

An agent-based simulation model of
Eldana saccharina Walker

Brian John van Vuuren



Dissertation presented for the degree of
Doctor of Philosophy
in the Faculty of Engineering at Stellenbosch University

Promoter: Dr. L Potgieter
Co-promoter: Prof. J Bekker

December 2016

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December , 2016

Abstract

An agent-based simulation model is formulated in this dissertation in order to simulate the population dynamics of *Eldana saccharina* Walker infestation in sugarcane. The simulation model comprises four distinct building blocks, namely a graphical user interface, the implementation of the life cycle and associated influence of temperature on *E. saccharina*, the mating process of *E. saccharina* and the oviposition and dispersal of *E. saccharina*. These building blocks are based on existing literature pertaining to the biology and behaviour of the pest and, in cases where the relevant literature is insufficient or non-existent, expert opinion and careful assumption. In order to select areas from existing sugarcane farms on which to execute location-specific experiments, functionality which allows *Geographic Information Systems* (GIS) data importation is included as a platform on which to run the simulation model. These data inform the model in respect of the shape and characteristics of the underlying sugarcane fields in which the simulated *E. saccharina* population interacts.

The model interacts and operates within an ANYLOGIC simulation software environment and, in so doing, aims to emulate the behaviour of a population of *E. saccharina* moths in sugarcane. It is anticipated that the model implementation may serve as a basis for facilitating future design and testing of control measures in order to suppress the pest and its consequent detrimental effect to sugarcane through infestation and feeding on interior stalk nutrients. Numerous working mathematical models of the pest exist in the literature, but, in all previous cases, intricate aspects of the stalk borer's biology have been aggregated on a population level and average population changes have been affected at discrete time steps. The resulting analyses therefore yield conclusions that do not necessarily reflect the continuous, changing nature of *E. saccharina* on a localised level. Using agent-based modelling, however, the pest's behaviour may be modelled in more detail so as to facilitate more thorough investigation of potential control strategies and their expected efficacy on the pest at different points in its life cycle.

The agent-based simulation model designed in this dissertation is subjected to a number of verification and validation techniques. Furthermore, a pilot sensitivity analysis is conducted to identify the most influential parameters in the simulation model. These parameters are then considered further in a comprehensive parameter variation analysis in order to illustrate the flexibility and diversity of the model in terms of the variety of scenarios pertaining to *E. saccharina* population behaviour that it can accommodate. In some cases, simplified implementations of control measures are also imposed on the pest within the model in order to further illustrate its implementation capabilities, as anticipated for future model development and use.

In light of this flexibility, the model is also presented as a computerised decision support and analysis tool, including the ability to upload and recreate a specific user's own sugarcane farm shapefile, as well as to alter a set of available parameters. This may aid in simulating specific behaviour in a simulation run in accordance with what has typically been observed by the user, or of hypothetical scenarios which require investigation. In turn, as the model is further

developed and detailed control measures are included as part of the simulation execution, it is believed that an appropriate response pertinent to the characteristics of the geographical area under consideration and the corresponding *E. saccharina* population present in this area may be predicted, allowing for control measure alteration and redesign so as to optimise the associated parameters or actions prior to in-field implementation.

In order to further refine the model and improve its accuracy, as well as ensure agreement between the existing modelling approaches and actual biological processes in nature, the entire simulation model of *E. saccharina* is subjected to an expert panel discussion. The experts comprising the panel encompass some of the key researchers pertaining to *E. saccharina* and other moth behaviour and population dynamics, both in South Africa and internationally. The simulation model is updated or adjusted according to suggestions made and new information shared by the expert panel in an attempt to simulate the pest as accurately as possible in accordance with the body of knowledge currently available.

Although several other approaches to modelling *E. saccharina* populations have been adopted in the past, no existing models implement such a low level of abstraction with respect to the biology of the pest. In addition, previous models are often case-specific, investigating specific control measures that are imposed on an aggregate level on a population of the pest. By actively simulating *E. saccharina*'s biological decision-making processes, intricate aspects pertaining to one or a number of interacting control strategies, as well as the manner in which they alter the pest's biology or behaviour, may easily be incorporated using an agent-based simulation modelling approach. Adopting a low level of abstraction also requires extensive information pertaining to the pest and, as such, areas where little understanding still exists with respect to the behaviour of *E. saccharina* have been highlighted and, consequently, may be prioritised for future entomological research by experts in the field. Finally, numerous options for future investigation into this problem, including model refinement, control measure design and testing and comparison to existing models, pose positive possibilities for the eventual establishment of a functional, integrated pest management programme for *E. saccharina*.

Uittreksel

'n Agent-gebaseerde simulasiemodel word in hierdie proefskrif geformuleer om die populasiedinamika van *Eldana saccharina* Walker infestering in suikerriet te modelleer. Die simulasiemodel bestaan uit vier verskillende boustene, naamlik 'n grafiese gebruikerskoppelvlak, die implementering van die lewensiklus en die gepaardgaande invloed van temperatuur op *E. saccharina*, die paringsproses, en die eierlegging en ruimtelike verspreiding van *E. saccharina*. Hierdie boustene is gebaseer op bestaande literatuur oor die biologie en gedrag van die pes en, in gevalle waar tersaaklike literatuur ontbreek, insette van kundiges en sorgvuldige aannames. Ten einde gebiede in bestaande suikerrietplase te kies waarin ligging-spesifieke eksperimente gedoen kan word, is funksionaliteit wat data-invoer vanuit Geografiese Inligtingstelsels (GIS) moontlik maak, ingesluit om as platform vir die simulasiemodel te dien. Hierdie data bedien die model met inligting rakende die vorm en eienskappe van die onderliggende suikerrietlande waarin die *E. saccharina* populasie interaktief is.

Die vier boustene van die model het op só 'n wyse interaksie met mekaar in 'n ANYLOGIC simulasiemodulering-omgewing dat die gedrag van 'n populasie *E. saccharina* motte in 'n suikerrietplantasie nageboots word. Daar word voorsien dat die modelimplementering die toekomstige ontwerp en toetsing van beheermaatreëls vir die onderdrukking van die pes en sy nadelige invloed op suikerriet deur infestering en tering op die interne voedingstowwe daarvan mag fasiliteer. Verskeie aanvaarde wiskundige modelle bestaan reeds vir die pes in die literatuur, maar in al hierdie gevalle word gespesialiseerde aspekte van die pes se biologie tot op populasievlak saamgevoeg en word gemiddelde populasieveranderinge by diskrete tydskritte geïmplementeer. Die gevolglike analyses lewer daarom gevolgtrekkings wat nie noodwendig die kontinue veranderende gedrag van *E. saccharina* op 'n lokale vlak goed saamvat nie. Met die gebruik van agent-gebaseerde simulasiemodulering, daarenteen, is dit moontlik om fyner besonderhede van die pes se gedrag met voldoende akkuraatheid te modelleer sodat dit moontlik word om die doeltreffendheid van potensiële pesbestrydingstrategieë op verskeie fases van die lewensiklus van *E. saccharina* deegliker te ondersoek.

Die agent-gebaseerde simulasiemodulering wat in hierdie proefskrif ontwerp word, word aan 'n aantal verifikasie- en valideringstegniese onderwerp. 'n Aanvanklike sensitiviteitsanalise word verder uitgevoer om die mees invloedryke parameters in die simulasiemodulering te identifiseer. Hierdie parameters word dan in 'n daaropvolgende, omvattende parameter variasie-analise ondersoek om sodoende die buigsaamheid en diversiteit van die model te demonstreer in terme van die verskeidenheid scenarios vir *E. saccharina* populasiegedrag wat daarin geakkommodeer kan word. In sommige gevalle word vereenvoudigde pesbestrydingsmaatreëls in die model ingebou om die implementeringmoontlikhede daarvan te illustreer.

In die lig van hierdie buigsaamheid word die model ook as 'n gerekenariseerde besluitsteun- en analisestelsel aangebied wat die vermoë insluit om die profiellêer van 'n spesifieke gebruiker se suikerrietplaas in te lees en te herskep. Die model laat ook toe dat 'n spesifieke stel parameters

varieer kan word. Dit laat toe dat spesifieke gedrag wat deur die gebruiker waargeneem is, gesimuleer kan word, asook hipotetiese scenarios wat ondersoek verg. Soos wat die model verder ontwikkel is, kan gedetailleerde beheermaatreëls as deel van die simulatie-uitvoering ingesluit word. Dit het tot gevolg dat die model die paslike reaksie lewer wat verband hou met die eienskappe van die geografiese gebied wat ondersoek word en die ooreenstemmende *E. saccharina* populasie in die gebied kan voorspel word. Dit laat verfyning en herontwerp van beheermaatreëls toe om geassosieerde parameters en aksies voor praktiese implementering te verfyn.

Ten einde die model verder te verfyn en die akkuraatheid daarvan te verbeter, en om ooreenstemming tussen modelleringbenaderings en werklike biologiese prosesse in die natuur te verseker word die volledige simulasiemodel onderwerp aan bespreking deur 'n paneel deskundiges. Hierdie deskundiges is van die voorste plaaslike en oorsese navorsers van *E. saccharina* en ander mot-gedrag. Die simulasiemodel word opdateer of aangepas volgens voorstelle wat gemaak is asook nuwe inligting wat deur die paneel gedeel is in 'n poging om die pes-gedrag so akkuraat as moontlik in ooreenstemming met huidige kennis te simuleer.

Alhoewel verskeie ander modelleringsbenaderings reeds in die verlede op populasies van *E. saccharina* toegepas is, bestaan deur geen model waarin daar só 'n lae vlak van abstraksie met betrekking tot die biologie van die pes bereik word nie. Verder was vorige modelle dikwels ook geval-spesifiek, in die sin dat spesifieke pesbestrydingsmaatreëls oorweeg is sonder dat toekomstige verbetering of uitbreiding van die modelle, of eksperimentering daarmee, dan op dieselfde vlak uitgevoer kon word soos wat in 'n agent-gebaseerde simulatiekonteks moontlik is nie. Aangesien die nastrewing van 'n lae vlak van modelabstraksie verder uitgebreide kennis oor die pes vereis, is dit ook moontlik om areas waar huidige kennis oor die gedrag van *E. saccharina* gebrekkig is, uit te wys. Hierdie areas mag vir toekomstige entomologiese navorsing deur kenners geprioritiseer word. Verskeie moontlikhede vir verdere toekomstige navorsing, insluitend modelverfyning, beheermaatreëlontwerp, en die toetsing en vergelyking van bestaande modelle, hou laastens ook moontlikhede in vir die uiteindelijke daarstelling van 'n funksionele, geïntegreerde pesbestrydingsprogram vir *E. saccharina*.

Acknowledgements

The author wishes to acknowledge the following people and institutions for their various contributions towards the completion of this work:

- The Almighty God who is the giver of every good gift. Every wonderful opportunity, unforgettable experience and lifelong friend I have been blessed by on this journey simply couldn't be by coincidence, and I know that He has crafted the path which I have been so privileged to walk.
- Dr L Potgieter, for her bravery in taking on a student from a different department with an entirely different academic background, and being willing to tirelessly impart all her knowledge pertaining to the problem which we tackled together. She always pushed me to do more and to do it better, whilst still being consistently supportive and understanding of my lifestyle, which is something I have valued ever so much. Her commitment and prioritisation of my work in spite of other professional and personal commitments is both admirable and hugely appreciated.
- Prof J Bekker for his wisdom, interest and, perhaps most of all, his wicked sense of humour. He came on board with this research at a late stage and excellently balanced the role of both being supportive and available, whilst still allowing me space to develop and mould my own project. He has also always treated me as an absolute equal — both in research and in a professional environment — and, for that, I am extremely grateful and look forward to working further as colleagues in the future.
- Prof JH van Vuuren, for being an absolute professional in his approach to all aspects of his work — particularly the lecture which first indicated to me the calibre of researcher that he is and resulted in this phenomenal, privileged position in which I spent the first two years of this research journey. He, too, has always had time to assist me in developing, not only as a young researcher, but also as a person and I am eternally grateful for both the friendship and guidance which he has afforded me during the completion of this work. It has been an honour to be associated with such a highly regarded researcher and I have learnt so much from the manner in which he engages with and respects his students as equals, always willing to go the extra mile for their benefit.
- Prof DE Conlong and the other staff at SASRI for always being open and available for questions pertaining to the problem at hand and showing a genuine interest in the research. Without their expertise and insight, this work would have never progressed as much as it did over the last two years.
- My friends of the Industrial Engineering Department and, in particular, the *Stellenbosch Unit for Operations Research and Engineering* (SUnORE) for their friendship and support

over the course of this research. The array of ‘disses’ and jokes which I have received about my work (and life in general) reassures me that they are all best friends and I will cherish the memories, tea times, laughter and other experiences we enjoyed together for many years to come. In particular, Dr MD Einhorn is thanked for introducing me to ANYLOGIC simulation software and assisting me in overcoming the steep learning curve pertaining to my complete lack of skill in the discipline of simulation modelling when beginning this project.

- Other friends and family who have supported me, distracted me (when needed) and shown a vested interest in my personal and academic development over the last two years. To those who have assisted me with part-time work and other financial support in a variety of forms, thank you for helping me make a success of this entire post-graduate study venture.
- My parents, Bridget and Les, for supporting me through the bold decision to take on my undergraduate study debt by terminating my bursary in order that I may pursue a postgraduate degree and showing interest along every step of the way — even when they had absolutely no idea what I was talking about. They are both shining examples to me and I respect and value every sacrifice they have made for me and the unwavering support and love which they have shown me in following my heart and forging my own path.
- My ‘second set’ of parents, Marilyn and Rob, for their continual support, advice and excitement for all that has come my way over the last few years. Your interest and investment of time, love and blessings has helped me immensely make a success of this adventure.
- My sister, Emma, for always remaining my biggest supporter and very best older sister — irrespective of the paths our lives have taken. She was my very first ‘teacher’ and, in many ways, this work is an expression of the love for learning, instilled in me by her.
- My other half, Courtney. She is surely the only other person who knows this simulation model as well as I do. She has been my rock and my sound board, my voice of reason and my most special person through this entire journey — It’s impossible to express how much you, as a person and companion, mean to me, but I will to spend the rest of my life trying anyway.

The Industrial Engineering Department is thanked for the use of its computing facilities and office space. The financial support of The *South African Sugarcane Research Institute* (SASRI) and the *HB and MJ Thom Trust* are hereby acknowledged and thanked for making this research possible. Any opinions, findings and conclusions or recommendations expressed in this dissertation are those of the author and do not necessarily reflect the views of Stellenbosch University, SASRI or the HB and MJ Thom Trust.

Table of Contents

Abstract	iii
Uittreksel	v
Acknowledgements	vii
List of Acronyms	xiii
List of Figures	xv
List of Tables	xix
1 Introduction	1
1.1 Background	1
1.2 Informal problem description	3
1.3 Dissertation scope and objectives	5
1.4 Research methodology	6
1.5 Dissertation organisation	7
2 Survey of biological literature	11
2.1 Sugarcane	11
2.2 <i>E. saccharina</i> on sugarcane	12
2.3 The life cycle of <i>E. saccharina</i>	13
2.4 The mating process of <i>E. saccharina</i>	15
2.5 The spatial movement of <i>E. saccharina</i>	16
2.6 The effect of temperature on <i>E. saccharina</i>	17
2.7 Existing control strategies for <i>E. saccharina</i>	19
2.8 An integrated pest management approach	20
2.9 Chapter summary	20

3	Computer simulation modelling	23
3.1	Simulation modelling	23
3.1.1	Types of simulation models	24
3.1.2	Simulation modelling approaches	25
3.1.3	Modelling components	26
3.2	Advantages and disadvantages of simulation	27
3.2.1	Advantages	28
3.2.2	Disadvantages	28
3.3	Steps in a typical simulation study	29
3.4	Simulation in the context of ecological systems	33
3.5	Chapter summary	34
4	An agent-based model of <i>E. saccharina</i>	37
4.1	The ANYLOGIC simulation modelling environment	38
4.2	General considerations of the proposed model	39
4.3	The graphical user interface (GUI)	40
4.3.1	General description of the GUI	40
4.3.2	Description of the constructed simulation environment in an ANYLOGIC modelling environment	43
4.3.3	Functionality assessment of the GUI	47
4.4	The life cycle of <i>E. saccharina</i>	52
4.4.1	General description of the implemented life cycle of <i>E. saccharina</i>	52
4.4.2	Description of the implemented life cycle in an ANYLOGIC modelling environment	54
4.4.3	Functionality assessment of the implemented life cycle	60
4.5	The mating process of <i>E. saccharina</i>	64
4.5.1	General description of the implemented mating process of <i>E. saccharina</i>	64
4.5.2	Description of the implemented mating process in an ANYLOGIC modelling environment	67
4.5.3	Verification of the implemented mating process functionality	73
4.6	The oviposition and dispersal of <i>E. saccharina</i>	75
4.6.1	General description of the implemented oviposition and distribution processes of <i>E. saccharina</i>	75
4.6.2	Description of the implemented oviposition and dispersal processes in an ANYLOGIC modelling environment	78
4.6.3	Verification of the implemented oviposition and distribution processes functionality	88

Table of Contents	xi
4.7 Chapter summary	92
5 Simulation model verification and validation	93
5.1 Model verification, output analysis and validation	93
5.2 Verification of the agent-based model of <i>E. saccharina</i>	94
5.3 Validation of the agent-based model of <i>E. saccharina</i>	96
5.3.1 Face validation	96
5.3.2 Comparison with video footage of <i>E. saccharina</i>	99
5.3.3 Parameter variation and sensitivity analysis	100
5.4 Chapter summary	107
6 Parameter variation analysis	109
6.1 The ScanRadius and Proximity parameters	110
6.2 The stage-specific random or predator mortality rates	119
6.3 The ‘rate’ of the Erlang distribution	123
6.4 The influence of the probability of scanning and displaying of agents	123
6.5 The influence of larvae dispersal among the sugarcane stalks	127
6.6 The oviposition site selection algorithms	132
6.6.1 General dispersal rates	132
6.6.2 The effect of the FlightAmbition parameter	142
6.6.3 Biased dispersal based on sugarcane maturity values	148
6.7 Chapter summary	152
7 A decision support and analysis tool	153
7.1 Motivation	153
7.2 Available parameters for variation in the decision support and analysis tool	155
7.3 Chapter summary	164
8 Model development through expert opinion	165
8.1 The Delphi technique	165
8.2 Expert panel discussion	167
8.2.1 Discussion topics	168
8.2.2 Simulation of the underlying sugarcane and larval infestation	169
8.2.3 Simulation of other notable elements	171
8.3 Changes made to the simulation model	172
8.3.1 The oviposition and dispersal process followed by female moths	172
8.3.2 Simulation of the underlying sugarcane and larval infestation	176

8.3.3	Other alterations made to the simulation model	180
8.3.4	Functionality assessment and comparison with previous performance . . .	182
8.4	Chapter summary	189
9	GIS map facilitation	191
9.1	Motivation	191
9.2	Importing shapefiles into ANYLOGIC	192
9.2.1	Description of the full sugarcane farm shape importation process	193
9.2.2	Description of the experiment site selection process	198
9.2.3	Description of the selected site importation into the model	198
9.3	Functionality assessment of the GIS importation process	208
9.4	Implementation of GIS features into simulation execution	210
9.4.1	Functional assessment of the sugarcane attribute attribution process . . .	210
9.4.2	Attributing specific characteristics of sugarcane stalks to each field	213
9.4.3	Incorporation of emitting and absorbing boundaries	216
9.4.4	Functionality assessment	219
9.4.5	Revised decision-support interface	228
9.5	Chapter summary	229
10	Conclusion	231
10.1	Dissertation summary	231
10.2	The contributions of this dissertation	233
10.3	Possible future work	235
10.3.1	Recommendations for future entomological research	236
10.3.2	Recommendations for future simulation modelling work	237
	References	241
A	Pilot sensitivity analysis data	247
B	Contents of the accompanying compact disc	265

List of Acronyms

AW-IPM: Area-wide integrated pest management system

IPM: Integrated pest management

SASRI: South African Sugarcane Research Institute

SIT: Sterile insect technique

GIS: Geographic information system

IRC: Interactive run controller

DD: Decimal degree

List of Figures

1.1	Sugar producing areas in South Africa	2
2.1	A sugarcane stem section	12
2.2	Furrows containing seed cane	12
2.3	Sugarcane of varying ages	13
2.4	Sugarcane of different ages and varieties	13
2.5	Life stages of <i>E. saccharina</i>	14
2.6	Borer related damage to a sugarcane stalk	15
3.1	Steps in a typical simulation study	31
4.1	General life cycle simulation framework	39
4.2	Screenshot of the GUI	41
4.3	Global parameters in the simulation	43
4.4	Underlying sugarcane layouts incorporated into the simulation model	47
4.5	Examples of hiding different simulation layers	48
4.6	Output of simulated sugarcane growth at typical temperatures	49
4.7	Output of simulated sugarcane growth at increased temperatures	49
4.8	Output of simulated sugarcane growth at decreased temperatures	50
4.9	Test scenarios for sugarcane maturity value setting	50
4.10	Flowchart of the general life cycle of <i>E. saccharina</i>	53
4.11	Life cycle statecharts division for female and male agents	55
4.12	Parameters for life cycle control of male and female agents	56
4.13	Variance in time taken to mature through the egg life stage	62
4.14	Variance in time taken to mature through the larvae outside stalk life stage	62
4.15	Variance in time taken to mature through the larvae inside stalk life stage	62
4.16	Variance in time taken to mature through the pupae life stage	63
4.17	Variance in time taken to mature through the adult life stage	63

4.18	Flowchart of the general mating process of <i>E. saccharina</i>	65
4.19	Inner working of the female mating behaviour control	67
4.20	Inner working of the male mating behaviour control	67
4.21	Inner working of the lek agent control	68
4.22	A mating sequence executed in the ANYLOGIC simulation	74
4.23	Flowchart of the general oviposition and dispersion processes of <i>E. saccharina</i>	76
4.24	Inner working of the female oviposition and distribution processes	79
4.25	Graphical representation of the oviposition site search process	82
4.26	Example of a typical collection of oviposition flights in a heading window	84
4.27	Graphical representation of the oviposition site search process	85
4.28	Example of a typical collection of oviposition flights in a moving window	86
4.29	Illustration of the FindMatureCane function executed in a simulation	89
4.30	Illustration of the MovingWindow function executed in a simulation	91
5.1	The verification and validation process of a simulation model	95
5.2	Comparison of simulation outputs to the idealised population growth curve	101
5.3	The starting conditions used in the sensitivity analysis experiments	103
6.1	Scan Radius set to 10m	111
6.2	Scan Radius set to 12.5m	111
6.3	Scan Radius set to 15m	112
6.4	Proximity set to 0.05m	112
6.5	Proximity set to 0.15m	113
6.6	Proximity set to 0.25m	113
6.7	Minima selected for mating	114
6.8	Maxima selected for mating	114
6.9	SIT release of 100 agents during first adult maturation	116
6.10	SIT release of 100 agents for different simulated conditions	117
6.11	The total number of non-surviving eggs oviposited	118
6.12	Four pesticide sprays of 60% potency released over 6 months	119
6.13	Two pesticide sprays of 60% potency released over 6 months	120
6.14	One pesticide spray of 60% potency released over 6 months	120
6.15	One pesticide spray of 40% potency released over 6 months	122
6.16	One pesticide spray of 80% potency released over 6 months	122
6.17	Lek creation and usage with increased display or search probabilities	125
6.18	Lek creation and usage with decreased display or search probabilities	126

6.19	Lek creation and usage with combined changing display and search probabilities	127
6.20	Sampled data set of <i>E. saccharina</i> infestation	128
6.21	Population growth of agents dispersing differing distances	129
6.22	Selected examples showing damage from typical spread of larvae	131
6.23	Selected examples showing damage from wider spread of larvae	132
6.24	Growth for parameter combinations using the ‘Moving Window’ algorithm	134
6.25	Dispersion using the ‘Moving Window’ algorithm with lower parameters	135
6.26	Dispersion using the ‘Moving Window’ algorithm with standard parameters . . .	136
6.27	Dispersion using the ‘Moving Window’ algorithm with higher parameters	137
6.28	Growth for parameter combinations using the ‘Find Mature Cane’ algorithm . .	138
6.29	Dispersion using the ‘Find Mature Cane’ algorithm with smaller parameters . . .	139
6.30	Dispersion using the ‘Find Mature Cane’ algorithm with medium parameters . .	140
6.31	Dispersion using the ‘Find Mature Cane’ algorithm with larger parameters . . .	141
6.32	Growth for each Erlang combination using the ‘Moving Window’ algorithm . . .	142
6.33	Growth for each Erlang combination using the ‘Find Mature Cane’ algorithm . .	143
6.34	Dispersion using the ‘Moving Window’ algorithm with a lower Erlang value . . .	144
6.35	Dispersion using the ‘Moving Window’ algorithm with a high Erlang value	145
6.36	Dispersion using the ‘Find Mature Cane’ algorithm with a lower Erlang value . .	146
6.37	Dispersion using the ‘Find Mature Cane’ algorithm with a higher Erlang value .	147
6.38	Dispersion for different sugarcane layouts using ‘Moving Window’ algorithm . . .	150
6.39	Dispersion for different sugarcane layouts using ‘Find Mature Cane’ algorithm . .	151
7.1	The display page seen upon model execution	155
7.2	The settings tab of the simulation model	156
8.1	The experts composing the panel during one of the discussion topics	167
8.2	Graphical interpretation of the revised search process during oviposition	173
8.3	Illustration of the FindOvipositionLocation function executed in a simulation . .	175
8.4	Examples of randomly generated soil profiles within the simulation space	179
8.5	Randomised physiological sugarcane age causing population extinction	183
8.6	Population dispersion with a very low probability of site selection	185
8.7	Population dispersion with a low probability of site selection	186
8.8	Population dispersion with a normal probability of site selection	187
8.9	Population dispersion with a high probability of site selection	188
9.1	The naming conventions used in the GIS importation process	193
9.2	Flowchart describing the importation process of the sugarcane farm shapes . . .	194

9.3	Required format of files for GIS importation	195
9.4	Flowchart describing the process followed to centre the polyline set	197
9.5	Flowchart describing the importation and point adjustment process of the selected polylines	200
9.6	Possible locations of points to be considered after importation	201
9.7	A hypothetical example of a polyline requiring adjustment	201
9.8	Possible positioning of reference field without any points falling in the presentation area	202
9.9	Duplication of polyline vertex points to allow for point splitting	203
9.10	Interpolation between points for correct adjustment	204
9.11	A hypothetical example illustrating the different inaccuracies dependant on plot- ting direction	206
9.12	Example of a case where points may be incorrectly adjusted by the adjustment algorithm	207
9.13	Incorrect adjustment resulting in inaccurately trimmed field shapes	207
9.14	The manner in which vertices were discovered	208
9.15	A typical GIS important process example	209
9.16	The resulting imported section of the GIS process	211
9.17	The resulting imported section of the GIS process	212
9.18	The manner in which characteristics are attributed to specific sugarcane fields . .	214
9.19	Assessment of the crop aging, colouring and harvesting process	215
9.20	Example of the absorbing and emitting boundary control	217
9.21	Details of the selected area upon which to execute a first illustrative example . .	220
9.22	First full model experimental testing	221
9.23	Screenshots of animation output upon first experiment conclusion	222
9.24	Details of the selected area upon which to execute a second illustrative example .	223
9.25	Second full model experimental testing	224
9.26	Screenshots of animation output upon second experiment conclusion	225
9.27	Details of the selected area upon which to execute an illustrative example	226
9.28	Third full model experimental testing	227
9.29	Screenshots of animation output upon third experiment conclusion	228
9.30	The revised simulation settings tab	229

List of Tables

2.1	Development thresholds for different life stages of <i>E. saccharina</i>	18
4.1	Different components employed in the simulation model	38
4.2	Results of tests done on cane maturity values	51
4.3	Probabilities of death for different life stages in the simulation model	61
4.4	Results of testing of sugarcane carrying capacity functionality	63
4.5	The remaining agents after a harvest event after one year	64
4.6	The remaining agents after a harvest event after two years	64
5.1	Default parameter values included in the sensitivity analysis	104
5.2	Aggregated damage indication data for the ‘Moving Window’ algorithm	105
5.3	Aggregated damage indication data for the ‘Find Mature Cane’ algorithm	105
5.4	Aggregated population data for the ‘Moving Window’ algorithm	106
5.5	Aggregated population data for the ‘Find Mature Cane’ algorithm	106
6.1	Damage indicators during testing of differing pesticide spray strategies	121
6.2	Damage indicators during testing of differing pesticide spray strengths	123
6.3	Damage indicators during testing of differing maximum dispersal distances	130
6.4	Damage indicators during testing of different oviposition algorithms	141
6.5	Damage indicators of different Erlang rates used in oviposition algorithms	148
6.6	Tally of agents residing in each area of different sugarcane maturities	152
A.1	Sensitivity analysis damage indicators using the ‘Moving Window’ algorithm	251
A.2	Sensitivity analysis damage indicators using the ‘Find Mature Cane’ algorithm	255
A.3	Sensitivity analysis population data using the ‘Moving Window’ algorithm	259
A.4	Sensitivity analysis population data using the ‘Find Mature Cane’ algorithm	263

CHAPTER 1

Introduction

Contents

1.1	Background	1
1.2	Informal problem description	3
1.3	Dissertation scope and objectives	5
1.4	Research methodology	6
1.5	Dissertation organisation	7

1.1 Background

Sugarcane cultivation in South Africa dates as far back as the early 1600s. In 1635, Portuguese explorers shipwrecked near the mouth of the Umzimkulu River, where they discovered that sugarcane was one of the crops grown by local inhabitants [80]. The sugar industry expanded following the establishment of the agricultural society in 1848 and, today, the South African sugar industry generates an estimated average direct annual income of R12 billion [81]. The industry is, thus, one of the world's leading cost competitive producers of high-quality sugar and makes a significant contribution to sustainable development, the South African national economy and employment, particularly in rural areas. Approximately one million people, or more than 2% of South Africa's population, depend on the sugar industry for a living.

The majority of the approximately 428 000 hectares of sugarcane grown by about 22 500 registered sugarcane growers in South Africa today are located in the province of KwaZulu-Natal. A substantial investment also exists in Mpumalanga, with some farming operations also occurring in the Eastern Cape. Fourteen sugar mills operate in these sugarcane growing regions, split among six milling companies who manufacture sugar in the country (see Figure 1.1). About 60% of this sugar is marketed in the South African Customs Union (SACU) whilst the balance is exported to markets in Africa, Asia and the Middle East [82].

The stalk borer, *Eldana saccharina* Walker, was first recorded in 1939 as a pest infiltrating sugarcane when a severe infestation occurred in the Umfolozi area. This infestation, however, did not spread and eventually died out in 1950 [32]. It then reappeared in the Hluhluwe area during the early 1970s [17] and has since spread to much of the local producing areas, but for inland crop areas where the insect is limited by the cooler temperature. Evidence suggests that *E. saccharina* infests sugarcane by virtue of its suitability for egg-laying in dead leaf material [4].

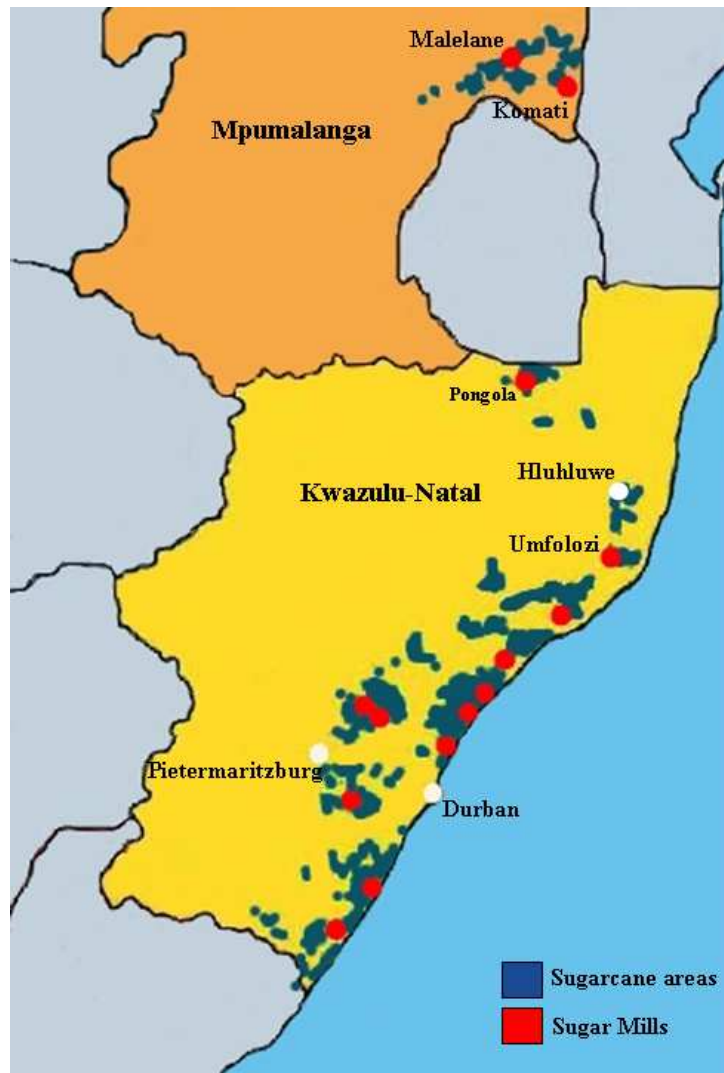


FIGURE 1.1: The sugar producing areas, as well as the location of the sugar mills, in South Africa [81].

This infestation results in decreased sugarcane quality (measured as a decrease in sucrose yield) and has a negative impact on total plant biomass [27].

In light of this, *E. saccharina* remains a topical concern in the sugar industry and the *South African Sugarcane Research Institute* (SASRI) continues to spearhead research efforts towards finding a means of effectively controlling the pest. A number of pest control measures currently exist, achieving only limited success in most cases. These include sugarcane variety development [23, 46], chemical control [32], biological control, habitat management [20, 19, 23] and, most recently, the *sterile insect technique* (SIT) [66].

SIT is a birth control method implemented by the release of a large number of sterile insects as part of SASRI's *area-wide integrated pest management programme* (AW-IPM). IPM systems aim to combine a series of control methods in order to achieve better overall pest management whilst decreasing the use of pesticides, thereby decreasing associated environmental problems. As a result, IPM is considered more sustainable in the long run. Ideally, a number of existing techniques should be applied in conjunction with one another to a sugarcane crop in order to achieve improved control of *E. saccharina*.

When applied together, research shows that the existing *E. saccharina* control methods can

reinforce one another [20]. In view of this knowledge, it is important to identify the shortcomings in the individual control methods, as well as the role of possible interactions between them. Determining these intricacies of either individual or interacting control methods by in-field testing is both costly and labour-intensive, and there is not always sufficient time available to conduct large-scale in-field tests in order to observe these effects. Moreover, a means for practical evaluation of the relative impact of interacting control strategies before they are actually applied to infested sugarcane is lacking.

1.2 Informal problem description

One identified means of attempting to address difficulties associated with the design and implementation of control measures to combat *E. saccharina* in sugarcane is that of mathematical and simulation modelling. Mathematical and simulation models of the pest which mimic their real-life behaviour and interaction may be further expanded, modelling imposition of IPM systems on the pest. This would, in turn, facilitate testing of a variety of different scenarios and parameters pertaining to control measures, affording some measure of confidence in the anticipated success of the venture before a large amount of time and money is spent conducting in-field implementation.

A number of working simulations have been proposed in the literature for evaluating the estimated effect of particular pest control methods imposed on sugarcane fields.

The first work of this nature was performed by Van Coller [86] and Hearne *et al.* [31] who employed a system of differential equations to model the change in population growth in the various stages of the life cycle of *E. saccharina*. This model primarily provided insight into the biological control of the pest through parasitoids, but did not explicitly take into account the spatial spread of an *E. saccharina* population. More specifically, the model facilitated the testing of policies for the timing, frequency and magnitude of parasitoid releases in terms of their relative effectiveness in the biological control of *E. saccharina*.

Horton *et al.* [35] later designed a model which was aimed at investigating the effects of insecticides and early cane cutting as control measures for *E. saccharina*. This model also explicitly included the effects of temperature on the pest's life cycle, but again assumed a homogeneous spread of the population over the spatial habitat. The function of this model was to produce an index measuring the extent of sugarcane damage under different temperature patterns in order to assist in determining an ideal time for harvesting.

Most recently, Potgieter *et al.* [65] developed a discretised reaction-diffusion model of the growth and spread of an *E. saccharina* population over time and space under the influence of the *sterile insect technique* (SIT). An accompanying SIT simulation tool was also developed whereby the effectiveness of the SIT can be investigated in different scenarios. Notably, the model and accompanying tool can be applied in the context of heterogeneous spatial domains — including realistic sugarcane field layouts by importing *geographic information system* (GIS) maps. Although the current design of the SIT simulation tool includes only the SIT as control measure, a framework for a computerised platform was developed which may be extended to a decision support system for use in an area-wide IPM programme.

A high-level design of such a simulation tool for investigating different IPM strategies against a number of pest species in sugarcane was presented by Potgieter [66], consisting of four interacting subsystems that simulate the population dynamics of different pest species, sugarcane growth, environmental dynamics and the cost of IPM strategies, respectively. One of the limitations

of the population dynamics model for use in an IPM context which was highlighted in the study is the difficulties surrounding the inclusion of interacting pest control strategy in the model. Possible improvements to the simulation tool were suggested, including adopting an agent-based modelling approach for the pest species population dynamics subsystem.

Despite the advancement in understanding of *E. saccharina* population growth and the relative success achieved by the aforementioned models, the design of each model is founded upon approximations of the pest on a population level. Local interactions of individual moths are not simulated or incorporated explicitly into the population dynamics; instead, approximate changes are executed in simulations at discrete time steps. The model results therefore do not necessarily reflect the continuous, changing nature of *E. saccharina* on a localized level. Furthermore, the models all focus on single control measures, limiting their development and flexibility in the context of IPM systems.

In an attempt to address the shortcomings of the existing simulation models described above, an approach is advocated in this dissertation where the individual members of a population of *E. saccharina* are simulated, thereby incorporating the effects of local interactions between individual stalk borers.

Agent-based modelling is the computational study of social agents interacting in an autonomous manner as evolving systems. It allows for the study of complex adaptive systems and facilitates investigations into how macro-phenomena develop from micro-level behaviour among heterogeneous sets of interacting agents [42]. By simulating *E. saccharina* moths as individual agents who are governed by their biological preferences and limitations, the resulting relationships between these agents can be used to predict population dynamics of the pest in a more realistic manner over space and time (based on local interactions) than is possible in the existing *E. saccharina* models.

It is anticipated that an autonomous, agent-based simulation model of *E. saccharina* may serve three primary functions. First, the model may be used to validate and support existing high-level averaging models of the stalk borer, such as those previously mentioned. This may assist in developing more advanced, accurate mathematical models for further understanding and prediction of the spreading of the pest. Secondly, by attempting to model the pest on such an individualised, detailed level, several areas are likely to be uncovered where a lack of understanding of the pest exists. In light of this, an agent-based model will require a number of important assumptions, incorporate the notion of randomness of behaviour and accommodate alternative manners for modelling certain aspects of the pest where entomological research cannot currently provide conclusive insight. This is anticipated to also assist in guiding the in-field research conducted on *E. saccharina* by uncovering areas where little or no knowledge of the pest exists. Finally, a working model of *E. saccharina* may provide a platform facilitating testing and evaluation of various pest control mechanisms, either alone or in combination, before expensive in-field testing is pursued.

A further important aspect of a computerised agent-based simulation model of the population dynamics of *E. saccharina* is the possibility of equipping the model with a visually intuitive interface facilitating the interpretation of its outputs by parties who do not have the necessary training required to understand complex mathematical models. Such a simulation model of *E. saccharina* can incorporate easily configurable variable factors affecting the spatial spread and temporal growth of the pest population. Users of the model can then investigate the effects of minor changes to the system and use these observed effects to predict the potential success of *E. saccharina* control measures in practice. This directly addresses a current shortcoming in the sugarcane industry where confidence levels associated with the expected outcomes of implementing proposed *E. saccharina* control methods, based on the predictions of existing

simulation models, are difficult to quantify or attribute to particular cases. Often, further experiments and methodology have to be developed simply to verify whether a mathematical model or control method will operate as expected. A realistic agent-based model, however, can assist in providing a simple, effective verification mechanism which, in turn, may increase confidence levels and creativity in the design of new *E. saccharina* control methods and IPM systems alike.

Such an agent-based simulation model of the population dynamics of *E. saccharina* is proposed, implemented and tested in this dissertation.

1.3 Dissertation scope and objectives

In order to facilitate the testing and optimisation of proposed pest control strategies for *E. saccharina* in an attempt to support the design of an AW-IPM system, a detailed, accurate simulation model of *E. saccharina* behaviour in sugarcane is required. In an attempt to provide a possible platform for such testing, the following objectives are pursued in this dissertation:

1. To *perform* a comprehensive survey of the literature in the various fields pertaining to this study, including:
 - (a) a biological background of the sugarcane plant,
 - (b) a detailed insight into the life cycle and associated activity of *E. saccharina*,
 - (c) other pest-orientated mathematical and simulation models or associated predictive software, and
 - (d) various computer simulation modelling techniques and approaches.
2. To *design* a robust simulation framework for investigating the population dynamics of *E. saccharina* moths interacting in a typical sugarcane environment with the highest possible degree of accuracy.

This requires:

 - (a) an underlying simulated environment within which the simulated agents may interact,
 - (b) an accurate implementation of the life cycle of the pest as well as the associated influence of temperature,
 - (c) a detailed, accurate implementation of the mating cycle of *E. saccharina*, and
 - (d) a comprehensive incorporation of the oviposition and consequent dispersal habits of *E. saccharina* post mating.
3. To *implement* the agent-based simulation model described in Objective 2 above in a user-friendly manner which is both intuitive and whose results are easy to interpret by its intended user.
4. To *validate* the simulation model of Objective 3 above using a variety of available techniques — particularly in the presence of a lack of high-quality, comprehensive experimental data.
5. To *facilitate* testing of various scenarios within the boundaries and scope of user-specified parameters in the form of sensitivity analyses, as well as to *shed light* on areas of the pest's biology where insufficient research has been performed. This should include a dedicated parameter setting functionality made available to the user upon model execution.

6. To further *develop* the model and increase its accuracy by subjecting it to a panel discussion whereby entomologists and in-field experts may scrutinize, refine and endorse model elements to ensure they best reflect the understood real-life process. Any suggested changes should then be incorporated into the simulation model.
7. To *incorporate* the importation of GIS shapefiles so that the simulation model may be executed on specific areas of real-life sugarcane farms, allowing for geographic features of the underlying sugarcane landscape to be better represented in the model.
8. To *suggest* possible future improvements or additions which may be included in the model, as well as other auxiliary studies which may stem from the work reported in this dissertation.

Where possible, the model proposed in this dissertation aims to include all small-scale complexities of the behaviour of *E. saccharina* in the most accurate manner possible. In areas where insufficient biological research exists to quantify or fully describe attributes or activities, sensible assumptions are made, based on insight from industry experts. Furthermore, where appropriate, certain aspects of *E. saccharina* behaviour are excluded or simplified for the sake of time and computational requirements.

This study forms part of a larger, ongoing body of research performed at Stellenbosch University, in association with SASRI. The simulation model proposed in this dissertation marks a departure point for computer-based simulation of an agent-based nature for *E. saccharina* control. For this reason, not all aspects of the model can be considered or included, and some aspects of a complete model are reserved for future work. Furthermore, owing to widespread uncertainty pertaining to many aspects of the biology and behaviour of *E. saccharina*, various aspects of the model are developed based on assumptions suggested by and insights gained from industry experts. The model proposed in this dissertation therefore aims to serve as a tool assisting in the ongoing investigation of the pest and, as such, facilitates evaluation of a number of possible scenario executions, as opposed to adopting a fixed set of parameter values and studying consequent behaviour.

The detailed implementation of an actual control strategy of *E. saccharina* falls outside the scope of this project. The model is, however, constructed so as to allow for easy, intuitive further development and improvement in order to eventually include and facilitate such functionality.

1.4 Research methodology

The methodological procedure followed in this dissertation to design and develop an agent-based model for investigating the population dynamics of *E. saccharina*, as described in §1.3, is as follows:

1. *Consult* the existing literature, as well as expert opinion, in respect of the biology and behaviour of *E. saccharina*, in order to formulate a comprehensive understanding of available knowledge pertaining to the life cycle and associated activity of the pest in sugarcane.
2. *Develop* a proficiency in agent-based modelling within an ANYLOGIC simulation software environment — particularly using a statechart-based modelling approach.
3. Iteratively *construct* an agent-based model of the well-defined aspects of the *E. saccharina* life-cycle and associated activities.

4. *Verify* the performance and execution of the model in respect of each new addition, both by animation output or, where necessary, expert opinion.
5. Further *develop* the model by making sensible assumptions, based on insight from similar studies and expert opinion, in modelling aspects of *E. saccharina* where little or no surety exists pertaining to the biological processes in question.
6. *Perform* an initial face validation, with the assistance of an industry expert, on the agent-based simulation model in order to determine the degree to which it reproduces the population dynamics of *E. saccharina* as observed in nature.
7. *Incorporate* any suggestions or changes proposed in light of the face validation mentioned above.
8. *Perform* other available means of validation on the simulation model, including video footage analysis and sensitivity analysis.
9. Using the results of the sensitivity analysis, *incorporate* the most notable parameters as user-changeable inputs in the simulation model in the form of a dedicated ‘settings’ tab, so as to facilitate scenario testing and ‘what-if’ analyses.
10. To *scrutinize* the entire model in detail, with the assistance of an expert panel as well as suggest further additions which may be made.
11. Based on input from the panel discussion, *improve* the simulation model by incorporating suggested changes to incorrectly implemented or insufficient model elements, as well as adding aspects which were previously excluded for lack of insight or understanding.
12. *Extend* the simulation model to facilitate the importation of GIS shapefiles so as to allow for simulation experiments to be executed within specific sugarcane farm field areas.

The simulation model construction and underlying assumptions are formally described in this dissertation. This includes a detailed description and analysis of its performance and the subsequent influence of notable model parameters. This report aims to facilitate a comprehensive understanding of the finer details of the agent-based simulation model so as to equip future research efforts which may stem from this study.

1.5 Dissertation organisation

Apart from this introductory chapter, this dissertation contains nine additional chapters. Chapter 2 provides the reader with an understanding of sugarcane growth in South Africa, the life cycle and important associated behavioural aspects of *E. saccharina* Walker in sugarcane, as well as existing control strategies aimed at combating the pest. This covers the relevant, existing biological information necessary for understanding the model assumptions made and approaches taken in the model construction.

The aim in Chapter 3 is to provide the reader with information pertaining to the different aspects and considerations of computer simulation modelling. The chapter contains a discussion on the different types of simulation models, as well as important modelling concepts, advantages and disadvantages, and describes the steps followed in a typical simulation study. The chapter then closes with a review of computer simulation modelling in an ecological context, citing various

appropriate software packages and existing examples of agent-based modelling applied to similar ecological problems.

In Chapter 4, the four fundamental building blocks composing the agent-based simulation model of *E. saccharina* proposed in this thesis are described in detail. The chapter opens with a short motivation of the software chosen for the simulation model development and the relevant aspects thereof. Following this, the manner in which the graphical user interface, life cycle and associated influence of temperature, mating cycle, and oviposition and dispersal cycle of *E. saccharina* are emulated and constructed in the model is described in detail. This includes a discussion on the parameters, functions and other notable model aspects. The description of each building block closes with a verification of its operation against the literature and assumptions upon which its implementation is based in order to determine the success with which it emulates the desired process.

Chapter 5 contains a description of the verification and validation processes followed to ensure that the interaction and functionality of all of the building blocks operating together as one simulation model functions correctly, and hence that an appropriate simulation model has been built. In this chapter, the difficulties surrounding the validation of such a simulation model, for which little or no accurate data exist, are discussed. In light of this, a face validation, brief video validation and sensitivity analysis are conducted in Chapter 5.

Based on the outcomes of the validation processes employed in Chapter 5, a parameter variation analysis is conducted in Chapter 6 in order to document the flexibility, functionality and diverse ability of the simulation model. In this chapter, various scenarios are emulated using the different parameter capabilities and the influence of changing each parameter on the model outcome is noted. Furthermore, a number of simplified experiments incorporating simulated control measures are conducted in this chapter in order to illustrate the manner in which the simulation model facilitates such testing.

Chapter 7 contains a step-by-step description of the agent-based simulation model of *E. saccharina* proposed in this dissertation, cast in the form of a decision-support tool which may be used in the industry to provide insight, conduct ‘what-if’ analyses and emulate a variety of possible scenarios pertaining to the biology and behaviour *E. saccharina*. Each of the parameters which exist as user-specified inclusions in the model are briefly discussed and the default value attributed to each of these parameters upon model execution is documented. Furthermore, a short description of the effect of increasing or decreasing each parameter is included for easy reference and understanding by model users.

In Chapter 8, details are given of an expert panel discussion conducted in an effort to refine and improve the model. The different discussions that took place are recorded and the input and suggestions from the panel are described. Based on the outcomes of these discussions, changes implemented in the simulation model are also explained, concluding with an assessment of the updated and newly implemented features of the simulation model through a number of experiments. These experiments are then discussed and, where relevant, compared to the outputs of the original model in an attempt to determine the degree to which the improvements have been implemented successfully.

Chapter 9 contains an in-depth description of the novel process designed to accommodate importation of sugarcane farm shapefiles into the model. This includes the conversion process from shapefiles into GIS coordinates, as well as the formatting, importation and presentation of these coordinates to the user. Furthermore, the user interface is detailed, including the process whereby a simulation experiment site may be selected by the user in order to perform location-specific simulation runs. The chapter also details further considerations which were required to

be implemented in light of the incorporation of GIS functionality, as well as some illustrative experimental results so as to demonstrate the functionality of the GIS features in the simulation model.

Finally, a short summary of the dissertation is presented in Chapter 10, together with an overview of the contributions made. The chapter closes with suggestions in respect of future work, pertaining both to entomology and simulation, which may improve or build on the work conducted in this study.

CHAPTER 2

Survey of biological literature

Contents

2.1	Sugarcane	11
2.2	<i>E. saccharina</i> on sugarcane	12
2.3	The life cycle of <i>E. saccharina</i>	13
2.4	The mating process of <i>E. saccharina</i>	15
2.5	The spatial movement of <i>E. saccharina</i>	16
2.6	The effect of temperature on <i>E. saccharina</i>	17
2.7	Existing control strategies for <i>E. saccharina</i>	19
2.8	An integrated pest management approach	20
2.9	Chapter summary	20

A short review on sugarcane in South Africa is given in this chapter, with a focus on the biology of the plant itself. This is followed by a discussion on the biology of *E. saccharina*, its infestation and growth in sugarcane, and its population dynamics. The material in this chapter provides a sufficient biological foundation for understanding the simulation model formulation and its underlying assumptions which will follow in later chapters.

2.1 Sugarcane

Sugarcane (*Saccharum officinarum* L.) is a tall perennial grass which forms part of the *Gramineae* family. This grass forms lateral shoots at the base which produce multiple stems, typically 3m high and 5cm in diameter. The stems then grow into cane stalks whose waxy interior contain a juice from which sucrose, used in commercial sugar production, is sourced [36]. Fully grown sugarcane stalks are up to 1.5m long and are approximately 7cm wide. They are divided by distinct nodes into internodes (see Figure 2.1) from which long, narrow leaves grow along the entire length of the stalk [66]. At the top of the plant there exists an apical bud which is responsible for elongation of the stalk. This bud eventually develops into a plume-like flower cluster [36, 66]. At each node in the stalk there is a bud sometimes also known as an ‘eye’. Unlike most plant types, the seeds produced by sugarcane are used to develop new varieties, rather than the planting of fresh crops. Instead, sections of sugarcane stalk are laid lengthwise in furrows and new plants stem from the buds located on the stalk nodes (see Figure 2.2) [84].

Sugarcane is best suited for growth in a tropical climate in fertile, moist soil [95]. During its various life stages, the plant becomes susceptible to a variety of diseases and infestations by pest species which can damage the crop and result in decreased yield for farmers [66].

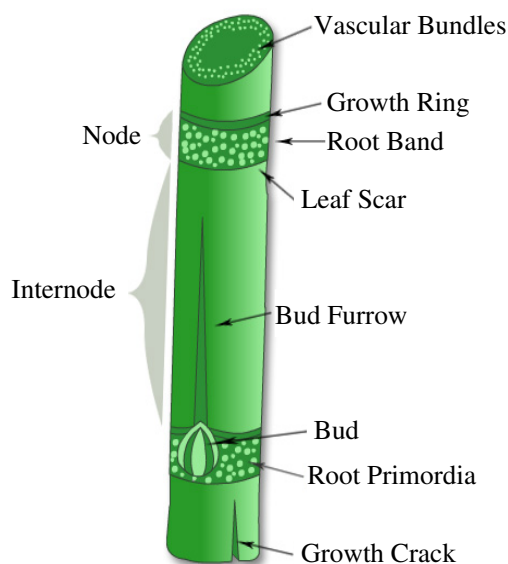


FIGURE 2.1: A sugarcane stem section [58].



FIGURE 2.2: Furrows with seed cane [84].

Growing periods differ owing to a variety of agroclimatic factors. In South Africa, maturation time varies from 12–24 months, depending on proximity to the coast. Coastal sugarcane usually matures within 12 months, whilst inland the process takes approximately 22 months [84]. Figure 2.3–2.4 show the growth of typical coastal sugarcane through a variety of ages in Kwa-Zulu Natal. Sugarcane mills are closed between the months of December through to March and, as such, farmers must manage their crops so as to ensure a maximum profit over the season. The option exists to leave some sugarcane unharvested during mill closure, known as carry-over cane, which is then harvested as very mature cane at the beginning of the new season. Mature sugarcane is, however, very vulnerable to infestation by pest species such as *E. saccharina* and, as a consequence, often delivers a reduced sucrose yield. For this reason, sugarcane is often cut early to avoid infestation losses, but the less mature sugarcane has a lower sucrose content, resulting in a loss of potential production which may have been achieved had the sugarcane been allowed to mature fully and remained un-infested [66]. Harvesting begins with burning and the cane is cut shortly thereafter [36].

2.2 *E. saccharina* on sugarcane

E. saccharina is an indigenous African stalk borer pest which is widely distributed in sub-Saharan Africa. The pest occurs naturally in wetland sedges and indigenous grasses in areas that experience warm temperatures and typically high rainfall [2, 66]. In tropical regions, the pest is renowned for attacking maize and sorghum, but, owing to the fact that it generally attacks mature crops of this type, it is not of economical concern [6].

In Southern Africa, the pest was first found to be present in sugarcane in the Umfolozi area from 1939 until approximately 1950. The pest appeared to infest sugarcane originally by virtue of its preference for dead-leaf material for oviposition, rather than it being a more nutritious



(a) Approximately 6 month old sugarcane.



(b) Sugarcane of more than 20 months old.

FIGURE 2.3: Sugarcane of varying ages [84].



FIGURE 2.4: A sugarcane field in Kwa-Zulu Natal consisting of cane of different ages and varieties [84].

host than its natural plant hosts such as Cyperaceae¹ [2, 4]. It re-appeared in the Hluhluwe area in 1970 [17], whereafter it became a widespread pest of sugarcane in Natal, Swaziland and the then Eastern Transvaal [6]. Since the 1970s, an increase in the use of nitrogen and potassium fertilizers in crop production has greatly increased the nutritious quality of sugarcane crops and has made it even more attractive to the stalk borer [6, 9].

2.3 The life cycle of *E. saccharina*

The life-cycle of *E. saccharina* is typical of that of insects and is depicted in Figure 2.5. Eggs hatch into larvae, after which pupation occurs and, finally, adult moths emerge [6]. The amount of time spent in each stage of the life cycle largely depends on both the quality of food, as well as the temperature experienced [65].

Although a female moth typically lays approximately 450 eggs post-mating, not all of these eggs are deposited at a single oviposition site. Instead, females usually search for various suitable sites, depositing batches of approximately 20 eggs at a time. Once the eggs have been oviposited, they typically hatch after 8–10 days. Larvae will then mature over the course of about 20 days in

¹Cyperaceae is a flowering plant which resembles rushes, examples of which include papyrus (*Cyperus papyrus*) and common basket grass (*Cyperus textilis*) [4].



FIGURE 2.5: Life stages of *E. saccharina* including (from top left to bottom right) eggs, larvae, pupae and fully grown moths [66].

summer or 60 days in winter. The immature larvae do not immediately penetrate the sugarcane stalk, but begin by feeding on the leaves of the sugarcane plant or other shards of organic matter. Once the larvae are sufficiently robust, they select a suitable host plant within their vicinity and bore into the sugarcane stalk where they remain as borers for the duration of their immature life. In the stalks, the larvae move backwards and forwards with equal ease; hollowing out large portions of the plant through consumption as they do so. During this feeding process, frass is pushed from the inner stalk through holes to the exterior of the plant, thus providing evidence of borer activity within the stalk. As many as 12 larvae may occupy a single internode in the stalk and all parts of the stalk may be infested. The lower region is, however, generally more heavily infested owing to its increased maturity and subsequently favourable mineral content [17]. This better quality food source is understood to contribute to maturation into stronger adult moths who are more competitive in terms of flight ability and flight speeds in the case of female moths, and superior mating displays and pheromone strengths in the case of males [19]. Once a larva is mature, it spins a protective cocoon and pupates therein. The pupae are either located within the hollow of the stalk, or balanced on its outside beneath a leaf sheath. Adult moths emerge from the pupae after approximately 10 days and have a lifespan of 5–7 days [17].

Owing to variation during the different stages of development time, generations of *E. saccharina* completely overlap, with all stages of the life cycle present at all times [66]. *E. saccharina* breeds continuously throughout the year, although particular life stages dominate each season, giving rise to seasonal cycles. Moth populations peak in early summer and again in autumn, whilst larval populations peak during winter, with a small peak also occurring late during the summer [7].

Mortality of the pest occurs haphazardly during the life cycle and natural enemies are not considered a notable factor in mortality [66]. Milling is considered the largest killer of the pest during the annual milling season beginning in May [7].



FIGURE 2.6: Damage to sugarcane stalks caused by larval boring in the stem of the plant [61].

2.4 The mating process of *E. saccharina*

The complex mating process followed by *E. saccharina* has been documented thoroughly by Atkinson [5].

Mating of *E. saccharina* occurs after sunset and is weather-dependent. Males typically emerge from pupae slightly before females and climb up the stalk of the plant to its canopy. Females are witnessed far more rarely on the plant canopy than males. Rain and even light winds inhibit mating and moths have been witnessed to wait beneath leaves in the canopy until the conditions become favourable for mating. During the first 10–15 minutes after emergence, the male's wings are pressed together over its back. Thereafter, once they are 'set,' they are folded back into the usual resting position. Approximately 30 minutes after the wings have set, the male begins to display.

The male's display consists of rapid wing flapping in bursts, accompanied by pencil hairs extending from the abdomen into a round brush. During this time, the male also releases an attractant pheromone into the surrounding air. This aphrodisiac secretion produced by the hair pencils on the male's abdomen as a long range attractant to females, the chemical structure of which has been analysed by Zagatti *et al* [50]. The aim of the display is to attract females for mating

purposes. Males remain static for the duration of the display which, in the absence of females, can continue through until dawn. Each moth can typically only maintain such behaviour over 15–20 minute periods, whereafter they sometimes move a short distance before restarting the process.

Although males can display singly, they far more frequently display in groups of 3–6 males, called *leks*, on the same plant. The plant on which they congregate is often one of the taller ones in a collection. Mating between moths usually occurs on the first night of emergence, after which females begin oviposition for the remainder of their lives.

During the mating process, a female approaches a lek of males upon being attracted by their display and the scent of the released pheromone. Once she has located the lek, one male is selected to mate with, after which the pair are typically not disturbed by any other moths in the near vicinity. It is generally understood that the most dominant male (that is, the male with the strongest pheromone or display) is selected by the female [19]. In a laboratory experiment, however, a female dropped into a box containing many males, some of which were displaying, appeared to walk and fly about until she encountered a displaying male, after which mating would ensue. This random means of selection has been attributed to the box being saturated with the odour of the male pheromone. It is believed that selection in nature is a more competitive process [5]. Kindl [48] also discussed the role of ultrasonic communication employed during courtship between species, confirming that male acoustic signals typically initiate the mating process in the associated *Aphomia sociella* moth species. The ultrasound signals are believed to also assist in deferring potential predation by moths [44].

Almost all females mate only once during their lifetimes owing to the large size and durability of the spermatophore, as well as the relatively short lifespan of the adult moth [17]. Walton [89] has, however, reported that, under laboratory conditions, females have been noted to mate up to three times. In the same study, males were shown to have the ability to mate up to six times, but more often seemed to do so approximately three times over their lifespan. The mating process lasts up to three hours. When disturbed during mating, female moths have been observed dragging their partners to alternative locations. This suggests that the pairs may seek shelter lower down the plant since they are more vulnerable to predation by spiders when on the plant canopy and, particularly, whilst mating [5].

About 24 hours after mating, a female will begin oviposition in preferred sites such as dead-leaf sheaths and areas between the plant and soil. As mentioned in §2.3, a female typically lays approximately 450 eggs in batches of approximately 20 per oviposition site [17].

2.5 The spatial movement of *E. saccharina*

Soon after a batch of *E. saccharina* eggs hatch from their oviposition sites, the neonatal larvae begin to disperse and do not bore into the stalk at the site of the hatching. Conlong [19] describes a process whereby larvae move away from other neonates by moving up the stalk at the oviposition site to the top of the cane plant. From here, larvae ‘parachute’ from the green leaf tips to surrounding plants. They move down the plant on which they land and bore into the more mature bottom part of new stalk, through cracks in the rind of the stalk, or through buds at each internode. This results in early dispersion within the local area surrounding the site where eggs were oviposited. The larvae then remain on their particular stalks until they become adults capable of walking or, at times, flying across the leaf canopy.

E. saccharina appears to fly more frequently during nights of light wind, possibly owing to the fact that mating and displaying is inhibited in the prevalence of wind [5]. Other than that, the pest is considered to be a relatively weak flier, but research has shown that female moths, being larger and stronger than males, have the potential to disperse over considerable distances [66]. It is suggested that female moths which disperse do so because they remain unmated by the end of their first or second night and are, thus, more likely to travel than those moths which have mated [5]. This would equate to approximately one third of female moths being involved in dispersal flights at the beginning and end of summer. A major reason for mated females flying is the search for oviposition sites. Although eggs are usually laid close to the adult's emergence site, some females fly up to 200m or more before ovipositing [17].

Existing studies related to other moth species indicate that mobile and sedentary genotypes exist in populations which display dispersal capacity in the field [76]. These studies also indicate that younger, unmated moths show greater capacity to travel long distances. The observations of Atkinson and Carnegie [8] corroborate these findings.

Berry *et al.* [13] have found that *E. saccharina* shows a distinct preference for infesting stalks with a high number of total internodes (in general, 16 internodes or more) as these stalks show higher growth potential and, in turn, are more nutritious and contain higher levels of nitrogen. The higher infestation levels in such stalks found in the study may have been by chance, or may be attributed to the female and/or her offspring possessing the ability to select taller potential stalks. In either case, the emergence of subsequent larvae will tend to be greater owing to the more nutritious food source which will, in turn, increase the density of the population in those areas. A study by Way [91] also found the developmental period of *E. saccharina* to decrease in a nitrogen-rich diet.

Conlong *et al.* [22] suggested that the preference for development in the vicinity of older cane is influenced by the greater availability of hidden egg-laying sites in dead-leaf material, or specific volatiles in dead-leaf material which attract the egg-laying female towards the aged sugarcane. Their study, however, confirmed that although more eggs were laid in quarters containing older plants, quantities of dead-leaf material did not influence female oviposition preference. This suggests the possibility that females generally leave their larvae to find the most suitable host plants for themselves.

Plant volatiles do, however, allow females to sense when oviposition niches are occupied and deter them from ovipositing in those areas which, in turn, aids in the dispersion of the pest throughout a sugarcane field [19].

2.6 The effect of temperature on *E. saccharina*

The rate of development of *E. saccharina* is strongly linked to the temperature that the moth experiences during its different life stages. Way [91] conducted a controlled laboratory experiment to determine the thermal constants and developmental thresholds for *E. saccharina*. Using a scale of *degree-days* ($^{\circ}\text{D}$), he documented the mean developmental periods for each life stage above a certain calculated lower threshold temperature. Degree-days indicate the total amount of exposure to heat each individual requires in order to develop sufficiently during each life stage. It is calculated as the integral of a function of time which varies with temperature and, as such, accumulates the pest's exposure to each average daily temperature, continually summing the temperatures until they exceed a pre-determined threshold and, in doing so, cause the pest to progress to its next life stage. The temperature used in the day-degree calculation is the difference between the actual temperature and a certain lower threshold or, if appropriate, an

upper threshold. This experiment was performed at several different temperatures, the results of which are summarised in Table 4.2. The threshold temperatures determined for the egg, larvae and pupae stages were 5.3 °C, 10.2 °C and 10.7 °C, respectively [49].

Temperature (°C)	Egg thermal constant (°D)	Larvae thermal constant (°D)	Pupae thermal constant (°D)
13	118	—	—
15	119	610	160.4
20	120	640	183.6
23	111	598	148.8
25	116	645	144.2
30	115	600	152.0
35	133	—	172.8
Average	119.0	618.6	160.3
STD	7.2	20.0	14.5

TABLE 2.1: Development thresholds for different life stages of *E. saccharina* [91].

Development occurs at an increased rate in warmer temperatures, such as those which are experienced in summer, as opposed to cooler temperatures more synonymous with winter months. This observation has been affirmed by Carnegie [17].

Temperature also plays an important role in the frequency of mating of the pest. In winter, mating is usually confined to the first 2–3 hours of the night. The temperature threshold of both sexes appears to be around 15°C. Below this temperature, the pest usually disappears from the stalk canopy on which it displays and mates [5].

Way [91] also found that, theoretically, *E. saccharina* could develop throughout the year in all sugarcane regions within South Africa, including the Natal Midlands, since the estimated thresholds at which the development occurs are above the average winter temperatures. One exception would be, very occasionally, in the Midlands where winter temperatures may dip below these thresholds. The study did note that, based on the observations of Price [68], the thresholds may be over-estimations of the actual values since the development rate is sigmoidal².

Little development of the pest occurs below 15°C and, as such, is considered to be the lower bound on maturation. Way [91] also confirmed that development has been recorded to slow significantly as tests approached 35°C, suggesting the upper threshold temperature for development may be near this figure.

In light of this, infestations are currently confined to warmer areas where sugarcane is farmed, such as the Natal coast where the July mean daily isotherm is about 16°C [6], as opposed to the Midlands which experience colder temperatures in the absence of the warm sea breeze. Increased concern is, however, being raised as daily mean temperatures in the country continue to rise, further increasing the possibility of a breakout of the pest in the previously un-infested Midlands [19].

²In the shape of an ‘S’, in that the independent variable begins to rise slowly before rising sharply and then flattening out again at the end.

2.7 Existing control strategies for *E. saccharina*

In an effort to combat the loss in revenue resulting from *E. saccharina* infestation, SASRI expends large amounts of effort into effectively managing the pest. Early control efforts were synonymous with that of typical crop maintenance and work on insecticides began in the mid 1970s. These were applied to the crop at all stages of growth and progress was recorded to further refine the insecticide treatment process. Immersion of the seedcane in insecticide solutions for 30 minutes was found to be effective in reducing the numbers of larvae, but approximately 85% of mortality was attributed to water alone and the effect of the insecticides was deemed relatively small [32]. Insecticides were only investigated intensively again from 2003, following the failure of biological control [56, 72]

In a consequent effort to increase sustainability and shift away from the environmental problems induced by pesticides, more sustainable means of control have since been developed. The most prevalent control means currently tested at SASRI include habitat management, biological control and, more recently, the *sterile insect technique* (SIT) [66].

Research suggests that certain cultural practices in farming may favour or disadvantage *E. saccharina* infestation. These include the use of more resistant sugarcane varieties, pre-trashing of dry leaf material in an effort to reduce preferred oviposition sites, and earlier harvesting since *E. saccharina* prefers sugarcane of an advanced age [18].

The sugarcane plant has many attributes which can attract or repel insects. *E. saccharina*, in particular, display a hierarchical preference in selecting a host plant habitat for ovipositing [23]. Research has shown that *E. saccharina* has a preference above sugarcane for plants from the Cyperaceae family for use as oviposition sites. Conversely, *Melinis minutiflora* Beauv., the indigenous African grass, attracts natural parasites of *E. saccharina* in the presence of sugarcane and, in doing so, repels the pest from inhabiting a particular area [23]. This has informed efforts to introduce so-called ‘push’ and ‘pull’ plants into sugarcane field layouts in order to repel the pest away from infesting the cane or attracting it to a preferred oviposition site, respectively [19].

Several natural enemies of *E. saccharina* exist in nature, most of which attack the borer whilst still in the larval stage [21]. The potential of deploying selected parasitoid species into sugarcane fields remains feasible, although knowledge of the role of a natural enemy in their habitat is of importance in order to successfully introduce a chosen enemy to a particular environment and understand its effects on host population density [20]. One possible hindrance pertaining this method of pest suppression lies in the fact that *E. saccharina* larvae generally do not feed on the green, vegetative parts of the sugarcane stalk. In light of this, herbivore-induced plant volatiles may not be released by the infested plants, resulting in predators and parasitoids not being attracted to the site of infestation [72].

SIT is a form of birth control used by entomologists whereby a large number of laboratory-reared sterile insects are released into a native population to mate with fertile organisms in order to prevent them from reproducing. Optimal release strategies are in the process of being tested. SIT is recognised as an effective method of suppression, eradication, containment and prevention of pest species [65].

Finally, the potential of the implementation of *Bacillus thuringiensis* (Bt) crops remains feasible as a means of combatting *E. saccharina* infestation. This method of engineering host plants to produce insecticidal toxins has already been proven to effectively control other species such as the Cotton Bollworm [93]. A proof of concept for the use of Bt sugarcane to suppress *E. saccharina* has already been achieved and further efforts are currently underway at SASRI to develop and produce an effective strand of Bt sugarcane for large-scale implementation in the industry [72].

2.8 An integrated pest management approach

Integrated pest management (IPM) systems combine a series of control methods to achieve better pest management whilst decreasing the use of pesticides. When instituted together, research has shown that existing methods for controlling *E. saccharina* may endorse one another [23]. In light of this it is important to identify the shortcomings in these individual control methods, as well as the role that pairwise interaction between them could play. These questions are best answered by physical testing and experimentation. There are, however, not always enough time and funding available to conduct large-scale in-field tests to observe these effects. Each of the existing individual strategies is under continual development and improvement, both through university research and at SASRI itself [19]. These further research efforts aim to overcome the existing shortcomings in the current, individual control strategies.

According to Conlong and Rutherford [23], using the information and research generated on more conventional control options and combining it with sound, ecologically based, new technologies, SASRI has developed an *area-wide integrated pest management system* (AW-IPM) in a further effort to combat *E. saccharina*. This AW-IPM focuses on, amongst others, delineation of within-species populations³, chemical ecology, habitat management, SIT and the utilisation of plant endophytic pathogens⁴ to produce a workable integrated control strategy.

The finer details of habitat management are currently being investigated by means of in-field tests at SASRI [19]. In particular, information is sought on what quantity of ‘push’ and ‘pull’ plants should be included among sugarcane crops, what type of plants can be most successfully implemented using this control method and whether or not the solution can viably impose a notable change in the concentration of *E. saccharina* in sugarcane.

Similarly, the intricate details of SIT are still being developed in a bid to achieve optimal release strategies of irradiated moths and to determine what quantity of sterile male moths is required to affect an existing swarm of fertile *E. saccharina* significantly. The effects of actual moth dispersal, when compared with pure random walk models, are also being studied [19, 66].

Working simulations have been developed to review the estimated effect of control solutions imposed on sugarcane fields; particularly that of an SIT simulation tool [66]. This simulation tool is based on a mathematical model derived as a result of a series of assumptions and simplifications. Such simulations are invaluable in saving time and money so as to ensure that the most viable option is investigated and found before implementing an IPM in a real sugarcane field environment.

2.9 Chapter summary

An overview of sugarcane in South Africa was given in this chapter, together with an introduction to the stalk borer, *E. saccharina*, and a description of its propensity to infest sugarcane. A detailed biological background of the pest was then presented. The *lek mating* process followed by *E. saccharina* was discussed and the entire mating cycle was described. Thereafter, information on the spatial movement of the pest in its adult phase was provided. This was followed by a description of the effect of temperature on the development of *E. saccharina* through its different life stages.

³Adaptation of information from closely related species such as translocation of natural enemies in order combat genetically similar pests.

⁴Inhabiting plants with micro-organisms, such as fungi and bacteria, which do not harm the plant itself, but can generate toxins which kill infesting species.

Existing control strategies proposed to suppress *E. saccharina* were then discussed, including pesticides, habitat management, biological control and SIT. An introduction to the notion of an integrated pest management system which may be used to control the pest effectively was finally provided in which the current state of development of the various control strategies, as well as the state of the current IPM in industry, was discussed.

CHAPTER 3

Computer simulation modelling

Contents

3.1	Simulation modelling	23
3.1.1	<i>Types of simulation models</i>	24
3.1.2	<i>Simulation modelling approaches</i>	25
3.1.3	<i>Modelling components</i>	26
3.2	Advantages and disadvantages of simulation	27
3.2.1	<i>Advantages</i>	28
3.2.2	<i>Disadvantages</i>	28
3.3	Steps in a typical simulation study	29
3.4	Simulation in the context of ecological systems	33
3.5	Chapter summary	34

In this chapter, the science of simulation modelling is described and the most important elements which comprise a simulation model are introduced. Furthermore, the different types of simulation which exist are briefly discussed, as well as the advantages and disadvantages of using simulation modelling as an approach to solving problems. The steps followed in a typical simulation study are then considered. Finally, simulation in the context of ecological systems is discussed, including a number of commercially available ecological simulation software packages. Examples of agent-based simulation modelling in ecological contexts is also discussed.

3.1 Simulation modelling

Simulation may be defined as the imitation of the operation of an existing or potential real-world process or system over time. It is used to describe and analyse the behaviour of a system and aid in the design or management of consequent physical systems [11]. Simulation allows for the generation of an artificial history of a system, and manufactured data may be collected from the simulation and used as if it emanated from the real system being studied, facilitating investigation and analysis of the performance of that system [25]. Since both real and conceptual systems may be modelled using simulation, it is an important planning and problem solving tool in numerous real-world application areas [11].

Simulation modelling affords an analyst the opportunity to optimise or investigate a system prior to implementation and is particularly useful in instances where experimentation with the

real system is impossible or expensive. According to Borschev and Filipov [16], simulation modelling differs from analytical modelling in that the results of an analytic (or static) model depend functionally on the system input in the form of a number of parameters. Analytical solutions do not always exist and may prove difficult to find. In such instances, simulation (or dynamic) modelling can be applied [54]. A simulation model is, effectively, a set of rules in the form of equations, flowcharts, state machines and other constructs that define how the modelled system will progress over time, given its current state. The state of a system is defined to be the collection of variables necessary to fully describe the system at a point in time [54]. Simulation is the process of model execution which progresses the model through a number of state changes over time.

Before developing a simulation model, it is important to understand the problem and desired outcome of the simulation. A variety of aspects regarding the system must be analysed prior to the model development. These include the availability of information pertaining to the system being modelled and the ease with which relevant data required to substantiate the model may be obtained. If data do not exist, the possibility of data collection, as well as the associated processes required for the collection thereof, must be considered [16]. Fundamental system characteristics must also be distinguished from minor, perhaps negligible system characteristics by the incorporation of assumptions and mathematical relationships which decrease model complexity.

The different levels of model abstraction which exist are a strategic level, an operational level and a physical level [94]. Selecting the correct level of abstraction and, consequently, detail of the model is very important as it directly influences the success with which a system can be effectively modelled and simulated.

Strategic-level simulations are intended to identify and analyse organisational issues assuming a high level of aggregation of objects being modelled [94]. The required amount of detail for this level of modelling is typically lower than that of operational or physical-level modelling. Examples of strategic models include marketplace and competition models, as well as models of general ecosystem development and population dynamics on a large scale [16].

A model requiring more detail than that of a strategic model lends itself to being designed on an operational level. Models on an operational level are typically used to make tactical decisions regarding the system being replicated and, thus, more accurate data or details are required to substantiate the system in a more reliable manner. Models such as those investigating the distribution of staff or inventory levels within a company, as well as modelling of supply chains and the analysis of particular performance measures all fall within the bounds of an operational-level simulation [94].

Finally, the most detailed level of abstraction is modelling at a physical level. Such a simulation is used to observe and analyse individual behaviour of different components of a system and reliable, correct information is required to validate their movements, actions and timings. Biological and ecological models are typically formulated on this level, as are models of transportation networks and passenger flow analyses — all of which employ microscopic behaviour traits to analyse the resulting macroscopic developments of a system [94].

3.1.1 Types of simulation models

Simulation models may be classified along three different dimensions. These are *static vs. dynamic*, *deterministic vs. stochastic* and *continuous vs. discrete* models [54].

A static simulation model (or analytical model) represents the state of a system at a particular

time-step, or can be used to represent a system in which time plays no role. As a result, the model is functionally dependent on the input parameters specified by the user [16]. Dynamic models, on the other hand, evolve over time and develop from their initial state by virtue of simulation functions which are established by the programmer.

Deterministic models makes use of model input parameters which are known values, as opposed to stochastic models which incorporate inputs containing some sort of randomness associated with the input parameters of the system. As a result, the outputs produced by such models exhibit some degree of uncertainty and must be duly considered as estimates or approximations of the true characteristics of the model.

Continuous systems change state continually with respect to time. For example, the velocity and position of a person running is either continually changing, or has the potential to change continually over the duration of the time period during which they run. In contrast, the queue at a shopping check-out, for example, only changes state at quantifiable time-steps when someone is either serviced by the teller, or joins the back of the queue. Such systems are called discrete systems and, in this respect, discrete models are appropriate for systems whose state changes occur instantaneously at countable points in time.

Simulation typically attempts to model real life systems and, as such, it is very seldom that the system can be classified as exclusively discrete or continuous. Law [53] suggests that it is neither feasible, nor necessary to have a one-to-one correspondence between every element of a system and every element of a model. In light of this, systems are regularly modelled using a combination of both discrete and continuous modelling approaches. In such cases, model behaviour is simulated by computing the state variables at small time steps, whilst the values of attributes of entities and global variables of the system are determined at discrete event times [11].

3.1.2 Simulation modelling approaches

Simulation modelling offers a variety of different approaches which may be used to replicate a real-world system. Choosing the correct approach is important to ensure that the model to be built represents the intended system as accurately and realistically as possible. In light of this, the level of abstraction, as well as the details dictated by the underlying problem, are governing factors in this selection process. There exist four primary modelling approaches in simulation modelling, namely discrete-event modelling, system dynamics modelling, dynamic systems modelling and agent-based modelling [94].

Each of the modelling approaches discussed below lends itself to specific levels of detail and abstraction, but it is not uncommon to follow an integrated approach whereby principles or, indeed, elements of the different approaches are employed to achieve a desired result [16].

Discrete-event modelling

In discrete-event modelling, states of a system are defined and the model progresses according to a series of activities which move it from state to state. The beginning and end of each activity are known as *events* and the aim of such a simulation is to gain insight into the manner in which entities interact at these events. This then informs the system's impending dynamic behaviour. The state of the model remains constant between consecutive events, but the dynamic working of the system evolves as the simulation time advances through these specific events [11].

System dynamics modelling

System dynamics modelling is used primarily in long-term, strategic models to observe effects of the interaction of certain structures and influencing factors on the system as a whole. It assumes a high level of aggregation and, as such, objects being modelled lose their individual properties, histories or dynamics. In system dynamics modelling, processes are typically implemented using feedback structures such as stock-flow diagrams and decision rules which govern the simulation commodities and their respective quantities or dynamics in the model [94].

Dynamic systems modelling

In contrast to system dynamics modelling, dynamic systems modelling employs integrated variables which have direct physical attributes (such as location, velocity or concentration) which are inherently continuous and are not considered as aggregates of a pool of entities. A dynamic systems model is reliant on a number of state variables which are governed by algebraic or differential equations of various forms. Dynamic systems are typically capable of much greater mathematical diversity and complexity than system dynamics modelling and tools have been developed to be embedded in the engineering design cycle and utilise graphic modelling computer languages [16, 94].

Agent-based modelling

As mentioned in §1.2, agent-based modelling is the computational study of social agents interacting in an autonomous manner as evolving systems and, as such, allows for the study of complex adaptive systems. This form of modelling facilitates investigations into how macro-phenomena emerge from micro-level behaviour among heterogeneous sets of interacting agents [42]. Essentially, agent-based modelling is a decentralised, individualistic-centred approach to model design [94] where the global system as a whole is not defined, but rather the behaviour of the individual entities, known as agents, is defined by means of a statechart.

From the interactions of these agents, the global behaviour of the system emerges and, for this reason, agent-based modelling is referred to as *bottom-up* modelling [16]. Examples of agents in such a simulation are people, animals, insects, vehicles, departments or companies [94].

3.1.3 Modelling components

Models encapsulate a number of different components which formulate their structure. The diversity of components included in a model varies according to the requirements of the system being developed. A typical model has a defined *system*, *system state variables*, *entities* and their individual *attributes*, *activities* and *events* which prompt changes in state variables. These elements as identified by Banks [11] are discussed in some detail in this section.

A system is the real life process or activity represented by a model. It is important to define the limits and boundaries of the system from the outset of the model creation in order to ensure that resulting model accurately represents the intended system. The model should strike a balance between being complex enough to answer the research questions raised, but not too complex for practical use. System definition is therefore one of the cardinal starting points in composing a simulation model.

System state variables are the collection of all the temporal information required to define what is happening within a system at a given point in time in sufficient detail. The system state variables are determined as a function of the purposes of the investigation. This means that the same system may make use of different state variables depending on the nature of the real-world process which requires modelling. A distinction between discrete-event and continuous models can be made on the definition of their system state variables. Discrete-event system state variables only change at defined points in time, known as *event times*, whilst system state variables in continuous models are typically defined by algebraic, differential or difference equations, causing them to change continuously over the period of the simulation.

An entity is an object of interest within a simulation which requires explicit definition. Entities can be dynamic in nature (e.g. moving through the system as the simulation progresses) or, alternatively, static in nature (e.g. fulfil a role of service to other entities). Furthermore, an entity may have attributes which pertain exclusively to it, considered to be local variables. These attributes characterise the entity entirely, although many entities can have the same or similar characteristics.

An activity is a process incorporated into a simulation whose duration is known prior to its execution during the simulation run. For this reason, when the activity begins, the end can be scheduled and further simulation action can be incorporated with certainty based upon completion of the activity. Such durations can be fixed or constant inputs to the model, random values drawn from theoretical or empirical probability distributions, equation results, data file inputs or based on the simulation state. A delay, unlike an activity, is a model component which has an indefinite duration and is the result of a combination of system conditions or events. In discrete-event simulation, activities cause time to advance, whilst delays arise when entities have to wait for some service or action. The commencement and termination of activities and delays occur as a result of events.

Events are occurrences which alter the state of the system being modelled. They are typically instantaneous occurrences in the simulation. Events can be internal or external in nature, known as *endogenous* and *exogenous*, respectively. *Endogenous* events occur within the system, such as a customer who was waiting in a queue but starts to receive service, since it is already within the system being simulated. *Exogenous* events, on the other hand, occur in the external environment to the simulation model, such as a customer joining a queue to await service. Although this may fall outside the simulation scope, the arrival of the customer proceeds to impinge on the system and, as a consequence, must be taken into consideration.

3.2 Advantages and disadvantages of simulation

As software and technological capabilities in the computer industry continue to increase rapidly, so competition in this sector intensifies at a rapid rate — offering hardware that is faster, more powerful and easier to master. Similarly, the simulation-software industry has exploited these advancements as a springboard for developing programs which allow for the construction of specific, detailed models with quicker completion times and improved accuracy. This has diversified and commercialised the simulation industry, resulting in a wider variety of applications. As a result, many companies now use this technology to reap the benefits that it offers for studying complex systems [11, 53].

The advantages of simulation are vast and have been discussed in detail by many authors such as Banks [11], Law [53] and Robinson [71], all of whom have significant experience in the field.

3.2.1 Advantages

The most fundamental advantage of simulation is that it can be used to test and investigate a potential system which does not yet exist. This allows for the optimal development and testing of the efficiency of a facility prior to its construction and implementation, affording the consideration, control or removal of various factors which are likely to introduce difficulties and deficiencies into the real-world system at a later stage.

In the same vein, systems which do currently exist, but require upgrading, optimisation or the addition of a new component often incur a considerable cost to the owner. Using simulation, it is possible to test a variety of new policies and strategies without physically committing any resources to the real-world implementation, whilst still completely controlling the system execution and its variables. This means that modifications or additions can be made and their impact analysed without the typical costs and disruptions, as well as iterations, necessary for their implementation in the real-world system.

Another important benefit of simulation is its ability to expand and contract time. This allows for lengthy activities spanning days or even weeks to be executed and their effect observed in mere minutes. Conversely, time may also be slowed down to analyse events which occur momentarily in real-world situations and, thus, are too brief to accurately investigate in a physical environment. Having such control over the model time may assist in acquiring a deep understanding of the practical operation of a system, as well as providing insight into the causes for the occurrence of certain phenomena.

By using simulation, insight is also gained into the interaction of important variables and entities which compose and dictate a system. This provides information on the interdependencies of the variables and how they co-exist within the simulation environment. In much the same way, problem diagnosis may also be simplified by the ability to analyse individual system traits and the effects of singular variables on a problem situation. Adapting this approach, a bottleneck analysis can be performed efficiently using simulation, and effective methods can be identified for alleviating bottlenecks in a system.

A particularly important trait of a simulation model is its ability to facilitate a visualisation of the underlying real-world system. Most software packages allow for accompanying animation with the use of simulation models which allows for effective insight into how the system operates and looks. This characteristic is especially important when proposing a system to a client who is not necessarily technically qualified, since visualisation of the system is inherently intuitive and makes communicating and describing the model to the client simpler.

Finally, simulation is a valuable tool which may contribute towards building consensus among decision makers with respect to a proposed policy or system. Simulation results obtained from a model which have been repeatedly tested, validated and analysed may be visually represented to the concerned parties to aid in the decision making process. In doing so, risk is minimised compared to simply relying on speculation, personal experience or opinions which may arise from proposed designs or concepts.

3.2.2 Disadvantages

As with any technology or approach to solving a problem, using simulation also has a number of disadvantages which may deter a potential user. A few of these disadvantages of adopting a simulation modelling approach, as detailed in the literature, are described in this section.

Despite the variety of uses of an accurate simulation model, a fair amount of expertise and finesse

is required to actually construct these models. Such skills are learned over time and through experience and specialised training. This also means that it is very much an individualistic task, since two people constructing the same model, although similar, will almost never develop identical models. In light of this, the expertise of the model developer must be well revered and respected. Lately, however, this drawback has been offset somewhat by the introduction of software packages with model frameworks and partially complete models which allows less skilled users simply to complete the model or input the relevant data in order to test their respective system.

Owing to the fact that inputs to a simulation are often random, it is sometimes difficult to ascertain whether the outputs from the model are a result of system interactions, or simply as a result of the input stochasticity. It is therefore important to have readily available, reliable input data to use in respect of the real-world system, together with an expected result if possible, so as to allow for calibration of the model and to ensure that it is operating in line with expectations. To this effect, many software packages provide output analysis functionalities and tools to assist in the result interpretation procedure.

An important drawback is that, although using a simulation model to test scenarios can be less expensive than the disruptions associated with real-world tests, the construction of an accurate model is a time-consuming process and commercially available simulation software, particularly software intended for more sophisticated and professional use, is often quite expensive. In light of this and the current rapid advances in technology, many software vendors release a variety of product versions, ranging from basic to professional, thereby reducing the capital expenses associated with implementation of the software where a complete professional package is unnecessary.

Finally, simulation can also be used inappropriately. In some cases, the formulation of an analytical solution is possible for a problem — perhaps even preferable — but simulation is instinctively turned to as a solution method. This has the potential to result in substandard, inaccurate models being developed which provide ‘cherry-picked’ solutions to a user who understand a problem sufficiently thoroughly and, thus, provides input data or scenarios which result in a distinctly favourable output for his model, rather than a true reflection of a typical output for that system.

On the other hand, simulations are also prone to the inclusion of a large number of input variables, often including a measure of randomness. Although the resulting model output may accurately represent the desired system, models are often intensively interconnected and complex, which restricts the ability to perform effective analysis in terms of parameter sensitivity analysis or variation effects. This results from an inability to identify the parameter or parameter set whose interaction drives a certain change in the model. Such models of complex systems also typically require a large computational cost. This also detracts from the time-related advantage of simulation modelling.

That said, there are very few instances in the real world where the problem is simple enough to make use of an exclusively analytical solution and so detailed simulation is almost always a valuable addition to an investigation or project.

3.3 Steps in a typical simulation study

In order to construct and execute a successful simulation model, certain steps need to be completed and, at times, repeated to ensure that each aspect of the model is correctly implemented

and emulated. Banks [11] proposed a framework outlining a suitable process to follow based on the works of Law and Kelton [54] and Banks *et al.* [10]. These steps are summarised graphically in Figure 3.1, and elaborated upon in this section.

1. *Problem formulation*: The starting point of any simulation is, unquestionably, the formulation of a problem statement. This must come directly from the clients or persons initiating the development of the simulation model and, if these persons are not the model builders themselves, the success of the venture depends fundamentally on a clear description of what is hoped to be achieved. This problem statement must account for the overall objectives of the study and the specific questions which must be answered. This will guide the model designer as to the detail and, consequently, the approach which should be followed in the model's construction.
2. *Project planning*: This step is, essentially, a 'project proposal' wherein the scope of the project and the various scenarios which will be investigated are stated. Furthermore, information about the performance measures which will be used to measure the efficiency of the various system configurations must also be included. The timeline of the study must also be proposed, as well as the various outputs anticipated after each stage of the project. Finally, the hardware and software requirements of the project must be wholly specified.
3. *Model conceptualisation*: At this stage, the real-world system under investigation (if applicable) is abstracted to a conceptual model by determining the mathematical and logical relationships which govern the model components and structure the system. Good models only extract *essential* features of a problem which characterise a system and then elaborate and iteratively modify these features until the model becomes a useful approximation of the respective real-world system. The number of features identified for model inclusion is directly proportional to the resulting model detail and, when deciding on these aspects, care should be taken to ensure that the aforementioned project objectives and performances can be met and that relevant data are available to serve as inputs to the system. Furthermore, the performance capability of the computer on which the simulation will be implemented should also be considered so as to ensure that an adequate balance between complexity and usability is achieved.
4. *Data collection*: Any model is only as good as the quality of the data which are used as input conditions to simulate scenarios. In the model construction process, both the building of the model and data collection should occur concurrently in order to ensure, at each stage, that relevant data are available for implementation at certain points in the simulation so that the predetermined objectives can be achieved. Changes in the complexity of the model will also introduce the need for more or different input data and this should be considered when iteratively developing the model. Apart from input information, sufficient data for system performance validation must, where possible, also be accessed in order to compare the system output with typical output data of the real-world system.
5. *Model translation*: This step involves converting the conceptual model which was developed into an operational model in a computer-recognisable form, using a suitable programming language or simulation software package. The use of common programming languages provides greater control to the user in terms of translating the model into executable code, but the use of modern specialised software packages greatly reduces programming time and often increases model efficiency.

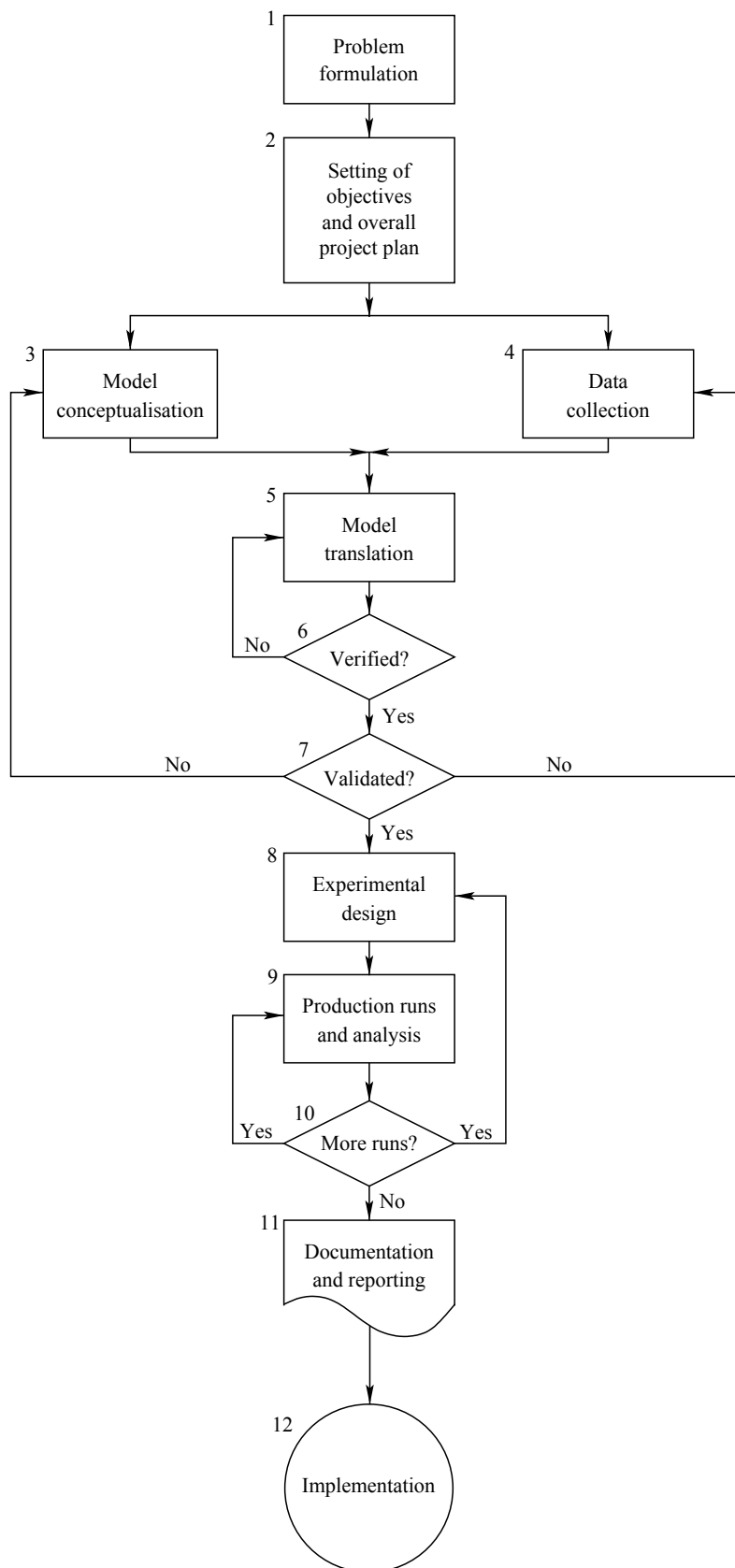


FIGURE 3.1: Steps in a typical simulation study [11].

6. *Model verification*: Verification concerns the implementation of the conceptual simulation model using the constructed computer model in order to assess whether or not the computer program is performing correctly. Ideally, verification should be a continual process throughout model construction to ensure that each new facet of the simulation operates as expected and contributes to the overall success of the model in its entirety. Debugging the entire model once complete is a difficult process and various interdependencies between variables are likely to be encountered, further complicating the process. Most simulation software suites provide efficient debuggers which aid in finding errors in models and interactive run controllers can also be used by the programmer to establish the success with which all commands are executed and, in doing so, verify the model's performance.
7. *Model validation*: Validation of a model is the process of determining whether that model is an adequately accurate representation of the real system. This is achieved through thorough comparison between the simulation model and subsequent real-world system, noting discrepancies between the two. Then, using data, case studies and expert opinions on the subject matter, the accuracy of the model is improved upon until the discrepancies are no longer relevant, or deemed to be at an acceptable standard.
8. *Experimental design*: Once the model is deemed to be an acceptable replica of the real-world system, the different system designs or scenarios which require investigation must be finalised. In each case, the length of the simulation run, the number of runs or replications and the starting conditions of the system must be decided upon and noted.
9. *Production runs and analysis*: The aforementioned production runs must then be executed and analysed in order to generate performance measures for each alternative scenario. This will allow for comparison between the different configurations at a later stage to either determine the most suitable or efficient configuration, or to gain a better understanding of the variety of possible situations which a real-world implementation of the system may encounter.
10. *Further production runs*: Based on the outcome of the initial production runs in the previous step, further production runs may be deemed necessary. In this case, the existing experiments which require further analysis must be re-run or, alternatively, the modification of existing experiments or, design of new ones, should be undertaken.
11. *Documentation and reporting*: Two types of documentation are necessary when conducting a simulation study, namely documentation detailing the program used to implement the simulation and model intricacies, and documentation of the results of the simulation study. In the event that another analyst wants to use the program to perform further testing, or another programmer wishes to pick up the model and develop or change it further, it is important that explanatory documentation accompanies the model explaining how it operates. This will decrease time and effort in understanding the model, as well as ensure that the model is operated properly and generates similar outputs as intended by the original designer. Documenting the results of the simulation study is obviously necessary to provide the initiator of the simulation exercise with a final answer or report on the original problem statement. That said, details of the progress of the model, including information on model specifications, prototype demonstrations, model animations and intermittent progress reports, are also important in order to provide a holistic presentation of the simulation study to the concerned parties.
12. *Implementation*: The final step of implementing the suggested process or configuration investigated by the study is inherently a combination of the success of the previous steps in

the process. The suitability and success of the model, as well as the quality and conviction with which the final report is presented will both play a crucial role in the decision maker's further steps.

3.4 Simulation in the context of ecological systems

The notion of modelling natural ecosystems in order to attempt to better understand or learn from them is by no means a new science. Initially, most mathematical models used simple algebraic relationships [15] and, later, diffusion models to analyse and better understand the local dispersal of insects [43]. In these cases, dispersion was modelled according to a mathematical relationship with a view to effectively simulating the population dynamics of a swarm at large.

In 1996, Minar *et al.* [59] released *The Swarm Simulation Software*, which is a toolkit for building multi-agent simulations. The toolkit supports a hierarchical modelling approach whereby agents exist in swarms of other agents in nested structures. The aim of the software is to provide open-source code and object-orientated libraries which facilitate building, analysing, displaying and experimenting with typical swarm models. Hiebeler [33] then developed a prototype of the Swarm Simulation Software in order to provide a general-purpose tool for building simulations involving a large number of agents behaving and interacting within a dynamic environment.

Various pieces of software with specific application to ecological systems have also been developed and implemented to simulate specific real-world environments.

Econet [45] is a simulation and analysis software package which integrates a dynamic simulation capability with steady-state network analysis and provides a simple, yet flexible, interface for users to develop sophisticated ecological models. This software was developed as a result of recent advances in experimental techniques which generated large amounts of data on biological systems and genetic networks. *Econet* is equipped with a web-based interface which uses an agent-based modelling approach and has the capability of modelling thousands of stocks and solving sophisticated numerical equation routines which require little input from the user.

GoldSim [28] is capable of creating realistic, complex, multi-media environmental systems for ecological and biological modelling. The software incorporates Monte Carlo simulation in order to represent uncertainty which is inherent in such systems and facilitates the evaluation of risk, potential environmental impacts and resource management decisions. *GoldSim* has sufficient power and flexibility to build a system of interacting models ranging from individual organisms to entire ecosystems.

RAMAS [3] is software intended for ecological and environmental risk analysis. More specifically, *RAMAS IRM 2.0* is a platform for modelling the risk of pest adaptation to genetically modified crops under a broad range of resistance management strategies. The software has the flexibility to accommodate all major insect crop pests by making use of user-defined life histories. One drawback of the software is that it is not visually intuitive, but rather runs an instantaneous simulation and outputs graphical and numeric data to be interpreted by the user, thus diminishing the effectiveness of its communication potential to parties for whom this form of experimentation is most pertinent.

A number of so-called *individualistic approach* models have since been developed for analysing, among others, dynamic swarms [26], predator-prey relationships [29] and the kinematics of neonatal rats [75]. Each of these models approach the problem by modelling the individual agents using their own data structures and internal state-variables, as well as action functions sent to the agent by other objects.

In particular, previous studies have been done on insect species using an agent-based modelling approach. Pratt *et al.* [67] used an agent-based modelling approach to simulate collective nest choice by the ant *Temnothorax albinpennis*. This model considers the ability of an ant colony to collectively choose the best of several nest sites in spite of the fact that many of the active ants who organise and move the colony only visit one potential site. This is attributed to the ability of an ant to recruit following ants after an independent assessment of a site. Other ants are recruited via slow tandem runs which eventually result in a quorum of ants being summoned to the site, generating a collective decision amongst the passive bulk of the colony. The behaviour of each ant in this model is governed by a network of states which are interconnected by transition probabilities. Further, each state belongs to one of four major phases in the decision making process, progression between which is governed by communication to and from fellow agents.

Gu and Novak [30] designed an agent-based model of mosquito foraging behaviour for malaria control. This model included hypothetical villages created in grid-based landscapes representing African rural communities. Each village contained twenty houses occupied by a number of residents assumed to be a normal variate with a mean of five and minimum of two. Seventy aquatic habitats were also randomly distributed across the landscapes, with the assumption being made that all houses and aquatic habitats were identical in their attractiveness as oviposition sites to females. This assured variations in the distributions of host-seeking mosquitoes and larval abundance among habitats is solely dependent upon their location relative to the foraging paths of mosquitoes.

The mosquito life cycle was modelled using a matrix transition which assumed constant rates of development and survival over the period of the simulation for populations of eggs, larvae and pupae. Adult female mosquitoes were instructed to search for suitable oviposition sites, bounded by a density-dependant threshold per habitat, since density-dependent effects have been observed on larval development, survival and body size of emerged adults. The movement of each emergent female was tracked as they sought hosts for oviposition. Foraging was treated as a moving window through the landscape, with the size of the window defined by the perceptual range and the movement of the window defined by the mosquito's maximal flight length. Through this, females spread through the simulated villages, selecting suitable oviposition sites and, in the process, transmitting malaria.

The success of such *agent-based* models encouraged a variety of applications of the simulation modelling approach, so much so that an annual conference called *Swarmfest* was established where international tool-users and tool-developers collaborate to present their agent-based models in order to share ideas and expertise in the hope of further developing and propagating agent-based modelling [74].

No formally agreed-upon ecological modelling software is earmarked as superior by users, but, in light of the aforementioned software which continues to be developed and improved upon, it appears that a general agent-based modelling approach is certainly the most favourable model type to assist in analysing problems of a complex nature such as natural systems. Most modellers also openly share their model frameworks or models in order to assist and further develop this field of simulation [74].

3.5 Chapter summary

In this chapter, an introduction to the discipline of simulation modelling, as well as the types of simulation models which exist, was given. The concept of abstraction, and the consequent role it plays in the selection of the level of simulation required, was discussed and, thereafter,

strategic-level, operational level and physical level simulations were described.

In §3.1.1, static and dynamic simulations were differentiated between and examples were given of each case. Thereafter, §3.1.2 detailed the four primary simulation modelling approaches, namely discrete-event modelling, system dynamics modelling, dynamic systems modelling and agent-based modelling. Each of these approaches were described in detail and their typical operation and underlying structure was also introduced. This was followed by §3.1.3 wherein typical modelling concepts and their associated terminology were described and defined in relation to their application in the field.

In §3.2, the advantages and disadvantages of simulation were discussed in detail and it was seen that most disadvantages are able to be overcome by the user owing to ongoing developments in the field. This was followed by Figure 3.1 in §3.3 which detailed the steps which are followed in a typical simulation study and explained the processes usually followed at each step.

Finally, the chapter concluded with §3.4 where a short overview of the development of simulation in an ecological context was given. Some of the more recent software packages which have been developed for use in the field such as *The Swarm Simulation Software*, *Econet*, *GoldSim* and *RAMAS*. These software packages were found to be tailor made to simulate specific scenarios and, in some cases, suffered shortcomings which were deemed to detract from the overriding purpose of simulation modelling of ecosystems. Examples of agent-based models which have been developed to simulate other ecological systems were also given, with specific reference being made to models of other insect types.

CHAPTER 4

An agent-based model of *E. saccharina*

Contents

4.1	The ANYLOGIC simulation modelling environment	38
4.2	General considerations of the proposed model	39
4.3	The graphical user interface (GUI)	40
4.3.1	General description of the GUI	40
4.3.2	Description of the constructed simulation environment in an ANYLOGIC modelling environment	43
4.3.3	Functionality assessment of the GUI	47
4.4	The life cycle of <i>E. saccharina</i>	52
4.4.1	General description of the implemented life cycle of <i>E. saccharina</i>	52
4.4.2	Description of the implemented life cycle in an ANYLOGIC modelling environment	54
4.4.3	Functionality assessment of the implemented life cycle	60
4.5	The mating process of <i>E. saccharina</i>	64
4.5.1	General description of the implemented mating process of <i>E. saccharina</i>	64
4.5.2	Description of the implemented mating process in an ANYLOGIC modelling environment	67
4.5.3	Verification of the implemented mating process functionality	73
4.6	The oviposition and dispersal of <i>E. saccharina</i>	75
4.6.1	General description of the implemented oviposition and distribution processes of <i>E. saccharina</i>	75
4.6.2	Description of the implemented oviposition and dispersal processes in an ANYLOGIC modelling environment	78
4.6.3	Verification of the implemented oviposition and distribution processes functionality	88
4.7	Chapter summary	92

An agent-based simulation model of *E. saccharina* was designed in an attempt to emulate the insect's behaviour and spatial distribution in sugarcane. The approaches taken and assumptions made during the design of this model, as well as the motivations behind these approaches, are discussed in this chapter. The implementation logic behind the workings of the model is also discussed. Each major aspect of the simulation, as well as the associated assumptions, is verified against existing knowledge of the biological process as described in §2 in order to determine the accuracy with which reconstruction has been achieved.

4.1 The ANYLOGIC simulation modelling environment

The agent-based model was developed and designed in the ANYLOGIC UNIVERSITY 7.3.1 [94] software suite. This software is a multi-method simulation modelling tool developed by *The Any-Logic Company* and supports all major modelling approaches. ANYLOGIC has a sophisticated suite of model building tools for different domain models, as well as a graphical modelling language which allows the user to extend simulation models using JAVA code [94]. ANYLOGIC was found to be a suitable choice of software owing to its sophisticated facilitation of the modelling approach for a number of reasons. The model architecture supported in ANYLOGIC allows for simple hierarchies to be defined and interconnectedness between agents is easily governed in the top-level agent tab. Furthermore, ANYLOGIC offers continuous, discrete and, importantly, GIS animation and mobility, which is useful for future work proposed to stem from this study. Agent communication and networks are also well defined and supported in ANYLOGIC, simplifying interaction instances involving a small group of agents selected from a large population, as is the case during the *lek* mating process of *E. saccharina*. Finally, the ability of an ANYLOGIC simulation model to dynamically create or destroy agents facilitates the development of population dynamics through multiple generations of the pest.

In order to translate *E. saccharina*'s population dynamics and behaviour into executable code in an ANYLOGIC simulation environment, several components were employed to construct the framework and code sequences which translate into agents who mimic the pest's behaviour. These components are shown in Table 4.1.








Name of component	Use in simulation	ANYLOGIC icon
Object class	Represents a separate agent who possesses its own internal structure governing its actions and decisions	
Parameter	Represents agent characteristics and behaviour and only changes when the behaviour of the agent is changed	
Variable	Stores results of model simulation or model object characteristics, changing over time	
Function	Executes a portion of code or returns the value of an expression each time it is called in the simulation	
Link	Creates links between agents along which messages can be sent for direct control and relationships between linked agents	
Event	Used to schedule action in the simulation using delays or timeouts	
Linked EXCEL spreadsheet	Used for platform-independent access and connectivity to MS EXCEL for reading from or writing to spreadsheets	

TABLE 4.1: *Different components employed in the simulation model.*

The general structure of each agent's life cycle was designed using a statechart-based approach in which each state represents a different point in the agent's life cycle. Each state imposes specific behavioural aspects on the pest, as determined by a combination of the appropriate variables, parameters and functions. States are connected by transitions which can be executed by a number of different triggers. When a transition is triggered, an agent moves from its current state into the new state as directed by the transition. In this manner, the agent moves through its life cycle and emulates *E. saccharina* in nature.

4.2 General considerations of the proposed model

The fundamental structure of the simulation is shown in Figure 4.1. This framework encapsulates the most notable life stages and activities of the pest which must be included in a simulation that aims to accurately mimic its behaviour and distribution. These attributes are discussed in finer detail by van Vuuren *et al.* [88].

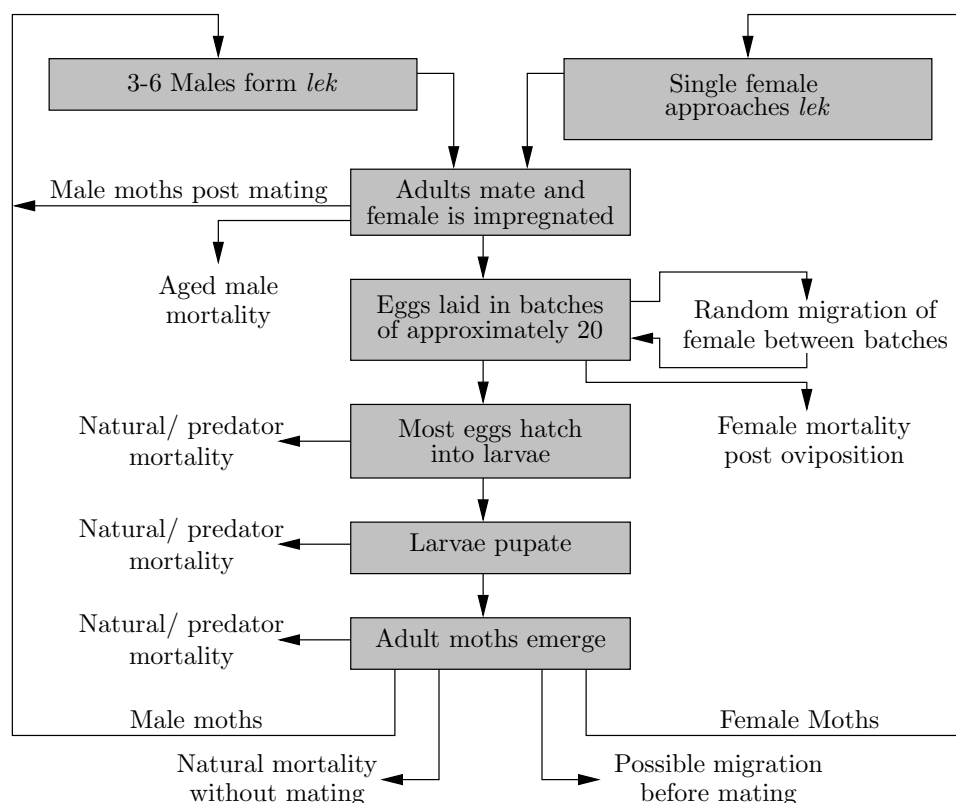


FIGURE 4.1: Flow diagram of the general life cycle followed by individual agents in the agent-based simulation model of the growth and spread of an *E. saccharina* population [88].

Four fundamental building blocks are included in the simulation model and are discussed in the sections that follow. These are the graphical user interface (*GUI*), the life cycle of *E. saccharina* and the associated influence of temperature on the pest, the mating process of *E. saccharina* and, finally, the oviposition and consequent in-field spatial distribution of the stalk borer within the simulated area. Each one of these building blocks is deemed to play a pivotal role in the reconstruction and simulation of the biological behaviour of *E. saccharina*.

Each of the following sections begins with a general overview of the modelling approach followed. This includes the relevant assumptions made in each process, derived from the literature and expert opinion discussed in §2. Thereafter, the manner in which the processes were constructed in an ANYLOGIC simulation environment is described in detail and all notable model building tools which were employed are introduced and explained. Finally, each implemented process is verified against the literature and assumptions on which it is based to determine the accuracy with which reconstruction has been achieved, as well as how effectively it emulates true *E. saccharina* behaviour.

4.3 The graphical user interface (*GUI*)




The *GUI* translates the functionality and interaction of all aspects of the model to the user in an understandable, informative manner. Careful consideration is given to ensuring that the display is both informative and visually intuitive to the simulation operator. The following text describes the *GUI* as seen by the user of the simulation, as well as the underlying simulation structure which governs, facilitates and links other model-composing elements.

4.3.1 General description of the *GUI*

In ANYLOGIC, the *GUI* is typically constructed in the **Main** tab and is the environment in which the default output of the simulation, as seen by an operator, is designed. In light of this, features which contribute to the functionality, flexibility and understanding of the model must be incorporated on this level in the simulation model. A screenshot of the *GUI* for the simulation model of *E. saccharina* is shown in Figure 4.2.

With reference to terminology introduced in §3.1.3, the *system* being represented in this model is a population of *E. saccharina* moths infesting part of a sugarcane plantation. The model boundary was chosen to be one hectare ($100m \times 100m$) of continuous simulation space. There are approximately 130 000 sugarcane stalks per hectare in a typical sugarcane plantation [19]. This is deemed suitable to provide enough complexity for prevailing behaviour and population dynamics to become evident in the simulation, whilst still not cluttering the simulation space and over-complicating analysis and investigation. The model was chosen to make use of an hourly time unit (*i.e.* one second represents an hour of activity in the simulation). This time unit was chosen to accommodate control of time-specific activity of the agents, whilst still allowing the simulation to progress acceptably quickly.

Almost all events which occur in the simulation are *endogenous* owing to the specified system boundaries. There do, however, exist instances where data or specific external events are incorporated by means of *exogenous* event triggers. Furthermore, in order to realistically simulate the given hectare as part of a larger sugarcane field, the possibility of moths entering and exiting the simulation space by means of an absorbing and emitting border condition was also incorporated as an *exogenous* event (discussed later in this chapter). A number of important attributes of the *GUI*, as labelled in Figure 4.2, are discussed below.

- (a) **Object classes:** Three separate object classes are defined in this simulation. A  **female** object class, a  **male** object class and a  **leks** object class accommodate independent control of each of the three agents in the simulation.
- (b) **Clock:** Certain behavioural aspects of *E. saccharina* are confined to specific times of the day. Furthermore, notable differences in the population dynamics of the pest are attributed to seasonal variation. In order to assist the user in determining the relative time and date being experienced by agents in the simulation, a display clock is incorporated which tracks these model attributes. In conjunction with this, a sun and moon icon are also included in the bottom left corner of the *GUI*. The sun icon is visible during simulated day time (06h00–18h00), whilst the moon icon is visible during simulated evening and night time (18h00–06h00).
- (c) **Layer Function:** In a simulation model containing 130 000 sugarcane stalks, there may be tens of thousands of agents present at a given time. Furthermore, these agents may

4.3. The graphical user interface (GUI)

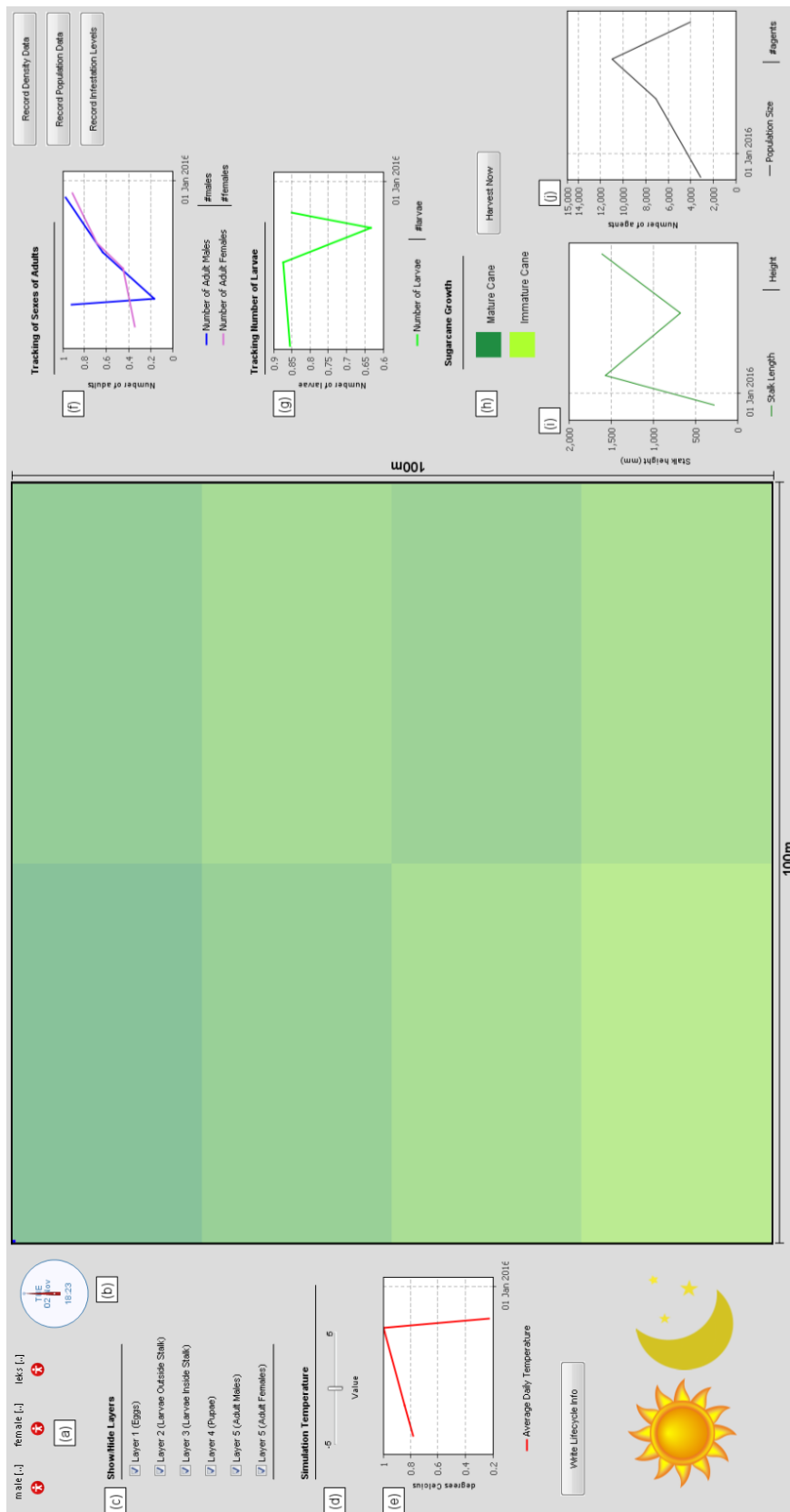






FIGURE 4.2: Screenshot of the GUI.


exist across all four life stages of the pest. A layer function is incorporated to show or hide life stages and/or, in the case of adults, different sexes of the moth. This functionality can be used both whilst the simulation is static or operational, so as to allow for specific analyses of certain life stages or attributes of the pest.

- (d) **Temperature adjustment bar:** In the simulation, temperatures are drawn from a linked EXCEL spreadsheet which contains a historical temperature profile for the simulated region. In this case, monthly averages of data collected during the period January 2011–September 2015 at SASRI (Mount Edgecombe, Kwa-Zulu Natal, South Africa) are incorporated into the model. In order to facilitate analysis of abnormal scenarios, a temperature-adjustment slider bar is incorporated. This bar increases or decreases the daily temperature value by a set amount for the duration of the entire simulation run. This attribute of the simulation environment may be used to simulate ‘out of the ordinary’ warm or cold periods, facilitating investigations of the pest propagation into previously uninfested areas, linked to a global rise in mean temperature as a result of climate change.
- (e) **Simulation temperature tracker:** This chart is incorporated to show a local history of the daily temperatures which have been incorporated into the simulation.
- (f) **Number of adults tracker:** Naturally, adult moths are responsible for propagating and distributing a population within sugarcane. Furthermore, females are more likely to disperse if they remain unmated for two nights post-emergence. This means that an imbalance in sexes of the adult moths may have a direct impact on the population’s behaviour and spread. In light of this, a chart which tracks the number of male and female moths in the simulation who are currently in the adult life stage is incorporated for ease of reference.
- (g) **Number of larvae tracker:** The robust larval stage of *E. saccharina* is primarily responsible for the damage caused to sugarcane when it bores into the stalk of the plant to feed on the inner nutrients. For this reason, a chart which tracks the number of larvae currently feeding on the plant’s interior nutrients is included to determine the level of infestation in the simulated space.
- (h) **Underlying sugarcane field structure:** The literature suggests that the maturity of sugarcane might play a role in the female’s choice of where to oviposit eggs post-mating [22]. Furthermore, in order to mature sufficiently and progress to the pupal stage of its life cycle, *E. saccharina* larvae require a certain amount of nutrients from the sugarcane stalk into which they bore. In order to facilitate realistic testing and analysis of these factors, three different underlying sugarcane crop arrangements were incorporated which divide the simulation space into a preset arrangement of mature and immature sugarcane within which the moths can disperse owing to their biological propensity. In conjunction, sugarcane growth and carrying capacity was simulated using inputs derived from the CANEGRO model [40].
- (i) **Stalk length tracking:** In order to view the growth progress of the sugarcane stalks in the simulation, a chart recording their height is included.
- (j) **Total population size tracking:** The total number of agents present in the simulation is recorded in this chart in order to provide the simulation operator with insight into the propagation of the population under the conditions being simulated.

4.3.2 Description of the constructed simulation environment in an ANYLOGIC modelling environment

The simulation environment is designed in the `Main` workspace. Although referred to as an object class, the `Main` class functions differently from other object classes in that the contents of this class are what is seen by the user when the simulation is executed. The statecharts and contents of other object classes are typically not viewed by the operator of the simulation, although they may be navigated to be viewed if required.

The  `Main` object class functions in the same manner as the  `male`,  `female` and  `Leks` object classes in that parameters, variables, functions and events can all be declared and controlled from its environment.

In the simulation, attributes declared and controlled by the  `Main` object class were chosen by virtue of being of ‘global’ importance to the simulation. That is to say that events, functions, parameters and variables declared in `Main` typically affect all other object classes and are, as such, accessed and incorporated into the separate, inner-agent workings of the other object classes. This is important for events or functions which should be identical in nature and timing for all agents in the simulation, as they can be controlled from a centralised point in the model and the information can be relayed identically to the different object classes. The ‘global’ declarations in `Main` are shown in Figure 4.3.

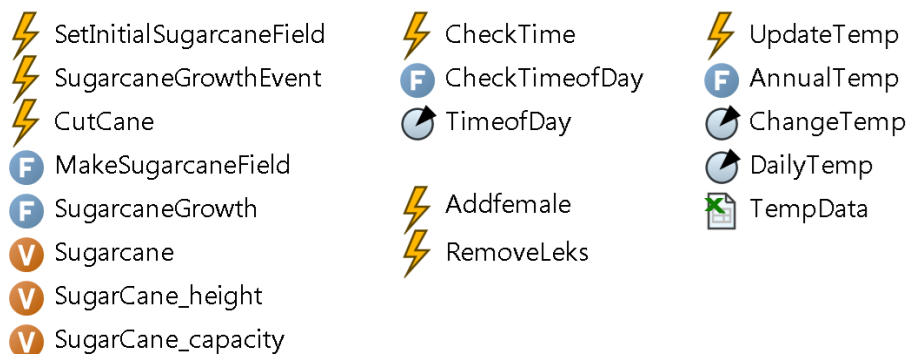








FIGURE 4.3: Global parameters declared in the `Main` environment of the simulation.

The manner in which each of the important attributes of the simulation environment discussed in §4.3.1 were incorporated into the ANYLOGIC simulation modelling environment in conjunction with the components shown in Figure 4.3 is discussed below.

- (a) **Object classes:** In order to facilitate separate behaviour control of both sexes of *E. saccharina*, an object class is defined for each gender, namely  `male` and  `female`, in which their individual statecharts are constructed. A further, ‘theoretical’ object class, namely  `Leks`, is also incorporated in order to assist in the creation and control of *leks* in the simulation model. The implementation of this approach is discussed further later in this chapter.
- (b) **Clock:** The clock is a standard agent provided for inclusion in ANYLOGIC projects and is incorporated here to convey the relative date and time being experienced by agents in the simulation model. The  `CheckTime` event is a cyclically-triggered event, set at the beginning of each model timestep. When triggered, it executes the  `CheckTimeOfDay` function. This function calls a built-in ANYLOGIC function named `getHourOfDay()` which

returns the current relative hour of the day according to the simulation. The  **TimeOfDay** parameter is then updated with this information and can be called by agents to test the relative current time of the simulation preceding time-specific activity.

- (c) **Layer Function:** In order to incorporate the functionality allowing agents of different life stages to be shown or hidden, the  **checkbox** feature is used. A conditional statement controlling dynamic visibility of the agent in each state was constructed and linked to the appropriate check box. As a result, when a box is checked, the corresponding agents in that state are displayed in the simulation space. This approach also accommodates the possibility of more than one state being displayed at a given time.
- (d) **Temperature adjustment bar:** In order to incorporate the temperature adjustment functionality, a slider control was added to the simulation environment. The value at which this bar is set is linked to the  **ChangeTemp** parameter and incorporated into the  **AnnualTemp** function, described in the next bullet point.
- (e) **Simulation temperature tracker:** In the simulation,  **UpdateTemp** is an event which occurs cyclically at each timestep in the model. When triggered, the  **AnnualTemp** function is called. This function uses the built-in ANYLOGIC functions `getMonth()` and `getDayOfMonth()` to determine the relative model calendar date and then searches through the linked EXCEL spreadsheet,  **TempData**, for the average temperature for that particular day in the simulated space. The  **TempData** file stores the temperature history of the simulated space as determined by an exterior data source. The returned temperature is sampled from a normal distribution over the historical temperature and an accompanying assumed standard deviation of 0.5°C^1 to provide some fluctuation in the subsequent simulation runs. As mentioned, this value may be increased or decreased by the value of the  **ChangeTemp** parameter and the result is stored as the  **DailyTemp** parameter.
The output data of the  **AnnualTemp** function are plotted in the chart to provide a temperature history profile for the simulation. Using this, the seasonal changes which occur in the simulation model may also be observed by the simulation user.
- (f) **Number of adults tracker:** Statistical functions are employed to tally the number of agents present in each life stage during the simulation. In this chart, the statistics describing the number of adult agents present at a given time are displayed. The chart updates once per model time-step (hourly).
- (g) **Number of larvae tracker:** Similarly, statistics tallying the number of agents of each sex in the **Larvae** state are displayed in this chart. This state represents robust larvae who have penetrated a sugarcane stalk and are busy feeding on its interior. In this case, since there is no difference in behaviour between larvae of differing sexes, the total number of larvae within stalks is represented as the sum of male and female larvae. The displayed data are also updated once per simulation time-step.
- (h) **Underlying sugarcane field structure:** In order to incorporate different underlying sugarcane field structures which may influence female agent movement during oviposition, as well as incorporate sugarcane growth and carrying capacity, the simulation space requires defined points, representing physical sugarcane stalks. These points must be attributed a relative ‘cane maturity’ value which can be assessed by a female to guide her

¹In order to ensure the most realistic yet current average temperature profile was used in the model, only five years of historical data was used. Since this is insufficient for calculation of an accurate standard deviation, the data was visually analysed and this value was seen to represent the fluctuations in the data sufficiently well.

choice of oviposition site, a measure of length to track sugarcane growth and, for simulation control purposes, a tally of the number of larvae who currently feed on or previously fed on that particular sugarcane stalk. This tally is used to determine whether sufficient stalk space (and corresponding nutrients) remain for a larva wishing to penetrate a particular stalk to mature sufficiently and progress to the **Pupae** state, and is termed the ‘carrying capacity’ of the sugarcane. In the simulation, a simplifying assumption is made that sugarcane interior which is destroyed by a feeding larva does not regenerate in any way and, therefore, the only manner in which a stalk can increase its capacity is through new growth over time.

Since the simulated space is continuous, an underlying, discrete platform was established to accommodate the aforementioned functionality. **V Sugar cane** is a two-dimensional array variable spanning the one hectare simulated sugarcane field. The array is chosen to store $1\,080 \times 1\,080$ discrete sugarcane maturity values. This resolution was selected as it accommodates sufficient, realistic inclusion and spacing of the approximately 130 000 sugarcane stalks grown in a hectare of sugarcane plantation. Similarly, **V Sugar cane height** is an array which stores the current height of the stalks in the simulation (and ‘0’ at array points where no physical stalk is located), while **V Sugar cane capacity** is an array that tracks the number of larvae who feed on a particular stalk within the array.

When larvae disperse upon maturing past the egg life stage (following the process discussed in §4.4), they are prompted to locate the specific array points representing sugarcane stalks into which they eventually bore. When this occurs, the **V Sugar cane height** and **V Sugar cane capacity** values of the particular stalk are assessed to determine whether there remain sufficient nutrients for the larvae to mature. The **V Sugar cane height** array is continually updated by means of the **⚡ Sugar cane Growth Event** which is called once per day. This event calls the **F Sugar cane Growth** function which, in turn, makes use of the CANEGRO model equations and corresponding simulated model temperature for that day to increase the respective height of each stalk in the simulation by the appropriate amount.

The CANEGRO model estimates the amount of stalk growth, $\ell(t)$ (in mm) on day t as

$$\ell(t) = 0.16(24)(-1.32 + 0.176(\tau_t - 10)) \quad (4.1)$$

for a two-year crop cycle [14], where τ_t represents the simulated temperature for day t . In this model, stalk development rate is assumed to be 16% of the leaf development rate which is, in turn, assumed to occur in day degrees. Leaf development has a threshold of 10°C. The average amount of stalk growth on day t for a one year cycle crop is given by

$$\ell'(t) = 0.16(48)(-1.32 + 0.176(\tau_t - 10)). \quad (4.2)$$

This was found to correspond to cane growth data from the coastal region in KwaZulu-Natal [27].

Expressions (4.1) and (4.2) were both included as options in the **F Sugar cane Growth** function and the appropriate expression, depending on the length of the crop cycle (as specified by the user), is utilised to increase the length of the stalks appropriately.

With reference to the effect of sugarcane maturity, a female searching for possible oviposition locations employs functions (discussed later in this chapter) which round off data points to ensure the discrete sugarcane array values are searched through, located and

assessed for suitability (according to relative maturity values stored in the **V** `Sugarcane` array) within the continuous movement space of the agent. The **F** `MakeSugarcaneField` function is responsible for attributing maturity values to the **V** `Sugarcane` array when a selection is made from the check boxes which set the underlying canefield structure in the simulation environment. The values are arbitrarily assigned and have no biological significance, but affect the future strength and competitiveness of the moths when they reach adulthood. In a full-scale, accurate simulation, the underlying sugarcane field would require a more detailed implementation, including sugarcane values which increase as a function of stalk height. Since this study is focused on modelling specifically the attributes of *E. saccharina*, the above-mentioned simplified inclusion of sugarcane maturity values was adopted to still allow for incorporation of the development and oviposition site effects attributed to sugarcane maturity. In essence, it acts as a ‘place holder’ for a more detailed, future development of this facet of the simulation model. In this instance, the maturity values operate more as an indication to agents of better quality sugarcane in certain areas, as opposed to actual, age-dependent maturity values.

In this simplified case, mature sugarcane is chosen to be represented by fixed higher values and assigned an average value of ‘1.1’ across a triangular distribution between 1 and 1.2. Conversely, less mature cane is attributed a fixed lower value of average ‘0.95’ across a triangular distribution from 0.85 to 1.05. The manner in which these values are incorporated into the strength and competitiveness of the adult moths is discussed in the next section. The **F** `MakeSugarcaneField` is set to attribute a maturity value to one in every three points per array line. This results in 129 600 of the available 1 166 400 array points to represent actual sugarcane stalks within the simulated space.

The logic sequence of the **F** `MakeSugarcaneField` function begins by determining which one of the boxes is checked. Four possible, simplified underlying sugarcane structures are included as options in the simulation in order to test the effect of different areas of maturity on population dispersal and dynamics. These layouts are depicted in Figure 4.4 and can be interpreted as follows:

- (i) **None**: In this scenario, shown in Figure 4.4(a), the **F** `MakeSugarcaneField` function attributes all data points in the **V** `Sugarcane` array where stalks exist to be mature sugarcane.
- (ii) **Split**: In this scenario, shown in Figure 4.4(b), the simulation space is split into two halves. The **F** `MakeSugarcaneField` function attributes mature sugarcane values to points in the first half where stalks exist, whilst stalks in the second half are attributed values of immature sugarcane.
- (iii) **Rows**: In this scenario, shown in Figure 4.4(c), the simulation space is split into four rows of equal breadth. The **F** `MakeSugarcaneField` function attributes mature sugarcane values to stalks in the first and third rows, whilst stalks in the second and fourth rows receive immature sugarcane values.
- (iv) **Checkered (2×2)**: In this scenario, shown in Figure 4.4(d), the simulation space is split into four quadrants of equal dimensions. The **F** `MakeSugarcaneField` function attributes mature sugarcane values to stalks in the first and fourth quadrants, whilst the second and third quadrants’ stalks receive immature sugarcane values.

The default arrangement of the underlying sugarcane structure is the **None** arrangement where all the sugarcane is assumed to be mature. ⚡ `SetInitialSugarcaneField` is a once-off event which executes the **F** `MakeSugarcaneField` function when the simulation starts

and, in doing so, attributes mature sugarcane values to all points in the **V** Sugarcane array. Since the selection of the underlying arrangement of the sugarcane is made prior to model execution (discussed in §5) it is only executed once per simulation run.

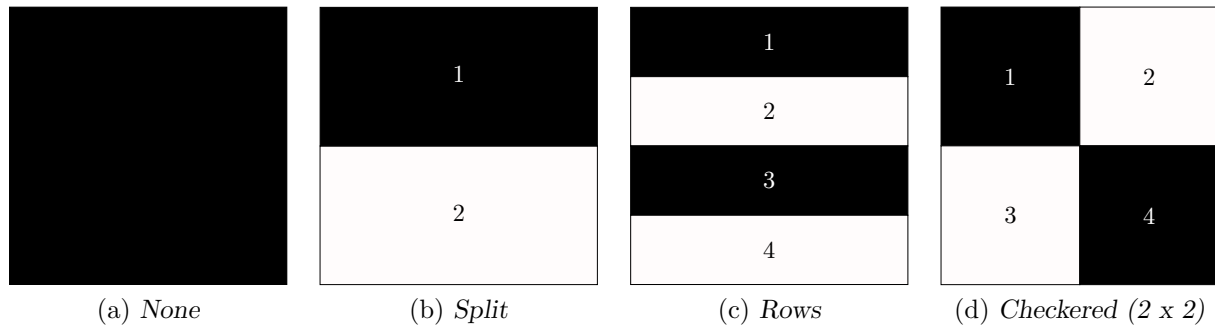


FIGURE 4.4: Possible underlying sugarcane layouts incorporated into the simulation model which may be changed by the user, where black indicates older, more mature sugarcane and white indicates younger, less mature sugarcane.

⚡ CutCane is an event which simulates the start of the annual milling season whereby cane is harvested before processing and refining. This event resets the **V** Sugarcane_height and **V** Sugarcane_capacity arrays to zero for each of the array points representing a sugarcane stalk. The event also simultaneously removes all pests which currently exist in the simulation, but for a very small percentage of the eggs and larvae (specified by the simulation user) which typically survive the harvesting period. The manner in which this is executed in the respective agents' statecharts is discussed in the next section. The **⚡ CutCane** event can be set to occur cyclically (annually or bi-annually, depending on the selected length of crop-cycle), or to be user-controlled in nature. If user-controlled, the harvesting season is simulated by selection of the control button which executes a sequence of code identical to that of the **⚡ CutCane** cyclic event. In line with the simplifying assumptions in respect of the sugarcane maturity values which operate as quality indicators in this simulation model, the harvesting function cuts all sugarcane within the simulated hectare, as opposed to staggered harvesting of different aged regions.

The **⚡ Addfemale** event adds a single female agent to the simulation who enters as an adult moth, flying into the simulation space from the hypothetical surrounding sugarcane. This is a rate-triggered event which occurs an average number of times per model time step. This rate is determined by the **🌀 NewArrivalRate** parameter.

⚡ RemoveLeks is an event incorporated to remove all *leks* which were created but not occupied during the course of the previous night. An occupied lek is removed through an action executed by the female that occupies it. **⚡ RemoveLeks** is executed once every 24 hours at 23h00 when mating of *E. saccharina* moths has subsided. The mating process is discussed in further detail later in this chapter.

4.3.3 Functionality assessment of the GUI

Owing to the fact that most of the outputs of the functionality included in the GUI are visual in nature, their correct implementation and operation could easily be verified by running the model and observing the different components in operation. Results for verification tests performed on

a number of the more complex components of the GUI in terms of correct implementation, the layer function and the underlying sugarcane structure, are reported in this section.

In order to test the layer function, the simulation was run a number of times and different combinations of life stages were hidden or shown. A series of snapshots illustrating the output of one such test is shown in Figure 4.5. As may be seen, the agents occupying states linked to selected layers remain visible whilst those in states linked to unselected layers are hidden in the simulation, indicating the functionality works as intended. Tests were conducted both on an active simulation, as well as a paused simulation and, in both cases, the functionality fulfilled its intended purpose.

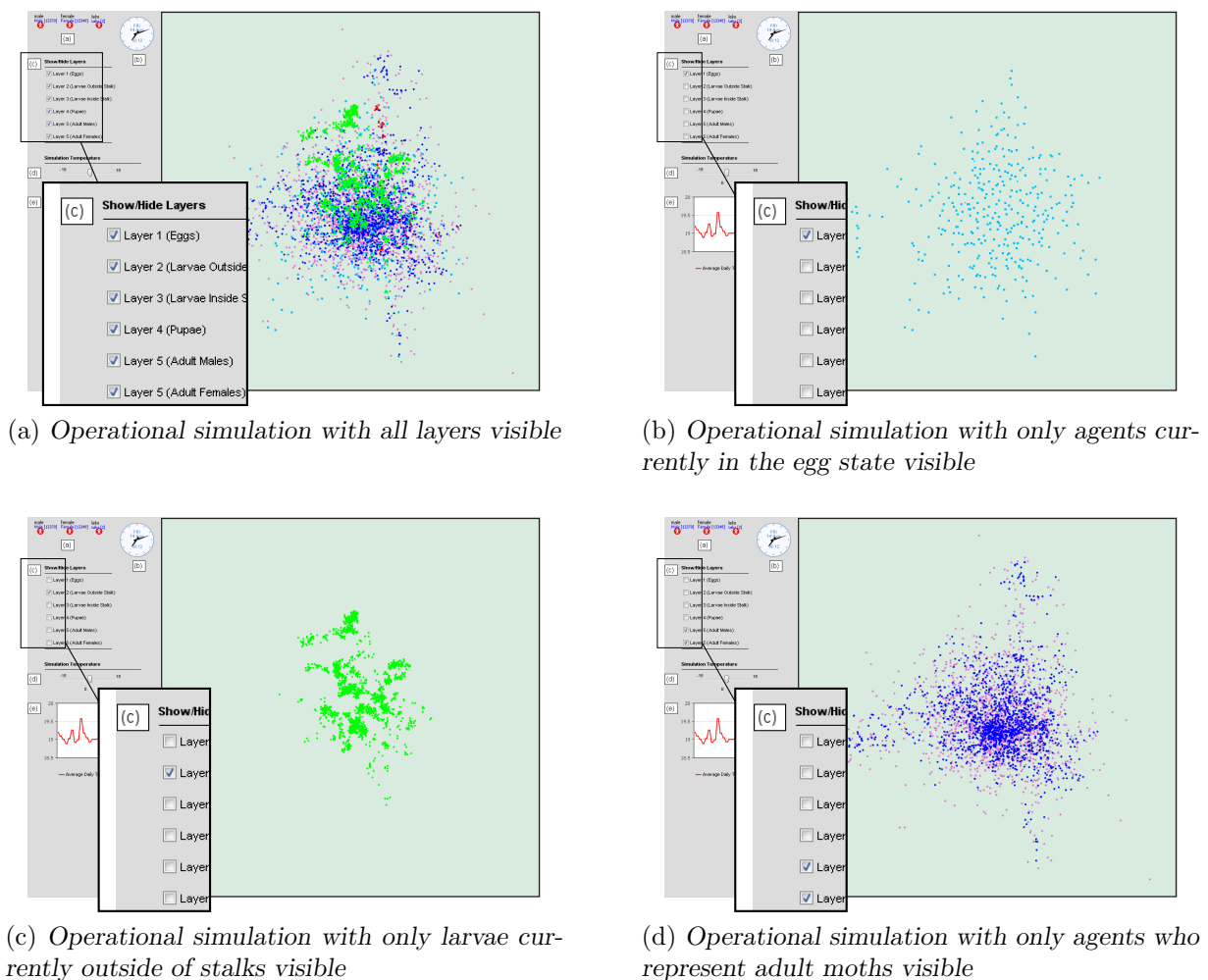
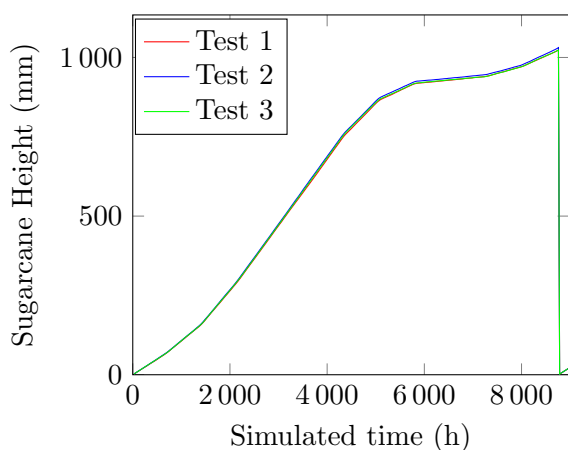


FIGURE 4.5: An example of a typical simulation run wherein different combinations of layers are chosen to be hidden illustrating the effective implementation and operation of the layer function in the GUI.

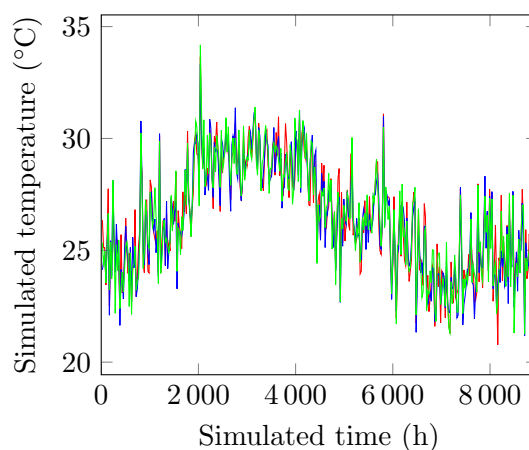
Testing of the implementation of the CANEGRO model was achieved in conjunction with the temperature read-in process. Tests were conducted to determine whether the growth of the sugarcane over different seasons within a year was consistent with observations in the field. Secondly, tests were performed to monitor the effect of higher temperatures on growth rate, as well as lower temperatures (including temperatures falling close to the 10°C threshold for leaf development²). Increasing and decreasing the read-in temperatures in the simulation was

²The CANEGRO model is derived from leaf development.

achieved using the temperature adjustment bar, thereby also testing its functionality. A harvest was included just prior to the termination of the test runs, indicating correct operation of the harvesting functionality.

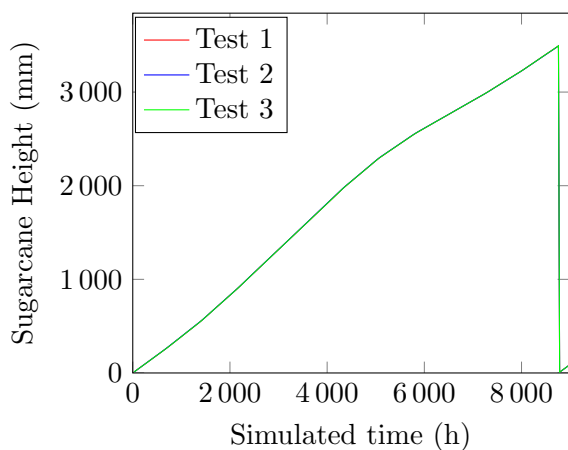


(a) Sugarcane height over simulated time.

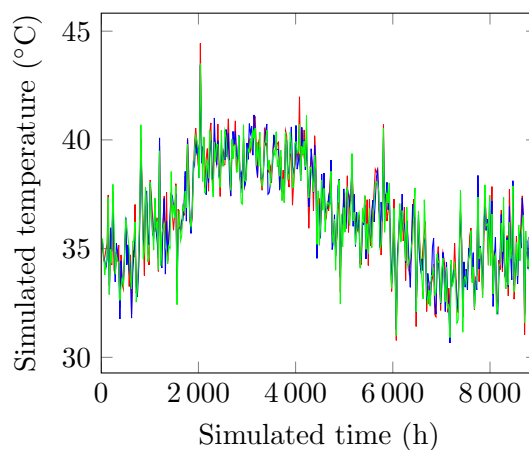


(b) Corresponding temperatures experienced in the simulation.

FIGURE 4.6: Sugarcane growth as governed by the CANEGRO model over the simulated period of one year exposed to typical annual KwaZulu-Natal temperatures.



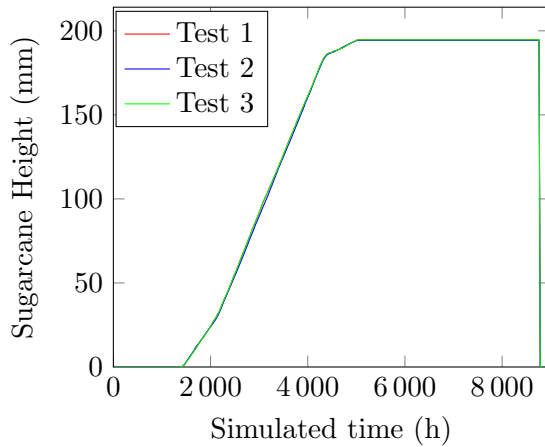
(a) Sugarcane height over simulated time.



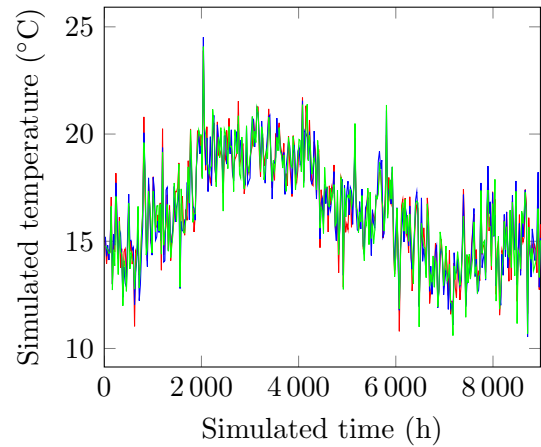
(b) Corresponding temperatures experienced in the simulation.

FIGURE 4.7: Sugarcane growth as governed by the CANEGRO model over the simulated period of one year exposed to temperatures inflated 10°C above typical annual KwaZulu-Natal temperatures.

As may be seen in Figures 4.6–4.8, the propagation of the sugarcane height is directly dependent on the simulated temperature being experienced by sugarcane stalks in the simulation. Figure 4.6(a) shows the typical growth pattern of sugarcane over the course of a year, growing faster during the warmer months and flattening out over the winter. In Figure 4.7(a), it can be seen that the sugarcane grows far quicker and, as a consequence, to a much greater final height owing to the inflated temperatures in the simulation. Conversely, the sugarcane in Figure 4.8(a) grows at a much slower pace, at times even halting growth completely when the temperatures are close to the growth threshold point.



(a) Sugarcane height over simulated time.



(b) Corresponding temperatures experienced in the simulation.

FIGURE 4.8: Sugarcane growth as governed by the CANEGRO model over the simulated period of one year exposed to temperatures deflated 10°C below typical annual KwaZulu-Natal temperatures.

A final set of tests were conducted to determine the maturity values attributed to sugarcane stalks for the different underlying structures. Three tests were performed for each available underlying arrangement and the corresponding maturity values of eight defined stalks in the simulation were recorded to determine whether the attributed maturity value was consistent with the expected range for that value based on a particular arrangement.

The four arrangements that were tested, as well as the array numbers corresponding to the stalks within the **V** Sugarcane array whose values were recorded in each case, are shown in Figure 4.9 and the outcomes of the tests are summarised in Table 4.3.3. As may be seen from these three replications of tests, the maturity values of the sampled sugarcane stalks in each arrangement case all fall within the anticipated respective value ranges, indicating that the **V** Sugarcane array is correctly established and that appropriate maturity values are attributed to the defined stalks.

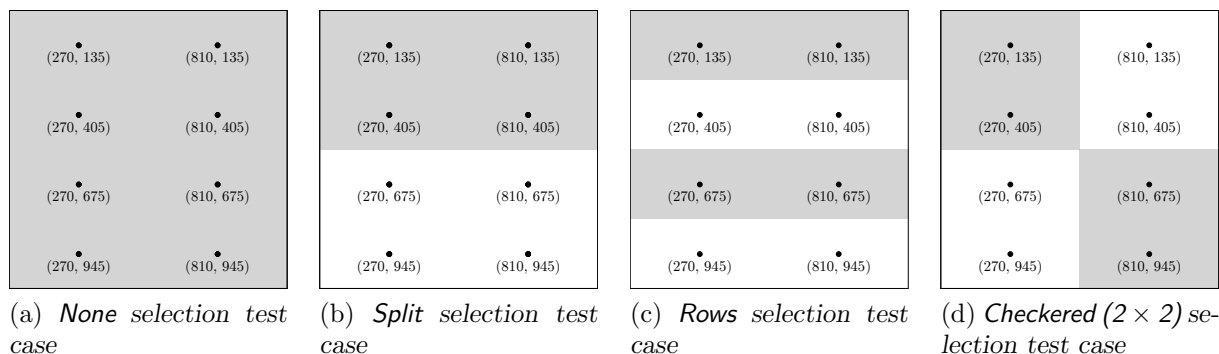


FIGURE 4.9: The location of test points used to ensure correct sugarcane maturity value setting during model operation for each of the four possible underlying sugarcane structure options in Figure 4.4. Areas which correspond to mature sugarcane are shaded grey.

Selection	Anticipated value	Measured co-ordinate	Result 1	Result 2	Result 3
None	Mature [1, 1.2]	(270, 135)	1.1096	1.0821	1.1120
		(810, 135)	1.0543	1.0661	1.1013
		(270, 405)	1.1587	1.0672	1.0438
		(810, 405)	1.0986	1.1927	1.1272
		(270, 675)	1.0868	1.0365	1.1725
		(810, 675)	1.0447	1.1260	1.0863
		(270, 945)	1.0800	1.0801	1.1733
		(810, 945)	1.1016	1.1437	1.1094
Split	Mature [1, 1.2]	(270, 135)	1.1421	1.1168	1.1193
		(810, 135)	1.0609	1.0825	1.1396
		(270, 405)	1.1945	1.1134	1.1174
		(810, 405)	1.1120	1.1749	1.1064
	Immature [0.85, 1.05]	(270, 675)	0.9444	0.9074	0.9175
		(810, 675)	0.9552	0.9943	0.9642
		(270, 945)	0.9409	1.0064	0.9077
		(810, 945)	0.9595	0.9675	0.9005
Rows	Mature [1, 1.2]	(270, 135)	1.0389	1.0945	1.0705
		(810, 135)	1.1240	1.0243	1.0265
	Immature [0.85, 1.05]	(270, 405)	0.9358	0.9041	0.9096
		(810, 405)	0.9291	1.0013	0.9112
	Mature [1, 1.2]	(270, 675)	1.0645	1.0783	1.0866
		(810, 675)	1.1959	1.0776	1.1072
	Immature [0.85, 1.05]	(270, 945)	0.9109	0.9315	0.9771
		(810, 945)	0.9725	0.8844	0.9714
Checkered (2 × 2)	Mature [1, 1.2]	(270, 135)	1.1349	1.1411	1.1782
	Immature [0.85, 1.05]	(810, 135)	0.9677	0.9182	0.9811
	Mature [1, 1.2]	(270, 405)	1.1953	1.0659	1.0392
	Immature [0.85, 1.05]	(810, 405)	0.9334	0.9987	1.0141
	Immature [0.85, 1.05]	(270, 675)	0.9332	0.8902	0.9411
	Mature [1, 1.2]	(810, 675)	1.1227	1.1423	1.0333
	Immature [0.85, 1.05]	(270, 945)	0.9044	0.9006	0.8830
	Mature [1, 1.2]	(810, 945)	1.1097	1.0973	1.0743

TABLE 4.2: Recorded maturity values of sugarcane stalks in the simulation set according to the available underlying sugarcane structure arrangements in Figure 4.4.

4.4 The life cycle of *E. saccharina*

Effectively implementing the typical life cycle of *E. saccharina* is paramount to developing a realistic, accurate model of the pest's behaviour. Using different states in order to represent different aspects, activities or, in this case, life stages of an agent is one of the available means of constructing an agent-based model. Transition trigger types include rate-triggered, condition-triggered, timeout-triggered, agent arrival-triggered or message-triggered transitions. Life cycle related transitions are primarily condition-triggered transitions that are dependent on the maturity of the pest at a given point in time. The life cycle construction also includes a number of parameters, variables, functions and events defined specifically for determining the behaviour of each agent, and are either predetermined, using information from the literature, or incorporated as random numbers drawn from an appropriate statistical distribution. The selection and calibration of parameters used in model construction is discussed in further detail in the next chapters.

4.4.1 General description of the implemented life cycle of *E. saccharina*

The logical progression through the life cycle of *E. saccharina* along with the associated activities as implemented in the simulation model are summarised in Figure 4.10.

The sex of an agent in the simulation is already defined in the egg life stage. The behaviour and life cycle of both male and female *E. saccharina* moths are, however, identical until they reach the adult life stages. In light of this, the description given in this section is appropriate for both sexes and, where appropriate, gender-specific activity is noted.

Eggs are laid in oviposition sites selected by the female (discussed in §4.6) and agents enter the simulation as eggs. Naturally, not all eggs will successfully hatch and progress to the next life stage. Furthermore, some eggs may be victims of predator mortality. If an agent's parameter governing this condition exceeds the selected probability of occurrence, the egg perishes and the agent is removed from the model³. No other activity occurs in this life stage and the egg simply matures as a result of the temperature experienced by the agent. Once the egg has matured sufficiently, it hatches and progresses to the next life stage.

The newly hatched larva is also subject to random or predator mortality as in the egg life stage, but according to a different probability. Being without any form of protection, the young larval life stage is the most vulnerable and, as a result, the probability of death will most likely be higher than in any other life stage. Larvae that do not perish proceed to disperse among the sugarcane stalks in the manner described in §2.5. Thereafter, the larvae are assumed not to move significantly for the remainder of the period spent outside of the sugarcane stalk. Once sufficiently robust and mature, each larva progresses to the next part of the larval life stage where it begins boring into a selected sugarcane stalk.

This part of the larval life stage begins with the larva searching in its immediate vicinity for the closest sugarcane stalk into which to bore [19]. The larva then moves towards this selected stalk and attempts to burrow into it. If the carrying capacity of the stalk has not yet been exceeded, the larva will successfully enter the stalk and begin feeding. If, however, the stalk is densely occupied and there remains insufficient space and nutrients for another larva to feed on in that particular stalk, it dies and is removed from the simulation. This was implemented according to

³The probability thresholds for predator or random mortality in all states are incorporated into the model using the best possible available measurement from in-field testing, but they are also changeable by the user of the simulation.

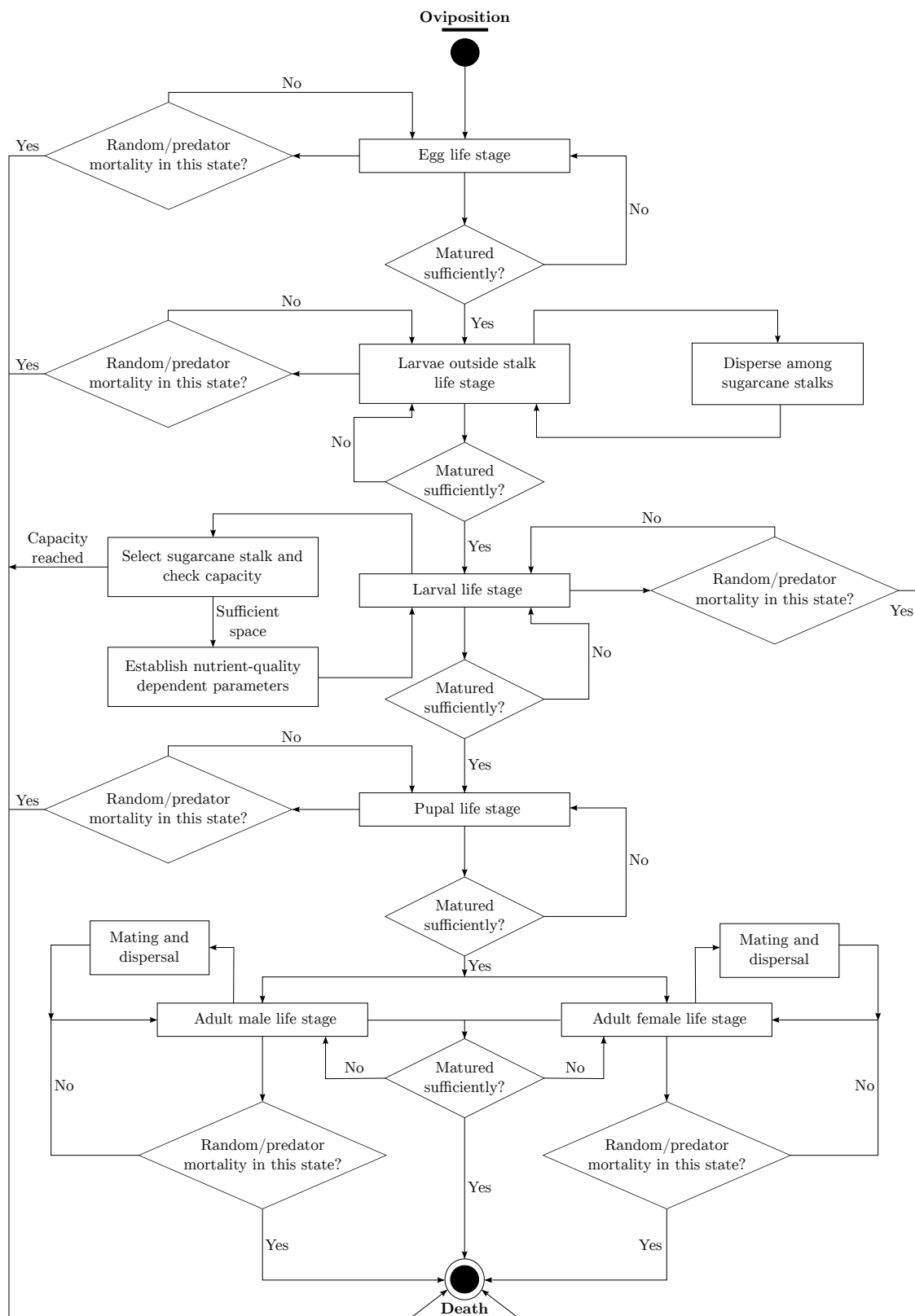


FIGURE 4.10: Flowchart of the general life cycle, including assumed associated activities, followed by *E. saccharina* agents in the simulation model.

the assumption that only a very small number of larvae can successfully change their choice of stalk in view of exceeded carrying capacity. Furthermore, in real life it has been observed that more larvae may feed on a stalk than can feasibly be accommodated and all may progress to pupation instead of death as assumed in the model. In such cases, however, the resulting adult moths have been observed to be very small, weak and uncompetitive in mating as a result of undernourishment [19] and are, therefore, assumed not to have significant future impact on the growth and spread of the population.




Random and predator mortality is far less likely in this life stage and, as such, the probability of these forms of death is likely to be low. The quality of the selected stalk as a food source for larvae affects the strength and competitiveness of the moth when it becomes an adult. Once the larva has matured sufficiently, it progresses to the next life stage and becomes a pupa.

Very little happens in the pupal life stage apart from the possibility of random or predator mortality. If a pupa does not perish and matures sufficiently, it progresses to the final life stage of the life cycle where it becomes an adult moth. The sex of the adult moth plays a determining role in its behaviour and decision making. The possibility of random or predator mortality still exists in the adult life stage and is incorporated in the same manner as in the other life stages.

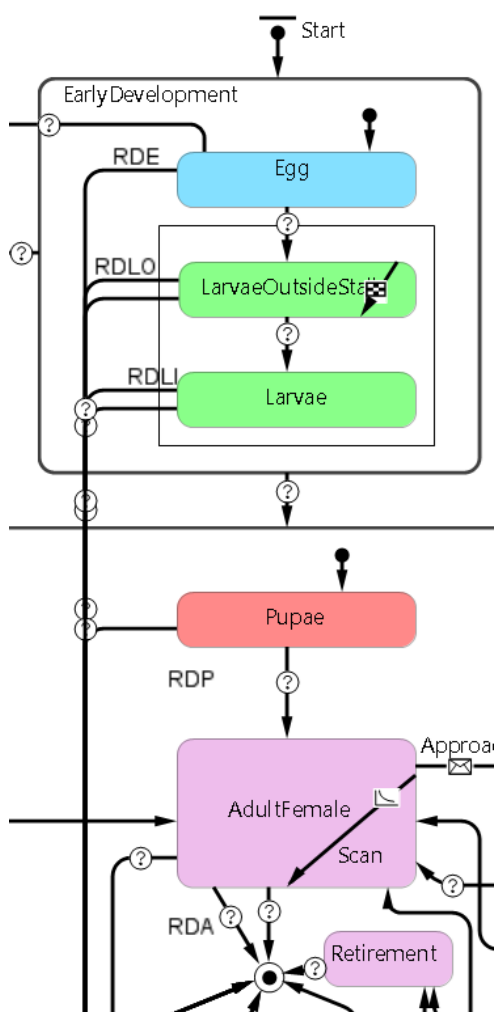
Once adults, male and female moths progress into the mating, oviposition and dispersal sub-cycles which are discussed later in this chapter. All mating and dispersal activity of *E. saccharina* occurs at night, and natural mortality of the pest depends on maturity driven by temperature exposure experienced primarily during the day. For this reason, it is assumed that the final progression from the adult stage to that of mortality never occurs during the aforementioned sub-cycles. Mortality will only take place once the adult moth has both matured sufficiently, as well as resides in the primary adult state post any adult activity.

4.4.2 Description of the implemented life cycle in an ANYLOGIC modelling environment

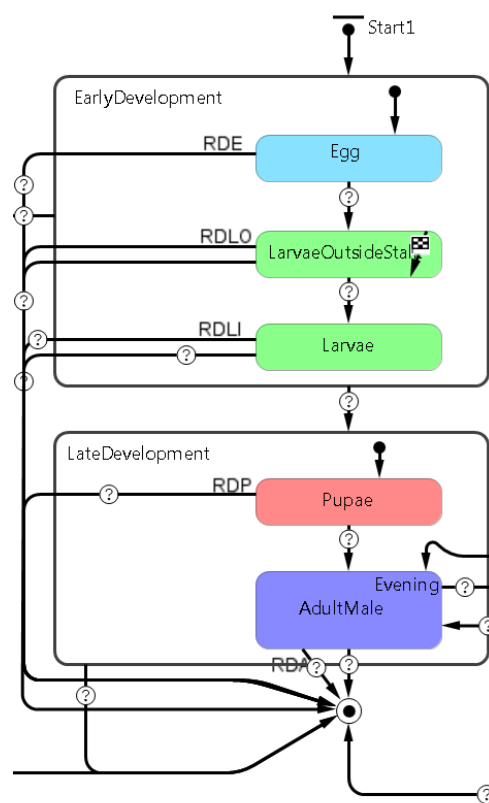
The partial statecharts of the female and male agents, as well as the associated model building tools responsible for the implementation of the lifecycle of *E. saccharina*, are shown in Figures 4.11(a) – 4.12(b).

As may be seen in both Figures 4.11(a) and 4.11(b), a newly generated agent of the  **male** or  **female** object class enters the simulation and moves immediately into the **Egg** state. In the case of the female, there exists a possibility that the agent entering the simulation is not a newly oviposited egg, but rather an external female entering the simulation space from the implied surrounding space. The two types of new female agents are distinguished by assessing an  **OutsideEnterer** parameter and, if the parameter indicates that the new female is, in fact, an adult migrating into the simulation space, the agent is diverted to another part of the statechart to be appropriately incorporated into the simulation model. This process is discussed later in this chapter.

In all states, the possibility of random or predator-related mortality exists as a condition-controlled transition from the current state to the final state, where the agent is removed from the simulation. Each of these transitions is controlled by a randomly generated ‘probability of death’ parameter which is compared to the predetermined probability threshold of the mortality transition in that state. This parameter is sampled a single time for each agent from a uniform distribution on the interval [0, 100] and the resulting sample is then attributed to the parameter which is used for conditional control on predator mortality transition in each respective state. Upon entering a certain state, the parameter which stores the sampled value is compared to the



(a) Screenshot of the life cycle control of the female in the simulation.



(b) Screenshot of the life cycle control of the male in the simulation.

FIGURE 4.11: Partial statecharts of the agents in an ANYLOGIC environment for the division of the simulation responsible for the accurate implementation and control of the lifecycle of *E. saccharina*.

threshold probability of the transition in that state and, depending on the outcome, the agent either perishes or continues to mature normally in the state.

In the **Egg** state, the transition accommodating the possibility of random or predator-related mortality is labelled RDE (*random death egg*) in both Figures 4.11(a) and 4.11(b). This transition is evaluated against the \odot **RandDeathEgg** parameter value of the agent upon entry to the **Egg** state to determine whether or not the agent perishes in this stage of its life cycle.

Transitions between different life stages of both male and female agents are triggered when the agent reaches a simulated measure of maturity. The determination of this measure, as well as the appropriate, experimentally determined maturity values required for progression through each life stage, was discussed in §2.6. Transitions in this part of the simulation model are conditional in nature. This means they are continually assessed and, when the transitional condition results in a **true** outcome, the agent progresses to the next state.

The \odot **Development** function is called cyclically by the ⚡ **SumDayTemps** event once per model time step. The logic sequence of the \odot **Development** function begins by using the \odot **Main**

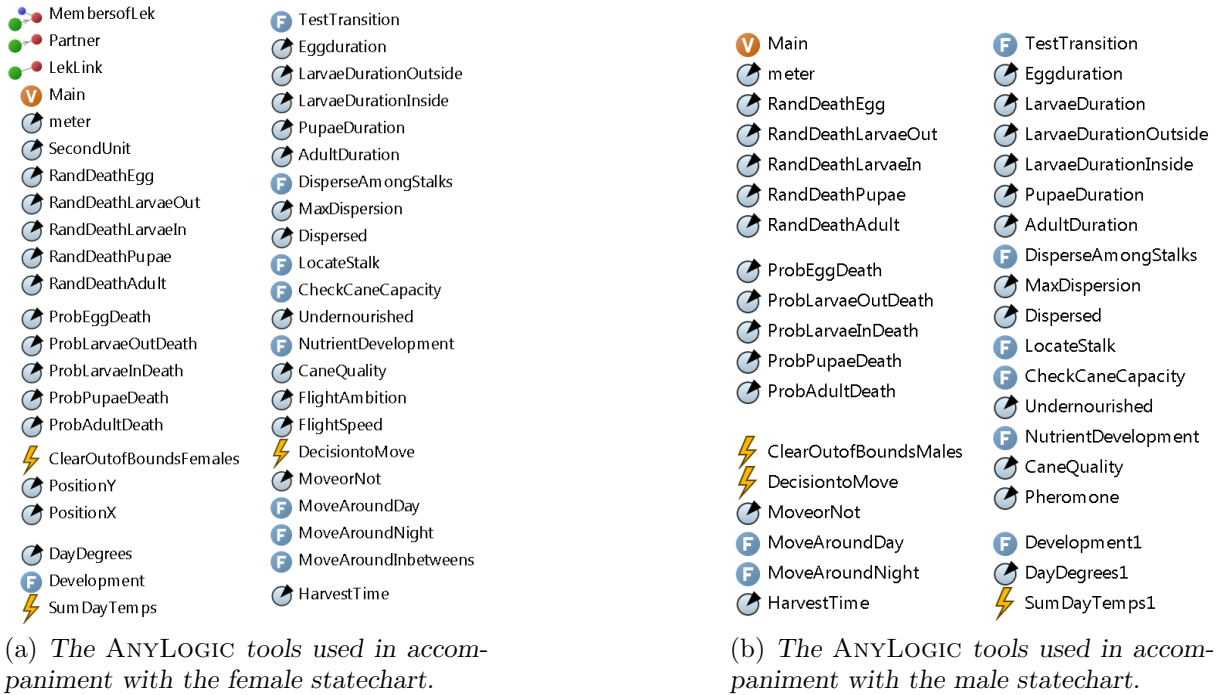


FIGURE 4.12: Sets of parameters used in an ANYLOGIC environment in the division of the simulation responsible for the accurate implementation and control of the lifecycle of *E. saccharina*.

variable to access the **Main** environment in order to obtain the value of the **DailyTemp** parameter (discussed in §4.3.2). This value is then divided by 24 to convert it to an ‘hourly contribution’ value and added to the current value of the **DayDegrees** parameter of the agent. This parameter stores the cumulative exposure to temperature and, consequently, the maturity of the pest up to the current time step. The **TestTransition** function uses the **DayDegrees** parameter to control the conditional transitions between life stages. The function determines what state the agent is in and then whether or not the accumulated day-degrees experienced by the agent is greater than the threshold value attributed to its corresponding current life stage.

In the **Egg** state, the **DayDegrees** parameter is compared to the **Eggduration** parameter to determine whether the agent should progress to the **LarvaeOutsideStalk** state. The value of the **Eggduration** parameter is drawn from a normal distribution using the mean and standard deviation values experimentally determined by Way [91].

After maturing sufficiently to progress from the **Egg** state, both male and female agents move into the larval stage of their life cycle. This stage is further divided into the larval period outside of the sugarcane stalk, as well as the period during which the larvae feed on the internal nutrients of the stalk into which they bore. Owing to the differences in behaviour and risk experienced by the larvae in each stage, a decision was made to model them as different states.

The first larval state is that of **LarvaeOutsideStalk**. In this state, the agent is once again prone to random or predator mortality. This is incorporated using the transition **RDLO** (*random death larvae outside*) which tests the value of the **RandDeathLarvaeOut** parameter against a preset probability threshold. The probability of mortality in this state is likely to be the highest since the pest is most vulnerable and unprotected in this stage of its life cycle. The value of the **RandDeathLarvaeOut** parameter is sampled once from a uniform distribution on the interval $[0, 100]$. A second transition between the **LarvaeOutsideStalk** state and the final state also

exists to facilitate the condition when the simulated temperature being experienced by the agents is too low for larvae to survive during the time period between hatching from eggs and burrowing into sugarcane stalks. This is also a condition-triggered transition occurring if the mean daily temperature being experienced in the simulation falls below 15°C, in line with the literature discussed in §2.6.

During the `LarvaeOutsideStalk` life stage, the **F** `DisperseAmongStalks` function is called to simulate the process described by Conlong [19] whereby neonate larvae move up surrounding sugarcane stalks and ‘parachute’ to other plants into which they will potentially bore. The **F** `DisperseAmongStalks` function draws a sample from a triangular distribution between zero and the maximum distance (in metres) that a larva is believed to be able to disperse⁴. This sample then determines the x and y displacement of a larva from its current location during dispersion. The new coordinates of the agent, x_{new} and y_{new} , are calculated as

$$x_{new} = U[x_{old} - (p \times m), x_{old} + (p \times m)]$$

and

$$y_{new} = U[y_{old} - (p \times m), y_{old} + (p \times m)],$$

where all values between the preset bounds have an equal probability of selection with x_{old} and y_{old} denoting the current coordinates of the agents. The maximum distance drawn from the triangular distribution is denoted by p and m is a scaling factor for converting the distances into metre values.

The **C** `meter` parameter is set to scale the visible simulation area to an equivalent hectare and is used as a reference for distance-restricted movement. Once the larvae have randomly dispersed to their specified locations, a search is performed using the **F** `LocateStalk` function to find the nearest sugarcane stalk within the underlying **V** `Sugarcane` array into which they bore. This function is called by an agent arrival-triggered transition to ensure that the new movement sequence is not initiated before the agent has completed its original dispersion. Although some literature suggests that larvae may have the ability to select sugarcane stalks based on some form of inherent intuition, the limited understanding of this process does not warrant its inclusion in the model. The simplifying assumption that a larva will simply select the nearest possible stalk for safety purposes was therefore made [19].

Upon arrival at the selected location, the agent does not move for the remainder of the time spent in this state. Progression from this state upon sufficient robustness and maturity is tested in the same manner as in the `Egg` state, with the **F** `TestTransition` function being used to determine when the accumulated **C** `DayDegrees` parameter exceeds the threshold as set by the **C** `LarvaeDurationOutside` parameter. Once this parameter is exceeded, the agent progresses to the `Larvae` state where it attempts to burrow into a sugarcane stalk.

The `Larvae` state represents the division of the *E. saccharina* life cycle where the pest has penetrated the selected sugarcane stalk and begins feeding on its internal nutrients. As in previous states, the possibility of random or predator mortality is controlled by a condition-triggered transition named RDL (*random death larvae*) which compares the value of the **C** `RandomDeathLarvaeIn` parameter against the probability of the transition being fired. The value of this parameter is, once again, sampled once from a uniform distribution on the interval [0, 100] and is evaluated upon entry to the state.

Random or predator mortality in the `Larvae` state is expected to be relatively low owing to the safety of being inside the stalk. Agents in this life stage are, however, prone to another form

⁴This value is built into the model, but the maximum value of the distribution is also a user-defined parameter.

of mortality as enforced by the carrying capacity of the sugarcane stalk into which they bore. As detailed in §4.3.2, an array similar to the **V** `Sugarcane` array also exists to track sugarcane stalk growth according to the CANEGRO model. This array, called **V** `Sugarcane_height`, stores the respective height of each stalk as it grows according to the temperatures experienced in the simulation. Furthermore, it has been observed and calculated that robust larvae require an average of approximately 40 mm of stalk length⁵ in which to bore in order to acquire sufficient nutrients for development to the pupal life stage [35, 66]. Using these two values, the number of agents each sugarcane stalk can accommodate at a given point in time in the model execution can be determined. In conjunction with this information, **V** `Sugarcane_capacity` is an array similar to **V** `Sugarcane`, which keeps track of the number of larvae that infest each sugarcane stalk at any point in time within the simulation. The capacity of a certain array is updated once a larva arrives at a respective stalk for feeding. The **F** `CheckCaneCapacity` function is called to determine whether or not that particular stalk has capacity to accommodate an additional agent. This function calculates whether the quotient of the current stalk length divided by the length required per larvae would be exceeded if the agent arriving at the stalk were to be added to the stalk's current capacity. If so, the agent does not penetrate the stalk and a transition is executed moving the agent to the death state where it is removed from the simulation. If, however, there exists capacity, the agent burrows into the stalk and that particular array point's capacity, as tracked by the **V** `Sugarcane_capacity` array, is incremented by one. The larva then survives the burrowing process and continues to exist in the `Larvae` state.




As described in §2.3, the quality of the nutrients consumed by larvae who successfully find stalks into which they can bore is understood to increase their competitiveness when they reach adulthood. In order to simulate this, a number of parameters affecting the future competitiveness of adult *E. saccharina* moths are therefore included and their values are set once an agent has successfully burrowed into a sugarcane stalk. The **F** `NutrientDevelopment` function determines the relative quality of the nutrients in the particular sugarcane stalk within the **V** `Sugarcane` array chosen by the larva in the previous state by assessing its preset maturity value. The agent's **C** `CaneQuality` parameter is then set to this maturity value. Subsequently, for female agents, the **C** `FlightSpeed` and **C** `FlightAmbition` parameters are then set and multiplied by this relative sugarcane strength. This means that female moths maturing from larvae who have fed on higher quality nutrients are likely to be larger and stronger, and will have the ability to fly further and quicker compared to moths maturing from larvae who have fed on lower quality nutrients [19]. In the case of male agents, the **C** `CaneQuality` parameter is incorporated as a scalar multiplier in establishing the strength of the **C** `Pheromone` parameter, since this is used to model the competitiveness of a particular male's display⁶.





Progression from the `Larvae` state (and, consequently, the larval life stage) occurs when the **C** `DayDegrees` parameter exceeds the threshold as set by the **C** `LarvaeDurationInside` parameter and the agents progress to the `Pupae` state. In doing so, an agent departs from what is termed in the simulation as the `EarlyDevelopment` stage. Agents who depart from this stage are assumed not to survive a harvest [19], whilst agents in this stage may still survive with a small probability. This is a user-defined probability and is compared to the value of the **C** `HarvestTime` parameter of the agent when the harvest occurs. This parameter is sampled from the interval [0, 100] and, if below the threshold probability, the agent survives the harvest. If not, a transition from the `EarlyDevelopment` state occurs and the agent is removed from the










⁵The average required length of stalk per larva is included as a user-changeable parameter.

⁶The values attributed to pheromone strengths are relative measures between males and do not represent any real actual biological value.




simulation space. The same transition exists for the `LateDevelopment` stage of an agent (representing all states post-larvae), but, as mentioned, this transition occurs for all agents in this stage when the harvest occurs.




No activity takes place in the `Pupae` state, but for the possibility of random or predator mortality, controlled by the condition-triggered RDP (*random death pupae*) transition that compares the value of the  `RandDeathPupae` parameter against the probability of random or predator mortality for this state. This probability will once again be lower than those of the other states owing to safer conditions for the agent. Agents progress from this state when the value of the  `DayDegrees` parameter exceeds the value of the  `PupaeDuration` parameter, indicating sufficient maturity to emerge as fully grown adult moths.



Once male and female agents enter the `AdultMale` and `AdultFemale` states, respectively, they begin to move randomly within their localised area, simulating walking or flying across the leaf canopy. This is modelled using an event called  `DecisiontoMove` which is a cyclically triggered event occurring, on average, twice per simulation time step. The event is called for both male and female agents and, when executed, first resets the  `MoveorNot` parameter to a sample from a uniform distribution on the interval $[0, 1]$. Thereafter, the  `MoveAroundDay` and  `MoveAroundNight` functions are called in succession. Both of these functions begin by first determining whether it is the appropriate time of day for the function to operate and, if not, do not progress further.




In the case of the female,  `MoveAroundDay` begins by determining whether the agent is in the `AdultFemale` state, as well as whether the simulated time falls between 06h00–18h00. If these conditions hold, the value of the  `MoveorNot` parameter is assessed to determine the distance over which the female may move. If the randomly sampled parameter is smaller than 0.65, no movement occurs. Values larger than 0.65, however, cause the agent to move on a scale that is proportional to the size of the sampled parameter. Similarly, the  `MoveAroundNight` function performs the same role during the simulated time period between 18h00–06h00. In this case, the movement scale is slightly larger since *E. saccharina* are more active during the night. Female agents also have a  `MoveAroundInbetweens` function which is called by the  `DecisiontoMove` event and performs a similar role to the other two movement functions.  `MoveAroundInbetweens`, however, only executes during the states when a female is waiting 24 hours post-mating, or is already engaged in the oviposition process, but is waiting until night falls before continuing to lay batches of eggs. Movement during this period is very localised and, for this reason, a separate movement function is established to replicate such small-scale movement. This phenomenon is discussed further later in this chapter. The  `DecisiontoMove` event concludes by updating the  `PositionX` and  `PositionY` parameters of the agent with its new Cartesian coordinates for use in subsequent mating and dispersal activity which occurs during the adult life stage.

The movement functions for the male operate in the same manner as those of the female, but, because the movement between the `AdultMale` and `LookingforLeks` states is already restricted by transitions dependent on the simulated time, the functions only check the current active state in an attempt to execute. Furthermore, the scales of movement are also appropriated and made smaller for males who are less frequent movers or fliers. Naturally, no function exists to accommodate movement during oviposition for the male as this agent does not lay eggs.

In the `AdultFemale` state, the  `MembersofLek`,  `Partner` and  `LekLink` links used by the female during mating are also established and cleared of any existing links to any other agents. This is of notable importance if the female returns to the adult state post-mating to

mate a second or third time.  **Partner** and  **LekLink** are single links, since the female is only ever associated with one *lek* or partner at a time, whilst  **MembersofLek** is a collection of links which operate in unison since it is responsible for connecting the female to all the members of the *lek* she is occupying. The manner in which these links operate is discussed in finer detail later in this chapter.

The  **ClearOutofBoundsFemales** and  **ClearOutofBoundsMales** are two cyclically triggered events which occur once per model time step. These events are responsible for removing an agent from the simulation if, during random movement, it exits the visible simulation space. This is achieved by assessing the agent's Cartesian coordinates and, if either the x or y coordinate of the agent falls outside of the simulation environment, it is immediately removed from the simulation. These events occur continuously throughout the simulation and remove any agent who exits the simulated area irrespective of the reason for its departure. This is of notable importance in the case of females who depart the simulation environment in search of oviposition sites.

The incorporated movement functions occur continuously during the adult states of the male and female life cycles and are only interrupted when the agents enter their mating and dispersal sub-cycles. These sub-cycles always return to the **AdultMale** and **AdultFemale/Retirement** states for the male and female agents, respectively. As with all other states, the possibility of random or predator mortality is modelled during adulthood by a condition-triggered transition **RDA** (*random death adult*) which is linked to the  **RandDeathAdult** parameter. Furthermore, the agent's death of natural causes is incorporated by a transition which compares the accumulated  **DayDegrees** experienced to the value of the  **AdultDuration** parameter and, when exceeded, moves the agent to the final state where it is removed from the simulation. For the sake of simplicity and model control, it is assumed that mortality can only occur in the **AdultMale** state for males (*i.e.* not during displaying or mating), and during the **AdultFemale** or **Laying** state for females (*i.e.* not during mating or oviposition).

4.4.3 Functionality assessment of the implemented life cycle

A series of test cases were considered to demonstrate correct implementation of some of the primary functionalities incorporated in the lifecycle building block of the simulation model. The test cases track the development and size of an initial population of 100 moths over time, without the presence of mating, until the population dies out. Simulations were run at both increased and decreased overall temperatures to assess the effect of temperature on the pests' development. Furthermore, the random/predator mortality functions were also assessed to verify that they operated as expected. It was ensured that the agents are sufficiently spaced so that the carrying capacity of the sugarcane stalks did not take effect in the simulation runs.

The random/predator mortality probabilities were set according to values proposed by Way [91] at a 26°C ambient temperature. In reality, mortality rates are actually a function of temperature and do not remain fixed. In this simulation model, the implementation of the random or predator mortality exists in the form of a transition which makes use of the accompanying probability of mortality in each life stage and is tested upon entry into each stage. This approach does not facilitate temperature-dependent mortality rates which vary over time. For this reason, the experimentally determined rates at a fixed temperature were implemented and the incorporation of a better means of modelling random and predator mortality which can make use of continually changing rates is proposed for future work.

These implemented values are shown in Table 4.3, along with the number of agents who progressed through each life stage during testing, taken as the average of 15 experimental runs at

differing temperatures. The promotion rate between states was also calculated to compare to the selected probability of death in each state.

State	Probability of death (%)	Number of agents who passed through state	Promotion Rate (%)
Egg	3	96.67	96.67
LarvaeOutsideStalk	11.5	86.33	89.31
Larvae	0.9	86.00	99.61
Pupae	0.7	85.00	98.84
Adult	20	69.00	81.18

TABLE 4.3: Probabilities for death by predator or random mortality initialised within the simulation model as described by Way [91].

As may be seen in Table 4.3, the mortality transitions were found to function as expected and the percentage of agents promoted between life stages is very close to the complement of the probability of death in each state.

In order to assess whether the effect of temperature was correctly implemented to influence the amount of time spent in each life stage, three sets of three experiments each were conducted. In the first set of experiments, temperatures were read into the simulation from a linked EXCEL worksheet without any alteration. The second set of experiments increased the read-in temperatures by a fixed amount of 7.5°C and the final set decreased read-in temperatures by a fixed amount of 3°C. For each experiment, the overall simulation time at the point of exit of each of the five states was recorded for each agent. The resulting spread of the time spent in each state at each of the three temperature conditions is shown in Figures 4.13 – 4.17. As may be seen in the figures, the experiments run at elevated temperatures (shown in blue) resulted in shorter times being required for agents to move through the life stages. By means of scale, a 7.5°C increase results in approximately 25% less time required to accumulate the required °D for a given state, assuming an average daily temperature of 25°C. The experiments at lower temperatures (shown in black) show that it took agents longer to move through the respective states, in line with expectations. In the case of temperature decrease, the results are closer to those of the normal temperature case (shown in red), because the fixed decrease was less than the fixed increase⁷. It may also be noted that, as the states progress, the disparity between the plots increases. This results from the differing accumulation of °D required by agents to progress through each state. During longer states, more time is lost or made up, depending on the temperature. Furthermore, the threshold values on each of the transitions are sampled from normal distributions determined by Way [91], meaning that in some experiments agents may have been the victims of consequent high or low samples. The difference between data sets plotted in the figures is also cumulative in nature, since it represents the total time spent passing from oviposition up to and including the state represented in the respective figure. This means that the separation between the different temperature profiles is continually growing through the progression to each state.

Upon examination, the plots demonstrate that the temperature-dependent nature of the development of *E. saccharina* is successfully implemented into the model framework.

The next test case was conducted to verify that the effect of the carrying capacity of the sugarcane was correctly implemented in the simulation model. Both the initial height of the sugarcane stalks and the average length of feeding required per larva are included as user-defined

⁷This was done to avoid temperatures dropping so low that larvae simply perish.

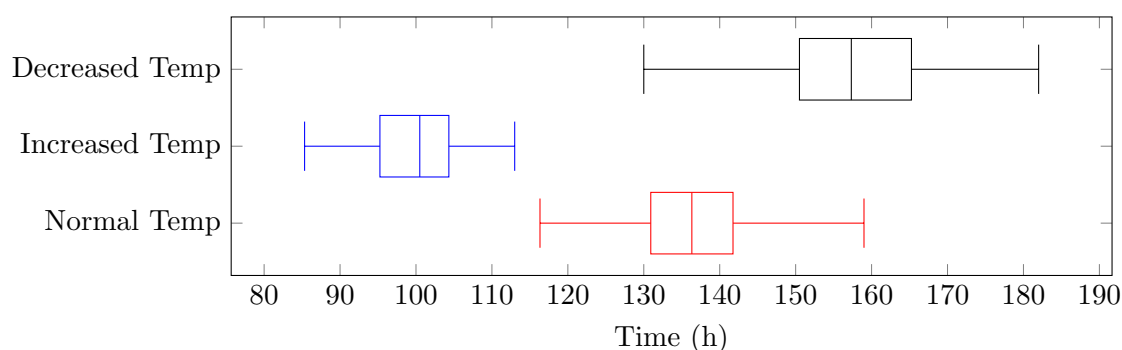


FIGURE 4.13: The difference in time taken for agents to progress through the egg life stage using averages determined by three experiments with a starting population of 100 agents.

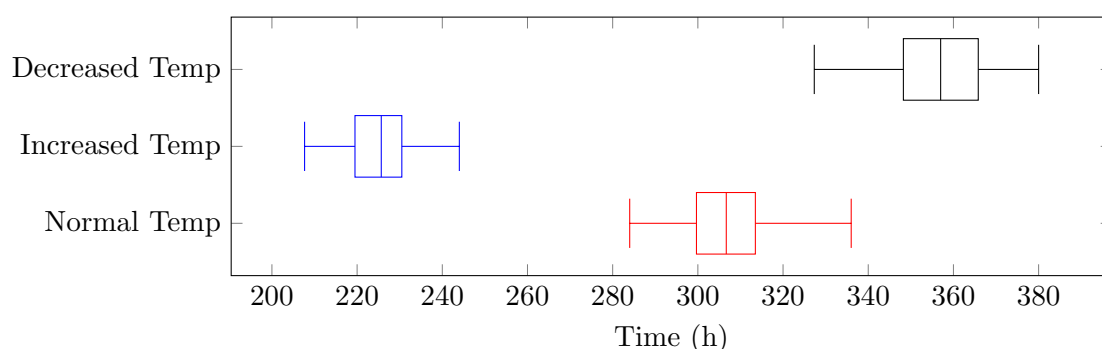


FIGURE 4.14: The difference in time taken for agents to progress through the larvae outside stalk life stage using averages determined by three experiments with a starting population of 100 agents.

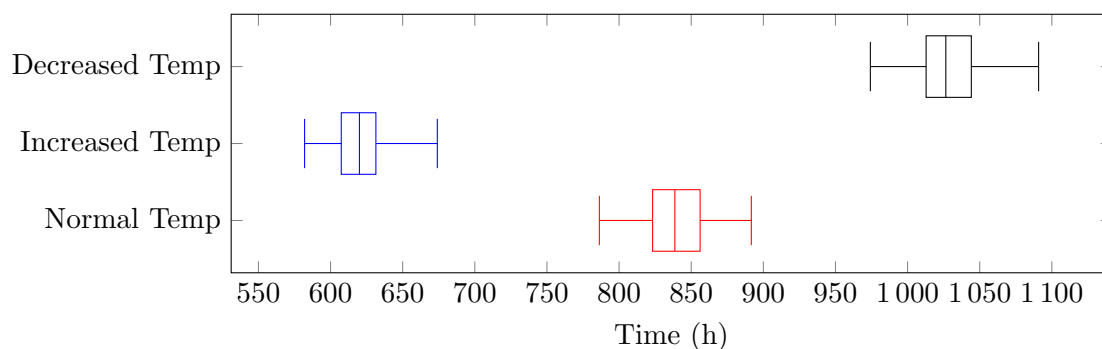


FIGURE 4.15: The difference in time taken for agents to progress through the larvae inside stalk life stage using averages determined by three experiments with a starting population of 100 agents.

parameters in the simulation model. These values were altered in a number of simulation runs to test the level of infestation the sugarcane stalks could carry by virtue of their current height. In the simulation runs, fifty agents were all directed to infest the same sugarcane stalk and all random or predator mortality probabilities were set to zero. The number of agents who survived the carrying capacity restrictions was then recorded and compared to the expected outcomes. The results of the experiments are summarised in Table 4.4.

As may be seen from the results, the carrying capacity was correctly implemented in the simulation model. The approximate theoretical capacity is given as a range between the time of the first larval infestation of a stalk and the last. There may be sufficient time during these two instances for further growth of the stalk to accommodate another agent. In cases where the actual total capacity (the number of agents surviving) matches the lower bound on the range, the stalk did not have sufficient time between infestations to grow and accommodate more agents.

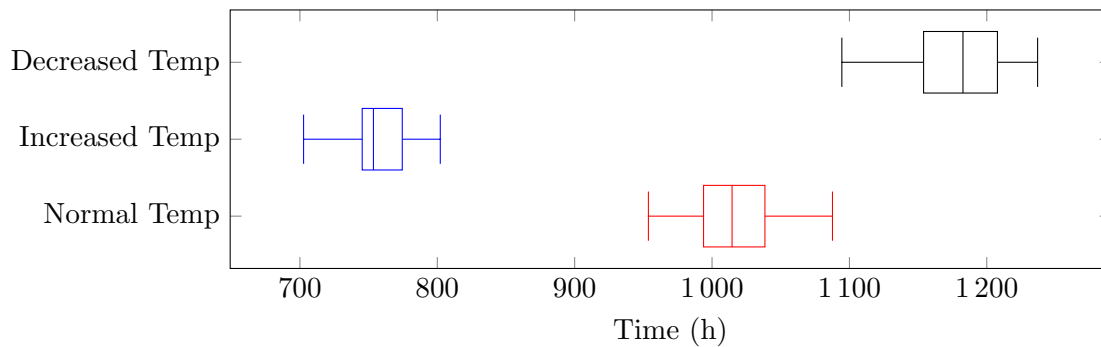


FIGURE 4.16: The difference in time taken for agents to progress through the pupae life stage using averages determined by three experiments with a starting population of 100 agents.

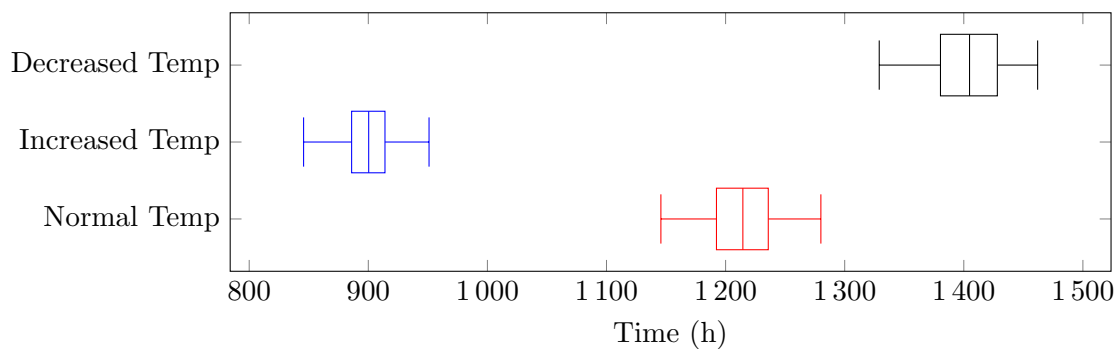


FIGURE 4.17: The difference in time taken for agents to progress through the adult life stage using averages determined by three experiments with a starting population of 100 agents.

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Initial population size	50	50	50	50	50	50
Length of stalk required per agent (<i>mm</i>)	40	5	40	5	40	5
Initial stalk length (<i>mm</i>)	0	0	50	50	100	100
Stalk length at first agent penetration (<i>mm</i>)	27	26	75	76	126	126
Approximate theoretical capacity ($\frac{\text{stalklength}}{\text{lengthrequired}}$)	0	5–6	1–2	15–16	3–4	25–26
Actual total capacity (<i>Number of agents surviving</i>)	0	6	2	16	3	25

TABLE 4.4: Results of a variety of tests to determine the accuracy with which the simplified sugarcane carrying capacity functionality was implemented in the simulation model.

A final test was conducted to verify that the harvesting function operated as expected. It is assumed that only a very small percentage of eggs and larvae will survive a harvest. This percentage is included in the simulation as a user-defined parameter, but was set at 3% for this experiment. To test the functionality, the simulation was run for a two-year cycle to allow the population to grow and spread. At the end of each year, a harvest was scheduled to be executed. The resulting population numbers for the tests are shown in Tables 4.5–4.6. As may be seen in these tables, approximately 3% of the total population remains in both cases after a harvest and all remaining agents reside either in the Egg or Larvae states. The harvesting function therefore operates as expected.

State	Number of agents pre-harvest	Number of agents post-harvest	Percentage remaining
Egg	4 012	47	1.17%
LarvaeOutsideStalk	32 749	914	2.79%
Larvae	1 946	143	7.35%
Pupae	0	0	0.00%
Adult	210	0	0.00%
All states (total)	38 917	1 104	2.84%

TABLE 4.5: The remaining agents per life stage, as well as the percentage of the original number of agents in each life stage remaining after a harvesting event was executed one year post initialisation.

State	Number of agents pre-harvest	Number of agents post-harvest	Percentage remaining
Egg	0	0	0.00%
LarvaeOutsideStalk	3 016	35	1.16%
Larvae	7 957	262	3.29%
Pupae	0	0	0.00%
Adult	0	0	0.00%
All states (total)	10 973	297	2.71%

TABLE 4.6: The remaining agents per life stage, as well as the percentage of the original number of agents in each life stage remaining after a harvesting event was executed one year after the previous harvesting event.

4.5 The mating process of *E. saccharina*

The mating process of *E. saccharina* is of notable importance as it is responsible for the spreading of the pest within sugarcane. Furthermore, pest control strategies, such as the *sterile insect technique*, affect the pest during mating [65] and it was therefore deemed vital to reconstruct the mating process in the simulation model with sufficient accuracy so as to mimic the in-field observations made by Atkinson [5] and Carnegie [17]. In all previous models of *E. saccharina*'s population growth and dispersal, an aggregation approach was followed which excluded small-scale intricacies such as the *lek* mating and mate selection processes. The facilitation of these processes may, however, significantly improve the accuracy of simulating *E. saccharina*'s population dynamics.

4.5.1 General description of the implemented mating process of *E. saccharina*

The logical progression followed and assumptions made to effectively implement *E. saccharina*'s mating process in the simulation model are summarised in Figure 4.18. This process is presented in detail by van Vuuren *et al.* [87].

Once a male moth has reached adulthood and the simulated time of day is appropriate for mating activity to commence, he attempts to create a *lek* together with males in his immediate vicinity. This time window is assumed to be 18h00–23h00 [5]. If there are 3–6 males in close

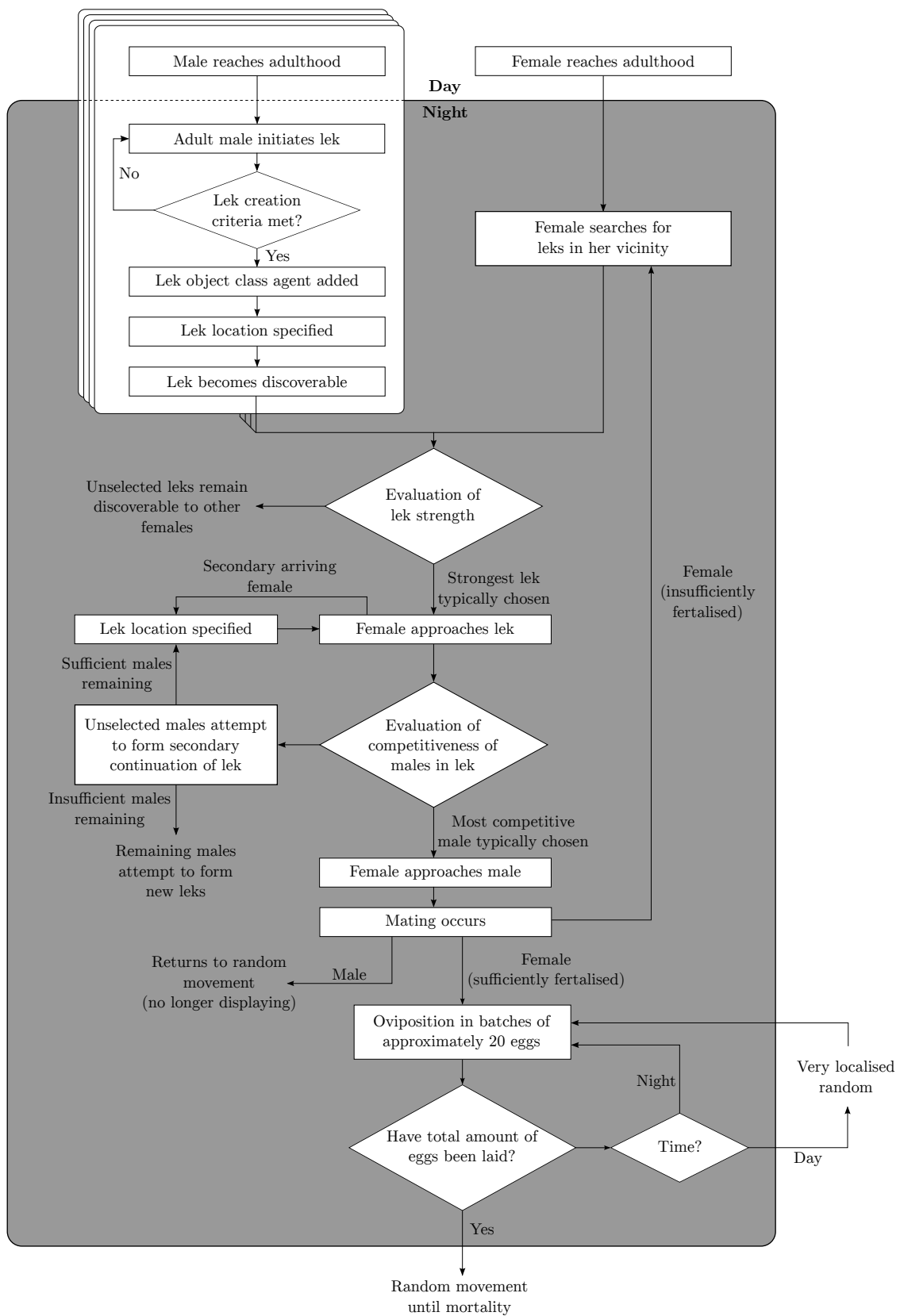


FIGURE 4.18: Flowchart of the general mating process, including assumptions, followed by *E. saccharina*.

enough proximity, a *lek* will be formed and the mating process will progress to the next stage⁸. If a male is drawn into joining a *lek* which was initiated by another male nearby, he loses the ability to initiate his ‘own’ *lek* until such time as the *lek* to which he belongs ceases to exist. This may occur due to a lack of sufficient males nearby, meaning that the *lek* never actually materialises, or it may materialise, but then dissipate once the female has chosen and approached her preferred partner from the *lek*.

When a *lek* is successfully initiated and a group of males become members of it, its location is assumed to be specified as the centroid of the males included in the *lek*. If n males with Cartesian coordinate positions $(x_1, y_1), \dots, (x_n, y_n)$ participate in the *lek* (when viewed from above), this point is (L_x, L_y) , where

$$L_x = \frac{1}{n} \sum_{i=1}^n x_i \quad (4.3)$$

and

$$L_y = \frac{1}{n} \sum_{i=1}^n y_i. \quad (4.4)$$

Once the *lek* has been located at this point, it becomes discoverable to females who are ready to mate and may potentially be attracted to it.

In a full-scale simulation, several *leks* are likely to materialise in the vicinity of a female at a given point in time. The simulation assumes that the female is aware of and attracted to all *leks* in her vicinity, but the *lek* which has the strongest attraction, based on the total pheromone strength of all the males in the *lek*, as well as its distance from her, has the highest probability of being selected.




It is possible that two or more females in the simulation may be attracted to and begin to approach the same *lek*. In accordance with the standard assumptions of a Poisson arrival process [92], no two females will arrive at a *lek* at precisely the same instant during the simulation. When a female arrives at a *lek*, she chooses a mate (assumed to be the most competitive male) with a certain probability⁹ and occupies this mate during the entire mating process. As a result, the male no longer participates in the *lek* [19]. As soon as the female has selected her mate, the other members of the *lek* are ‘freed’ and once again become available to participate in a remaining *lek* (provided that a sufficient number of unmated males remain). This weaker *lek* essentially functions as a continuation of the previous *lek* and its centroid is typically very close to that of the previous *lek*. As a result, other females already in the process of approaching the original *lek* move further to this new centroid and select a mate from the remaining males who participate in that *lek*. This may occur several times until fewer than three males remain as participants of the remaining *lek*, at which point, it dissipates — no longer attracting females [19, 61].


Mating ensues for two to three hours. Due to the limited time available per night for mating, it is assumed that males will only mate once per night. After mating, the female departs in order to prepare for oviposition. In line with assumptions made in the literature, the female only oviposits batches of eggs during the early hours of the night and, during the day, will remain relatively static, moving randomly on a very small scale (resembling walking), until such time as she can oviposit again. Once the female has laid all of her eggs, she typically does not partake in any further activity and perishes after a few days.

⁸This proximity is included as a user-changeable parameter.

⁹This probability is included as a user-modifiable parameter.

4.5.2 Description of the implemented mating process in an ANYLOGIC modelling environment

Successful implementation of the *E. saccharina* mating process in the ANYLOGIC modelling environment required the use of three independent object classes, namely  **male**,  **female** and  **Leks**.

A screenshot of the division of each object class' lifecycle in the ANYLOGIC project responsible for mating is shown in Figures 4.19(a), 4.20(a) and 4.21(a), along with the accompanying model building components shown in Figures 4.19(b), 4.20(b) and 4.21(b) which are employed in conjunction with each statechart to effectively implement the mating process. In the case of the  **Leks** object class, the division responsible for mating represents the entire statechart and all model building components, since this object class is only implemented during mating.

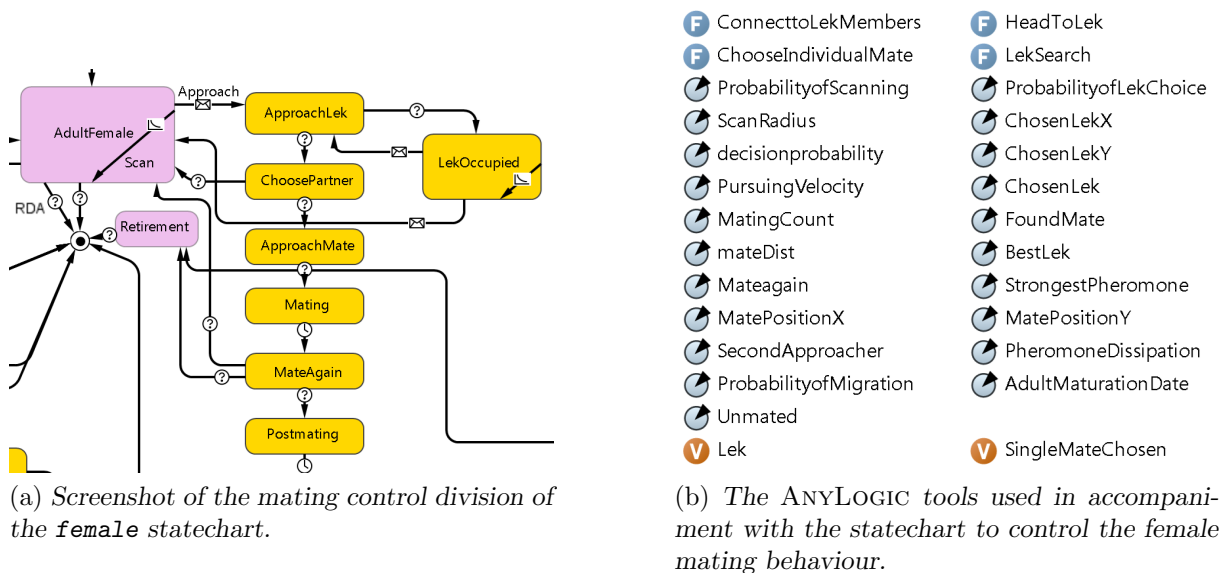


FIGURE 4.19: Inner working of the female mating behaviour control.

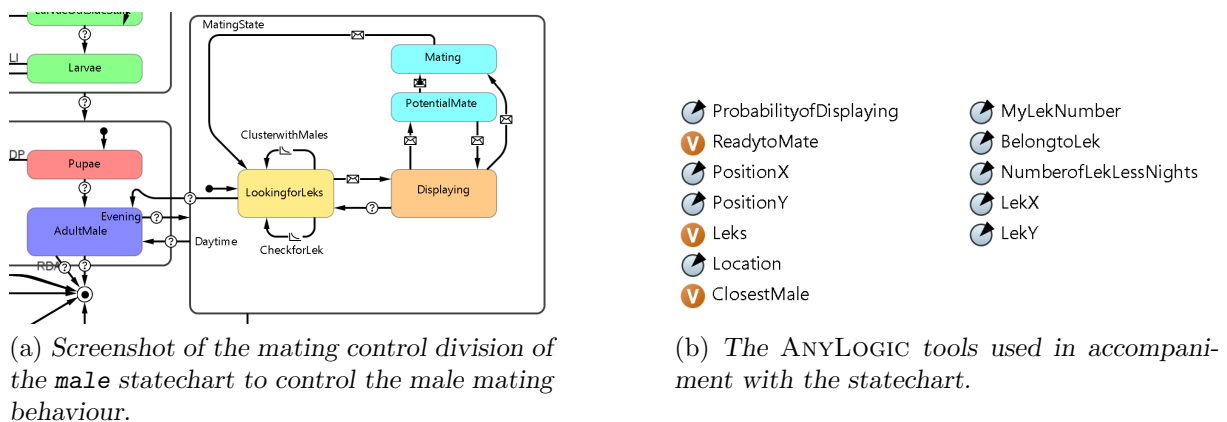





FIGURE 4.20: Inner working of the male mating behaviour control.

In the simulation,  **male** and  **female** object classes exist to facilitate separate control of both sexes of *E. saccharina*, since their behaviour varies distinctly. The  **Leks** object class was also incorporated to assist in the creation and control of *leks* in the simulation model.

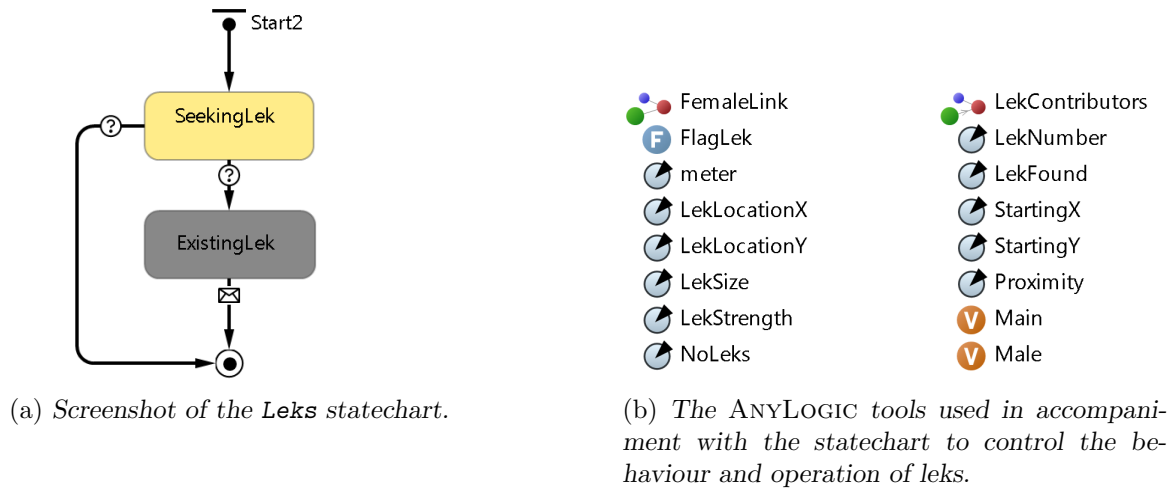











FIGURE 4.21: Inner working of the lek agent control used in the mating process.

As a result of using continuous space in the simulation, it is not possible for females to search in defined, discrete locations in an attempt to identify *leks* for mating. Furthermore, for the purposes of *lek* selection by females, specific attributes, such as a pheromone strength and positional coordinates, are required to be possessed by *leks*. In the case of displaying males, an entity is required to which they can ‘belong’ so as to ensure that they do not form part of multiple *leks* at any given time within the simulation, which would result in a misrepresentation of the number of females who can feasibly mate per night. In light of these requirements, defining  **Leks** as a separate object class accommodates superior control and efficiency within the model. An individual  **Leks** object is, however, a ‘hidden’ attribute of the final simulation and only contributes to control on the back-end of the simulation model.

Upon reaching adulthood, individual males are responsible for initiating *leks* during night-time displays. As may be seen in Figure 4.20(a), male moths progress from the **AdultMale** state of random movement to the **LookingforLeks** state at the appropriate time of day. In this state, two rate-triggered transitions are called continuously by the statechart. **ClusterwithMales** is a function which prompts the male to seek another male within close proximity. The searching male moves toward the nearest other adult male in an attempt improve the chances of forming a competitive *lek* to attract females. The second transition, **CheckforLek**, creates a new agent of the  **Leks** object class in an attempt to initiate a *lek* with these nearby males. The male agents are responsible for the initial addition of an agent to the  **Leks** object class only; thereafter, they do not operate or control the *lek*’s behaviour in any way. Since both the **ClusterwithMales** and **CheckforLek** transitions are sourced from the **LookingforLeks** state, they cannot be executed simultaneously and will wait until the male has either completed approaching fellow males, or has unsuccessfully initiated a *lek* (hence remaining in the **LookingforLeks** state) before either of the transitions are called again. Only males who do not currently belong to *leks* possess the ability to initiate them since the successful execution of a *lek* causes a male to progress to the **Displaying** state where the **CheckforLek** transition is no longer triggered. As may be seen in Figure 4.20(b), each male has a  **BelongtoLek** parameter. This is a Boolean parameter which is set to **false** as long as the male does not belong to a *lek*.



When the **CheckforLek** transition is called and the male generates the addition of a new object to the  **Leks** object class, the individual  **Leks** ‘agent’ immediately enters the **SeekingLek** state, as may be seen in Figure 4.21(a). The two parameters,  **StartingX** and  **StartingY**, are set by the male who initiates the *lek* to his current Cartesian coordinates. Upon entering the



SeekingLek state, the *lek* is instantly relocated to these coordinates. This ensures that, when seeking males in the surrounding area to form part of this attempted *lek*, the male who originally initiated it will definitely form part of the *lek*, should it materialise. Furthermore, the proximity of males is taken relative to the male who attempts to initiate a *lek*. In the **SeekingLek** state, the *lek* is not discoverable to females as it has not yet technically been ‘formed’. In the **SeekingLek** state, the **F** **FlagLek** function is called. This function begins by searching through all of the males in the simulation and determining which males are in the **LookingforLeks** state and, as a result thereof, may form part of the *lek*. The function then proceeds to consider all of these males, confirming whether or not they are part of a *lek* by evaluating their **B** **BelongtoLek** parameters, as well as determining whether or not they are within close enough proximity to form part of this particular *lek*. This is done by comparing the distance from the *lek* to a male’s preset **P** **Proximity** parameter, specifying the maximum distance allowed between any two agents¹⁰. If both of these conditions are met, a final test is performed to confirm that the *lek* has not, as of yet, reached its capacity of males. If not, the male being assessed will be added to the *lek*. The *x* and *y* coordinates of this newly added male are then contributed to a running tally of all Cartesian coordinates of all of the males contributing to the *lek* to determine its location, according to (4.3) and (4.4). Furthermore, the **L** **LekStrength** parameter of the *lek*, storing its relative competitiveness, is updated by adding the **P** **Pheromone** strength of the male to this **L** **LekStrength** parameter. After this, the **M** **MyLekNumber** parameter of the male is updated and set to equal the **L** **LekNumber** of the *lek*. This is a unique, sixteen-digit number used to ‘brand’ males as part of a particular *lek* and simplify communication and organisation between them. They are also associated with a particular *lek* via the **C** **LekContributors** connection. Finally, the **L** **LekSize** parameter is incremented once to keep track of the current size of the impending *lek*.

Once all of the males have been considered, the size of the *lek* is assessed. If the **L** **LekSize** counter is found to be larger than or equal to three, the *lek* is large enough to ‘materialise’ and, in doing so, become discoverable to females. The *x* and *y* coordinates of the *lek* are calculated and assigned to **L** **LekLocationX** and **L** **LekLocationY**, respectively. Thereafter, a message is sent via the **C** **LekContributors** connection which prompts males to move from the **LookingforLeks** state to the **Displaying** state, as they now form part of a *lek* and can begin attempting to attract a female. This transition in the male statechart is accompanied by an action which sets the male’s **B** **BelongtoLek** parameter to **true**. The **L** **LekFound** boolean parameter within the statechart of the *lek* itself is then also set to **true** which triggers the transition between the **SeekingLek** and **ExistingLek** states, causing the *lek* to ‘materialise’ in the simulation space and thereby become discoverable to females.



If, after searching through all of the males in an attempt to form a *lek*, the **L** **LekSize** counter value is still found to be less than three, the **N** **NoLeks** Boolean parameter is set to **true**, triggering the transition which skips the **ExistingLek** stage and immediately terminates the *lek*. This means the *lek* does not materialise and is never discoverable to females. In the termination state, the **T** **Leks** agent is removed from the object class and any male agents who were associated with the *lek* via the **C** **ContributorsofLek** connection are disconnected. Furthermore, as a safety precaution, a loop is run during *lek* termination which searches for any males in the simulation whose **L** **LekNumber** has been set to that of the *lek* which is being removed. If any are found, their **B** **BelongtoLek** parameter is set to **false** (in case, for some reason, it was set to **true** during the attempted *lek* creation process). This ensures that these males then become eligible to initiate or form part of *leks* once again.









¹⁰This distance is included as a user-changeable parameter.

A counter parameter called  **NumberOfLekLessNights** is incorporated to track the number of nights a male goes without successfully forming part of a *lek*. This counter is incremented each time the male returns from the **LookingforLeks** state to the **AdultMale** state, since it means the male was not part of a *lek* by the time the mating time period ended. If the counter reaches three, meaning the male goes three nights without forming part of a *lek*, a random movement sequence is run which moves the male on a larger scale than the localised random movement typically experienced by adult males during the day. This follows intuition in that, if the male is unable to form a *lek* he is most likely to undertake some form of dispersal flight in the hope of finding a more densely populated region where he may find other males to display with and, consequently, a mate.  **NumberOfLekLessNights** is reset to zero each time the male enters the **Displaying** state, since this only occurs if he does indeed form part of a *lek*.





As may be seen in Figure 4.19(a), the female performs a scan function upon reaching the **AdultFemale** state. This scan is performed, on average, five times per simulated hour. This discrete number of search instances was chosen to reduce computational expense when a large number of females are present in the simulation. This scan frequency was also deemed sufficient since the random movement by male agents is small-scale and, as a result, their arrangement (and, therefore, the possibility of new *leks* being formed) does not change significantly over small time intervals. The transition triggers execution of a function called  **LekSearch**. The logic sequence of this function begins by first determining whether it is an appropriate time for mating to ensue. If so, a test is performed in respect of each *lek* to determine whether enough males are part of the *lek*, as well as whether or not the *lek* is sufficiently close to the female. If both of these conditions are met, the pheromone strength of the *lek* is scaled as a function of its distance from the female. The scaling of the *lek* strength value accounts for dissipation of the pheromone strength with distance from the female. This is caused by physical barriers (such as plants), air resistance and wind between the female and the *lek*, and may result in selection of a weaker *lek* by the female which is closer to her, instead of a stronger *lek* further away [19]. Little is known about the finer workings of pheromone dissipation and, for this reason, the strength of the pheromone as perceived by the female is assumed to decay at an exponential rate with respect to distance. For a *lek* with a strength equal to the *lek*'s unscaled  **Lekstrength** parameter L_u , the relative scaled strength of the *lek* being assessed, L_s (referred to *ScaleLek* in the simulation model) is calculated using the formula





$$L_s = L_u \lambda e^{\lambda d},$$




where λ , modelled by the  **PheromoneDissipation** parameter, refers to the rate of decay of the exponential function and d refers to the straight line distance from the female to the *lek*. Owing to a widespread lack of knowledge in this area of the *E. saccharina* life cycle, the rate of decay is chosen to be unity. Since the female does not consider *leks* which fall outside of the user-defined  **ScanRadius** parameter, the relative strength of the *lek* is bounded from reaching all females, in spite of the asymptotic nature of the exponential function.

A test is then performed to determine whether the scaled value is the strongest *lek* which has been found so far by comparing it to the value of the  **BestLek** parameter. This parameter is used to keep track of the current strongest *lek* discovered. If it surpasses the value of competitiveness of the current strongest *lek* found and the probability threshold is met¹¹, the  **BestLek** parameter is updated with this value and the  **ChosenLek** parameter is updated with the *lek* number, drawn from the  **LekNumber** parameter in the  **Leks** object class. Similarly, the  **ChosenLekX** and  **ChosenLekY** parameters are set to the  **LekLocationX**






¹¹The probability threshold is incorporated to model the possibility of a female choosing not to approach a certain *lek* or, perhaps, choose a less competitive *lek* over a more competitive one for no specific biological reason.




and  **LekLocationY** parameters of the  **Leks** object class agent so that the *lek* can later be located, in the event that no stronger *leks* are found and selected. Finally, the female disengages from the previously chosen *lek* and engages with the new *lek* via the  **LekLink** connection. This connection remains until disengaged by the female, either to replace it with a stronger *lek*, or once the *lek* has been occupied. During *lek* occupation, the female sends a message to the  **Leks** agent which triggers a transition in its statechart and it is removed from the simulation.









If a *lek* is successfully found by the female, the function concludes by triggering the **Approach** transition to move the female to the **ApproachLek** state. If not, the female remains in the **AdultFemale** state, moving randomly on a small scale until the **Scan** transition is triggered again to retest for *leks* in her vicinity. In the **ApproachLek** state, the  **HeadToLek** function is called which sets the female in flight with a speed equal to her  **FlightSpeed** parameter and instructs her to move from her current location to the *lek* defined by the  **ChosenLekX** and  **ChosenLekY** parameters set in the previous step.



In the event that more than one female approaches the same *lek*, the first arriving female will ‘occupy’ the *lek* and transition to the **ChoosePartner** state. Since the **Scan** function is called at discrete, random time instants (on average, five times per model time step), females will detect *leks* at different times and be different distances from these *leks*. In light of this, it is not always the first female to detect a *lek* who will arrive there first. Once the first female ‘occupies’ the *lek*, the  **LekLink** connection between all other approaching females and that particular *lek* is broken, and these females transition to the **LekOccupied** state. In this state, a female continues on her original flight path to arrive at the location of the *lek* originally identified. Here the **LekSearch** function is called in rapid succession (within a smaller search radius) to force the female to identify a new *lek* which may materialise from the remaining males, if enough remain to form a competitive *lek*. This ‘new’ *lek* functions fundamentally as the same *lek* as that originally approached. This management of the multiple arrival instances allows for the necessary amount of control over the remaining agents and excludes the male chosen for mating from the previous *lek*. The female will then approach this ‘new’ *lek* by returning to the **ApproachLek** state as before. A control parameter, called  **SecondApproacher**, is incorporated to inform the female in the event that insufficient males remain from the original *lek* which she was approaching and it dissipates. This triggers a transition back from the **LekOccupied** state to the original **AdultFemale** state if the  **LekSearch** function is called unsuccessfully a threshold number of times, indicating no secondary *lek* has formed. When a female is first to arrive at a *lek* and ‘occupies’ it, a conditional test is performed to ensure that her coordinates match those of the *lek*. If the match is successful, she transitions to the **ChoosePartner** state.




During the simulation, adult male moths move randomly within their local area. This random movement means that they have the ability to leave or join *leks* during the mating time window. Once part of a *lek*, males are assumed to be primarily static and the random movement in the simulation subsides.

In the **ChoosePartner** state, two functions are called consecutively, namely  **ConnecttoLekMembers**, followed by  **ChooseIndividualMate**.  **ConnecttoLekMembers** is responsible for connecting the female to the group of static males who compose the selected *lek*. The logic sequence of the function begins by connecting to the males who form part of the selected *lek* by exploiting the  **ContributorsofLek** connection which exists between the *lek* and associated males. Any males who are part of this connection then become associated with the female *via* the  **MembersofLek** connection and are sent a message through this connection which triggers a transition in the male statechart, moving them from the **Displaying** state to the **PotentialMate** state.

Once all the males comprising the chosen *lek* have been associated with the female and have moved to the `PotentialMate` state, the female sends a message to the  `Leks` agent which is currently occupied *via* the  `LekLink` connection, triggering a transition from the `ExistingLek` to the termination state. This removes the *lek* and sets the  `BelongtoLek` parameter of all males who contribute to the *lek* to `false`. In spite of this, all males who compose the *lek* are now in the `PotentialMate` state and therefore cannot yet begin forming new *leks*.

 `ChooseIndividualMate` is then called to determine the most competitive male in the *lek* for mating purposes. This function follows a similar logic to the  `LekSearch` function but, in this case, it searches through all male agents in the simulation, determining whether they are in the appropriate life stage for mating and, if so, testing again whether or not they belong to the *lek* chosen by the female. If these conditions are met, the strength of each male's pheromone is evaluated and compared to that of the other males comprising the *lek*. The current strongest pheromone is stored by the female using the  `StrongestPheromone` parameter. If the pheromone strength of the male is stronger than the previously strongest pheromone encountered by the female and a probability threshold is exceeded¹², she disengages from her previous best partner, connects to this male *via* the  `Partner` connection and saves the partner's *x* and *y* coordinates in the  `MatePositionX` and  `MatePositionY` parameters, respectively. This connection overrides the previously established  `MembersofLek` connection that the female has with all males forming the *lek* and allows for a direct message line between the female and chosen male. Once the function has considered all of the males who contribute to the *lek*, a message is sent *via* the  `Partner` connection to the chosen male, triggering his transition from the `PotentialMate` state to the `Mating` State.

As the female approaches her chosen mate, she disengages from all the other males forming part of the *lek* by breaking the  `MembersofLek` connection. Before this connection is broken, a message is sent to the *lek* members, triggering their transition back to the `Displaying` state. Here, since the  `BelongtoLek` parameter was set to `false` when the *lek* was removed, a second transition is immediately triggered, moving the male from the `Displaying` state to the `LookingforLeks` state where they can, once again, form secondary or new *leks*. If, by the time they reach this state, the mating time has expired, a final transition is triggered which moves the males to the `AdultMales` state where they return to small-scale random movement until the next mating window period.

When the female arrives at the coordinates, mating ensues for a randomly selected time period of between two and three hours before proceeding to the oviposition and consequent dispersal division of the simulation. Once mating is complete, the female sends a message to the male with which she mated, triggering a transition in his statechart from `Mating` state back to the `Displaying` state and breaking the  `Partner` connection. The transition resets the  `BelongtoLek` to `true` (it was set to `false` when the original *lek* was removed). This means the mated male cannot form part of a new *lek* until the mating time period expires and the transition back to the `AdultMale` state where the  `BelongtoLek` is, once again, set to `false` to allow for *lek* formation on subsequent nights.

In some rare instances after mating, the eggs present in the abdomen of the female have not been sufficiently fertilised during mating and further fertilisation is required. If this is the case, when the female moves from the `Mating` to the `MateAgain` state, there is a small possibility that she is re-routed back to the `AdultFemale` state to begin searching for another mate. This process of 're-mating' may happen up to three times in exceptional circumstances. If, after three attempts




¹²The motivation for inclusion of this probability threshold is the same as in the case of the *lek* search, as well as the fact that it accurately incorporates the possibility of females competing for specific males within *leks*.

at fertilisation, the female is still not able to oviposit, the oviposition cycle is skipped and she moves to the **Retirement** state until mortality, never ovipositing any eggs. Once the female is sufficiently fertilised after any number of mating instances, she progresses to the **Postmating** state to begin the gestation period.

4.5.3 Verification of the implemented mating process functionality

In order to determine the success with which the mating process of *E. saccharina* has been implemented, a comparison was performed to determine the accuracy with which the important mating aspects described in §2, as well as the assumptions mentioned in §4.5.1, were incorporated into the model. The sequence of a typical mating instance in the simulation is illustrated by a collection of images in Figure 4.22. In the figure, male agents are represented as blue squares, whilst female agents are coloured pink.

While weather effects (other than temperature) are not considered in the model, the assumed time at which mating activities occur, as well as the assumed sequence of events (derived from the literature in §2), was incorporated. Note that, for the purposes of observation and verification, several aspects have been included in the simulation which would typically not form part of the final model output. For example, male agents change colour depending on their respective states in order to facilitate observation of the effects and execution of the associations between the males and the mating female. During the displaying stage, males are coloured orange (indicating entry into the **Displaying** state), although not captured in any of the images, the potential mates identified as contributing to a *lek* briefly turn light blue (indicating entry into the **PotentialMate** state), and the male selected from the *lek* as a mate by the female remains light blue for a prolonged time period (indicating entry into the **Mating** state). The remaining males return first to the **LookingforLeks** state and then, since three or more remain, a secondary *lek* is immediately formed, progressing them to the **Displaying** state once again.

Furthermore, the pheromone strength of a male moth is shown as a number attached to the corresponding agent (these are simply representative values and are not attributed any specific unit). The centroid location of *leks* which are currently discoverable by females in the simulation are shown as black squares and the ‘strength’ of the *lek* (by virtue of its composing agents) is shown as a bold sum value above the *lek* element. The female’s search radius boundary is included as a wide-spaced dotted black line and the  **LekLink**,  **MembersofLek** and  **Partner** links are shown as a red dashed line, a fine-spaced black dotted line and a solid black line, respectively. The females in this test run are instructed always to choose the strongest male.

Referring to the various parts of Figure 4.22, the following interpretations are offered:

- (a) All of the agents are in the pupae stage and, as a result, remain static in their respective locations.
- (b) Four of the males and both females have reached adulthood, but no mating occurs since it is still daytime. It may be seen that the arrangement of the agents differs slightly from that in Figure 4.22(a). This is due to random movement by agents within their local area during the adult phase. The boundaries of the females’ search radii may be seen in the top right-hand corner of the figure.
- (c) Night falls and the four mature males have transitioned to the **LookingforLeks** state. Here they were found to be in close enough proximity to form a *lek* and, as a result, the first

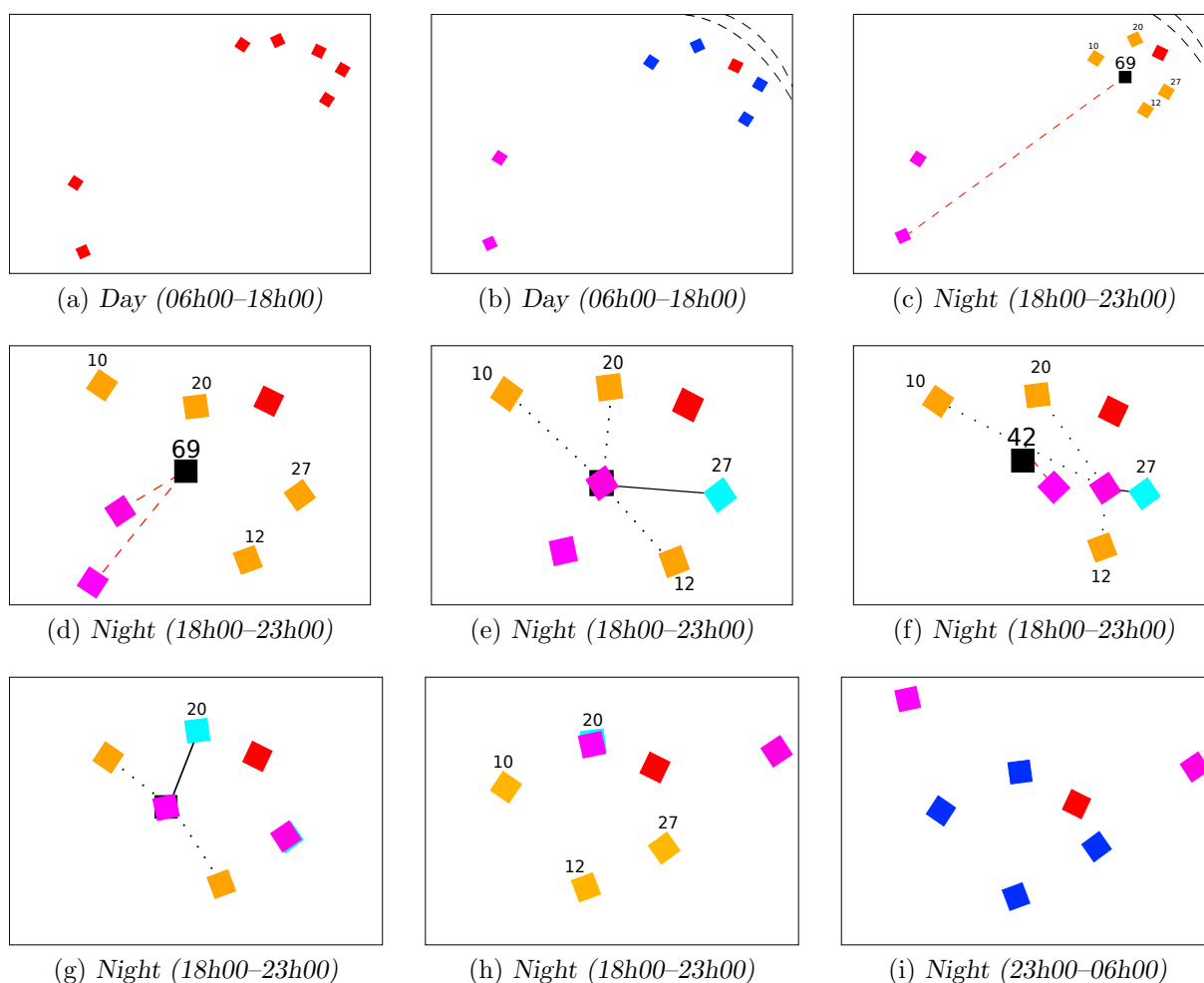




FIGURE 4.22: A mating sequence executed in the ANYLOGIC simulation.

attempted addition of an agent from the Leks object class has successfully materialised and progressed to the `ExistingLek` state to become discoverable to females. One of the females has already discovered this *lek*.

- (d) Both females have discovered the *lek* and are heading towards it. It is evident that the female who was closer to the *lek* initially will arrive first.
- (e) The first female to arrive at the *lek* has occupied it, identified the most competitive male in the *lek* and connected to him as her mate. The male has a pheromone strength of 27, has progressed through the `PotentialMate` state and now resides in the `Mating` state — hence the light blue colouring of the agent. The other males have already entered the `PotentialMate` state and then returned to the `Displaying` state and, further, to the `LookingforLeks` state where they are attempting to form a secondary, resulting *lek*. The second female has now moved to the `LekOccupied` state and is searching for the *lek* which may result from the remaining males of the original *lek*.
- (f) Now that the first female is approaching her mate, she has executed a transition to remove the *lek* she occupied. This causes the `BelongtoLek` parameter to be set to `false` which means that the males are now able to form part of a new *lek*. Since three unmated males still remain in close proximity from the previous *lek*, a second *lek* immediately materialises

nearby. As may be seen, this *lek* has an inferior strength to that of the original *lek* as one male is no longer contributing to the display.

- (g) The first female is in the process of mating with her partner and, in the meantime, the ‘new’ *lek* location is being approached by the second female who originally approached the first *lek*. She then chooses the most competitive remaining male and engages it as her partner.
- (h) The first female has completed the mating process and has now moved to the oviposition state where she will begin laying eggs after about 24 hours. The mated male has returned to the **Displaying** state, but a further *lek* does not transpire owing to the fact that his  **BelongtoLek** parameter was set to **true** during the transition and, as a result, no further *lek* has formed since there remain insufficient displaying, unmated males from the original *lek* to contribute towards continuing the *lek*. The second female is in the process of mating at this point.
- (i) The time window for mating has expired and the males are no longer displaying. In the **AdultMale** state, all males’  **BelongtoLek** parameters are reset to **false** so as to ensure that they will be able to form *leks* during the next mating time period. All agents also return to random movement and females do not lay eggs until the following evening, in the meantime moving only over very small distances.

4.6 The oviposition and dispersal of *E. saccharina* female moths

The largest propagator of *E. saccharina* infestation in sugarcane is considered to be the oviposition and dispersal flights of female moths. It may cause both a population explosion in a certain area where localised oviposition occurs, or spread a population to previously unaffected areas. That said, there are many unanswered questions and unknown aspects with respect to the oviposition site selection process, as well as the propensity of some females to fly longer distances [19]. It was therefore deemed crucial to implement the oviposition and dispersal flights of the females based on valid, sensible assumptions so as to facilitate testing of different possibilities of the decision making process followed by female moths during dispersal in order to progressively make sense of the manner in which the pest spreads throughout its habitat. Effective implementation of these aspects may significantly improve the accuracy of simulating *E. saccharina*’s population dynamics, as well as shed light on some of the unanswered questions surrounding their behaviour.

4.6.1 General description of the implemented oviposition and distribution processes of *E. saccharina*

The logical progression followed and assumptions made to effectively implement the oviposition and dispersal cycles of *E. saccharina* in the simulation model are summarised in Figure 4.23.

Once a male and female moth have completed the mating process, the male returns to random localised movement for the remainder of that night. During subsequent nights, males may mate a second or even third time. Females only begin oviposition approximately 24 hours after mating. For this reason, they wait close to the mating site, moving only on a very small scale, until they are ready to oviposit. Oviposition usually takes place over the next one to two nights.

Some females have the ability to fly considerable distances in search of oviposition sites. In such cases, a female may wish to fly to areas which theoretically exist outside of the simulated

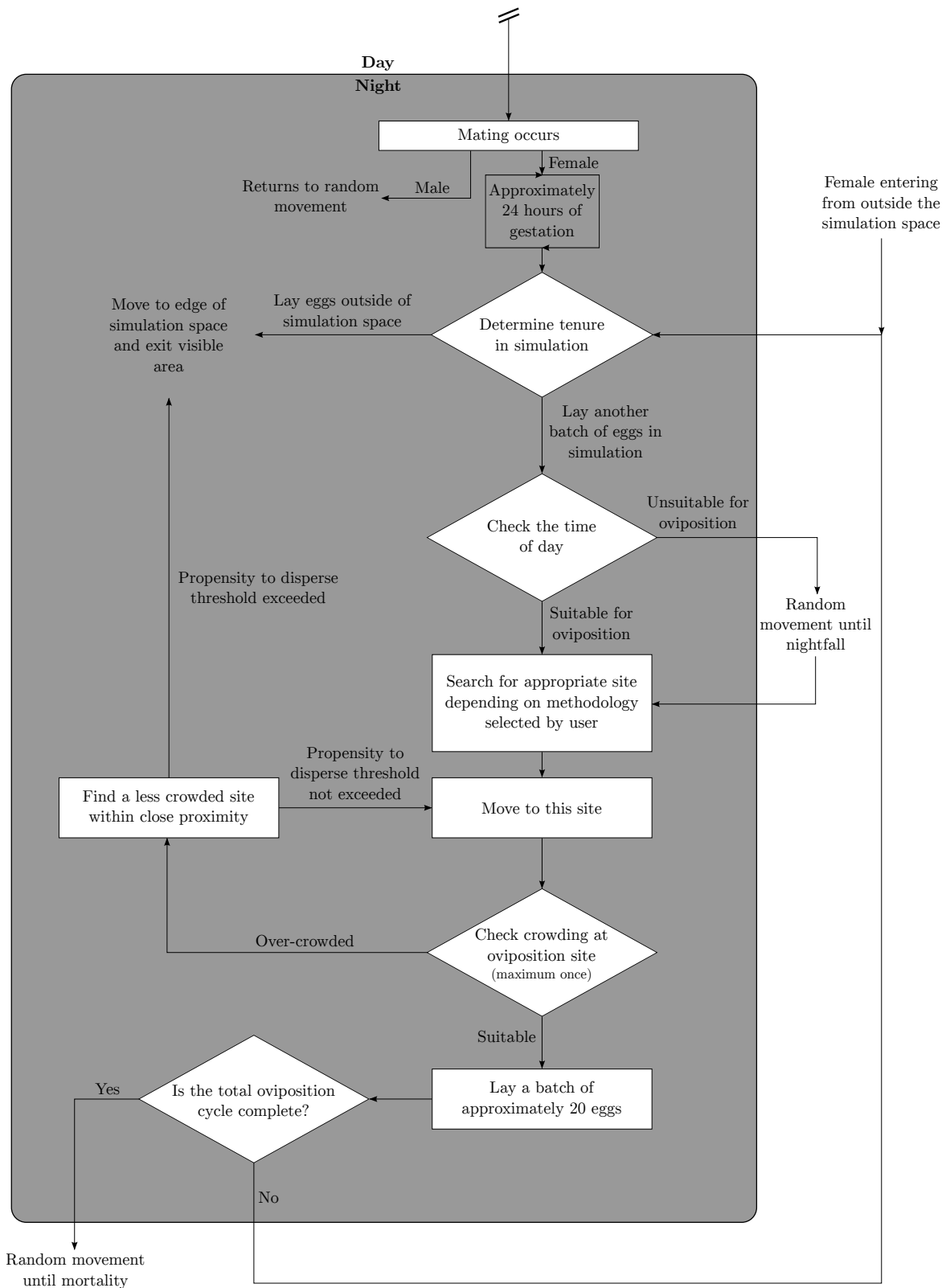


FIGURE 4.23: Flowchart of the general oviposition and dispersion processes, including assumptions, followed by *E. saccharina* post-mating in the simulation model.

environment. This departure from the simulation space can occur at any stage between batches of eggs laid by the female. During departure from the simulation space, the female moves toward the edge of the simulation environment and then exits the visible area and, in doing so, no longer exists in the simulation. If a female wishes to continue laying eggs in the visible simulation space, the process begins by assessing the time of day. Since females are understood to only oviposit during the early hours of the evening (the same time period during which mating is assumed to take place), the oviposition process in the simulation begins by assessing whether or not this time period is currently being experienced. If not, the female remains in her current location, moving on a very localised scale (resembling walking) until the appropriate time for oviposition to ensue.

Once this time period is entered, the next step in the oviposition process involves the selection of a suitable location to lay eggs. Conlong [22] has suggested that females may have an inherent ability to seek out more mature sugarcane in which to oviposit, but at the same time, have been observed to oviposit in almost any cryptic site in the absence of leaf sheaths. Furthermore, it is speculated that males may even play a role in the selection of oviposition sites for females [19]. This notion is consistent with the research of Berry [13] who found, using in-field experimentation, that sugarcane stalks with a higher number of internodes were found to be more heavily infested by *E. saccharina*. This suggests that shoots exhibiting the greatest growth potential attract the insect, possibly due to nitrogen content. In light of this, an *E. saccharina* population may inherently gravitate toward these regions and, as a result, male displays would be more heavily congested and, hence, stronger in these areas of more mature cane. This means that females may congregate in these areas to mate and, since most eggs are laid close to the emergence site, so the population would continue to grow in the region where the males are located. These assumptions and insights surrounding the selection of an oviposition location are intuitively included in the simulation model by two processes which incorporate the preference for more mature sugarcane in different manners.

In the first process, the route for oviposition is modelled by performing a search to find the direction in which the female should move from her current location to encounter the most mature sugarcane and, by extension, good oviposition sites. The search is confined to areas within a radius specified by the user. Once the most suitable direction has been found, the female moves off within a randomly generated ‘direction window’, centred around the primary ‘best’ direction discovered. Oviposition occurs at intervals, the distance between which are informed, in part, by the flight ambition of the female. Each time oviposition occurs, a new direction within the ‘direction window’ is chartered. The female consistently moves away from the mating location until the final batch of eggs is oviposited within close proximity of the intersection point between the original direction in which the female set off and the maximum distance over which she was willing to fly, indicated by her flight ambition.

The second process is, in essence, the converse of the first. In this instance, the route for oviposition is modelled by setting the female off in a random direction from her current location, with the maximum flight distance specified by her flight ambition. Whilst flying, a search window of a user-defined length is established in which the female can assess the quality of the sugarcane stalks over which she flies. This approach was inspired by the agent-based model of mosquito foraging behaviour of Gu *et al.* [30]. Upon discovery of a stalk within a search window which meets the user-defined quality criteria threshold, the female diverts from her route in order to make use of this oviposition site, before continuing towards her original destination post oviposition. This process continues until six batches of eggs have been oviposited and the cycle is over, or until the female reaches the outer boundary of her flight ambition. At this point, if

she has not yet oviposited all of the batches of eggs, she will oviposit the remaining batches in the vicinity of this location.

Both algorithms assume cognitive ability of *E. saccharina* in decision making processes. This assumption is based on expert opinion [19], as well as other research which has been conducted on similar insects [30, 67].

Upon arrival at an oviposition site in both search paradigms, the female assesses the surroundings in respect of how densely populated the oviposition sites in the area are. In line with Conlong's suggestions [19, 22], if the oviposition sites in the current location are occupied by existing eggs, the female attempts to find an area nearby with more available sites (that is, where less eggs exist). Once located, the female then moves to this area to oviposit a batch of eggs. This attempt at finding a less congested site only occurs once and, if the new site is also densely populated, oviposition by the female will still occur, since *E. saccharina* have been observed to resort to almost any site (even against the sugarcane stalk stem base) when no cryptic sites are available [61].

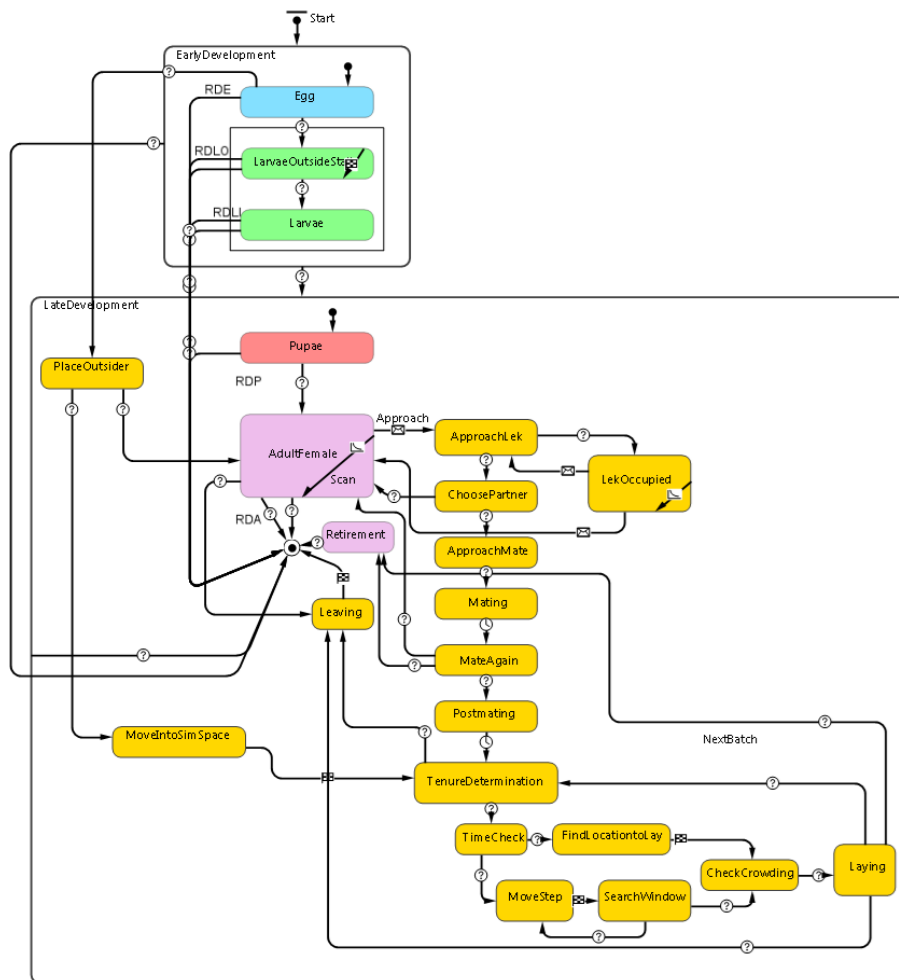
Once the female has oviposited a batch of eggs, the cycle begins again with an assessment of whether or not the female must leave the simulation space. If not, the female continues using the specified means of oviposition site selection in order to find an appropriate location for the next batch of eggs, as described above. Once six batches of eggs have been oviposited, the female returns to random movement until mortality.

The oviposition and dispersal processes also model the possibility of female moths entering the simulation environment from the hypothetical surrounding simulation space. Since males are not strong fliers, it is assumed in the simulation that no males enter the simulation space from the surrounding areas. When a female enters the visible environment, it is assumed that, for the most part, she does so in search of oviposition sites as a far flying female. There also exists a small possibility that the reason for dispersal is that the female remains unmated after the first few nights after emergence [5]. Female enterers who are in the process of oviposition immediately enter the oviposition cycle, whilst females coming in search of mates join the adult life stage to begin looking for *leks* at the appropriate time.

4.6.2 Description of the implemented oviposition and dispersal processes in an ANYLOGIC modelling environment

Other than random movement, male agents in the ANYLOGIC simulation do not disperse within the simulation space on a notable scale. This, in combination with the fact that they play no active role in the oviposition process, means that no additional elements are required in the male statechart for the purposes of oviposition and dispersal. Once mated, males remain relatively inactive for the remainder of that particular night and, for nights thereafter, may mate with different females.

The female controls the oviposition process and is the more likely than the male to disperse and spread within the simulation environment, as well as the implied surrounding areas. For this reason, further elements exist in the female statechart to facilitate these actions. Since decisions related to oviposition and dispersal (including possible adult females entering into the simulation space from the surroundings) take affect at a variety of different points within the female statechart, Figure 4.24(a) contains a screenshot of the entire female statechart, encapsulating the elements employed to effectively imitate the pest's oviposition and dispersal. Figure 4.24(b) contains the ANYLOGIC model components used in conjunction with the statechart in order to implement all oviposition and dispersion-related activity into the simulation model.



(a) Screenshot of the *female* statechart, including oviposition and distribution control attributes.

- | | | |
|------------------------|-------------------------|-----------------------|
| Ⓛ LeaveSimulationSpace | Ⓛ Leaver | Ⓛ OvipositSiteX |
| Ⓛ TimetoDepart | Ⓛ KeepGoing | Ⓛ OvipositSiteY |
| Ⓛ AssessCrowding | Ⓛ HeadingDirection | Ⓛ FinishedLaying |
| Ⓛ Oviposit | Ⓛ CantFindaSpot | Ⓛ NumberofBatches |
| Ⓛ SetUpOutsideEnterer | Ⓛ Finalx | Ⓛ FindBetterSpot |
| Ⓛ OutsideEntererPlacer | Ⓛ Finaly | Ⓛ OutsideEnterer |
| Ⓛ FindMatureCane | Ⓛ HeadingWindowStart | Ⓛ TotalDisplacement |
| Ⓛ MovingWindow | Ⓛ HeadingWindowEnd | Ⓛ Heading |
| Ⓛ InitialX | Ⓛ CaneSearchRadius | Ⓛ RunningTotal |
| Ⓛ InitialY | Ⓛ AnticipatedFinalX | Ⓛ BestDirectionAve |
| Ⓛ FirstBatchinSimSpace | Ⓛ AnticipatedFinalY | Ⓛ WindowSize |
| Ⓛ TypeofDispersion | Ⓛ WindowX | Ⓛ FinalBatchProximity |
| Ⓛ CaneQualityThreshold | Ⓛ WindowY | Ⓛ CurrentBest |
| Ⓛ BestCane | Ⓛ StartingX | Ⓛ PerceptionDistance |
| Ⓛ WindowCounter | Ⓛ StartingY | Ⓛ WindowIncrements |
| Ⓛ FoundLocation | Ⓛ RandomMovingDirection | |
| Ⓛ LocationX | | |
| Ⓛ LocationY | | |

(b) The ANYLOGIC tools used in accompaniment with the statechart to control the female oviposition and distribution behaviour in the simulation model.

FIGURE 4.24: Inner working of the statechart and parameters used in the female object class to effectively implement the oviposition and distribution behaviour of *E. saccharina* female moths.

After mating, the male returns to the `LookingforLeks` state (as described in §4.5). As may be seen in Figure 4.24(a), the female progresses to the `Postmating` stage, provided fertilisation has occurred, where she remains for 24 hours of simulated time. In this state, the `F MoveAroundInbetweens` function simulates very small-scale movement of the female. Once a day has passed, the female is ready to oviposit and a timeout transition is executed to move the female to the `TenureDetermination` state.

The `TenureDetermination` state is responsible for identifying females who are set to leave the simulation space and travel to the hypothetical surrounding areas and route their movement thereto. Such long-range flying is usually undertaken by larger, stronger females and, as such, females with a greater `F FlightAmbition` (which, in turn, comes as a result of both genetics, as well as good-quality nutrient consumption whilst boring in the sugarcane stalk). These stronger moths are typically the females who depart from the simulation space. For the most part, however, eggs are laid relatively close to the site where mating occurred [19].

The `TenureDetermination` state begins by setting the Boolean parameter `F FinishedLaying` to `false`. Since the `TenureDetermination` state is revisited after each batch of eggs is oviposited, the female must indicate that she has completed oviposition before the transition back to this state can be executed. This is indicated by the `F FinishedLaying` parameter being set to `true` in a different state and, after entering the `TenureDetermination` state, it is reset for the next mating instance. The `F Leaver` parameter, which is linked to a female's `F FlightAmbition` value, is then assessed and, if above a certain threshold value, the `F TimetoDepart` function is called. If below the threshold value, the agent is not a 'leaver' and simply progresses to the `TimeCheck` state to continue with the oviposition cycle.

In the `F TimetoDepart` function (which is only called if the female is deemed to be a 'leaver'), a `randomTrue` test is conducted and, if the result is `true`, the female departs from the simulation space. The Boolean parameter `F KeepGoing` is set to `false` if the female must depart from the simulation space and progress to the `Leaving` state, or `true` if she should remain in the simulated environment and progress to the `TimeCheck` state in order to proceed with oviposition. Although a female is flagged at an early stage in the simulation to be an agent who will eventually depart from the simulation space, functionality exists for a certain number of batches of eggs to be oviposited within the simulation space by that female before she departs in search of further oviposition locations. Since the `TenureDetermination` state is revisited between each batch of eggs oviposited, a leaving female may do so at any time during oviposition, or not at all.

In the `TimeCheck` state, a test is performed to assess whether the simulated time is suitable for oviposition. Although a lack of surety exists surrounding the matter, it is generally assumed that oviposition by females occurs in much the same time frame as mating [19]. For this reason, if the current simulated time is between 18h00–23h00, it is deemed suitable for oviposition and the female will progress through the conditional transition to either the `FindLocationtoLay` or `MoveStep` state, depending on the choice of oviposition location algorithm specified by the user. If not, the female remains in the `TimeCheck` state, subject only to very localised movement, as in the `Postmating` state, initiated by the `F MoveAroundInbetweens` function which executes cyclically. Once the time period is suitable, the female begins the process of identifying where to oviposit her eggs.

As mentioned, two algorithms were developed to model the oviposition site selection process. The choice of algorithm is specified before model execution. During a simulation run, the `F TypeofDispersion` parameter is assessed accordingly to determine which division of the female statechart should be entered to guide the oviposition site selection process.

The first algorithm is employed in the `FindLocationtoLay` state where the `F` `FindMatureCane` function is called. As discussed in §4.6.1, *E. saccharina* is assumed to have a preference for selecting mature sugarcane owing to a variety of factors. `F` `FindMatureCane` simulates the method by which this may occur. The logic sequence of the function begins by saving the initial x and y coordinates of the female using the `InitialX` and `InitialY` parameters. Thereafter, a search is undertaken in 12° increments of the 360° around the female. At each angular search instance, five equi-spaced locations distanced between the female and a user-specified radius from the female is specified. This radius can be any distance from the female's location up to and including a distance equivalent to that of her `FlightAmbition`. At each of these search locations, the surrounding eight points of the `Sugarcane` array, as well as the located point itself, are assessed. At discrete points where stalks are located (*i.e.* the maturity value of the point is greater than zero), the stalk 'value' is added to the `RunningTotal` parameter and a locally declared counter is incremented. In the event that the location being searched falls outside of the simulation environment, an average value of '1' is attributed to that particular location. This may occur in instances where a female is very near to the boundary of the simulated space on one or more sides of the circular search. Using the running total and counter, an average 'quality' value is then determined for the points identified in that particular angular direction. If this average exceeds the previous best average of another direction, the `BestDirectionAve` parameter is set to this average and the `Heading` parameter is updated with the new best angle. Thereafter, the `Finalx` and `Finaly` parameters are updated, saving the x and y coordinates of the corresponding point on the radius of the agent's `FlightAmbition` at this particular angle, respectively. A graphical interpretation of the case when a new best angular direction is found for a particular search is shown in Figure 4.25 which portrays an example of the female searching at an angular direction of ϕ .

Once all thirty directions have been considered and the locally optimal direction information has been saved, a 'heading window' is determined within which the female may move on route to her final oviposition location. This window is centered on the `Heading` parameter, set during the previous search and is randomly determined for each agent within a general bound specified by the user. The `HeadingWindowStart` (H_s) is specified as

$$H_s = H - \text{uniform}(\theta_{lower}, \theta_{upper}),$$

where H represents the best direction discovered during the search, `uniform` is a distribution where all values from the lower to upper bound have equal probability of selection, and θ_{lower} and θ_{upper} are minimum and maximum bounds specified by the user on the size of the half-window, respectively. The second half of the window, referred to as `HeadingWindowEnd`, encompasses the same angle, but on the other side of the `Heading` line.

The aforementioned search sequence and window setup only executes once per oviposition cycle when the `NumberofBatches` parameter of the female is zero (that is, no batches of eggs have yet been oviposited) and the results are used in all oviposition instances which follow.

At the start of the oviposition flight, the direction of flight is taken to be a sample from between the `HeadingWindowStart` and `HeadingWindowEnd` bounds. The distance to be travelled is calculated by dividing the `FlightAmbition` of the female by the total number of remaining oviposition instances (in the first case, six) and multiplying this result by a random sample from a uniform distribution on the interval $[0.8, 1.2]$ for diversity of results. Once this direction and distance have been established, the new location of the female is determined by converting

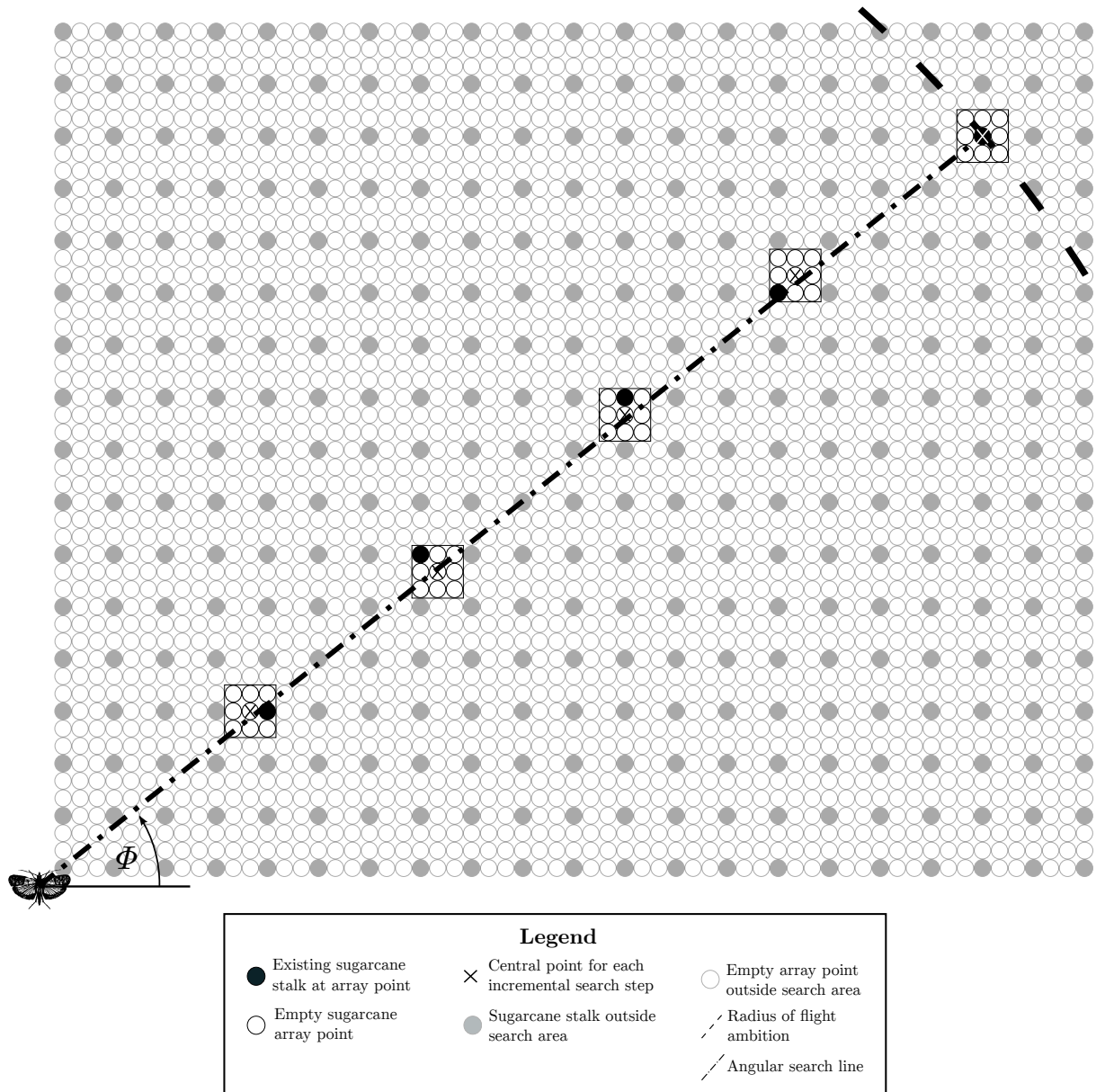


FIGURE 4.25: Graphical interpretation of the search process followed by female moths when determining the most suitable direction to head in search of good oviposition sites.

the polar coordinates to Cartesian coordinates, adding the result to the initial coordinates of the female (saved as InitialX and InitialY) and instructing the female to move to that location. Finally, the TotalDisplacement parameter is used to save the straight-line distance between the female's new position and initial position.

For every subsequent oviposition site, the new location is specified in relation to the initial coordinates of the female. This is done in order to keep movement confined to the initially determined 'heading window', as well as to ensure that the female remains within the original bound set by her FlightAmbition parameter. The direction chosen by the female is therefore, once again, sampled from between the $\text{HeadingWindowStart}$ and HeadingWindowEnd parameters, but

the distance travelled is calculated as

$$d = \frac{(F - d_{total}) \times \text{uniform}(0.8, 1.2)}{6 - N_{batches}} + \Delta d,$$

where F denotes the flight ambition of the female, and d_{total} and $N_{batches}$ denote the current total displacement of the female from her initial location and the number of batches already oviposited, respectively.

This distance is then used to specify the new location of the female with reference, as mentioned, to her initial coordinates. This can be done because the formula accounts for the portion of distance within the agent's **FlightAmbition** which has already been covered (by subtracting **TotalDisplacement** from **FlightAmbition** in the numerator of the expression), and then equally compensates for the fact that the remaining distance must be relatively equally distributed over the remaining batches of eggs to be oviposited (by decreasing the denominator in the sum by the **NumberofBatches** which have already been oviposited). This result is then added to the **TotalDisplacement** to account for distance already travelled between earlier batches, specifying her new oviposition location relative to her initial coordinates.

The final batch of eggs is set to be oviposited in a user-specified proximity to the originally determined 'best' location, captured by the **FinalX** and **Finaly** parameters. This proximity, relative to the **FinalX** and **Finaly** parameters, is specified by the **FinalBatchProximity** parameter. An example of a typical collection of oviposition flights undertaken by a female moth is shown in Figure 4.26. In the figure, the location specified by the **FinalX** and **Finaly** parameters is indicated as a black star shown on the intersection point between the line indicating the best discovered angle (the **Heading** parameter) and the radius created by the female's **FlightAmbition**. Note that in this case, the search radius within which the female assesses the sugarcane quality to determine the best direction is equal to her flight ambition.

The second available algorithm describing the female's process of locating oviposition sites is employed in the **MoveStep** state where the **MovingWindow** function is called. As opposed to the **FindMatureCane** function where the female oviposits eggs at random points within the best direction that could be sensed, this function takes on a completely random direction and attempts specific, sufficiently high-quality oviposition sites in the vicinity of this direction.

The **MovingWindow** function assesses which part of the algorithm should be executed (as determined by the state in which the agent currently resides). The division of the algorithm associated with the **MoveStep** state begins by resetting the **FoundLocation** parameter to zero. This parameter controls progression to states after site location and must therefore be reset between each site search instance. The remainder of the algorithm simply instructs the agent to move an increment forward along the random search line. Increment sizes are user-specified and stored by the **WindowIncrements** parameter. The total distance travelled in increments by the agents is stored as the **WindowCounter** parameter. Once the value of this parameter exceeds the **FlightAmbition** of the female, the function no longer instructs her to continue to move along the random direction line. The female is now instructed to move to a random location in the near vicinity since the desired oviposition locations were not found during the search process and the female must now resort to ovipositing in random locations [19]. Once the female has arrived at the instructed location, be it random or at the next increment in the random direction, the arrival-triggered transition moves her to the **SearchWindow** state.

Here the same **MovingWindow** function is called but, based on the different state in which the female resides, a different part of the algorithm is executed. This part of code begins by

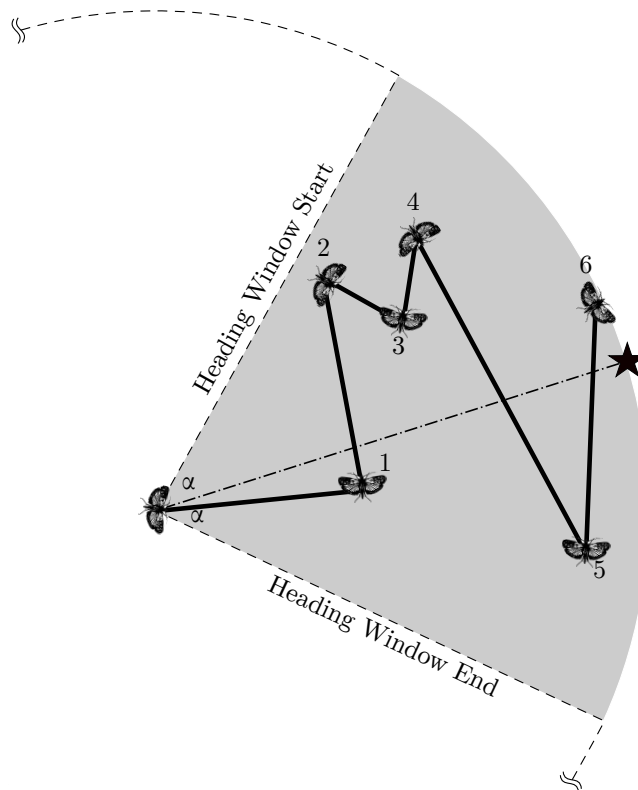


FIGURE 4.26: Graphical description of a possible collection of flights undertaken by a female moth between oviposition sites following the FindMatureCane approach.

determining whether or not the female has exceeded the bounds of her FlightAmbition and must thereby resort to random oviposition, or whether she is still moving incrementally along the line. If the distance is still less than the flight ambition, a search is performed within a window of a certain radius around the moth. This radius is specified according to the user-defined $\text{PerceptionDistance}$ parameter. The search process assesses the underlying Sugarcane array which fall within its bounds to determine the maturity value of the physical stalks which are encountered. Each discovered stalk is compared to the user-specified $\text{CaneQualityThreshold}$ which places a minimum bound on acceptable maturity values of the stalks. If this bound is exceeded, and the stalk discovered has a better maturity value than any other stalks identified within that particular window, the location of this stalk is captured in the LocationX and LocationY parameters and the BestCane parameter is updated with this maturity value.

If, by the end of the search through the window, no sites which exceed the $\text{CaneQualityThreshold}$ in quality have been found, the female does not enter the next phase of the oviposition process and simply returns to the MoveStep state. If, however, a suitable location has been found, the female moves to this location and then progresses to the CheckCrowding state for further instruction in the cycle.

If the distance incrementally travelled by the female exceeds her FlightAmbition when entering the SearchWindow state, the search process is skipped over and, since the female will then concede to oviposition in random locations, transition to the CheckCrowding state occurs.

A graphical representation of the **F** MovingWindow function may be seen in Figure 4.27. In this illustrative figure, the female does not deviate from the original flight path, but assesses sugarcane in incremental search windows at different points along the search line. Note that, in cases where the increment between windows is small and the search window size is large, the female may assess the same sugarcane stalk in two different, but consecutive windows. If this particular stalk is of a high quality value, it may result in the female re-visiting the stalk a number of times. Such areas which are regularly visited and re-visited by females are referred to as ‘hot spots’ and are typical in in-field observations of *E. saccharina* [19]. They result in population clusters within a field which is typical of the spread of the pest in nature.

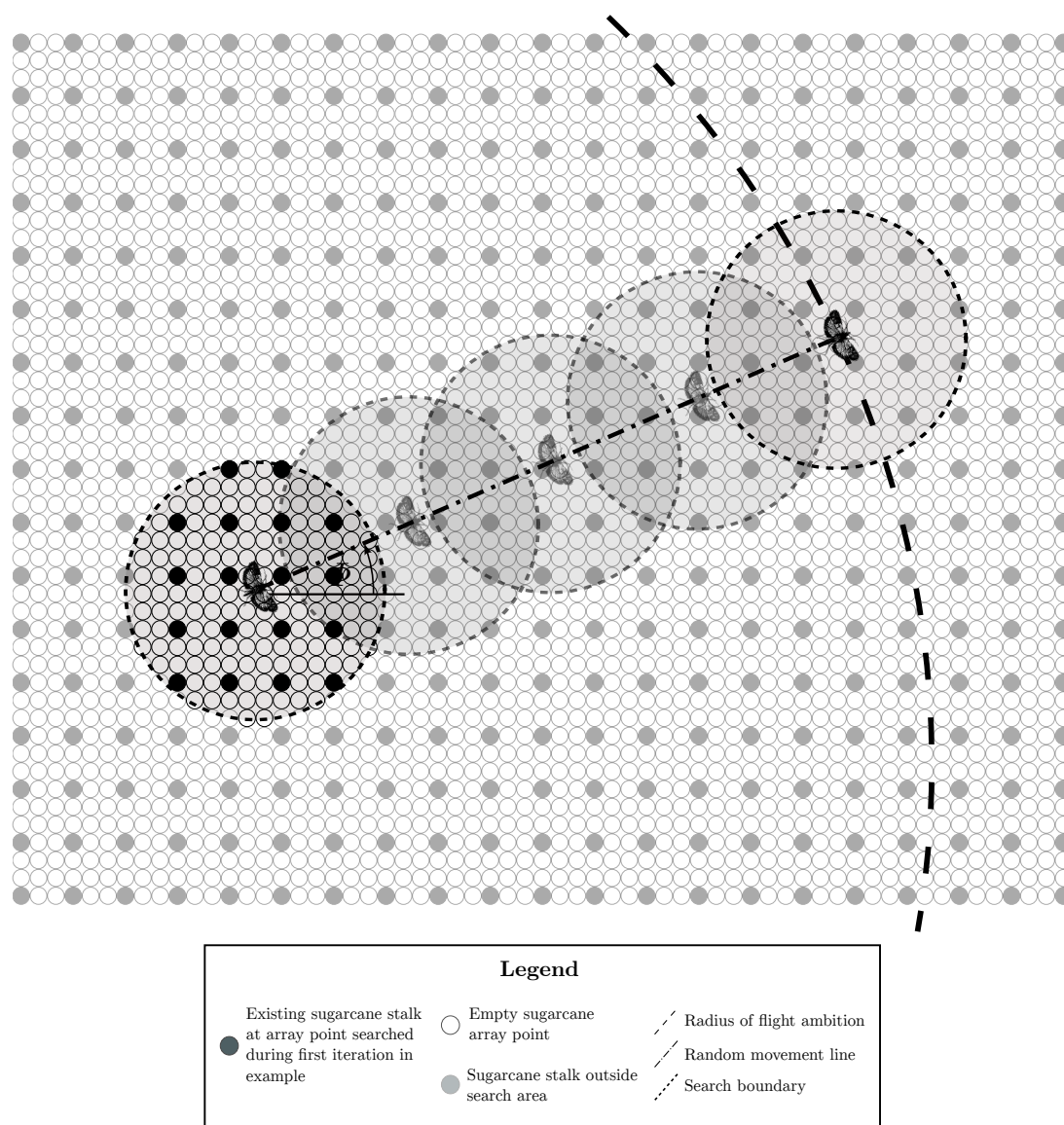


FIGURE 4.27: Graphical interpretation of the process followed by females whereby a random direction is undertaken and searches are performed at small increments along the movement line so as to determine suitable oviposition sites falling within a specified search radius.

An intuitive example of a possible flight sequence executed by the female following the moving window algorithm is shown in Figure 4.28. This is a simplified illustration of the function execution, but provides an idea of how movement in a random direction with small deviations from this path when appropriate oviposition locations are discovered within the search window

may occur. Sites where eggs were oviposited in this example are shown as numbers and the direction lines followed by the female in each case are shown for reference.

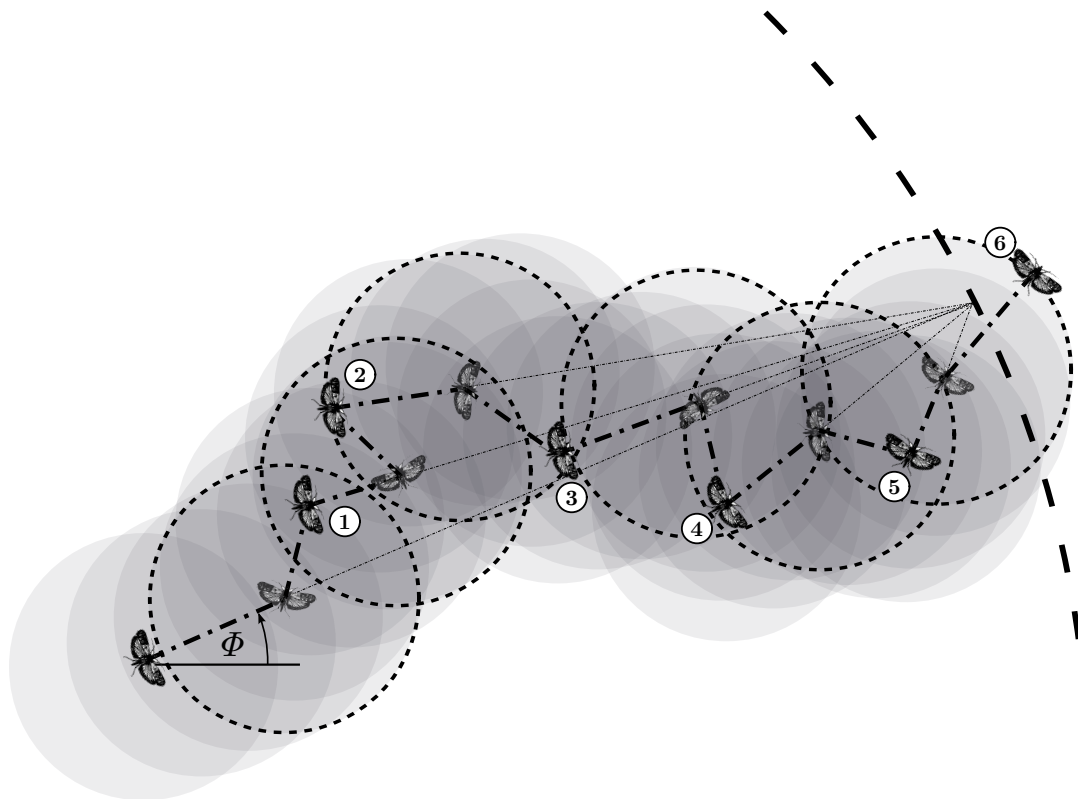


FIGURE 4.28: Graphical description of a possible collection of flights undertaken by a female moth between oviposition sites following the MovingWindow approach. The shaded circles without a distinct border indicate unsuccessful search instances by the female, whilst the shaded circles with a dotted border represent search windows in which a suitable sugarcane stalk was found. The female then proceeds to move to this stalk within the window, the location of which is indicated as an encircled batch number.

Once at the desired location between batches of eggs in each oviposition process, the female agent progresses to the **F** CheckCrowding state. This state begins by assessing the density of eggs which have already been oviposited in cryptic locations at the site. Despite a possible intuition of where more mature sugarcane may be located for oviposition, the female is not likely to be able to assess how many other females have previously chosen the same location for oviposition. In light of this, the female attempt to and find an alternate location very nearby the originally identified one where she can oviposit in a less crowded space. To achieve this, the **F** AssessCrowding function is called. The logic sequence of this function begins by filtering through all of the agents in the simulation and determining which are currently in the Egg state. Such agents are then assessed in terms of their proximity to the female who is searching for oviposition sites. Intuitively, this simulates the occurrence of suitable, cryptic sites that the female may identify in her current location which are already occupied. If the area is deemed sufficiently uninhabited for oviposition to occur, the female progresses to the Laying state. If the area is found to be crowded, the function tests several nearby locations for less crowded conditions. In line with the propensity of females to resort to almost any oviposition site when required, the female will eventually choose one of the sites nearby the identified location, even if it remains to be relatively crowded. If she is forced to settle for a crowded site, however, the **C** CantFindaSpot parameter is updated with a sample on the interval $[0,1]$ before progression to the Laying state. This parameter is then evaluated upon entry to the Laying state and, in rare circumstances, may

trigger the transition which instructs the female to depart from the simulation space in search of more favourable oviposition conditions. If not, the female conceded to ovipositing either at the originally identified location (if it was found to be sufficiently uncrowded) or at an alternative location found whilst attempting to avoid crowding, and the oviposition process is executed. This process begins by setting the OvipositSiteX and OvipositSiteY parameters to the current x and y Cartesian co-ordinates of the female's location, and then executing the `Oviposit` function.

The `Oviposition` function logic consists of two simple for-loops which generate new agents. Since *E. saccharina* generally oviposit in batches of approximately 20 eggs, the first for-loop dynamically adds a random number between 8–12 females to the simulation and places them at the oviposition site. Thereafter, the second loop also generates a random number between 8–12 male agents and places them, along with the females, at the oviposition site. All of these agents enter directly into the `Egg` state in their respective statecharts and, in the case of a female agent, it is ensured that the `OutsideEnterer` parameter does not exceed the threshold value on the transition to the `PlaceOutsider` state, meaning that it will never function as a female agent entering from the surrounding areas into the simulation space. After both males and females have been oviposited, the `NumberOfBatches` parameter is incremented once to keep track of how many batches have been oviposited during this cycle. Finally, the function sets the `FinishedLaying` parameter to `true`, indicating that transition from the `Laying` state to other states may occur.

From the `Laying` state, there are three possible states that the female may enter. First, there exists a transition from the `Laying` state to the `Leaving` state, as described earlier. Alternatively, if the `FinishedLaying` parameter is set to `true`, but the agent has not yet reached mortality and has oviposited fewer than six batches of eggs, the `NextBatch` transition moves the female back to the `TenureDetermination` state to begin the oviposition process again. This, once again, starts by testing whether the female should depart from the simulation space. The final possibility exists when the female has completed the oviposition of all six batches of eggs. In this instance, a transition takes her to the `Retirement` state where she is subject to random movement controlled by the `MoveAroundDay` and `MoveAroundNight` functions (as in the `AdultFemale` state) and, eventually, moves to the final mortality state once the `TestTransition` function determines that sufficient maturity has been reached.

The oviposition and dispersal cycle is also responsible for the functionality of female agents entering the simulation space from the implied surrounding areas. Such agents are generated dynamically according to a user-controlled parameter called `NewArrivalRate`, as discussed in 4.3.2. Each time a female agent is generated by this rate-controlled event, its `OutsideEnterer` parameter is set to 1. This is picked up by a conditional transition when the newly created agent enters the `Egg` state and, as a result, it is immediately re-routed from the `Egg` state and subsequent traditional life cycle to the `PlaceOutsider` state. Here the `SetUpOutsideEnterer` function is called. This function begins by establishing a number of the parameters of the female related to mating. The `NumberOfBatches`, `MatingCount`, `GrandTotalNumberOviposits`, `Leaver` and `LayBeforeLeaving` parameters are all set to random values in order to simulate the notion that the female agent has existed in the space outside of the simulation space and is entering in the middle of the oviposition process, or as a result of being a far flier. After these parameters are set, the agent is randomly placed at a point on the periphery of the simulation space. This is the location from which the female will enter the simulation.

From the `PlaceOutsider` state, two possible transitions exist. First, a `randomTrue` transition occurs which moves a certain proportion of entering females directly to the `AdultFemale` state. This represents females who have undertaken dispersal as a result of remaining unmated for the first few nights after emergence. In this case, the `NumberOfBatches` is set to 0. Thereafter, entry into the simulation space is initiated by the `OutsideEntererPlacer` function. The agent is then set on a flight path over a distance which results in the entering agent arriving somewhere in the vicinity of simulation boundary through which it entered. For example, if the agent is placed for entry near the top left corner of the simulation environment, it will most likely settle somewhere in the top left quadrant of the simulation. The distance is sampled from a truncated normal distribution with a mean which translates to approximately 25 metres and a standard deviation of roughly 10 metres, with a minimum of 0 metres. This means that crossing of the entire simulation space by an entering agent will require a flight distance of a length which is very uncommon and only likely to happen in extreme cases of sampling from the normal distribution. Once the female arrives at the location, she continues in the `AdultFemale` state in the same manner as any female who has matured through all life stages to arrive there.

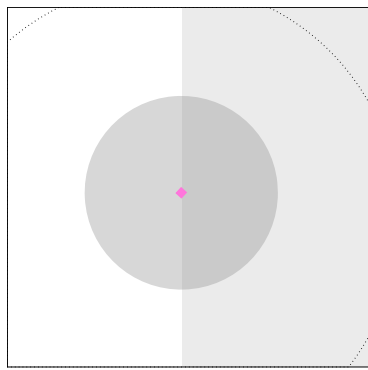
If, however, the more likely case of a female entering in search of oviposition sites occurs, the transition between the `PlaceOutsider` and `MoveIntoSimSpace` states is assessed according to a condition which determines the simulated time of day being experienced by the agents. This is in light of the assumption that females would only travel such significant distances during the night. If the simulated time indicates daytime, the agent is withheld from the simulation space until nightfall. When this occurs, the agent progresses to the `MoveIntoSimSpace` state where the `OutsideEntererPlacer` is called to move the agent to an appropriate location in the same manner as described above. Upon arrival at this location, the female enters the `TenureDetermination` state and, in doing so, joins the oviposition process in the same manner as other agents, moving through the appropriate states to oviposit eggs to complete its number of batches (some of which are assumed to have been oviposited outside of the simulation space).

4.6.3 Verification of the implemented oviposition and distribution processes functionality

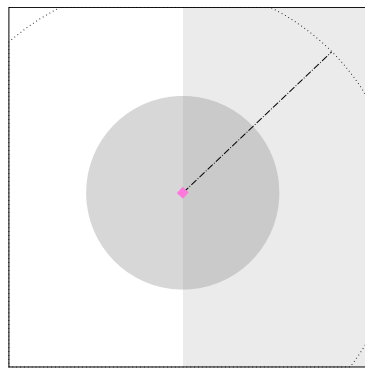
The implementation of the oviposition and dispersal cycle may largely be assessed by viewing the model during operation. Two such representative instances are shown in Figures 4.29(a)–4.29(i) and 4.30(a)–4.30(i) in order to determine the success with which different possibilities in respect of how the female selects oviposition sites, as described in §4.6.2, were incorporated into the simulation model. The figures show a sequential execution of each of the two algorithms, and include representative lines and shaded regions to indicate the effect of the user-defined parameters in each case. Such inclusions are purely illustrative and are excluded from the final model. In the figures, female agents are represented as pink squares and newly oviposited eggs as teal squares. The white and grey areas within the figures represent sugarcane of a lower and higher maturity, respectively, whilst the shaded grey circle surrounding the female indicates the user-defined radius within which the agent is able to sense the quality of the underlying sugarcane.

Figures 4.29(a)–4.29(i) represent the typical execution of the `FindMatureCane` function within the simulation. The processes illustrated in Figures 4.29(a)–4.29(c) are common in the execution of all modifications of the algorithm. The remaining figures describe three different outcomes which may materialise based on the parameters selected by the user.

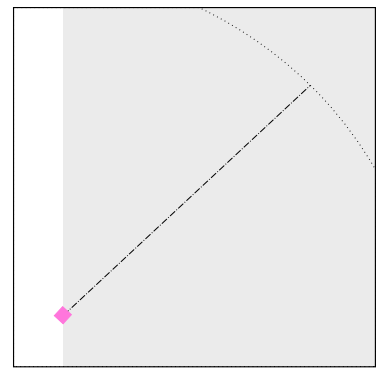
Referring to the various parts of Figure 4.29, the following interpretations are offered:



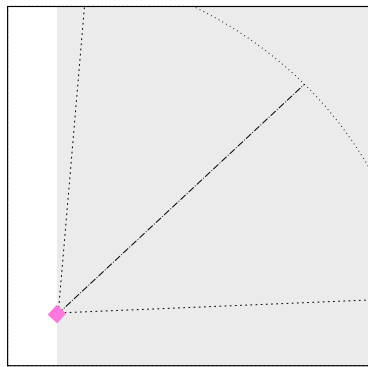
(a) A search is performed of the surrounding area.



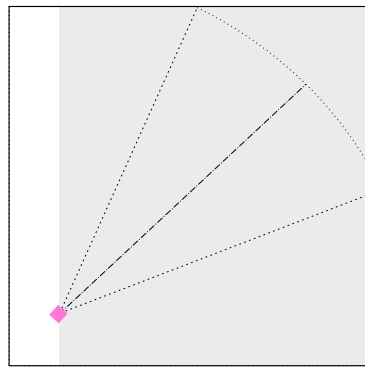
(b) The most suitable direction is determined.



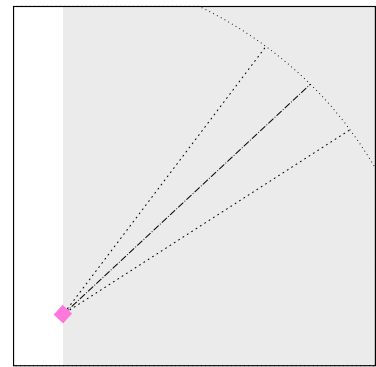
(c) The flight path is set from the current location to the flight ambition bound.



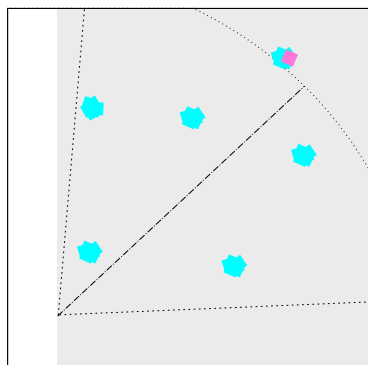
(d) Possibility 1: A large oviposition window is established.



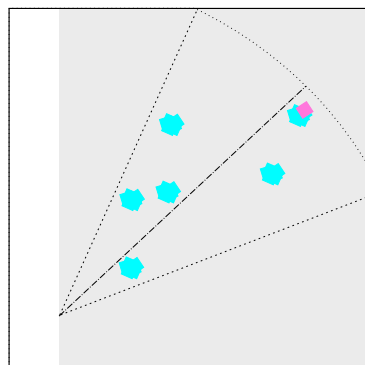
(e) Possibility 2: A medium oviposition window is established.



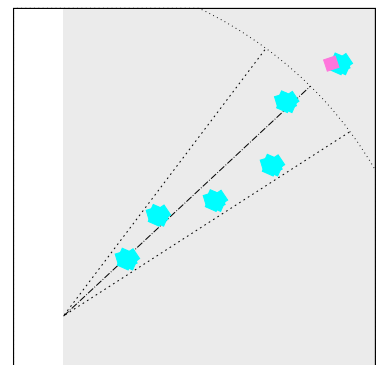
(f) Possibility 3: A narrow oviposition window is established.



(g) Outcome 1: A diverse spread of eggs results within the window.



(h) Outcome 2: A balanced spread of eggs results within the window.



(i) Outcome 3: A concentrated spread of eggs results within the window.

FIGURE 4.29: The *FindMatureCane* oviposition site search function executed in an ANYLOGIC simulation run.

- (a) The female performs a search within the bounds of a radius specified by the user. The maturity values of sugarcane stalks along the search lines are averaged to determine the ‘best’ direction to depart so as to find good-quality oviposition sites.
- (b) The ‘best’ direction has been discovered by the female. Note that, as expected, this

- direction falls within the space indicating more mature sugarcane.
- (c) The female now sets a flight path from the current location to the intersection point between the ‘best’ direction line and the maximum distance over which she is willing to fly, based on genetic predisposition, as well as the quality of sugarcane on which she fed as a larva. Note that, if desired, this distance of the flight ambition radius is not necessarily the same as the radius of the area in which she searched to determine the ‘best’ direction. The search space simply informs the direction and the female assumes that the sugarcane beyond the searched area will be of equally good quality. This may, of course, not always be the case, but she will still take off in this direction.
 - (d) The size range of the symmetrical oviposition window established about the ‘best’ direction line is specified by the user. In this case, the window was chosen to be large and, as a result, there is a large area encompassed by the window in which the female may now oviposit eggs. Note that some eggs may even end up being oviposited very close to less mature cane, in spite of the original departure direction.
 - (e) In this alternative case, the size range was specified to be medium. This means that the encompassed area surrounds the ‘best’ direction more closely and the resulting eggs are oviposited more in the region of the departure point of the female.
 - (f) The third case represents a small range selected for the oviposition window. Here, eggs are oviposited much closer to the ‘best’ direction line and there is a far smaller area in which the female may deviate from the original route. In all cases, the final batch of eggs is oviposited in proximity to the intersection point between the ‘best’ direction line and the flight ambition radius. This degree of proximity may be specified by the user.
 - (g) Figure 4.29(g) shows the typical outcome of specifying a large oviposition window. Although all within the mature sugarcane space, the eggs are widely distributed and are not necessarily located at the high-quality stalks originally discovered by the female.
 - (h) The outcome of the medium window of Figure 4.30(e) shows eggs that are far more condensed around the ‘best’ direction line. This means the resulting larvae, although not located in the perceived best locations, will be closer to sugarcane of a higher quality and, as a result, probably in sugarcane with a sufficiently high nutrient content.
 - (i) The final outcome shows eggs which are oviposited very close to the ‘best’ direction line. This means the resulting larvae will most likely infest the best (or very close to) quality stalks the female could locate during her search.

Figures 4.30(a)–4.30(i) represent the typical execution of one of the possible cases of the **F** **MovingWindow** function within the simulation. Referring to the various parts of Figure 4.30, the following interpretations are offered:

- (a) The gestation period has expired and the female is ready to begin oviposition. Note that the initial position of the female is identical to that of Figure 4.29(a), in that the underlying field is split into half mature (light grey) and half immature (white) sugarcane.
- (b) A random direction is assumed by the female. In this case, the direction leads the female away from the mature sugarcane. This is the main difference between the two algorithms. The direction line intersects with the radius formed by the flight ambition of the female to form the anticipated final destination of the flight.

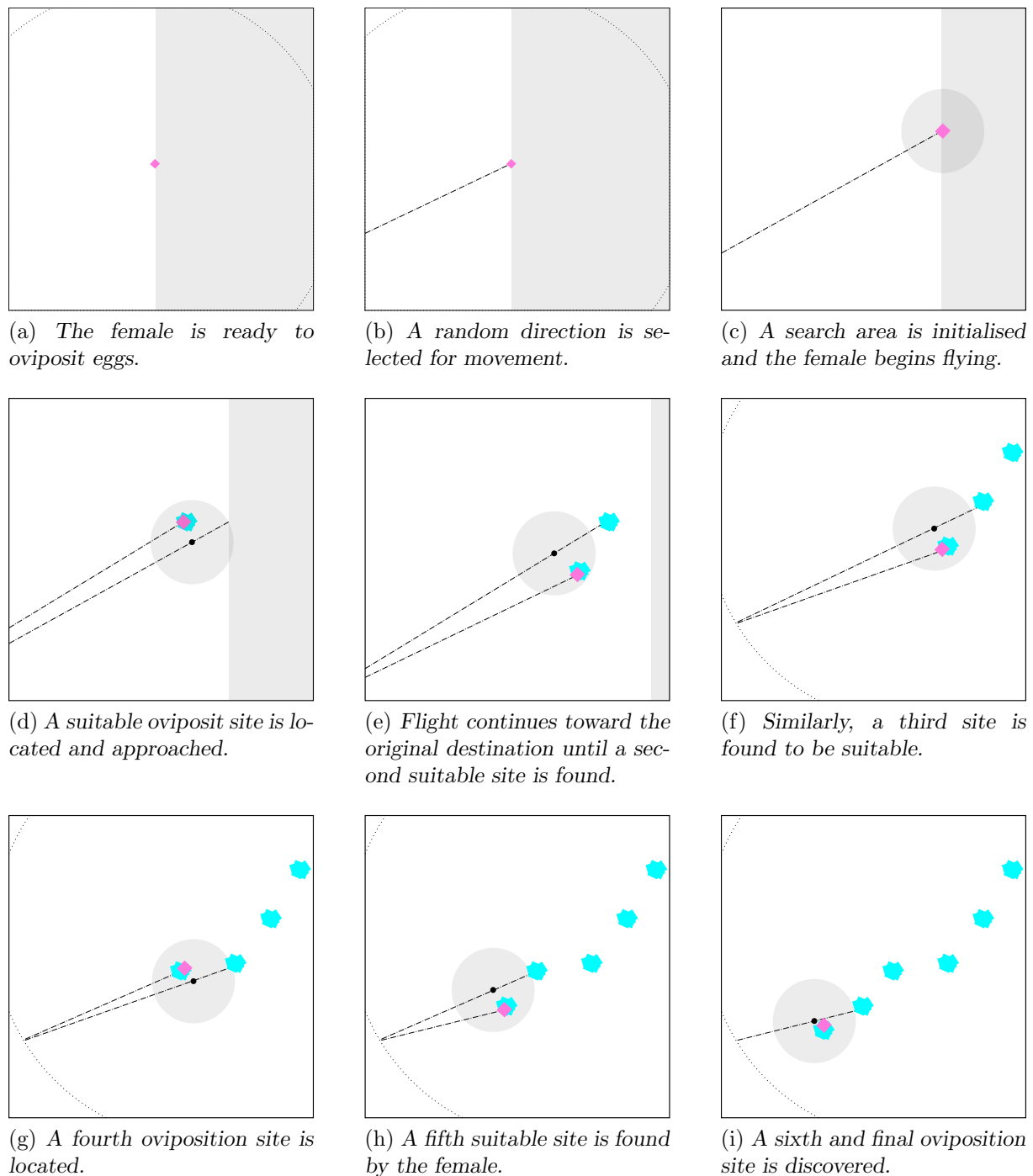


FIGURE 4.30: The MovingWindow oviposition site search function executed in an ANYLOGIC simulation run.

- (c) A search window area of a user-defined size is initialised around the female. It is assumed that the female can sense good-quality cane within this window. The agent moves along the direction line incrementally, searching the sugarcane stalks captured by the window until one such stalk meets the user-defined criteria of what is deemed 'quality'.
- (d) A sugarcane stalk which satisfies the quality threshold of the female has been located within the search window. She thus deviates from her path to oviposit a batch of eggs at

the discovered site. Once the eggs have been oviposited, a new direction line is assumed. This direction aims toward the same original destination of the female.

- (e) The female moves along the new line incrementally until she finds a second suitable oviposition site. Once found, she deviates once again to oviposit a second batch of eggs within the moving window area.
- (f) This process continues between batches. In Figure 4.30(f), the female has found an appropriate oviposition site for the third batch of eggs.
- (g) The space between incremental search instances is specified by the user and can be enforced to be a minimum distance. Spacings between consecutive batches of eggs, however, depend on the frequency with which suitable sites are located. In Figure 4.30(g), a fourth batch of eggs has been oviposited at a suitable site.
- (h) The female continues on her flight path, still directed towards the original destination set by the random direction line and the radius of her flight ambition. A fifth suitable site is found.
- (i) The sixth and final batch of eggs is oviposited at an appropriate site to complete the oviposition cycle. In the event that fewer than six appropriate sites are found between the initial location of the female and the radius of her flight ambition, the remaining batches (in some cases where the quality threshold is very high, this may be all six batches) are oviposited randomly within the vicinity of the destination location.

These two illustrative examples show successful implementation of the two possible oviposition site search algorithms implemented in the simulation model. The resulting spread of eggs represents dispersal typically seen in the field to a satisfactory degree from an observation perspective, based on the limited literature which exists on oviposition.

4.7 Chapter summary

In this chapter, the approaches taken and assumptions made in the design of an agent-based model of *E. saccharina* were described on a conceptual level, accompanied by a detailed description of the implementation logic within an ANYLOGIC simulation environment. The model consists of four major building blocks, namely the GUI, the life cycle and associated influence of temperature, the mating process and the oviposition and dispersal of *E. saccharina*. In each of these sub-models, the biological processes encapsulated were derived from literature and, in the absence of literature, expert opinion and sensible assumptions. Each sub-model was then verified against existing knowledge of the process to determine the accuracy with which reconstruction of the process was achieved. In all cases, the implemented processes were found to adequately mimic the in-field behaviour of *E. saccharina*.

CHAPTER 5

Simulation model verification and validation

Contents

5.1 Model verification, output analysis and validation	93
5.2 Verification of the agent-based model of <i>E. saccharina</i>	94
5.3 Validation of the agent-based model of <i>E. saccharina</i>	96
5.3.1 <i>Face validation</i>	96
5.3.2 <i>Comparison with video footage of E. saccharina</i>	99
5.3.3 <i>Parameter variation and sensitivity analysis</i>	100
5.4 Chapter summary	107

In this chapter, further verification methods applied to the simulation model in its entirety (to be considered in conjunction with those performed on each model building block in §4) are discussed. Thereafter, the validation process is described. This consists of a face validation by an industry expert, a short analysis of video footage of *E. saccharina* mating in captivity and a pilot sensitivity analysis. Since the sensitivity analysis results of §5.3.3 were deemed inconclusive in terms of being able to draw definitive conclusions about the parameter effects in the simulation model, a parameter variation analysis was proposed.

5.1 Model verification, output analysis and validation

One of the most important parts of the simulation process involves the attempt by a simulator to determine whether or not a simulation model is an accurate representation of the actual system being studied. Law [54] defines *verification* as the process of debugging a computer program in order to determine whether it performs as intended. *Validation*, on the other hand, consists of determining whether a simulation model (not a computer program) is an accurate representation of the real-world system being studied. This is, in turn, contrasted with *output analysis* which is concerned with determining the simulation model's (not necessarily the system's) true parameters or characteristics. Banks [12] suggests that this form of analysis of simulation models aims to increase the credibility of the model to an acceptable level so as to enable use of the model in decision-making processes which pertain to the real-world system.

In §4, each section describing the four fundamental building blocks of the simulation model concluded with a simple building block verification indicating the successful functional implementation of the assumed process being modelled. In a full-scale model, these four building blocks are required to interact and operate progressively during the execution of a simulation

run. According to Banks [12], verification is utilised in the comparison of the conceptual model to the computer representation that implements that conception. The process of verification asks the following questions: *Is the model implemented correctly in the computer? Are the input parameters and logical structure of the model correctly represented?* Law [54] emphasises this step as the successful debugging of a simulation model so that it is free from any errors of logic.

Law [54] also suggests output analysis to be historically lacking in the field of simulation modelling. This is primarily attributed to the fact that simulation data are rarely, if ever, independent and, as such, classical statistical analyses based on observations are not directly applicable. In the case of the simulation model presented in this dissertation, the output of the model is largely dependent on the input values provided for parameters used during model execution. In such a biological simulation model, these values would ideally be based on the results of in-field and experimental measurements as well as tests conducted in respect of a particular aspect of *E. saccharina*'s biology. Unfortunately, such detailed data are not often available and, furthermore, are not currently acquirable through research. In light of this, a simulation of this nature bears more significance in terms of employing changeable inputs which can be determined by the simulation operator. In this case, the output of the simulation can be examined by professionals in the field or compared to data which have been collected physically or from video footage drawn from experiments performed on *E. saccharina*. Discrepancies noted between the simulation and real-world behaviour of the pest can then be addressed in an attempt to improve the model inputs or, perhaps more likely, inform in-field researchers of areas of the pest's biology where significant lack of insight exists. In doing so, a better understanding of *E. saccharina* and, by extension, better insight into accurate parameter selection may be achieved. The above-mentioned form of output analysis and parameter selection lends itself largely to the process of validation described by Banks [12].

Whilst verification is concerned with building a model correctly, validation is concerned with building the correct model. In light of this, the simulation model's validation is progressively achieved by careful selection of parameters, as well as incremental testing and sensitivity analysis so as to determine the most suitable conditions under which to execute the simulation in order to accurately represent the underlying real-world system. This process of calibration is repeated until the desired level of model accuracy has been reached, and the model is continually updated in line with insights and input gained from in-field studies which continue to be conducted.

The iterative process of verification, validation and calibration is summarised in Figure 5.1.

5.2 Verification of the agent-based model of *E. saccharina*

Law [54] proposes several methods of simulation verification, many of which are applicable in this study. During development, the entire simulation was continually compiled and executed in an iterative manner so as to ensure constant debugging of code errors and execution of the simulation model with each new addition to the model. This method of verification is further evident in the separate verification of each of the model building blocks, conducted in §4. In each of these local verifications, the method proposed by Law, whereby the model is run under simplifying assumptions in order to identify its true characteristics, was employed. This allows for focused analysis of each of the building blocks which compose the final model. These building blocks were then progressively integrated into the complete model, ensuring their integration and temporal interaction in the manner specified in [88] and, at each point in the integration of the total model, executing the simulation to ensure an absence of errors, as well as fluid interaction between the different model building blocks.

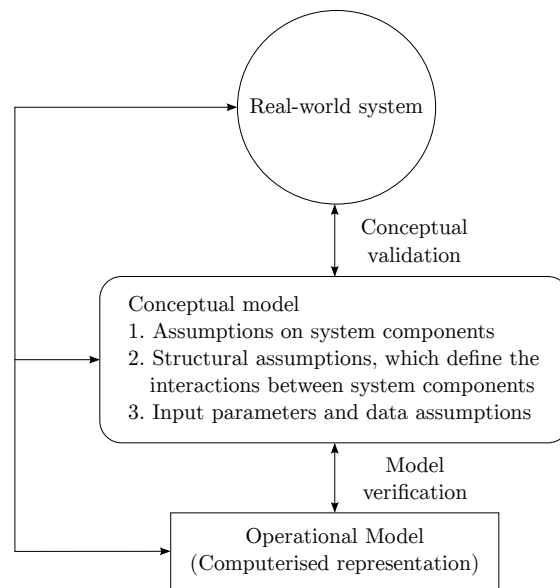


FIGURE 5.1: The verification and validation process of a simulation model [12].

Both Banks [12] and Law [54] recommend the implementation and development of a model in a dedicated simulation software package, equipped with an *interactive run controller* (IRC) or debugger, as well as a graphics terminal or animation feature, in order to visually detect any irregularities in the behaviour of the system. A pre-existing IRC and debugger is implemented within ANYLOGIC UNIVERSITY 7.1.2. Any model which is to run in ANYLOGIC requires its underlying code to first be compiled. During compilation, the debugger scans through the written code for discrepancies and syntax errors. If located, the compilation is interrupted and the error is reported in the output terminal of the program. This report details the type of error, as well as its location within the simulation project code, and suggestions for its possible rectification are proposed. Once the model is free from any errors detected by the IRC during compilation, it may be run. At times, the IRC may detect an error whilst the model is running. Such errors are often of the type *null pointer*, whereby an agent in the simulation is instructed to complete an action which is not completely specified within the underlying code. When this occurs, the model is suspended and an error report is generated which describes the nature of the error, as well as its location within the main JAVASCRIPT code file being executed as the simulation runs. Such error detection and reporting was essential in the development of the simulation model implementation reported in this dissertation as, during the testing of each new component added to the simulation, syntax and logic errors could be visited individually and rectified continually. This ensured that the simulation did not progress too far so as to make revisiting and editing implemented code sequences unnecessarily complicated.

Debugging was conducted at the base speed of the simulation (*i.e.* one second of real time is equivalent to one hour of simulation time) as well as virtual time (*i.e.* the maximum speed at which the simulation could be run). Running the model at base speed allowed for visual inspection of the correctness of newly added code sequences, verified by the subsequent expected behaviour of agents. Where necessary, the model was run at a speed slower than the base speed of the simulation to ensure that small-scale model intricacies were operating correctly as per intended implementation. Virtual time execution of the model was used to ensure that newly added code sequences did not result in errors or null pointers over an extended period of time in the simulation.

Law [54] also proposes the use of a *trace* during model verification in order to evaluate each possible program path, as well as the program's ability to deal with extreme conditions. The *trace* feature of ANYLOGIC was comprehensively used during verification of the agent-based model developed in this dissertation to evaluate parameter values during simulation runs, as well as to indicate progression of agents between states during sub-cycle execution. Several traces are offered by ANYLOGIC, the most commonly used of which includes the display of model parameters within the simulated space, as well as the printing of structurally included lines to the output terminal of the program conveying information, action and parameter changes of agents in the executed model. Printed lines were also used as *flags* to indicate that a certain sequence of code was being executed sequentially as intended. The visual animation capabilities of ANYLOGIC were also used in order to verify agent behaviour. Agent appearance and movement were altered depending on the state in which it resided or a sequence of code being executed in order to visually cue the successful implementation of anticipated agent behaviour. The combination of these two verification techniques facilitated comprehensive verification, allowing one to determine, from the animation capabilities, a more abstracted confirmation that the components of the model were behaving in a logical manner (*e.g.* agents progressing through life stages, changing colour upon entry of each new state), whilst the trace allowed for verification of precision and accurate implementation of model logic (*e.g.* progression between states is prompted by temperature exposure, the values of which are drawn from a corresponding EXCEL sheet).

5.3 Validation of the agent-based model of *E. saccharina*

Three means of validation were conducted in order to gain insight into the accuracy with which the simulation output mimics the in-field behaviour of *E. saccharina*. These methods were insight gained from industry expert opinion, comparison of simulation output to video footage and parameter sensitivity analysis. Each of these three steps in the model validation process is discussed in further detail in this section.

5.3.1 Face validation

Law [52] suggests that a simulation model attains credibility when a decision maker or key project personnel accepts the model as "correct." He formulates the following questions which should be answered during such face validation:

- (i) Do you agree with the assumptions being made in the model?
- (ii) Are the processes employed in the model to recreate appropriate scenarios verified? (*i.e.* do they perform as expected, based on the model specifications?)
- (iii) Have you been involved with the design and iterative development of the model during its construction?
- (iv) Are the model developers reputable researchers and do they demonstrate professionalism and criticality and produce quality outputs?
- (v) Is the animation output of the simulation model compelling and does it represent, to the best of your knowledge and expertise, a similar situation to that which is experienced in the real-life scenario which the model is attempting to recreate?

Conlong [19] is affiliated with *SASRI* and is one of the leading experts in South Africa on *E. saccharina* behaviour and control. Conlong's insight into the intricacies of *E. saccharina* behaviour, both as individuals and as a population, is amongst the best currently available and, as such, he was deemed a reliable expert in terms of reviewing the simulation model, and providing commentary and feedback on its credibility. Conlong was consulted throughout the course of the development of the research and, once the appropriate point in the study had been reached, a meeting was conducted whereby assumptions, inclusions, exclusions and general approaches used in the model could be described and discussed in order to ascertain how realistically and appropriately these elements had been incorporated. This was done both in principle from a design perspective, as well as based on the visible animation output of the running model. More specifically, Conlong was asked to assist in adding validity to areas of the simulation model that are based on aspects of *E. saccharina* in which there exist little or no surety about the exact behaviour of the pest, or where no relevant, reliable data are available against which to test the output of a simulation model.

The elements of the model which were discussed with Conlong are briefly described and the primary approach and assumptions are reiterated (discussed fully in §4) for convenience. Input from Conlong, where relevant, is also included with respect to each discussed aspect. It is important to note that the face validation process initially conducted was simply an interim validation to evaluate the model's accuracy and realism with which it mimics the behaviour of *E. saccharina* as it was best understood at the particular stage in the study. This served as a progress point from which to depart with more intensive testing which was conducted using the model. A formal, comprehensive face validation, consisting of a number of experts from a variety of areas of expertise, is reported on at a later stage in this dissertation.

The selection of oviposition sites:

The two available means through which the female agent can be informed to select oviposition sites in the simulation were described to Conlong. The point was affirmed that no conclusive evidence exists regarding the manner in which this process occurs in nature and that the algorithms had been developed by consultation of similar studies conducted on insects, as well as intuition based on the typical resulting spread and propagation of an *E. saccharina* population. A discussion was conducted on the supposed ability of the female to sense more mature sugarcane which, in turn, may provide more suitable cryptic sites for oviposition. Conlong commented that the female's sense is not necessarily restricted to mature sugarcane, but possibly dead material which is found on this sugarcane. The manner in which the algorithms of §4.6.2, through different approaches, allow the user to specify the adeptness of this sense was described, as well as the converse case where a large degree of randomness can be chosen to be present in the process. All possible modifications and selection criteria within these algorithms which have been incorporated as user-defined parameters were explained to Conlong and his input was inquired with respect to any other implementations which should be made. No further additions or inclusions were suggested at this stage in the research as significant insight from an entomological view point does not exist.

The dispersion of newly hatched larvae:

In a previous communication, Conlong described a process whereby neonatal larvae climb up sugarcane stalks under which they hatch and 'parachute' off the leaves to surrounding stalks before burrowing in to feed on the interior nutrients (§2.5). The manner in which this early dispersal is implemented in the simulation model of this dissertation was discussed with Conlong, as well as the realistic distances across which larvae may be able to move during this process. The following stage whereby larvae then select a stalk into which to

burrow was also discussed, including the assumption made in the model that larvae will select the stalk nearest to their landing site for feeding purposes post-dispersion. Conlong responded that there may be some form of selection process followed by larvae in order to choose a stalk for burrowing, but little or no insight exists pertaining to this possible process and, as such, he was content with the implemented assumption.

The underlying sugarcane structure:

The manner in which an array was generated to represent physical sugarcane stalks within the continuous model space, as well as the manner in which larvae and ovipositing females locate these specific stalks for burrowing and oviposition site purposes, respectively, was described to Conlong. Furthermore, the inclusion of the simplified CANEGRO model which is used to simulate the growth of the sugarcane over time was demonstrated. A follow-up discussion to a conversation with Conlong at an earlier stage in the research also took place on the relative maturity values assigned to the sugarcane stalks (and subsequent implementation of four possible underlying sugarcane structures) in the simulation. Furthermore, the previously agreed-upon manner in which more mature sugarcane positively affects the future strength and competitiveness of the agents was also reiterated. Conlong agreed with this implementation and the associated assumptions, commenting that it formed a good departure point for such an inclusion in the model which may be developed further as future work, accompanied by possible further insight into the process as it occurs in nature.

The carrying capacity of a sugarcane stalk:

The process implemented to accommodate the promotion rate of larvae feeding within sugarcane stalks to pupae, upon consumption of sufficient nutrients, was described to Conlong. The simplifying assumption that, once fed on, damaged areas of sugarcane stalks do not regenerate in any manner to accommodate further feeding larvae, but new stalk portions which grow at a rate determined by the CANEGRO model may provide new feeding sites, was discussed. Stemming from this discussion, the assumption made that a stalk's capacity is determined by the quotient of its current length and the length of stalk required per larvae to take in sufficient nutrients for maturation (as defined by the user), as well as the assumption that any larvae which attempt to feed on a particular stalk that is already infested up to its capacity do not progress to the next life stage and perish, was analysed. Conlong described the actual process followed in nature, whereby a sugarcane stalk may be infested beyond its capacity and larvae who receive insufficient nutrients may still pupate — developing into very weak, non-competitive moths. It was thus concluded that it is valid to assume that these moths will have very little effect on the propagation of a population and, as a simplifying assumption, may be removed at the larvae stage of the life cycle.

The absorbing boundary condition:

The inclusion of the absorbing boundary condition within the simulation model, whereby agents have the ability to enter and exit the simulated space during its execution, was presented to Conlong. The model assumes that only females who either cannot find a mate, or (in some cases) cannot find sufficiently good quality, unoccupied oviposition sites may disperse and leave the simulation space. Similarly, only females who are currently ovipositing may enter the simulation space, in search of oviposition sites. Furthermore, any agents who exit the simulated hectare by means of small-scale random movement as the population spreads to the boundaries of the simulated space are also removed from the model. Following a discussion about the nature of dispersal flights among *E saccharina*, it was agreed upon that these were good assumptions to include in the model. Upon observ-

ing the animation output of females entering the simulation space to oviposit eggs, Conlong commented that the resulting spread of eggs very closely mimicked what is typically found in nature. It was further agreed that the rate/likelihood of entering females should be a user-controlled parameter for scenario testing purposes, since no certainty exists about the rate of such an occurrence in the field.

The mating window and possibility of multiple mating instances by a single female:

The handling of cases whereby a female may, according to a certain probability, mate more than once was discussed with Conlong and it was found that this element had been incorrectly implemented in the simulation. Although rare, a single female has the potential to mate up to three times, but this only occurs when the sperm of one male insufficiently fertilizes the eggs already present in the abdomen of the female. This means that even females who mate three times will still only lay the typical 120 eggs in total, as opposed to the number of mating instances multiplied with the typical 120 eggs, as was previously implemented in the model. Conlong commented that a female may, in fact, oviposit far more eggs in a single oviposition cycle, but typically only 120 of these eggs survive and, as such, this is an appropriate number to implement in the simulation model.

Another important consideration which required inclusion in the model is the approximate two-day post-emergence mating window imposed on females before they become incapable/unwilling to mate. This means that if a female does not successfully mate within 48 hours after emergence from the pupae, it is assumed that she does not mate at all during her life time. This important change was noted as of potential importance to the propagation of a population and number of eggs oviposited in the simulation.

In all cases, discussions were held on the implementation of the aforementioned divisions within the simulation model and the back-end code and statecharts were stepped through with Conlong for thorough inspection. This was accompanied by a thorough analysis of the animation output of the processes being executed in the simulation itself, as well as the general performance of the model as a whole when executed over a prolonged period of time, incorporating large numbers of agents.

5.3.2 Comparison with video footage of *E. saccharina*

Video footage taken of *E. saccharina* moths interacting and mating was obtained from Mudavanhu [61]. In the videos, small numbers of male and female moths are allowed to interact in a clear box and their resulting behaviour is captured. Most pertinently, the videos aimed to document mating activity and behaviour of the pest.

The activity which occurs in the videos is in line with the observations of Atkinson [5] as documented in the literature. The moths in this case are, however, in captivity and not in nature as was the case in [5]. This means that the box was likely to be saturated with the odour of the attractant pheromone and, as such, is not a good indicator of behaviour pertaining to the search, location and approach process which occurs before mating.

The most notable activity witnessed in the videos which is not formally documented in the literature is the conflict which often transpires between females competing for the same *lek*. In cases where more than one female approached the same *lek* (in this case, there were only sufficient numbers of males to form a single *lek*), females moths were observed to fight or compete to ‘occupy’ and, if victorious, make use of the *lek*. Inferior females either abandoned the *lek* and moved away from the vicinity of the conflict, or remained in close proximity, attempting to

find a male after the more dominant female had begun mating. This phenomenon was discussed further with Conlong [19] who confirmed that this sort of conflict is believed to occur in the field too.

In the simulation model, females do not ‘compete’ as such to occupy *leks* and, in all cases, the first arriving female takes preference to occupy a *lek*. In order to simulate the case where a first arriving female does not necessarily take preference of occupation of a *lek*, the selection of a mating partner from the contributors to that *lek* is implemented according to a certain probability. In the simulation model, there exists a 75% probability that the female will consider a male as a potential mate. If the probability threshold is not exceeded during the execution of the **F** ChooseIndividualMate function, the female skips over this particular male and does not consider him as a potential mate. The effect of this results in the female sometimes not choosing the male with the strongest display from a *lek* for mating purposes. This means that a second (or later) arriving female who occupies the *lek* formed by the remaining males once the first arriving female has begun mating sometimes has the opportunity to select and mate with the original, most competitive male who forms part of that *lek*. This effectively models a scenario where the first arriving female may have been ‘fought off’ by a second arriving female who would then go on to select the strongest male for mating, leaving the original first arriving female to select a mate from the remaining *lek* contributors.

All further mating activity which was observed in the video footage appeared to be both in line with the literature, as well as with what had been implemented in the simulation model.

5.3.3 Parameter variation and sensitivity analysis

Figure 5.2 contains the results of one-year simulation runs using each of the two available oviposition site selection algorithms included in the simulation model (as described in §4.6.2). The figures show the full population numbers as tracked throughout the simulation runs in an attempt to provide insight the population dynamics as a whole. Each simulation runs for a full year, starting from a recent harvest (October) and the results are represented as e/100s¹ values over the simulated hectare for comparative purposes. In these simulations, all parameters are set to the ‘best guess’ values which were determined through model contribution insight, experimental analysis and various testing scenarios.

An idealised population curve which is typically predicted for an *E. saccharina* population in sugarcane is included in the figure. This idealised curve represents the expected growth of an *E. saccharina* population in sugarcane as derived from the data-validated model of Potgieter [64] and expert insight from Conlong [19]. Typically, a population is expected to propagate slowly when infestation levels are low, with an increasingly exponential growth as the population size increases. This growth curve then flattens out during the colder winter months when the pest takes longer to develop and, as such, the population propagation slows down.

As may be seen in the figure, the general population curve shape is similar to the population growth generated by the simulation to some extent. There are, however, some significant differences. Very large peaks are encountered at intermittent steps in time. These peaks result from intensive mating and population propagation when many adult moths exist in the simulation space, but numbers are significantly reduced when agents begin penetrating sugarcane stalks and the carrying capacity of the sugarcane takes effect. Such a phenomenon would not typically be observed in nature owing to the manner in which the data are collected at discrete time steps which are far apart and the results of which are averaged over the test area. The simulation

¹Eldana per 100 stalks. This is a typical measure of infestation levels.

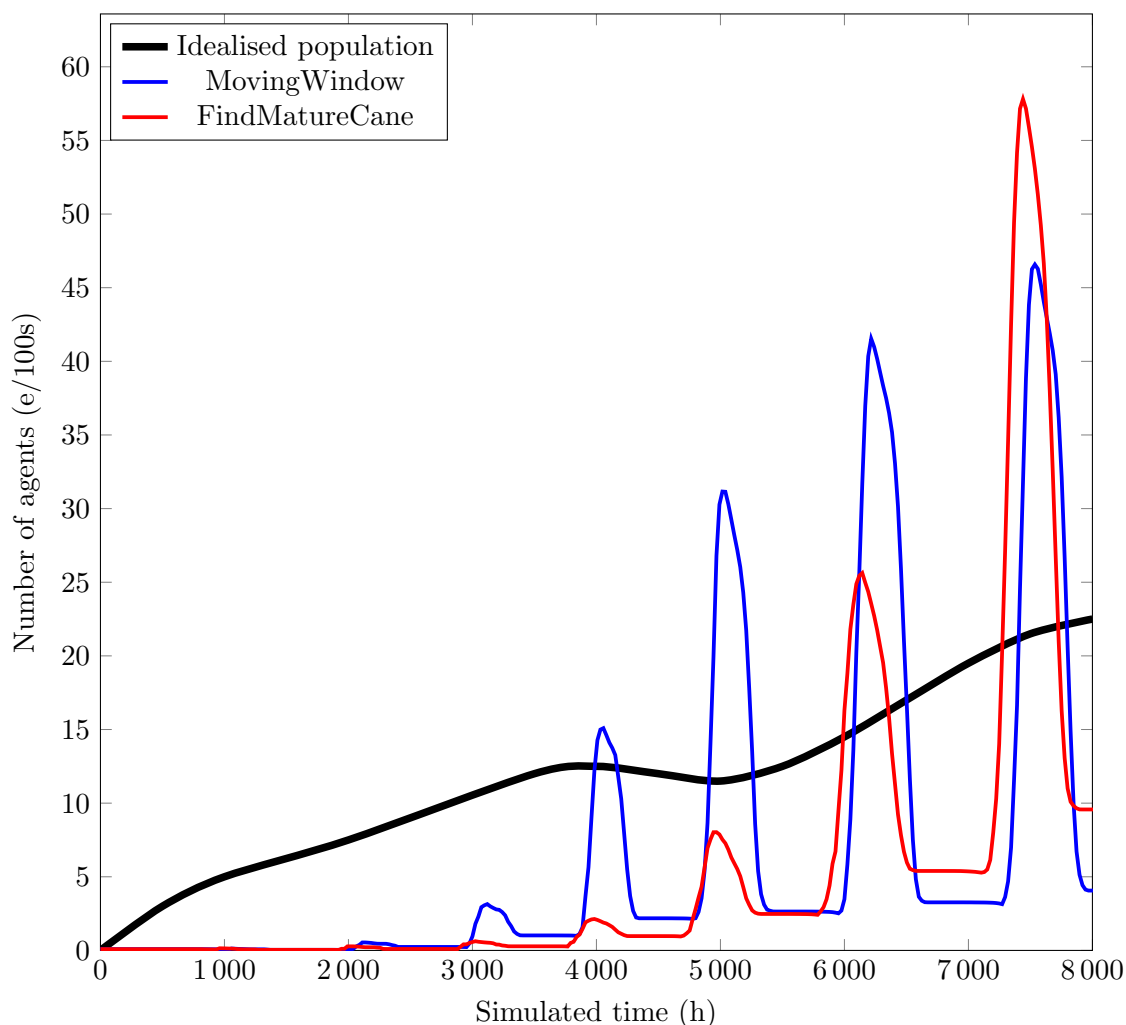


FIGURE 5.2: A comparison between the total number of agents per 100 simulated sugarcane stalks for a full year using both oviposition site selection algorithms of §4.6.2 compared to the idealised population growth curve which is believed to mimic the manner in which a population of moths grows within a field.

data also show that the population takes longer to propagate than is anticipated by the idealised curve, but, when it does, growth occurs at a faster exponential rate than is assumed by the idealised curve. Infestation levels achieved by the stable simulated population (the population which is not affected by the carrying capacity of the sugarcane) are also notably lower than the typical infestation levels which, after a year, can range anywhere from 20–60 e/100s.

In light of these differences, it follows that the factors influencing the population growth and behaviour require investigation and quantification in order to validate the model.

The most common and, often, insightful method of model validation involves recreating scenarios already recorded in real-life experiments, comparing the output data of the simulation model to those which were gathered during the in-field experiments in order to determine the degree to which they agree. In the case of *E. saccharina*, data are limited and sporadic owing to a number of complications, both in conducting experiments, as well as in the actual gathering of the relevant data. Ultimately, the simulation model in this study aims to serve as a tool within the investigation and planning process of establishing manners to control *E. saccharina* and, as

such, should possess the ability to recreate any given behaviour or population dynamics desired by the operator.

A pilot sensitivity analysis was, thus, conducted using the model to determine the parameters which alter the simulation behaviour in the most significant way and, in doing so, discover the tools available to model users to manipulate the simulation behaviour so as to mimic the specific scenario which they would like to investigate. Sensitivity analysis consists of changing the values of the input and internal parameters of a simulation model and determining the resulting effect on the model behaviour or output [73].

One important aspect in any sensitivity analysis is to define the starting conditions which are assumed at the beginning of a simulation run. In nature, a population of *E. saccharina* is largely eradicated during the harvesting season when sugarcane is cut and removed from the field. In almost all cases, however, a very small percentage of eggs and larvae remain in the roots and stalk bases of the sugarcane [19]. These remaining eggs and larvae are then responsible for the propagation of another infestation as the sugarcane grows post-harvest.

In order to simulate the propagation of a population of *E. saccharina* post-harvesting, the initial infestation and infested area of the simulated hectare space are included in the simulation as user-defined parameters. Using these parameters, the initial number of agents in the simulation can be determined by

$$N_{initial} = \frac{\delta}{100} \times A \times 129\,600,$$

where $N_{initial}$ represents the number of agents which populate the simulation space, δ represents the initial infestation level ($e/100s$) and A represents the initial area infested. This is then multiplied by the total number of stalks in the hectare so as to calculate the corresponding number of agents constituting an infestation of $\delta e/100s$ over area A .

The initial condition specification of a simulation is of utmost importance when performing a sensitivity analysis as it generally has a direct influence on the outcome of a simulation execution. For this reason, initial conditions were simplified and kept constant over the various runs, so as to minimise the influence of the initial spread of agents on the simulation outcome. All simulations conducted over the course of the sensitivity analysis assume an initial infestation level of 0.1 $e/100s$ and an infestation area of 100% of the simulated hectare. Typically, the starting population is randomly dispersed across the simulated space, resulting in small pods of agents in areas where infestation typically propagates. This initial layout varies greatly between runs and results in a large range of rates of population expansion (which directly depends on the number of pods or clusters in the population). In an attempt to mitigate these effects, twelve defined nodes were established among which the agents were spaced within the simulation space. A snapshot of the layout of these nodes within the simulation space is shown in Figure 5.3. These starting conditions were found to provide the most similar arrangement and infestation levels over the different simulation runs. It is, however, recommended that different initial conditions under which simulations can be run and the resulting impact on the population dynamics and propagation be investigated in future work.

The simulation model presented in this dissertation makes use of a large number of variables and it would be impossible to perform an in-depth sensitivity analysis on each component within the time-frame of this study in order to determine its effect on the overall model. For this reason, the analysis was confined to the variables that are included in the model as user-defined parameters as they will serve as tools which model users can employ to recreate specific simulation scenarios.

In order to determine which user-defined parameters exert the most notable change on the performance of the simulation when varied, ten simulation runs lasting for a simulated time of

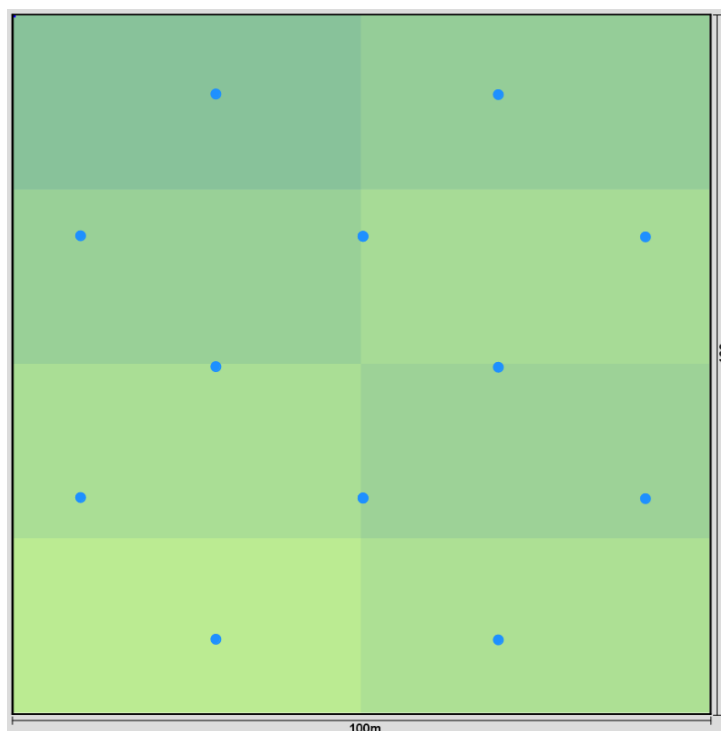


FIGURE 5.3: The starting conditions used in each of the experiments conducted in the pilot sensitivity analysis. Agents were uniformly distributed amongst the twelve nodes.

six months each were conducted. Each of the relevant parameters were tested at a ‘best guess’ base level, a small increase and a small decrease. The increase and decrease were typically 10% upwards and downwards, respectively, but in some cases a specific higher and lower value were selected in the region of the percentage change. In total, nine parameters were tested² for each of the two oviposition site selection algorithms of §4.6.2. The algorithms, in turn, each have two parameter sets themselves which required testing. Each case was then repeated ten times in an attempt to mitigate exceptional cases or out-of-the-ordinary behaviour. This resulted in 352, six-month long simulations being executed — an equivalent ‘real time’ of 176 years. In each case, a number of informative statistics were recorded from each simulation run. The results of the ten experimental runs for each case investigated were averaged and are shown in Tables 5.2–5.5. Table 5.1 contains a list of the parameters which were varied, as well as ‘keys’ with which the parameters are referenced in subsequent results tables. This table also indicates the unit of measure used for each parameter varied, as well as the ‘best guess’ value, established from model building insight, performance testing and, where possible, expert opinion. The full set of results for each independent test conducted during the sensitivity analysis may be found in Tables A.1–A.4 in Appendix A.

In the tables, the mean infestation level ($I_{\mu_{total}}$) is a measure of the average number of larvae which infested each stalk in the simulated space over the duration of each entire simulation run. This differs from the current infestation level ($I_{\mu_{current}}$), which represents the average number of agents larvae present in each stalk at the time instant when the simulation ended. The standard deviation of the total infestation over the duration of the simulation run ($I_{\sigma_{total}}$) is a comparison between the infestation of each individual stalk over the course to the ($I_{\mu_{total}}$) of that run and gives an indication to what degree the infestation was evenly spread throughout

²In some cases, parameters were tested as ‘sets’ owing to similarities in their areas of influence within the model.

Parameter	Key	Unit	Default Value
ScanRadius	(SR)	metres	12.5
Proximity	(Prox.)	metres	0.15
<i>Probability of mortality:</i>			
	($P_{mortality}$)		
Egg		% true	3
LarvaeOutsideStalk		% true	11.5
LarvaeInsideStalk		% true	0.9
Pupae		% true	0.7
Adult		% true	20
Erlang rate (FlightAmbition)	(λ_{Erlang})	None	1.5
ProbabilityofDisplaying	($P_{disp.}$)	% true	75
ProbabilityofScanning	(P_{scan})	% true	75
MaxDispersion (larvae)	(LD_{max})	metres	0.5
<i>Moving Window Algorithm:</i>			
Perception Distance	(PD)	metres	2.5
Window Increments	(WI)	% of FlightAmbition	2.5
CaneQualityThreshold	(CQT)	None	1.15
<i>FindMatureCane Algorithm:</i>			
CaneSearchRadius	(CSR)	metres	15
WindowSize	(WS)	radians	0.52
FinalBatchProximity	($BatchProx_{final}$)	metres	0.5

TABLE 5.1: Default parameter values included in the sensitivity analysis with a reference key, unit of measurement and default, ‘best guess’ value.

the simulated space. The current standard deviation ($I_{\sigma_{current}}$) performs the same calculation, but exclusively for the infestation present at the time instant when the simulation is terminated. The ‘e/100s’ value represents the number of larvae and eggs currently present in the simulation as a measure per 100 simulated sugarcane stalks, whilst the ‘max. e/100s’ value represents the highest such infestation levels experienced over the course of the simulation. Similarly, the record of the ‘Current eggs & larvae’ and ‘Max. eggs & larvae’ represent the total agents in those states at the termination of the simulation and the highest recorded level during the course of the simulation run, respectively. It was chosen to record the number of agents in the **Egg** and **Larvae** states because the greatest changes to the population numbers in subsequent simulation runs were observed in these states. The carrying capacity, in particular, resulted in a large number of larvae failing to mature to the **Pupae** state and, since the sensitivity analysis aimed to test which parameters had the most significant influence on aspects of the simulation model such as this, the number of agents in these states was deemed important to record. Finally, a maximum total population level reached is recorded, as indicated by the ‘Max population’ column.

When assessing the averaged sensitivity analysis results presented in Tables 5.2–5.5, the outcomes of the sensitivity analysis appear to agree with intuition and experience gained from using the model. Increases in certain parameters, such as SR, Prox., LD_{max} and λ_{Erlang} , appear to cause a general subsequent increase in the population numbers and damage indication data for both oviposition site selection algorithms, but exceptions to these general trends also exist.

In examining the full set of sensitivity analysis data in Appendix A, it may be seen that there are large ranges of output values from simulation runs when using identical parameter values and, as such, the sensitivity analysis does not offer a definitive conclusion on the influence of

Varied parameter	Value	$I_{\mu_{total}}$	$I_{\sigma_{total}}$	$I_{\mu_{current}}$	$I_{\sigma_{current}}$	e/100s	max e/100s
None		0.06	0.69	0.05	0.52	4.63	11.61
SR	15	0.07	0.69	0.05	0.51	4.78	12.23
	10	0.06	0.63	0.04	0.47	4.14	11.24
Prox.	0.05	0.01	0.26	0.01	0.17	0.63	1.10
	0.25	0.14	1.03	0.10	0.79	10.07	35.28
PD & WI	2.75	0.04	0.55	0.03	0.42	2.90	6.25
	2.25	0.06	0.65	0.04	0.49	4.44	9.96
CQT	1.05	0.07	0.69	0.05	0.51	4.74	13.18
	1.175	0.05	0.59	0.04	0.43	3.65	8.89
$P_{mortality}$	↑10%	0.04	0.52	0.03	0.38	2.82	7.55
	↓10%	0.05	0.56	0.03	0.43	3.41	8.43
λ_{Erlang}	1.25	0.04	0.50	0.03	0.37	2.55	6.52
	1.75	0.06	0.64	0.04	0.48	4.20	9.46
$P_{disp./scan}$	0.65	0.03	0.48	0.02	0.35	2.29	5.02
	0.85	0.10	0.85	0.07	0.63	7.16	20.80
LD_{max}	0.25	0.01	0.30	0.01	0.21	0.73	1.89
	0.75	0.13	0.97	0.10	0.74	9.77	24.26

TABLE 5.2: Aggregated damage indication data collected during the sensitivity analysis using the ‘Moving Window’ oviposition site selection algorithm of §4.6.2.

Varied parameter	Value	$I_{\mu_{total}}$	$I_{\sigma_{total}}$	$I_{\mu_{current}}$	$I_{\sigma_{current}}$	e/100s	max e/100s
None		0.05	0.59	0.04	0.44	3.68	9.44
SR	15	0.24	1.07	0.21	0.97	20.79	21.24
	10	0.25	1.08	0.22	0.98	21.45	21.86
Prox.	0.05	0.23	1.03	0.19	0.93	19.18	19.44
	0.25	0.48	1.48	0.37	1.37	37.09	10.72
CSR & WS	13.5 & 0.26	0.31	1.22	0.27	1.13	27.30	29.15
	16.5 & 0.79	0.34	1.41	0.30	1.30	30.20	32.05
$BatchProx_f$	1.05	0.16	0.88	0.13	0.79	13.22	13.79
	1.175	0.26	1.10	0.22	1.00	22.03	22.64
$P_{mortality}$	↑10%	0.24	1.05	0.20	0.95	20.12	20.67
	↓10%	0.33	1.27	0.28	1.14	27.88	28.97
λ_{Erlang}	1.25	0.43	1.44	0.37	1.31	36.96	37.72
	1.75	0.23	1.07	0.20	0.97	19.46	21.20
$P_{disp./scan}$	0.65	0.14	0.80	0.12	0.72	12.09	12.49
	0.85	0.16	0.87	0.13	0.78	13.22	13.46
LD_{max}	0.25	0.31	1.26	0.27	1.15	26.88	27.41
	0.75	0.18	0.98	0.15	0.89	15.33	15.79

TABLE 5.3: Aggregated damage indication data collected during the sensitivity analysis using the ‘Find Mature Cane’ oviposition site selection algorithm of §4.6.2.

Varied parameter	Value	Current eggs & larvae	Max. eggs & larvae	Max population
None		6 014	15 094	18 263
SR	15	6 212	15 903	19 714
	10	5 378	14 613	17 491
Prox.	0.05	818	1 426	1 705
	0.25	13 088	45 858	56 793
PD & WI	2.75	3 775	8 123	9 929
	2.25	5 766	12 954	16 514
CQT	1.05	6 160	17 130	21 138
	1.175	4 742	11 553	14 473
$P_{mortality}$	↑10%	3 660	9 810	11 119
	↓10%	4 435	10 961	13 220
λ_{Erlang}	1.25	3 316	8 480	10 126
	1.75	5 462	12 294	15 325
$P_{disp./scan}$	0.65	2 981	6 521	8 035
	0.85	9 310	27 042	33 227
LD_{max}	0.25	945	2 457	2 881
	0.75	12 700	31 543	40 353

TABLE 5.4: Aggregated population data collected during the sensitivity analysis using the ‘Moving Window’ oviposition site selection algorithm of §4.6.2.

Varied parameter	Value	Current eggs & larvae	Max. eggs & larvae	Max population
None		4 780	12 272	14 843
SR	15	27 032	27 608	32 312
	10	27 891	28 422	33 261
Prox.	0.05	24 928	25 274	29 523
	0.25	48 211	52 935	60 758
CSR & WS	13.5 & 0.26	35 492	37 896	43 563
	16.5 & 0.79	39 264	41 669	47 958
$BatchProx_{final}$	1.05	17 190	17 923	20 764
	1.175	28 634	29 431	34 469
$P_{mortality}$	↑10%	26 161	26 872	31 320
	↓10%	36 249	37 665	43 959
λ_{Erlang}	1.25	48 054	49 033	57 614
	1.75	25 293	27 566	31 696
$P_{disp./scan}$	0.65	15 718	16 235	18 746
	0.85	17 186	17 501	20 541
LD_{max}	0.25	34 938	35 630	41 840
	0.75	19 929	20 527	24 336

TABLE 5.5: Aggregated population data collected during the sensitivity analysis using the ‘Find Mature Cane’ oviposition site selection algorithm of §4.6.2.

each parameter. Some individual simulation runs show expected results, based on the parameter varied, but are countered by other runs with resulting statistics of, sometimes, an entirely different order of magnitude. This diversity and inconsistency of results may be attributed to the large number of assumptions and accompanying randomness incorporated into the model. In the numerous cases where insufficient knowledge of *E. saccharina* exists to accurately construct a process which mimics its real-life behavior, an assumed or derived process was implemented which makes use, in most cases, of several random variables or samples from appropriate distributions as input parameter values which direct movement and behaviour. This results in a large range of possible behavioural outcomes for each agent in the simulation; a phenomenon which is intensified when thousands of such agents interact in the same space.

In time, as more entomological research is conducted, however, it is anticipated that these assumed processes may be refined so as to accurately replicate the true behaviour of *E. saccharina* and, in doing so, reduce randomness in the model. Until such time, the best possible solution that can be achieved is to incorporate the assumptions in a manner which allows for a diversity of outcomes in subsequent simulation runs in an attempt to mimic a variety of possible scenarios of the resulting population dynamics.

In spite of the poor correlation of parameter variation to resulting model performance, the numerous iterations of model design and corresponding test executions of the model have provided insight into the general influence of the different parameters within the simulation model performance. In all cases where assumptions were incorporated into the model, a number of experimental simulations were conducted to test these assumptions, resulting in a large range of model results. These results, along with expert opinion and insight, informed further model refinements and assisted in better quality assumptions and iterative improvements of the model in terms of more accurately reflecting the pest's behaviour. Future developments in accuracy, as mentioned, will require insight from entomological research and experimentation.

In view of the insight gained from model construction, as well as the pilot sensitivity analysis whose results were reported on in this section, it was deemed necessary to attempt to investigate and give a general indication of the influence of the most notable parameters included in the simulation model. These parameters are implemented as user-specified parameters in the simulation model and may be used to recreate a diverse range of simulation runs which tend towards different population dynamics and outcomes. Paramount to harnessing the power of these changeable parameters is an understanding of the manner in which they are perceived to influence the simulation outcome in the general case. For this reason, an in-depth parameter variation analysis was conducted to assist in explaining the effect of each variable inclusion in the model. The results of this analysis are presented in the following chapter.

5.4 Chapter summary

In this chapter, the various different verification techniques used during model construction were discussed. These included the use of an IRC, animation output analysis and traces in computer code. Thereafter, the validation process followed was described. This process included a face validation by Conlong [19], a leading industry expert on *E. saccharina* who has been closely involved with the construction of the model since the start of the research. Thereafter, videos which were taken of *E. saccharina* mating in captivity were examined and compared to the implementation of the mating process in the simulation model. Finally, a pilot sensitivity analysis was conducted on the most notable parameters in the simulation in order to determine the manner in which they influence the model outputs. Although insightful, the results from

the sensitivity analysis were not consistent enough to be regarded as conclusive and a parameter variation analysis was proposed in order to assist in understanding the conceptual effects of each parameter in the simulation model so that it may be used as a tool for scenario recreation and analysis going forward.

CHAPTER 6

Parameter variation analysis

Contents




6.1	The ScanRadius and Proximity parameters	110
6.2	The stage-specific random or predator mortality rates	119
6.3	The ‘rate’ of the Erlang distribution	123
6.4	The influence of the probability of scanning and displaying of agents	123
6.5	The influence of larvae dispersal among the sugarcane stalks	127
6.6	The oviposition site selection algorithms	132
	6.6.1 General dispersal rates	132
	6.6.2 The effect of the FlightAmbition parameter	142
	6.6.3 Biased dispersal based on sugarcane maturity values	148
6.7	Chapter summary	152





In this chapter, the manner in which the most significant parameters influence the execution of the simulation and the behaviour of the agents therein are analysed more thoroughly. These parameters were identified by insight from model construction and further informed by the results of the pilot sensitivity analysis. Where appropriate, simplified forms of control measures which link to these parameters are implemented in the simulation model to illustrate the flexibility and capability of the model to act as a decision support tool for pest management. Auxiliary experiments are conducted over a time frame of approximately 6 months from various initial conditions depending on the type of analysis being done. This is done in an attempt to isolate, as best possible, the effects of the parameters in question and minimise influence of other random variables or samples employed in the simulation. Each parameter and consequent investigation is conducted in an effort to provide rudimentary insight to and understanding of the nature of the influence of the parameter. An in-depth analysis should, however, be conducted for each specific case in future research. This should also be accompanied by intensive entomological research to assist in substantiating and/or minimising assumptions in each case. In all experiments, but for those pertaining specifically to comparisons between the two available oviposition site selection algorithms, the **F** MovingWindow function is employed in the model to select oviposition sites. This algorithm was chosen as it is less computationally expensive and is similar in construct to egg-laying site selection algorithms seen in other research, providing credibility toward its approach.







The analysis conducted serves to validate the abilities and diversity of the model, and affirms its benefit as a tool for scenario recreation and ‘what-if’ analyses pertaining to *E. saccharina*.



Each of the relevant parameters is discussed based on understanding gathered both from model construction and numerous testing and debugging. Where relevant, these observations are linked to and substantiated by an accompanying experiment.



6.1 The ScanRadius and Proximity parameters

Both the  ScanRadius and  Proximity parameters affect the model during the mating cycles of *E. saccharina*. Numerous test simulations revealed that, during the early stages of lower infestation, smaller  ScanRadius values often resulted in populations dying out because females were not near enough to *leks* in order to be attracted for mating. Similarly, in cases where the proximity bound for *lek* formation was too strict, very few (or no) *leks* were formed and, as before, the population frequently failed to progress.

Conversely, in cases where the population did explode within the simulated space, it was noted that both the  ScanRadius and  Proximity parameters played a less important role in the further propagation of the population. This is attributed to the density of agents within the simulation space: Even for small values of the  ScanRadius and  Proximity parameters, such large numbers of available agents within the simulated space meant that even strict conditions were always satisfied and infestation increased with population propagation.

From these observations, it may be deduced that both the  ScanRadius and  Proximity parameters have the most pronounced effect on the simulation during the early stages of infestation post-harvesting. In cases where the  ScanRadius is large, initial populations are likely to grow at a faster rate than in cases when it is smaller. Similarly, if the value of the  Proximity parameter is large, more *leks* form and, as a result, more mating typically occurs than in cases when it is very small. Practically, this may suggest that control measures such as SIT, which affect the pest at the point of mating, should be investigated in terms of implementation at low population densities. Irradiated moths have different mating capabilities which would be maximised at a time where the mating behaviour of the pest appears to be most vulnerable. Naturally, if the actual  ScanRadius and  Proximity parameter values in nature are too small, SIT may fail to operate effectively at densities which are too low. On the other hand, however, releasing at large densities would require more irradiated moths which is, in turn, more expensive. In light of this, using the model to simulate the control measure may provide important insight into the release strategies and numbers of released moths when imposing SIT.

The observations discussed above were substantiated by performing a number of 6-month long simulation runs with varied values attributed to the  ScanRadius and  Proximity parameters and observing the rate of growth of the population. In each case, the initial conditions for the experiments represented an initial infestation level of 0.1 e/100s spread across the entire simulation space. To assist in mimicking the population post-harvesting, a function was called which clustered agents in close proximity with one another. Such resulting ‘pods’ effectively mimic small groups of agents in stalks who survive the harvesting period. The results of the tests are shown in Figures 6.1–6.6.

As may be seen in Figures 6.1–6.3, a larger  ScanRadius appears to result in a larger population. In the experiments conducted, it was noted that there exists a so-called ‘threshold’ value in a sense that once the population size reaches this number, it generally explodes during the next mating season, irrespective of the size of the  ScanRadius adopted by females during mating. This threshold appeared to be once the population of larvae and eggs reaches approximately 2000 agents.

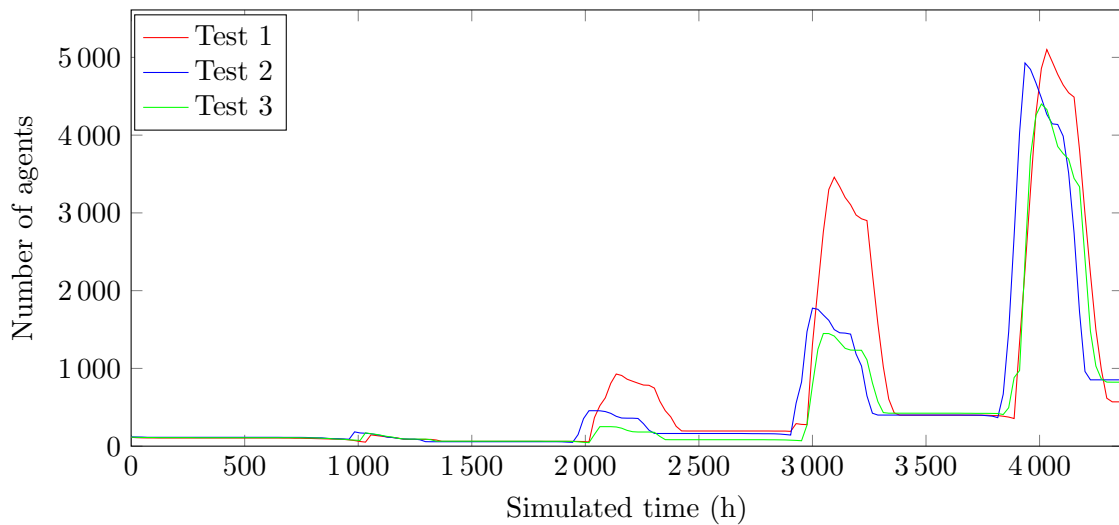


FIGURE 6.1: Scan Radius set to 10m.

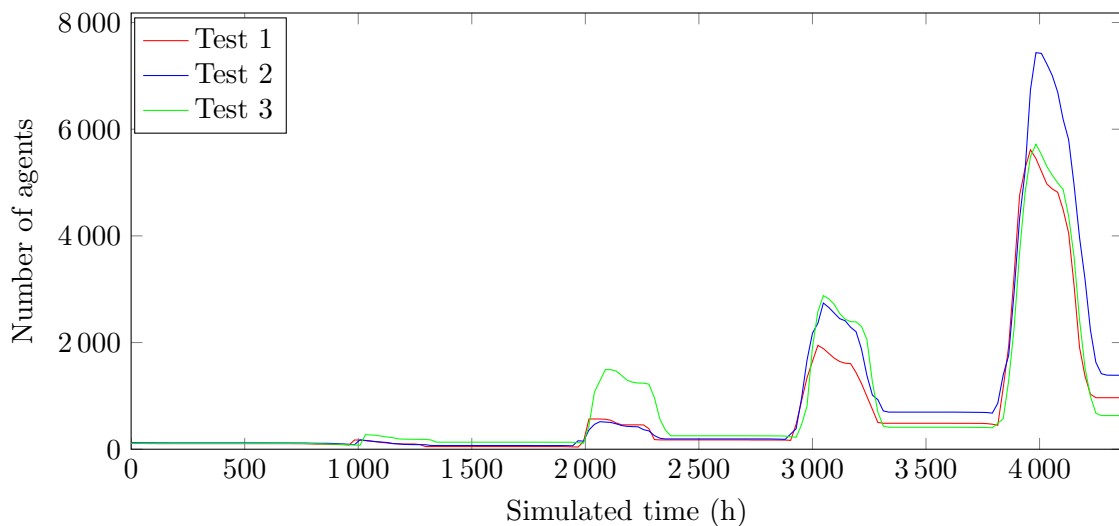
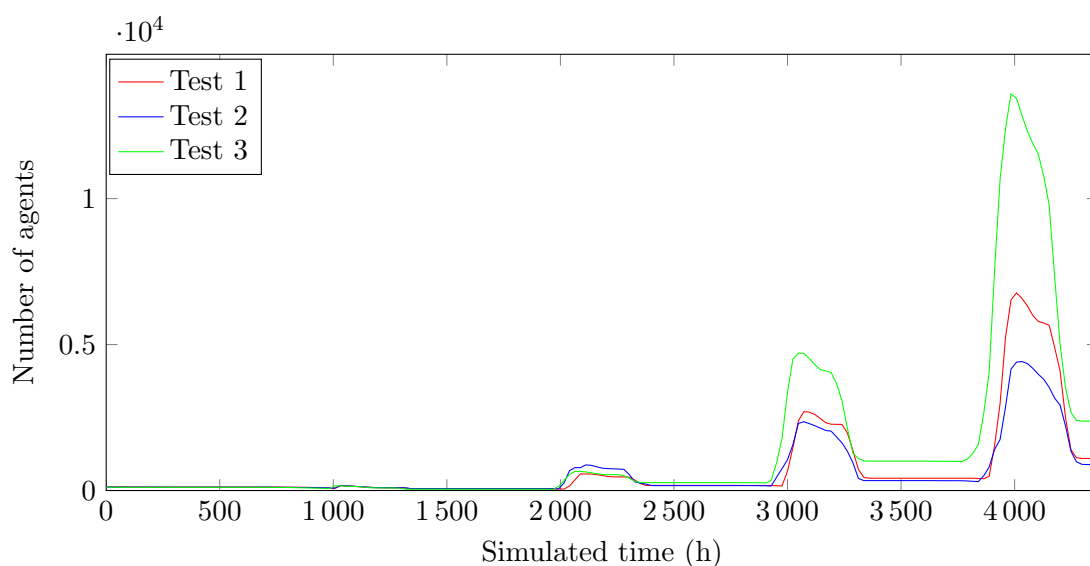
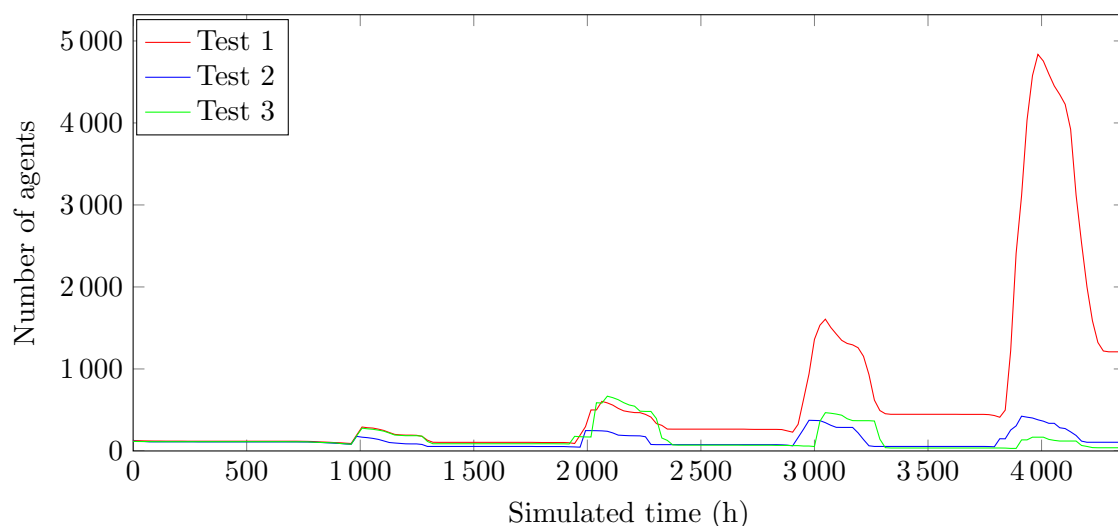




FIGURE 6.2: Scan Radius set to 12.5m.



Owing to the random dispersion of the population, in many of the simulation runs, an initial infestation failed to propagate and the agents simply died. Such simulation runs were not recorded, but it was noted that a smaller ScanRadius value generally required far more attempts to generate an infestation breakout than higher ScanRadius values in an experiment. This is intuitive in that larger values allow for mating to occur between agents who are slightly further spaced than in the case of smaller values. As such, smaller ScanRadius values often resulted in only one or two mating instances during the first population cycle which meant the population took longer to reach the ‘threshold’ value where it began to explode in size.

The resulting behaviour caused by variation of the Proximity parameter is similar in nature to that of the ScanRadius parameter. Imposing strict, small bounds on the minimum distance between males over which a *lek* can form resulted in slow propagation of the population and required a very large number of attempts in order to generate an infestation breakout. As

FIGURE 6.3: *Scan Radius set to 15m.*FIGURE 6.4: *Proximity set to 0.05m.*

may be seen in Figures 6.4–6.6, larger proximities generally result in a larger population and more rapid growth to reach the ‘threshold’ value, at which point the effect of the size of the mating threshold parameters appears to decrease.

In order to assess the effect of a large or small  **ScanRadius** and  **Proximity** parameter combination, further experiments were conducted, the results of which are shown in Figures 6.7–6.8.

As is evident from the results shown in the figures, a combination of low parameters attributed to the  **ScanRadius** and  **Proximity** parameters results in a slower propagation of the population as opposed to the case when higher values are chosen. It may be seen in Figure 6.7 that the population takes longer to reach the ‘threshold’ value from where population numbers explode and, as such, reaches a lower total population value over the course of the 6-month

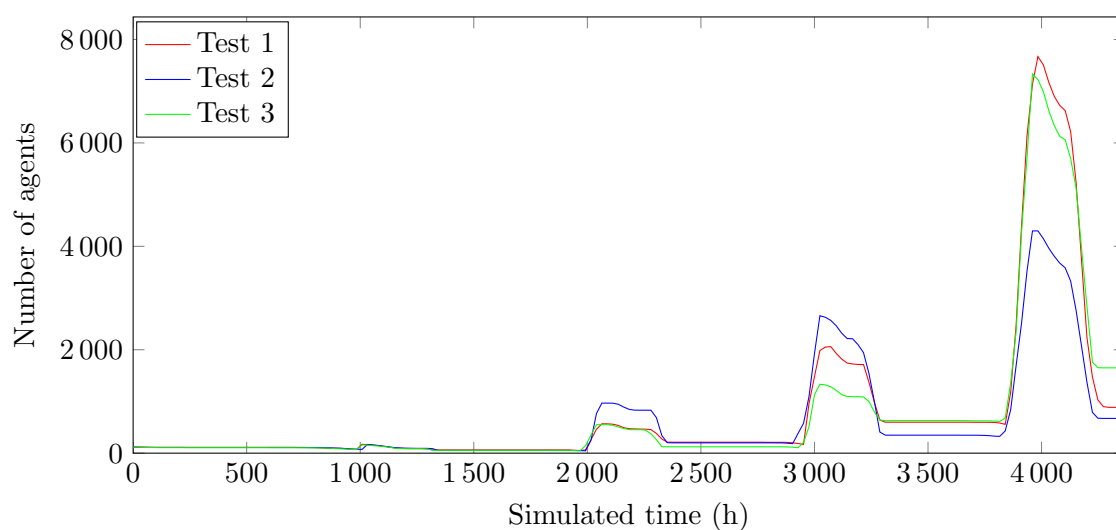


FIGURE 6.5: Proximity set to 0.15m.

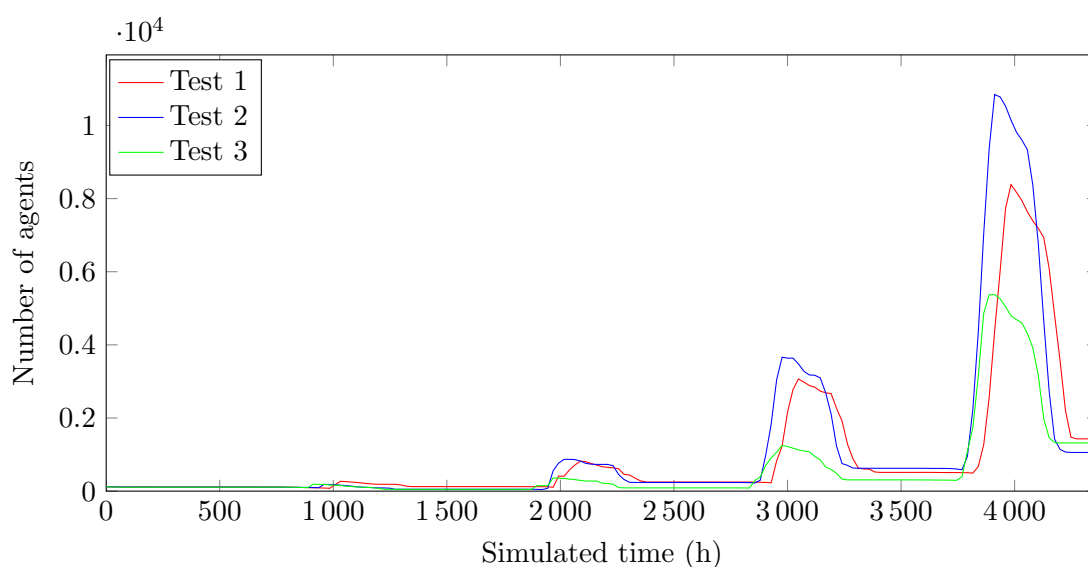




FIGURE 6.6: Proximity set to 0.25m.

simulated period. In Figure 6.8, the threshold point is reached at a significantly earlier stage during the simulation and, as such, the population reaches a much larger size in the experiment.

The observations pertaining to the values of the  ScanRadius and  Proximity parameters shed light on issues which must be considered or investigated pertaining to the successful implementation of SIT as a control measure on *E. saccharina*. The number of irradiated moths released, as well as the frequency or timing of the release is of utmost importance when employing this control measure. Using an agent-based model of *E. saccharina* may accommodate testing of different release strategies and investigation of their predicted affect on a population of *E. saccharina*. In order to illustrate the possibility of such testing, a simplified example of SIT was developed and incorporated into the simulation model to test its effect on a number of cases.

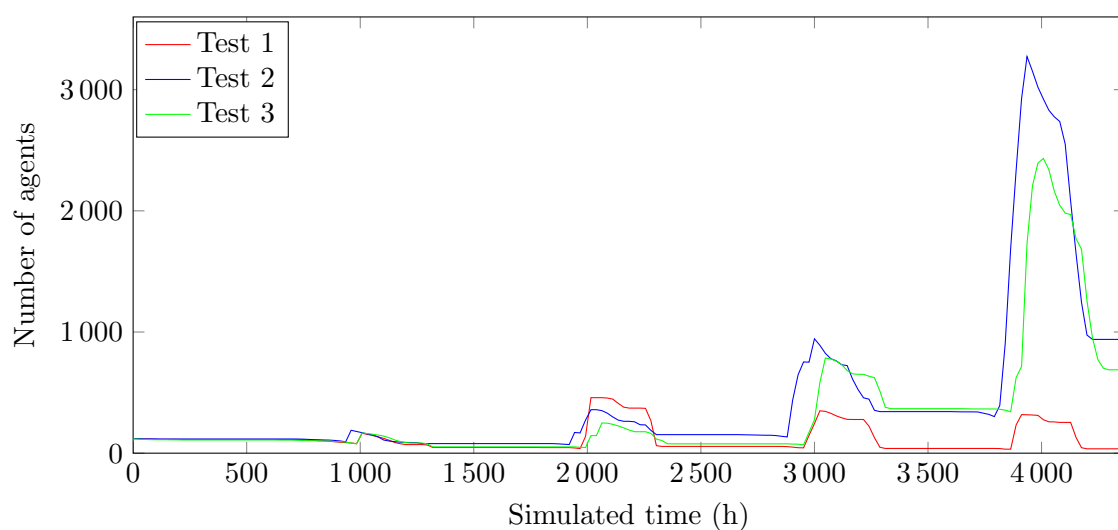


FIGURE 6.7: Minimum ‘scan radius’ and ‘proximity’ parameter values selected for mating.

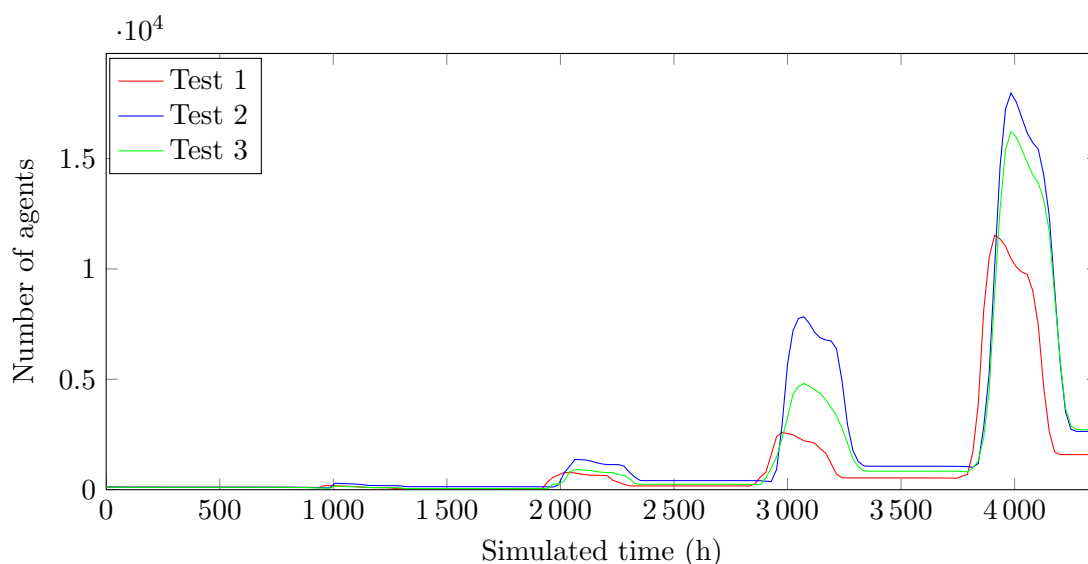


FIGURE 6.8: Maximum ‘scan radius’ and ‘proximity’ parameter values selected for mating.

The simplified version of SIT, as incorporated in the agent-based model, includes both irradiated female and male moths which are released amongst a population of *E. saccharina* at a certain time-step in the simulation. The dosage of radiation typically given to moths used in SIT results in sterile females, but only semi-sterile males, since females are more sensitive to the radiation than males [89]. This means that when an irradiated female mates with a non-irradiated male, she produces offspring which do not survive. An irradiated male, however, when mating with a non-irradiated female, will result in a generation of eggs which will survive, but are themselves sterile. This means that, should these offspring go on to mate with non-irradiated moths in the future, their offspring will not survive.

This simplified version of SIT was incorporated in terms of parameters which were used to track the sterility of females and males in the simulation and their consequent offspring. The results of the different experiments conducted are shown in Figures 6.9(a)–6.11(a). In the first



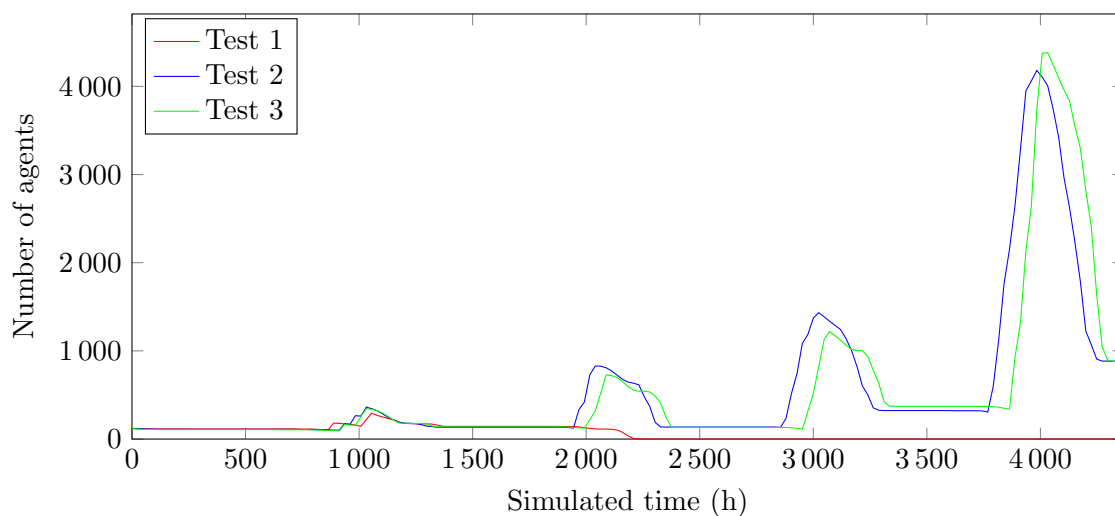
experiment, an SIT population of 50 semi-sterile males and 50 semi-sterile females was released amongst the existing, non-sterile *E. saccharina* population. Not all of these agents survive post-release and some perish with a probability equal to that of non-sterile agents in their respective adult states. This population represented an initial infestation of 0.1 e/100s (approximately 130 agents) concentrated within the centre of the simulated space, infesting 10% of the simulated area. The sterile population was released at the point when the non-sterile population began to mature from the `Pupae` state to the `AdultMale` and `AdultFemale` states, allowing all agents equal opportunity to mate with one another. The  ScanRadius and  Proximity parameters were set to their ‘best guess’ values of 15m and 0.15m, respectively.

Figure 6.9(a) shows the population growth over time for three experiments incorporating the release of an SIT population, whilst Figure 6.9(b) is a cumulative count of all agents present in the simulation which are, in some way, affected by SIT, other than those agents introduced to the simulation by means of the SIT release (*i.e.* non-sterile females who have mated with semi-sterile males and will now propagate sterility in the population, as well as any offspring resulting from a mating instance involving an agent released as part of the SIT population).

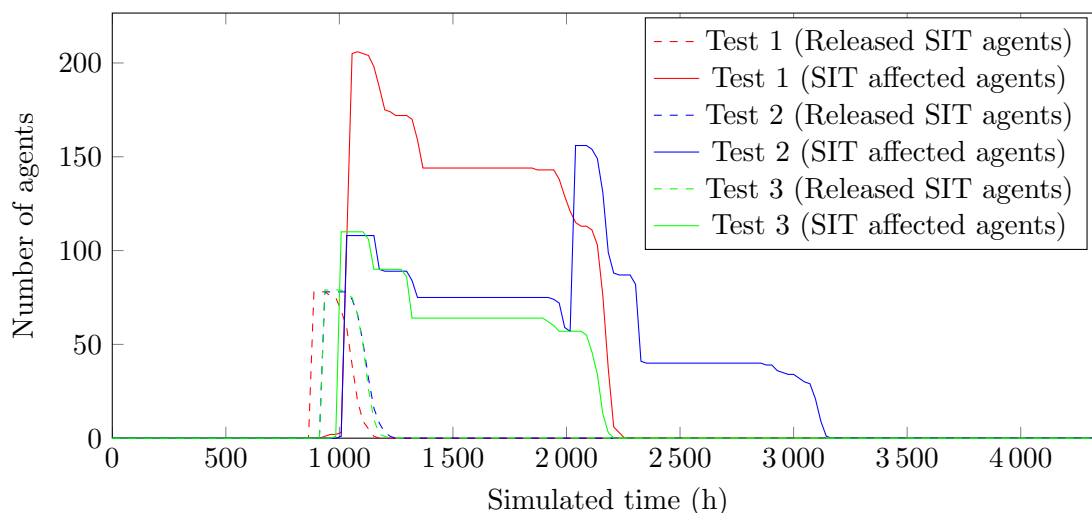
In Figure 6.9(a), the SIT release in ‘Test 1’ was a success and caused the entire *E. saccharina* to be exterminated. In ‘Test 2’ and ‘Test 3’, however, the SIT population interacted and mated with the non-sterile agents, propagating sterility within the population, but there were too many incidents of non-sterile mating interaction still present in these experiments, allowing the population to persevere through the SIT release and continue to propagate. The introduction of sterile agents in these cases did, however, assist in slowing population development somewhat. In Figure 6.9(b), the sudden rise in the number of SIT-affected agents results from a period of mating and oviposition within the simulated population. As may be seen in ‘Test 1’, a large number of agents were affected by SIT (either through mating with semi-sterile males or being oviposited after a mating instance involved an SIT affected agent). This resulted in the population dying out, since the next generation oviposited by the numerous SIT agents does not survive. In ‘Test 2’ and ‘Test 3’, fewer agents were affected by exposure to the sterile population and, as such, the non-sterile population managed to persevere and survive past the point where all SIT agents present in the simulation had died out. This is in spite of ‘Test 2’ showing a significant second peak in the number of SIT-affected agents as a result of mating between SIT-affected agents. This may be attributed to the sterile agents being concentrated in a small portion of the simulated space, or simply too many non-sterile agents being present for the sterile population to maintain a notable effect.

In the second set of SIT-related experiments, the results of which are shown in Figures 6.10(a) and 6.10(b), ‘Test 1’ and ‘Test 2’ involved the same initial conditions as the previous experiments, but the SIT population in ‘Test 2’ was only released during the second instance of maturation of agents from the `Pupae` state to the `AdultMale` and `AdultFemale` states. In ‘Test 3’, the release timing occurred once again at the first maturation instance as before, but, in this case, the initial infestation level was doubled (approximately 260 agents), whilst the number of released SIT agents remained fixed at 100 agents in total.

As may be seen in Figure 6.10(a), the effect of SIT release is minimal when moths are released later and with a larger initial infestation, represented by ‘Test 2’ and ‘Test 3’, respectively. Considered in conjunction with the subsequent number of SIT-affected agents in each test each case, shown in Figure 6.10(b), it may be seen that a larger number of agents were affected by SIT in ‘Test 3’, when the initial infestation was larger. This follows intuition since more mating instances between sterilised and non-sterilised agents would typically occur. In spite of this, the abundance of non-sterile agents resulted in the unaffected population continuing to flourish, minimally impacted by the SIT release. This corresponds with the sensitivity analyses



(a) Population growth.



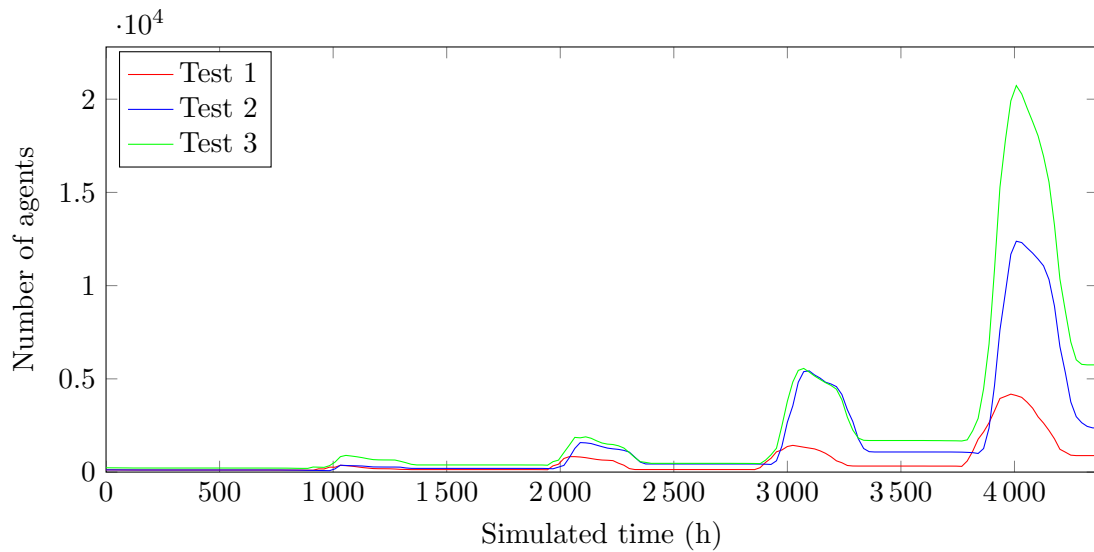
(b) Sterile agents present in simulation.

FIGURE 6.9: SIT release of 100 agents during the first cycle of adult maturation.

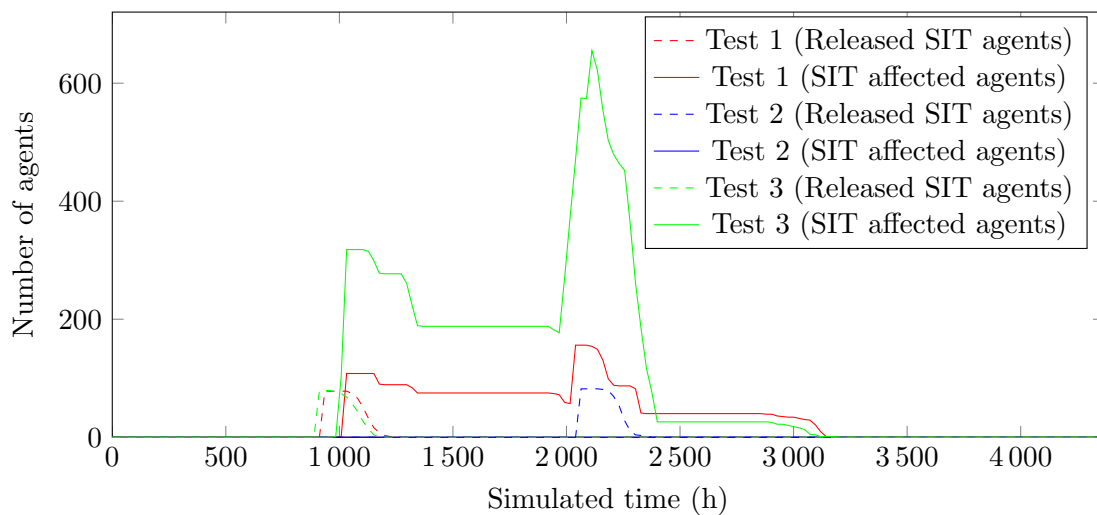
by Potgieter *et al.* [64] where it was shown that the success of SIT is dependent on the initial population size.

When released at a later stage, as shown in ‘Test 2’ of Figure 6.10(b), the SIT population did not manage to infiltrate the non-sterile population in order to result in a notable number of SIT-affected agents. This is largely as a result of the fixed release site of the SIT population. During the early release tests, the bulk of the non-sterile population still resides within the central 10% of the simulation space and, since the SIT population is also released within this same central area, the interaction between the two populations is maximised. When released at a later stage, the non-sterile population has had time to disperse away from the centre of the simulated region, resulting in far less interaction with the sterile agents. This highlights the importance of the release sites selected for SIT, corresponding with the observations made by conducted by Potgieter *et al.* [65].

Figures 6.11(a) and 6.11(b) track the number of eggs which are oviposited, but fail to survive,



(a) Population growth.

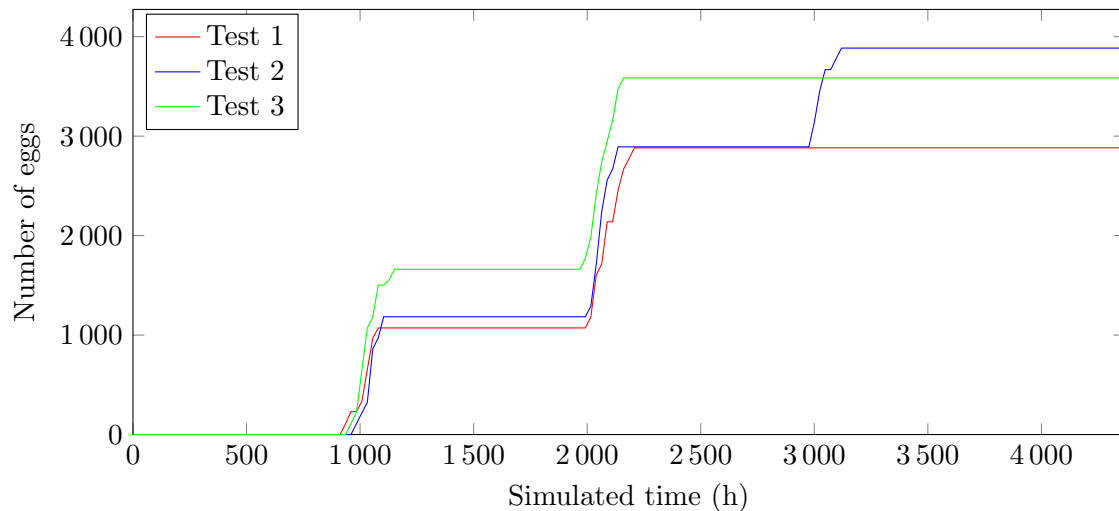


(b) Sterile agents present in simulation.

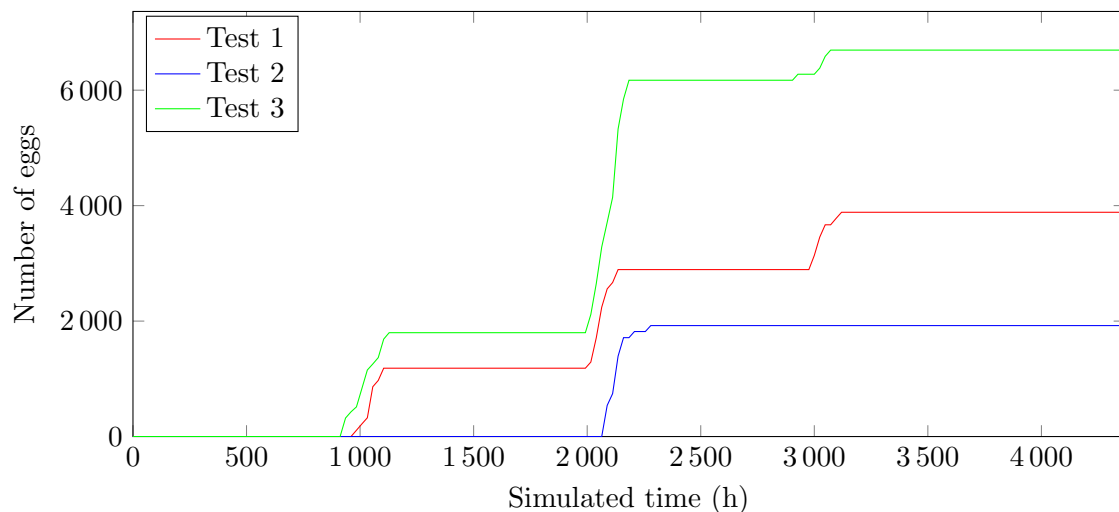
FIGURE 6.10: SIT release of 100 agents for different simulated conditions.

in the simulation. Such eggs are either oviposited by females who were released during the SIT implementation and mated successfully with a male, or they result from mating interactions by the sterile generation of agents who are oviposited by females who mated with semi-sterile males during the SIT release. Both figures illustrate that in all cases, the implementation of SIT did result in the suppression of the pest in some manner, although the significance of these numbers of non-surviving eggs must be considered in the context of the size of the greater population of which it forms a part. As may be seen in Figure 6.11(b), despite little or no agents being affected by the SIT release in ‘Test 2’, a number of non-surviving eggs were still oviposited. These eggs therefore must result from fully sterile, released females who mate either with other semi-sterile or non-sterile males. This suggests that, despite the SIT population being released in a different location to where the non-sterile population resided in the simulation, some females may still be attracted to and occupy *leks* formed by non-sterile males further away. The semi-sterile males released during SIT do not, however, travel over significant distances and, as such, the *leks*

formed by these males are likely to be overpowered or less attractive to the non-sterile females than those in close quarters, formed by non-sterile males. This is a general concern pertaining to SIT [19].





(a) The number of eggs oviposited in the simulation which fail to survive owing to parents having been affected by SIT during the experiments documented in Figures 6.9.





(b) The number of eggs oviposited in the simulation which fail to survive owing to parents having been affected by SIT during the experiments documented in Figures 6.10.

FIGURE 6.11: The total number of non-surviving eggs oviposited as a result of parents being affected by SIT.

This simplified implementation of SIT does not thoroughly incorporate the many intricate factors pertaining to the accurate implementation of the control measure in a real-life case, but was included to illustrate the flexibility of the model in its ability to accommodate control measure testing and tracking of the impact which the measures have. As mentioned previously, in the test cases, the implementation of SIT certainly appeared to have more of an effect when infestation levels are still relatively low and the size of the released population is similar to that of the non-sterile population. Similarly to the influence of the  ScanRadius and  Proximity parameters, once the number of agents reaches a ‘threshold’ point, the population appears to

propagate in spite of any hindrances which may attempt to be imposed. More intricate factors pertaining to release timing, strategies and sterile population sizes are proposed as future work to build on the work done by Potgieter [66].

6.2 The stage-specific random or predator mortality rates

As noted in Chapter 4, random and predator mortality is actually a temperature-dependent phenomenon, but was chosen to be a fixed value owing to the manner in which it is simulated in this model. Simulation runs in which the mortality rates were fluctuated by small amounts showed little or no emergent behaviour and no definitive model outcomes can be attributed to these parameter values. As in the case of the  **ScanRadius** and  **Proximity** parameters, the period in the simulation of most significant influence was seen to be during stages of low infestation where portions of the population being eradicated by means of random or predator mortality have the potential to halt further mating and consequent population propagation.

In reality, experiments have been carried out to determine mortality rates with a fair amount of accuracy under laboratory conditions [91]. Based on the understanding and model insight, minor differences between the recorded values and actual, in-field values are likely to have little or no effect in the model. That said, the fluctuation of these parameters can also be used to test the effect of applying pesticides in terms of determining when is the most suitable time to spray crops. Experiments were conducted to mimic this control measure and investigate differing spraying strategies and pesticide strengths on a population.

The first set of experiments concerned differing frequencies of pesticide spraying on a crop. In this simplified case, the pesticide was deemed to exclusively affect larvae which had not yet penetrated a sugarcane stalk for feeding and had a 60% potency (*i.e.* each larva outside of a sugarcane stalk at the time of pesticide spraying had a 60% chance of mortality). The initial conditions in these experiments were the same as those used in the SIT experiments in the previous section. The results of these test cases are documented in Figures 6.12–6.14 and the data collected on the population dynamics are shown in Table 6.1.

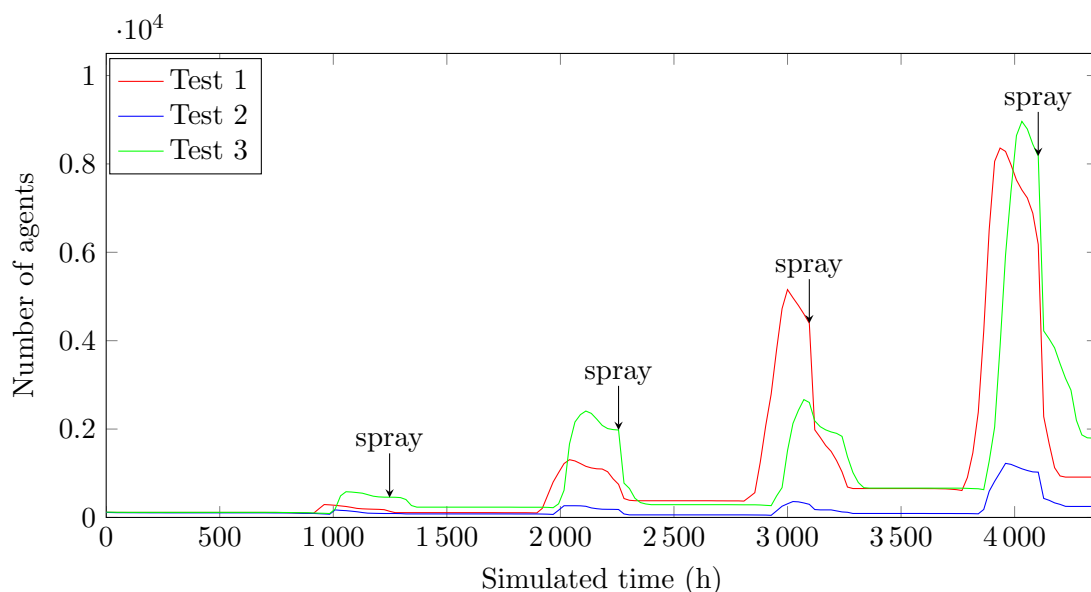


FIGURE 6.12: Four pesticide sprays of 60% potency released over 6 months.

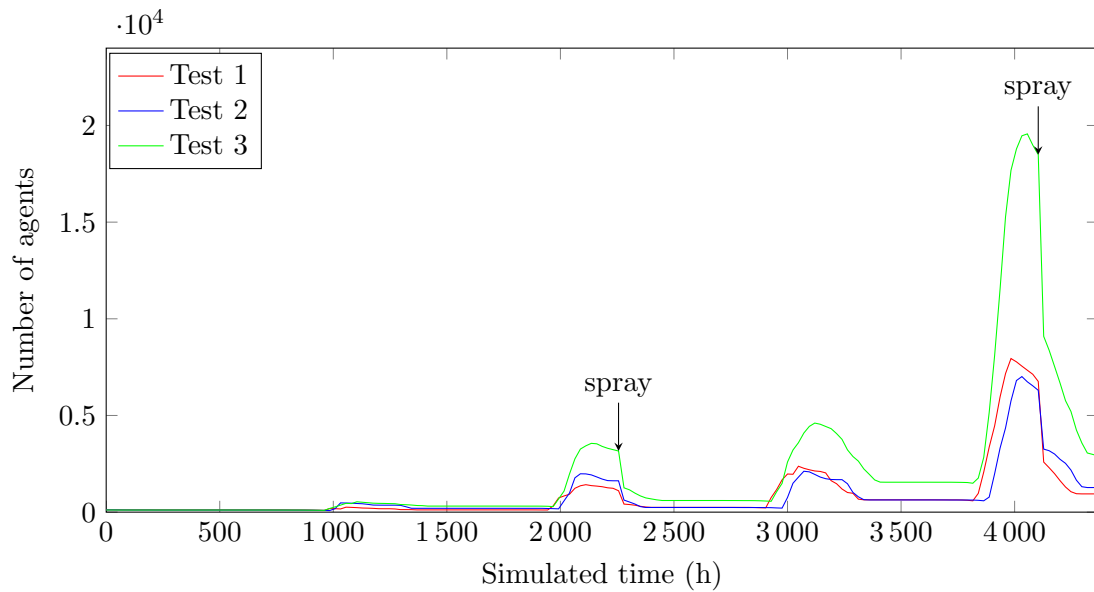


FIGURE 6.13: Two pesticide sprays of 60% potency released over 6 months.

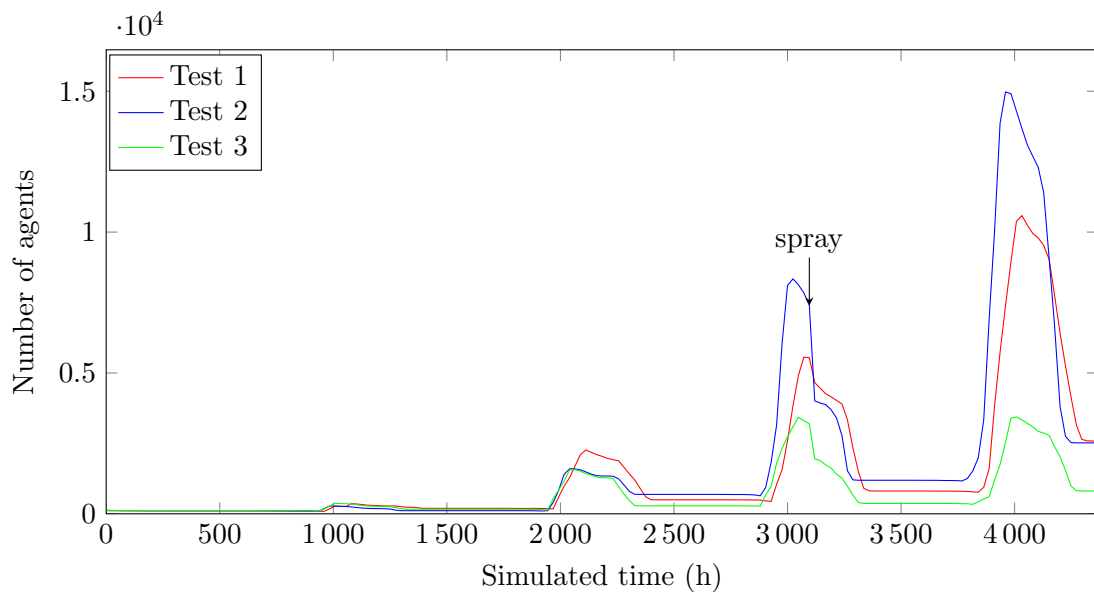


FIGURE 6.14: One pesticide spray of 60% potency released over 6 months.

Figures 6.12–6.14 show, as one might expect, that more frequent sprays of pesticide generally result in populations which grow at a slower rate. Table 6.1 confirms this observation, showing that the mean infestation levels ($I_{\mu_{total}}$) also generally remain lower during more frequent sprays.

The rate at which the population spreads is also an important aspect to be considered. There exist various complexities surrounding the detailed calculation of a dispersal rate in an agent-based model which falls outside the scope of this general discussion. That said, the standard deviation of the total infestation over the duration of the simulation run ($I_{\sigma_{total}}$), as well as the current standard deviation of the population at the time instant when the simulation ended ($I_{\sigma_{current}}$), were calculated in an effort to indicate the degree to which the infestation has spread

Strength	# sprays	I_{μ_total}	I_{σ_total}	$I_{\mu_current}$	$I_{\sigma_current}$	e/100s	max e/100s	Max pop.
60%	4	0.017	0.366	0.007	0.181	0.702	5.672	8 359
	4	0.005	0.160	0.002	0.103	0.192	0.888	1 224
	4	0.024	0.434	0.014	0.276	1.381	5.876	8 962
60%	2	0.016	0.360	0.007	0.186	0.726	4.912	7 945
	2	0.019	0.381	0.010	0.218	0.978	4.758	7 012
	2	0.042	0.590	0.022	0.349	2.224	11.871	19 565
60%	1	0.032	0.525	0.020	0.348	1.955	7.018	10 585
	1	0.036	0.552	0.019	0.334	1.937	10.286	14 973
	1	0.013	0.336	0.006	0.184	0.622	2.308	3 440

TABLE 6.1: *Damage indicators during testing of differing pesticide spray strategies.*

across the simulation space. The standard deviation is calculated by comparing the infestation level of each individual stalk to the mean infestation level, both over the entire duration of the simulation run, as well as at the end. A larger standard deviation suggests that the population has spread to some degree within the simulation space, since the infestation levels of the sugarcane stalks are notably different. This does not, however, consider the distribution of this spread and the measure may be skewed as a result of a localised infestation ‘pod’ concentrated at one point in the outlying areas of the simulation. As mentioned, a more detailed analysis of the dispersal of agents during simulation experiments is therefore necessary. As may be seen in Table 6.1, I_{σ_total} is nevertheless generally larger in experiments with less frequent sprays, suggesting that there is more significant dispersal of agents in these cases.

Whilst conducting the experiments, the importance of the timing of the pesticide application became evident. As mentioned, a assumption was made that the pesticide only affects larvae which are yet to burrow into stalks [24]. The timing of the spray should therefore typically be chosen so as to maximise the number of agents exposed to the poison. In the experimental cases, the timing of the pesticide sprays were fixed and identical for all tests. If during a certain experiment, a population matured at a slower or greater rate than usual, the pesticide spray could have little or no impact on the population. This phenomenon is well illustrated at the second pesticide spray as applied to ‘Test 1’ in Figure 6.12 where it appeared that all larvae had already penetrated sugarcane stalks by the time the pesticide was sprayed.

A further set of experiments was conducted to determine the effect of pesticides with different strengths or potencies. These experiments involved one spray of the pesticide at the same time step within the simulated 6-month period for comparison to the results shown in Figure 6.14. The results corresponding to pesticides with differing strengths are summarised in Figures 6.15 and 6.16, and information about the population dynamics is recorded in Table 6.2.

As may be seen when comparing Figures 6.13, 6.15 and 6.16, a stronger pesticide assists in keeping populations lower and prolongs the time taken to reach the ‘threshold’ value from which population explosion propagates. In all cases of the pesticide use, the number of larvae that perish as a result of the control measure is further decreased by the effect of carrying capacity when the remaining larvae begin burrowing into sugarcane stalks. This suggests that the effectiveness of the pesticide is not as much the number of agents it eradicates, but rather the manner in which it ‘thins’ the remaining population to make further mating and oviposition more difficult and assist in slowing the population growth.

The data in Table 6.2 should be interpreted similarly to those of Table 6.1 in that the mean total infestation and standard deviation of that infestation is lower for instances where a stronger

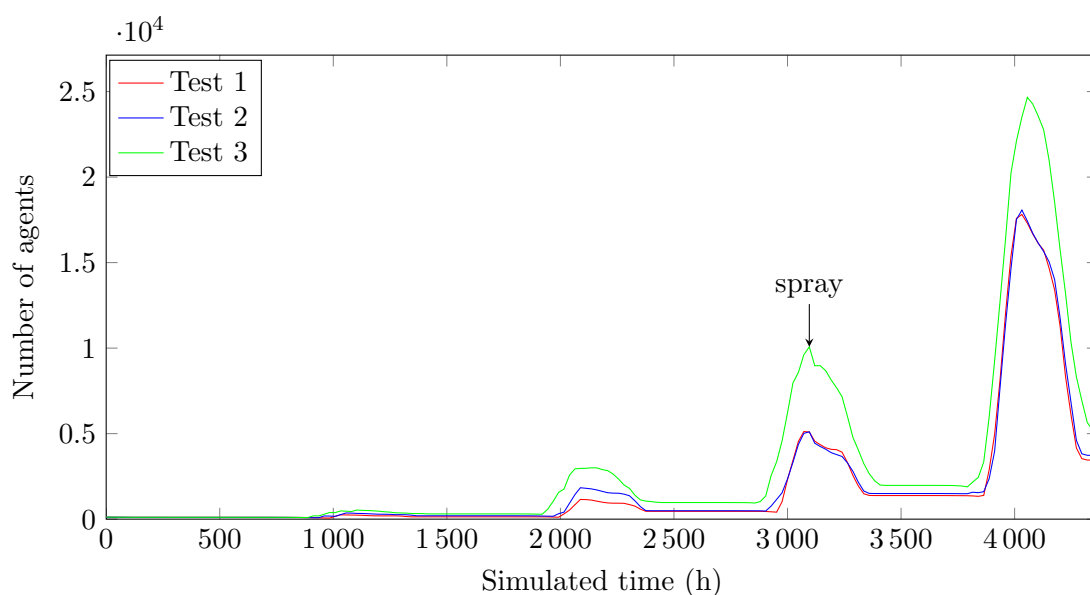


FIGURE 6.15: One pesticide spray of 40% potency released over 6 months.

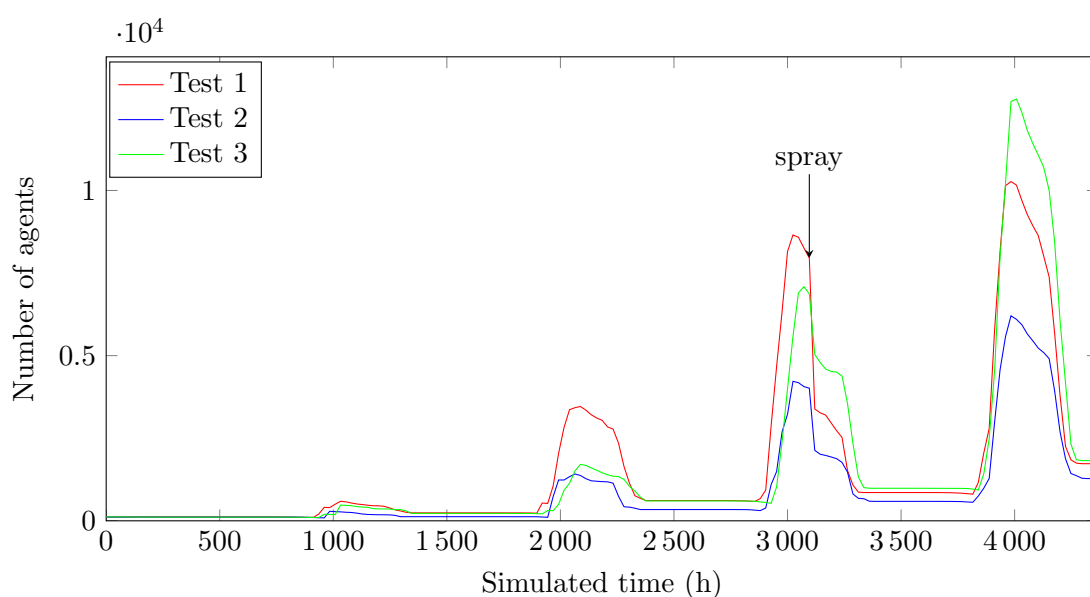


FIGURE 6.16: One pesticide spray of 80% potency released over 6 months.

pesticide is used. In this case, the maximum $e/100$ s reached in each case also correlates to the nature of the pesticide where a larger infestation was observed during the experiments using a weaker pesticide. This is typical of an experiment where the population numbers are higher, but is not always the case, as observed in Table 6.1 where higher maximum and final $e/100$ s were observed to be higher, even in instances involving a greater number of pesticide applications.



These experiments pertaining to the use of pesticides are not intended to draw accurate conclusions about the type of pesticide or spraying strategy which should be used to best suppress an *E. saccharina* population. They do, however, effectively indicate the possibility for using the agent-based model in future work to model and analyse the pesticide effect as a temporary

Strength	# sprays	I_{μ_total}	I_{σ_total}	$I_{\mu_current}$	$I_{\sigma_current}$	e/100s	max e/100s	Max pop.
40%	1	0.043	0.617	0.027	0.410	2.657	11.786	17 813
	1	0.047	0.647	0.029	0.425	2.871	12.100	18 070
	1	0.066	0.752	0.040	0.479	3.945	13.609	24 665
60%	1	0.032	0.525	0.020	0.348	1.955	7.018	10 585
	1	0.036	0.552	0.019	0.334	1.937	10.286	14 973
	1	0.013	0.336	0.006	0.184	0.622	2.308	3 440
80%	1	0.028	0.494	0.013	0.272	1.330	7.118	10 270
	1	0.019	0.397	0.010	0.235	0.983	4.193	6 206
	1	0.029	0.502	0.014	0.270	1.402	8.888	12 769

TABLE 6.2: *Damage indicators during testing of differing pesticide spray strengths.*


increase in the transition threshold controlling the random or predator mortality of an agent within its statechart.

6.3 The ‘rate’ of the Erlang distribution

For each female agent, the  **FlightAmbition** parameter is set by means of a sample from an Erlang distribution. The Erlang distribution is a special case of the well-known gamma distribution where the curve shape typically consists of a peak which rises initially and then levels out as distance from the origin increases. The severity and location of this peak is determined by the ‘rate’ parameter (λ) of the distribution. In turn, when selecting the rate of the Erlang distribution used to determine the  **FlightAmbition** value for the female agents in the simulation, the value chosen by the user effects the number of so called ‘far fliers’ within the female population.

Far fliers, in this simulation, are females who have a flight ambition in excess of 50m. The distance over which female moths are willing to fly has the most pronounced effect during oviposition as both algorithms make use of this parameter when making decisions on how the eggs will be spread within the simulated space. Generally speaking, it was observed that the dispersal rate of the population depends largely on the number of far flying females present in that population. This phenomenon is, however, also dependent on the type of oviposition algorithm selected in the simulation runs and, for this reason, is investigated in conjunction with the respective oviposition site selection algorithms later in this chapter.

6.4 The influence of the probability of scanning and displaying of agents

An assumption is made in the simulation model that not all males and females in a real-life population of *E. saccharina* would attempt to mate all of the time. This is modelled by a probability threshold imposed on the agents each time night falls and the conditions are suitable for mating to ensue. In the case of the female, each time the  **LekSearch** function is called, the female only successfully performs a search for *leks* in a nearby vicinity if a random sample from a uniform distribution exceeds the user-defined probability threshold for searching. Similarly, a male will only attempt to join a *lek* if the user-defined probability of displaying is exceeded and,

in turn, the male enters a state where the possibility of *lek* contribution exists.

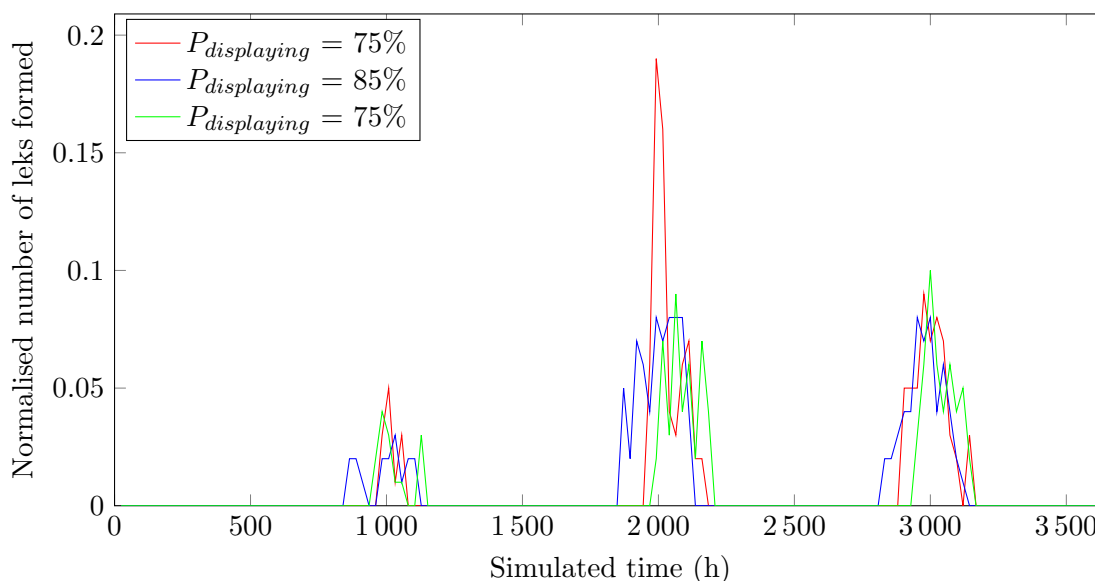
In both cases, very low probabilities of displaying and searching have the potential to throttle a population and cause it to die out owing to a lack of subsequent mating and oviposition. It is, however, anticipated that such low values are unrealistic in nature. Fluctuations around the assumed value of 75% have little to no effect when the population is already thriving and large in number because of the numerous available agents to form or find *leks* and propagate mating. In smaller, more spread out populations where there are fewer agents available in proximity to mate, however, the probability thresholds may have the potential to affect the growth of a population through restricting the number of successful mating instances observed. This may be of importance in the development or design of control measures which affect the scanning or displaying process of female and male moths respectively because, if either process can be sufficiently inhibited or disrupted, it may allow for effective combating of smaller populations. This may potentially be in the form of a counter pheromone scent released within a population which scrambles the pheromone profile created by the male moths during mating, making it more difficult for females to perceive *leks* within close proximity.

In order to test the effects of small fluctuations of the probabilities of displaying for males and searching for *leks* for females, a series of experiments were conducted with different combinations of these probabilities. The number of adult agents was recorded each evening, as well as the number of *leks* created and the number of *leks* successfully discovered and made use of by females that evening. The number of *leks* was then ‘normalised’ by dividing the tally by the number of adult males present in the simulation on that evening in order to compare consequent simulation runs. This was done since a small number of *leks* may also be attributed to a small number of adult males present in the simulation and not necessarily the fact that these males are not displaying. By normalising the value, it may be considered as a ratio of *leks* formed to adult males existing in the simulation. The number of *leks* used was also calculated as a percentage of the available *leks* on a given night. Three experiments for each case were conducted and the result was taken as an average of the recorded cases.

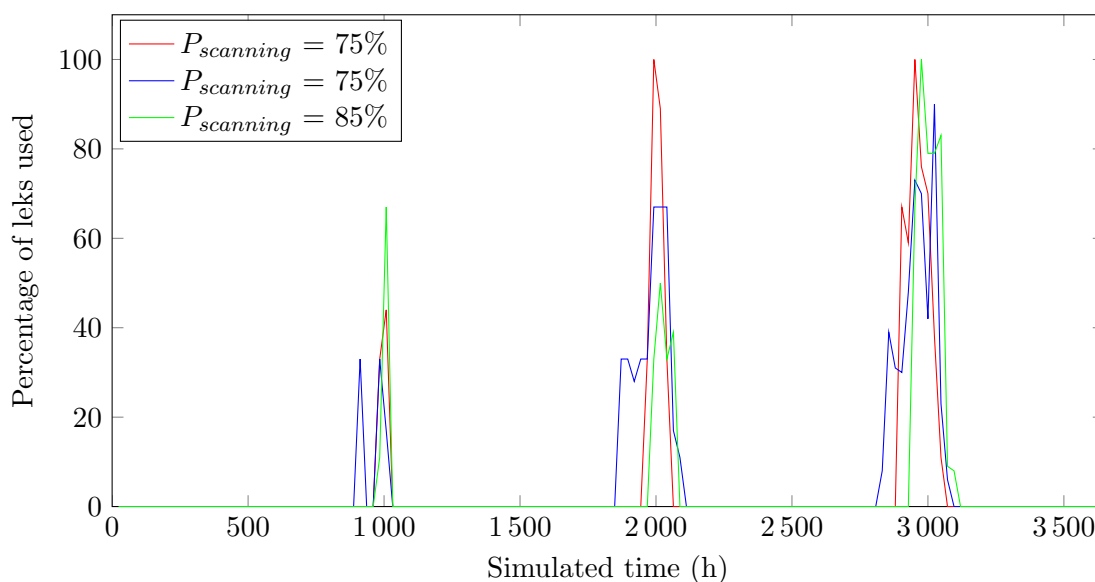
Owing to the fact that the number of male and female adult agents present in the simulation remains relatively similar at any given time, it follows that, in cases where many *leks* are formed owing to an abundance of males, an equal abundance of females will exist to potentially discover and make use of those *leks*. The opposite situation also holds for times when few *leks* are formed. For this reason, the normalised number of *leks* formed as well as the percentage of *leks* discovered and used must be considered in combination as an indicative data reading for each given time step.

In these experiments, an initial infestation of 0.1 *e*/100s was distributed across 100% of the simulated space. The results for the different combinations tested are shown in Figures 6.17–6.19. In each test case in the experiments, the results in Figures (a) and (b) should be considered in combination since the data in each case is generated from the same experimental run (*i.e.* the red lines in Figure 6.17(a) and 6.17(b) are measured at the same time during the same experiment).

Figures 6.17(a) and 6.17(b) show little distinct influence of a higher probability of displaying or searching. In fact, in some cases, more *leks* were formed and discovered during experiments with lower selected probabilities imposed. What can be seen in Figure 6.17(a), however, is that the band of the normalised number of *leks* formed at 85% begins slightly earlier in each case where the population is maturing to adulthood. This suggests that, even when only a small percentage of the population has successfully matured to the `AdultMale` state, the fact that a greater percentage of males are displaying results in *leks* being formed earlier, whereas in the other cases, *leks* were only formed at a slightly later point in time — probably once more agents



(a) Normalised number of leks formed.

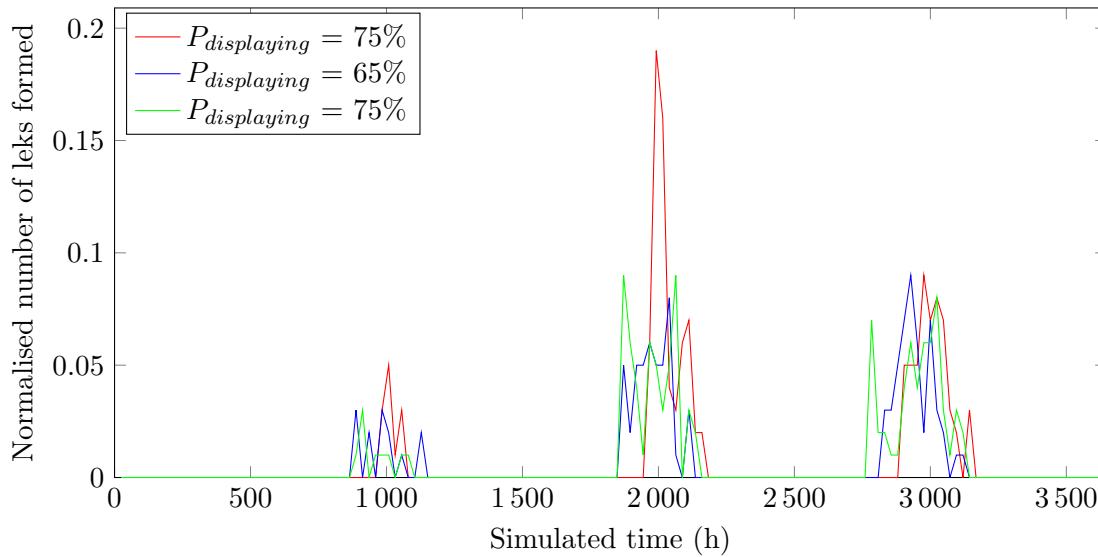


(b) Percentage of leks used.

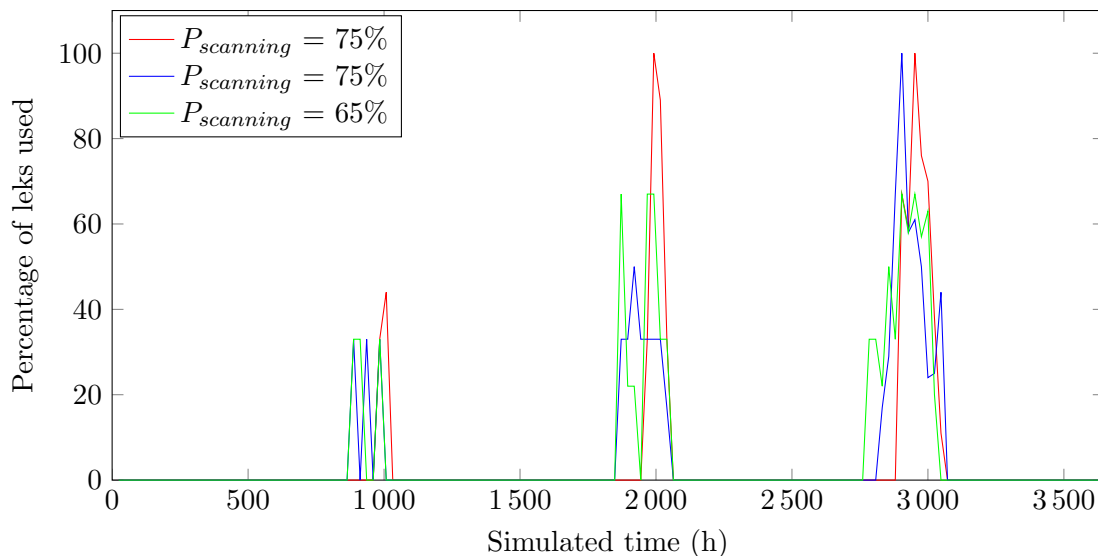
FIGURE 6.17: Lek creation and usage as a result of increased probability of displaying for males or probability of scanning for females.

have matured to adults and were available to contribute to them. This shows that a higher probability of displaying means that fewer adult males are typically required for *leks* to begin becoming available within a population. A similar phenomenon may be seen in Figure 6.17(b), although this case is also dependent on the availability of *leks* within the simulation to be discovered.

In cases where lower probabilities of displaying and searching were employed, Figures 6.18(a) and 6.18(b) once again show little definitive influence within the simulation. Figure 6.18(a) shows that the normalised number of *leks* formed was generally low for all cases and, as such, it appears that a decrease of 10% in the probability of males displaying has little to no effect on the possibility of mating. The percentage of *leks* discovered in Figure 6.18(b) does remain



(a) Normalised number of leks formed.

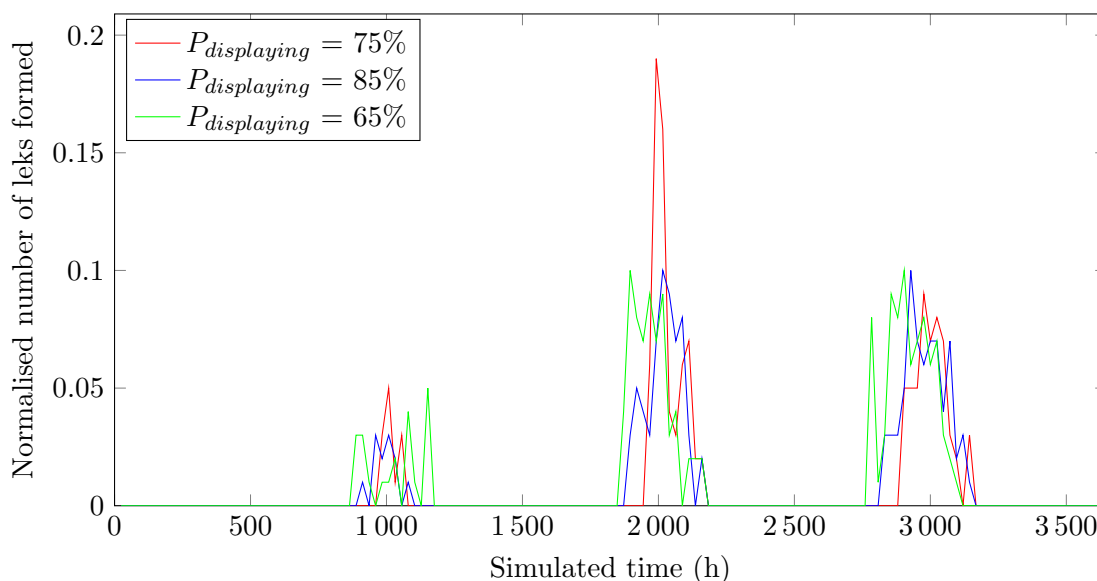


(b) Percentage of leks used.

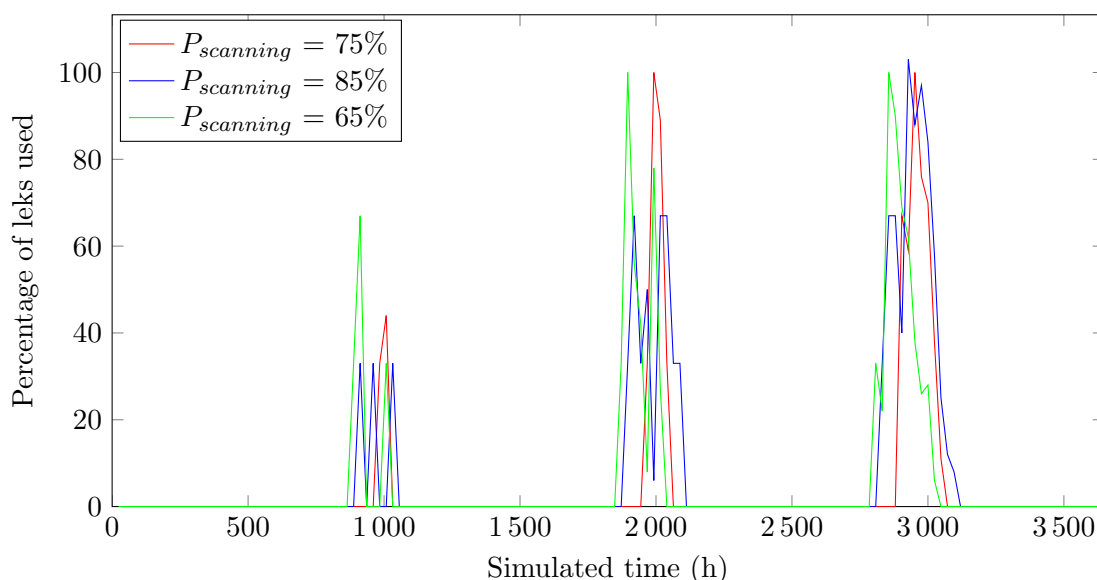
FIGURE 6.18: Lek creation and usage as a result of decreased probability of displaying for males or probability of scanning for females.

consistently low when fewer females are searching, but this may also, once again, be attributed to a lower number of *leks* available for discovery.

In experiments where both probabilities were either simultaneously increased or decreased, no apparent influence is noticeable in Figures 6.19(a) and 6.19(b). This, along with the results of the previous experiments pertaining to displaying and searching probabilities, suggest that the number of *leks* formed and discovered during a simulation is strongly dependent on the agent arrangement within the simulated space. In all cases, the random organisation and position of the agents was a defining factor in the number of *leks* which are formed and discovered during the execution of the simulation, which is more dependent on the ScanRadius and Proximity parameters than on the imposed probabilities. Large changes in the probability of displaying or searching are likely to influence the propagation of the simulation when agent numbers are low



(a) Normalised number of leks formed.



(b) Percentage of leks used.

FIGURE 6.19: Lek creation and usage as a result of a combination of increased or decreased probability of displaying for males and probability of scanning for females.

and infestation is perhaps concentrated within a certain area, but small fluctuations are seen to have little to no effect on the occurrence of mating — particularly as the population increases in size.

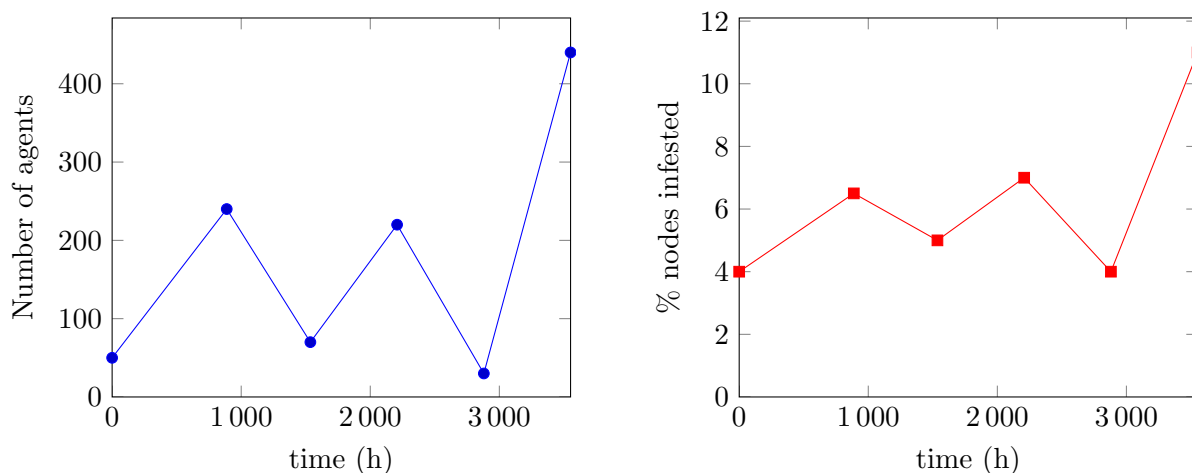
6.5 The influence of larvae dispersal among the sugarcane stalks

In mimicking the process whereby neonatal larvae climb sugarcane stalks and ‘parachute’ from leaves to surrounding plants before burrowing into the stalks, as described in Chapter 2, the distance covered during the dispersal process is modelled by taking a sample from a symmetrical

triangular distribution — the maximum upper bound of which is set by the user. Smaller values of the dispersal distance naturally result in a more densely clumped group of larvae which means more larvae end up burrowing into the same sugarcane stalk. This has the consequence of larger numbers of larvae perishing as a result of the imposition of the carrying capacity of the sugarcane stalks. Only a certain number of larvae can be accommodated by each individual stalk and, in cases where many larvae select the same stalk, fewer agents are promoted to the Pupae state and end up becoming active propagators of the population. As anticipated, the converse is true for larger dispersal distances whereby less mortality is evident as a result of the carrying capacity of the sugarcane.

In the simulation model, up to approximately 80% of the population can perish owing to the carrying capacity of the sugarcane when the dispersal distance is small, but the population will never be completely eradicated by this phenomenon. It simply slows the growth for the first few generations post-harvesting. Once the population reaches a critical mass whereby exponential growth occurs, even very large numbers of carrying capacity-related mortality fail to affect the population explosion and resulting infestation.

In the sporadic data that are available on in-field observation of *E. saccharina*, the phenomenon of carrying capacity appears evident in the population readings which are recorded. One example of such data is shown in Figure 6.20¹. As is the case in this figure, the available data sets often consist of only three to six readings over the course of a large number of months. For this reason, they do not bode well for comparison with the large data sets acquired from the simulation. What is, however, of notable importance, is the manner in which the population fluctuates between successive readings. This can possibly be attributed to the effect of the carrying capacity of the sugarcane.



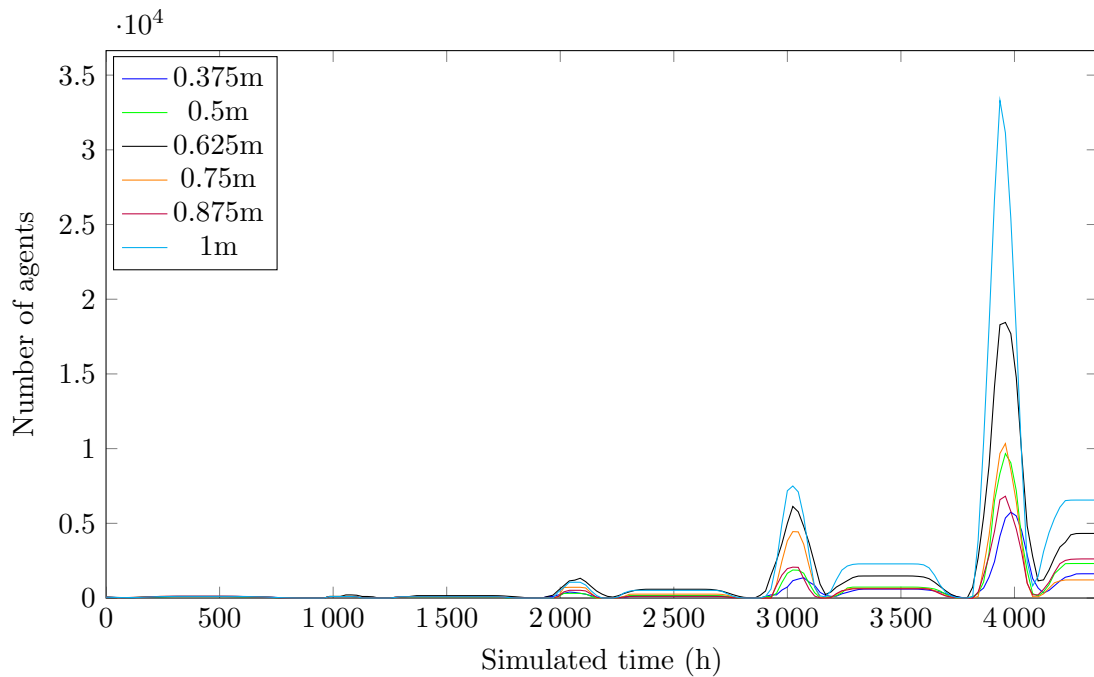
(a) Data set of *E. saccharina* population growth.

(b) Data set of percentage nodes infested.

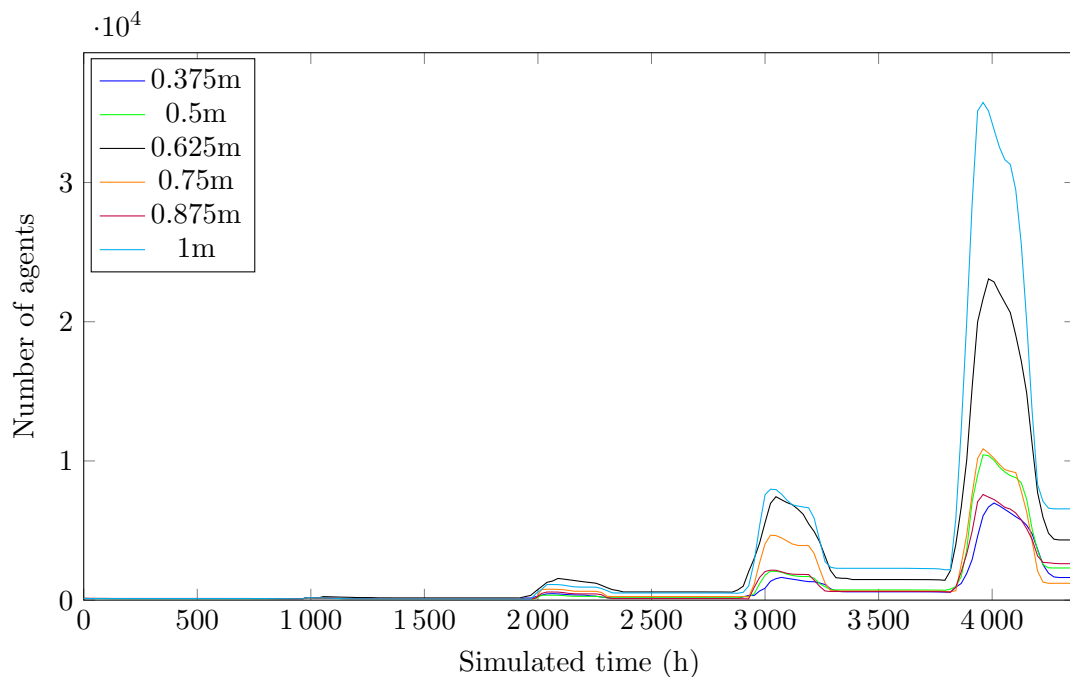
FIGURE 6.20: Extrapolated data set of in-field *E. saccharina* infestation and damage levels.

Experiments were conducted to determine the effect that the assumed maximum distance dispersed by neonatal larvae has on the effect of the carrying capacity and, consequently, population growth. The results of a selected number of these experiments are shown in Figures 6.21(a) and 6.21(b). The initial conditions in these experiments again consisted of an infestation of 0.1 e/100s distributed across 100% of the simulated area.

¹The data set was extrapolated to represent an equivalent 1ha field.



(a) Population growth (Larvae and Eggs).



(b) Population growth (All agents).

FIGURE 6.21: Population growth of agents dispersing differing distances.

As may be seen from the figures, the population typically grows at a greater rate when larvae disperse over greater distances. This is intuitive in that greater dispersal distances means that a larger number of sugarcane stalks are reached by the spreading population and the effect of the carrying capacity on the agents is less pronounced, with more agents being promoted to the Pupae state. When considering the figures in conjunction with the data in Table 6.3, it

$\max_{dispersion}$	$I_{\mu_{total}}$	$I_{\sigma_{total}}$	$I_{\mu_{current}}$	$I_{\sigma_{current}}$	e/100s	max e/100s	Max pop.
0.375	0.020	0.397	0.013	0.286	1.249	4.421	6 977
0.5	0.026	0.475	0.018	0.351	1.777	7.435	10 443
0.625	0.052	0.658	0.033	0.451	3.326	14.188	23 077
0.75	0.018	0.381	0.009	0.227	0.932	7.952	10 869
0.875	0.028	0.446	0.020	0.345	2.014	5.242	7 590
1	0.074	0.775	0.051	0.547	5.044	25.622	35 749

TABLE 6.3: *Damage indicators during testing of differing maximum dispersal distances for neonatal larvae.*

may be seen that the values for the standard deviation of both the current and total infestation levels generally increase with a larger dispersal distance, suggesting a greater spread of resulting agents. The mean infestation levels also increase owing to the larger resulting populations. This is, however, only a generalised observation and some cases, such as that of $\max_{dispersion} = 0.625m$ resulted in a larger, more diversely spread infestation than almost all other cases. This unpredictable behaviour is attributed to the role of random variables in these experiments.

In order to view the resulting damage caused to the sugarcane in each experiment, colour plots were generated to show the infestation state of the sugarcane post the six-month simulated period. The sugarcane length in the simulation begins at $200mm$ and grows to approximately $440mm$ within the 6 months. Each larva is assumed to require nutrients from $40mm$ worth of sugarcane stalk in order to sufficiently mature and progress to the Pupae state. A count was kept of the total number of agents who successfully infested and fed on each sugarcane stalk during each experiment run. Using the final height of the sugarcane and the amount of stalk required per larvae, the total infestation was converted to a percentage of stalk damaged, as indicated by the colour of the area in each figure.

The distance covered by larvae during the ‘parachuting’ dispersal process is likely to be relatively low on average (possibly less than $1m$ [19]). The infestation results for the experiments with lower $\max_{dispersion}$ values are shown in Figures 6.22(a)–6.22(d). As may be seen in these figures, the size of the infested area appears to increase with a corresponding increase in maximum dispersal distance and consequent greater spread of agents within the simulated space. Furthermore, the percentage of stalk damaged also increases, probably owing to the larger population which results from a decreased carrying capacity effect. This effect is not seen in Figures 6.22(d), however, where appears even more localised than in cases employing a lower maximum dispersal distance. This may suggest a possibility that higher maximum dispersal thresholds introduce other complications into the propagation of a population such as the proximity to mates upon reaching the respective adult life stages.

That said, the percentage of stalk damaged in Figure 6.22(d) is very high in the infested area. In such an instance, the initial population breakout most probably resulted in an arrangement of agents which did not promote widespread mating and dispersion and, instead, the population simply remained in the localised area and continued to feed on this sugarcane.

The same analysis was conducted in two experiments which make use of larger $\max_{dispersion}$ values and the results are shown in Figures 6.22(e)–6.22(f). As may be seen from these figures, a similar trend is observed when compared with Figures 6.22(a)–6.22(c), with the infested region and intensity growing relatively proportionally with an increasing maximum dispersal distance. The percentage of stalks infested in areas with the highest damage, however, is significantly higher than in Figures 6.22(a)–6.22(d). This suggests that the differences in dispersal distances

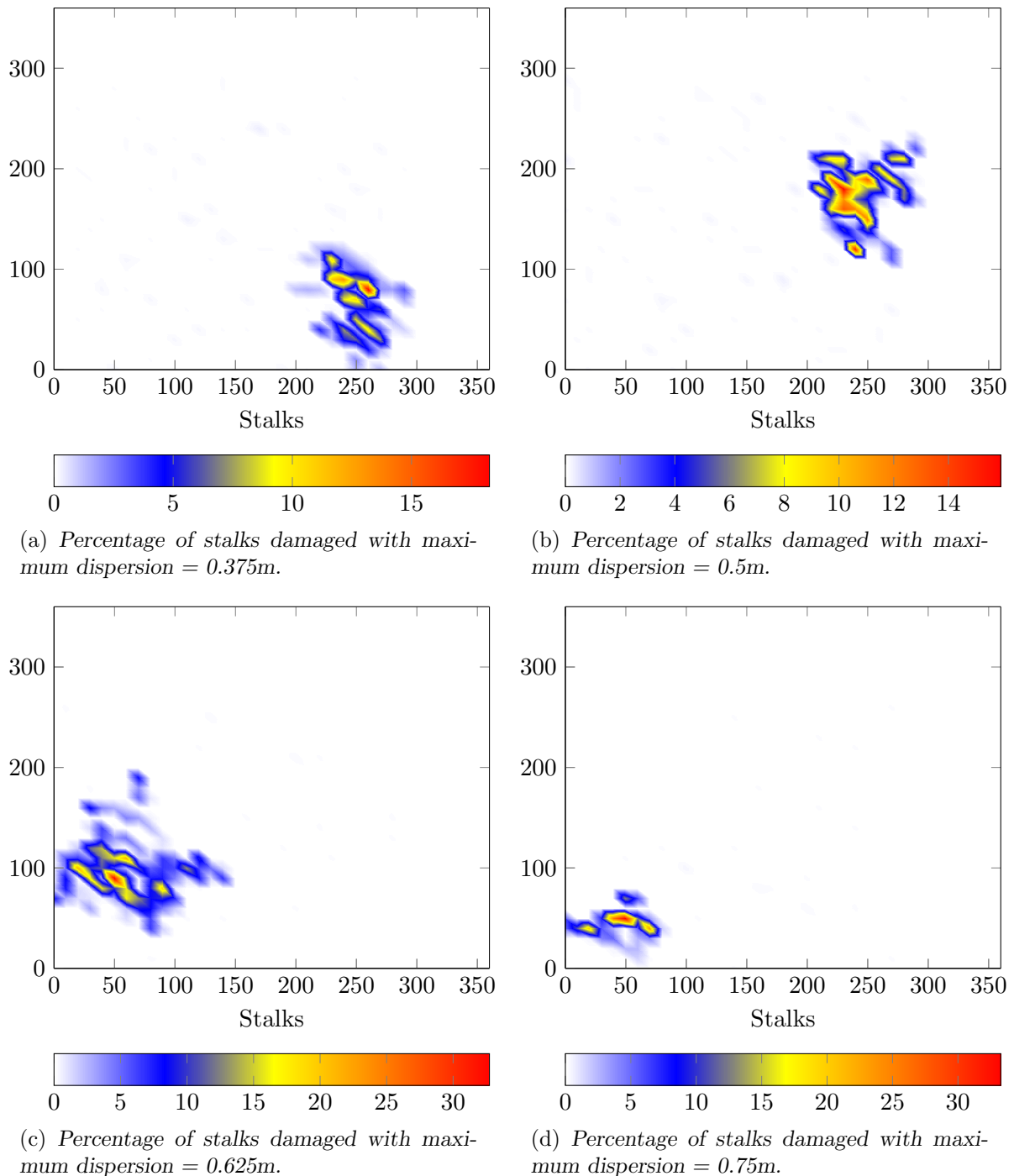


FIGURE 6.22: Selected examples showing notable damage as a result of typical spread of larvae.

in the two sets of figures are perhaps on too small a scale to perceive in these figures, but large enough to effect the resulting infestation levels, owing to a decreased effect of carrying capacity of the sugarcane.

As mentioned, all reported experiments are single cases and would require much more intensive investigation and consequent simulation runs in order to draw conclusive results from emergent behaviour. It can, however, be concluded that the maximum larval dispersal distance plays a notable role in the propagation of the population and the resulting infestation levels within the

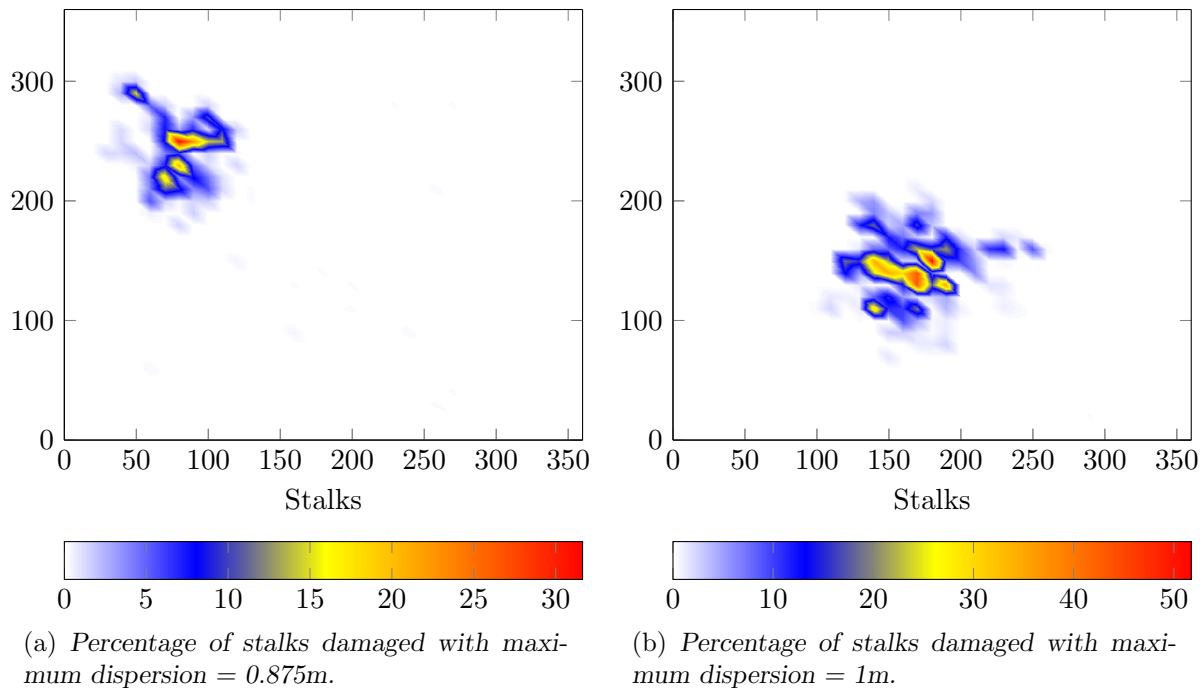


FIGURE 6.23: Selected examples showing significant damage as a result of wider spread of larvae.

sugarcane in the simulated space. The agent-based model presented in this dissertation provides a useful platform for testing of a variety of scenarios, as well as the ability to further investigate the consequences of these differing scenarios.

6.6 The oviposition site selection algorithms

Oviposition site selection plays a critical role in the behaviour and propagation of a population of *E. saccharina* in sugarcane, but is very poorly understood in terms of what process is followed in nature. The two algorithms included for this purpose in §4.6 can affect a large number of different resulting arrangements of the population using the available parameters which compose the functions. Each of the three primary attributes of both oviposition site selection algorithms are discussed in this section.

6.6.1 General dispersal rates

A very important property of the two oviposition site selection algorithms is the manner in which they influence the dispersal of a population. In general, the **F** FindMatureCane function causes a greater spread of agents in the simulation than the **F** MovingWindow function. This is in light of the fact that the **F** FindMatureCane function forces the female to fly the full distance of her **F** FlightAmbition parameter during an oviposition cycle. As mentioned earlier, this spread is intensified when the rate of the Erlang distribution from which the **F** FlightAmbition parameter is sampled is chosen to favour larger values. Adjusting the **F** CaneSearchRadius has a significant influence in the dispersal of the agents toward areas of more mature sugarcane, but can, in conjunction with a varied **F** WindowSize, promote a diverse spread of oviposited eggs within the specified direction of movement.

In the **F** `MovingWindow` function, females aim to fly the entire distance modelled by their **C** `FlightAmbition`, but halt flight once six batches of eggs have been oviposited. This means that if either the **C** `CaneQualityThreshold` is low, or the female happens to have set off in a random direction which leads to high quality sugarcane, suitable oviposition sites are generally found in close proximity to the mating site and the resulting spread of eggs is not as large as in the case of the alternative algorithm. Simply increasing the **C** `PerceptionDistance` can result in subsequent search windows overlapping. The same stalks are consequently seen as potential oviposition sites for a number of windows and are eventually considered multiple times to satisfy the **C** `CaneQualityThreshold` bound. If this occurs, the female may return to the same spot on numerous occasions to oviposit batches of eggs within one oviposition cycle. This phenomenon is intensified when the **C** `CaneQualityThreshold` is high since few alternative locations typically exist which are of sufficient quality. This resulting effect was discussed with Conlong [19] and it was confirmed that, in nature, certain ‘hotspots’ like these are typically observed to which *E. saccharina* females appear to return from time to time to oviposit. For this reason, no constraint was placed on the number of times a female may visit the same oviposition site, other than the **F** `AssessCrowding` function which will deter females from certain oviposition sites when they become too crowded with eggs. For this reason, the user may also set the **C** `WindowIncrements` parameter. This parameter forces a certain distance to be covered by the female between each batch of eggs and is linked to the value of the **C** `FlightAmbition` parameter. Increasing this parameter hence increases the spread of eggs oviposited and can assist in increasing dispersal distance. A possible consequence to this approach, however, is that there may only exist a very small number of suitable oviposition sites in the randomly chosen direction which satisfy the **C** `CaneQualityThreshold` parameter and, if a large distance is forced between each search window, such sites may only be utilised once by the female, meaning that she may come to the end of her flight distance without having oviposited all of the batches of eggs and thereby resort to random oviposition. This is not a favourable outcome because the sites chosen in this instance have no modelled influence and therefore do not assist in logical, structured simulation development. For this reason, increasing the **C** `WindowIncrements` parameter is therefore not a recommended means of increasing agent dispersal. It is rather suggested that the Erlang rate parameter be increased which, in turn, results in a typically higher **C** `FlightAmbition` parameter from which the **C** `WindowIncrements` are calculated.

The difference in dispersion between the two algorithms is also attributed to the movement about the direction in which the female is instructed to move. In the case of the **F** `FindMatureCane` function, the algorithm allows for more diverse movement (within a window bound falling in a range specified by the user) and results in a wider spread of batches of eggs by each female. Conversely, the **F** `MovingWindow` function typically results in batches of eggs being oviposited in close proximity to the direction line in which the female departs. This, of course, depends on the size of the **C** `PerceptionDistance` window in which the searches are performed, but, in general, the spread of batches of eggs is far more consistent and linear than the **F** `FindMatureCane` algorithm.

These differences in dispersal also effect population growth since a more diverse, wider spread of eggs results in less competition between larvae for sugarcane stalks to feed on and, in turn, a less significant carrying capacity effect. This accommodates a more exponential growth of the population and can lead to intensely computationally expensive simulation runs. The more consistent, narrow spread of oviposited eggs resulting from the **F** `MovingWindow` function causes more competition between burrowing larvae and, as a result, a greater influence of carrying capacity is observed. For this reason, the population tends to grow at a slower rate, particularly

when infestation levels are still relatively low and there are fewer agents to propagate mating.

The influence of the aforementioned parameters and their resulting spread within sugarcane fields of varying maturity was tested in a series of simulation runs. The first set of experiments aimed to illustrate the degree to which varying the internal parameters of each oviposition site selection algorithm effected the dispersion of *E. saccharina* within the simulated space. The experiments were run for six months and the population growth was tracked throughout this period. Furthermore, four dispersion ‘snapshots’ were taken at fixed times during the simulation runs to assist in visualising the resulting population spread. All experimental runs were populated with 30 agents (15 male and 15 female) randomly located within a $10m \times 10m$ square in the centre of the simulated space. This initial condition was chosen so as to illustrate the spread of the population from the centre of the simulation space outwards, as well as to avoid initial population explosions resulting from a large number of initial agents and reduce subsequent computational expense.

The **F** MovingWindow function was assessed first by varying the **P** PerceptionDistance and **C** CaneQualityThreshold parameters between low, medium and high values (the ‘medium’ parameters represent the ‘best guess’ values) and analysing the differing effects on population propagation and dispersal.

The growth of the entire simulated population over the course of the experiment is tracked in Figure 6.24.

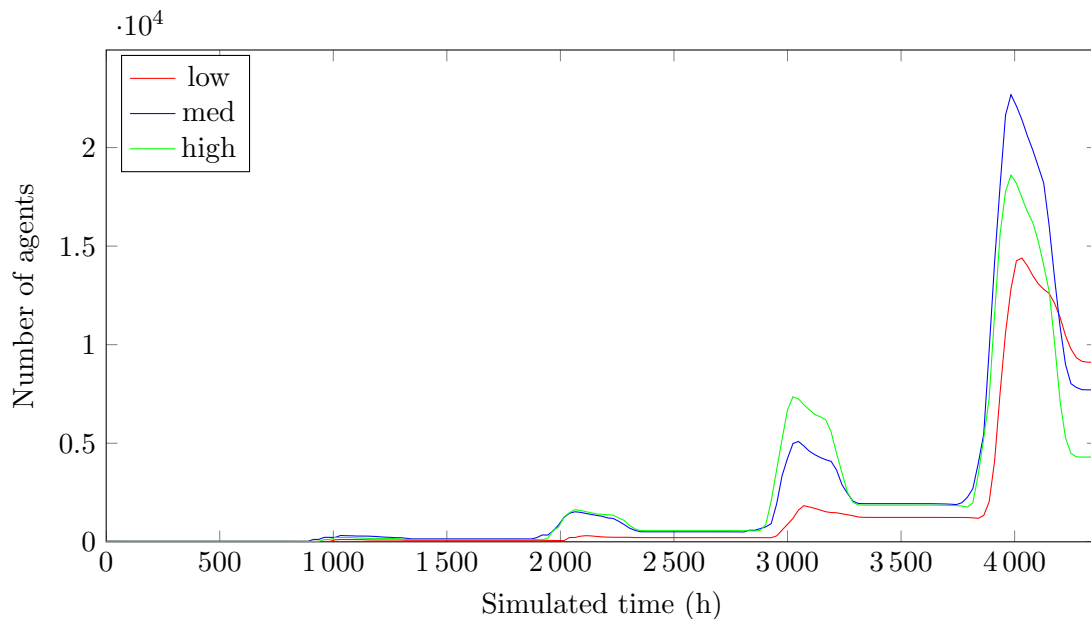


FIGURE 6.24: The growth of the agent population using the low, medium and high values for the parameters used in the implementation of the ‘Moving Window’ oviposition site selection algorithm.

As may be seen in Figure 6.24, the population appears to grow at the largest rate when using the ‘medium’ values for the **P** PerceptionDistance and **C** CaneQualityThreshold parameters. The decreased population growth in the case of the low parameter values is attributed to the intensified effect of the carrying capacity of the sugarcane. Since agents do not spread amongst the sugarcane as much using these parameters, more larvae fail to be promoted to the Pupae state and, as such, the population growth is inhibited. In the case of large parameter values, the population becomes relatively widespread in a short space of time. This means that the effect

of the carrying capacity of the sugarcane is typically less in this case, but has the adverse effect of minimising instances where the mating conditions can be met. Furthermore, since agents are more spread, fewer *leks* are formed and fewer females are within range of discovering these *leks*, resulting in a slower growth of the population. The ‘best guess’ case strikes a balance between these two population regulating factors and, as such, benefits from less influence of either growth inhibitor.

Figures 6.25–6.27 show the dispersion of the pest by means of four snapshots taken at identical times during the respective simulation runs².

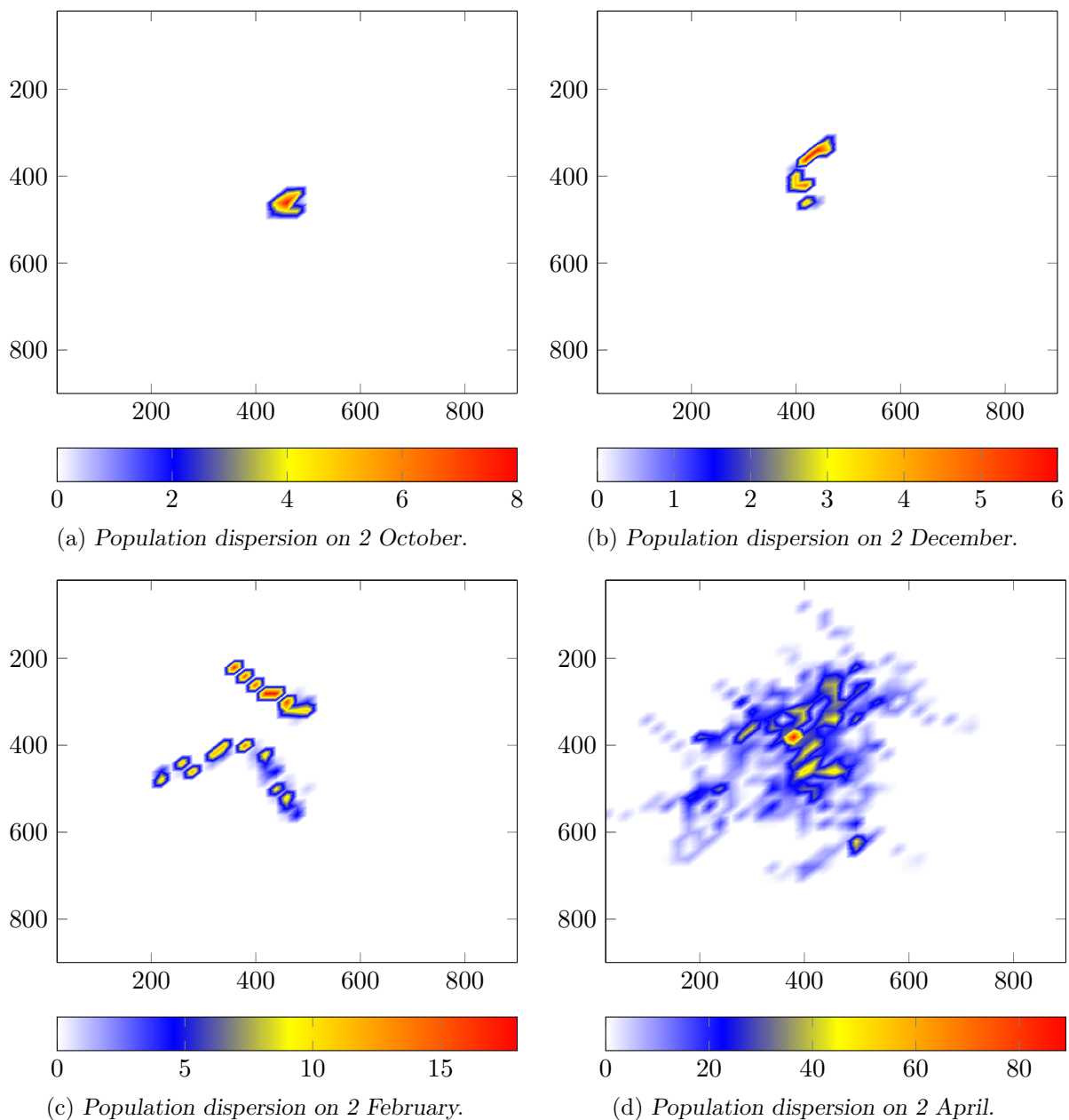


FIGURE 6.25: Population dispersion using the ‘Moving Window’ algorithm with a smaller search window size and lower sugarcane quality threshold.

²All simulations were run over a hypothetical, six-month time period starting on 2 October and concluding on 2 April. Snapshots were taken on 2 October, 2 December, 2 February and 2 April.

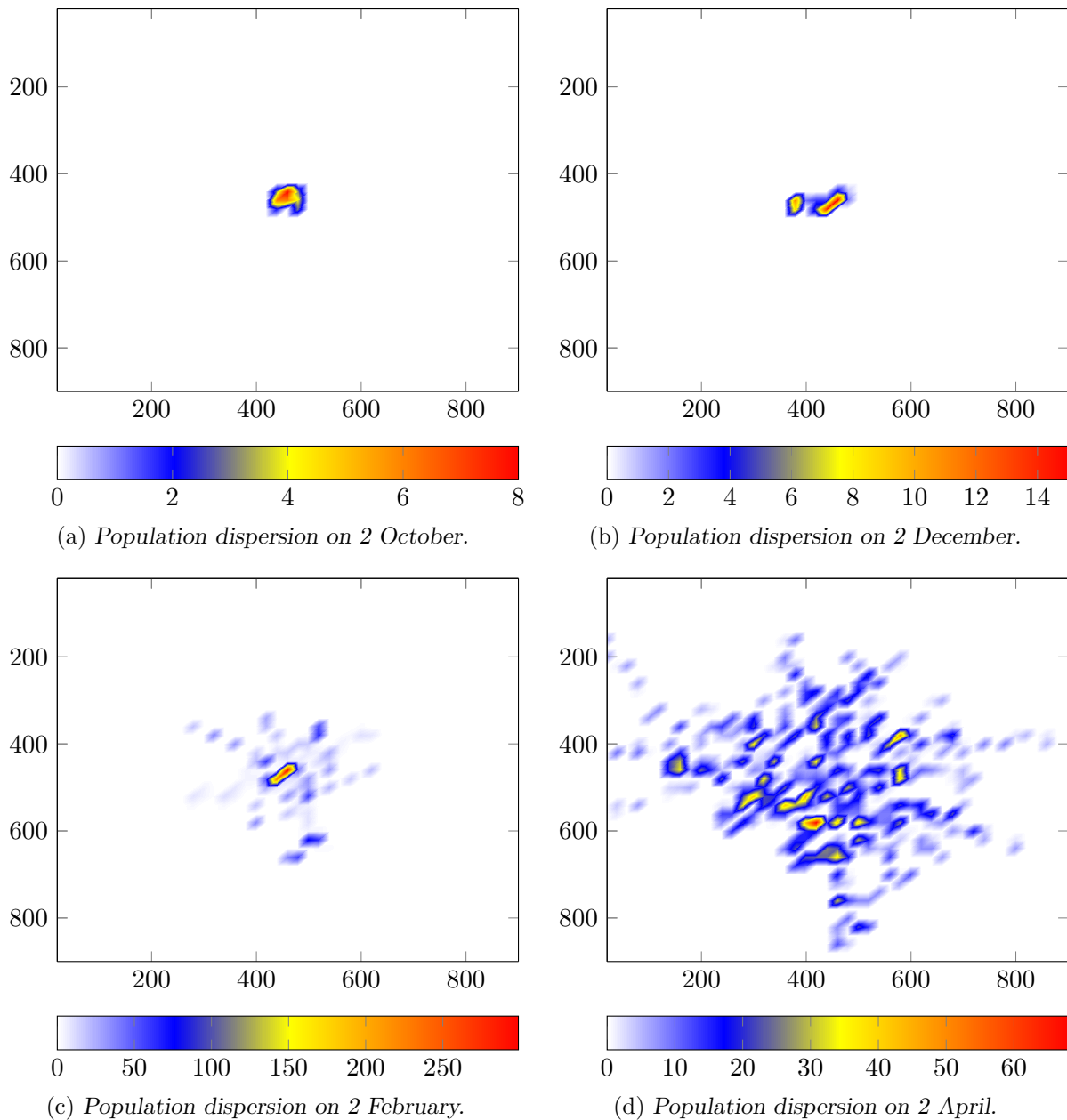


FIGURE 6.26: Population dispersion using the ‘Moving Window’ algorithm with a standard search window size and sugarcane quality threshold.

As may be seen in Figures 6.25–6.27, the population certainly does appear to spread at a faster rate when the $\text{PerceptionDistance}$ and $\text{CaneQualityThreshold}$ parameters increase in magnitude, since the agent distribution in Figures 6.25(d) and 6.26(d) appears somewhat more widespread than Figure 6.24(d). This supports the observations pertaining to the parameters of the MovingWindow oviposition site selection algorithm and illustrates the manner in which the parameters used in the function may be fluctuated to affect different dispersal rates in the simulation.

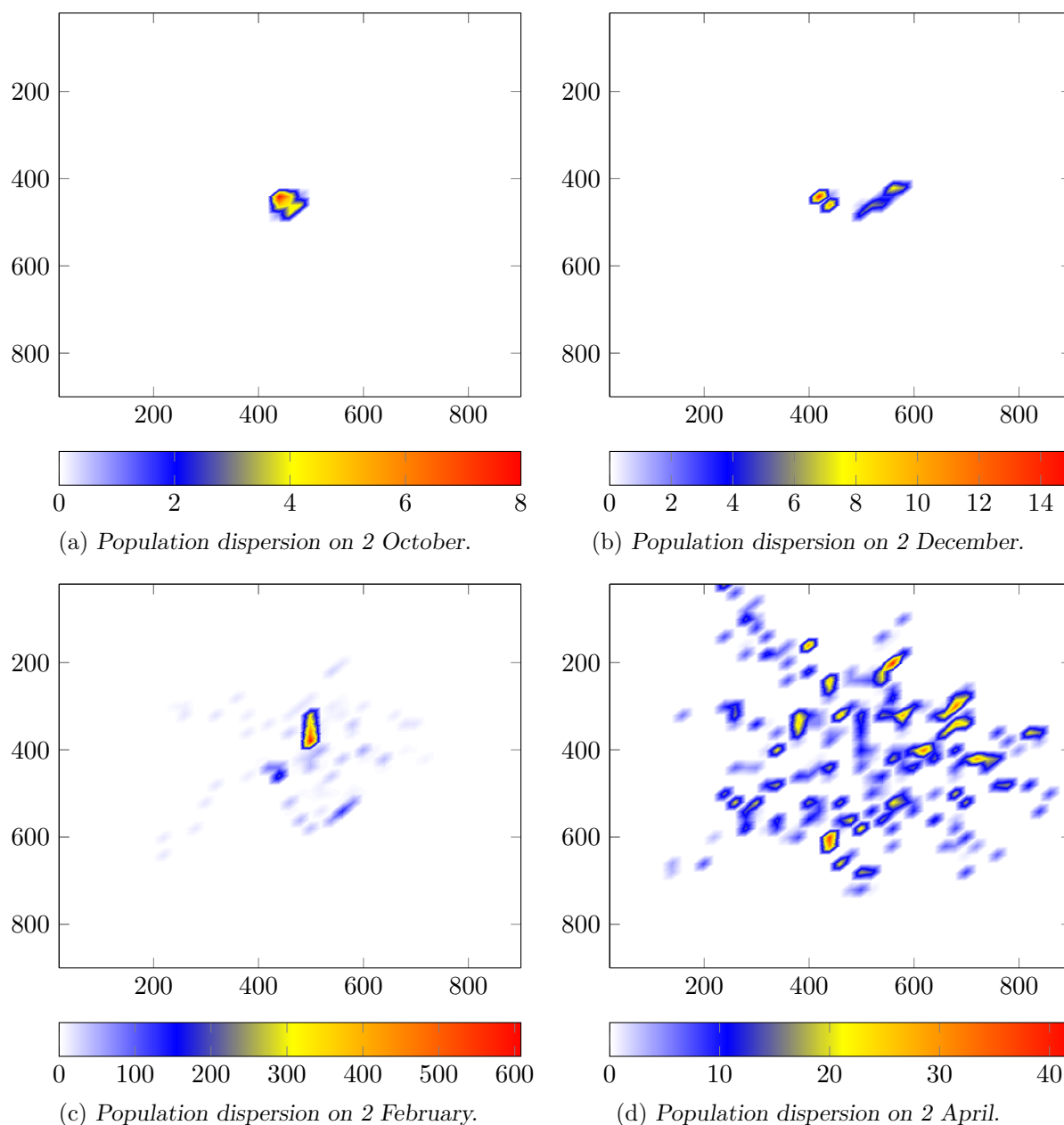


FIGURE 6.27: Population dispersion using the ‘Moving Window’ algorithm with a large search window size and high sugarcane quality threshold.

The same analysis was conducted for the **F** FindMatureCane oviposition site selection algorithm, this time by fluctuating the **C** CaneSearchRadius and **W** WindowSize parameters to determine the effect on population propagation and resulting dispersal. Figure 6.28 illustrates the total population growth over the course of the simulations and Figures 6.29–6.31 represent identically timed snapshots to those in Figures 6.25–6.27, in order to illustrate the accompanying dispersion of the varied parameters using the **F** FindMatureCane algorithm.

In Figure 6.28, a very similar phenomenon to the population tracking for the **F** MovingWindow oviposition site selection algorithm is observed. Here, once again, the largest population is observed for the ‘best guess’ or ‘medium’ choice of parameters. This result comes again as a

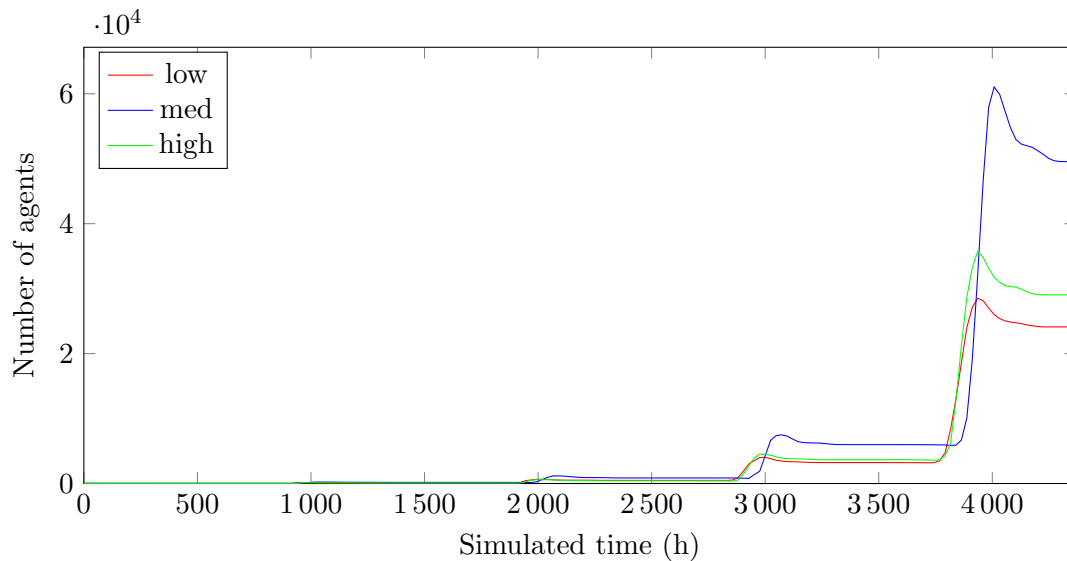


FIGURE 6.28: The growth of the agent population using the low, medium and high values for the parameters used in the implementation of the ‘Find Mature Cane’ oviposition site selection algorithm.

balance between the dispersal of agents at a fast enough rate to reduce the effect of the carrying capacity of the sugarcane, whilst still maintaining close enough proximity to promote mating.

As may be seen in the figure, there is a distinct reduction in the number of agents who perish as a result of the carrying capacity effect when compared with Figure 6.24, resulting in an entirely different shape population curve. This affirms the observation that the **F** FindMatureCane algorithm inherently promotes greater dispersal within the simulated space when compared to the **F** MovingWindow function. The total population numbers also reaches higher values in this case as a result of more agents surviving the promotion between the larvae and pupae life stages and, in turn, going on to mate when they reach the adult life stage.

As expected, the dispersion plots for the **F** FindMatureCane algorithm exhibit considerable spread, as seen in Figures 6.29–6.31. Even in the case where smaller values are selected for the **C** CaneSearchRadius and **C** WindowSize parameters, the resulting dispersion is still relatively vast in comparison to all cases for the **F** MovingWindow algorithm. It appears that the dispersion shown in Figure 6.30(d) is lower than that of Figures 6.28(d) and 6.29(d), but this is actually as a result of a larger number of agents departing from the simulated region owing to such widespread dispersion when the parameter values are increased. These agents are not represented in the figure and, as such, the result shown in the figure tends to be misleading.

The difference in the type of dispersion between the two algorithms is also noticeable. The **F** MovingWindow algorithm generates a somewhat more consistent dispersion, generally tapering off from the centre of the simulated space toward the edges with a few small peaks occurring randomly within the vicinity of the general population. The **F** FindMatureCane, however, promotes a notably more erratic dispersion comprising of large collection of agents in the centre of simulated space surrounded by a number of small pods scattered on the outskirts of the bulk of the population. These scattered pods also extend all the way to the outermost bounds of the simulated space using this algorithm, whereas, in the case of the **F** MovingWindow algorithm, the population did not typically extend to the simulation boundary.

The resulting information pertaining to the parameter variation of both the **F** MovingWindow

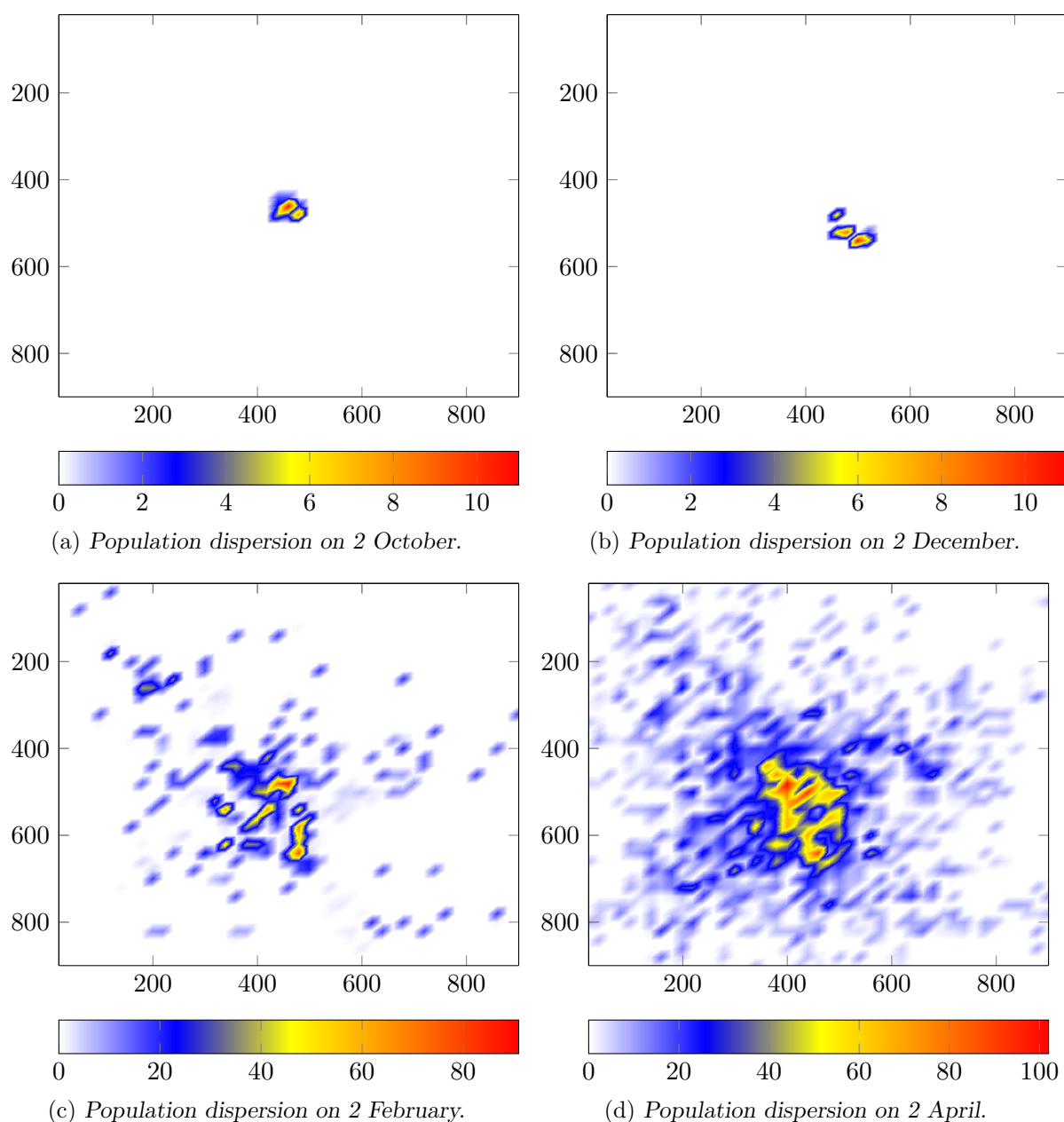


FIGURE 6.29: Population dispersion using the ‘Find Mature Cane’ algorithm with a smaller cane search radius and window size.

and **F** FindMatureCane oviposition site selection algorithms is shown in Table 6.4. The mean infestation level and standard deviation of the infestation is determined in the same manner as in previous experiments. As may be seen, the measures are significantly higher in the **F** FindMatureCane (FMC) function case owing to the wider dispersion and population numbers than in the **F** MovingWindow (MW) algorithm case. It should be reiterated, however, that standard deviation as a measure alone does not necessarily imply uniform dispersal and may be skewed by certain occurrences or outbreaks within the population. For this reason, it is important to consider the measures in conjunction with the corresponding population density plots.

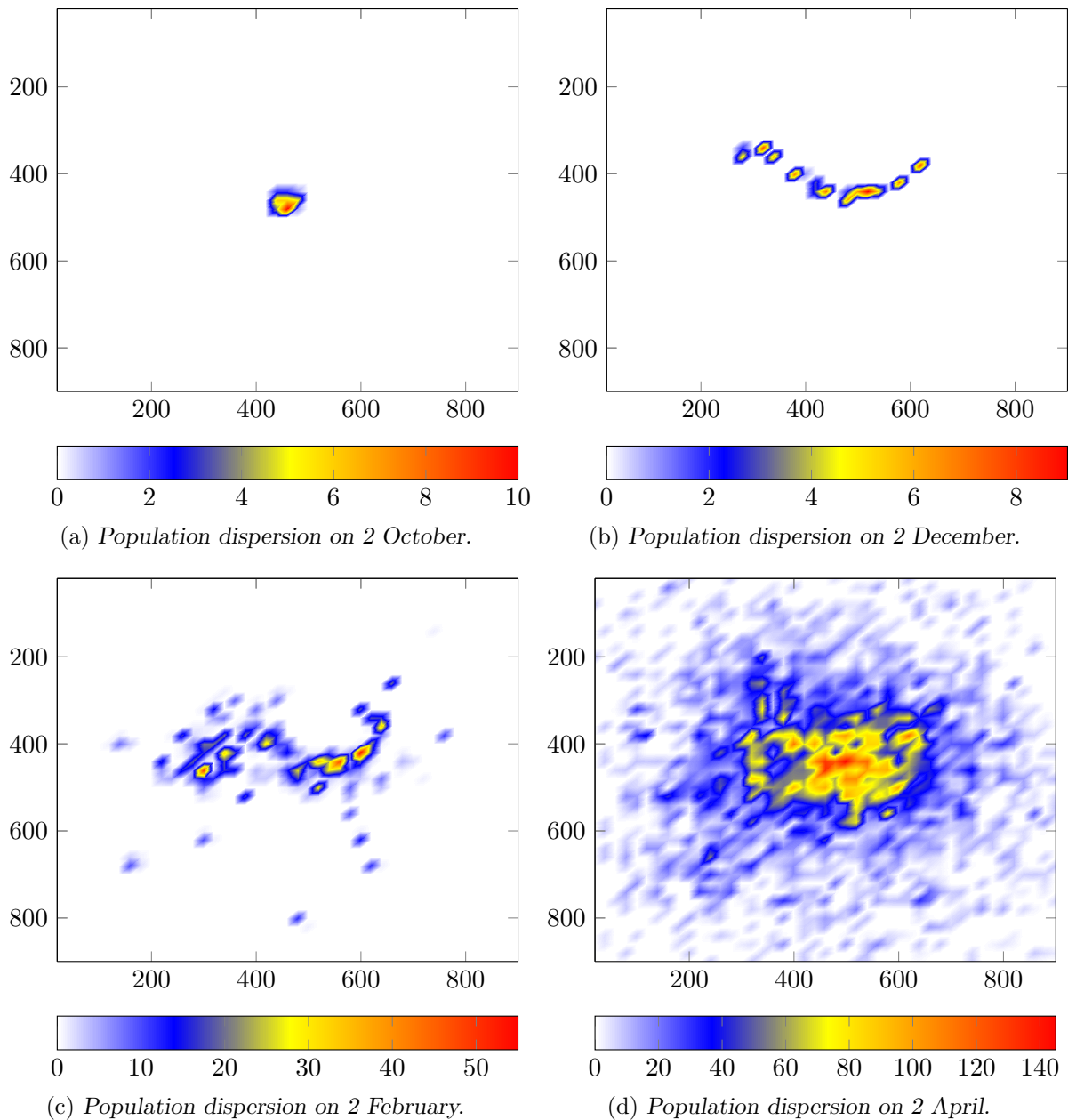


FIGURE 6.30: Population dispersion using the ‘Find Mature Cane’ algorithm with a medium cane search radius and window size.

The inflated damage indicators in the case of the **F** FindMatureCane algorithm are also accompanied by higher $e/100$ s numbers which indicate infestations in excess of 100% more severe than those recorded using the **F** MovingWindow. These data affirm the ability of the simulation model to produce a diversity of dispersal rates and infestation levels using either oviposition site selection algorithm and its accompanying parameters.

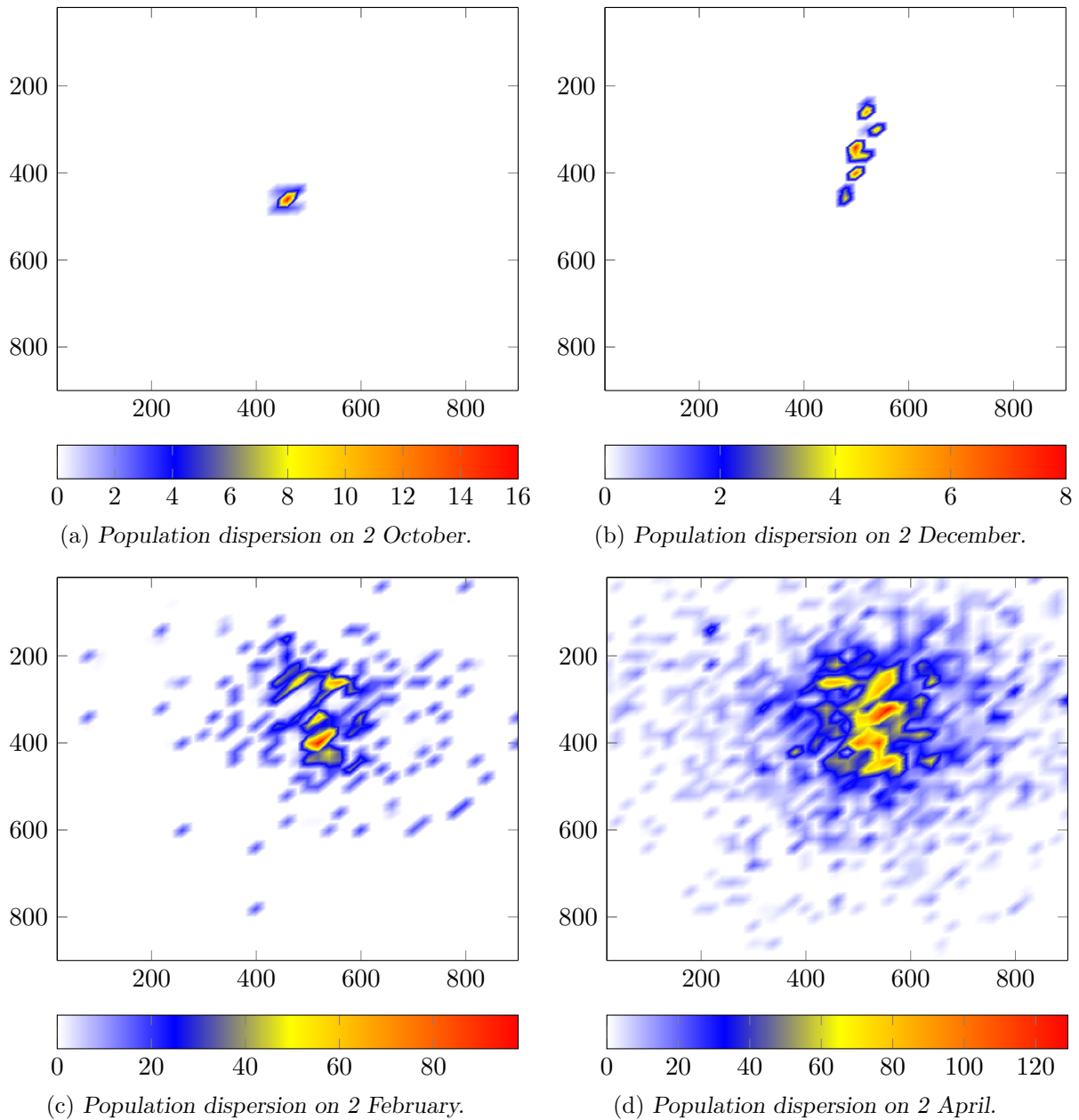


FIGURE 6.31: Population dispersion using the ‘Find Mature Cane’ algorithm with a large cane search radius and window size.

Case	I_{μ_total}	I_{σ_total}	$I_{\mu_current}$	$I_{\sigma_current}$	e/100s	max e/100s	Max pop.
Low MW	0.083	0.748	0.070	0.650	7.003	9.780	14 394
Med MW	0.080	0.807	0.059	0.620	5.929	14.191	22 681
High MW	0.054	0.664	0.033	0.437	3.308	11.692	18 600
Low FMC	0.217	1.044	0.186	0.952	18.545	18.545	28 518
Med FMC	0.441	1.558	0.382	1.416	38.116	40.885	61 049
High FMC	0.259	1.165	0.224	1.059	22.345	23.807	35 786

TABLE 6.4: Damage indicators during testing of differing parameter values for both oviposition site selection algorithms.

6.6.2 The effect of the FlightAmbition parameter

Both oviposition site selection algorithms make use of the FlightAmbition parameter of a female in their execution. Values for each female agent's FlightAmbition are drawn from an Erlang distribution which has a 'rate' parameter (λ) specified by the user. By increasing this rate, more females are likely to have high FlightAmbition parameters which, in turn, will affect both of the oviposition algorithms. Increasing the Erlang rate differs from varying the oviposition algorithm parameters in that it does not affect the inner operation of the algorithms, but rather influences the oviposition spread. Experiments were conducted using both oviposition algorithms to determine the effect of an increased or a decreased Erlang rate parameter which, in turn, results in a larger or smaller FlightAmbition parameter of females in the simulation. These results are shown in Figures 6.32–6.36(d). The surface plots for both 'medium cases' are not repeated as they are identical to those in Figures 6.26 and 6.30 from the MovingWindow and FindMatureCane algorithms, respectively.

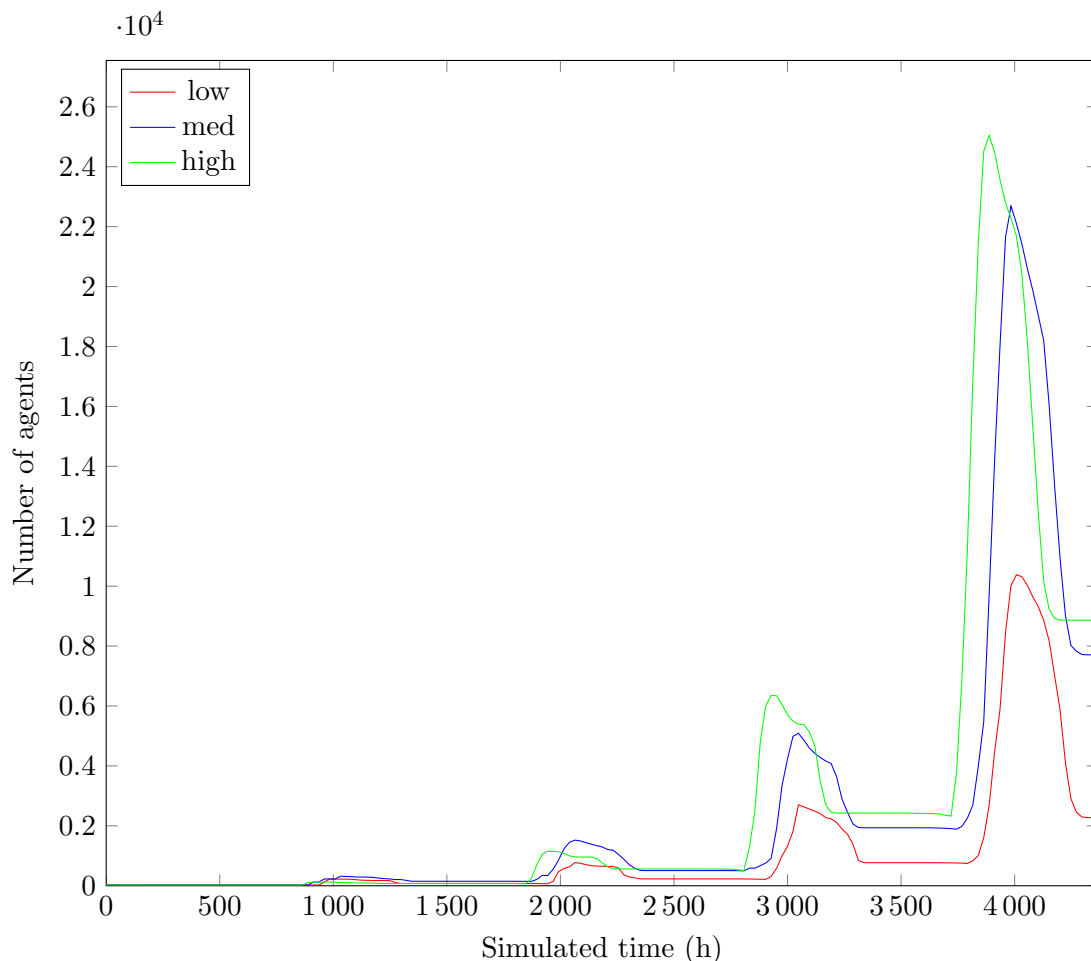


FIGURE 6.32: Population growth for differing Erlang rate parameters used in the execution of the 'Moving Window' oviposition site selection algorithm.

As may be seen in Figure 6.32, increasing the Erlang parameter has resulted in a decreased effect of the carrying capacity of sugarcane and, as such, the 'high' instance results in the largest population growth. Since the MovingWindow algorithm results in dispersal at a slower rate,

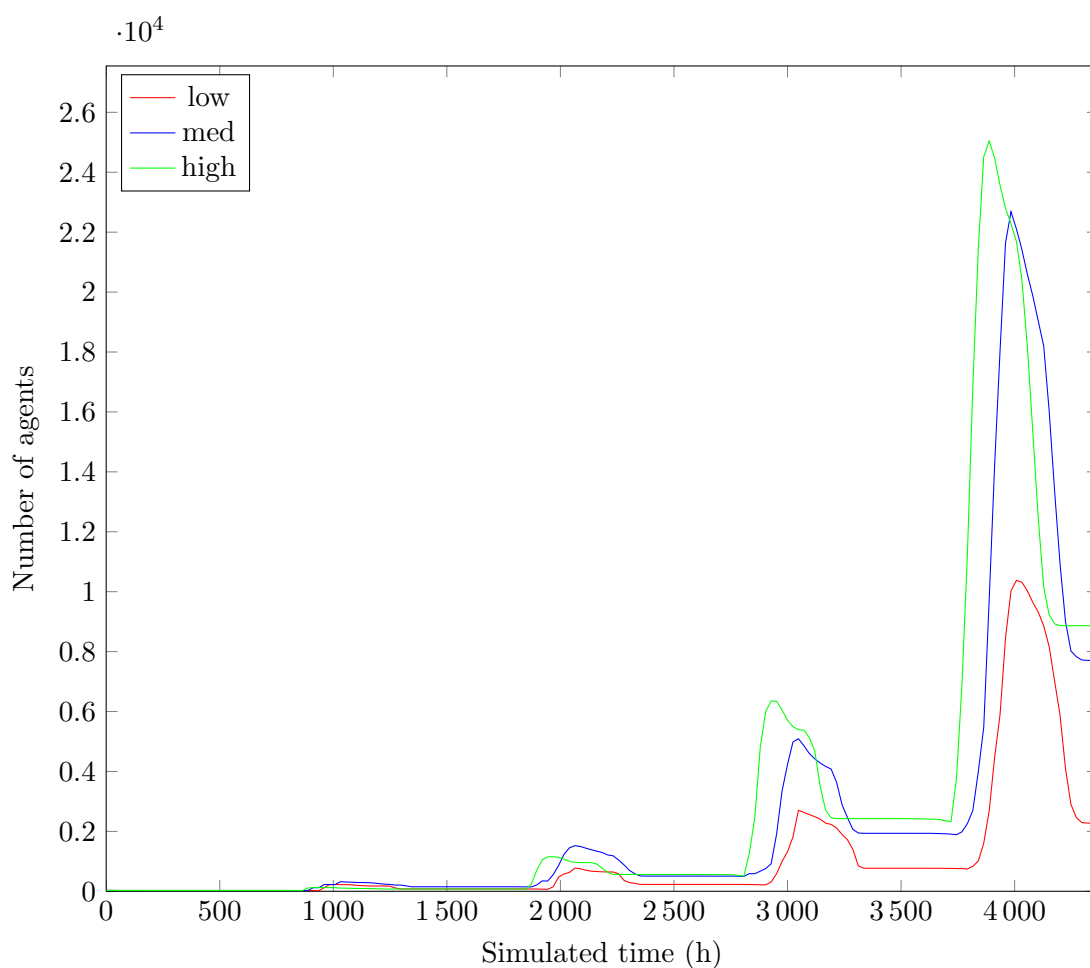


FIGURE 6.33: Population growth for differing Erlang rate parameters used in the execution of the ‘Find Mature Cane’ oviposition site selection algorithm.

there is less adverse effect of agents becoming out of proximity for mating which results in an increased population growth.

In Figure 6.33, however, a similar trend to that observed in Figure 6.28 is noted, where the population has evidently spread so vastly according to a ‘high’ Erlang rate parameter value that mating occurrences drop and the population propagation slows as a result thereof. The population growth when using a ‘low’ Erlang value may be seen to still be affected by the carrying capacity of the sugarcane.

The population dispersion shown in Figures 6.34 and 6.35 appear to be similar in shape to the results documented in Figures 6.25–6.27, but as predicted, appear to be scaled smaller and larger for a lower and higher Erlang rate parameter, respectively. As may be seen in Figure 6.34(d), the dispersion reaches up to the boundary of the simulation space using the **F** MovingWindow algorithm which was rarely observed to occur by simply varying the internal algorithm parameters. That said, the population still maintains the tapered appearance from the centre of the simulation outwards. This suggests the behaviour of the algorithm to be the same, but the distances covered by the female agents when selecting oviposition sites appear to have increased.

This scaling phenomenon is also very noticeable in Figures 6.36 and 6.37 which make use of

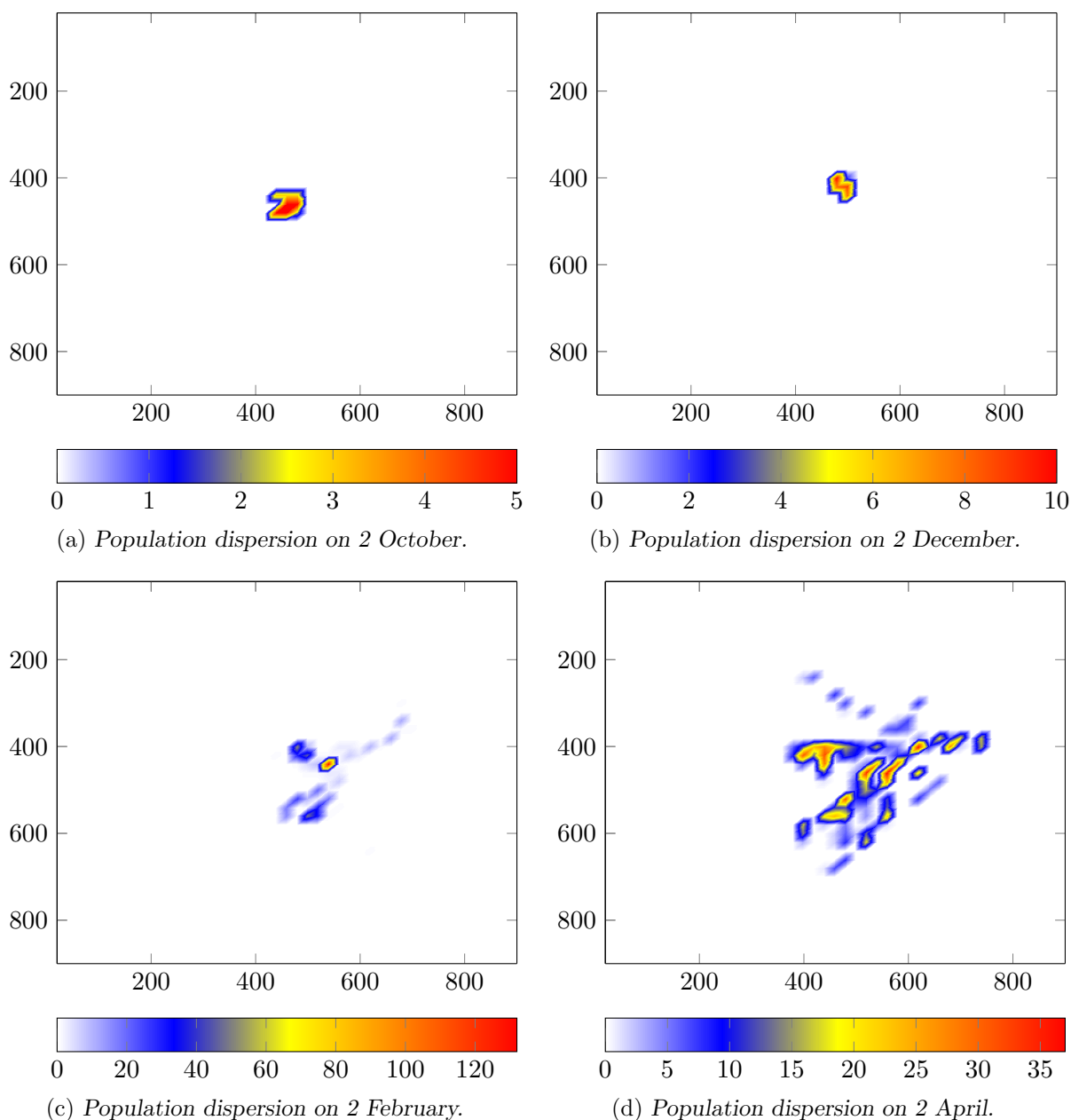


FIGURE 6.34: Population dispersion using the ‘Moving Window’ algorithm with the Erlang value which determines the distribution from which the ‘Flight Ambition’ parameter is lower.

the **F** FindMatureCane algorithm. Figure 6.35(d) shows a significantly clumped population in the centre of the simulated space surrounded by a few clusters of agents dispersed toward the outskirts of the hectare. These results correspond to those of diffusion-type models such as those employed by Potgieter *et al.* [65]. Diffusion models assume average properties and, as such, do not include notions such as ‘far flying’ females, but, rather, show gradual spread over time. In the above-mentioned case of the simulation model, the lower Erlang rate employed results in fewer ‘far flying’ females being generated when sampling values for the **F** FlightAmbition parameters of the females, simulating a process which corresponds to typical diffusion. The small clusters of agents which are evident on the outskirts of the simulation, however, are being caused by the female agents who, despite the decreased probability, were still attributed high

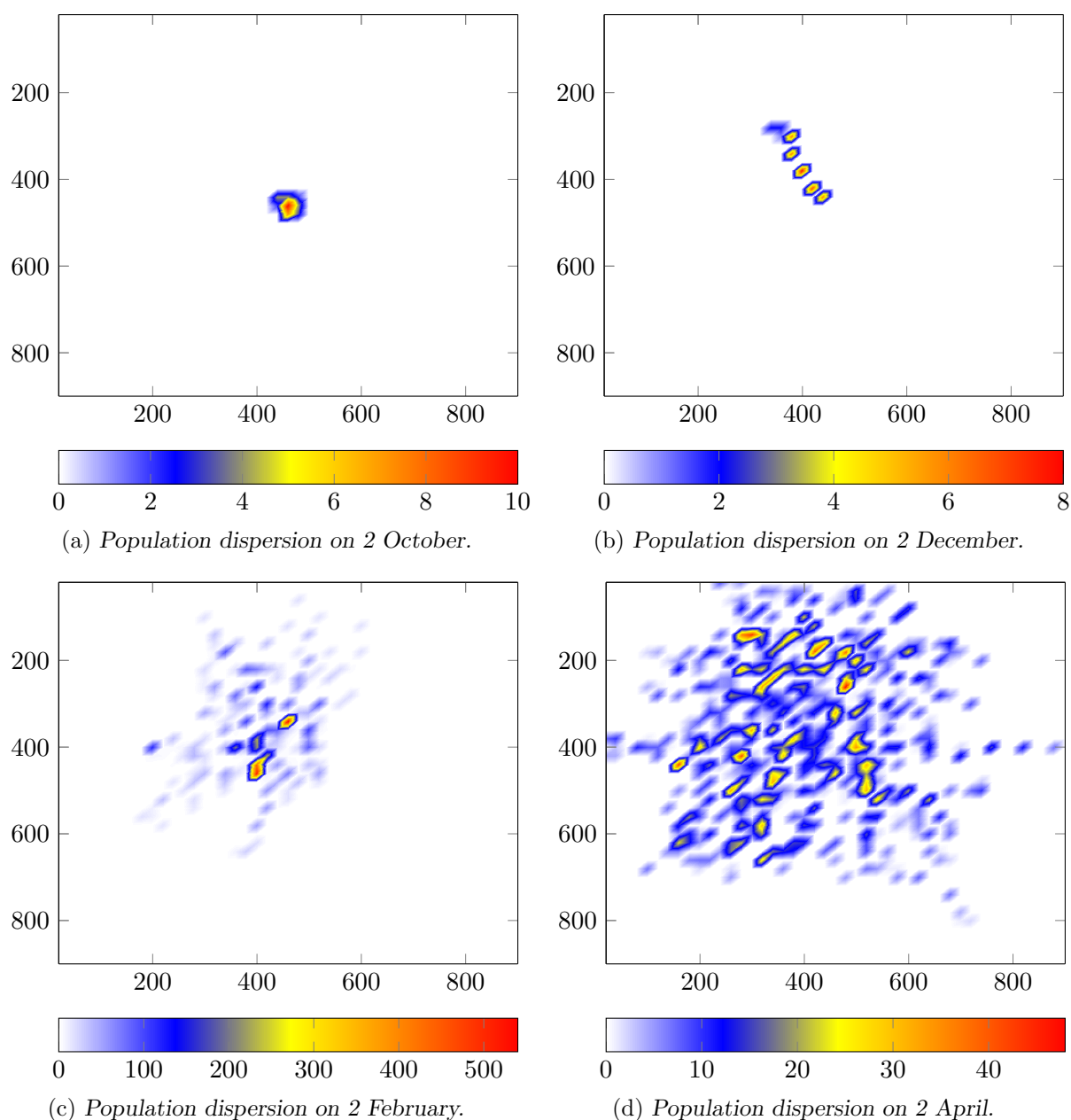


FIGURE 6.35: Population dispersion using the ‘Moving Window’ algorithm with the Erlang value which determines the distribution from which the ‘Flight Ambition’ parameter is higher.

🌀 FlightAmbition values as sampled from the Erlang distribution.

Figure 6.36(d) shows a very diversely spread population where, once again, many agents reside on the boundaries of the simulated space. This suggests that the population spreads at a rapid rate and, since females employing the 🌀 FindMatureCane algorithm are instructed to travel roughly the entire distance of their 🌀 FlightAmbition parameter, the subsequent larger number of far fliers is evident from the flatter, more erratic distribution of the population. This result corresponds more to the in-field observations of Berry *et al.* [13] who reported that, in experiments monitoring *E. saccharina* damage in sugarcane crops, the borders of the trail areas indicated infestation level equal to those of the central areas. This was attributed to either invasion from outside the field, as well as internal spread within the field. Berry also identified

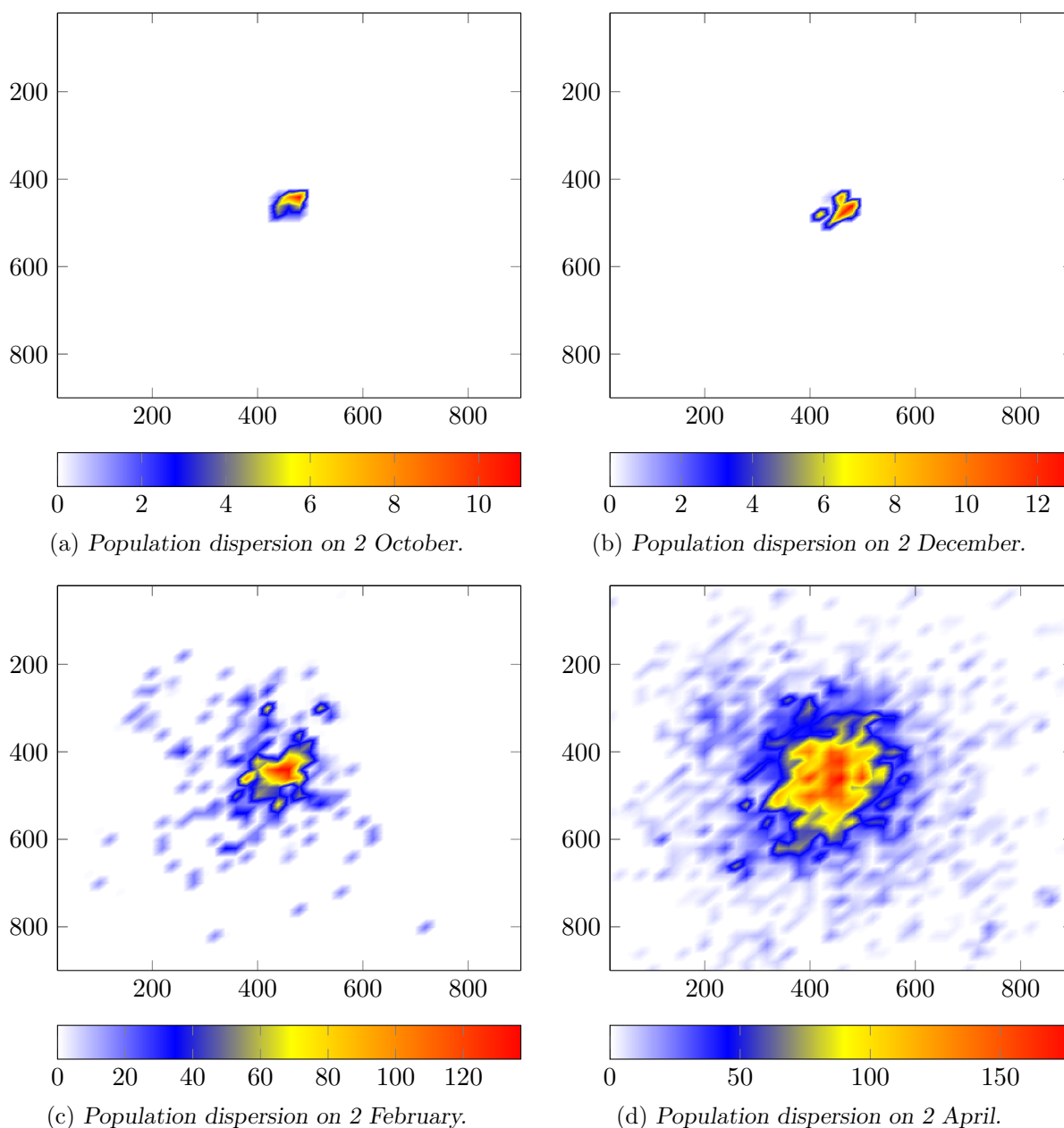


FIGURE 6.36: Population dispersion using the ‘Find Mature Cane’ algorithm with the Erlang value which determines the distribution from which the ‘Flight Ambition’ parameter is lower.

different levels of infestation between adjacent stalks, resulting in a clustered arrangement of high and low areas of infestation across the field, dissimilar to the ‘diffusion-like’ dispersal observed in the case of the **F** MovingWindow algorithm.

The damage indicators for the two different algorithms and their respective Erlang rate parameters are summarised in Table 6.5. As may be seen in the table, and when considered in conjunction with the corresponding figures, these damage indicators and associated infestation levels appear to correlate with the different oviposition algorithms in much the same manner as observed in Table 6.4, with a larger mean and standard deviation of damage resulting from experiments using the **F** FindMatureCane algorithm as opposed to the **F** MovingWindow. This is accompanied by higher $e/100s$ values for these experiments. There is, however, a noticeable

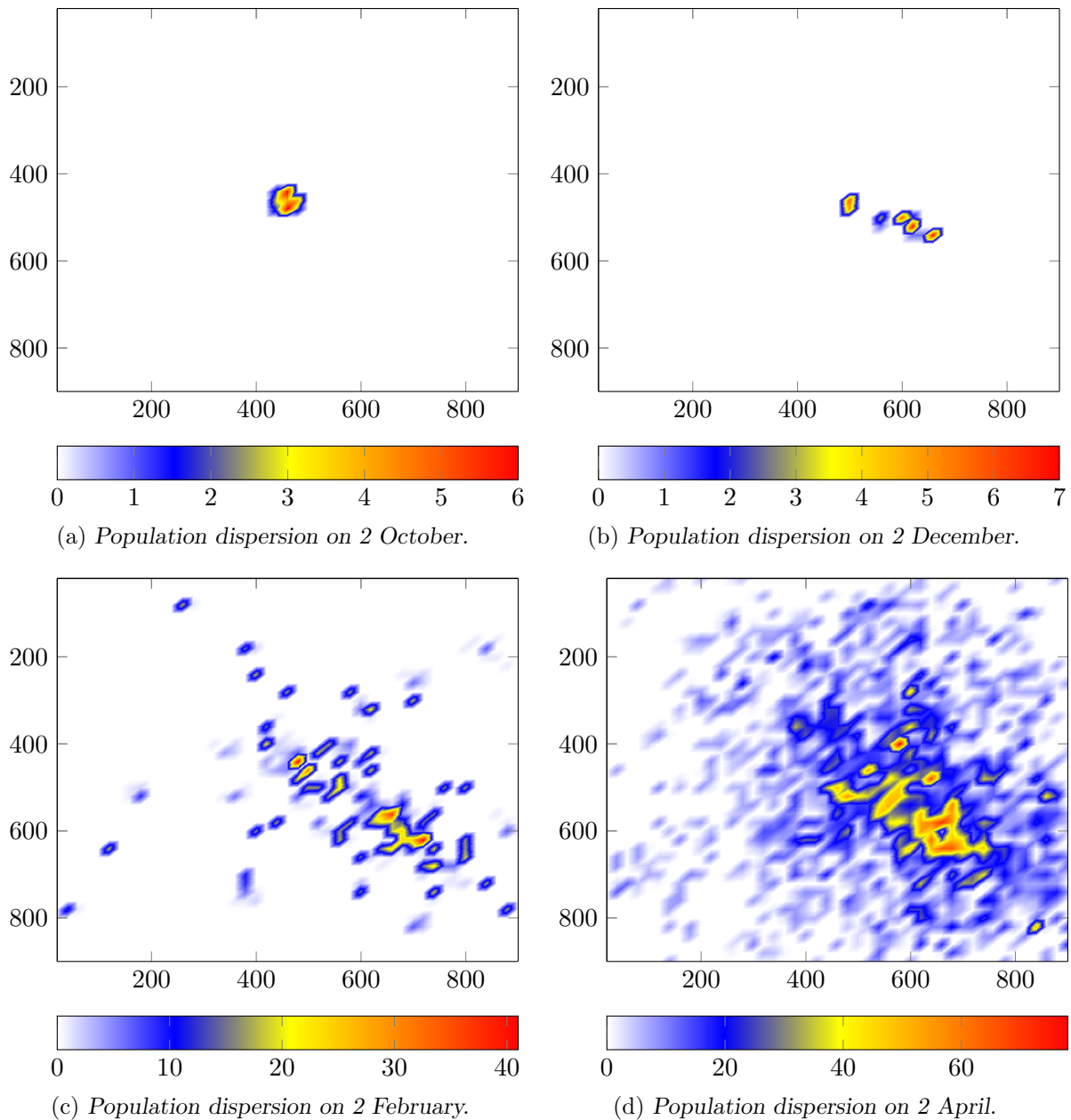


FIGURE 6.37: Population dispersion using the 'Find Mature Cane' algorithm with the Erlang value which determines the distribution from which the 'Flight Ambition' parameter is higher.

decrease in the 'FMC high λ ' case shown in the table. As mentioned, this is most probably as a result of agents departing from the simulated space on a regular basis during this particular experiment, but may also be as a result of the influence of random variables which had an adverse effect on the growth and spread of the population in this test case.

Case	I_{μ_total}	I_{σ_total}	$I_{\mu_current}$	$I_{\sigma_current}$	e/100s	max e/100s	Max pop.
MW low λ	0.026	0.466	0.018	0.327	1.750	6.750	10 379
MW med λ	0.080	0.807	0.059	0.620	5.929	14.191	22 681
MW high λ	0.093	0.831	0.068	0.629	6.818	16.571	25 043
FMC low λ	0.382	1.535	0.336	1.384	33.486	35.551	56 247
FMC med λ	0.441	1.558	0.382	1.416	38.116	40.885	61 049
FMC high λ	0.222	1.034	0.189	0.944	18.838	20.083	29 762




TABLE 6.5: *Damage indicators during testing of differing Erlang rate parameters affecting the ‘Flight Ambition’ values using both oviposition algorithms.*

6.6.3 Biased dispersal based on sugarcane maturity values


Although both oviposition site selection algorithms are similar in the manner in which they incorporate some measure of influence of sugarcane maturity values combined with some measure of random movement, the execution and resulting spread of eggs from the two algorithms is vastly different. Conceptually, the **F** FindMatureCane function is more prone to result in a population spreading towards areas of more mature sugarcane over time. This is attributed to the construction of the algorithm which first determines the most appropriate direction (in terms of sugarcane maturity) in which to move, and then moves within a defined ‘window’ to oviposit eggs in this general direction. Manipulation of the **C** CaneSearchRadius and **W** WindowSize determine the degree to which the female concedes to ovipositing primarily in areas of more mature sugarcane. If the **C** CaneSearchRadius is larger, a greater percentage of females will discover more mature sugarcane pre-oviposition, irrespective of their current location in the simulated space. In addition, if the **W** WindowSize sample comes from a smaller range, the female will stick more closely to the identified direction. Both of these factors influence a more direct instruction to female agents to oviposit in areas of more mature sugarcane and, as the **C** CaneSearchRadius is made smaller and the range from which the **W** WindowSize sample is increased, so there is less chance of discovery of better quality sugarcane and a greater influence of the randomness incorporated into the site selection algorithm. The consequence of this is a more gradual spread of the population toward better quality sugarcane, accompanied by a greater spread of the population within the simulated space, including in areas of less mature sugarcane.


In the case of the **F** MovingWindow algorithm, the female is less likely to move directly toward more mature sugarcane since the initial direction of movement is randomly chosen in the algorithm. This means that there is equal probability of movement away from or towards such better quality areas. Once mobile, the size of the search windows, referred to in the simulation as the **P** PerceptionDistance, in conjunction with the **C** CaneQualityThreshold, intuitively determine the degree to which the oviposition cycle will result in dispersal toward areas of better quality sugarcane. A higher **C** CaneQualityThreshold will result in a more intensive search conducted by the female within each moving window, since finding a sugarcane stalk with a maturity that exceeds the threshold will be more difficult. In cases where the **P** PerceptionDistance is large and perhaps encompasses areas of both mature and immature sugarcane, imposing a more strict **C** CaneQualityThreshold can assist in pushing a female within the search area toward more mature sugarcane. Conversely, in instances where the **C** CaneQualityThreshold is lower, the female generally finds appropriate oviposition areas with ease and adheres quite closely to the original direction of departure. This results in a


spread which is largely attributable to the initial random direction selection.


In the event that the  **CaneQualityThreshold** is made too strict (even in cases where the  **PerceptionDistance** is large), a regularly occurring result is that very few or no instances of sugarcane with such a high quality are discovered during the flight. This results in the female resorting to random oviposition site selection at the outermost boundary of the  **FlightAmbition** distance which favour no specific quality of sugarcane.

A series of further experiments were conducted to determine the degree to which the two oviposition algorithms result in dispersal toward areas of higher sugarcane maturity. Although population dispersion results from many years of growth and spread of *E. saccharina*, the experiments in this case were restricted to 6 months long, simply to provide an indication as to whether the algorithm promotes such type of dispersion or not. This was done because the maturity values of the sugarcane stalks in the simulation are fixed over time instead of increasing as a function of sugarcane height as is the case in nature. For this reason, it would not be accurate to run these experiments for a prolonged period of time, particularly over a harvesting period.

The starting conditions for these experiments were identical to those in the assessment of the resulting dispersion of the oviposition algorithms. The final population dispersions for each of the four possible underlying sugarcane field layouts after six months using the  **MovingWindow** oviposition site selection algorithm are shown in Figures 6.37(a)–6.37(d). Note that, in these plots, the direction of the vertical-axis has been reversed so as to allow the underlying sugarcane arrangement in the figures to match that which is seen when executing the simulation (*i.e.* the mature and immature areas appear in the same orientation).

As may be seen in Figure 6.37(a)–6.37(d), the  **MovingWindow** algorithm does not result in a notable dispersal towards areas of sugarcane with a higher maturity value. Owing to the random initial direction of movement, the only manner in which the female may discover mature sugarcane is if the random direction leads the oviposition process toward a mature area. In Figure 6.37(b), it appears that more of the random direction movement occurred towards the top half of the simulated space, resulting in more of the population residing in that area. As the underlying structure becomes more complex, however, such as in Figures 6.37(c) and 6.37(d), it may clearly be seen that the influence of the maturity of the sugarcane values has little to no effect on where the female oviposits eggs and the subsequent population develops.

In a similar fashion to previous tests, the population numbers generated when using the  **FindMatureCane** are again far greater in the experiments conducted to determine the effect of the underlying sugarcane structure when employing this algorithm. In this case, Figures 6.38(a)–6.38(d) show a distinct dispersal to mature areas of sugarcane in all different layouts. Owing to the brief simulated time period, it is expected that not all agents will reside exclusively in these mature areas, but, even after only six months, the concentration of the population appears to be far higher in areas where the sugarcane is more mature.

In order to gain a more definitive insight into the data of Figures 6.38 and 6.39, the number of agents residing in each sugarcane maturity division (mature or immature) of the simulated space in each case was tallied. The results of these calculations are shown in Table 6.6. The results show that the figures are, to a certain extent, misleading in terms of the actual spread of the population. In the experiments making use of the  **MovingWindow** algorithm, the ‘rows’ and ‘checkered’ cases resulted in nearly a 50-50 split of the population within the different areas. This makes sense since the female has equal probability of moving in any direction from the location at which she begins oviposition and, since in both cases the central starting condition is surrounded by an equal ratio of mature and immature sugarcane, so it follows that the population is likely

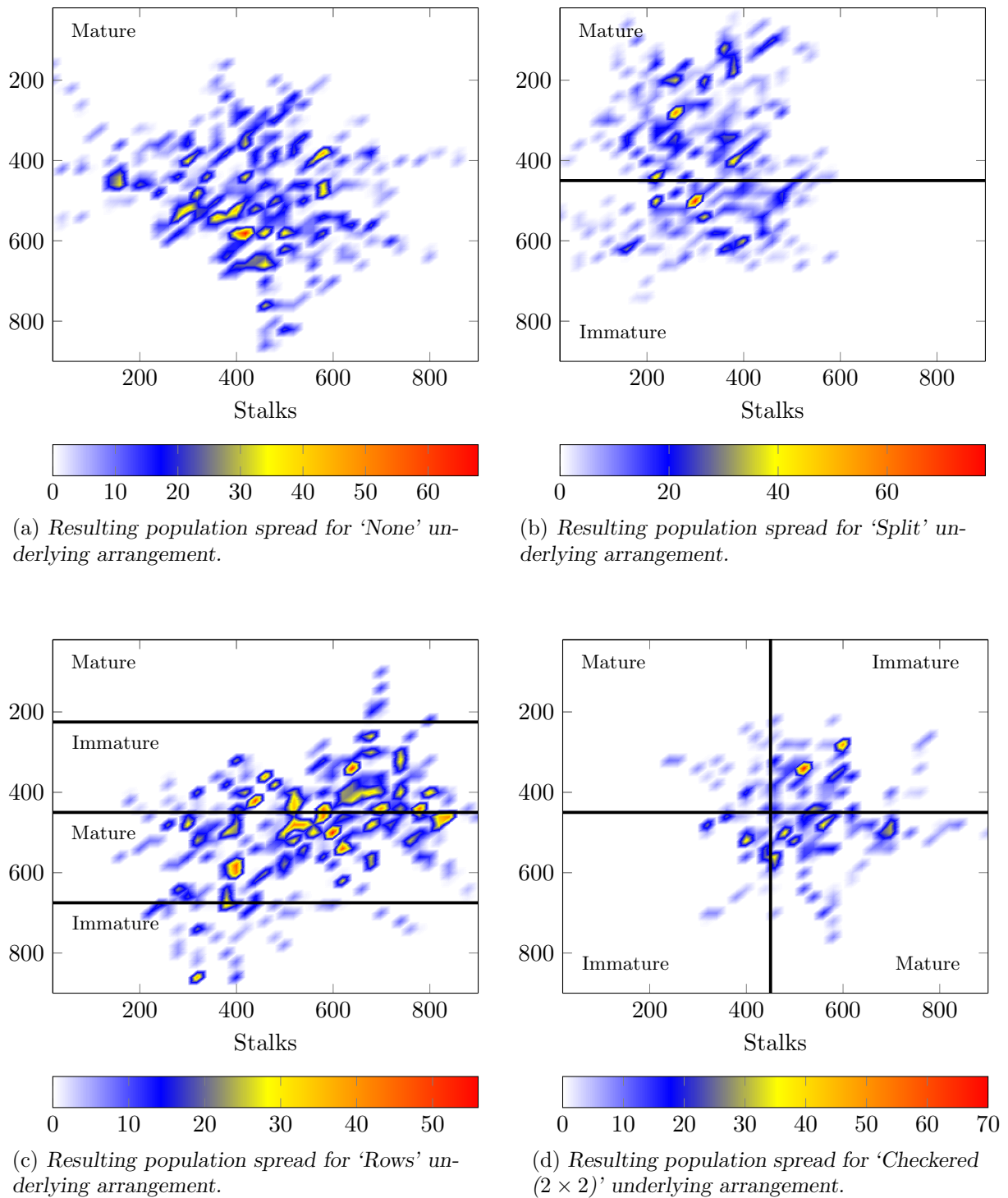


FIGURE 6.38: Population dispersion after six months for different underlying sugarcane maturity values using the 'Moving Window' oviposition site selection algorithm.

to propagate at equal rates within the differing areas. In the 'split' case, more agents were observed to reside in the mature sugarcane area. This is specific to the particular simulation run being reported on and is probably due to the first number of oviposition instances being attributed with a direction which led the female to the mature sugarcane areas.

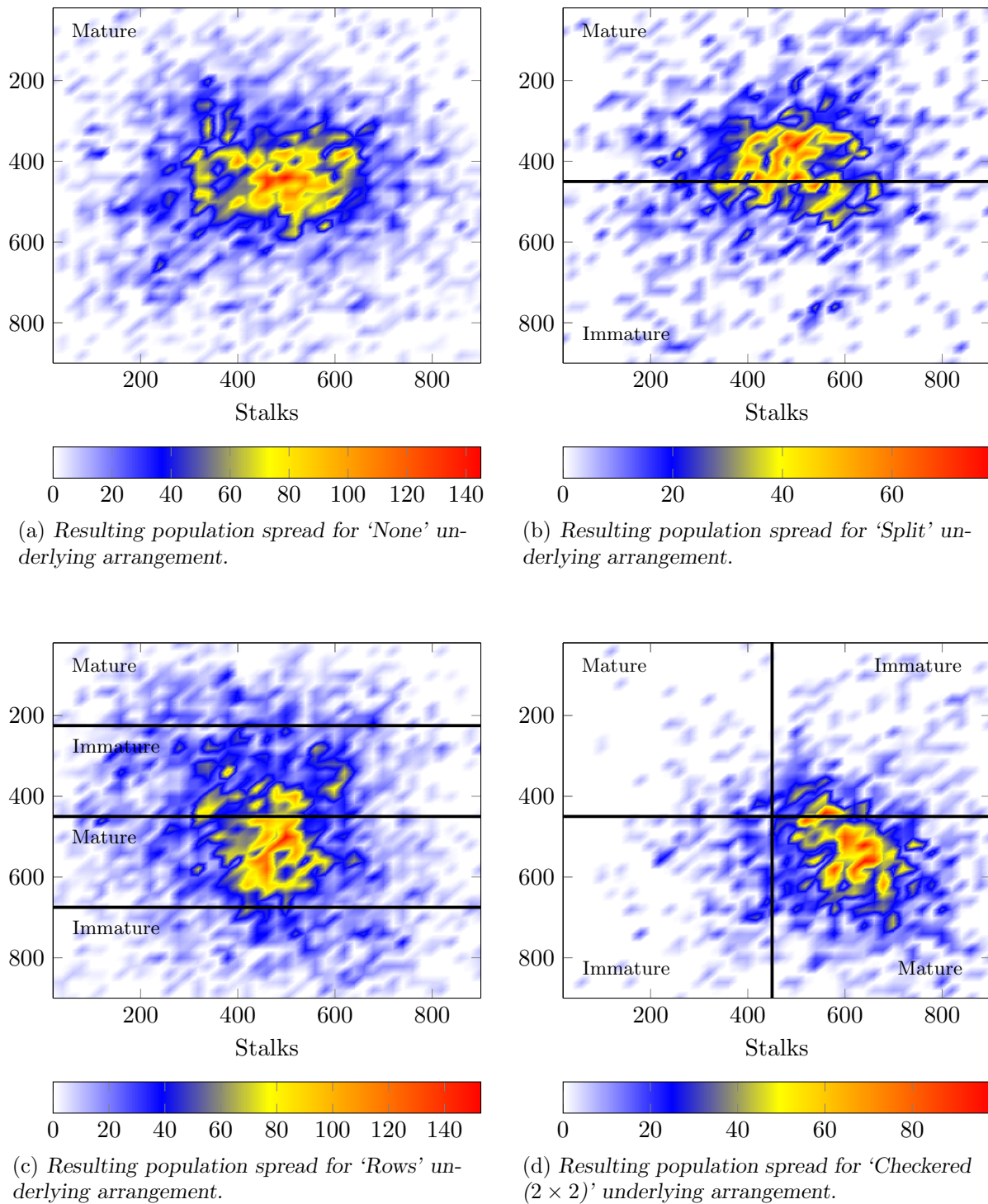


FIGURE 6.39: Population dispersion after six months for different underlying sugarcane maturity values using the 'Find Mature Cane' oviposition site selection algorithm.

In the experiments which employed the **F** FindMatureCane algorithm, the data suggest that the effectiveness of the algorithm in dispersal toward areas of more mature sugarcane is not as influential as deduced from Figures 6.38(b)–6.38(d). That said, in all cases, more agents were observed to reside in mature sugarcane after the six-month simulated period. The poorest performance was observed in the 'Rows' layout case. This is likely due to the fact that the

	Mature Area	%	Immature Area	%
None MW	3 820	100%	0	0%
Split MW	1 958	61%	1 262	39%
Rows MW	1 998	51%	1 912	49%
Checkered MW	875	47%	979	53%
None FMC	24 823	100%	0	0%
Split FMC	7 229	61%	4 629	39%
Rows FMC	15 400	53%	13 773	47%
Checkered FMC	7 996	65%	4 305	35%

TABLE 6.6: Tally of agents residing in each area of different sugarcane maturities.

rows are too narrow in width (25m) in the simulated space and, since many females possess a **F**FlightAmbition in excess of this width, even those females who move towards the rows with mature sugarcane are likely to oviposit only some of their eggs within a certain row and eventually progress into the next row containing immature sugarcane to complete their oviposition cycle. In reality, however, a sugarcane field is unlikely to be diversified on such a small scale and, as such, this incidence of females moving through a mature area to a neighbouring immature area is unlikely to occur in a simulation. In spite of this, the gathered data give a good indication of the difference in effectiveness in terms of the manner in which the two algorithms aid in the dispersal of *E. saccharina*, as well as which underlying sugarcane structures provide the most notable biased dispersion.

Neither oviposition site selection algorithm is seen to be more correct or more suitable than the other and both appear to have merit in the manner in which the resulting spread of oviposited eggs emulates what is observed in-field. The **F**FindMatureCane does, however, result in a far more computationally expensive simulation instance and, at times, the number of agents becomes too large for the simulation run to advance. Both algorithms provide diverse properties and ability to model a variety of different instances which may be required for experiments by simulation users and, as such, are useful inclusions in the model.

6.7 Chapter summary

In this chapter, a parameter variation analysis was performed to determine the effects of changes in the most notable parameters in the simulation. In some cases, specific control measures were also included in the experimental runs to illustrate the diverse applicability of the simulation model and its ability to facilitate a number of different investigations. The possibilities afforded to the simulation model in terms of the diversity of scenarios which may be recreated by adjusting each parameter value was discussed and evaluated. The model was shown to be a flexible, scalable, modifiable tool with which a number of population dynamics of *E. saccharina* may be simulated. This functionality is deemed to assist in investigations and ‘what-if’ analyses, in an attempt to guide both entomological research pertaining to *E. saccharina*, as well as control measure development and design.

CHAPTER 7

A decision support and analysis tool

Contents

7.1	Motivation	153
7.2	Available parameters for variation in the decision support and analysis tool	155
7.3	Chapter summary	164

As concluded in Chapter 6, the agent-based simulation model designed and presented in this dissertation is best used as a tool for investigating a variety of different scenarios which may occur in an *E. saccharina* population in nature. In doing so, confidence in and better understanding of the most notable behavioural aspects of the pest and the relevant population dynamics may be established. Furthermore, as previously shown, pilot analyses of control measures imposed on the pest in the model allows for gaining insight into how a population may react under different circumstances.

The notions of decision-support and analysis by means of scenario generation are common in the field of Operations Research. This chapter opens with a motivation in §7.1 endorsing an implementation of the model as a decision support tool for use by farmers and researchers in the agricultural field. Thereafter, the settings tab incorporated into the simulation model, which may be used for scenario testing and investigation is described in §7.2, as is the effect of each user-defined parameter, including those analysed in Chapter 6. The chapter closes with a brief summary in §7.3.

7.1 Motivation

The Franz Edelman Award aims to recognise outstanding examples of innovation in research and analytics which improve organisations around the world and, often, change people’s lives [39]. This award is considered to be the most prestigious group award in operations research [41] and has only once ever been won by a South African team. In 1996, the *South African National Defence Force* (SANDF) received the award for their work entitled *Guns or butter: Decision support for determining the size and shape of the South African National Defence Force*. This work aimed to aid in determining the optimal size and shape of the SANDF in the absence of a military threat. It was conducted in association with industry partners, supporting the *joint strategic management process* (JSMP) with *operations research/military systems* (OR/MS) based decision support models. The team created models including a risk model of the probability and potential impact of various defence contingencies, a model of growth potential of force

structure elements, a cost model, a mixed-integer-programming model of all the relationships and a manual model allowing the user to select different tasks and strategies [38]. The results of the project indicated clear opportunities for major additional savings in the force design (size and shape) for SANDF.

Once handed over to SANDF, the theoretical model was intensively tested on a variety of situations, using many different parameter combinations, before it was considered for application in practice. In doing so, the team responsible for management of the force size and shape became so well accustomed to the problem and understood the influence of different parameters and variables in the solution construction so thoroughly that the model was never formally introduced into practice in order to solve the force design problem. Instead, the experienced staff who had worked closely with the model were able to use their expertise, as well as the insight gained from the decision-support system, to formulate problem solutions on their own accord [41].

The development of an agent-based model of *E. saccharina* is a long-term study which will require extensive entomological insight and the corresponding improvement of the simulation model, as well as future research in the fields of both entomology and simulation. There exist many unanswered questions pertaining to the biology and behaviour of the pest which, as a result, cannot be accurately included in the model at present. These questions will have to be answered by in-field research if a very accurate model is ever to be constructed of the pest. Similarly, the approach taken in the construction of the model, as well as the manner in which some attributes of the pest have been incorporated into the simulation, may be approached in a different manner, possibly providing more accurate results. There are also aspects not included in this phase of the research which must be included at a later stage in order to facilitate complete, sound testing on the pest's behaviour in different scenarios.

In light of these shortcomings in the simulation model presented in this dissertation, in a similar vein to the SANDF situation previously described, it is desired that an incarnation of the model should exist as a decision-support tool that can be used to educate, inform and enlighten users to the effect of certain attributes or influences in the real-life issue of *E. saccharina* infestation in sugarcane fields. In a sector as well developed as that of sugarcane and agriculture, it is unlikely that solutions or suggestions made by a computer simulation will be blindly followed or incorporated by researchers and farmers — particularly in a case such as this where the model necessarily incorporates a large number of assumptions. That said, the problem of *E. saccharina* has been investigated with limited success for many years and each new development is generally informed by investigative research in one aspect of the problem. A diverse, facilitative model, such as the one presented in this dissertation, along with continuous potential improvements and refinements may serve a purpose similar to that of the *Guns or butter* model in assisting entomologists and farmers in gaining insight into the various aspects of the infestation and how they can be altered to achieve a desired result. This tool may also inform future research efforts and assist in periodically uncovering the most pertinent questions which require answers, both with respect to the biology and behaviour of *E. saccharina*, as well as with respect to the finer details of relevant control strategies imposed on a population of the pest.

It thereby follows that such a decision support tool must facilitate parameter selection and varying of what are seen at this stage to be the most influential parameters in the simulation model so as to allow for thorough testing and scenario analysis. For this reason, the simulation model presented in this dissertation was implemented and equipped with a 'settings' tab, consisting of a number of important parameters as determined by the model builders in conjunction with industry experts, so as to allow for user-defined inputs and parameter variation in each simulation run. This increases the possibilities in terms of the usability and flexibility of the simulation

model and contributes to its implementation as an education tool capable of generating insight and decision support in the industry.

7.2 Available parameters for variation in the decision support and analysis tool

The start-up screen that is seen by the user upon model execution is pictured in Figure 7.1. Upon selection of the button, the ‘Simulation Settings’ screen is shown. In this tab, all of the parameters available for variation in the simulation model are shown. Depending on the type of oviposition site selection algorithm selected, the screen may either look like that shown in Figure 7.2(a) (when the **F** FindMatureCane is set to be employed), or in Figure 7.2(b) (when the **F** MovingWindow algorithm is selected for use).

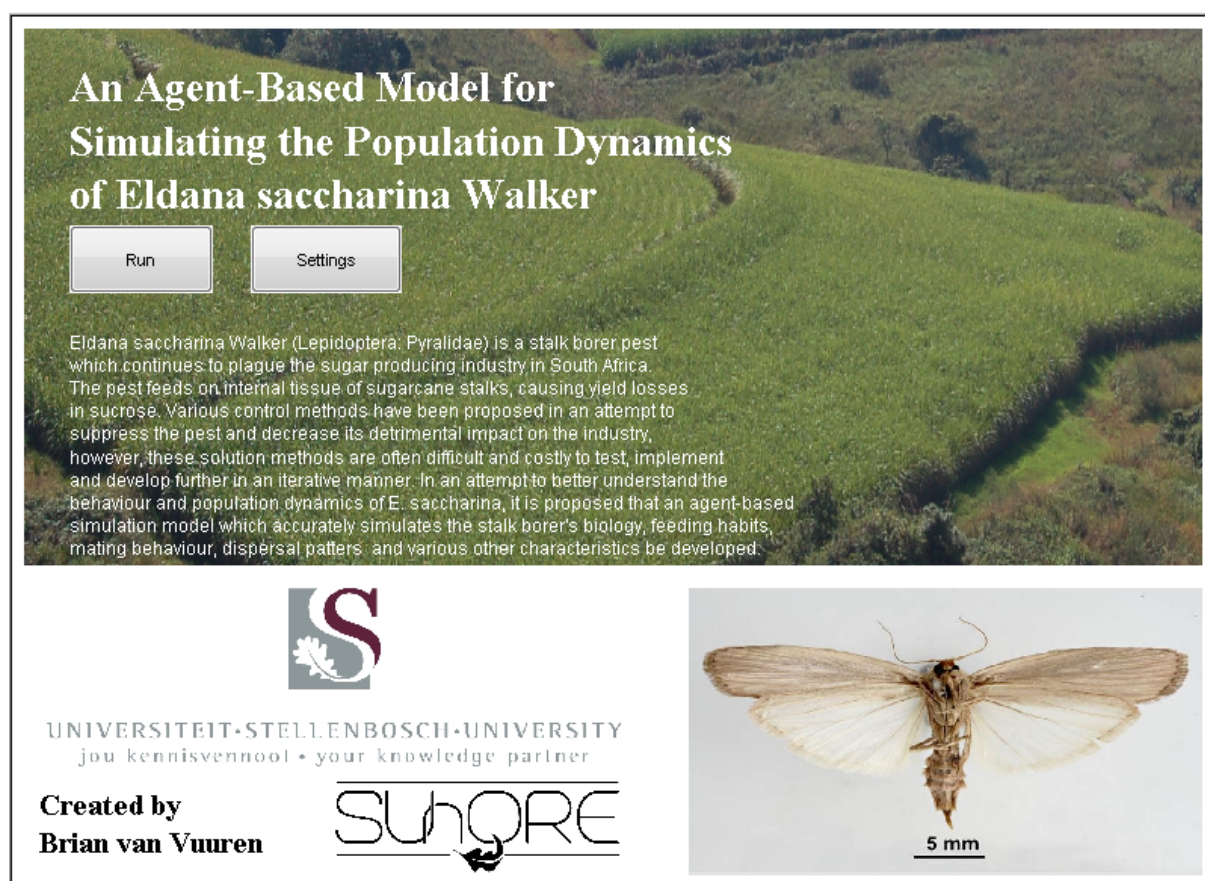
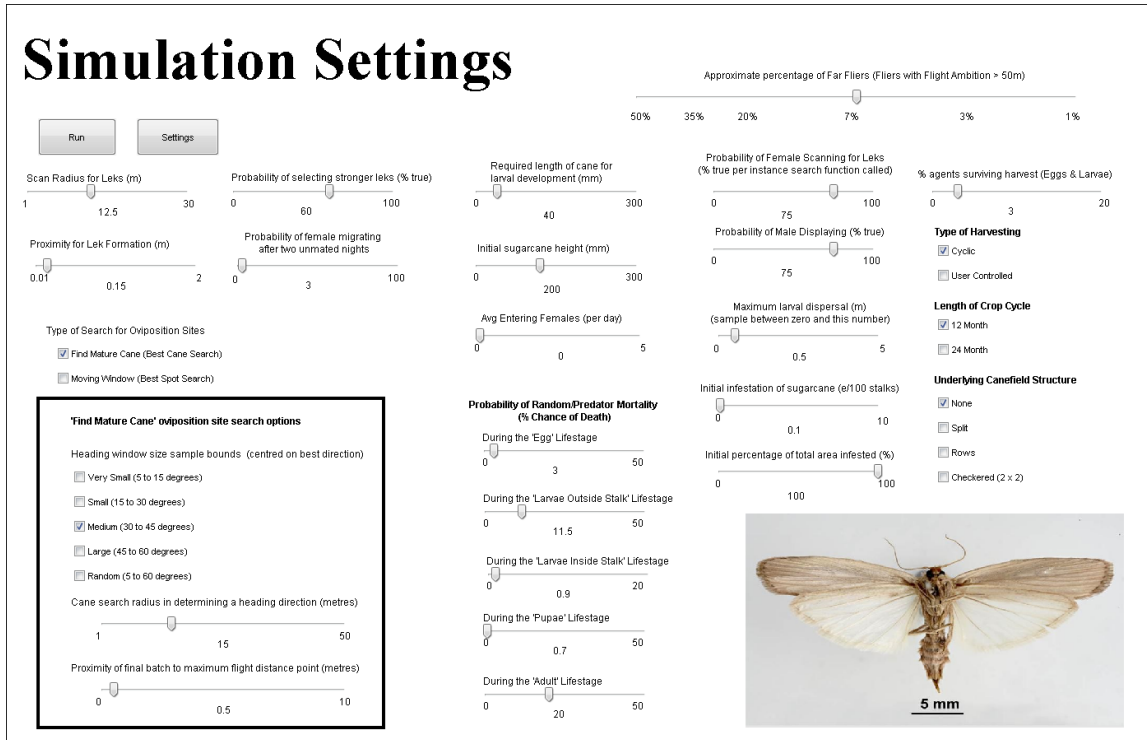
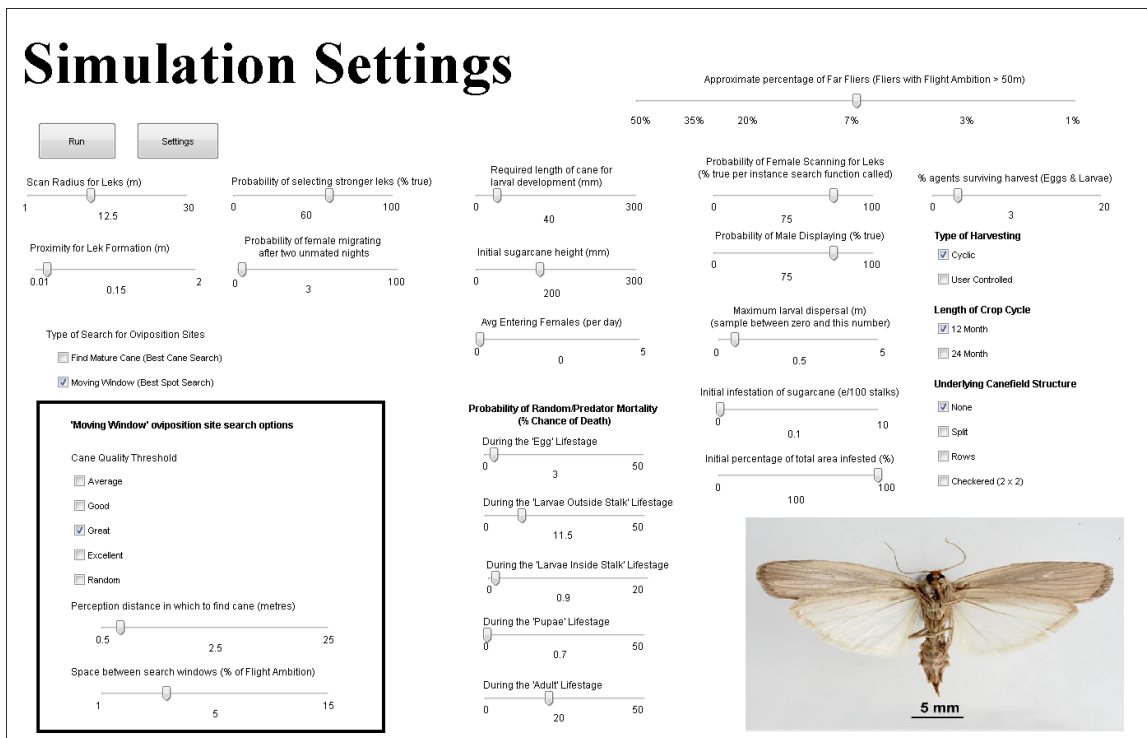


FIGURE 7.1: The display page seen by the user upon model execution whereby the model can either be run ‘as-is’, or the settings tab can be accessed in order to alter user-defined parameters.

A brief explanation of the functionality controlled by each parameter in the settings tab is provided, as well as a ‘best guess’ value attributed to it by default in the simulation. This is accompanied by a brief review of the effect of increasing or decreasing the parameter value (or, where appropriate, selecting a specific tickbox). This is a high-level summary of the descriptions provided in Chapters 4 and 6, and is intended for easy reference by future simulation users, once the necessary level of understanding has been acquired from these two chapters.



(a) The settings tab of the simulation with the 'Find Mature Cane' oviposition site search algorithm selected



(b) The settings tab of the simulation with the 'Moving Window' oviposition site search algorithm selected

FIGURE 7.2: The tab(s) which show the parameter available to the user to be altered for specific simulation runs.

Scan Radius for Leks (m)

Explanation: Sets the maximum distance at which a female will be able to sense a *lek*. The pheromone strength of the *lek* will then be scaled as a function of distance to the female.

Default value: 12.5m.

Effect of increase: During the early stages of the simulation, population numbers increase at a greater rate with a larger scan radius as more mating instances occur.

Effect of decrease: Populations struggle to increase in size during low infestation levels as fewer *leks* are discovered by females.

Proximity for Lek Formation (m)

Explanation: Sets the maximum distance allowed between two males who wish to contribute towards the same *lek*.

Default value: 0.15m.

Effect of increase: Population numbers increase at a greater rate as more *leks* are formed between males, promoting mating.

Effect of decrease: Infestations struggle to progress and often die out as a result of very few *leks* being formed and little mating activity.

Probability of selecting stronger leks (% true)

Explanation: Attributes a more natural means of *lek* selection where the female's selection process is guided, but not exclusively governed, by the relative strength of the discovered *leks*. This is also important for analysis pertaining to SIT, since irradiated males and females may have different competitiveness levels than those of non-irradiated moths. This phenomenon may be modelled using this parameter. In a non-SIT scenario, however, this probability has little to no effect on the overall population growth since all females still mate and oviposit eggs. For this reason, it was excluded from the parameter variation analysis.

Default value: 60%.

Effect of increase: *Lek* selection and occupation is more predictable and rudimentary. The first female to discover and approach a strong *lek* typically occupies it.

Effect of decrease: The relative attractive strength of *leks* plays a decreasingly prominent role in female selection.

Probability of female migrating after two unmated nights

Explanation: Models the observations by Atkinson [4] whereby unmated females migrate after two unmated nights in search of alternative male populations for mating. In this case, the migrating agents are instructed to leave the simulation space completely.

Default value: 3%. This value was discussed and determined with the assistance of Conlong [19]. It is, however, nearly impossible to accurately determine and exists only as a 'best guess' value. It was therefore excluded in the parameter variation analysis.

Effect of increase: More females depart from the simulation after two unmated nights. May lead to shrinkage of the female population and an overpopulation of males.

Effect of decrease: Fewer or no females depart from the simulation space — even if they remain unmated.

Required length of cane for larval development (mm)

Explanation: This parameter represents the length of available sugarcane which must exist in a stalk if a larva is to successfully penetrate and feed on a particular stalk. If not, the larva will perish as a result of the carrying capacity of the sugarcane.

Default value: 40mm. This value can be determined experimentally as a function of temperature and sugarcane quality, but was included as a user-defined parameter in light of the simplified implementation of sugarcane in the simulation and, in doing so, can be used to account for possible difference in nutrition quality between different sugarcane varieties.

Effect of increase: More sugarcane stalk length is required per larva and, as such, each stalk can accommodate fewer agents. The effect of the carrying capacity is greater and more larvae perish.

Effect of decrease: The effect of the carrying capacity decreases since more larvae can be accommodated in each stalk.

Initial sugarcane height (mm)

Explanation: This parameter sets the height of the sugarcane at the start of the simulation and accommodates the initiation of a simulation at any point in time relative to a harvest.

Default value: 200mm.

Effect of increase: There is more available stalk length for larvae to feed on during the early stages of the simulation and, as such, the effect of the carrying capacity is less.

Effect of decrease: More larvae perish as a result of carrying capacity since there is less available sugarcane on which to feed during the early stages of the simulation when the sugarcane has not yet had time to grow sufficiently.

Avg Entering Females (per day)


Explanation: Models the average number of females who enter into the simulated space in the adult state through the emitting boundary condition. These females are assumed to be entering the simulation in search of suitable oviposition sites and, therefore, are in the middle of an oviposition cycle. This is an important parameter for scenario testing such as, for example, consideration of farms employing good farming practices with respect to pest control which may be adjacent to other farms with no pest management system. In this case, the possibility of infestation from neighbouring farms into the field must be accounted for.

Default value: 0. There currently exists no means to determine this value and it is relevant to specific cases under investigation. As such, it is simply an extra feature of the simulation for functionality and scenario testing purposes.

Effect of increase: More females enter the simulation from the hypothetical surroundings and the population expands at a greater rate, since more eggs are oviposited in the simulated space. This also results in a mixing of the generational cycles within the population since eggs are oviposited at any given time by entering agents and do not conform to the typical cycle observed in a static population.

Effect of decrease: The population consists only of agents originating from mating instances of existing agents residing within the simulated space. This results in cycles of population maturation and fewer instances of all life stages being present at one point in time in the simulation.

Probability of Female Scanning for Leks (% true per instance search function called)

Explanation: Controls the probability with which a female will perform a scan for *leks* each time the  **LekSearch** function is called. This is done in an attempt to model a more ‘natural’ process whereby not all females will attempt to mate all of the time.

Default value: 75%.

Effect of increase: The effects of altering this parameter were seen to be minimal in §6. A greater percentage of the available *leks* are likely to be discovered for higher parameter values.

Effect of decrease: Similarly, decreasing this parameter has little effect on model execution. If it is set to be very low, the population may fail to propagate in the presence of no mating instances.

Probability of Male Displaying (% true)

Explanation: This parameter aims to mimic a more natural choice of males to display and controls the probability with which a male will enter the **LookingforLeks** state whereby he becomes available to contribute to the formation of a *lek*.

Default value: 75%.

Effect of increase: Increase of this parameter was also seen to have little effect on model execution in §6, but it was noted that *leks* were typically available at a slightly earlier stage in the maturation cycle for higher probabilities since a greater percentage of the available adult males attempted to contribute to their formation.

Effect of decrease: A greater number of adult males are typically required within a certain area before *leks* will begin to form, since fewer of them will be displaying.

Maximum larval dispersal (m) (sample between zero and this number)


Explanation: A sample is drawn from a symmetrical triangular distribution between zero and the number attributed to this parameter. This sample then determines the distance covered by the neonatal larvae during the ‘parachuting’ process before stalk selection and penetration. In effect, this parameter controls the larvae distribution in the simulation.

Default value: 0.5m.



Effect of increase: Larvae disperse over generally greater distances and a larger number of sugarcane stalks are infested, resulting in less competition and a smaller effect of carrying capacity.


Effect of decrease: Larvae dispersion is very localised and large portions of each larvae pod perish as competition for feeding on the small number of stalks amongst which they disperse is high. The effect of the carrying capacity of the sugarcane is distinctly more pronounced in this case.

Approximate percentage of Far Fliers (Fliers with Flight Ambition > 50m)

Explanation: This parameter sets the ‘rate’ parameter (λ) of the Erlang distribution from which the  **FlightAmbition** parameters of female agents is sampled. That said, selection of a rate parameter for a distribution is not intuitive for users and, as such, the selection was translated into an approximate resulting consequence (hence the exponential scale of the selection bar).

Default value: 7% (equivalent to $\lambda = 1.5$).

Effect of increase: The samples drawn from for the  FlightAmbition parameters of females are generally higher and the dispersal of the population, particularly that of females during oviposition, is greater. An increased  FlightAmbition appears to ‘up-scale’ the distances covered during oviposition and promotes greater dispersal by females.

Effect of decrease: The samples drawn from for the  FlightAmbition parameters of females are generally lower and the subsequent dispersal of the population, including that occurring during oviposition, is lower.

Initial infestation of sugarcane (e/100s)

Explanation: This parameter determines the number of agents with which the simulation is populated upon execution. It translates the initial infestation to a value based on the percentage of the 129 600 stalks infested in the simulation.

Default value: 0.1.

Effect of increase: A larger number of agents are included in the starting population of a simulation run and, as such, the resulting population typically propagates faster.

Effect of decrease: Fewer agents compose the starting population. If the initial infestation level is too low, the population will frequently die out before sufficient propagation can occur.

Initial percentage of total area infested (%)

Explanation: This parameter, in conjunction with the initial infestation level, determines the number of agents who populate the starting simulation. Furthermore, it controls the initial placement of these agents. The percentage infested area selected exists as a scaled square area in the centre of the simulated space. In the case of 100% initial infestation, agents are spread over the entire simulated hectare.

Default value: 100%.

Effect of increase: Agents are more diversely spread within the simulation space and initial mating activity is low. This also depends on the accompanying infestation level.

Effect of decrease: Agent distribution over the simulated area is more clumped and the population typically propagates faster. Once again, this relies on the accompanying initial infestation level.

Probability of Random or Predator Mortality (During the ‘Egg’ Lifestage)

Explanation: Models the fixed probability with which an agent will perish whilst still an egg.

Default value: 3%.

Effect of increase: A larger percentage of eggs perish in the simulation. For small fluctuations, this effect is not noticeable.

Effect of decrease: Fewer eggs perish in the simulation as a result of predator or random mortality.

Probability of Random or Predator Mortality (During the ‘Larvae Outside Stalk’ Lifestage)

Explanation: Models the fixed probability with which an agent will perish as a larva before it penetrates a sugarcane stalk.

Default value: 11.5%.

Effect of increase: A larger number of larvae outside the stalk perish in the simulation. This functionality can be used to simulate the inclusion of pesticides which affect the pest during this life stage.

Effect of decrease: Fewer larvae outside of stalks perish. Small fluctuations in this parameter value were found to have little effect on the resulting population growth.

Probability of Random or Predator Mortality (During the 'Larvae Inside Stalk' Lifestage)

Explanation: Models the fixed probability with which an agent will perish as a larva inside a sugarcane stalk.

Default value: 0.9%.

Effect of increase: A larger number of larvae will perish whilst feeding for reasons other than that of the carrying capacity of the sugarcane. This functionality may be used to simulate the effects of biological control measures since biological agents typically attack larvae during this life stage.

Effect of decrease: Fewer larvae will perish, but those who do, do so as a consequence of the carrying capacity of sugarcane, rather than random or predator mortality.

Probability of Random or Predator Mortality (During the 'Pupae' Lifestage)

Explanation: Models the fixed probability with which an agent will perish as a pupa.

Default value: 0.7%.

Effect of increase: A larger number of pupae fail to progress to the adult life stages.

Effect of decrease: More pupae mature into adult moths.

Probability of Random or Predator Mortality (During the 'Adult' Lifestage)

Explanation: Models the fixed probability with which an agent will perish prematurely through factors other than maturation.





Default value: 20%.

Effect of increase: A larger number adult agents will perish as a result of factors other than age or maturity.

Effect of decrease: Fewer adult agents will perish randomly and mortality will be primarily as a result of age and maturation. Fluctuation of this parameter was seen to have little influence on simulation results.

Checkbox name: Type of Search for Oviposition Sites: Find Mature Cane (Best Cane Search)
(default case is unselected)

Heading window size sample bounds (centered on best direction)

Explanation: This selection box specifies the bounds within which the heading window around the best identified direction by the female is generated. The sample indicates the angle between the  HeadingWindowStart and  Heading lines and, similarly, the  Heading and  HeadingWindowEnd lines.

Default value: Medium (30 to 45 degrees).

Effect of increase: The sampled window size is larger which results in the female adhering less strictly to the best identified direction during oviposition. This results in more sporadically oviposited eggs, as well as less direct dispersal toward areas of more mature sugarcane.

Effect of decrease: The sampled window size is narrower which results in the eggs oviposited by the female at sites in close proximity to the best identified direction. The resulting dispersion of eggs is less sporadic and more linear in appearance; and the agents typically disperse faster toward areas of more mature sugarcane.

Cane search radius in determining a heading direction (metres)




Explanation: This parameter specifies the radius of the search area within which the female identifies the maturity of the surrounding sugarcane stalks in order to determine the most suitable direction.

Default value: 15m.


Effect of increase: The female has a greater chance of discovering mature sugarcane and, as such, the population will tend toward these areas at an accelerated rate.


Effect of decrease: Females who are located far from mature sugarcane areas are more likely to embark on oviposition flights into areas of immature sugarcane. This results in a slower rate of dispersal toward areas of high sugarcane maturity.

Proximity of final batch to maximum flight distance point (metres)

Explanation: This parameter forces the female agent to oviposit the final batch of eggs at a location within this proximity to the intersection point of her accompanying  FlightAmbition and the  Heading direction line. In essence, it imposes a specification on the female as to how much of her  FlightAmbition must be covered during oviposition.

Default value: 0.5m.

Effect of increase: During oviposition, it is possible for the female to fly across a smaller portion of her  FlightAmbition and, as such, the population is less diversely spread within the heading window.


Effect of decrease: Females are forced to cover most or all of their  FlightAmbition distance during oviposition, resulting in widespread population dispersal within the simulated space.

Checkbox name: Type of Search for Oviposition Sites: Moving Window (Best Spot Search)
(*default case is selected*)

Cane Quality Threshold

Explanation: This selection box imposes a minimum quality of sugarcane which will suffice as a suitable oviposition site for the female. This quality bound is attributed to the sugarcane maturity value of each stalk, but, since the values attributed to sugarcane quality are simply indicative and have no biological significance in the simulation, it is not intuitive to the user how to specify a specific number. For this reason, the numbers were translated into descriptive measures in each case.

Default value: Great.

Effect of increase: Suitable oviposition sites are more difficult to find and the female is more likely to divert from the original departure direction and move towards specific sugarcane stalks of a suitable quality which fall within her search window. This increases the diversity of the resulting dispersion of oviposited eggs. Very high quality thresholds imposed on oviposition increase the possibility of not satisfying the constraint before coming to the end of the flight distance as determined by the  FlightAmbition of the female. If this occurs before all six batches of eggs are oviposited, the female will resort to random location selection in the vicinity of the end point of the flight.

Effect of decrease: Suitable sugarcane stalks for oviposition are easily discovered, typically in close proximity to the random direction line along which the female flies. This results in a more linear spread of oviposited eggs in the flight direction.

Perception distance in which to find cane (metres)



Explanation: This parameter specifies the radius of the circular window centered around the female within which she assesses stalks in an attempt to find a stalk of a suitable quality to serve as an oviposition site. If the distance between a stalk and the female's current location is smaller than the distance specified by this parameter, it is considered as a potential oviposition site and its maturity will then be assessed by the female.

Default value: 2.5m.

Effect of increase: A large number of stalks are considered by the female and there is a greater possibility of satisfying the quality threshold. In cases where the random direction of flight is close to the boundary between two areas of differing maturity, a larger perception distance may assist in pushing the female toward the more mature sugarcane. Very large perception distances can cause overlapping of subsequent windows which results in the same stalks being assessed on multiple occasions. This promotes the development of 'hotspots' (discussed in §6).

Effect of decrease: Fewer stalks are considered as potential oviposition sites by the female. This decreases the possibility of discovering a stalk suitable for oviposition and can result in more frequent random site selection once the full flight distance has been covered.

Space between search windows (% of Flight Ambition)

Explanation: This parameter specifies a minimum distance which must be covered by the female between each search window. It is relative to the  FlightAmbition parameter of the female, meaning far fliers will have larger minimum distances imposed between windows than females with a small  FlightAmbition.

Default value: 5%.

Effect of increase: Distances between batches of eggs are larger and there is a smaller likelihood of 'hotspots' developing. If the parameter is increased too much, the female may not be able to search a sufficient number of windows during the oviposition site selection process and, as such, will resort to random site selection at the end of the flight in order to oviposit the remaining batches of eggs.

Effect of decrease: Distances between batches of eggs are smaller and, as such, the population is more concentrated. This also increases the possibility of search windows overlapping, resulting in the development of 'hotspots' which are repetitively exploited by the female.

% Agents surviving harvest (Eggs & Larvae)

Explanation: This parameter sets the probability with which an agent still in the Egg or LarvaeInsideStalk state will remain after the harvest function.

Default value: 3%.

Effect of increase: Larger numbers of eggs and larvae remain post-harvesting and a second generation infestation propagates with ease and typically faster.

Effect of decrease: Fewer eggs and larvae remain and the second generation propagates at a slower rate, sometimes failing to progress at all.

Selection Boxes:**Type of harvesting**

Explanation: This selection box determines whether the harvest should occur as an automatic event on a date pre-specified within the simulation construction, or whether the user will execute the harvest by means of a click control button during the simulation.

Default value: User Controlled.

Length of Crop Cycle

Explanation: This selection box determines the type of crop cycle employed in the simulation and, consequently, the sugarcane growth as determined by the respective CANEGRO model.

Default value: 12 Month.

Underlying Canefield Structure

Explanation: This selection box specifies the arrangement of the underlying sugarcane field in terms of the options presented in §4.3.2.

Default value: All Mature.

7.3 Chapter summary

This chapter began with a motivation endorsing the presentation of the agent-based model of *E. saccharina* as a decision-support tool which assists in educating and equipping in-field experts to gain a better understanding of the problem at hand. Reference was made to the award-winning *Guns or butter* decision-support system which, despite high accolade, was never formally used in practice, but simply aided to facilitate gaining an in-depth understanding of a variety of possible scenarios pertaining to the modelled problem. Following this motivation, the ‘settings’ tab of the simulation model was shown and discussed for reference to potential future users. Each user-modifiable parameter was briefly described and the default value associated with the parameter during simulation execution was provided. Following this, a short analysis of the potential effects of increasing or decreasing each particular parameter was provided in order to assist the reader in scenario generation and model execution understanding.

CHAPTER 8

Model development through expert opinion

Contents

8.1	The Delphi technique	165
8.2	Expert panel discussion	167
	8.2.1 <i>Discussion topics</i>	168
	8.2.2 <i>Simulation of the underlying sugarcane and larval infestation</i>	169
	8.2.3 <i>Simulation of other notable elements</i>	171
8.3	Changes made to the simulation model	172
	8.3.1 <i>The oviposition and dispersal process followed by female moths</i>	172
	8.3.2 <i>Simulation of the underlying sugarcane and larval infestation</i>	176
	8.3.3 <i>Other alterations made to the simulation model</i>	180
	8.3.4 <i>Functionality assessment and comparison with previous performance</i>	182
8.4	Chapter summary	189

In this chapter, the well-known Delphi technique is briefly reviewed as a group-based process that may be used to collect the opinions of experts on a particular subject in an unbiased manner so as to reach consensus regarding the subject matter. Thereafter, the details, responses and suggestions obtained during an application of this technique conducted in association with relevant industry experts in order to refine and improve the agent-based model of *E.saccharina* are noted. The implementation of subsequent improvements or changes to the model in light of this discussion, where possible, are then described, and suggestions for future work stemming from the discussion are also noted.

8.1 The Delphi technique

The Delphi technique is defined as a process involving the interaction between a researcher and a group of identified experts on a specified topic, usually through a series of questionnaires [97]. The technique is used to gain consensus, typically pertaining to future trends and projections in some context, by using a systematic process of information gathering. Skutsch and Hall [79] identify the Delphi technique as a useful method for gaining judgements on complex matters in cases where precise information may not be available.

Linstone and Turoff [57] mention two of the typical instances that may lead to use of the Delphi technique. First, instances where a problem does not lend itself to analytical techniques, but

can benefit from subjective judgements on a collective basis may be assisted by Delphi-type discussions. Furthermore, cases where individuals are required to contribute to the analysis of a broad, complex problem and may represent diverse backgrounds with respect to experience or expertise. The technique is therefore an ideal manner in which to critically evaluate and refine the simulation model of *E. saccharina* presented in this dissertation. This is in light of the fact that many aspects of the biological behaviour of the pest are still not thoroughly understood, and remain unquantified in terms of specific attributes. Furthermore, new information is constantly being discovered through entomological experimentation or deduced from in-field observations. Not all of this knowledge is yet formally recorded or published in the literature as it is often based on the opinion and interpretation of experts in the field. For this reason, discussions surrounding the poorly understood aspects of the pest are paramount to advancing the knowledge base of *E. saccharina* and, more importantly, providing guidance in the modelling process in order to ensure that the simulation model represents the known behaviour and observations of the pest as currently understood by experts.

The basic steps of the Delphi process are outlined by Pfeffer [63]:

1. A first questionnaire is given to the panel of experts requesting opinions and judgements, based on their experience, as well as a list of recommendations regarding the subject matter under scrutiny.
2. During the second round, a collective list of the responses, judgements and recommendations is distributed to each expert and he or she is asked to rate or evaluate each item in terms of its importance.
3. The third questionnaire includes the list, ratings and consensus, if any. Experts are then asked to reconsider their opinions, discussing possible reasons for disagreements if a group consensus has not been reached.

Such a process may be conducted in a central location with all contributors present, or via telecommunication and email. In either case, it is important that responses are kept anonymous in order to promote honest, diverse viewpoints.

The Delphi technique, as mentioned, is of particular benefit when other methods are not adequate or appropriate for data collection. The technique therefore leverages subjective judgements on a collective basis by experts in a particular field. Furthermore, a larger number of contributors can typically be accommodated using a Delphi technique than during informal discussion, and they may also contribute from a remote location, using the cyclic feedback system. This increases the efficiency of face-to-face interaction and assists in avoiding disagreements among individual members. This further mitigates domination by quantity or strength of personality by certain members of a discussion and allows for all experts' opinions to be considered equally through anonymous submissions [96].

The Delphi technique is, however, not without limitation. The consensus that may be reached by contributors may not necessarily be a true consensus, but rather a product of speculation or manipulated consensus [60]. Linstone *et al.* [57] identify a number of typical reasons that may cause the Delphi technique to fail. Imposing preconceptions of a problem upon the respondent group may disallow contributions of other perspectives related to the problem, leading to a misconceived consensus from the group. Furthermore, poor summarising and presentation of group responses in pursuit of common interpretations of the evaluation scales employed may also corrupt the processes' advancement. Finally, the failure to explore disagreement within the

group of experts may cause those with opposing viewpoints to relinquish their opinion, leading to an artificial consensus.

Ultimately, the outcome of the Delphi sequence is simply an opinion as valid as the opinions of the experts who comprise the panel [96]. Results should, however, be interpreted statistically or informatively, rather than in terms of a definitive majority vote. In the case of this research, responses were sought leading to refinement and consensus in the modelling approach and assumptions made in the agent-based simulation model of *E. saccharina* presented in this dissertation. Furthermore, where possible, any consensus reached on further additions or improvements which may be made to the model were duly considered and, if pertinent to the scope of the study, implemented.

8.2 Expert panel discussion

An expert discussion was held at Stellenbosch University on 20–21 April 2016, incorporating a selected panel of participants whose fields of expertise were deemed beneficial in the context of improving the accuracy of the *E. saccharina* agent-based simulation model of Chapters 4–7 and agreement between this model and actual biological processes in nature. Addison [1], Conlong [19], Mudhavundu [61], Pringle [69], Rutherford [72] and Way [90] formed the panel, affording a diverse array of entomological insights, not only with respect to *E. saccharina* behaviour, but also to other related species, such as *Cydia pomonella* (the codling moth). This collection of participants encompasses some of the key researchers pertaining to *E. saccharina* and other moth behaviour and population dynamics, both in South Africa and abroad. The discussion members may be seen in Figure 8.1.



FIGURE 8.1: The experts composing the panel during one of the topic discussions.

The discussion made use of an adapted Delphi technique, similar in approach to that described in §8.1. This was done in an attempt to achieve consensus pertaining to a number of assumptions made within the simulation model due to a lack of insight or information available in the literature. The discussion also covered the implementation of well understood *E. saccharina* behaviour and population dynamics in the simulation model of Chapters 4–7 so as to ensure their correct interpretation and incorporation into the model. These elements were discussed in order to ensure their correct implementation to the best knowledge of the participating experts.

All participants of the discussion were physically present and, as such, it was deemed unnecessary to perform three rounds of feedback for continual review by the contributors. Instead, the Delphi technique was modified to suit its application in this particular forum. Owing to the factual, experience-based nature of the subject matter, relatively little conflict arose in opinions and the experts appeared comfortable to share their experience and opinions, questioning one another where necessary. In light of this, the need for anonymity in the feedback phase was diminished and it was decided only to conduct a single, personal feedback on each of the primary discussion points. From this feedback, a list of responses, judgements and recommendations based on each conversation was compiled and distributed to the panel for review, in line with the traditional Delphi technique. Thereafter, an open conversation was held to allow for scrutiny, discussion and refinement of the opinions provided in order to reach consensus about the topics at hand. The details of this discussion were minuted and, as such, no further rounds of feedback were required.

8.2.1 Discussion topics

The conversation began with a thorough overview of the current simulation model described in Chapters 4–7 in order to inform the participants clearly of what had already been implemented in the model, as well as the manner in which each module functioned within the model. Panel members were encouraged to ask questions in order to ensure a comprehensive understanding of the implementation of the different processes within the model.

Thereafter, a number of pre-selected aspects of the model were posed for open discussion and scrutiny. These aspects were selected by the author based on a perceived lack of clarity and conviction as a result of the limited literature available. Experts on the panel were prompted to discuss openly, scrutinize and provide feedback pertaining to these aspects in an attempt to seek out areas requiring improvement, or identify where incorrect assumptions or interpretations had been made. These discussions are reported in the following sections. The consensus reached on each topic in light of the anonymous feedback and open discussions held is also highlighted.

Discussion of the oviposition and dispersal process of *E. saccharina*

The manner in which the oviposition and dispersal process of *E. saccharina* is emulated in the simulation model, as described in §4.6, was discussed in detail with the panel. It became clear at an early stage during this discussion that the process employed by females in selecting oviposition sites (by means of the two available oviposition site selection algorithms detailed in §4.6.2) had been notably over-complicated and that the moths' decision making process was somewhat misappropriated. The spatial working memory assumed to be inherent in female moths appears not to be as proficient as was included in the original model.

It was explained by panel members that male *E. saccharina* moths are believed to be far more sensitive to pheromones and plant volatiles than females. This is evident from the fact that individual males have the ability to seek out other single, displaying males in order to form *leks*, whereas female moths are only attracted to *leks* composed of a number of males displaying at once. Furthermore, these males are more susceptible to plant volatiles that are released by stressed, physiologically aged sugarcane, indicating its preference as a food source for larvae. Sugarcane can become stressed owing to a variety of reasons, the most common of which is shortage of water supply.

As a result of this, males, instead of females, are deemed to gravitate toward areas where the underlying sugarcane provides a suitable food source for offspring. This is also consistent with

the fact that females are the larger sex of the species and, although expected to have a greater ability to disperse over long distances, their larger abdomen is believed to restrict movement in most cases, but for mating purposes. Site location is believed to be largely visual on the broader scale, with the presence of a distinct leaf canopy attracting males due to the greater possibility of forming *leks*. More specific location selection by males is attributed to chemical cues released by the underlying sugarcane, the details of which are not yet thoroughly understood. Once males have settled in a specific display location, they attract females to this vicinity by means of *lekking*. This draws females towards the areas where mating occurs.

Once the gestation period of a mated female has passed, the model of Chapters 4–7 simulates an intensive search process undertaken by females to determine suitable, cryptic egg laying sites in which to oviposit. The experts now believe, however, that this is a far less ‘calculated’ process, and that the female is more likely to oviposit eggs in very close proximity to the mating site. This process is thought to be aided by physiologically and chronologically aged sugarcane, where more cryptic oviposition sites are available in leaf sheaths and buds on the sugarcane stalks, meaning that suitable sites for oviposition are readily available in close proximity to the female.

It was noted that congestion and crowding of oviposition sites is an aspect that would cause females to disperse slightly further from the site of mating in some cases, thus inevitably causing the population to disperse slowly outwards from the central hub of infestation. Heavily infested areas may fail to provide suitable cryptic oviposition sites for females, prompting them to disperse away from their current positions, thus increasing their search area and subsequent chances of successful discovery of suitable sites.

In light of the above discussion, the spatial movement of an *E. saccharina* population causes, in effect, a ‘self-fulfilling’ incidence where generations of the pest do not move significantly, since they inherently find themselves in suitable habitats causing intense infestation in patches located in stressed, aged cane throughout a sugarcane farm. Females, as such, are not required to move large distances in order to locate *leks* and end up mating in close proximity to their site of emergence.

In spite of this information, it was agreed by all experts that a small percentage of females are still likely to disperse on a larger scale (so called far-fliers), potentially spreading the population to previously uninfested areas. This was confirmed to be the case in codling moth populations in order to promote outbreeding [1, 47, 76], but *E. saccharina* infestations are of such a density, and the larval mortality exhibits such a high probability, that little chance of inbreeding exists. Instead, only a small percentage of *E. saccharina* females would typically disperse widely post-mating, resulting in their eggs possibly being oviposited in younger sugarcane which serves as an inferior food source to that in which mating occurred. In such cases, the resulting larvae will still feed on the younger sugarcane, but development will take slightly longer owing to the inferior nutrient content of the food source. Once matured to adulthood, male moths will once again attempt to move, where possible, to better quality sugarcane, thereby focussing the resulting infestation towards the most suitable areas within a particular newly infested field. Some females may also fly over greater distances if they are insufficiently fertilised during the first mating instance. This was confirmed to typically occur until the second night after emergence, but not as late as the fourth night, as was currently accommodated in the model.

8.2.2 Simulation of the underlying sugarcane and larval infestation

The simulation implementation of the underlying sugarcane field within which the simulated *E. saccharina* population resides and interacts was also discussed in detail with the panel. The

growth and development of sugarcane in the model, as well as its effect on the simulated larvae which infest it, as discussed in §4.3.2 and §4.4.2 respectively, was scrutinised by the panel in order to determine the accuracy with which the model emulates the real-world process.

The discussion brought to light the necessity of simulating not exclusively the chronological age of sugarcane (as was captured in the current model), but also the physiological age of the sugarcane, as it plays a notable role in infestation outbreaks. The physiological age of sugarcane is a measure of how well or poorly the sugarcane is functioning relative to its chronological age. This is judged in terms of the plants' physiological development. On sugarcane farms, some poor soil patches typically exist in which the sugarcane is more stressed. Furthermore, during periods of low rainfall, sugarcane also becomes more stressed and, in turn, physiologically older. This 'older' sugarcane typically serves as a better food source to *E. saccharina* because, although the internodes are shorter in stressed sugarcane, they are densely packed with nutrients which decreases the amount of feeding required per larva to mature and become a pupa.

Before stalk infestation, neonatal larvae feed by scavenging [90]. As such, food availability in the form of yeasts and microbes on sugarcane stalks influence survival. In light of this, more stressed plants also aid in survival of larvae before they bore into the stalk owing to the availability of food during this stage, meaning more larvae reach a level of maturity where they can begin to penetrate sugarcane stalks.

Stemming from this observation, a discussion pertaining to the carrying capacity of the sugarcane ensued among the panel members and it was soon realised to be inaccurately implemented in the simulation model of Chapters 4–7. In nature, only a single larva will typically occupy an internode in a sugarcane stalk. Furthermore, it was found to be uncommon to witness more than one larvae per stalk at a given time during in-field observations. The number of internodes of a sugarcane stalk is therefore of more significance than its physical height when determining carrying capacity. In cases where a stalk is not currently infested, the number of internodes may be used to determine whether or not that particular stalk can further accommodate and provide sufficient nutrition to current infesting larva.

It was agreed by the panel that, once infested, an internode of a sugarcane stalk was incapable of regenerating in any significant manner so as to facilitate a future larva in the same node. Instead, only new internodes which continue to grow at the top of the plant as it ages may accommodate future infesting larvae. As such, the stalk is expected to be infested from the base upwards. The experts reported that damage of one to two internodes is common in a stalk. A very heavy infestation may be characterised by infestations of ten or more internodes per stalk.

Stalks usually begin to form four months after seed cane is planted and the rate of internode development depends on both sugarcane growth, and the variety of the sugarcane in question. Furthermore, as mentioned above, factors such as water supply, stress and the physiological age of sugarcane also influence internode size. Coastal varieties of sugarcane were reported to typically grow 16–18 internodes, whilst inland varieties grow 20–24 internodes on average. In all cases, however, a larva typically requires a full internode of feeding in a certain type and quality of sugarcane in order to consume sufficient nutrients to develop to the pupal life stage. In cases where stalks possess shorter internodes, the nutrient supply is typically dense, with the opposite being true for longer internodes. It is therefore estimated that sugarcane possessing a higher nutrient content may accelerate maturity through the larval lifestage by up to five days when compared to lower nutrient cane owing to the decreased feeding time.

The manner in which larvae disperse before they penetrate sugarcane stalks, as well as the stalk selection process, was finally also discussed by the panel. It was confirmed that the 'parachuting' method of dispersal employed in the simulation was accurate, with larvae typically able to move

in the region of 1.5m to either side of the stalk upon which they hatch. Dispersal distances in healthy sugarcane has also been found to be greater as food may be in short supply. This is aided by leaves in green sugarcane which overlap one another, potentially allowing larvae to move between rows of sugarcane. Once dispersed, it was found that larvae typically avoid burrowing into stalks in which other larvae currently reside by means of identifying frass on the stalk surface, indicating active current infestation. That said, in cases where dispersal distances are lower, the subsequent lower predation levels may result in more than one larvae selecting a particular stalk when food is in short supply for a greater larvae generation. This can promote cannibalism between larvae. Furthermore, larvae are also believed to have an inherent ability to select better quality sugarcane stalks to infest, but this was agreed by panel members to be assisted by generally hatching in higher quality sugarcane.

In general, physiologically aged, stressed sugarcane typically offers more trash and microbes on its surface, aiding neonatal development and promoting greater infestations. In the case of shorter internodes, bunched dead leaf sheaths also result in more oviposition sites in areas densely packed with nutrition. These elements promote development, resulting in a survival rate of up to ten times more than sugarcane with long internodes.

8.2.3 Simulation of other notable elements

Several other elements of the model of Chapters 4–7, including those that were deemed to be successfully implemented modules based on biologically documented literature on *E. saccharina* behaviour, were also discussed with the expert panel in order to ensure accurate implementation.

Upon discussion of the implementation of the mating process, it was agreed by panel members that the *lek* mating procedure followed by *E. saccharina* has been acceptably captured in the model using the method described in §4.5.2. Upon further discussion of the typical proximity required by males for a *lek* to form, it was suggested that this distance may be a fluctuating threshold, increasing in magnitude in order to accommodate low infestation, but with a preference to closer proximity in the abundance of other males.

It was, however, uncovered that the manner in which oviposited eggs are simulated in the model was inaccurate. A female moth will typically oviposit between 500–600 eggs in discrete batches. Females who feed on better quality food sources in the larvae stage typically mature to larger, stronger moths who are more competitive and oviposit more eggs. Mortality in the egg stage is very high and, as such, a total of approximately 100–120 eggs will typically survive to hatch into neonatal larvae. The model incorrectly assumed that the female oviposited only a few large batches of eggs (typically six) and, of these, approximately 20 eggs per batch survived in a certain oviposition site. It came to light that, in nature, approximately 30 batches of eggs are often oviposited by a female, each of approximately 20 eggs and, of these, a very small percentage will survive to hatch into larvae. This number is expected to be as low as three or four per batch. Consequently, it was found that neonatal larvae mortality is actually far higher in nature than the laboratory figures documented by Way [91] and typically only a single neonate in a batch of oviposited eggs will survive to bore into sugarcane.

These survival probabilities depend on a number of factors. Neonate mortality is attributed, in some cases, to desiccation, but mainly to predation. In the presence of good quality nutrients, larvae are not required to disperse over large distances and, as such, predation by ants is minimised. Furthermore, stressed sugarcane with shorter internodes means that there are more sites for boring and larvae are not required to move as far up the sugarcane stalk, further decreasing the chance of predator mortality. The shorter internodes also mean that there are bunched

leaf sheaths with more oviposition sites in high-nutrition sugarcane, aiding in decreasing the requirement for neonate dispersal.

8.3 Changes made to the simulation model

In view of the insights provided by the expert panel, a number of changes were implemented in the model to, where possible, allow for a better representation of real-world processes. In some cases, the detailed implementation and modelling of certain aspects was deemed to fall outside of the scope of this study and will require a separate study in order to be captured effectively. Such cases pertain both to the behaviour of *E. saccharina*, as well as to modelling the growth of the underlying sugarcane in the simulation model. For this reason, a simplified interpretation of these aspects was included in the model in such a manner that it would allow for future development and refining.

The changes implemented in the simulation model of Chapters 4–7 are discussed in the following sections.

8.3.1 The oviposition and dispersal process followed by female moths

The inaccuracies in the currently employed oviposition process were addressed and replaced by functionality which mimics the actual process, as discussed in §8.2.1. The **F**FindMatureCane and **F**MovingWindow functions were replaced by a single function employing a new algorithm called **F**FindOvipositionLocation. This function employs a tiered search approach whereby the mated female examines the availability of oviposition sites within her immediate surroundings (*i.e.* close to the mating site). If an uncrowded stalk with suitable sites is discovered, the female moves to this site and oviposits a batch of eggs. This process is repeated until all batches of eggs have been oviposited. If no suitable site is discovered during a search, the radius within which the female searches is incrementally increased, considering sugarcane stalks of a greater distance away, until a suitable site is discovered.

The result of this algorithm effectively mimics the basic process described by the experts during the panel discussion. Females attempt to remain and oviposit in these areas where mating occurred as these are understood to be good sites for larvae to thrive. As mentioned during the discussion, the primary factor that may drive females to disperse from the mating site during oviposition is crowding and a lack of oviposition sites as a result thereof, with a further small percentage of ‘far fliers’ embarking on longer dispersal flights as a result of a genetic predisposition to do so. Females who do not successfully mate within the first two nights after emergence are also prone to resort to longer dispersal flights in an attempt to locate other potential mates. It was, however, deemed unnecessary to model physical ‘searching’ flights of the female during oviposition (*i.e.* the flight path followed by the female as she assesses different sites for oviposition), as this would have no significant outcome on the development of the population. For this reason, the alternative algorithm was designed with the search process preceding movement between each batch of oviposited eggs and the female flying directly to an identified oviposition location.

The working of **F**FindOvipositionLocation algorithm is illustrated graphically in Figure 8.2 where the darker shaded areas close to the female’s mating site have a greater probability of being selected as oviposition sites by the female, decreasing with distance from the female to outlying areas that are only considered in high density populations. The approach is anticipated

to result in a spread of eggs and consequent population dispersal that is reaction-diffusion-like in appearance.

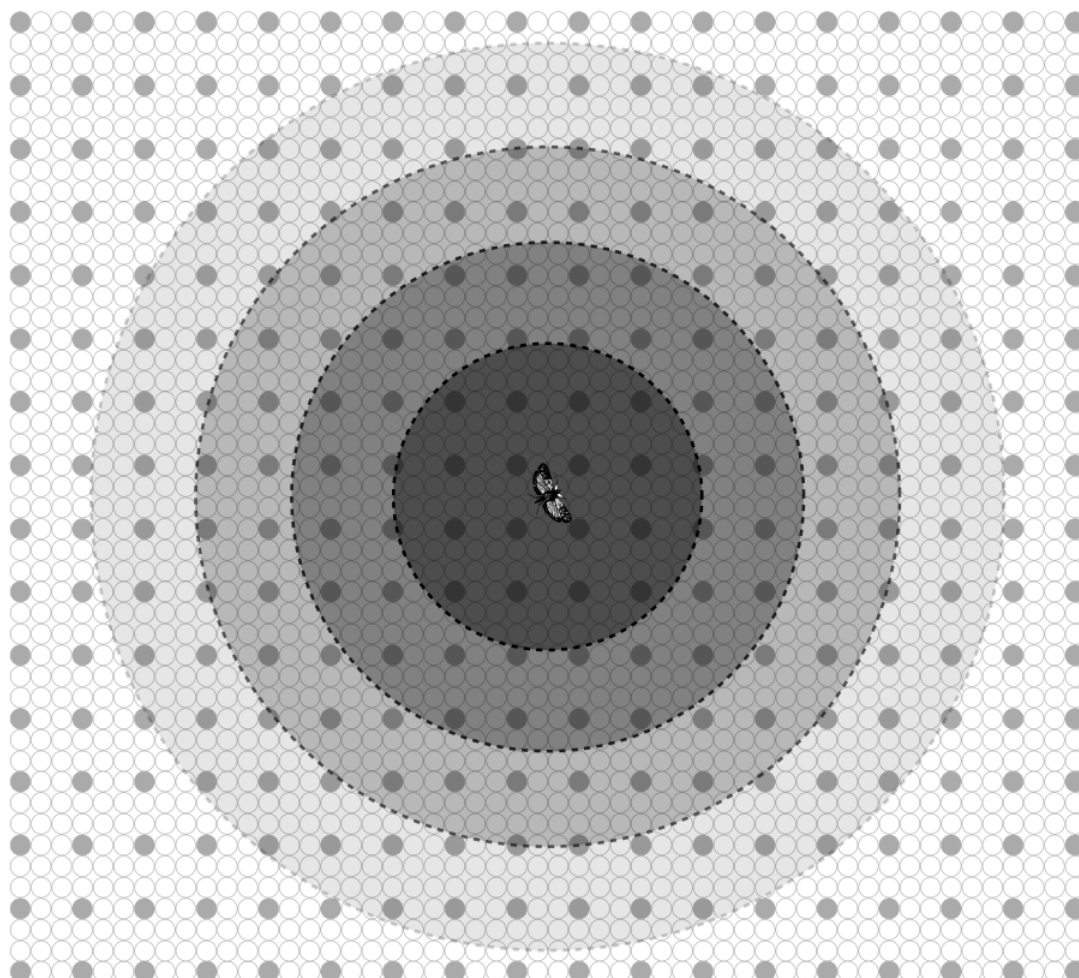


FIGURE 8.2: Graphical interpretation of the revised search process followed by female moths to find sufficiently sparsely populated oviposition sites.

The manner in which this algorithm was implemented in the ANYLOGIC simulation software environment involved the introduction of an **V** Egg_Dispersal array. This array mimics the **V** Sugarcane array in construction, but, instead of keeping track of the properties of the sugarcane stalk represented by each array point, the **V** Egg_Dispersal array stores the number of batches of eggs currently existing on a particular stalk (*i.e.* occupying an oviposition site). That is, the array comprises 129 600 points (representing the number of stalks in a typical hectare of sugarcane), and records each case where a female selects a particular internode site in order to oviposit eggs. Furthermore, once all eggs have either perished or progressed to the next life stage and the oviposition site becomes available once again, the array notes this and re-opens these sites to searching females.

When a new **T** male or **T** female agent is added to the population through oviposition and enters the Egg state, its x and y coordinates are passed to a global function **F** UpdateEggDispersal. This function then records those eggs constituting a batch oviposited by a female as occupying one internode on a stalk within the **V** Egg_Dispersal array, denoted by the agent's positions. Similarly, as each egg matures and the agent moves from the Egg to the Larvae state, or if an

agent perishes as a result of random or predator mortality, the function is once again called to record a decrease in the number of eggs on an internode of the corresponding array point. Once no eggs of a particular batch remain, the occupied internode site becomes available once again for oviposition by another female.

During the **F** FindOvipositionLocation algorithm execution, a female searches through points stored in the **V** Egg_Dispersal array which fall within a specified distance according to each iteration and, as such, a number of oviposition sites may be considered. It is assumed that each node on a sugarcane stalk can provide one cryptic oviposition site by means of a bud or leaf sheaths on the stalk. This includes the node assumed to be at the base of the plant where the root structure begins. Upon assessment of a particular stalk as a site for oviposition, the female assesses the number of available internodes and, consequently, the number of available oviposition sites as recorded by the **F** UpdateEggDispersal array. If a site exists, the female occupies it with a certain probability and oviposits there. If no site is found (or the probability threshold is not met), the search continues amongst the stalks in her immediate vicinity. If, after all stalks within the specified distance threshold have been searched, no suitable oviposition site has been discovered, the search radius is increased to consider a larger number of stalks in a slightly larger surrounding area and the process is repeated. This process continues, with the search radius increasing incrementally, until a suitable site has been located.

Other, more intricate factors which aid in driving the oviposition process are not yet thoroughly understood and, as such, their inclusion and development in the simulation model are left for future work.

Figures 8.3(a)–8.3(i) contain an example of the typical execution of the **F** FindOvipositionLocation function within the simulation. Referring to the various parts of Figure 8.3, the following interpretations are offered:

- (a) The female has progressed past the gestation period and, as such, begins searching for suitable oviposition sites amongst the sugarcane stalks in the immediate vicinity of the mating site.
- (b) No suitable sites are discovered during the search, suggesting that this is a crowded area. The female then increases the radius of the search area and continues to assess the sugarcane stalks within this range.
- (c) This process is repeated until a sufficiently uncrowded site is discovered by the female.
- (d) This site is approached and a batch of eggs is oviposited. The female then re-establishes a local search window, starting again with the closest sugarcane stalks.
- (e) During the second search, another suitable site is discovered. Sites are approached as soon as they are discovered and, as such, suitable sites within previous search windows, which were not considered before the female found an alternative suitable site may be discovered during subsequent searches and the flight pattern of the female may therefore go back and forth between areas.
- (f) This second site is approached and a further batch of eggs is oviposited.
- (g) This process continues in the same manner. It follows that the distances between batches may begin to decrease in length as the search process naturally moves the female away from densely populated areas, thereby increasing the chance that other suitable sites are discovered in the close vicinity of the new areas into which she moves.

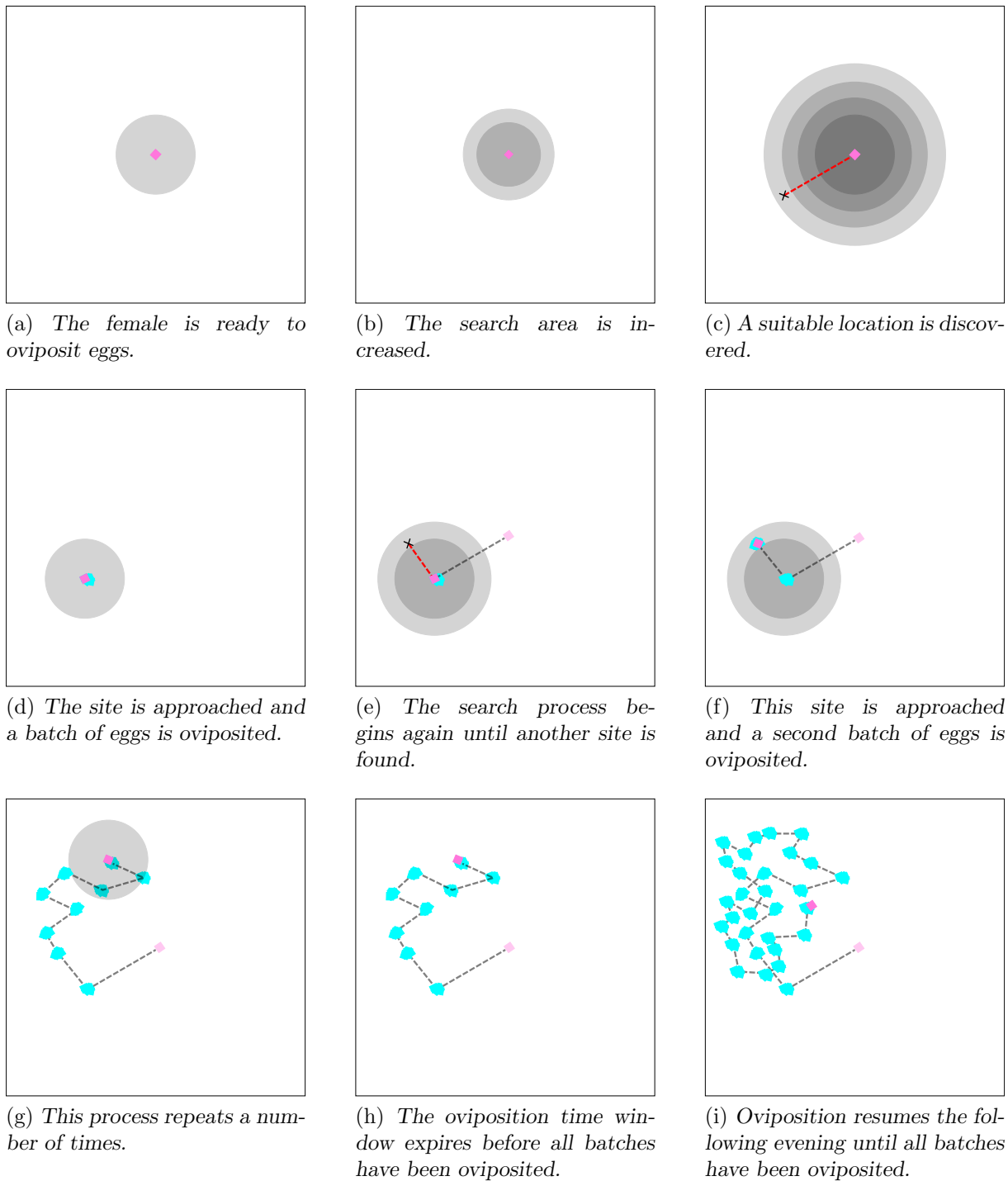


FIGURE 8.3: The **F** *FindOvipositionLocation* oviposition site search function executed in an ANY-LOGIC simulation run.

- (h) If the oviposition cycle is not complete, but the time window for oviposition activity concludes for the night, the female returns to random, small-scale movement until night falls again.
- (i) Once suitable, the female returns to the oviposition process, ovipositing further batches until approximately 30 batches, typically of 20 eggs each, have been oviposited.

8.3.2 The underlying sugarcane and larval infestation

The typical number of internodes per sugarcane stalk varies, depending on a number of factors. Different varieties of sugarcane and the location in which they are grown affects the typical number of internodes. Furthermore, stress and chronological age may also alter the number of nodes witnessed per stalk. During the expert discussion, it became increasingly clear that the simulation of sugarcane growth and development should be based on a number of important factors in order to be accurately captured. The consequent behaviour of the sugarcane, in turn, has the ability to affect *E. saccharina* population development and dynamics. The finer details pertaining to the accurate modelling of sugarcane in the simulation was deemed to fall outside of the scope of this study, as it aims primarily to mimic *E. saccharina* behaviour and population dynamics. That said, a sensible, simplified sugarcane component of the model is required in the interim until a more advanced model can be designed and implemented. Without the underlying sugarcane, the behaviour and dynamics of *E. saccharina* would be illogical and nonsensical, since much of the decision making processes of the pest appear to be influenced by its habitat.

In light of this, a simplified, yet logical modelling approach of the sugarcane, considering the most notable, well-defined influences imposed on *E. saccharina* is implemented in the simulation model, with some of the finer details (some of which still require further in-field research) left for future work.

As mentioned, the manner in which the carrying capacity of the underlying sugarcane was implemented in the simulation model was inaccurate and the necessary changes were therefore made to its implementation. Based on the insights gained from the expert panel that larvae typically require the nutrients of a full internode of sugarcane stalk (irrespective of length or variety of the sugarcane) in order to mature sufficiently, the way in which the sugarcane growth is modelled was appended. The CANEGRO model implementation was retained in the simulation model and still used to model physical sugarcane stalk growth, however, this is then translated into a number of internodes per stalk, providing a better indication of the ability of a particular stalk to host a larva.

Information pertaining to the variety of sugarcane plant simulated and, in conjunction, the typical number of internodes per stalk, is included in the model by means of a linked attribute file containing information pertaining to specific sugarcane farms. This attribute file forms part of the GIS importation process where specific sugarcane farms may be imported into the simulation model so as to execute location-specific simulation runs. The manner in which this is achieved is discussed later in this dissertation. If no typical number of internodes is included for a particular field or variety in the attribute file, a default value of 20 internodes is attributed to that field or variety.

This typical internode value is recorded on a per-field basis with all stalks in a certain field assuming the same average internode value. This information is read from the attribute file and stored in an array called **Ⓜ Sugarcane_TypInternodes** which mimics the **Ⓜ Sugarcane**. Here, a typical number of internodes is set for each stalk, depending on the information provided by the model user with respect to each field. On average, a healthy sugarcane stalk will grow to approximately three metres in height. Using this information, in conjunction with the typical number of internodes per stalk, a further array called **Ⓜ Sugarcane_internodes** is established to translate the current height of the sugarcane stalks into an ‘internode’ value. This value can then be assessed when a larva attempts to infest a stalk so as to determine whether or not there exists sufficient nutrients and space for development.

Confirmed to be accurate during the expert discussion, the model continues to be based on the assumption that sugarcane stalks do not regenerate in any significant manner post-infestation. This means that, once a larva has occupied and fed on a certain internode, future larvae may only be accommodated in stalks which have grown further internodes for infestation purposes. Furthermore, the expert panel noted that typically only a single larva occupies a stalk at any given time. For this reason, two further arrays, namely **V** `Sugarcane_capacity` and **V** `Sugarcane_totalinfestation` are established to manage this process. When a larva selects and penetrates an unoccupied stalk, the corresponding array point stored in **V** `Sugarcane_capacity` is incremented. Similarly, as it departs the `Larvae` state to begin pupation, the array point is decremented. This means that, when other larvae attempt to select the stalks for boring, they can only do so if the **V** `Sugarcane_capacity` of a particular stalk is zero (meaning it is unoccupied). When a larva moves to the `Pupae` state, the corresponding value of the stalk on which it fed is also incremented in the **V** `Sugarcane_totalinfestation` array. This array keeps track of the number of internodes per stalk that have already been infested and, as such, cannot accommodate further larvae. This value can then be subtracted from the **V** `Sugarcane_internodes` to determine whether a stalk still has uninfested internodes, or whether further growth must first occur before boring sites exist on the stalk again. If an available internode exists on a stalk, and the associated **V** `Sugarcane_capacity` value is zero, a larva can successfully select and infest that stalk.

The revised process therefore begins when the eggs mature to the `LarvaeOutsideStalk` state. Upon entry, the **F** `DisperseAmongStalks` function is called, mimicking the ‘parachuting’ process employed by neonates in order to find stalks in which to bore. Once completed, the **F** `LocateStalk` function moves the dispersed larva to the nearest stalk as a potential feeding site. Once at the stalk, a function called **F** `CheckCapacity` is called which determines whether the stalk has an available internode for feeding, as well as whether it is occupied or vacant, effectively simulating the real-life process where surface frass and volatiles cue larvae regarding the suitability of a stalk as a feeding site. If the outcome of the **F** `CheckCapacity` function is positive and the stalk is vacant, the larvae selects and feeds on the stalk. If not, the **F** `FindOtherStalk` function is called, prompting the neonate to search for an alternative stalk in close vicinity as a potential infestation site. Once a new stalk has been located, the **F** `CheckCapacity` function is once again called to determine whether the new stalk may be infested or not. This reselection process is performed a maximum of three times. If, by the third iteration, the neonate has still not located a suitably grown, unoccupied stalk for boring, it is removed from the simulation. This is deemed to account for cases in nature where larvae may resort to infesting stalks without sufficient stalk nutrients out of desperation, leading to under-development and an inability to mate effectively, as well as associated cases where larvae infest stalks in which another larva already resides. This leads to cannibalism and one of the larvae typically perishes. This is effectively captured by the process modelled when insufficient boring sites exist in a local environment.

A further aspect of the manner in which sugarcane is simulated in the model which required attention was the inclusion of a metric indicating the physiological age and, consequently, ‘desirability’ of a certain area of sugarcane for infestation by the pest. As noted by the expert panel, sugarcane of an advanced physiological age is more prone to heavy infestations as it is more stressed than physiologically young plants. As such, such stalks serve as better food sources and decrease development time. Furthermore, the availability of yeasts and microbes on sugarcane stalks is more abundant on physiologically older sugarcane, causing an increase in the promotion rate between the `LarvaeOutsideStalk` and `Larvae` states. Dispersal in aged, stressed sugarcane

is also typically lower since more suitable infestation sites exist. This has the added effect of lower predation on larvae in these fields and greater survival rates of larvae, thereby increasing infestation intensity.

In order to incorporate these aspects of sugarcane physiological age into the simulation model, a physiological age scale is incorporated into the sugarcane module of the model so as to influence the aforementioned areas. In a full-scale sugarcane simulation model, the incorporation of physiological age typically requires information pertaining to the soil quality and rainfall history of a particular area where sugarcane is grown. Each of these factors can play a role in physiological age. As a simplifying assumption, and in an attempt to simulate the response of *E. saccharina* in physiologically aged sugarcane, an algorithm which generates a randomised soil profile, consisting of ‘pods’ of good quality soil in amongst average or poor quality soil, is incorporated into the simulation model. This algorithm gives rise to differing growth patterns within a field. In areas assumed, by the soil profile, to be good growing sites, the physiological age of the sugarcane is assumed to be lower, whilst sugarcane in poor growing conditions is assumed to be more stressed and, as a consequence, physiologically older. Examples of the differing areas of physiologically aged, sugarcane as generated using the algorithm, are shown in Figure 8.4. A further assumption is made that these areas do not change significantly over time and the physiological sugarcane age profile of the simulated area remains constant during model execution. Although a notably simplified implementation, this basis serves the purpose of facilitating decisions made by *E. saccharina* moths which depend directly on the sugarcane in which it resides in the simulation model developed in this dissertation. A more comprehensive, accurate sugarcane model, and subsequent refinement of its effect on the pest’s behaviour, is reserved for future work.

The differing areas arising from the random physiological sugarcane age profile are mapped to an appropriate scale value and stored for each stalk in a static array called **V Sugarcane_PhysAge**. These array values are then implemented in the model at points in *E. saccharina*’s response or decision-making process where physiological sugarcane age is deemed to play a role. The selected scale ranges from 1 to 10, where 10 represents the most physiologically stressed sugarcane stalks present in the simulation space and 1 represents a healthy sugarcane stalk. For each stalk, the randomly aged physiological age value attributed to the stalk in the aforementioned algorithm is compared to the most stressed stalk in the simulation space. The ratio between these two stalks is then converted to a corresponding scaling factor of 1–10. Points or areas in the **V Sugarcane_PhysAge** array which contain higher values are then interpreted as more desirable areas for infestation owing to the factors previously explained. This scale was chosen as it may be easily manipulated to be suitably implemented at points in the simulation model where it is required to alter *E. saccharina* development rates or behaviour, as explained in the following section.

The manner in which the values stored in the **V Sugarcane_PhysAge** array are incorporated into the simulation model is as follows:

- The **MaxDispersion** parameter establishes the upper limit of the triangular distribution from which the dispersal distance of an agent is sampled, where the mode value of the distribution represents the value most likely to be sampled in each case. In the simulation model, the mode of the distribution is weighted for each agent according to the physiological age of the sugarcane in which it resides. This scaling factor shifts the most likely sample towards zero in physiologically older sugarcane, or towards the **MaxDispersion** value in healthy sugarcane. Sugarcane which is more advanced in physiological age requires less dispersion since good food sources are generally more readily available and,

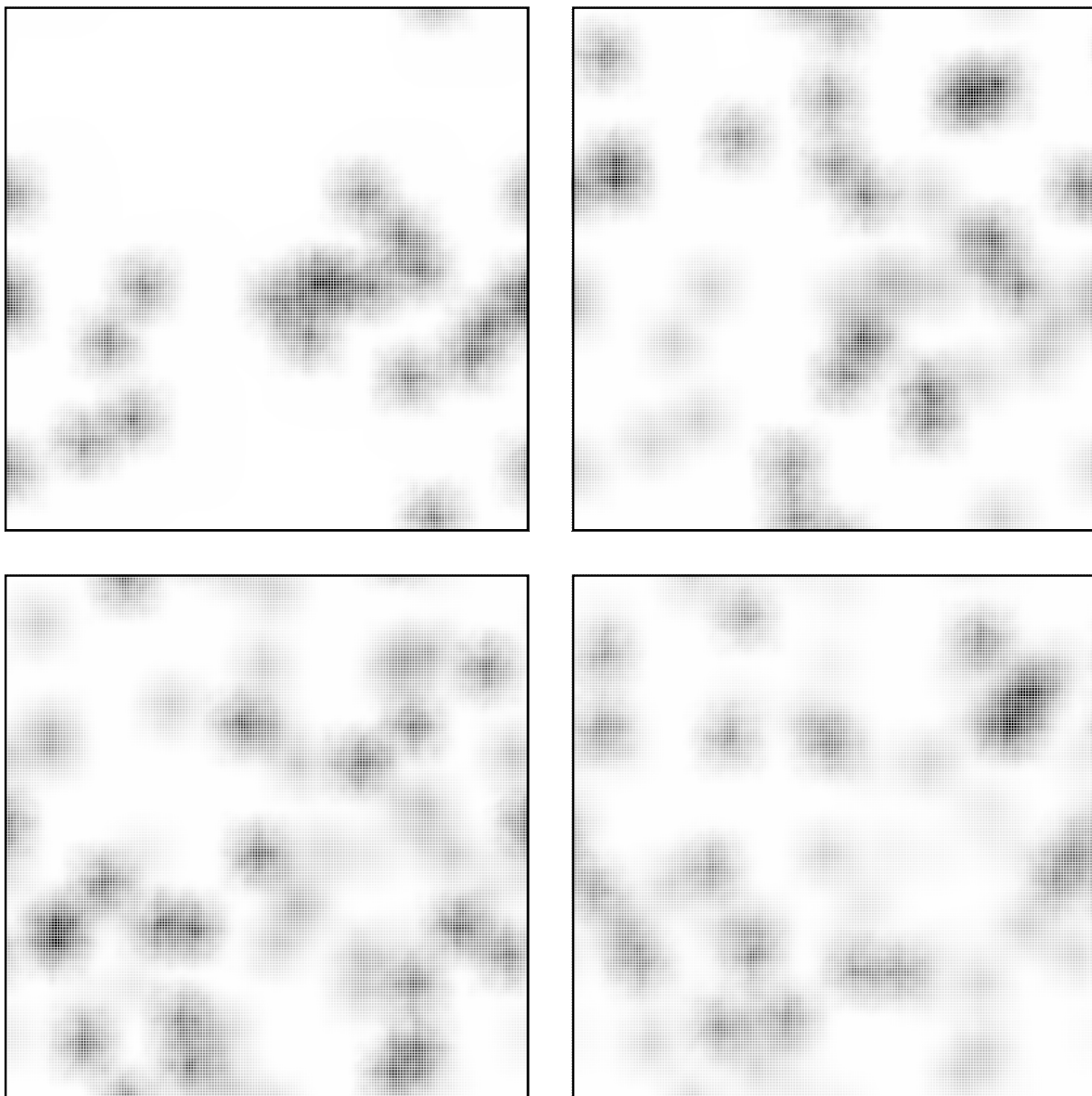







FIGURE 8.4: Examples of outputs from the algorithm which generates randomised soil profiles within the simulation space, giving rise to sugarcane of differing physiological ages. Poor quality soil and, consequently, physiologically aged sugarcane is represented by the darker areas in the profiles, whilst the lighter areas represent healthy sugarcane in more favourable growing conditions.

as such, the mode of the distribution is weighted by the scaling factor toward the lower bound. Conversely, larvae residing in physiologically younger sugarcane disperse according to a distribution weighted in favour of the  **MaxDispersion** value, as established by the user, since a greater search is generally required to locate a site for boring in the absence of high nutrient quality stalks. A function called  **DetermineCaneQualityforAgent** is established in the  **Main** simulation environment which is called by agents upon creation within the simulation environment. This function receives the x and y coordinates of the agent and returns the corresponding physiological age of the sugarcane in which the agent resides, as tracked by the  **SugarCanePhysAge** array. For each agent, this value is stored as a  **CaneQuality** parameter within the agent's statechart and, in turn,

used when establishing the mode of the distribution from which the dispersal distance is sampled.

- The effect of the RDLO (*random death larvae outside*) transition, which models the probability of random or predator mortality in the `LarvaeOutsideStalk` state, is diminished by the appropriate scale value pertaining to the **V** `Sugarcane_PhysAge`, stored as the agent's **C** `CaneQuality` parameter, based on the location of a larva upon maturing and progressing from the `Egg` state. This is achieved by increasing the probability of the specific agent surpassing the threshold for survival in cases when it feeds on sugarcane of a greater physiological age. Implementing this increase results in a survival rate of approximately ten times more in cases of heavily stressed sugarcane where physiological age is at a maximum. This appropriation is based on the insight provided by the expert panel. This survival rate increase is attributed to increased yeast and microbes for feeding, as well as decreased dispersal distances and consequently decreased predation. The increased availability of oviposition sites in these areas, more densely packed with nutrition, are also captured by this survival rate increase as it causes the population to develop at a greater rate in these areas.
- The expert panel also noted that better quality nutrients encountered by feeding larvae may accelerate the maturation rate between the `Larvae` and `Pupae` states, as well as advantage the moth when it reaches the adult state by increasing its competitiveness in mating, as well as its ability to disperse. In order to capture this in the simulation model, the **C** `CaneQuality` value attributed to the agent, based on the sugarcane stalk on which it feeds, is implemented the `Larvae` state in a mathematical equation which results in larvae feeding on physiologically older sugarcane to mature at a greater rate than those in younger sugarcane. The equation is implemented such that larvae feeding on the maximum physiologically aged sugarcane stalks to mature in a time period of approximately five days quicker than those feeding in the youngest sugarcane, depending on the temperatures experienced in both cases.

In accordance with observations conveyed by the expert panel, a function called **F** `SeekOutPhysiologicallyAgedCane` was added to the male's statechart. This function is called in the same instance as the clustering algorithm during the `LookingforLeks` state with an equal probability of the male agent either attempting to move towards a fellow male agent who is displaying, or calling the aforementioned function. When the **F** `SeekOutPhysiologicallyAgedCane` is executed, a search is performed by the male agent within a randomly sampled search radius. During this search, the physiological age of the nearby surrounding sugarcane is assessed and the male is attracted to sugarcane of the most advanced physiological age discovered. The agent then proceeds to move to this area. The function is written as a dynamic function whereby search distances, as well as the regularity with which search instances are initiated, are initiated at random instances, with only a certain probability of successfully occurring in each case. This approach is taken so as to not definitively direct each agent in the adult **M** `male` population to move to the most stressed sugarcane areas as soon as it emerges from the pupa, but rather to simulate a slow gravitation of the general population to these areas over time, as suggested to be the case observed in nature.

8.3.3 Other alterations made to the simulation model

Other aspects of the simulation model that were found to be inconsistent with what is observed in nature were, for the most part, inherently incorporated as a result of the aforementioned

alterations and additions made to the model. The number of batches of eggs oviposited by the female was increased during the oviposition cycle, in conjunction with a consequent increase in both the `Egg` and `LarvaeOutsideStalk` mortality probabilities by means of incorporation of the **V** `Sugarcane_PhysAge` array. Furthermore, some entirely new ‘probability of mortality’ values informed by the expert panel for certain life stages are implemented. This was done in light of feedback from Way [90] that the values published in [91] had been observed under laboratory conditions, whilst the actual mortality rates during these stages has been observed to be significantly higher in the field.

One further aspect of the model identified as a point for consideration by the expert panel pertains to the minimum distance allowed between two males if they are to successfully form a *lek*. As mentioned in §8.2.3, in cases where males are few in number and widely spread (typically times of low infestation), it is still possible for *leks* to be formed, although, when possible, it is preferable for proximity to be close between *lek* contributors. This requires a dynamic assessment of the level of infestation by a male before a *lek* is initiated in order to establish the tolerance within which other males must reside in order to be considered contributing to the *lek*.

In order to simulate this phenomenon, a detailed study is required so as to establish a system of rules or conditions under which the minimum distance threshold between males contributing to a common *lek* may increase or decrease, or even exist as a dynamic parameter dependent on the state of the agent’s surrounding. This correlation between population density and mean individual fitness of a species is called the Allee effect [51, 83]. Such a detailed implementation falls outside the scope of this study and, in turn, requires further entomological insight and research in order to quantify and understand the driving factors of the process.

It was therefore decided to implement a simplified approach in this study which increases the likelihood of male agents moving over larger distances in an attempt to cluster and, in turn, form *leks* during times of low infestation. This approach required the current infestation level of the field to be determined in order to cue the male’s actions. This infestation level is determined using a function **F** `FieldDensityAssessment` which determines the number of agents currently existing in the simulated area who reside in the respective `AdultMale` or `AdultFemale` state. This total is considered in conjunction with the number of sugarcane stalks in order to return an $e/100s$ value for the area, indicating the adult population density. Originally, this adult density measure was determined for the entire simulated hectare space, but this was subsequently changed to a per-field value in light of the inclusion of GIS shapefiles as discussed in the following chapter. The infestation level of the male’s current field in which it resides then informs the male of the density of the surrounding population of adult agents and, furthermore, its ability to form *leks* and mate.

In cases where lower infestation levels are observed, the clustering algorithm which called at a certain rate by males during the `LookingforLeks` state is altered to increase the frequency with which males attempt to move toward their closest potential fellow *lek* contributors. Furthermore, the probability threshold imposed on this action is also increased during low infestations to make clustering between adult males more likely. This, in turn, results in more active movement of males towards one another in times of low infestation, allowing *leks* to form more easily and thus preventing populations from dying out, particularly during model initiation.

Finally, movement of adult *E. saccharina* agents in the simulation during daylight was reduced in frequency in light of insight that the moths typically only move during the day if disturbed by predators [55, 72]

8.3.4 Functionality assessment and comparison with previous performance

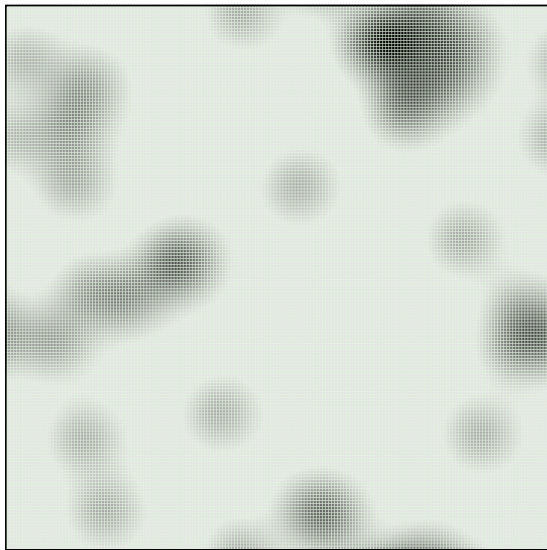
In order to assess the success with which the new knowledge acquired during the expert discussion was implemented and changed in the simulation model, a number of experiments similar to those conducted in §6.6 were carried out. These experiments also aimed at eliciting the differences between the newly implemented, more biologically accurate oviposition process when compared to the previously adopted algorithms. As in the previous cases, the initial conditions applied to these experiments consisted of a small population of *E. saccharina* eggs concentrated into a $10m \times 10m$ square in the centre of the simulated hectare. In these experiments, however, owing to the higher mortality rates implemented in the different life stages of the model, a larger starting population was required in order for sufficient agents to mature through to their respective adult stages and initiate the growth of the population. In the previously conducted tests, the model was populated with 15 male and 15 female agents initially. Through experimentation, it was determined that a starting population of 120 male and 120 female agents in conjunction with the higher mortality rates resulted in the fostering of an adult population of a similar size to those previously encountered. As was the case in the previous experiments, the rate of incoming agents entering the simulation space from the hypothetical surrounding sugarcane areas was assumed to be zero. Experiments were once again conducted over the same selected dates as those in §6.6, lasting six months in total per experiment. All sugarcane stalks present in the simulation were assumed to possess 20 internodes at full growth.

Other than providing a comparison of the dispersal of the population between the different oviposition algorithms, the experiments also provided a means to test the degree to which the population favours physiologically aged sugarcane. This was facilitated by the incorporation of a button which, when selected, executes an algorithm that converts the physiological age scale value of each sugarcane stalk (ranging from 1 to 10) to a coloured box (where stalks of physiological age 1 are white, ranging to physiological age 10 stalks which are black). In doing so, this action proceeds to convey the physiological sugarcane age profile of the simulated hectare of sugarcane to the user, overlaid on the simulation space. The creation and incorporation of this number of presentation shapes, however, consumes a significant amount of memory during each simulation run, slowing down the execution of the experiment. For this reason, the visibility of the physiological sugarcane profile should only be activated intermittently for assessment purposes and then once again be disabled by reselecting the button, in order to allow primary memory reallocation to the execution of the moth's behaviour.

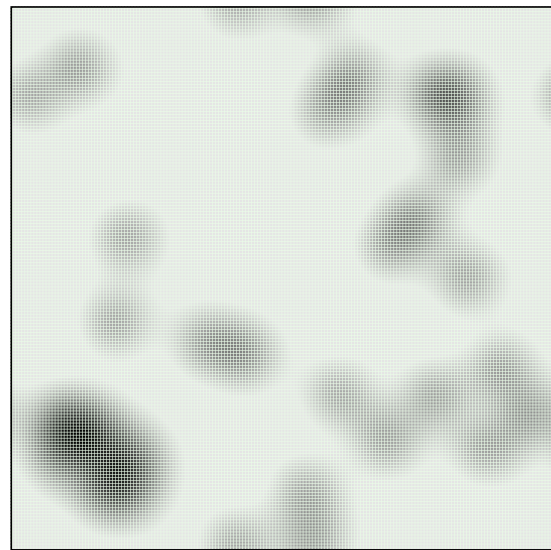
A selection of the results generated by the experiments are documented in this section. These results illustrate notable aspects of the modified model which are further discussed.

As discussed in §8.3.2, a randomised physiological profile is generated for each simulation run. This means the initial population may be located in 'good' or 'poor' areas in terms of development (relating to the physiological age of the sugarcane in which it resides), depending on the randomly generated profile. It was observed that in some cases, examples of which are shown in Figure 8.5, populations failed to establish themselves and break out in a notable manner, lasting only two or three generations before dying out completely. This is attributed to the initial population being located in the centre of the simulated hectare. In the cases illustrated in Figures 8.5(a)–8.5(b), the central area is occupied by sugarcane which is notably healthy and not of an advanced physiological age. Although populations of *E. saccharina* are able to survive in such sugarcane, the model is constructed in such a manner that agents residing in these areas experience very high predator mortality rates and development is slightly slower during the feeding stage, based on the recommendations from the expert panel discussion previously mentioned. In light of this, it follows that, for such a small initial population, cases of insuffi-

cient mating instances to overcome the high mortality rates may occur and the population is, thus, prone to extinction. It is, however, possible in some experiments that adult males may sufficiently disperse, moving into more suitable sugarcane (as is the case in nature [72]), but this does not always happen in time, nor does it occur predictably and definitively (for reasons explained previously). This means that the effect of dispersal is not always sufficiently pronounced in a short enough space of time to reduce mortality rates, resulting in the population failing to progress.



(a) Underlying sugarcane physiological age profile in an example of an unsuccessful experiment.



(b) Underlying sugarcane physiological age profile in a second example of an unsuccessful experiment.

FIGURE 8.5: Randomised physiological sugarcane age profiles which led to population extinction owing to insufficient mating instances occurring to combat increased mortality rates experienced in healthy areas of sugarcane in the centre of the simulation space.

In cases where the initial population is located in or near to stressed sugarcane, however, populations were observed to thrive and develop at very high rates owing to the reduced mortality rates, as well as the dense population of male and female agents, further promoting mating and oviposition. Unlike the previous **F** `MovingWindow` and **F** `FindMatureCane` oviposition algorithms, which included a number of user-changeable parameters, the **F** `FindOvipositionLocation` oviposition algorithm is far simpler and based on the expert panel insight which suggests that there is no formalised process of oviposition site selection, other than an attempt by the female to remain and oviposit near to the site of mating. This means that there are not as many modifiable parameters for significantly altering the execution of the algorithm and the corresponding dispersal of eggs. This is sensible, since it is firmly believed that dispersal of the population is more strongly influenced by males moving to sites of stressed sugarcane, as modelled by the **F** `SeekOutPhysiologicallyAgedCane` function.

One aspect of the **F** `FindOvipositionLocation` algorithm that may be changed is the probability with which a female will select an unoccupied oviposition site. As detailed in §8.3.1, the female considers vacant internode buds on sugarcane stalks to be suitable sites for oviposition. If no such vacant sites are, however, discovered in close proximity to the site of mating, the algorithm causes the female to expand the search area to further areas of sugarcane. In order to mimic the process more effectively, a probability threshold is included in the algorithm in

an effort to reproduce the more natural behaviour where not each and every site is effectively discovered and occupied, but rather, for no specific reason, a female may choose not to oviposit in a particular site as would probably be the case in nature. When a vacant mating site is discovered, a random sample is made and, if the probability threshold is overcome by the random sample, the site is selected. If not, the female does not make use of the vacant site and continues searching. It was anticipated that decreasing the probability would result in greater search areas requiring consideration by the female and, as a result thereof, more dispersal of eggs during oviposition. This phenomenon is, however, not expected to be apparent on a large scale. Owing to 129 600 stalks being included in the simulated hectare, movement between stalks by a single female during oviposition is not likely to be perceptible on a macro scale in the simulation model. Evidence of greater dispersal, however, is expected to be noticeable in the size of the population within the established cohorts of larvae. This follows in view of the fact that a small increase in dispersal of the eggs results in a greater possibility of larvae discovering vacant stalks into which to bore and, as a result, lower mortality in the *Larvae* state by virtue of the effect of carrying capacity.

Figures 8.6–8.9 contain graphical representations of the results of experiments executed with increasing probabilities of site selection using the **F** `FindOvipositionLocation` oviposition algorithm. In each example, the underlying physiological profile of the sugarcane is also shown so as to facilitate assessment of the degree to which the male agents' movement causes the population gravitate toward areas which are more physiologically aged. The results of each experiment are discussed and certain attributes of the revised process are highlighted in the context in which they affected the progression of the experiments. As may be seen, the population densities are recorded and reported at the outset of the experiment, as well as two months into the experiment and at the end of the six-month experiment. The population size after two months represents the first cohort of larvae which result from mating interactions between the initial population after experiment initiation and therefore provide insight into how the resulting population is established. This population then progresses through a number of further generations (details of which are omitted for conciseness) until the final cohort of larvae in the six-month experiment is recorded and shown. This provides insight into how the population is fostered and progresses, as well as whether dispersal toward areas of physiological sugarcane is evident. In both cases, owing to the maturation rates of *E. saccharina*, the population comprises almost entirely of agents in the *Larvae* state. This means a more realistic indication of the population size and dispersal is reported on, since the large mortality occurring in the preceding states has already taken effect in the population and, as such, no inflated population sizes, suggesting inaccurate infestation values, are mistakenly reported.

In the first experiment, shown in Figure 8.6, the probability of oviposition site selection by the female in each case is 20%. This value is expected to result in greater dispersal within the population. As may be seen in Figure 8.6(a), the initial population location is very near to a physiologically aged patch of sugarcane and, as such, the population is expected to disperse towards and thrive in this area.

From the initial cohort, pictured in Figure 8.6(c), it may be seen that not much dispersal has occurred at this stage of the experiment, but that the population has grown significantly in number. This is attributed, as mentioned, to a probable decrease in the effect of the carrying capacity of the sugarcane. In comparison with Figures 8.7(c), 8.8(c) and 8.9(c), all of which employ a higher probability of site selection, the population numbers of the populations are greater. This effect is also more pronounced by virtue of the initial population being located within physiologically aged sugarcane which has a decreased larvae mortality, as discussed in §8.3.2. In Figure 8.6(d), the population has grown significantly at the conclusion of the exper-

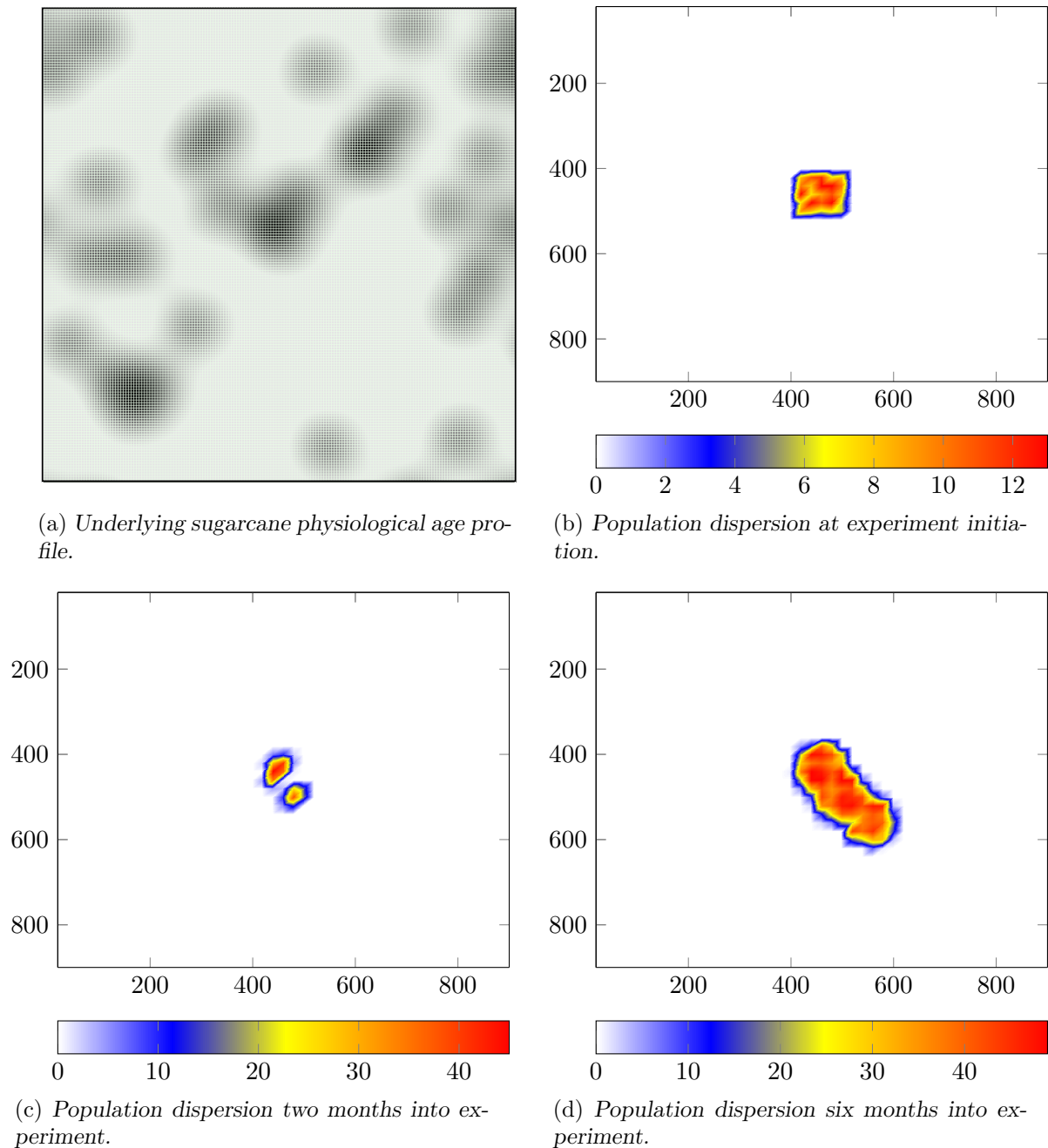


FIGURE 8.6: Population dispersion observed upon the application of the revised oviposition site selection algorithm, adopting a very low probability of site selection.

iment and has further notably infested the area of physiologically aged sugarcane. That said, dispersal toward the bottom right corner has also occurred. This may be attributed to crowding which occurred in the high-quality sugarcane, forcing female agents to disperse on a larger scale in search of oviposition sites. This, in turn, may have also been propagated by the low probability of oviposition site selection. It is, however, important to note that, in all cases, a number of random behaviour procedures continuously occur in the model and, as such, not all resulting activity may be definitively attributed to specific aspects of the simulation model, but may rather vary on a case-by-case basis.

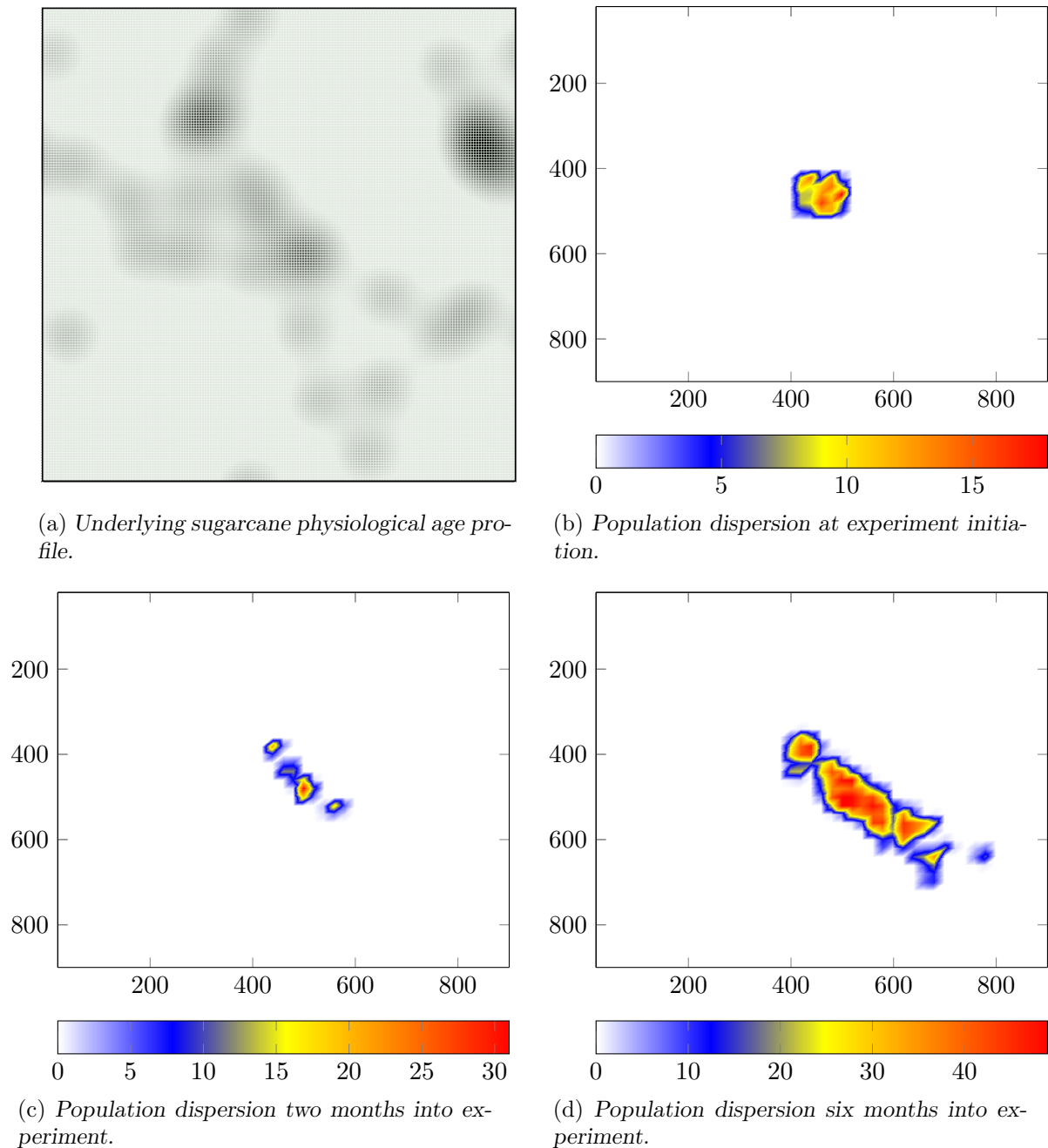


FIGURE 8.7: Population dispersion observed upon the application of the revised oviposition site selection algorithm, adopting a low probability of site selection.

Figure 8.7 shows the result of a similar experiment where the probability of oviposition site selection by the female was chosen to be 40%. The underlying physiological sugarcane profile in this experiment is also shown in Figure 8.7(b). As may be seen in Figure 8.7(c), the population numbers are slightly lower than those recorded in the previous experiment. This may be as a result of a more pronounced effect of the carrying capacity of sugarcane owing to less small-scale dispersal during oviposition site selection, as well as slightly less stressed sugarcane in the vicinity of the initial population, increasing larvae mortality during the stalk selection process. As may be seen in Figure 8.7(d), the population ended up dispersing significantly, following the physiologically aged sugarcane quite notably. The population is also somewhat more fragmented

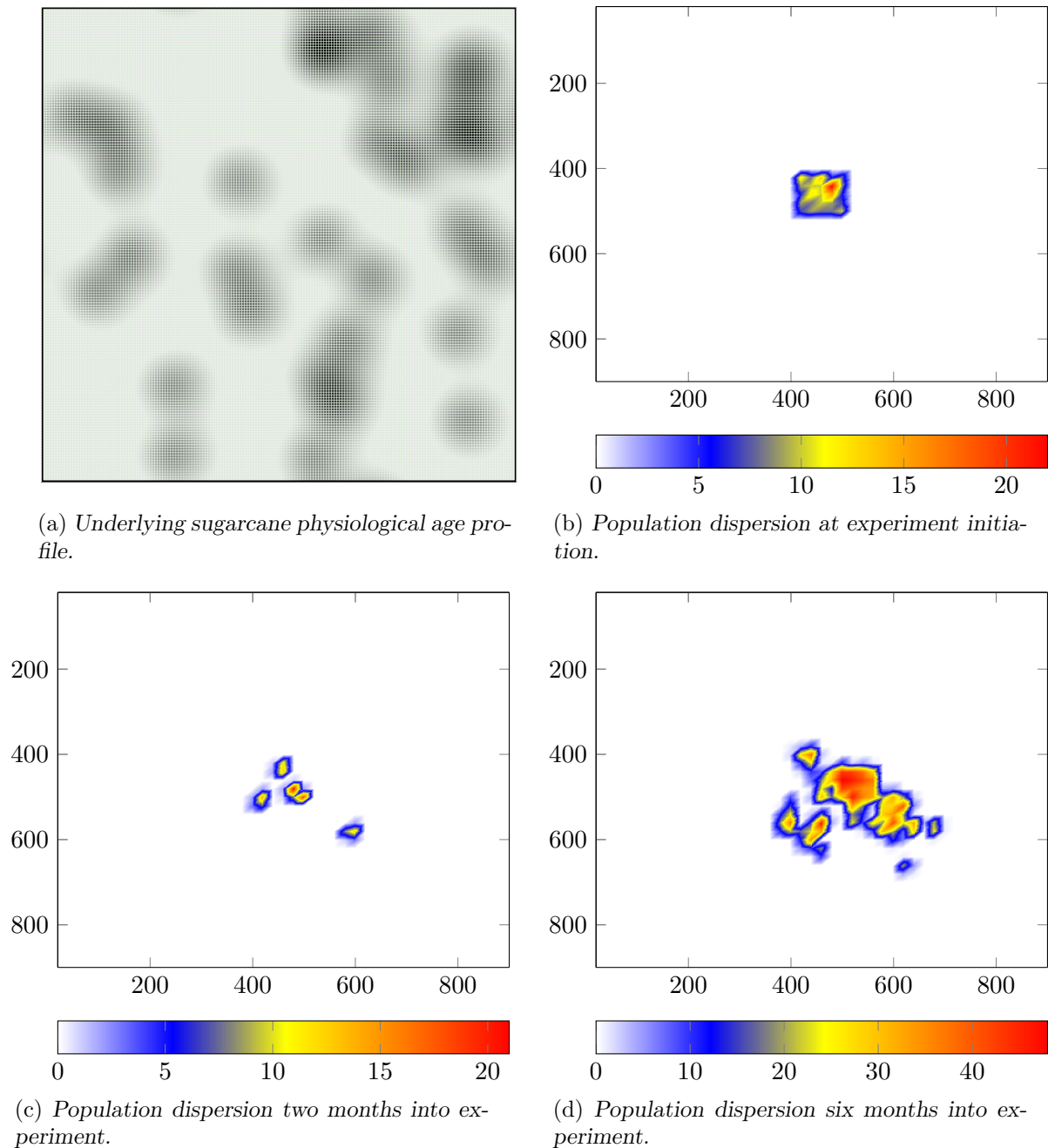


FIGURE 8.8: Population dispersion observed upon the application of the revised oviposition site selection algorithm, adopting a normal probability of site selection.

than in the previous case, probably as a result of various suitable habitation sites and, as such, split dispersal by male adult agents.

A similar phenomenon may be seen in Figure 8.8. In this experiment, the probability of oviposition site selection was 60%¹. As may be seen in Figure 8.8(d), several smaller populations exist within the simulation space in light of the various stressed areas, pictured in Figure 8.8(a), to which male agents dispersed. The initial population growth, shown in Figure 8.8(c), is similar to that of Figure 8.7(c), with slightly lower population numbers observed, but over a greater

¹This is the value with which the algorithm is populated by default upon model execution.

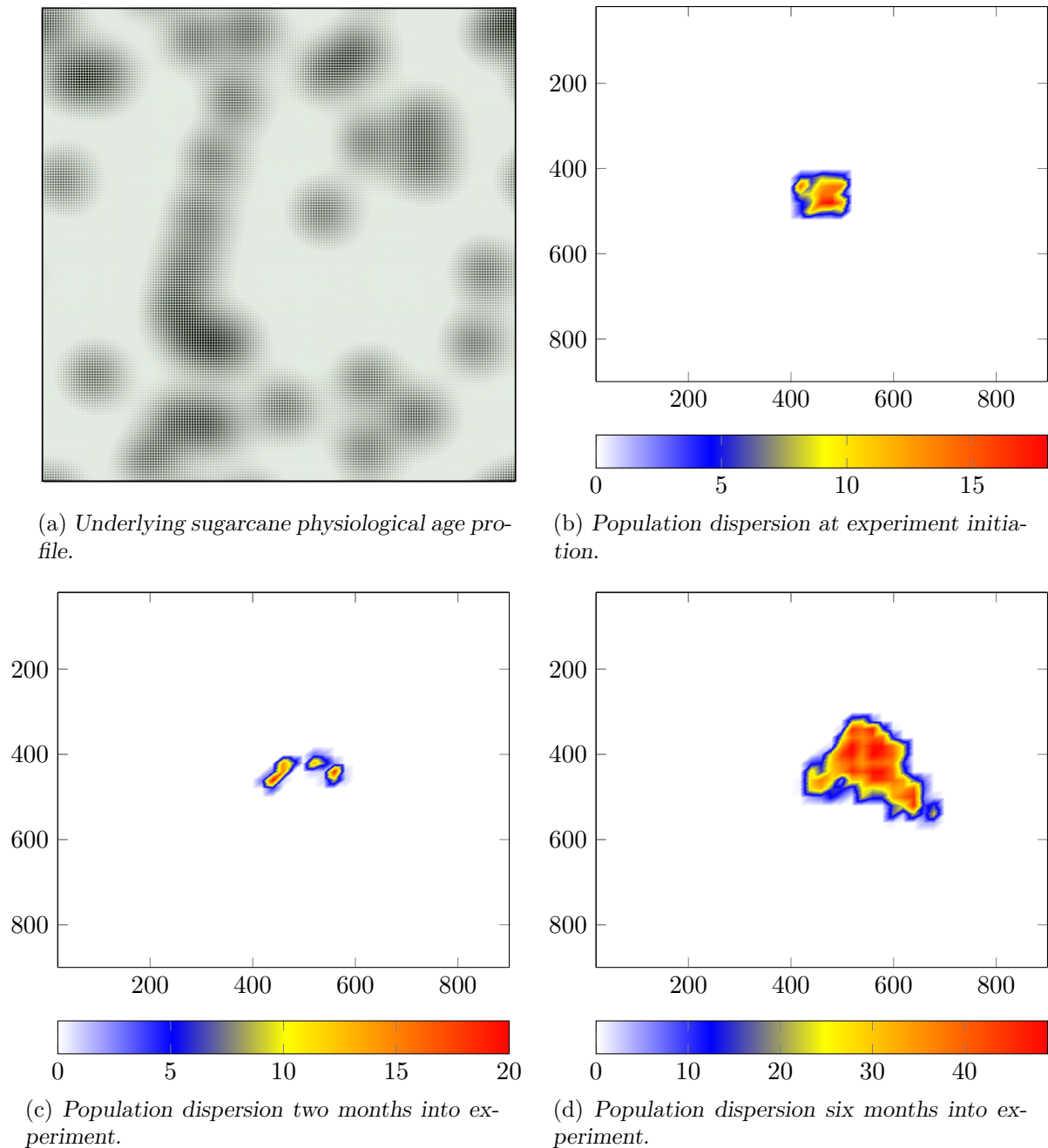


FIGURE 8.9: Population dispersion observed upon the application of the revised oviposition site selection algorithm, adopting a high probability of site selection.

number of small cohorts. This suggests that dispersal occurred at a greater rate during the early stages of this experiment in comparison to the previous cases.

In the final experiment, the probability of oviposition site selection by the female was set to 80% and the results are shown in Figure 8.9. As may be seen in Figure 8.9(a), the initial population site is surrounded by physiologically aged sugarcane. Following this, Figure 8.9(c) shows that early dispersal favoured the stressed cane to the top right of the initial population. In this figure, population numbers are lower than in all previous cases, possibly owing to the more pronounced effect of carrying capacity since females select almost all available oviposition sites, restricting

small-scale dispersal. As is seen in Figure 8.9(d), the population favoured the area to which it originally dispersed and no other factions of agents formed a breakaway infestation in the nearby physiologically aged areas. This is a typical example of the effect of randomness in the model and is not directly attributed to any particular model aspect. The phenomenon may, however, have been intensified by the decreased dispersal during oviposition, but it is unlikely that this attribute had a controlling effect.

In all of the above-mentioned cases, the resulting populations mimic, with greater accuracy, the observations by experts in the field. Populations aim to remain in areas of physiologically aged sugarcane, dispersing only as a result of overcrowding or in search of physiologically aged sugarcane. This results generally in reaction-diffusion-like dispersal, consisting of a dense central point and dispersing outwards in terms of population numbers. This corresponds largely with the results of diffusion-type models such as those employed by Potgieter *et al.* [65]. The distance of agents from the initial population site after six months also corresponds to expert observations who commented that the species is not known to be a far-flier. This, of course, excludes the approximate 10% of the female population which is genetically prone to flying greater distances. These agents are not reflected in the results as they departed from the simulation space during the course of the experiment execution.

In general, the results do not correspond strongly with the observations of Berry *et al.* [13] who reported equal infestation levels across an experimentation site. It should, however, be noted that the experimental timeline was particularly short and that the subsequent infestation outbreaks caused by far flying females are not reflected in these results, a phenomenon to which Berry did attribute some of the boundary infestation levels recorded in [13]. In cases such as those reported in Figure 8.8(d), however, the different levels of infestation between adjacent stalks or areas do appear to agree more with Berry's observations, who commented that the dispersal of *E. saccharina* resulted in a clustered arrangement of high and low areas of infestation across a field.

Overall, the newly implemented oviposition algorithm, in conjunction with the role of physiological sugarcane age in the dispersal process is believed to more accurately mimic the process as it is currently understood to occur in nature. Further research is required in order to comprehensively understand the process, suggestions as to which are made later in this dissertation. Such improvements to the simulation process may only occur in light of further entomological knowledge and understanding.

8.4 Chapter summary

In this chapter, the Delphi technique was reviewed as a method for finding consensus among experts pertaining to topics which are not thoroughly understood, or documented in the literature. An application of a revised version of this method was then applied in the context of an expert panel discussion whereby entomologists gathered to critique and refine the agent-based model of *E. saccharina* of Chapters 4–7 in an attempt to improve its accuracy and realism. The results of these discussions were recorded, with specific reference to areas of the simulation model where aspects of the pest's behaviour were incorrectly implemented and where further in-field observations had led to increased insight so as to allow for refinement of the model. The discussed changes were then implemented in the model and tested so as to determine the degree to which the improvements agree with the recommendations of the expert panel. The changes were found to reflect the biological processes more accurately as they are currently understood, whilst still retaining the similarities to previous studies in terms of resulting population dispersal — part of which was captured by the previously implemented algorithms.

CHAPTER 9

GIS map facilitation

Contents

9.1	Motivation	191
9.2	Importing shapefiles into ANYLOGIC	192
	9.2.1 <i>Description of the full sugarcane farm shape importation process</i>	193
	9.2.2 <i>Description of the experiment site selection process</i>	198
	9.2.3 <i>Description of the selected site importation into the model</i>	198
9.3	Functionality assessment of the GIS importation process	208
9.4	Implementation of GIS features into simulation execution	210
	9.4.1 <i>Functional assessment of the sugarcane attribute attribution process</i>	210
	9.4.2 <i>Attributing specific characteristics of sugarcane stalks to each field</i>	213
	9.4.3 <i>Incorporation of emitting and absorbing boundaries</i>	216
	9.4.4 <i>Functionality assessment</i>	219
	9.4.5 <i>Revised decision-support interface</i>	228
9.5	Chapter summary	229

In this chapter, the motivation is presented for the inclusion of a module in the simulation model of Chapters 4–8 which allows for the importation of GIS shapefiles and their associated attributes. This is followed by an explanation of the approach employed. The newly developed algorithm is discussed in detail, including the various aspects which required consideration during its development. The output of the importation method is then tested in order to determine the accuracy with which it was implemented. Following this, the manner in which characteristics are attributed to sugarcane are discussed and tested, as well as changes to the boundary behaviour of the simulation model in light of the GIS map inclusion. The entire module is then tested in conjunction with the rest of the simulation model to illustrate its diversity and applicability.

9.1 Motivation

In Chapter 6, an in-depth parameter variation analysis was conducted in order to illustrate the capacity and flexibility of the simulation model of Chapters 4–8, as well as its ability to recreate and simulate desirable population dynamics, as per user requirements. By selecting the user-defined inputs in differing arrangements, the consequent behaviour and development of the *E. saccharina* population was shown to be effectively manipulated. This is of notable importance because farmers and researchers who may make use of the decision-support tool are likely to

experience differing behavioural traits and population dynamics of *E. saccharina*, depending on the location of the sugarcane crop under consideration. Climatic factors, soil characteristics, and sugarcane age and variety have all been identified to alter both *E. saccharina*'s response to sugarcane as a habitat [90], as well as the subsequent development of a population within that sugarcane.

It follows that, should a farmer or researcher wish to simulate the behaviour of a particular population of *E. saccharina* whose behaviour is attributed to a particular habitat or farm location, so the model should be able to facilitate the specific attributes of the area, such as field shapes, sugarcane varieties and differing ages of fields. Such information about specific sugarcane farms typically exists in geographic shapefiles. A shapefile stores non-topological geometry, as well as attribute information, pertaining to the spatial features in a data set. A feature's geometry is typically stored as a shape comprising a set of vector coordinates. Areas within shapefiles are represented as closed loop polygonal shapes and each attribute record has a one-to-one relationship with the associated shape record [85]. Shapefiles can typically be imported into *Geographic Information System* (GIS) or simulation software for use in analysis or modelling, allowing for the real-life shape and accurately scaled size of its contents, as well as the attributive information pertaining to the area, to be inspected and appropriately used.

9.2 Importing shapefiles into ANYLOGIC

The ANYLOGIC simulation software offers functionality that accommodates the direct importation of shapefiles into a simulation model [94]. That said, such direct importation results in a static image file which is not interactive with the model or user and simply exists as an underlying 'background' representing the shape of the area under investigation. In the simulation model of Chapters 4–8, dynamic interaction with individual sugarcane fields detailed in the shapefile is required in order to attribute individual sugarcane stalks to the appropriate areas within the simulation space. This means the Cartesian coordinates of the individual field boundaries have to be accessible in the model. This, in turn, allows for the identification of the specific areas where certain sugarcane stalks should be generated in order to execute a geographically accurate simulation run. A novel algorithm was therefore developed for importing the GIS information into the simulation model and still facilitate interactive, dynamic accessibility post-importation.

This approach was advocated by Schumann [77], a world-leading simulation expert who performed similar work in applications bound to the use of shapefiles in the ANYLOGIC simulation software and, in turn, required a novel means of overcoming the aforementioned shortcomings of using the software. He programmatically extracted details of the shapefiles in order to access them in an ANYLOGIC environment for the purpose of simulating unmanned aircraft routing along road networks [78]. His expertise was therefore sought regarding the design of the importation process which is discussed in this chapter.

The process of importing GIS features into the simulation model in order to execute geographically specific simulation experiments comprises three parts, namely the full sugarcane farm shape importation, the experimental site selection process and the section importation of the experiment site into the model. Each of these steps occurs sequentially and requires user input. During the explanation of these processes, a number of naming conventions are employed in order to refer to specific components included in the importation algorithmic process. These naming conventions are defined in Figure 9.1. The *presentation area* refers to the $900\text{pix} \times 900\text{pix}$ area in which the simulation is run. This area has the capacity to simulate 129 600 sugarcane stalks and represents 1ha of farm area. The *reference field* is a 1ha square which is scaled relative to

the size of the imported sugarcane field and used to select a specific area upon which to execute the simulation. The elements contained within this square are extracted and scaled to fill the presentation area during model execution.

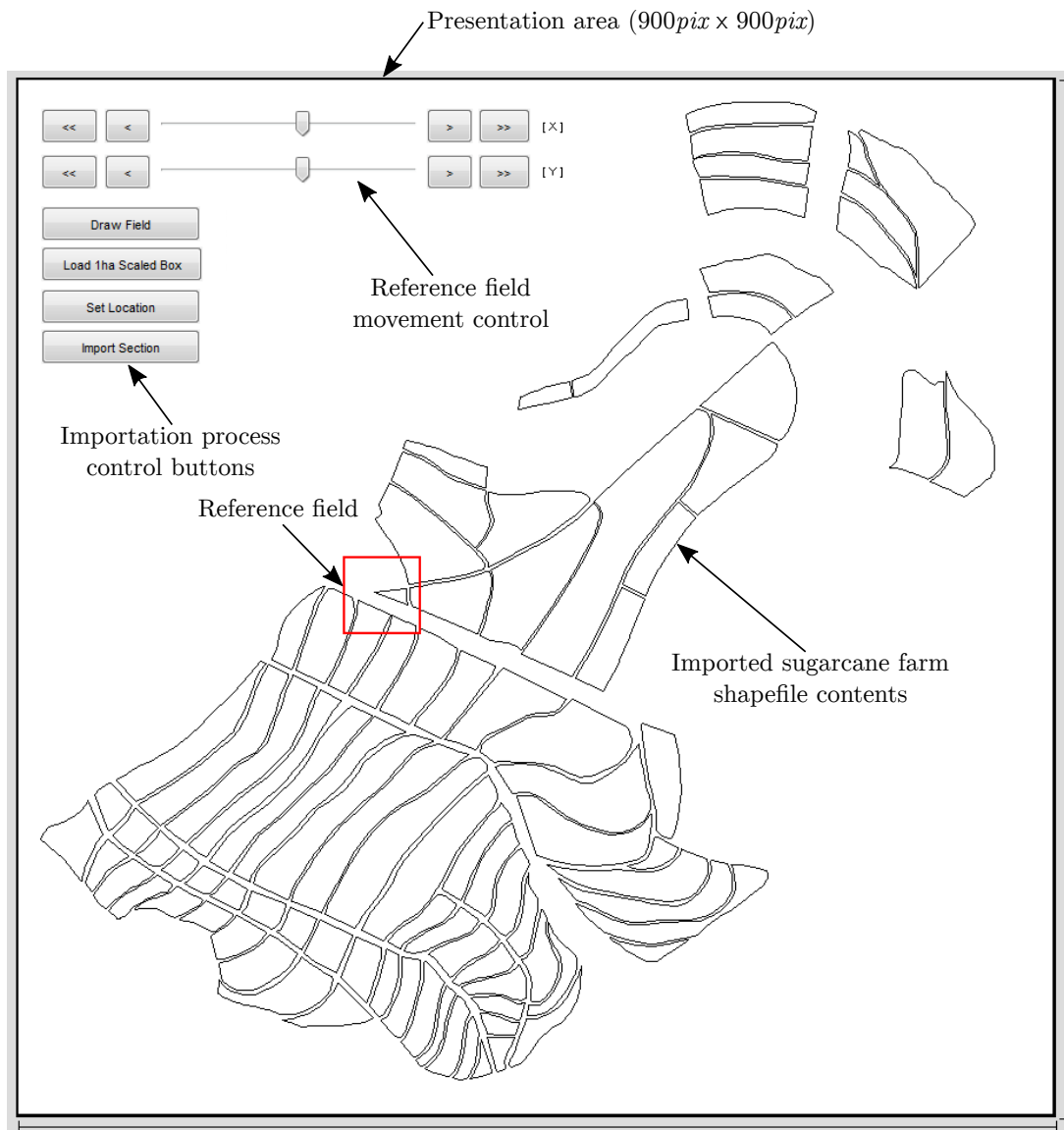


FIGURE 9.1: The naming conventions used in the GIS importation process.

9.2.1 Description of the full sugarcane farm shape importation process

The main steps in the full sugarcane farm importation process are shown in Figure 9.2 and the following explanations of each step are offered (using the same numbering as in the figure):

1. The information stored in the shapefiles (*.shp* and *.shx*) must first be extracted into a recognisable format for importation into ANYLOGIC. In this case, it was decided to make use of an EXCEL format (*.xlsx*). Extraction may be achieved by a range of freeware GIS softwares, as well as other open-source programming softwares, such as Python [70]. Two

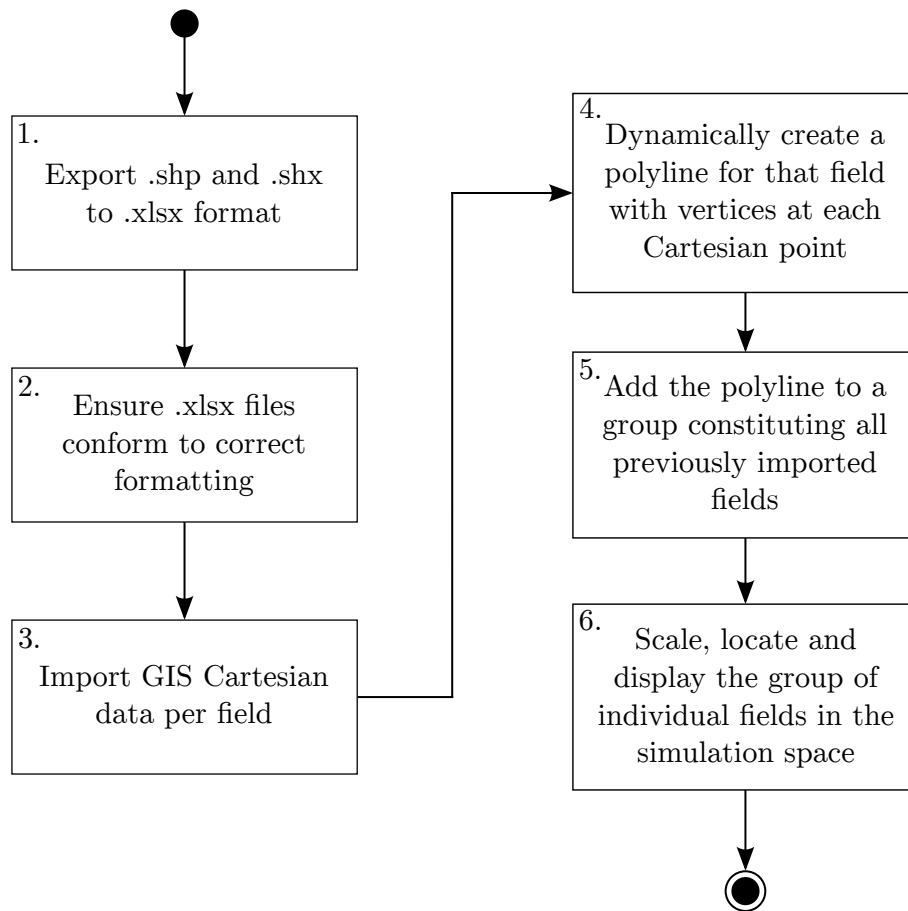


FIGURE 9.2: Flowchart describing the importation process of the sugarcane farm shapes.

files are required, namely the GIS shapepoints file, as well as the associated attribute file. The GIS shapepoints file contains the extracted GIS coordinates in *decimal degree* (DD) format, consisting of a series of x and y coordinates mapping each shape in the file. This is an alternative to using the common degrees, minutes and seconds format and is widely employed in GIS systems. Positive y coordinates indicate latitudes north of the equator, whilst latitudes south of the equator are represented by negative y values. This is similar for x values, with positive and negative x coordinates representing eastern and western longitudes relative to the Prime Meridian, respectively.

2. Once the relevant information has been extracted, it must be formatted correctly within the *Excel* file to ensure correct interpretation by the algorithms. The formatting of the two files required to facilitate the GIS importation process are shown in Figure 9.3 of the GIS shapepoints file is shown in Figure 9.3(a). As may be seen in the figure, no headings should exist in the file. Furthermore, the Cartesian coordinates for each field should be recorded in a single line, alternating between the x and y coordinates for each point in the shape. This data format may be achieved as the generic output from extraction using the Python software package. Importantly, the shape of each field may be defined by a differing number of point pairs. In light of this, the number of data entries in each row (represented by ‘n’ in the figure) will not necessarily be the same. Furthermore, the extracted separate arrays storing all of the separated x and associated y points will be of length $\frac{n}{2}$ owing to the coordinates being combined in a single line. The attribute file may include a range of details pertaining to the farm under consideration. In this case, the information provided

per field was stored in the format “*FieldNumber*”, “*Area*”, “*Perimeter*”, “*ID*”, “*Age*”, “*Variety*”, “*Internodes*”. The formatting of this file is illustrated in Figure 9.3(b). As may be seen in the figure, the headings are assumed to be stored in the first row of the sheet and the algorithm is designed to interpret the data according the aforementioned order of columned information.

	A	B	C	D	•	•	•	$n-1$	n
1	x_{11}	y_{11}	x_{12}	y_{12}	•	•	•	$x_{1\frac{n}{2}}$	$y_{1\frac{n}{2}}$
2	x_{21}	y_{21}	x_{22}	y_{22}	•	•	•	$x_{2\frac{n}{2}}$	$y_{2\frac{n}{2}}$
3	x_{31}	y_{31}	x_{32}	y_{32}	•	•	•	$x_{3\frac{n}{2}}$	$y_{3\frac{n}{2}}$
•	•	•	•	•				•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•				•	•
m	x_{m1}	y_{m1}	x_{m2}	y_{m2}	•	•	•	$x_{m\frac{n}{2}}$	$y_{m\frac{n}{2}}$

(a) The required format for GIS coordinates in the Excel file in order to be effectively imported into the simulation model using the automated process.

	A	B	C	D	E	F	G
1	FieldNumber	Area	Perimeter	ID	Age	Variety	Internodes
2	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
m	•	•	•	•	•	•	•

(b) The required format for attributes stored in the Excel file in order to be effectively imported into the simulation model using the automated process.

FIGURE 9.3: The required formatting of the two Excel files in order to effectively implement GIS field shapes into the model and populate them with the relevant information.

- Once these two files have been correctly established and linked to the simulation model, the fields are ready to be interpreted as presentation shapes. For each line in the GIS shapepoints file, the corresponding x and y coordinates are extracted into two separate arrays. An offset is established using the x and y coordinates of the first field in the shapepoints file. This offset assists in correctly locating the output shapes within the simulation model space. Since GIS coordinates may be sourced from anywhere in the world, the corresponding GIS values are indicative of their geographic location¹. By establishing an offset, the geographic location of the features represented in the shapefile is nullified and only the relative distances between points are considered by the algorithm.

¹South Africa’s largest sugarcane farming area, Kwa-Zulu Natal, is situated roughly at negative 30 and 31 decimal degrees of latitude and longitude, respectively.

4. Each x and y point pair is then plotted as the vertices of a polyline in the simulation model. A polyline is a presentation shape consisting of a variable number of vertices and, as such, can be used to contract a multi-sided feature in the simulation model [94]. Polylines are dynamically created as per necessity when recreating a sugarcane farm field layout. A polyline with the correct number of points, as determined by the length of the corresponding x and y arrays, is added to the presentation space and each point is set according to the data drawn from the GIS shapepoints file. Each polyline is also coloured according to the current age of the sugarcane, as indicated by the “Age” column in the attribute file. The scale of colours used depends on the user’s selection of the crop cycle (12 months or 24 months) under the settings tab prior to the simulation execution.
5. In order to ensure that the collection of dynamically created polylines are all located relative to the same origin point, as well as remain on the same presentation layer within the simulation, the creation of each field is concluded by its addition to a presentation group called ‘Fields’. A group serves as a collector of elements and keeps track of all the fields created. Furthermore, changes may be imposed to a group as a whole, thereby affecting all of the shapes which compose it. For example, if the origin point of a group is changed, the relative origin from where each of its composing fields are drawn will also be changed.
6. Once the data of all fields stored in the GIS shapepoints file have been extracted and a polyline has been created to map each field, the entire ‘Fields’ group is dynamically scaled according to its contents so as to ensure that it can be wholly presented to the user within the $900pix \times 900pix$ presentation area. This is achieved by recording both the highest and lowest x and y values encountered during the importation process. These maximum x and y distances are then considered and, using the larger of the two, a scale is generated which results in that constraining distance being appropriately scaled in order to fit within a $900pix \times 900pix$ presentation area restriction. The same scale is then applied to the entire system of polylines to ensure that no horizontal or vertical distance between any two points exceeds the $900pix$ constraint.

The system of scaled polylines is then also dynamically centered within the presentation area. This is achieved following the process illustrated in Figure 9.4, where x_{min} and y_{min} refer to the previously determined minimum x and y values encountered during the importation process respectively, x_{max} and y_{max} refer to the maximum x and y values encountered during the importation process, respectively, H_{shift} refers to a horizontal shift incremental counter, and V_{shift} refers to a vertical shift incremental counter.

As may be seen in the figure, this part of the algorithm increments or decrements the H_{shift} and V_{shift} values continually until the extreme points within the polyline group fall within the presentation space and, by extension, all points are then effectively scaled and located to complete the importation process. As mentioned, the scaling process conducted previously ensures that it is possible to find such a centroid point around which to arrange fields so that no fields exceed the presentation area limits during initial importation.

This concludes the initial importation division of the algorithm. Once this division of the algorithm has been executed, the user should be presented with a mark-up sugarcane farm mimicking the contents of the shapefile which was processed by the simulation model. The entire process was designed to remain generic and, as such, is capable of successfully importing geographical information of any sugarcane farm and effectively presenting the result to the user.

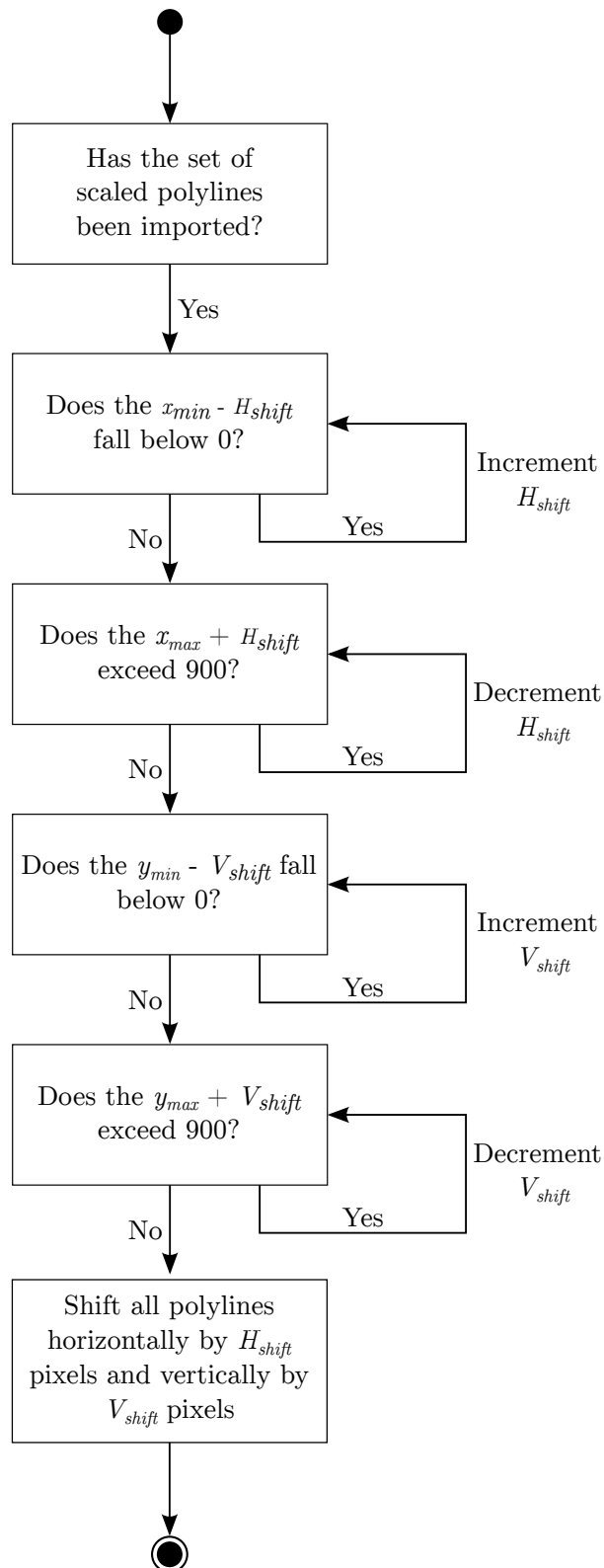


FIGURE 9.4: Flowchart describing the process followed in order to shift the polyline set sufficiently vertically and horizontally so that all presentable points fall within the $900\text{pix} \times 900\text{pix}$ presentation area.

9.2.2 Description of the experiment site selection process

Once the sugarcane farm under consideration has been imported into the simulation model, the model user is required to choose a one hectare-sized area from the farm on which the simulation should be executed. In order to achieve this, a reference field of size $100m \times 100m$ relative to the imported sugarcane field must be generated. It follows that, once scaled to fit the $900pix \times 900pix$ presentation area, the relative size of the fields will differ between different farms and geographic locations and the size of $1ha$ relative to the rest of the farm will depend on the shapefile which was imported.

The algorithm makes use of a subfunction which is responsible for determining the relative size of a $1ha$ reference field in relation to the imported fields. This is achieved by comparing the scaled length of the polyline mapping of one of the fields imported during the previous step (*i.e.* the length of the polyline which is seen by the user) with the actual perimeter of that same field as recorded in the attribute file. Using these two values, a scale is established to determine the relative size of $1m$ in the presentation space.

An appropriately scaled $1ha$ ($100m \times 100m$) reference field is consequently placed into the presentation space and the user is able to place this field anywhere within the imported sugarcane farm. As such, one entire small field may be captured, or portions of several fields and the surrounding pathways between them may be selected. Anything captured within the borders of the reference field will ultimately be imported into the final $1ha$ simulation space (including fields of different age and variety sugarcane, as well as paths and fire-breaks).

Once the location of the reference field has been finalised by the user, an extraction process is carried out to determine which fields must be included in the simulation. This division of the algorithm considers each point inside the reference field square and determines whether or not it is contained in any of the polylines encircling individual fields. If a point within the square is found to be located in a field, that field's number is saved for future reference. Once all points within the square have been assessed, the result is saved as a vector storing the field numbers of all fields which are partly or fully located inside the boundaries of the reference field.

9.2.3 Description of the selected site importation into the model

Once the impacted fields have been successfully extracted according to the process described in §9.2.2, the corresponding elements captured in the reference field are scaled and placed to occupy the entire $900pix \times 900pix$ presentation area. The selection importation process is similar to that of the initial importation process, detailed in §9.2.1, but only involves the particular fields identified during extraction. Furthermore, since the reference field may capture small parts of a number of fields, with the remainder of the field continuing outside of the reference field boundary, the corresponding polyline shapes have to be trimmed to fit the presentation area exactly.

In terms of functionality, the purpose of importing the fields as polyline shape elements is to allow for assigning characteristics to the **V Sugarcane** array points, as described in §4.3.2. By extracting the individual field GIS data from the shapefiles and incorporating them into the model as dynamic, interactive presentation elements, the location of each corresponding **V Sugarcane** array point may be assessed to determine within which field it falls and, as such, which properties it should possess. In order to achieve this outcome, a simple comparison similar to that between the points within the reference field and the polylines discussed in §9.2.2 is employed.

Once the underlying sugarcane field has been established, however, the purpose of the polyline shapes is not exclusively to demarcate the underlying areas visually for the user's benefit, but is also required to control the inflow and outflow of agents at the boundaries of the simulation area, outside of which the rest of the sugarcane farm is assumed to still exist. It is therefore important for the point adjustment and trimming process to occur on the polylines that are included in the *1ha* simulation. The manner in which the inflow and outflow of agents at the simulation boundaries is incorporated is discussed at a later stage.

The steps followed to import what is selected by the positioning of the reference field, as well as the process followed to trim the included fields effectively to the boundaries of the presentation area, is summarised in Figure 9.5. The following descriptions are offered for each step in the algorithm (using the same numbering as in the figure):

1. The process begins by accessing the array containing the field numbers of all fields that have to be represented in the presentation area by virtue of their inclusion in the reference field site selection by the user. This is a simple global parameter array and is accessed in the `Main` simulation environment.
2. The scale difference between the reference field and the presentation area is then determined. This factor is recorded in order to be applied to each of the fields that appear in the final simulation space. This is in light of the fact that the reference field size represents *1ha* relative to the actual size of the fields represented in the initial importation process and, as such, if it is to be enlarged to fit the presentation area, so the shapes captured should equally be enlarged to effectively duplicate the elements that it captures.
3. Each of the affected fields is then accessed once again in the GIS shapepoints file and saved into a separate array in the simulation model. The correct scaling and offsets are applied to the field points so as to correctly locate it in the same position relative to the presentation area as selected by the user when specifying the reference field.
4. Once the fields have been correctly placed relative to the presentation area, it follows that there will exist points in some fields which fall outside the boundaries of the *1ha* space. These will be fields which were not selected in their entirety by the user, but rather, only form part of a field included within the boundaries of the reference field. In such cases, outlying points must be altered and moved to the nearest boundary to ensure that the shapes do not extend past the presentation area boundaries. As mentioned, these boundaries are used to govern the inflow and outflow of agents within the executed simulation. Typical examples of where points may be located and corrected are shown in Figure 9.6.

As may be seen in the figures, a simple solution to correcting outlying points may appear to move them to the nearest point on the presentation boundary. In cases such as those illustrated in Figure 9.6(b), this is a good approach, since the nearest corner is the only logical point for relocation. If this approach is followed in cases such as those in Figure 9.6(a) and 9.6(c), however, inaccuracies may occur in the resulting trimmed field.

A hypothetical example of an imported field shape which possesses points inside and outside the presentation area is shown in Figure 9.7(a). As may be seen in the figure, points 1, 11, 10 and 7 are required to be moved to the boundary line in order to effectively trim the shape to fit within the presentation area. When simply keeping the x coordinate of these points fixed, setting the y coordinate equal to the boundary value generates an inaccurate result, as illustrated in Figure 9.7(b). As may be seen in the resulting field (coloured red), the alteration does not accurately map the portion of the field which

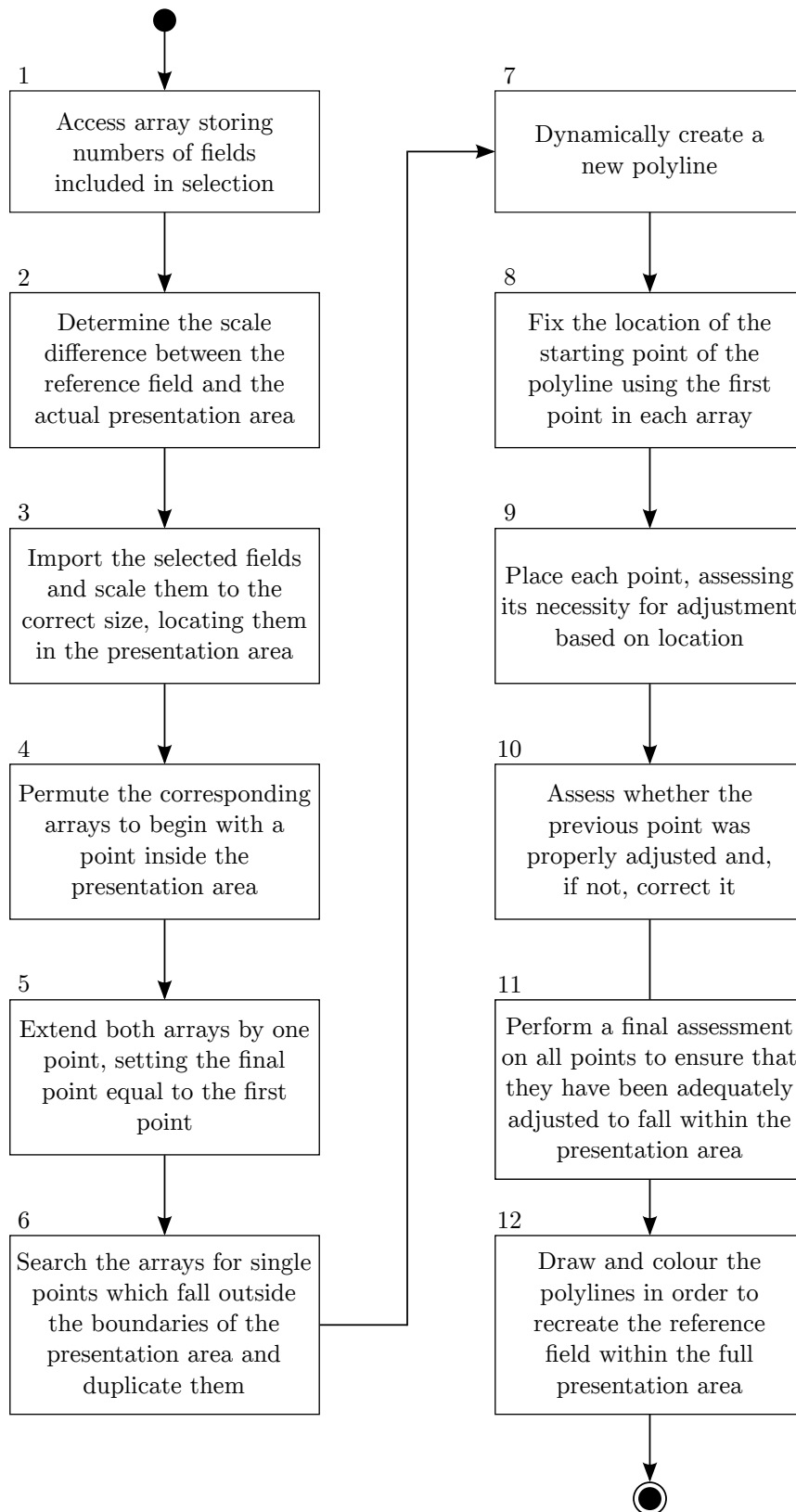


FIGURE 9.5: Flowchart describing the process followed in order to import the fields included in the reference field and then adjust the outlying points and scale them appropriately so as to fit exactly into the presentation area.

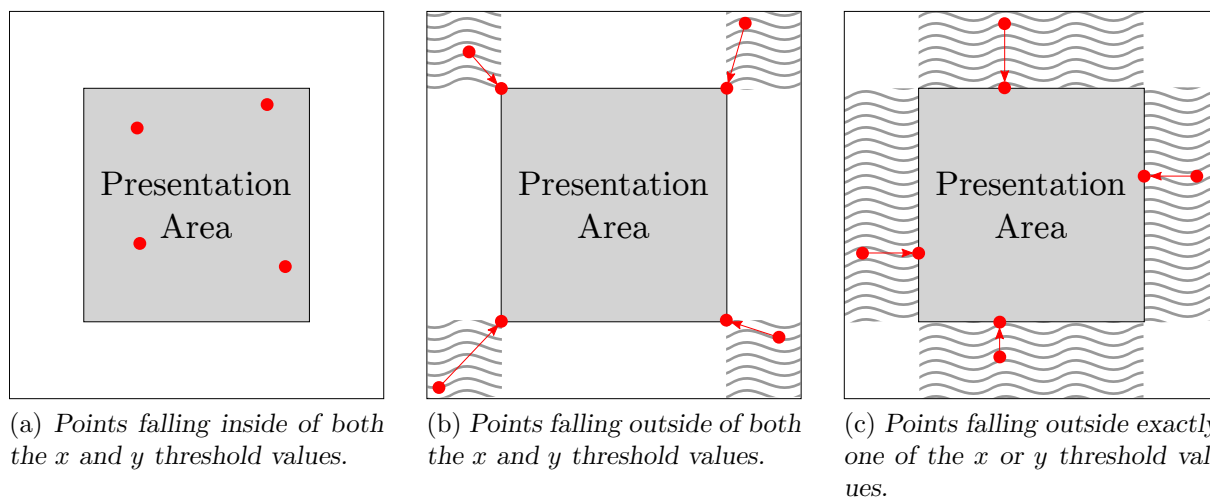


FIGURE 9.6: Examples of the possible locations where vertices of the imported fields may lie and, as such, require consideration during the adjustment process in order to move all field boundaries so as not to exceed the presentation area.

resides within the presentation area. For this reason, each vertex point in the polyline must be considered independently and effectively manipulated so as to accurately trim the polyline.

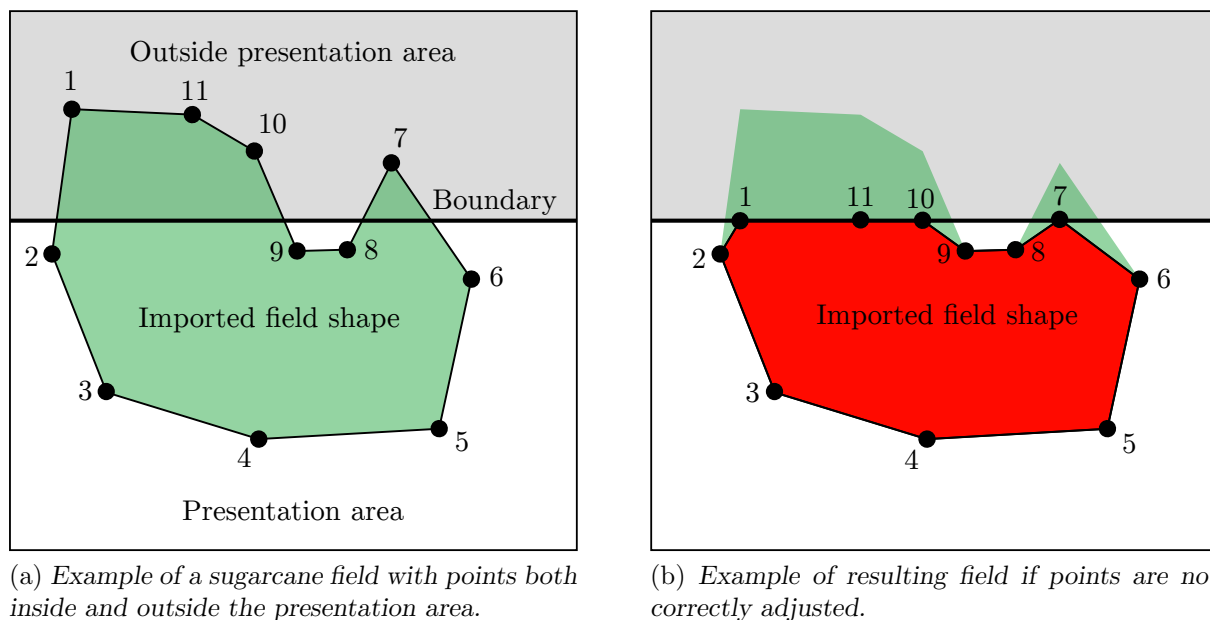


FIGURE 9.7: A hypothetical example of a polyline requiring adjustment in order to only be visible within the presentation area of the simulation model.

In order to simplify this process, the first point which in the polyline should, where possible, be one that need not be altered (*i.e.* a point which falls inside the presentation area). This point will then act as an anchor from which to plot the rest of the vertices. In most cases, since the polyline inherently has some portion of its area inside the reference field, there should be at least one such polyline point to serve as an anchor. At this stage in the algorithm, both the x and y arrays are searched to find a coordinate pair whose entries

both fall within the presentation space (*i.e.* $0 < x < 900$ and $0 < y < 900$). Once located, the array is rearranged so that this pair of coordinates is moved to the front of the array and the remaining elements are shifted along within the array in the correct order. It is possible to select locations of the reference field where no points' field vertices fall in the presentation area. One such example is shown in Figure 9.8. This occurs when a selection in the corner of the reference field is made to intercept a relatively long, straight portion of the imported field. Such straight portions of real sugarcane fields are uncommon and, furthermore, selecting the reference field to fall in such a position as that shown in Figure 9.8 would result in a particularly uninformative simulation experiment. For this reason, although the corresponding trimmed polyline may be slightly inaccurate if this position was selected, it was deemed sufficiently unlikely and, as such, was not considered.

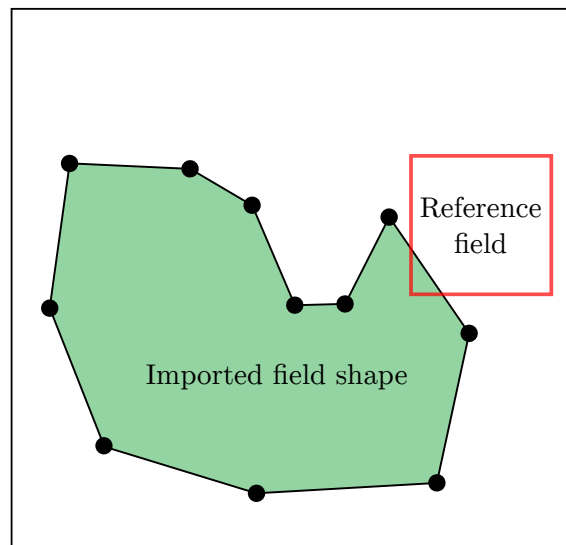
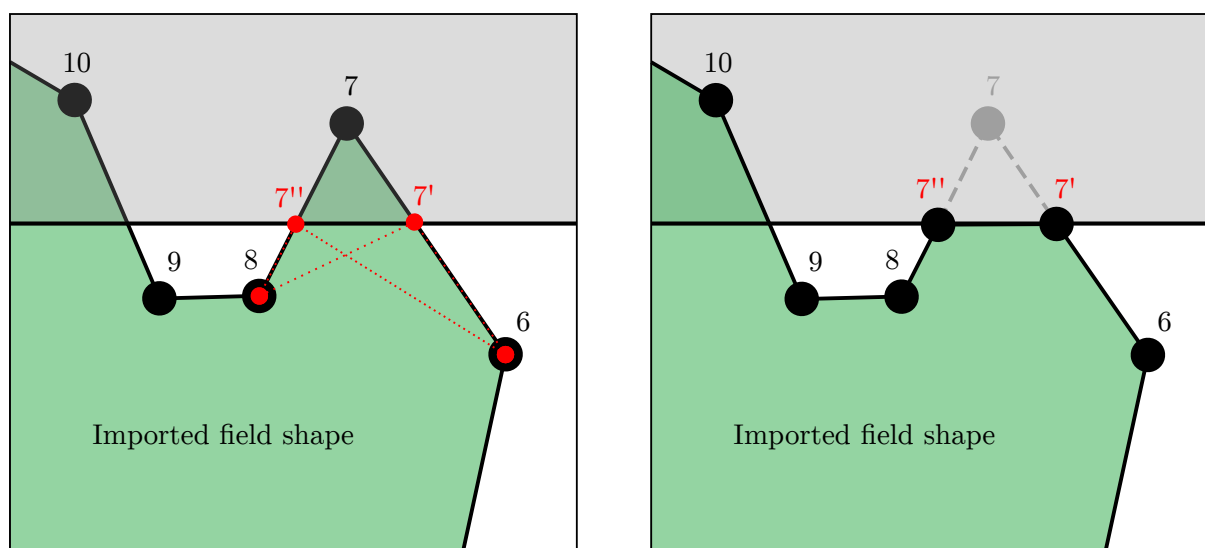


FIGURE 9.8: Possible positioning of the reference field without any polyline vertices falling within the presentation area which may result in inaccuracies when attempting to trim the fields effectively to the presentation area boundary lines.

5. For reasons similar to those motivated in the previous step, the first point in the array is duplicated and appended to the end of the array, making it one element longer. This results in the polyline starting and ending in the same place, both points typically not requiring any alteration. This once again aids in simplifying the trimming process by allowing the final point in the polyline to serve as a second anchor to the shape.
6. In light of the issues encountered in the examples illustrated in Figure 9.6, a more sophisticated method of altering points which fall outside the presentation area was required to map the visible portions of the fields accurately. The method employed by the algorithm uses interpolation between two points (for example, points 1 and 2 in Figure 9.7(a)) in order to locate the interception point where the polyline crosses the presentation area boundary. Once located, point 1 can then be moved to the interception point to ensure that no part of the visible field is cut off by the adjusted polyline, as was the case in Figure 9.7(b). This interpolation process is described in more detail in the steps that follow.

One potential drawback of this method occurs for point networks like those of points 6, 7 and 8. In this case, point 7 is required to be adjusted and moved to two different interception points, as illustrated in Figure 9.9(a). As may be seen in the figure, moving from point 6 to point 7, the interpolation process will relocate point 7 to point 7'. When

moving from point 7 to point 8, however, point 7 will be moved a second time — from 7' to 7''. As such, a single point cannot satisfy the requirements for accurate trimming of the shape. The algorithm therefore performs a search through the arrays to find any points which fall outside the presentation area, but fall between two points that are both within the presentation area. Once these points have been located, they are duplicated and placed as consecutive points in the array. This means that, during the point adjustment process, the same point may be split between two boundary intersection points and effectively satisfy the trimming requirements, as illustrated in Figure 9.9(b), where point 7 was duplicated and, subsequently, points 7' and 7'' could be moved to the location determined by interpolation between point 7 and points 6 and 8, respectively.



(a) An example of a single point which is required to be moved to two different locations through interpolation

(b) The result of duplicating a point so that it may be split between two locations

FIGURE 9.9: Example of a case where duplication of a point is required since both the previous and following polyline vertices fall within the presentation area, thereby allowing for the point to be split and interpolated between two locations.

7. At this stage, the arrays storing the vertices of the polylines which map the shape of the fields have been prepared sufficiently and the resulting presentation shapes may be drawn into the simulation model. In order to begin this process, a new polyline is dynamically created and added to a group named 'finalsection'. This group holds and maintains all of the final fields that are imported into the simulation model.
8. Since the starting point is known to be located within the presentation area, its location is fixed as the first point of the polyline and no assessments need to be made in order to determine whether adjustment is required.
9. Each following polyline vertex point is then placed according to the locations stored in the respective arrays. When placing a point, various checks are performed to determine whether any adjustment is required. In each case, one of, three possible situations is encountered:

- (i) *No adjustment is required:* This occurs when both the x and y coordinates fall within the presentation area and, as such, the vertex may simply be fixed according to the array values.
- (ii) *Adjustment is required on both coordinates:* This occurs when both the x and y coordinates of the vertex either exceed or fall below the presentation area boundaries. In such cases, the point is adjusted to meet the nearest corner of the presentation area.
- (iii) *Adjustment is required on one coordinate:* This occurs when either the x or the y coordinate of the vertex exceeds the presentation area boundary, exclusively. In this case, the coordinate which falls outside the boundaries is adjusted to equal the value of the closest boundary. When this occurs, however, the value of the previous point placed as a vertex must be assessed in order to determine whether or not the line joining the two vertices crosses a boundary line of the presentation area. If the previous point also required adjustment of one coordinate (*i.e.* it, too, exceeded or fell short of the presentation area at the same boundary as the current point), then it follows that the line between the two vertices does not cross a boundary.

If, however, the previous point falls completely within the presentation area, it follows that the line joining the two vertices crosses a boundary and, as such, interpolation must be carried out to find the interception point with the boundary and place the adjusted vertex at that point. The manner in which this interpolation is conducted is shown in Figure 9.10. The gradient between the two points is calculated as $m = \frac{\delta y}{\delta x}$. In the case of the figure, the x coordinate of point 2 is then required to be increased by a value of x_{dist} in order to fall on the intersection point I_{23} . This distance is calculated using the value of y_{dist} in conjunction with the gradient. The value of y_{dist} is simply the distance between the y coordinate and the boundary line (in this case, either 0 or 900). Once the corresponding x_{dist} value has been determined in the example shown in Figure 9.10, it is added to the x coordinate, whilst the y coordinate is simply set to equal the boundary, thereby successfully locating the vertex at the point I_{23} .

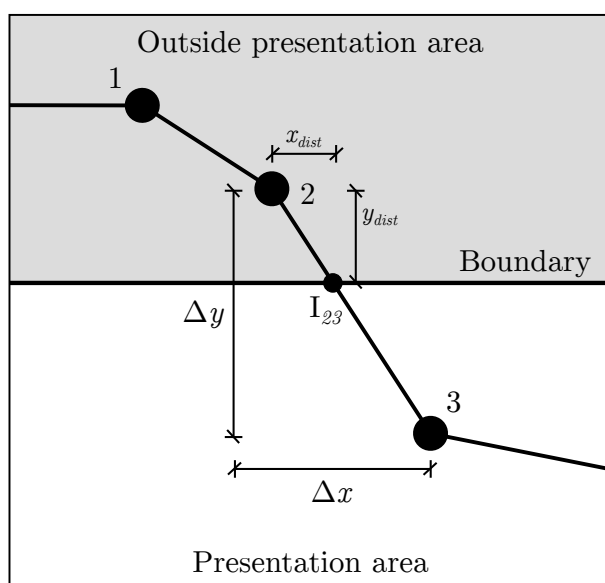


FIGURE 9.10: Example of a case where interpolation is required between two points in order to effectively adjust the point outside the presentation area so that it is located at the intersection point between the original polyline and the boundary line.

This interpolation is required to be suitable for a number of cases. Either the x or y coordinate may need adjusting using interpolation, depending on where outside the presentation area the coordinates exceed or fall short of the boundaries. Furthermore, either coordinate may be required to be increased or decreased, depending on the arrangement of the consequent vertices. For this reason, the algorithm considers every possible arrangement, ensuring that the correct manipulation is performed on the point in question, depending on its relative orientation.

10. In the cases described above, the algorithm will successfully interpolate any points on a vertex requiring adjustment, depending on the location of the previous vertex. Successful interpolation therefore depends on the direction in which the polyline is plotted. In Figure 9.11(a), the polyline may be plotted clockwise from points 2–9, or anti-clockwise from points 9–2. This means that, when a point exits the presentation area and refers back to the previous point, the algorithm picks up that the adjustment of the point requires interpolation. This is shown to be considered correctly between points 2 and 1 in Figure 9.11(b) where the polyline vertices are being plotted in a clockwise direction. Consequently, when points 11 and 10 are placed, a reference back to the previous point indicates, in both cases, that it also fell outside of the presentation area and, as such, does not require interpolation. In this example, the x coordinate of the vertex simply remains the same, whilst the y coordinate is set equal to the boundary value.

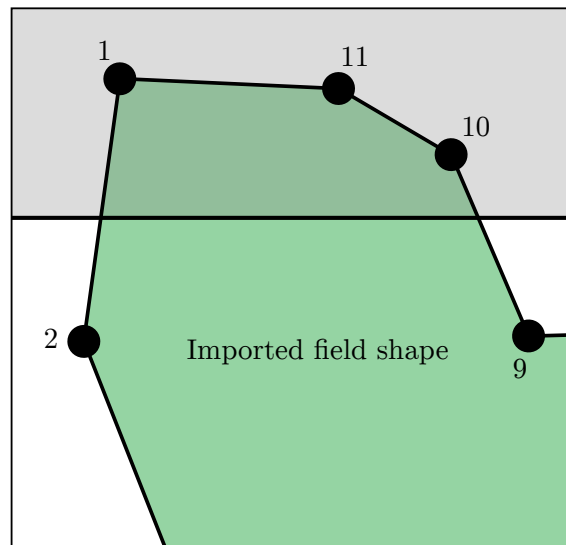
Then, upon re-entry into the presentation area, both coordinates of point 9 fall completely within the boundaries and, as such, the algorithm is not prompted to evaluate the previous point. For this reason, point 9 is simply located at its correct coordinates. As may be seen in Figure 9.11(b), this results in an inaccuracy between points 10 and 9 because no interpolation occurred between the points.

In the converse case, depicted in Figure 9.11(c), the polyline is plotted in an anti-clockwise direction and, in this case, the interpolation occurs correctly between points 9 and 10 when the polyline exits the presentation area, but once again is inaccurate upon re-entry between points 1 and 2.

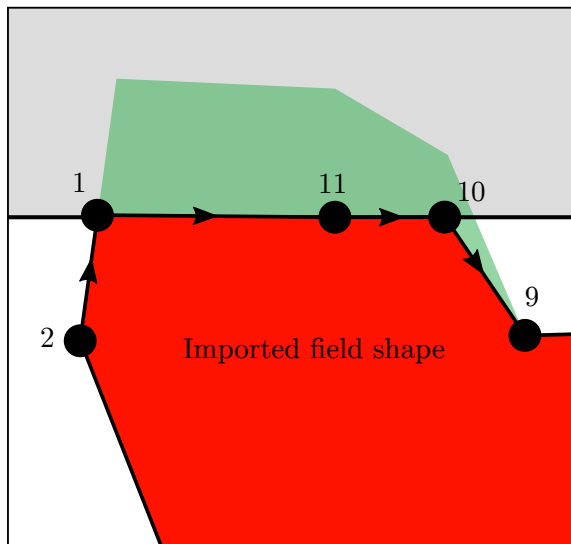
In light of this, a second test is incorporated in the algorithm which is performed when a vertex has both x or y coordinates within the presentation space. During this test, the previously plotted vertex is assessed in the same manner as in (iii) above in order to determine whether or not re-entry into the presentation space has occurred during the location of this vertex. If this is found to be the case, interpolation between the current and previous vertices occurs and the previous vertex is adjusted in order to map the intersection point between the polyline and boundary line accurately.

In cases such as those described in 6, both forms of interpolation will occur in the algorithm but, owing to the fact that the single outlying point was duplicated, the correct adjustment may still be facilitated by moving one of each of the duplicate points to the appropriate intersection site.

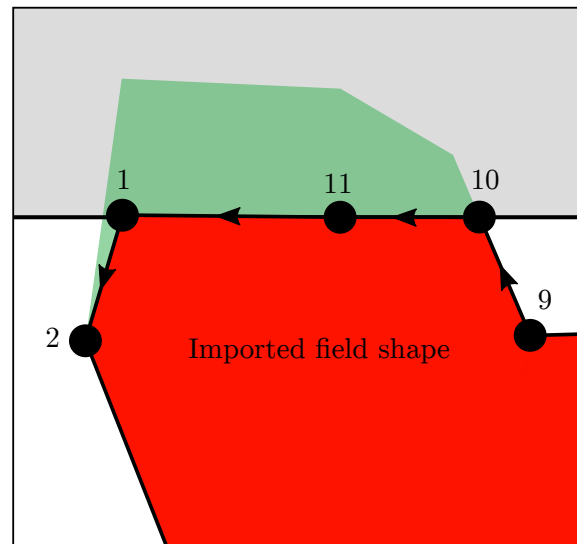
11. At this stage, all possibilities have been considered and the polyline is believed to be, under most circumstances, adequately adjusted so as to only plot points which fall within the presentation area, with all outlying points being accurately relocated along the perimeter of this area. A final assessment of all points is conducted to ensure that no points still exceed or fall short of any presentation area boundaries, or have been relocated incorrectly during the adjustment process. An example of such a case is shown in Figure 9.12. As may be seen in the figure, point 2 falls in the outer quadrant (the case illustrated in Figure 9.6(b)) and, as such, will be moved to the nearest corner of the presentation area



(a) Example of a location where the direction of plotting the polyline affects the adjustment process.



(b) Clockwise plotting results in an inaccuracy between points 10 and 9.



(c) Anti-clockwise plotting results in an inaccuracy between points 1 and 2.

FIGURE 9.11: A hypothetical example of the potential inaccuracies that may be encountered when only assessing the location of the previous point when a vertex falls outside of the presentation area. The locations of these inaccuracies in the resulting polyline depends on the direction in which the points are plotted.

at point 2'. In this case, however, the point actually requires adjustment to intercept the right-hand boundary at point 2'', based on the vertex to which it is joined inside the presentation area.

A second possible case where mis-adjustment may take place is shown in Figure 9.13. As may be seen in Figure 9.13(a), the arrangement of the outlying points is such that the manner in which the algorithm interprets and adjusts them does not result in an accurate representation of the trimmed field. This means fields which intercept boundaries, particularly those near the corners of the presentation area, require this final assessment.

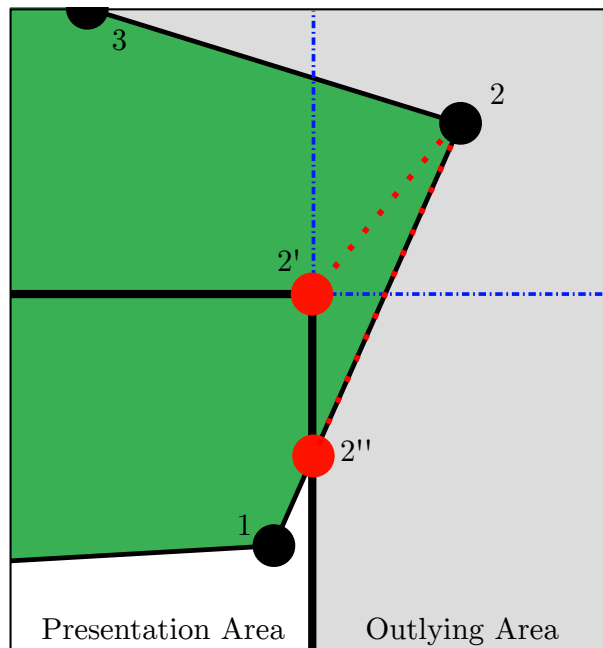
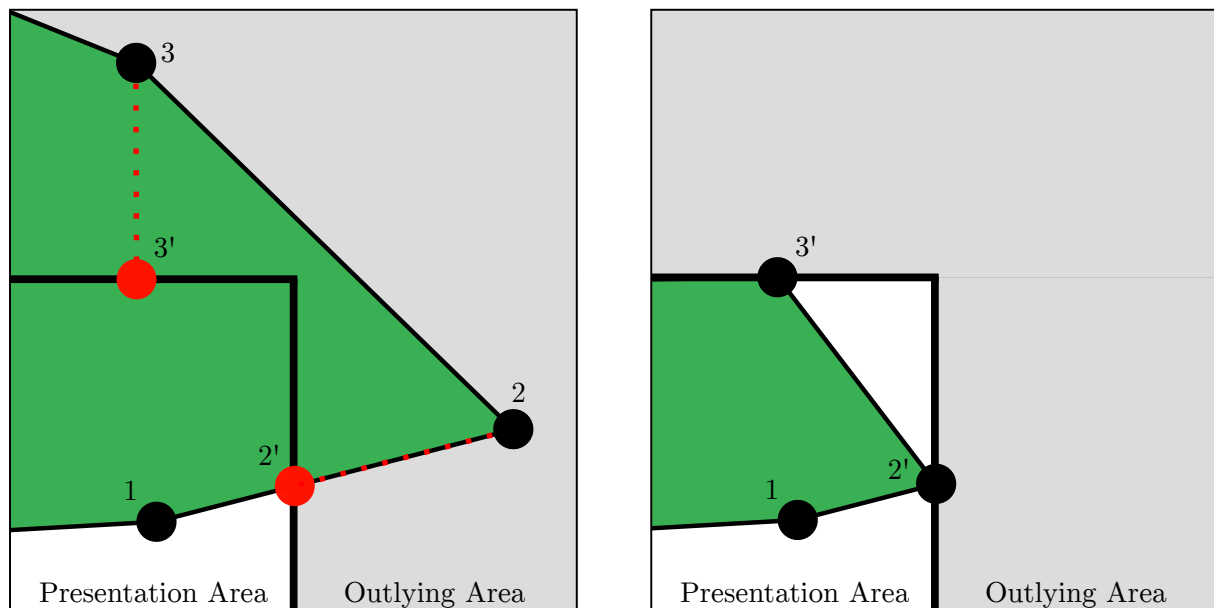


FIGURE 9.12: Example of a case where points may be incorrectly adjusted by the vertex adjustment algorithm owing to exceptional circumstances where adjacent vertices are long distances apart.



(a) Interpretation of where points should be adjusted to based on the adjustment algorithm.

(b) The resulting trimmed field.

FIGURE 9.13: Example of a case where the points interpreted by the algorithm are insufficient to specify the complete area to be covered by the final trimmed field and, as such, an inaccurate field shape is generated.

The final assessment begins by determining the actual vertices that the trimmed polyline should pass through. This is achieved by comparing each pixel point in the presentation area to those points both vertically and horizontally adjacent to it. Using the untrimmed fields which were originally imported into the model through the extraction process, each

point is assessed as to whether or not it lies within a particular field. A common assessment case is illustrated in Figure 9.14. In the illustrated case, point (i, j) is compared to the adjacent points as indicated in the figure. If a situation is encountered where the points $(i - 1, j)$, (i, j) and $(i + 1, j)$ do not all fall within a specific field, as well as $(i, j - 1)$, (i, j) and $(i, j + 1)$ not all falling within the same field, then it follows that the point (i, j) must be a vertex of the polyline, since it indicates a change in both the x and y directions (as denoted by the green and white colouring of the pixels in the figure). Owing to the fact that only points within the presentation space will be considered, a special case is included for points which fall on a boundary line. Following this process, all of the appropriate vertices may be discovered. These vertices are, however, not discovered in a sequential, circular order, but rather by a sweeping search algorithm. This means the points may not simply be translated into polyline shapes, as the resulting polylines would form connections between incorrect nodes, causing intersections between lines and misshapen fields.

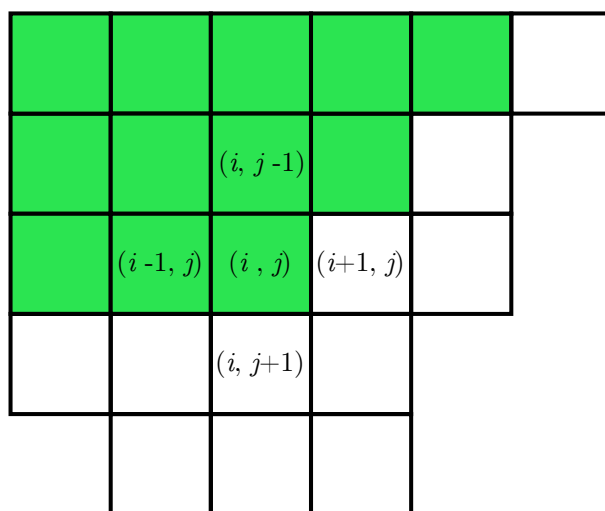


FIGURE 9.14: Example of the comparison which was conducted between adjacent pixel points in order to determine the location of the polyline vertices.

For this reason, a comparison is conducted between the vertices and the trimmed shape in order to determine the degree to which they agree with one another. In cases where incorrect vertices have been adjusted (as was the case in Figure 9.12) or a vertex has been incorrectly excluded (as shown in Figure 9.13), the necessary manipulations are conducted to ensure that all vertices are correctly included, thereby ensuring that an accurate, complete trimming process is conducted on all included field shapes.

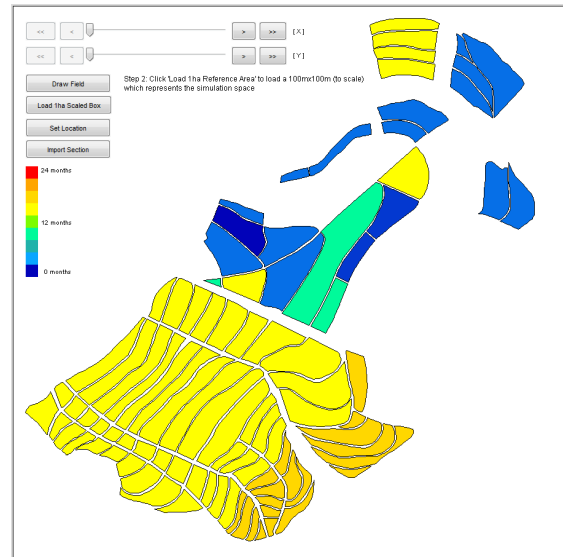
12. During the final step, each polyline is plotted and appropriately coloured within the presentation area according to the initial age of the sugarcane, as determined from the associated attribute file. Darker shades of green are used to indicate sugarcane of an advanced age.

9.3 Functionality assessment of the GIS importation process

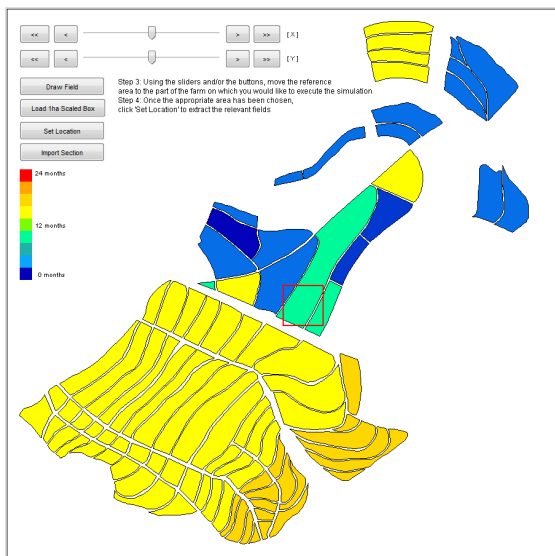
In order to assess the success with which the GIS importation process was implemented in the simulation model, the process was executed several times using an example shapefile of a sugarcane farm located at Eston [37]. An example of the typical execution process is illustrated below in Figures 9.15(a)–9.15(d) and the following interpretations of each figure are offered.



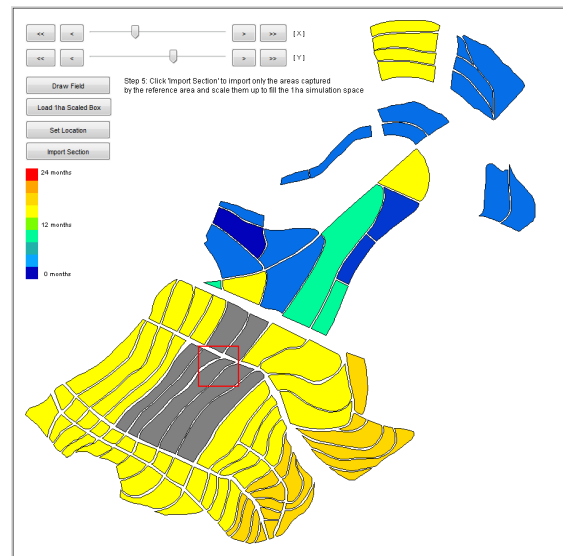
(a) The opening screen of the GIS site selection process.



(b) An example of a shapefile whose contents have been extracted, plotted as polylines, scaled and centered within the presentation area.



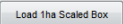
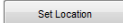
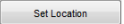
(c) A scaled reference field, representing 1ha, generated based on the imported shapefile contents.



(d) The shaded fields are all fractionally (or wholly) located within the now fixed selected area and must be included extracted, scaled and trimmed for inclusion in the presentation area.


FIGURE 9.15: An illustrative example of different steps followed in order to import a sugarcane farm shapefile into the simulation model. This begins by the selection of a 1 ha area from within the farm's area, and extract the corresponding information in order to replicate that selection in the presentation area. The simulation may then be executed within this space.

Figure 9.15(a) is the initial screen encountered by the user once the user-defined parameters have been selected and the simulation has commenced. Here, the user is prompted to click the button which imports the GIS information. The result of clicking this button is shown in Figure 9.15(b) which is generated by the process described in §9.2.1. As may be seen, the shapefile is scaled and centred to fit comfortably within the presentation area. In this

example, the shapefile contains sugarcane which follows a two-year crop cycle, as indicated by the colour scale bar. At this stage, the user is prompted to click the  button which calculates the size of a 1 ha area relative the imported shapefile and loads the reference field into the presentation area. The result of this action is shown in Figure 9.15(c). The user is then prompted to move the reference field to the chosen location by either clicking the location, or making use of the toolbar buttons for incremental, accurate movement. Once a location has been decided, the user is instructed to click the  button which fixes the location and identifies the fields which intercept this area according to the process described in §9.2.2. These fields are indicated by grey shading, as shown in Figure 9.15(d) where the user is then prompted to import the contents of the selected area to fill the entire presentation area by clicking the  button.

The result of this importation is shown in Figure 9.16. As may be seen in the figure, each polygon representing a field is scaled to fit the corresponding presentation area, mimicking the area selected in the previous step. Furthermore, the shapes have been trimmed so as not to extend past the boundaries of the presentation area according to the process described in §9.2.3.

9.4 Implementation of GIS features into simulation execution

Once the underlying 1ha portion of the sugarcane farm (selected by the user as the landscape within which to execute the simulation model) has successfully been imported into the presentation area, the corresponding fields are used to establish the underlying  **Sugarcane** array, representing the actual sugarcane stalks which fall within that hectare. In each case, a stalk is assessed by virtue of its location within the imported field. If it falls within the area covered by a particular field, the corresponding characteristics of the sugarcane grown in that field, as documented by the attribute file, are allocated to the stalk. If it falls within the spaces between fields, no sugarcane is grown in these areas and, as such, a value of ‘0’ is attributed to the stalk. This also means that the *E. saccharina* population will veer away from these areas as they are unsuitable as food sources for larvae. This is consistent with what is observed in the field where, but for exceptional, singular circumstances which occur in error, *E. saccharina* larvae and pupae are typically not found on the footpaths and fire-breaks between fields.

9.4.1 Functional assessment of the sugarcane attribute attribution process

An experiment was first conducted to assess the output of the process during which certain information is attributed to each sugarcane stalk based on the shape within which it resides. In this experiment, the same area as that shown in Figure 9.15(d) was selected from the example sugarcane farm and imported into the presentation area. Then, during the value attribution process, the field number (as recorded in the attribute file) was allocated to each sugarcane stalk within the boundaries of a particular field. Sugarcane stalks which do not fall within any field were attributed a value of ‘0’.


Thereafter, an algorithm was executed which locates a square at each  **Sugarcane** array point, and colours it depending on the value of the underlying array, with zero points being attributed white squares. These squares were then plotted within the presentation space and compared to the output of the importation process, as shown in Figure 9.16, so as to assess the accuracy with which specific features of certain fields could be attributed to the simulated sugarcane stalks which reside within them. The output of the experiment is shown in Figure 9.17. presentation shapes.



FIGURE 9.16: The resulting imported, trimmed and scaled selection of the area selected by the user in Figure 9.15(d).

It should be noted that, in this figure, there may be up to 129 600 small squares plotted (depending on the selection made by the user). This amount a finer detail than is offered by the pixels available in the presentation area for representing shapes. As a result, the location of the incorporated squares is moved to the nearest pixel value and results in the squares bunching in groups of four. In light of this, what appears as one square in Figure 9.17 is actually four small squares in close proximity, the space between which is indistinguishable. On the boundaries of the fields, however, smaller squares (which appear as parts of the larger squares, owing to the scale) may be seen where field boundaries pass through a collection of stalks. An enlarged view of the boundary is included to show the fine detail of the stalks. In spite of the altered appearance in the figure, the stalk locations remain equally spaced within the fields in the simulation model and are perceived at the correct coordinates by the agents, since arrays such as the **V** Sugarcane array offer a finer detail than

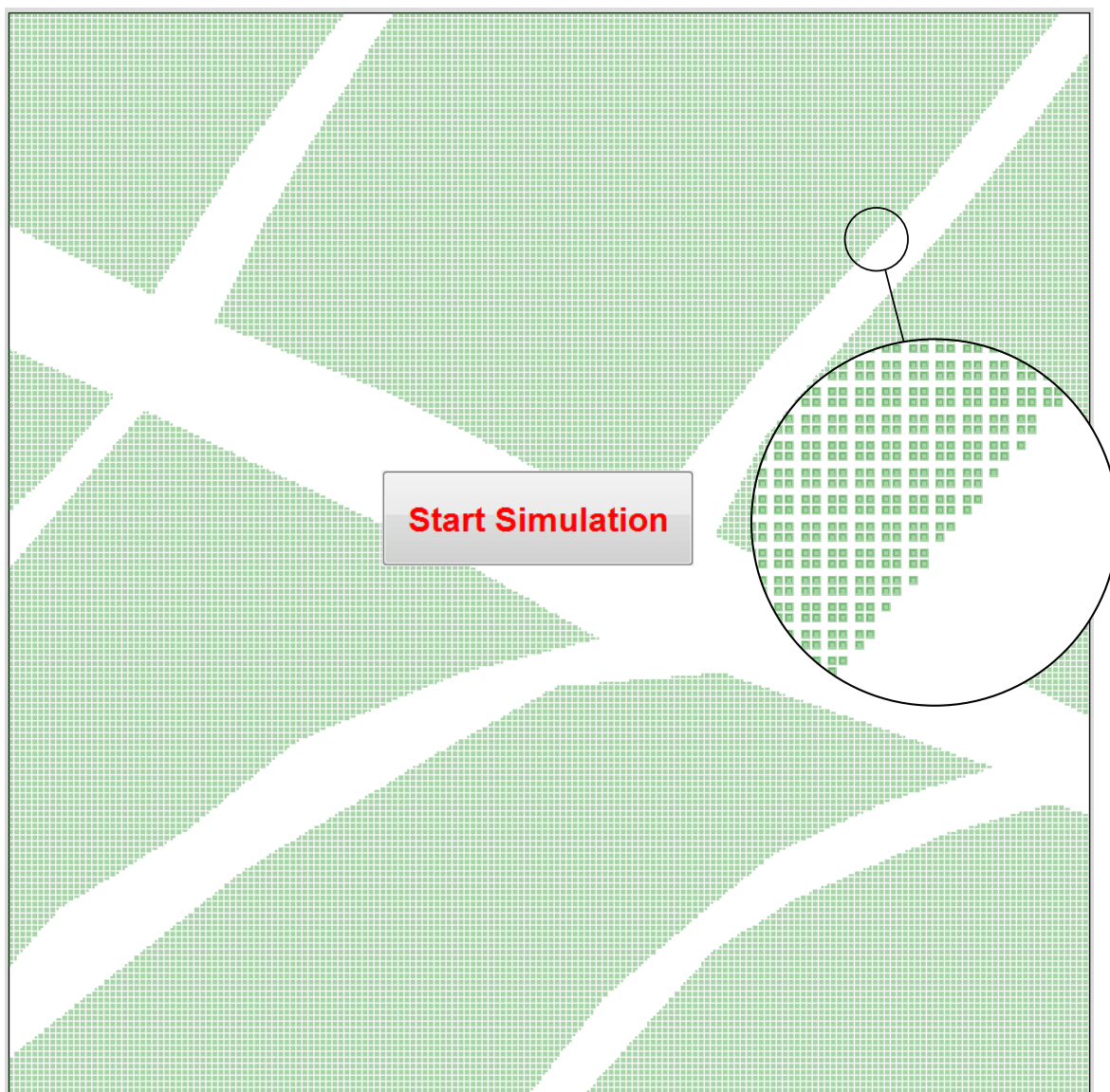


FIGURE 9.17: *The resulting points which are linked to fields shown as colour squares, based on the selection made in Figure 9.15(d).*


As may be seen in the figure, the experiment yielded field shapes consistent with those achieved by the output of the algorithm, illustrated in Figure 9.16. It was thus concluded that the process of attributing characteristics to stalks residing in a particular area functions as desired. The field shapes in Figure 9.17 are, of course, not as smooth as those in Figure 9.16 since the experimental case considers each individual potential stalk location, outputting a presentation square upon encountering an existing stalk and, as such, resulting in a similar, yet slightly discretised-looking field layout. Although useful for testing purposes, this method of showing the field shapes was deemed unsuitable for use in the model as a visual demarcation of the selected field shaped since the inclusion of each presentation shape makes use of part of the model's memory during execution and, as such, including thousands of these squares consumes a large portion of the model's memory, causing the model to execute at a slower rate. As such, the previously described algorithm was incorporated as it makes use of a smaller number of presentation shapes which cover the space of an entire fields as opposed to individual stalks, thereby preserving memory.

9.4.2 Attributing specific characteristics of sugarcane stalks to each field

As discussed in §8.3.2 and §8.3.3, various shortcomings pertaining to the manner in which the underlying sugarcane in the model of Chapters 4–8 is implemented were addressed and rectified as best possible, in light of suggestions and insight gained from the expert panel. In each case, various characteristics were attributed to the specific stalks within the simulated space, such as the typical number of internodes (stored in the **V** `Sugarcane_TypInternodes` array) and the physiological age of the sugarcane (stored in the **V** `Sugarcane_PhysAge` array). As a result of the addition of the GIS importation functionality, there now exists the possibility for a number of fields to be present within the simulation space, depending on the simulation area selection made by the user. This may, in turn, result in sugarcane which is of different varieties confined to specific areas, meaning the attribution of the typical number of internodes per stalk, as well the physiological age of the different fields, is required to be managed carefully within the presentation area, among the fields present.

In order to achieve this, the **F** `MakeSugarcaneField` function (discussed in §4.3.2) was modified to correctly assign correct information on a per-stalk basis. The static maturity value previously used to indicate sugarcane of an advanced age was removed from the model and replaced by the aforementioned arrays to store specific, accurate information pertaining fields which compose actual sugarcane farms. In order to attribute the correct characteristics to each stalk, an algorithm similar to that used during the extraction process described in §9.2.2 was constructed which assesses the relative location of each sugarcane stalk and determines in which imported field (if any) it falls. As previously mentioned, all fields inside the reference field location selected by the user and, as a result thereof, within the final presentation area, form part of the ‘finalsection’ group and, as such, can easily be accessed and filtered. The attribution process is shown graphically in Figure 9.18.

The **F** `MakeSugarcaneField` function cycles through each field, extracting the linked information pertaining to the variety of sugarcane in the field, the typical number of internodes per stalk and the physiological age of the sugarcane in the field from the associated attribute file. With reference to the example shown in Figure 9.18, it may be seen that three different fields contain points where sugarcane stalks are located within the small area shown, with two stalks not falling within any field. Each of the 129 600 points which can possibly accommodate a sugarcane stalk in the simulated hectare is considered as forming part of a particular field. The remainder of the array points within the **V** `Sugarcane` array represent space between stalks in the fields and, as such, are not considered for attribution of stalk characteristics. If the polyline which maps the perimeter of a field is found to contain the point, the appropriate characteristics are attributed to the stalk represented by that point within the different arrays. If not, the point forms part of the pathways and fire-breaks present within a sugarcane farm and, as such, remains as a ‘null’ point in all arrays. This means that these points possess no desirable properties for oviposition or infestation and, as such, the *E. saccharina* population will not move into these areas.

In light of the incorporation of specific sugarcane stalk attributes pertaining to different fields represented within the simulation model, the manner in which harvesting is managed in the simulation model was subsequently altered. A harvesting schedule is determined by the user on either a 12 or 24 month cycle. Once selected, harvesting occurs automatically according to this schedule and, as such, the user-controlled,  button was removed from the model. The actual details pertaining to the harvesting process of sugarcane is a comprehensive skill and science, and should ideally be considered and implemented in a full-scale, real-world simulation model by means of a separate study. For this reason, the auto-harvesting function was maintained to function in a similar manner as previously described in §4.3.2 whereby all stalks

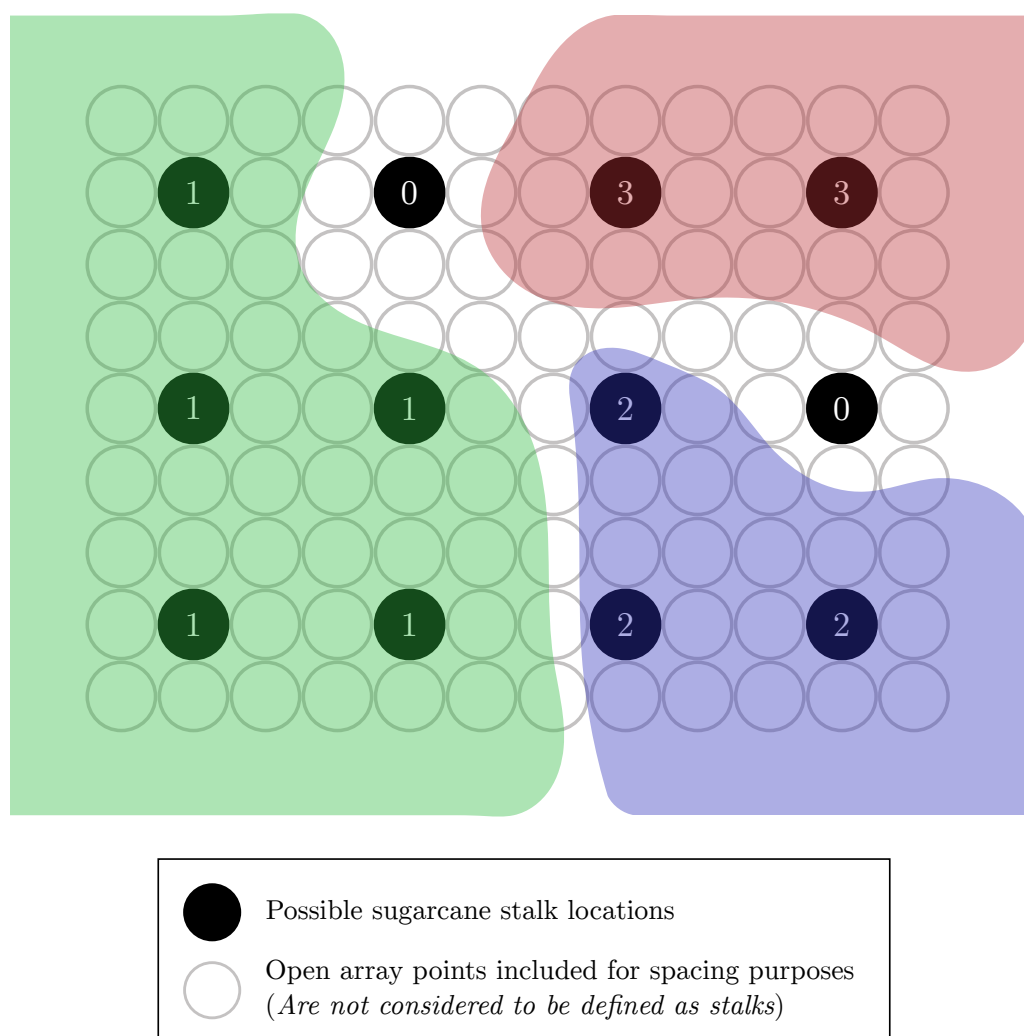


FIGURE 9.18: The manner in which characteristics are attributed to specific sugarcane fields by comparison their location in relation to the imported field shapes.

are immediately harvested once their full cycle has been reached (*i.e.* they reach an age of either 12 or 24 months). When this occurs, the **V** `Sugarcane_height` values within the field being harvested are set to 50mm. This is done in view of insight gained from the expert panel discussion (as described in Chapter 8) that ratoon crop stalks are cut to just above soil level during harvesting.

The current age of the stalks for each field is tracked by the **F** `AgingCane` function which increments each crop's age once per month. In order to observe the ageing process in the simulation model, the colour of each field is also darkened as the crop ages. Once the crop age reaches the user-defined crop cycle age, it is harvested and reset to zero. The colour of the harvested field is then subsequently reset to a lighter colour to indicate young sugarcane.

The ageing and colouring processes were tested by means of executing the simulation model over the course of a year in the absence of any agents. The results of the experiment are shown in Figure 9.19. As may be seen in Figure 9.19(a), the selected area covers three fields of differing ages. The field in the top right-most corner of the figure has a crop age of 14 months at the initiation of the experiment, as determined from the linked attribute file. The central and left-most field begin the experiment at ages of 1 month and 6 months, respectively. The crop cycle

length selected for this experiment is 24 months.

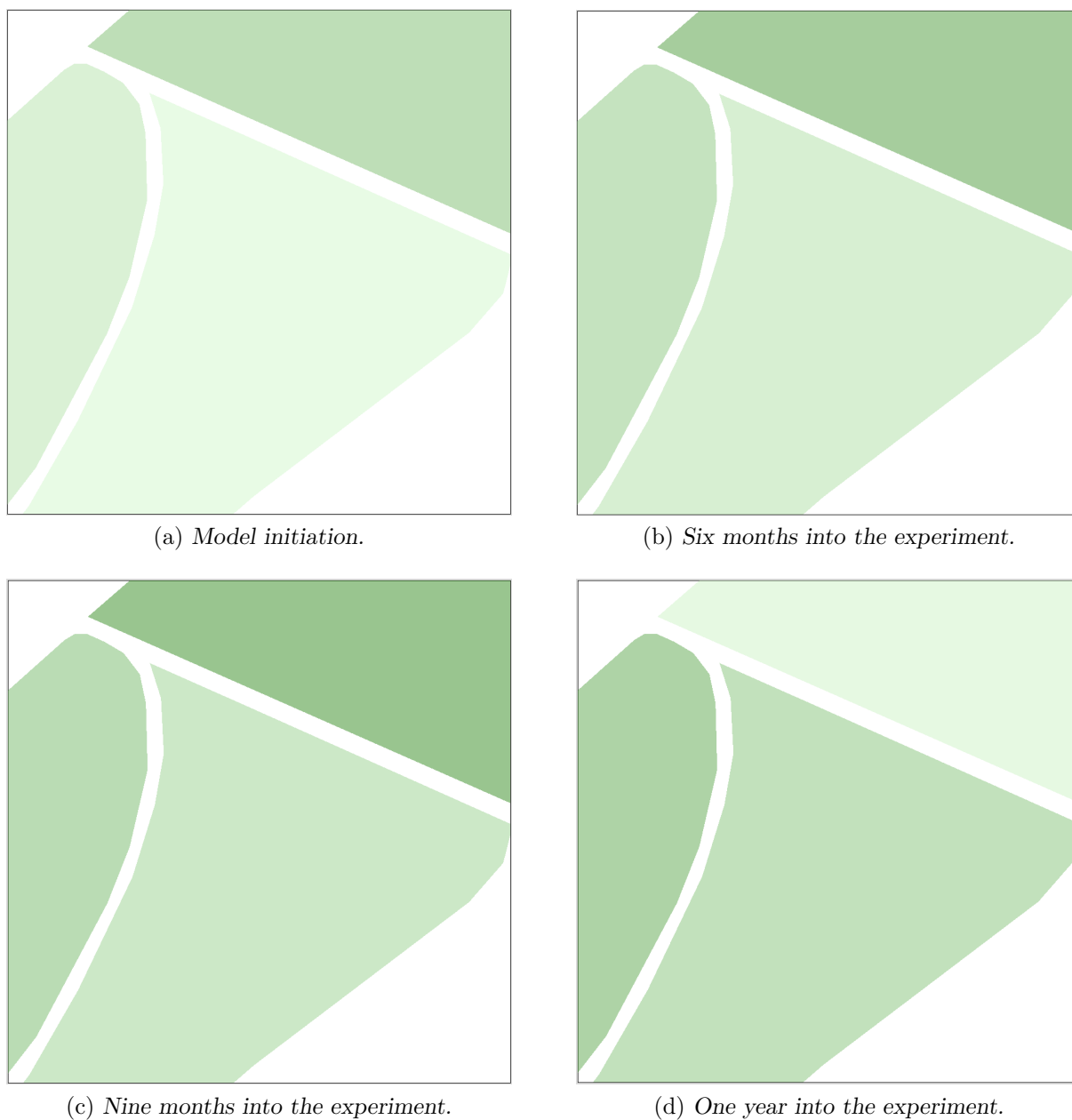


FIGURE 9.19: The results of an experiment conducted to assess the process through which crop ages are tracked and corresponding fields are coloured according to their age, as well as harvested once they reach the user-defined crop-cycle limit.

As may be seen in Figures 9.19(b)–9.19(c), the fields darken progressively as the model runs and the crops age. One year into the experiment, the top right-most field has exceeded the 24 crop cycle and, as such, has been harvested. This is seen by the colour reset and consequently a light field colour in Figure 9.19(d). In light of these results, as well as other experiments executed in order to assess the functionality of other sugarcane-related modules of the simulation, the sugarcane attribution functions, as well as the age management and harvesting functions, were all found to be working effectively and in correspondence with the specific field shapes imported into the simulation model.

9.4.3 Incorporation of emitting and absorbing boundaries

As a result of the fact that the simulation model represents a pre-selected hectare which forms part of an entire sugarcane farm, it follows that it is likely that there exist other infestations of *E. saccharina* populations in the areas outside the boundaries of the presentation area. As mentioned in §8.3.1, experts estimate that approximately 10% of females within an *E. saccharina* population are ‘far fliers’ and are likely to disperse over greater distances in order to promote outbreeding as opposed to typical females who generally oviposit close to the site of mating. This greater dispersal can, in turn, cause new infestations to break out in previously uninfested areas or fields. Such far fliers which form part of the simulated *E. saccharina* population residing within the presentation area are likely to disperse to the outlying areas which fall beyond the presentation area boundaries into fields that were not selected for simulation purposes by the user at model initiation.

In §4.3.2, the absorbing and emitting boundary control system incorporated into the model was discussed. This inclusion allows far-flying females to ‘exit’ the simulation space, as well as facilitates functionality to allow similar far-flying females assumed to exist within areas surrounding the simulation space to enter into the presentation area and oviposit eggs. During the model analyses, however, it was decided to exclude incoming far-flying females owing to a lack of information related to a quantification of the rate of foreign enterers through the presentation area boundaries.

In order to incorporate this functionality into the updated simulation model, a modified zero-flux boundary condition was implemented. This means that far-flying females which disperse over the boundaries of the presentation area for oviposition must be balanced by similar females who consequently enter into the presentation area for the same purpose. This condition is only pertinent to far-flying female agents and does not relate to agents who exit the presentation area by means of small-scale random movement. Females entering the simulation space are assumed to be mated and therefore begin immediately with the oviposition site selection process upon arrival.

When a far-flying female exits the simulation space, the coordinates of the female’s location prior to her departure flight are transferred to the **F**BoundaryControl function. These coordinates are then linked in the function to identify the field within the simulation space from which she departed. Since the actual process in nature is not thoroughly understood and almost impossible to quantify comprehensively, a simplifying rule is imposed whereby a responding entering female will select an oviposition site within the field from of which the exiting female departed. Once the female has departed, a randomly assigned time period of between 0–336 hours passes before a responding female is generated and prompted to enter the simulation space from a random location. This is done in order to create a number of cohorts of *E. saccharina* to exist concurrently at different life stages within the simulation model. In all previous experiments, the exclusion of the emitting boundary condition, in conjunction with the initial conditions selected for the experiment, resulted in a single cohort of moths passing through the different life stages at much the same time. Although preferential for the purposes of the experiments, this is not the case in nature and, particularly in a selected experimental site, new cohorts are typically initiated by entering females in search of oviposition sites. By generating an incoming female at a random time within the two weeks following a corresponding female’s exit, the zero-flux boundary condition is, to some degree, maintained, whilst still allowing for diversity and dynamism in the experiments.

The arrival location or the responding female is randomly selected as a point which typically falls within the correct field. In order to simulate the possibility of infestation outbreaks in

previously uninfested fields, 10% of incoming females randomly select one of the fields captured within the presentation area as a site for oviposition. This, once again, provides the diversity and dynamism typical of a real-world study. As mentioned during the expert panel discussion (described in Chapter 8), the biological role of ‘far-flying’ females is to promote out-breeding and, as such, they often end up ovipositing in inferior sugarcane to that from which they departed. The consequently fostered population may then disperse back towards sugarcane of a better quality and advanced physiological age if such sugarcane can be located nearby.

Although simplified, the implementation of the process accommodates the outbreak of new infestations in previously uninfested areas or parts of fields and, consequently, new cohorts of agents passing through the life stages of the pest within the area under study, as has been observed in nature [72, 90].

An example of the execution of the absorbing and emitting zero-flux boundary control process is shown in Figure 9.20. It should be noted that this is a hypothetical example and that the maturation times between the life stages shown are not necessarily realistic. The example was, however, constructed for illustrative purposes. The following interpretations are offered for each of the parts of the figure:

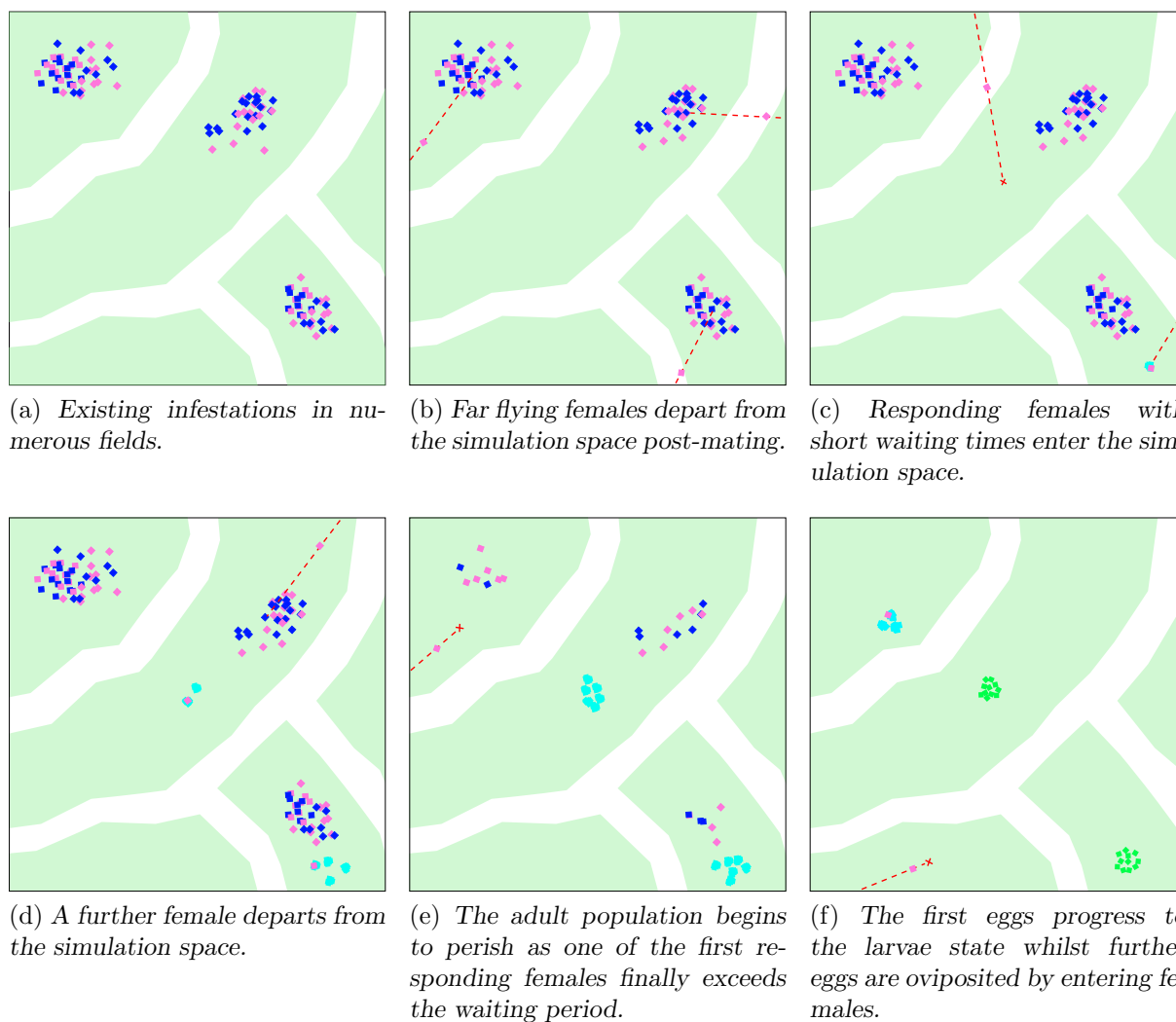


FIGURE 9.20: Illustrative example of the absorbing and emitting boundary control as executed during simulation operation.

- (a) Figure 9.20(a) shows a typically selected area for simulation with a number of infestations existing within the different fields. Only agents residing in their respective adult life stages are shown in this initial figure as such agents are able to exit the simulation space post-mating.
- (b) In Figure 9.20(b), a number of mating instances have occurred and, as such, some females are ready to begin the oviposition process. As mentioned, 10% of these females are genetically assumed to be far-fliers and hence begin to depart the simulation space post-mating. The flight paths of the exiting females are illustrated by means of a red dashed line. Females typically exit the simulation space toward the quadrant boundary of the quadrant within the simulation space from which they depart (*i.e.* a female who departs from a position located in the top left quadrant of the simulation space will typically leave through a point on the top or left boundary within this quadrant).
- (c) Once the females have exited the simulation space, a random waiting time is executed before the responding female enters. As may be seen in Figure 9.20(c), the responding female generated in response to a female having exited over the right-hand presentation area boundary has already exceeded the waiting period and has begun moving toward the randomly determined arrival location within the field from which the outgoing female departed. No female has yet entered in response to the female who departed through the left hand boundary. Finally, the female generated in response to the exiting female from the bottom right field has already moved to the respective randomly selected location within the field and has begun oviposition. This is in light of a short waiting time sample attributed to this female.
- (d) In Figure 9.20(d), two of the entering females have arrived at their respective locations and have begun ovipositing eggs. The third female's waiting period has still not expired and, as such, no responding agent has been generated. In the meantime, further mating has also occurred and a far-flying female has been prompted to depart from the simulation space through the top boundary of the presentation area.
- (e) Figure 9.20(e) shows that a portion of the original adult population has begun to perish as a result of age. Furthermore, the two first-responding emitted females have completed their respective oviposition cycles and have also perished. As may be seen in the top left quadrant, the female in response to the earlier departing female in that quadrant has surpassed the waiting period and has been instructed by the algorithm to oviposit in the same field from which the original female departed.
- (f) Finally, in Figure 9.20(f), all of the original adult agents have perished, whilst the first oviposited eggs have progressed to the `LarvaeOutsideStalk` state. As a result of high mortality rates experienced in this state, it may be seen that only a few larvae still remain from these batches of eggs. The female in the top left quadrant has also begun oviposition at the appropriate location. As may be seen at the bottom of the figure, the female generated in response to the departing female shown in Figure 9.20(d) has exceeded the waiting period. In this case, this responding female is part of the 10% of the population who are instructed to oviposit in a random field, as opposed to the same field from which the exiting female departed. This female has randomly selected a site in the bottom field and is heading toward it. The staggered time between these oviposition instances will, in time and with further generations, give rise to distinct cohorts of the pest residing in different life stages, as is the case in nature.

Owing to a lack of thorough understanding of the drivers of *E. saccharina* female dispersal, as well as the complex nature of thoroughly quantifying or specifying the manner in which it occurs, the aforementioned assumed process was deemed to provide suitable variability and effectively mimic the real-life process observed in nature, consisting of a number of cohorts of the pest being present within an observation area, as well as the possibility of new infestations breaking out as time progresses.

9.4.4 Functionality assessment

In order to assess the manner in which the GIS importation functionality operates when incorporated into the simulation model, a number of tests and experiments were conducted. Importantly, the inclusion of this functionality in the simulation model is intended to serve as a platform from which future studies may be launched. In particular, the ability to select a specific area within which to execute simulation experiments provides the potential to replicate existing or future real-world studies which have been conducted in areas on certain sugarcane farms. Before such a venture can feasibly be undertaken, as well as produce results with a reasonable amount of accuracy, however, a number of further improvements or refinements should be made to the model. Most notable of these improvements is the inclusion of a full-scale, accurate sugarcane model which considers all elements affecting the growth and development of the sugarcane plant, such as soil quality, aspect, rainfall and others. Furthermore, several biological aspects of the pest also require further insight and information which may be attained from real-world entomological studies, or further in-field observations in the future. The finer details of these future work propositions are discussed later in this dissertation.

Because of the above-mentioned limitations, the aim of the experimental analyses conducted on the complete model aimed more to illustrate the operation of the different modules combining the simulation model, rather than to generate definitive outcomes to be assessed and scrutinized. That said, it is important to use the results generated by the simulation experiments to assess the degree to which the simulated *E. saccharina* population continued to behave as predicted or witnessed in previous, simpler experiments and to make the necessary adjustments if this is not found to be the case.

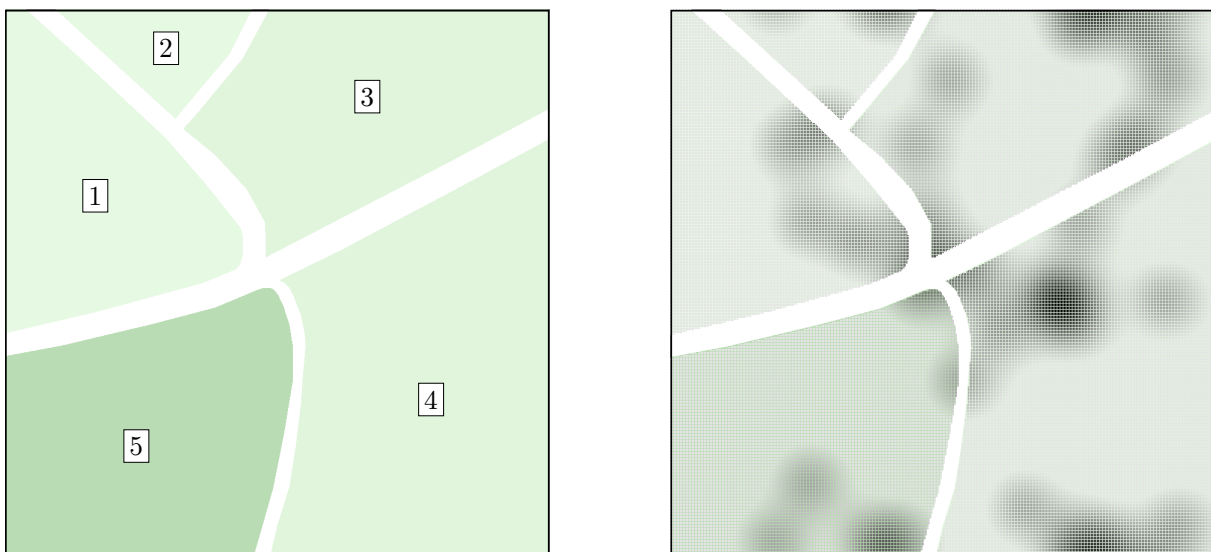
A simplifying assumption was made in the model of an initial *E. saccharina* infestation of 0.1 e/100s across 100% of the area in each field (*i.e.* uniformly distributed within each field). This means that no fields were excluded from infestation initially. Owing to the different initial ages of the fields (as specified in the associated attribute file), however, the lengths of the stalks in each field were different. This means that, as a result of the effect of carrying capacity, initial infestations in fields of very new sugarcane (0–4 months) were unlikely to be able to support larval feeding and, as such, initial infestations died out in these fields. Conversely, in older fields, more larvae survived, increasing the potential of fostering infestations in these areas. It should be noted that future infestations were still capable of breaking out in fields with young sugarcane where the initial population did not survive by means of the 10% of incoming females who randomly select a field for oviposition. No selection process is, however, followed by females who embark on long flights and, as such, the resulting infestation outbreak does not necessarily occur in sugarcane of a more advanced physiological age.

A further inhibitor of the simulation model is the computational expense of simulating a large number of agents. As was the case in previous experiments, the simulation model struggles to run simulation instances representing time spans of longer than 6 months. Ideally, detailed experiments should be performed over the course of a year or more, but the computational burden is too great on a single, regular computer to achieve this goal at present. Suggestions are

made later in this dissertation in respect of possible means to decrease the computational expense of the model and, in doing so, hopefully allow for longer simulation runs to be conducted. Such enhancements, however, fall outside the scope of this study and, as such, simulation experiments were confined to 6-month time periods only.

A typical experimental progression is described below. In Figure 9.21, the details of the area that was selected from the Eston [37] shapefile are shown, whilst the development and dispersal of the population is shown in Figure 9.22. Finally, for a more insightful view on the animation output of the model implementation during execution, Figures 9.23(a)–9.23(c) contain screenshots taken of the animation output at the end of the simulation experiment run. For analysis purposes, the

button was selected to show the physiological age profile superimposed on the field shapes in the images.



(a) The area selected for a first the simulation experiment.

(b) The randomly generated physiological age of the sugarcane in the selected area.

FIGURE 9.21: Details of the shape and physiological age of the area of sugarcane farm selected from the imported shapefile on which to run an illustrative experiment.

The agent densities reported refer exclusively to agents residing in the **Egg**, **LarvaeOutsideStalk** and **Larvae** states. Although, typically, the larvae inside stalks and pupae are the only agents that can be collected and counted during an in-field experiment, in the simulation, the states most indicative of the population progression, development and behaviour are those previously mentioned and are hence tracked. This aids in attempting to explain the effects of what is observed in a simulation run's output. In future studies, should the output of the model have to be compared to an in-field study, the tracked and recorded life stages may easily be altered to suit the experiment requirements.

As may be seen by the colour of the fields in Figure 9.21(a), the fields are of differing initial ages. Fields 1 and 2 begin the simulation as newly harvested fields (zero months old), whilst fields 3 and 4 are four months old, whilst field 5 is seven months old. It is expected that no initial infestation will survive in fields 1 and 2 upon model initiation owing the unavailability of feeding sites. In Figure 9.21(b), the physiological profile of the selected area is shown, with darker areas once again representing sugarcane of a more advanced physiological age, where *E. saccharina* populations are anticipated to thrive.

As may be seen from in Figure 9.22(c), no infestations broke out in the two newly-harvested

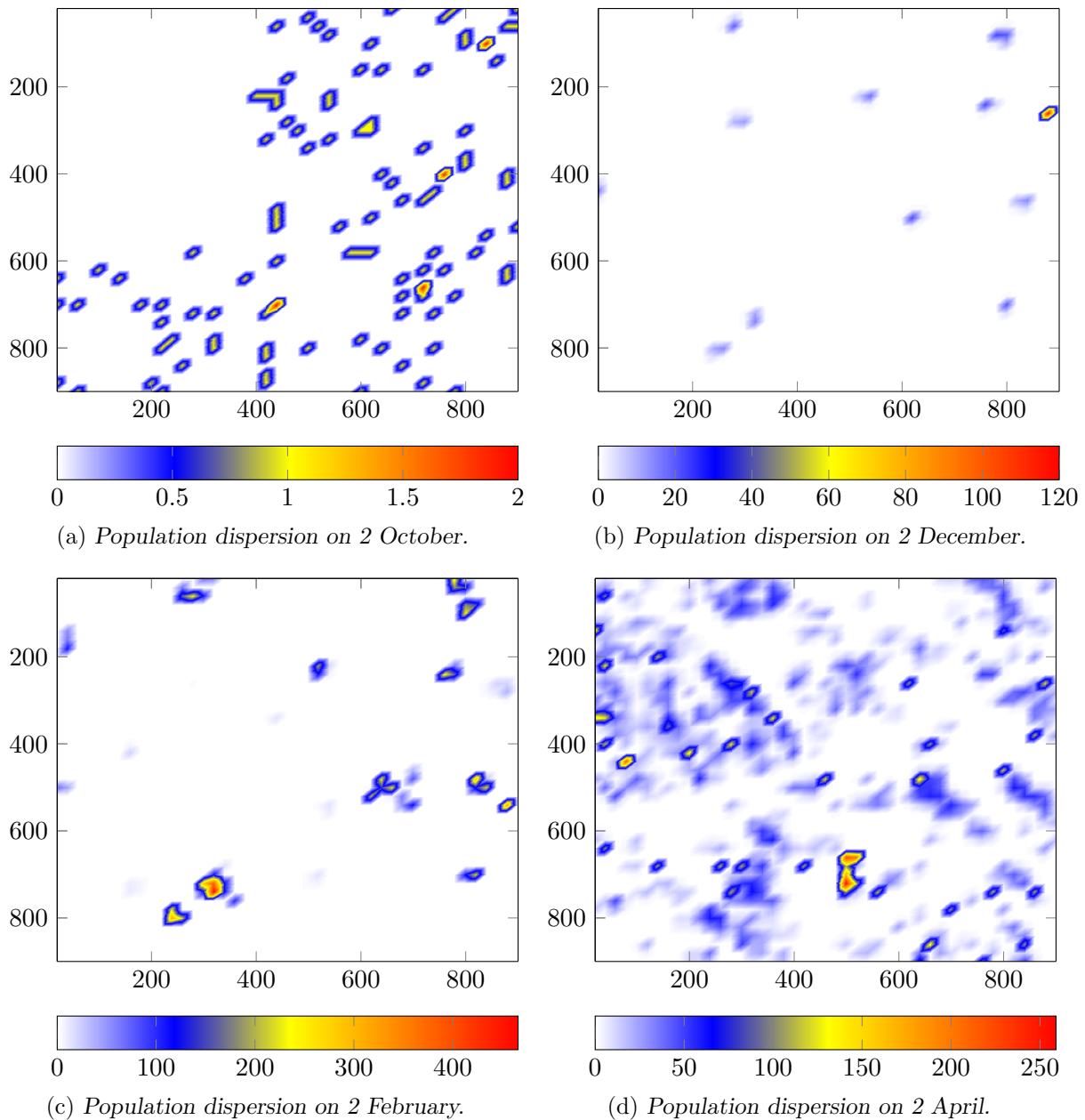


FIGURE 9.22: First full model experimental testing.

fields, as expected, in spite of the uniform initial infestation distribution. With time, however, far flying females proceeded to oviposit in these areas, causing secondary batches of agents attempting to infest the sugarcane, as pictured in Figures 9.22(b) and 9.22(c). These infestations struggled to establish themselves and thrive since sufficient growth must first occur before the pest can be supported by the stalk. For the sake of simplicity in the model, it was assumed that, as soon as a new stalk had grown to the length of a typical internode, it could support a larva for feeding. Although in nature, the process is understood not to be as binary as the model assumes, there have also been recorded cases of *E. saccharina* larvae burrowing into the plant roots in order to gain access to nutrients when there are no stalks available above ground. In light of the possibility of some larvae surviving in this manner, it was deemed feasible to include the assumption in the management process of a carrying capacity.

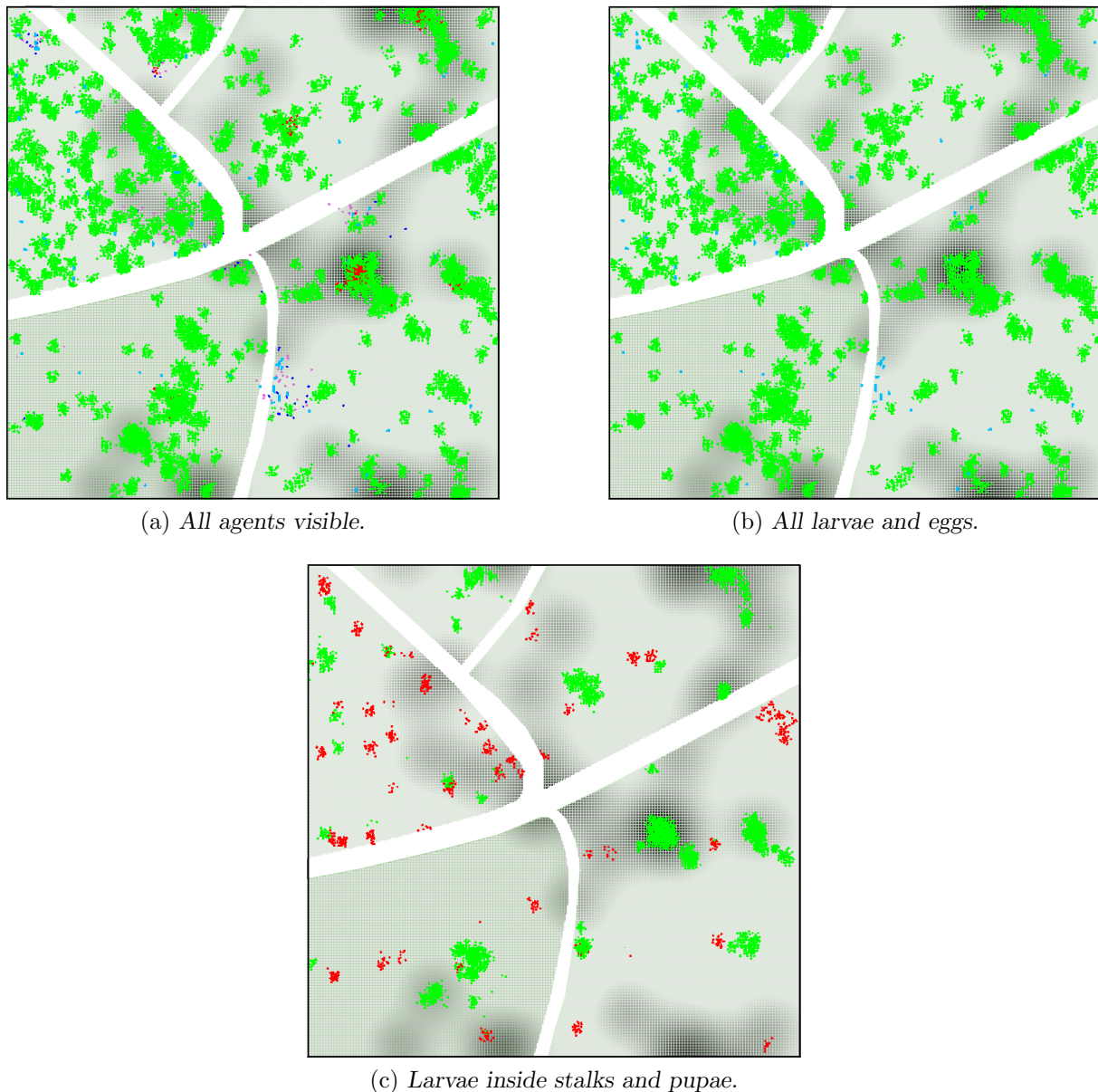


FIGURE 9.23: Screenshots of the animation output of the simulation model in the context of the experiment in Figure 9.22 upon conclusion of the simulation run with the physiological age profile superimposed over the fields.

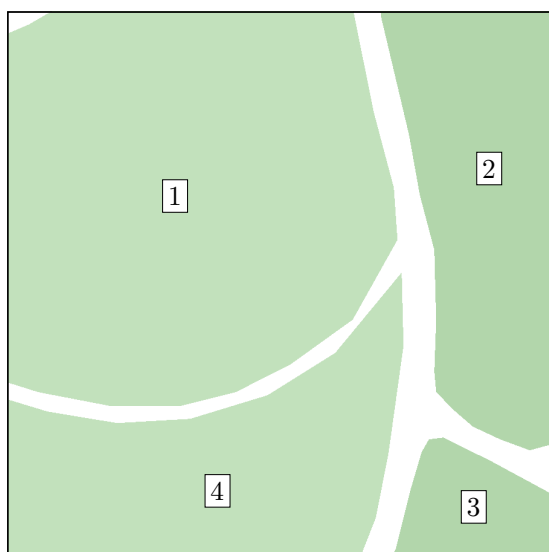
When considering Figure 9.22(d), it may be seen that, by the end of the 6-month simulation period, the infestations which occurred as a consequence of far-flying females ovipositing in fields 1 and 2 did eventually manage to progress and establish themselves. This seems plausible since, as the sugarcane grows, there are more available sites for larval boring. It may also be seen that the infestations levels in fields are widely spread, with less clumping of agents. This may be as a result of fewer oviposition sites being available in the shorter sugarcane and, as a result, eggs being slightly more widely spread, thus promoting dispersal. This is likely to be balanced by the fact that, particularly in field 1, most of the sugarcane is of a low physiological age. This means that infestations would have been hampered further by high mortality rates of larvae in this field. Furthermore, the lack of abundant physiologically aged sugarcane would have resulted in male agents continually dispersing in an attempt to come across more stressed sugarcane. This

would have resulted in small populations being infrequently static or satisfied with a certain area for infestation.

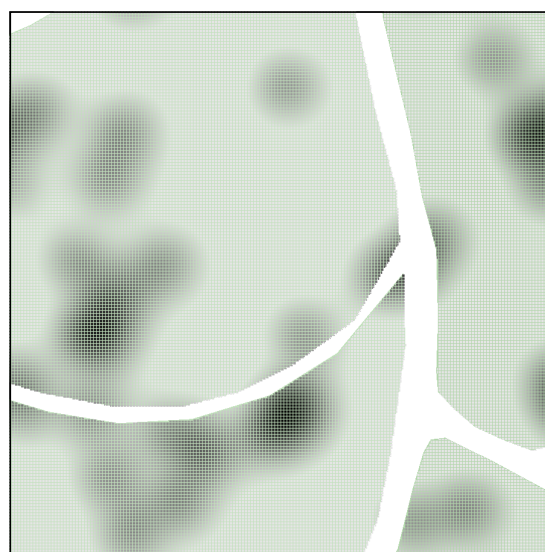
Infestations in fields 3–5 appear to be more clumped in nature, with infestation levels slightly higher than in fields 1–2, as well as larger. This may be attributed to the converse case experienced in these fields as opposed to that in fields 1 and 2. In this case, more oviposition sites are available, diminishing the need for dispersal during oviposition. Furthermore, these fields encompass a larger area of physiologically aged sugarcane. When comparing the population dispersal in Figure 9.22(d) to the physiological age profile in Figure 9.21(b), it may be seen that a number of the larger infestations correspond to areas of physiologically aged sugarcane. That said, there are many eggs and larvae residing in less stressed sugarcane. This is, once again, attributed to the random site selection for oviposition by the incoming females. It is predicted that, should the model be executed over an extended period of time, the infestations in the more stressed sugarcane would grow notably in size and intensity, whilst the growth of infestations in other areas may not be so notable.

Figure 9.23(a) shows all of the agents that exist in the simulation experiment at its conclusion, whilst, in Figure 9.23(b), the layer function was used to hide all agents barring those that reside in `Egg`, `LarvaeOutsideStalk` and `Larvae` states. This output therefore reflects what is shown in Figure 9.22(d). Interestingly, the most heavily infested areas identified as red areas in Figure 9.22(d) are not as apparent in Figure 9.23(b). This is because, in the snapshot, indication of the density of agents in a particular area is given. Furthermore, the granularity of the samples taken to generate the plot in Figure 9.22(d) groups small areas of agents together to be represented as a point in the graph. Lastly, Figure 9.23(c) shows the agents which would feasibly be able to be discovered during an in-field experiment analysis of this area, that is, the larvae inside stalks, as well as the pupae.

Figures 9.24, 9.25 and 9.26 illustrate the results of a second, similar experiment conducted on the full model. In this case, the site selected comprises four fields, none of which are initially newly harvested sugarcane.



(a) The area selected for a second simulation experiment.



(b) The randomly generated physiological age of the sugarcane in the selected area.

FIGURE 9.24: Details of the shape and physiological age of the area of sugarcane farm selected from the imported shapefile on which to run a second illustrative experiment.

In Figure 9.24(a), fields 1 and 4 began the simulation at age 13 months, whilst fields 2 and 3 started at 17 months. As may be seen from the physiological profile of the experimental site in Figure 9.24(b), field 4 in particular contained a large area of physiologically aged sugarcane and it was thus anticipated that the *E. saccharina* population would thrive in this region.

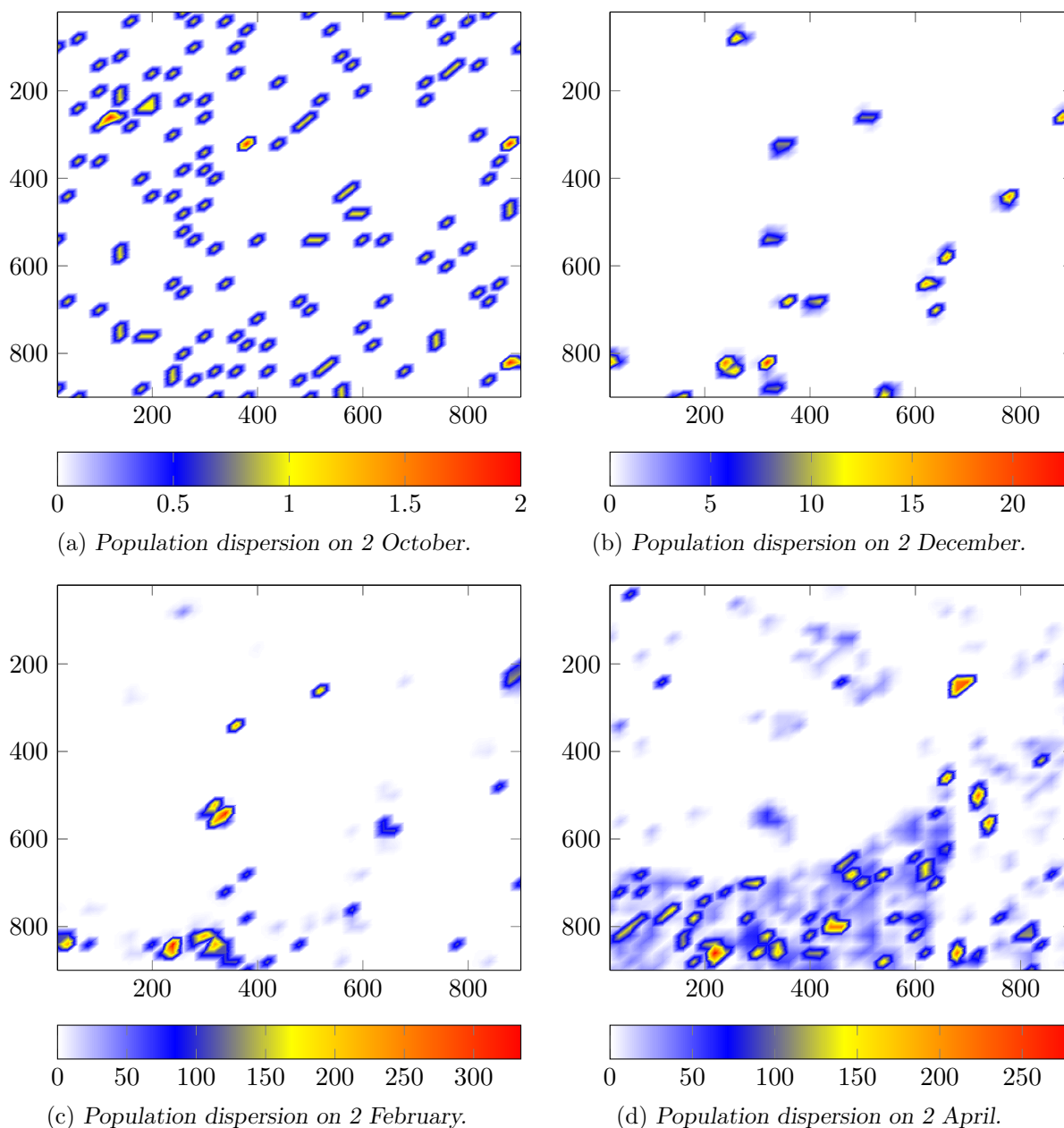
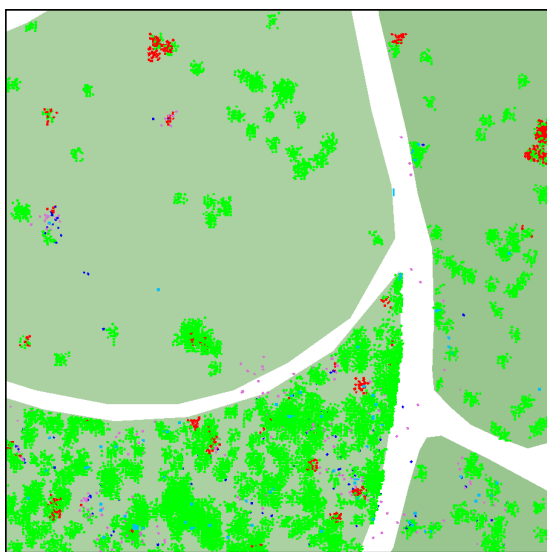


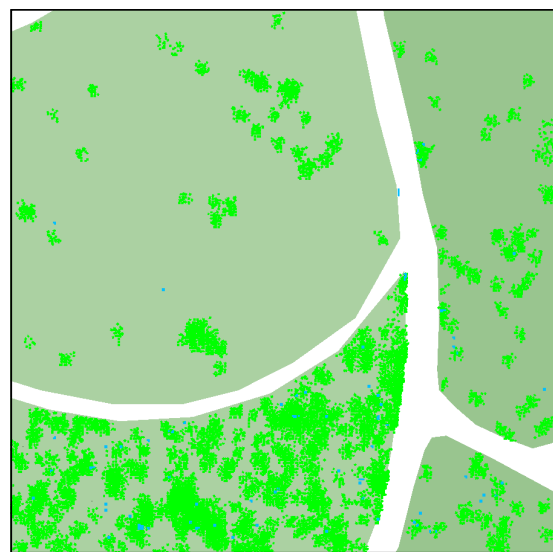
FIGURE 9.25: Second full model experimental testing.

Owing to the availability of internodes for feeding from the outset in this experiment, larvae were able to survive in all fields at the initial state, as may be seen in Figure 9.25(a). As such, a number of small infestations broke out across all the fields, some of which were already in the region of the physiologically aged sugarcane available in the field. As may be seen in Figures 9.25(c)–9.25(d), as expected, the population did indeed thrive in field 4, as well as field 3. It should be noted that, when an entering female forms part of the 10% of incoming females generated by outgoing far-fliers and randomly selects a field for oviposition, the selected is equally

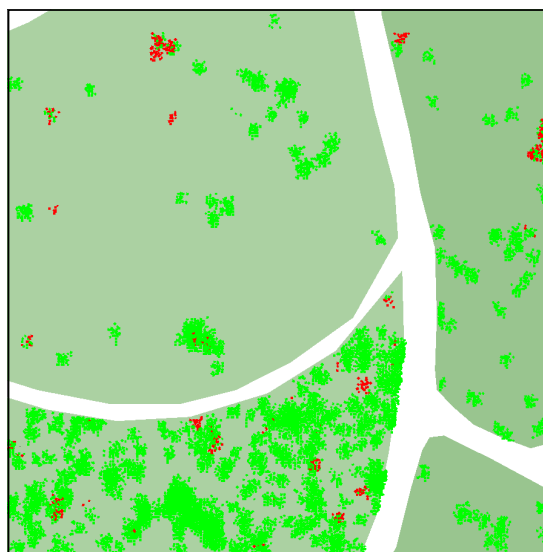
weighted for all of the fields within the simulation area. This means that, in cases where only a small portion of a field is included in the presentation area, there is an equal probability of infestations in these small areas as there is in, for example, a largely included field such as field 1. This may further result in increased infestation levels in these fields as opposed to others in the simulation, attributed to the smaller simulated area. In this experiment, however, this is not a dominant case since infestation levels in field 4, as may be seen in Figure 9.25(d), are far higher than in any of the other fields, despite covering a significant area in the presentation space. As more information is acquired through testing and in-field observation about the decision-making process and spatial working memory of female *E. saccharina* moths and, in particular, far-fliers, this implementation should be revisited so as to more accurately capture the site selection process of incoming females.



(a) All agents visible.



(b) All larvae and eggs.

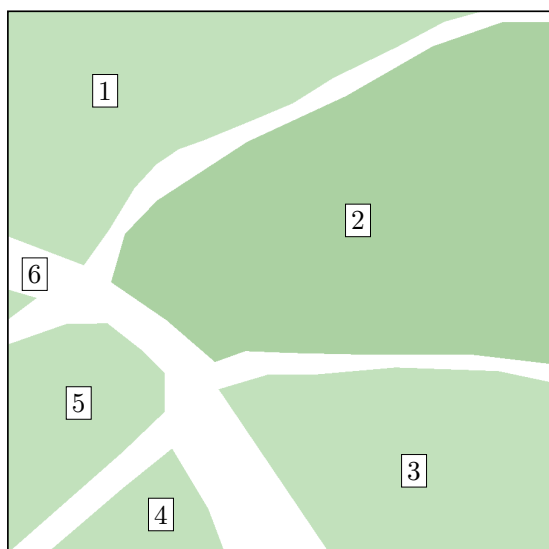


(c) Larvae inside stalks and pupae.

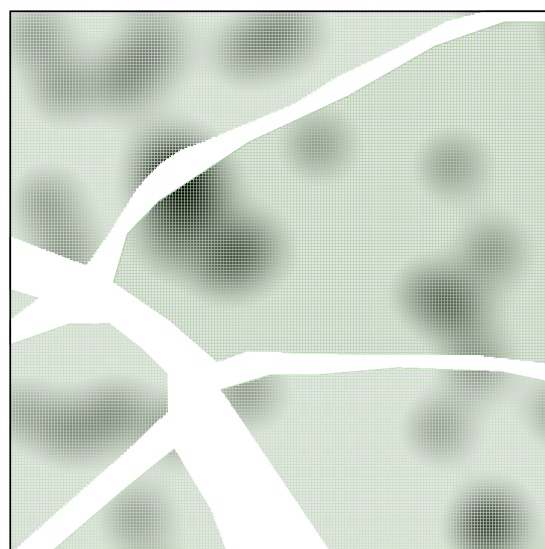
FIGURE 9.26: Screenshots of the animation output of the simulation model in the context of the experiment in Figure 9.25 upon conclusion of the simulation run with the physiological age profile superimposed over the fields.

As may be when comparing Figures 9.26(a) and 9.26(b) of the snapshots taken at the conclusion of the 6-month simulation experiment, the population in this experiment is dominated by the larval life stages (shown in green). This may mean that, on average, the sampled waiting time of incoming females was lower which means, in turn, that there is less of a time gap between the different cohorts of infesting moths. In all cases, however, many more larvae exist in the simulation space than compared to the latter life stages. This is because of the very high mortality rate experienced by neonate larvae and, as such, far fewer agents progress to the pupal and adult life stages.

For a third and final reported experiment, the sugarcane attribute file was accessed and the age of one of the fields (field 2 in Figure 9.27(a)) was increased to begin at 19 months in the simulation experiment. This meant the field would be harvested when it reached 24 months during the course of a 6-month simulation run. The surrounding fields in this experiment all began with 13-month old sugarcane. The physiological profile of the selected area is also shown in Figure 9.27(b).



(a) The area selected for a third simulation experiment.

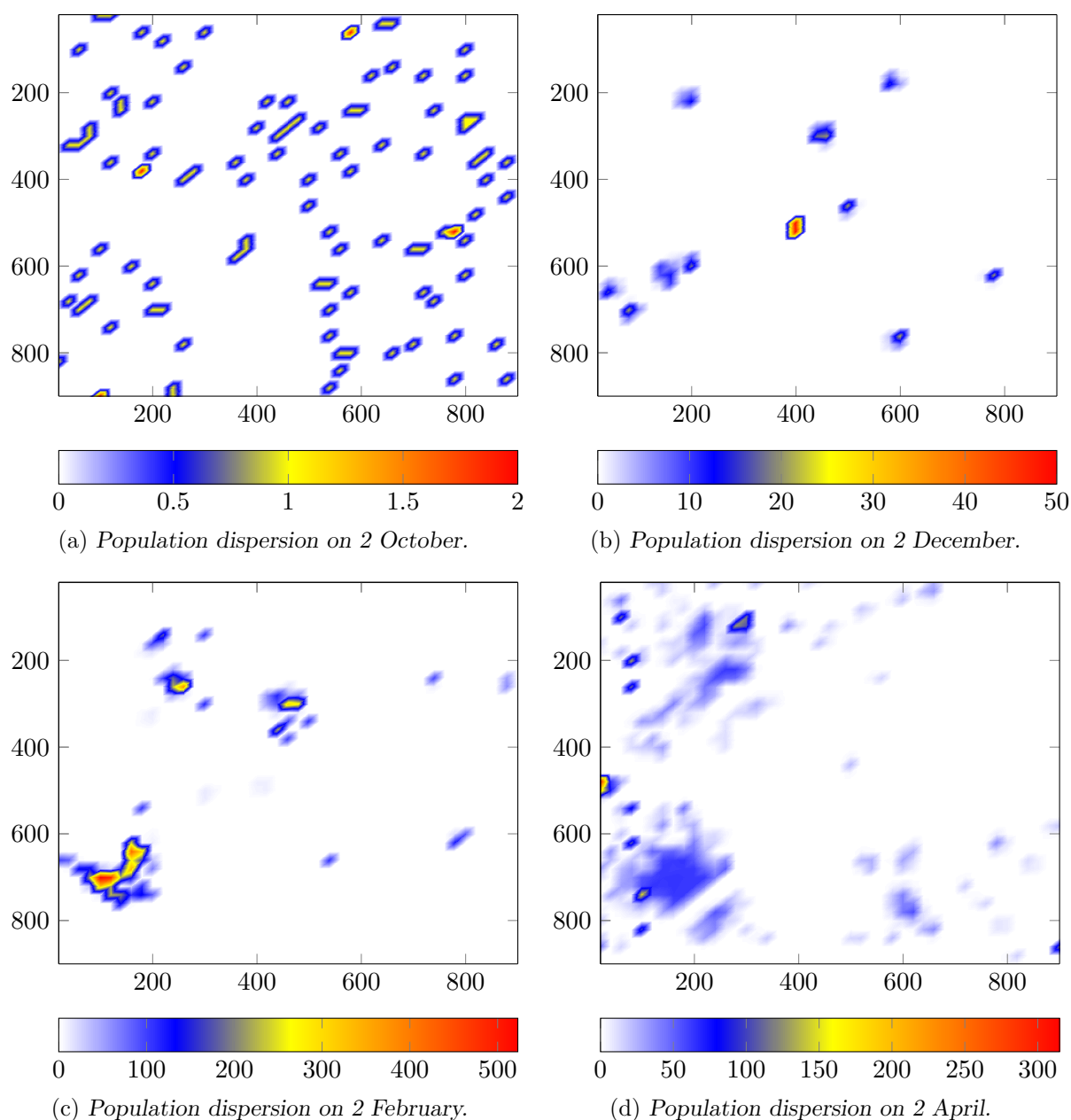


(b) The randomly generated physiological age of the sugarcane in the selected area.

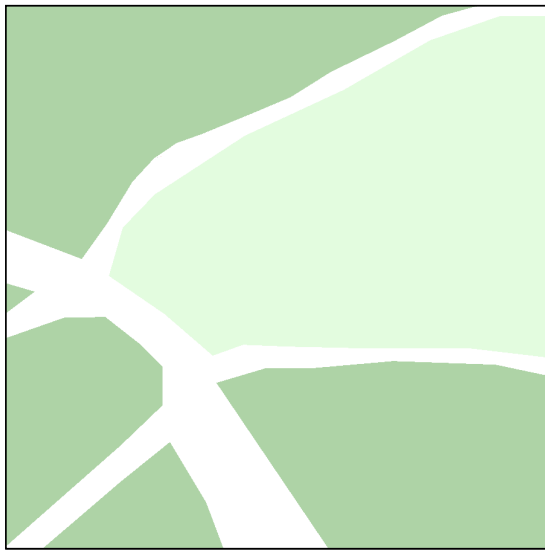
FIGURE 9.27: Details of the shape and physiological age of the area of sugarcane farm selected from the imported shapefile on which to run an illustrative experiment.

The effect of harvesting field 2 during the execution of the model may be seen in Figure 9.28. In Figures 9.28(a)–9.28(c), the population grew slowly, with various small infestations occurring within the simulated space. Harvesting took place between Figures 9.28(c) and 9.28(d), once the appropriate field had reached an age of 24 months. Harvesting operates in the same manner as explained in §4.3.2: only fields that are at the appropriate age are harvested and the population residing in that field is decreased accordingly. The option to control the harvesting schedule was removed from the simulation and it was assumed that harvesting automatically occurs on fields once they reach an age of either 12 or 24 months, depending on the selection made by the user. As may be seen in Figure 9.28(d), by the time the dispersal data were recorded in the simulation model, a number of small infestations had re-established themselves in field 2 as it began to grow for the second time, although most of the area covered by the field remained uninfested.

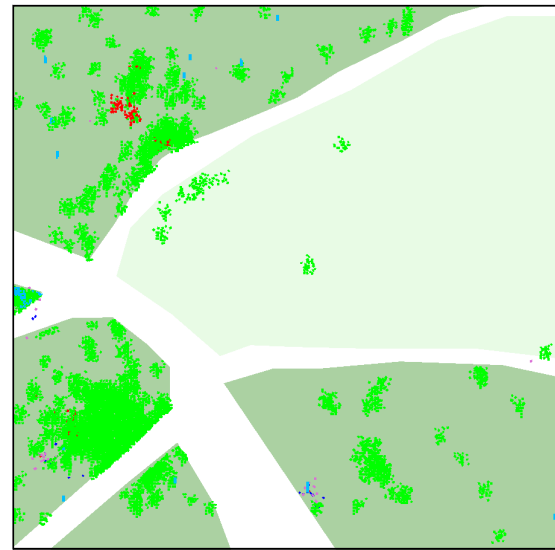
Snapshots of the population within the simulation were once again taken upon model conclu-

FIGURE 9.28: *Third full model experimental testing.*

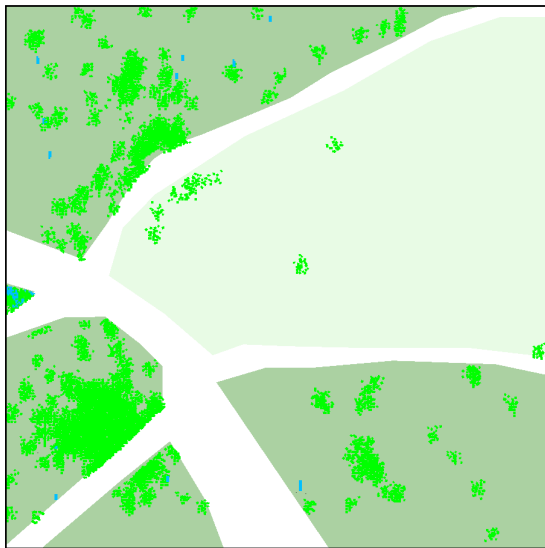
sion. Figure 9.29(a) is an additional snapshot taken of the field colours. As may be seen, the surrounding fields have all become darker with age when compared to Figure 9.27(a). Field 2, however, is significantly lighter because of the harvesting which occurred. As may be seen in Figures 9.29(b)–9.29(d), infestations are spread across all of the fields, with a notable concentration of agents occurring in field 5 within the physiologically aged sugarcane. Field 6 is also heavily infested, but this is attributed to the phenomenon discussed previously whereby an equal probability of random infestation among the fields is deemed to result in heavier infestation in fields covering a small area. It may be argued that the probability of selection of the fields in the case of random selection by incoming females should be performed as a weighted probability depending on the area covered by the field. Although sensible, this would detract from the aim of this inclusion — to promote population dynamism and facilitate outbreaks in previously uninfested



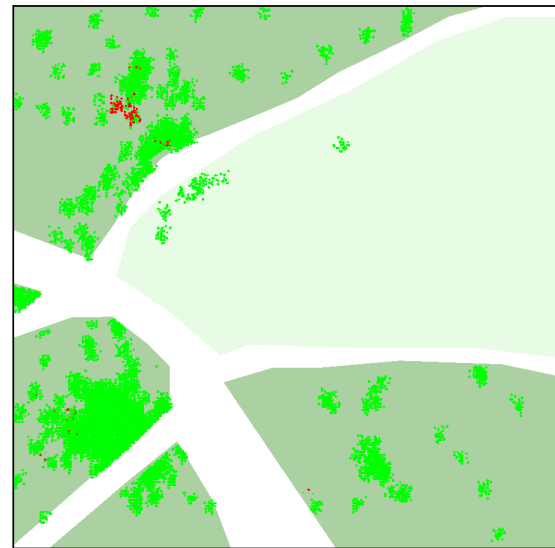
(a) *The area selected for the simulation experiment post-harvesting.*



(b) *All agents visible.*



(c) *All larvae and eggs.*




(d) *Larvae inside stalks and pupae.*

FIGURE 9.29: Screenshots of the animation output of the simulation model in the context of the experiment in Figure 9.28 upon conclusion of the simulation run with the physiological age profile superimposed over the fields.

areas and fields. It was deemed sufficient to postpone further modifications on the management of this feature for future studies, possibly assisted by further insights and observations from field experts.

9.4.5 Revised decision-support interface

In light of the alterations made to the model in §8, as well as the functionality added in this chapter, the decision-support tool presented in §7 was modified to exclude parameters that are no longer employed in the simulation model, as well as facilitate new inclusions pertaining to the

added functionalities of the simulation model. The user-specified attributes of the simulation model now include both the the selection process presented to the simulation user upon model execution in order to select the experimental site (as detailed in Figure 9.3), as well as the revised settings tab which may be accessed by the user upon clicking the  button before commencing a simulation run. The revised settings tab is shown in Figure 9.30. For the most part, functionality that is no longer applicable to the revised model was removed from the settings tab (shown in Figures 7.2(a) and 7.2(b)). A further user-specified parameter has been added which pertains to the probability with which the female moth selects an available site during oviposition. As was seen in §8.3.4, the result of a change in the probability threshold for oviposition is not as pronounced in the simulation as parameter changes in the previous oviposition algorithms were observed to be in §6.6. It is, however, deemed sensible to include this probability threshold of the new oviposition algorithm as a user-specified parameter for testing purposes, particularly as the simulated process is further refined.

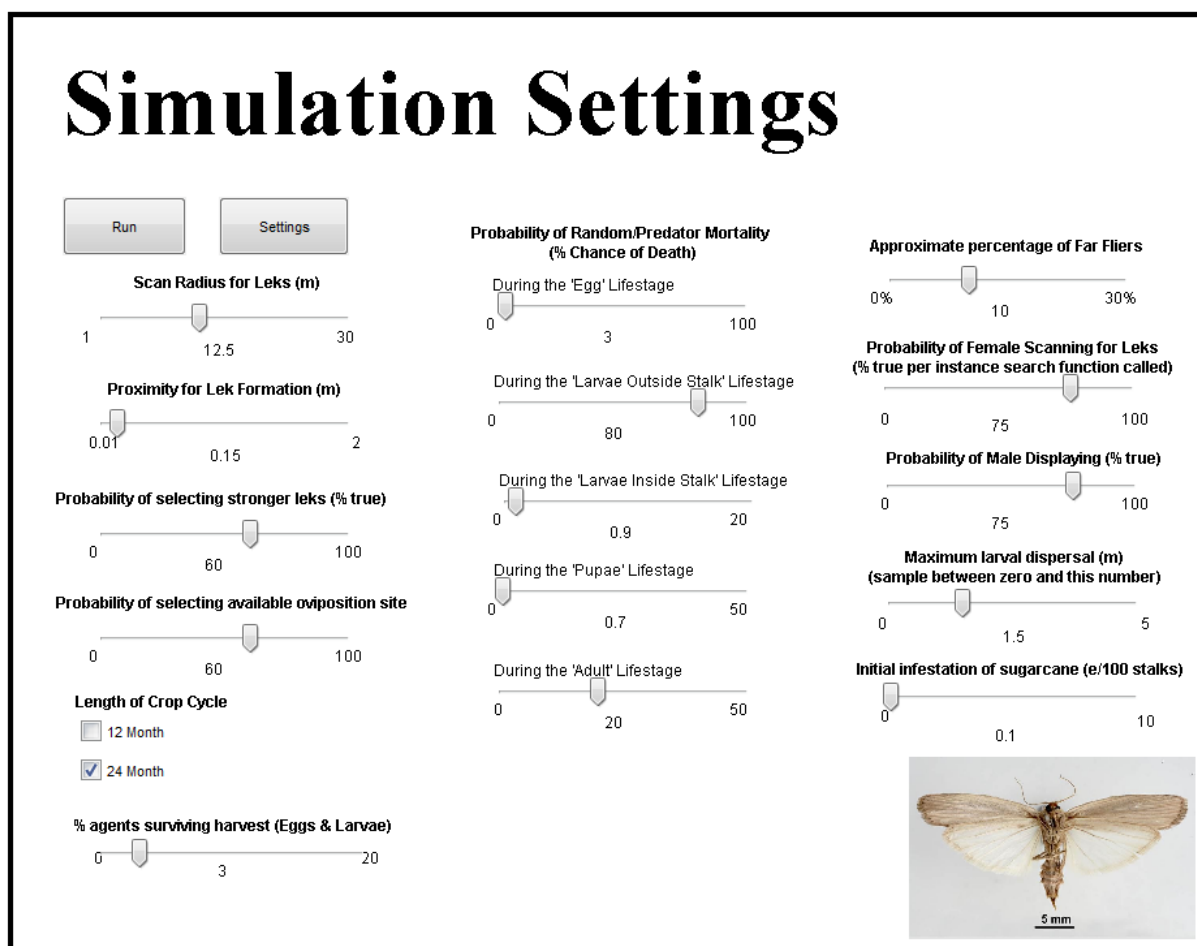


FIGURE 9.30: *The revised simulation settings tab.*

9.5 Chapter summary

In this chapter, the process design methodology and employed in order to convert sugarcane shapefiles into GIS coordinates and further import these coordinates into the simulation model of Chapters 4–8 in order to generate farm field shapes was detailed. A new process was required

in order to achieve this functionality, because no existing feature in the ANYLOGIC simulation software version employed in this study was available. From the imported field shapes, the process was explained whereby a user may make a *1ha* size selection from the farm on which to execute the model. The various considerations and difficulties involved in extracting, scaling and recreating the selection within the presentation area were discussed and the solutions presented. Furthermore, a number of other alterations to the behaviour of *E. saccharina* agents were required to integrate with the revised underlying sugarcane structure employed in the simulation model via the importation process. These changes were explained and their implementation in the model was described. The chapter closed with a number of simulation experiments conducted, using the full model in conjunction with the GIS importation process, in order to illustrate the operation of the different model components in unison. The revised settings tab, in which the user may alter parameter configurations for specific simulation runs, was finally shown in light of changes made to the simulation model during its refinement.

CHAPTER 10

Conclusion

Contents

10.1 Dissertation summary	231
10.2 The contributions of this dissertation	233
10.3 Possible future work	235
10.3.1 <i>Recommendations for future entomological research</i>	236
10.3.2 <i>Recommendations for future simulation modelling work</i>	237

This chapter opens with a summary of the contents of this dissertation in §10.1, followed by an appraisal of the contributions of the dissertation in §10.2. The chapter closes in §10.3 with a number of suggestions in respect of possible future work stemming from this study.

10.1 Dissertation summary

The introductory chapter of this dissertation, Chapter 1, contained a brief description of the development of the sugar industry of South Africa, from the first records of sugarcane cultivation in 1635 through to existing today as one of the leading producers of high quality sugar in the world. This was followed by an introduction to *Eldana saccharina* Walker, a stalk borer pest which infests sugarcane by virtue of its suitability for egg laying sites in dead leaf material, causing considerable yield losses to the industry each year. Mathematical and simulation modelling were described as possible means to better understand the pest, and develop control strategies for combatting its spread in sugarcane. Following this, a number of existing models of the pest were described. Agent-based computer simulation modelling was identified as the next logical step in the research, and the development of such a model of the pest was subsequently proposed. The scope and objectives of this study were then detailed and a brief outline of the dissertation organisation was given.

In Chapter 2, a literature review was presented of sugarcane growth and cultivation, in accordance with Dissertation Objective 1(a). A thorough discussion of the existing knowledge surrounding the biological attributes and behaviour of *E. saccharina* followed, in fulfilment of Dissertation Objective 1(b). The chapter closed with a review of existing mathematical and simulation models which have been developed to emulate *E. saccharina* behaviour, as well as an introduction to the notion of an IPM system, in partial fulfilment of Dissertation Objective 1(c).

Chapter 3 contained a review of various computer simulation modelling aspects, techniques and approaches found in the literature, in fulfilment of Dissertation Objective 1(d). The advantages and disadvantages of using simulation as a research tool were also considered, as well as the typical steps in a simulation study. In further fulfilment of Dissertation Objective 1(c), the chapter closed with a discussion on commercially available software packages used in ecological modelling, as well as existing agent-based models of insects.

In Chapter 4, a detailed description of the tools and techniques employed in this study, as well as the assumptions made in the design of a novel agent-based model of *E. saccharina*, was presented. The four primary building blocks of the model were described conceptually and then elucidated within an ANYLOGIC simulation environment. Each building block was tested against the literature and assumptions informing its construction in order to ensure that it operates as intended. The description in §4.3 pertaining to the GUI of the simulation model fulfils both Dissertation Objective 2(a) and 3, whilst the description of the modelled life cycle of *E. saccharina* in §4.4 fulfils Dissertation Objective 1(b). In §4.5, the novel manner in which the mating process of the pest was simulated fulfils Dissertation Objective 2(c) and, finally, the manner in which the oviposition and dispersal of *E. saccharina* is simulated is described in §4.6, stands in fulfilment of Dissertation Objective 2(d).

Dissertation Objective 4 was achieved in Chapter 5 where various techniques for model verification and validation were applied. Methods such as a preliminary face validation, video analysis validation and a sensitivity analysis were incorporated in view of insufficient data for use in comprehensive model critique.

The parameter variation analysis conducted in Chapter 6 recreated and described a number of different scenarios in which the simulation model was used and detailed the impact of each parameter varied to achieve each outcome. In conjunction with the description of the simulation model as a decision support tool in Chapter 7, this analysis demonstrates the flexibility and diversity of the model in view of the possibility of user-specification of the type of simulation which should be executed, in fulfilment of Dissertation Objective 5. This diverse capability of the model is intended to facilitate ‘what-if’ scenario analyses, as well as associated control measure design and analysis in an attempt to better understand the problem at hand, as well as the behaviour and interaction of *E. saccharina*.

Dissertation Objective 6 was fulfilled in Chapter 8. An expert panel met for a workshop in order to scrutinize and discuss important aspects of the simulation model to ensure their correct implementation, as well as collect the most recent, notable information developments on *E. saccharina* as attained from in-field studies and observations. Various improvements to existing model features, as well as suggested additions came out of this discussion and their incorporation into the simulation model was detailed in this chapter, concluding with a number of illustrative example experiments in order to demonstrate the functionality of the improved model.

In Chapter 9, Dissertation Objective 7 was fulfilled by the addition of functionality to the simulation model which accommodates the importation of GIS features extracted from shapefiles of sugarcane farms. The design and incorporation of this functionality was discussed, as well as the procedure followed by the user in order to make use of this feature in the simulation model. Further features which were required to be added in view of the addition of GIS data importation were also detailed in this chapter and the new module was tested in conjunction with the rest of the model.

10.2 The contributions of this dissertation

The main contributions of this dissertation towards a better understanding of the behaviour and interaction of *E. saccharina* in sugarcane, as well as developing a means of facilitating the design and testing of control measures which may be employed to suppress or eradicate the pest, are presented in this section.

Contribution 10.1 *The design of a user-friendly, agent-based simulation model of the population dynamics of *E. saccharina* in sugarcane.*

An agent-based model which incorporates all of the most notable aspects of *E. saccharina*'s life cycle and behaviour was built in the simulation software suite ANYLOGIC UNIVERSITY 7.3.1 [94]. The pest's life cycle and associated influence of temperature, its mating and oviposition process, and dispersal mechanism were modelled according to information presented in §2.3–2.6. These sub-models were linked to a suitable graphical user interface which employs a simulated underlying hectare of sugarcane in which the agents interact. The interface also accommodates various useful features and informative charts which may be evaluated during model execution. The model was also extended to incorporate the importation of GIS shapefiles in order to allow location-specific simulation experiments to be conducted by the user.

Contribution 10.2 *The incorporation of a novel manner of simulating the small-scale intricacies of the mating process of *E. saccharina*.*

In all previous models of *E. saccharina*'s population growth and dispersal, an aggregation approach was followed which excluded small-scale intricacies such as the *lek* mating and mate selection processes [31, 34, 65]. Furthermore, in these aforementioned models, the growth and spread direction and the rate at which the pest spreads were required as specified information during model construction. The agent-based model presented in this dissertation, however, makes use of the novel incorporation of a hypothetical object agent class that facilitates simulation of a mating process which accurately mimics the in-field observations of Atkinson [5] and Carnegie [17]. This allows individual agents to select mating sites and, in turn, naturally facilitate the spread of the population.

Furthermore, simulation of the competitive nature of the mate selection process incorporates the possibility of stronger offspring which may disperse over longer distances, as well as the prevalence of unmated females which are more likely to disperse over large distances if they remain unmated by the end of their first or second night after emergence [5]. From an IPM perspective, the SIT is a topical control measure currently under investigation at SASRI [62, 65, 89]. The SIT has a direct impact on the competitiveness of males and females, which may change their behaviour during the *lek* mating process. The rate at which sterility spreads within a population is directly affected by the selective, competitive mating process within the *lek*. Including the *lek* mating procedure in the simulation model described in [88] is therefore an important contribution in terms of accurately describing the imposition of the SIT on an *E. saccharina* population.

Contribution 10.3 *The design and implementation of a refined algorithm for simulating the oviposition site selection process believed to be employed by *E. saccharina* females.*

Some measure of uncertainty is always likely to exist in respect of the manner in which female *E. saccharina* moths select sites in which to oviposit their eggs. In light of this, and in collaboration with an expert panel, two initial possible oviposition site selection algorithms were discussed and analysed based on their outputs and performance. From this, an improved, biologically more accurate algorithm was incorporated into the simulation model for employment by model users.

The algorithm is based on a number of assumptions and insights from the literature and aims to produce a spread of *E. saccharina* eggs which mimics those observed in nature. Furthermore, the role of the adult male moths in dispersing the population towards areas containing suitable sites for larval survival was incorporated into the revised algorithm, based on recent insights from in-field observations. The refined oviposition algorithm also employed density-dependent dispersal, as recommended by the industry professionals and adequately captures the pest's inclination to remain in areas of suitable sugarcane for oviposition unless sites become restricted in availability due to crowding. This algorithm is deemed to be the most accurate manner of simulating the oviposition process at the current stage of entomological research.

Contribution 10.4 *The development of a decision-support tool for investigating a variety of scenarios pertaining to the behaviour and interaction of *E. saccharina*.*

The agent-based model of *E. saccharina* designed in this dissertation was implemented as a computerised simulation tool which incorporates a 'settings' tab containing the most notable, influential model parameters. These parameters may be specified by the user to simulate a variety of possible scenarios and their default values were determined during model construction with the help of industry experts. Each of the parameters included was thoroughly investigated and its consequent effect on the simulation model was described in Chapters 6 and 7. Based on the outcomes of the expert meeting described in Chapter 8, various parameters were removed from the user-defined list in light of improvements in functionality in the simulation model and, as such, became void for use in the model. It is anticipated that the remaining available parameters for user-specification may not only help future model users gain an advanced understanding and insight into the effective use and manipulation of the simulation model in terms of facilitating a variety of possible outcomes, but that it may also inform future research efforts. This may be achieved by either highlighting areas where little or no surety exists in respect of the pest's behaviour, or indicating where more advanced, accurate means of simulation modelling may be required to better emulate the behaviour of the pest within a simulated environment.

Contribution 10.5 *Facilitation of an expert panel discussion and the subsequent documentation of the most recent knowledge pertaining to *E. saccharina* presently.*

The expert panel discussion conducted during the research reported in this dissertation, and the documentation thereof, serves as a record of the most recent knowledge pertaining to *E. saccharina* available in the field. Using this knowledge, the simulation model could be refined and improved so as to better reflect the observations and behaviour experienced and witnessed in the field by industry professionals. It is anticipated that this record of knowledge and discussion will also assist future studies pertaining to the pest by making more information available to assist in modelling or researching efforts. In light of the outcomes of the discussion and the consequent improvements and additions made to the simulation model, it is now deemed to be as accurate as currently possible. All known aspects of the pest's behaviour have been incorporated into the model, with sensible, endorsed assumptions being documented at points where surety is still lacking. The manner in which model elements requiring further improvement was incorporated also aims to endorse and accommodate future research efforts stemming from this study and building on the work which has already been conducted.

Contribution 10.6 *The design and incorporation of functionality to allow for the importation of GIS sugarcane field maps into the simulation model implemented in the ANYLOGIC simulation software environment.*

By facilitating the importation of GIS sugarcane field maps and their dynamic recreation as useable model presentation elements, the simulation model thereby attains the ability to move towards specific simulation experiments conducted on selected areas of sugarcane farms. In line

with the ultimate goal of this model — to facilitate specific control measure testing on sugarcane farms — the incorporation of this module is both a significant and important contribution in the context of the study. Although limited in capacity pertaining to the size of the area that may be selected for model execution, the generic implementation methodology ensures seamless integration and the ability to execute larger models during subsequent studies, either by decreasing model computational expense, or by accessing more computing memory during experiments so as to assist in increasing model execution speeds. No such documentation of the GIS functionality offering the dynamism and integration required in this simulation model was found to exist in the ANYLOGIC simulation software version available at the time of this study [94]. This claim was endorsed by Schumann [77], one of the top ANYLOGIC experts in the world, who has recognised the functionality as a novel contribution to the software’s functionality and use.

Contribution 10.7 *Assisting in the prioritisation of future aspects of entomological research at SASRI.*

During the design of the agent-based model of *E. saccharina* presented in this dissertation, liaison with experts in the field, who are employed at SASRI, regularly took place in order to gain insight into the behaviour and biological processes pertaining to *E. saccharina* which are not documented in the literature. The experience and opinions of these experts were used to inform the assumptions made in order to include such aspects in the simulation model. It is, however, acknowledged that in order to construct a full-scale, accurate model of the pest, many aspects of the pest’s biology require further advanced insight and definitive quantification obtained from in-field experimental analysis. The questions posed during the construction of the simulation model described in this dissertation have not only assisted in uncovering areas where insufficient research exists in respect of the behaviour of *E. saccharina*, but the further analyses conducted using the model have provided insight into what are believed to be the most important aspects of the pest’s biology in terms of their potential influence on the model outputs. This may assist in prioritising future entomological research at SASRI, ensuring that the most pressing questions are afforded sufficient attention in order to progress understanding of the pest, as well as to increase the simulation model’s accuracy, in the most efficient manner possible [19].

10.3 Possible future work

As is the case with most research projects, there was not sufficient time available to investigate all of the detailed aspects of the problem considered in this dissertation. The simulation tool presented in this study is by no means complete: techniques, approaches and accompanying assumptions may all be improved upon or altered completely, and a number of additional elements may be added to increase the complexity, flexibility and reality of the model, given more time. This may result in a model that achieves better performance and/or provides more realistic results. Moreover, this study has uncovered a number of unanswered questions with respect to the biology of *E. saccharina* that must necessarily be researched and answered before a simulation model can be implementable which truly mimics the pest’s behaviour accurately. This section therefore contains a number of suggestions in respect of possible future research, from both an entomological point of view and a simulation modelling point of view, which may be pursued as follow-up work to the contributions of this dissertation.

10.3.1 Recommendations for future entomological research

Where possible, parameters employed in the simulation model presented in this dissertation were taken from previous in-field studies conducted on the growth and spread of *E. saccharina* in sugarcane. In cases where the appropriate value of an attribute was unknown or had not yet been estimated experimentally, the parameter was attributed a best estimation value as determined by experts in the field, and then made available to be modified by the user (within realistic bounds) in the settings division of the simulation model. A number of these parameters were found to have a significant effect on the output of the model and it is therefore crucial that more surety is attained with respect to their actual values in order to incorporate further biological realism into the simulation model. The following recommendations are made in respect of possible future entomological research related to *E. saccharina*, which has been identified as important during the construction of the simulation model.

Proposal 10.1 *Investigate the finer workings of the lek attraction process.*

At present, little surety exists about important aspects of the *E. saccharina* mating process, such as the maximum radius within which a female moth will search for *leks*, the manner in which the attractant pheromone strength of a *lek* dissipates over distance, and the maximum distances which may exist between males who contribute to the same *lek*. Each of these factors plays a crucial role in the simulation execution and small changes in the attributed parameter values are capable of altering the outcome of the simulation runs significantly. Further insights into or confidence intervals for these attributes of *E. saccharina* will enable more accurate simulation execution and, consequently, better quality results.

Proposal 10.2 *Measure the typical flight distances of *E. saccharina* moths.*

In the simulation model, 10% of females are assumed to be genetically inclined ‘far-fliers’ who disperse notably from their sites of emergence. Male moths are also assumed not to fly over significant distances, dispersing mainly towards nearby sites of physiologically aged sugarcane. In both cases, these assumptions are based on expert observation [19, 90] and are deemed to accurately mimic behaviour of the respective sexes of *E. saccharina* moths in nature. That said, there may be other aspects of *E. saccharina* movement that are not thoroughly understood or quantified, including the maximum distance both sexes are likely to cover during their lifespan. The assumptions made by not including this information in the simulation model have a direct impact on the growth and spread rates exhibited by the simulated *E. saccharina* population and strongly influence the dispersion and subsequent infestation within a selected area. Mark-recapture experiments are generally conducted to provide more confidence in estimating these distances and are recommended to be continued and refined for both sexes of the moth. A comprehensive simulation of the pest’s movement is critical in order to ensure a realistic response to control measures implemented within the simulation model at a later stage.

Proposal 10.3 *Investigate the spatial working memory of *E. saccharina* moths.*

In the simulation model described in this dissertation, each decision function executed by the agent is called from an independent location within that agent’s statechart. Moreover, no information attained during the execution of previous functions is stored by the agent. This presumes that agents do not possess the ability to remember any attributes of their environment, which may influence their decision making process over time. Although a valid assumption in this case, this may not necessarily be true in nature. In particular, the female is especially suspected of possessing some form of memory influence during

the oviposition site selection process. The ability to make influenced decisions based on experience or exposure to the environment in which the female moth finds itself is likely to alter the behaviour of the agents notably in the simulation. It is thus recommended that entomological research be conducted in an effort to determine to what degree behavioural traits or decisions are influenced by the spatial working memory of *E. saccharina*.

Proposal 10.4 *Refine understanding of the role of plant volatiles in population dynamics.*

Experts believe that volatiles released by sugarcane may guide the dispersal of the pest within a field [72, 90]. More specifically, stressed sugarcane is deemed to release volatiles that are perceived by male moths who, in turn, move toward these stressed areas to *lek*. In the simulation model in this dissertation, this process is captured, but is rudimentary in its application. The extent to which these volatiles affect male movement is neither quantified nor fully understood, and is simply applied as a direct movement, under some probabilistic conditions, by the male towards stressed sugarcane falling within a randomly selected radius. Furthermore, different varieties of sugarcane may release volatiles with differing intensities or attraction capabilities, thereby altering the process further and causing different dispersal between heterogeneous fields by males. It is recommended that the finer details of the role of plant volatiles be investigated in a comprehensive study in order to ensure that this aspect may be properly implemented and captured in later models.

Proposal 10.5 *Further investigate the decision making process employed during the selection of oviposition sites.*

The refined oviposition site selection algorithm implemented in this dissertation is deemed to effectively re-create the spread of eggs and the resulting population growth as it is observed in nature. That said, the actual simulated process is greatly simplified and does not include a number of possible influences in light of a lack of understanding of their role in the real-life process. Aspects such as volatiles which are believed to be released by stalks, indicating infestation, which may influence the selection process of the female should be investigated and their role incorporated into the oviposition site selection process so as to allow for more accurate capturing of the process.

10.3.2 Recommendations for future simulation modelling work

The simulation model presented in this dissertation is a novel approach to simulating and predicting the population dynamics of *E. saccharina* in that it employs an agent-based modelling approach. This approach offers several tools and methodologies which may be used to construct models accommodating various methods of controlling agent behaviour, as well as incorporating a variety of aspects pertaining to the agent modelled. The diversity of approaches and available tools inevitably means that there may exist alternate, computationally more efficient manners in which to model a population of *E. saccharina* moths than the approach described in this dissertation. Furthermore, given sufficient time, insight and accompanying successful entomological research results, virtually all quantifiable aspects of the pest may be incorporated into the simulation model to influence its execution. The following recommendations are made in respect of possible future alternative approaches toward simulating the population dynamics of *E. saccharina*, as well as additional inclusions in or improvements to the simulation model presented in this dissertation.

Proposal 10.6 *Construct an agent-based model which is not statechart-controlled in nature.*

In ANYLOGIC, several approaches may be employed to construct an agent-based model. The simulation model in this dissertation employs a statechart-based approach where each different stage in the agent's life cycle is modelled as a different state in which different actions are executed and specific functions are called. The agent then progresses through these states based on a number of conditional transitions which determine the path followed through the statechart.

An alternative approach to such a statechart-based model is the notion of a continually-executing event, similar to a main script in a typical computer programme. This event then contains interrupt functions which execute at the appropriate times, according to the position of the lifecycle in which the agent finds itself. Execution of these interrupt sequences have to be controlled by parameters which track agent behaviour and development, thereby possibly increasing model complexity and parameter tracking. That said, such an approach is likely to facilitate a less computationally expensive simulation model requiring fewer code sequences and transition checks than in the case of a statechart approach.

Proposal 10.7 *Partition the simulation space so as to decrease computational complexity.*

Another means of decreasing simulation complexity whilst still maintaining acceptable model execution speeds may be to partition the simulation space into smaller areas and execute simulations of these divisions on different machines. Such parallelisation of the simulation will require intensive, linked programming in order to ensure that agents can progress fluidly through the absorbing and emitting boundaries of each simulated area into adjacent space. Once implemented, however, such an approach will result in a platform for intensive, thorough model design without running the risk of overburdening the single processor on which the simulation model is executed.

Proposal 10.8 *Employ dynamic systems modelling for generalised aspects of the pest.*

A further possibility in pursuit of a reduction in the computational burden of the simulation model is to identify general, common aspects of the pest's behaviour (such as the life stages preceding mating, oviposition and dispersal) and an attempt to incorporate these attributes in a 'global' manner. This alteration will reduce the number of individual events executed per time step in the simulation should, in turn, reduce the computational expense of executing simulations. Future research may even include consideration of a dual-simulation approach — employing a dynamic systems approach (in which individual attributes of elements are maintained despite a generalised modelling approach) for common aspects of agents and then diverging to an agent-based approach where individuality becomes more prevalent in the simulation.

Proposal 10.9 *Employ cloud computing facilities to execute the simulation model.*

It is believed that, in the near future, the ANYLOGIC simulation software package will be available with the ability to run models from a cloud-based server [77]. In this case, a very large amount of *random access memory* (RAM) may be accessed by the model through the cloud. It is believed that, at present, it is the RAM capabilities of a typical processor which cause the model to execute at a slower rate. By employing more RAM, the model may be extendedable in size without major modifications to its construction, allowing for more comprehensive studies on entire sugarcane farms to be conducted.

Proposal 10.10 *Include the effects of all weather conditions on pest behaviour.*

At present, the simulation model only accommodates the effect of temperature change on the pest development and behaviour. In reality, however, all weather conditions affect the population. Rain may, for example, inhibit mating altogether, while wind is likely to

increase the distances travelled by agents. As further research is conducted on the pest's reaction to certain weather conditions, the accuracy of the model may be improved by incorporating all weather aspects that are thought to affect its behaviour and development.

Proposal 10.11 *Include comprehensive management of simulated sugarcane growth, physiological age and variety.*

In the simulation model presented in this dissertation, only the stalk height growth equation of the CANEGRO model [40] is used to simulate sugarcane growth. Furthermore, the physiological profile of the sugarcane is determined by a randomised profile generated per simulation run.

It is recommended that an independent study be conducted whereby the existing sugarcane simulation models may be compiled and extended to serve as a functional basis on which to operate this model of *E. saccharina* biological behaviour. During the progression of the current study, it has become increasingly evident that the underlying sugarcane plays a prolific role in a number of the pest's decision making processes, thereby notably altering its development and dispersal. Furthermore, different varieties of sugarcane provide different nutrient profiles and maturation properties to *E. saccharina* larvae. Stressed sugarcane is also deemed to affect the development of *E. saccharina* in a different manner than normal sugarcane, as was included only in a simplified manner in this simulation model. For this reason, and in line with the pursuit of a comprehensive model of *E. saccharina* population dynamics in sugarcane, it is important to include these subtle differences comprehensively in the underlying sugarcane in order to affect the behaviour and development of larvae accordingly.

Proposal 10.12 *Perform a real-world case study using the simulation model.*

In general, the data available for validation and testing of the simulation model put forward in this dissertation are sporadic and specific in nature. It is therefore difficult to fully determine the flexibility and accuracy of the model by comparison to real-world studies which have been conducted under specific conditions. Liaisons with staff at SASRI have been undertaken to possibly perform a specific case study in the future [19]. Such a study would only be sensible following several important further inclusions in the model aimed at increasing its realism and accuracy. Thereafter, the model may be applied to a typical industry problem.

Proposal 10.13 *Compare the output of the agent-based simulation model with the solution of a corresponding, existing mathematical model.*

In order to determine the degree to which the agent-based simulation model presented in this dissertation corresponds with existing mathematical models of the pest, it is proposed that a comparison between the outputs of the simulation model and other mathematical models be conducted. Most existing mathematical models simulate specific scenarios of the pest, such as the imposition of a control measure, and it may therefore be necessary to replicate the respective control measure in the simulation model in order to allow for result comparison. This may also be achieved in the form of a real-world case study investigated using both modelling techniques in order to determine whether or not the outcomes of each model agree and, where relevant, link to the existing data for case studies.

Proposal 10.14 *Develop a deterministic version of the simulation model and perform a sensitivity analysis on this model version.*

As discussed in §5.3.3, the outcomes of the sensitivity analysis performed on the simulation model were largely inconclusive owing to the many random processes which occur, either to promote dynamism and realistic behaviour in the *E. saccharina* behaviour, or as inclusions

where little surety exists pertaining to appropriate parameter values. In light of this, it is proposed that a study be conducted in which the random model elements are removed, as best possible, in order to promote consistent behaviour. The model could then be executed a number of times within a range of differing input parameter values. The corresponding changes in the model may then be more definitively attributed to the fluctuated parameter and, as such, a better understanding of the sensitivity and corresponding role of each of the parameters may be achieved.

Proposal 10.15 *Implement a population control measure in the simulation model.*

The pinnacle of this area of research is also the reason behind its inception — to accurately simulate the detailed imposition of a population control measure on *E. saccharina* and reliably determine the most important parameters pertaining to the expected success of the control measure preceding in-field testing. Several important improvements previously detailed in this chapter should, however, be included before any reliable conclusions may be drawn in respect of the success of a control measure, but it certainly remains the destination to which the simulation model research effort should be aimed. Such simulation of control measures will greatly assist researchers and farmers in the sector towards gaining confidence and answering questions pertaining to attributes of these strategies before they are taken to in-field experiments in an effort to save both time and money in line with the objectives of this study.

References

- [1] ADDISON MF, 2016, Researcher (Conservation Ecology) at *Stellenbosch University, Stellenbosch*, [Personal Communication], Contactable at mfa@sun.ac.za.
- [2] AKOL AM, 2011, *Eldana saccharina Walker, 1865 — African Sugarcane Stalkborer*, [Online], [Cited October 2014], Available from http://keys.lucidcentral.org/keys/v3/eafrinet/maize_pests/key/maize_pests/Media/Html/Eldana_saccharina_Walker_1865_-_African_Sugarcane_Stalkborer.htm.
- [3] APPLIED BIOMATHEMATICS, 2013, *RAMAS IRM Version 2.0*, [Online], [Cited November 2014], Available from <http://www.ramas.com/indexEnv.htm>.
- [4] ATKINSON PR, 1979, *Distribution and natural hosts of Eldana saccharina Walker in Natal, its oviposition sites and feeding patterns*, Proceedings of the South African Sugar Technologists Association, **42(2)**, pp. 111–115.
- [5] ATKINSON PR, 1981, *Mating behaviour and activity patterns of Eldana saccharina Walker (Lepidoptera: Pyralidae)*, Journal of the Entomological Society of Southern Africa, **44(2)**, pp. 265–280.
- [6] ATKINSON PR, 1980, *On the biology, distribution and natural host-plants of Eldana saccharina Walker (Lepidoptera: Pyralidae)*, Journal of the Entomological Society of Southern Africa, **43(2)**, pp. 171–194.
- [7] ATKINSON PR, 1984, *Seasonal cycles of Eldana borer in relation to available control measures*, Proceedings of the South African Sugar Technologists Association, **58**, pp. 165–167.
- [8] ATKINSON PR & CARNEGIE AJM, 1989, *Population dynamics of the sugarcane borer, Eldana saccharina Walker (Lepidoptera: Pyralidae), in Natal, South Africa*, Bulletin of Entomological Research, **79(1)**, pp. 61–80.
- [9] ATKINSON PR, CARNEGIE AJM & SMAILL RJ, 1981, *A history of the outbreaks of Eldana saccharina Walker in Natal*, Proceedings of the South African Sugarcane Technologists Association, **55**, pp. 111–115.
- [10] BANKS J, 1996, *Interpreting simulation software checklists*, ORMS Today, **22(3)**, pp. 74–78.
- [11] BANKS J, 1998, *Principles of simulation*, pp. 3–30, in *Handbook of simulation: Principles, methodology, advances, applications, and practice*, Wiley-Interscience, New York (NY).
- [12] BANKS J, CARSON J, NELSON BL & NICOL DM, 2001, *Verification and validation of simulation models*, pp. 367–397 in FABRYCKY WJ & MIZA JH (EDS), *Discrete-event system simulation*, Prentice-Hall, Upper Saddle River (NJ).

- [13] BERRY SD, LESLIE GW, SPAULL VW & CADET P, 2010, *Within-field damage and distribution patterns of the stalk borer, Eldana saccharina (Lepidoptera: Pyralidae), in sugarcane and a comparison with nematode damage*, Bulletin of Entomological Research, **100(04)**, pp. 373–385.
- [14] BEZUIDENHOUT C, 2000, *A model review and proposed mechanistic tiller model for the Canegro sugarcane crop model*, MTech Thesis, Technikon Natal, Durban.
- [15] BOGGS CL, 1990, *A general model of the role of male-donated nutrients in female insects' reproduction*, American Naturalist, **136(5)**, pp. 598–617.
- [16] BORSHCHEV A & FILIPPOV A, 2004, *From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools*, Proceedings of the 22nd International Conference of the System Dynamics Society, **22**.
- [17] CARNEGIE AJM, 1974, *A recrudescence of the borer Eldana saccharina Walker (Lepidoptera: Pyralididae)*, Proceedings of the South African Sugar Technologists Association, **48**, pp. 107–110.
- [18] CARNEGIE AJM, 1981, *Combating Eldana saccharina Walker: A progress report*, Proceedings of the South African Sugar Technologists Association, **55**, pp. 116–119.
- [19] CONLONG DE, 2014–2016, Senior Entomologist (Crop biology resource centre) at the South African Sugarcane Research Institute, Mount Edgecombe, [Personal Communication], Contactable at Des.Conlong@sugar.org.za.
- [20] CONLONG DE, 1990, *A study of pest-parasitoid relationships in natural habitats: An aid towards the biological control of Eldana saccharina (Lepidoptera: Pyralidae) in sugarcane*, Proceedings of the South African Sugar Technologists Association, **64**, pp. 111–115.
- [21] CONLONG DE, 2001, *Biological control of indigenous African stemborers: What do we know?*, International Journal of Tropical Insect Science, **21(04)**, pp. 267–274.
- [22] CONLONG DE, KASL B & BYRNE M, 2007, *Independent kids, or motherly moms? Implications for integrated pest management of Eldana saccharina Walker (Lepidoptera: Pyralidae)*, Proceedings of the International Society of Sugar Cane Technologists, **26**, pp. 787–796.
- [23] CONLONG DE & RUTHERFORD RS, 2009, *Conventional and new biological and habitat interventions for integrated pest management systems: Review and case studies using Eldana saccharina Walker (Lepidoptera: Pyralidae)*, pp. 241–261 in PESHIN R & DHAWAN A (EDS), *Integrated pest management: innovation-development process*, Springer, New York (NY).
- [24] DOWNING KJ, LESLIE G & THOMSON JA, 2000, *Biocontrol of the sugarcane borer Eldana saccharina by expression of the Bacillus thuringiensis cry1Ac7 and Serratia marcescens chiA genes in sugarcane-associated bacteria*, Applied and Environmental Microbiology, **66(7)**, pp. 2804–2810.
- [25] EINHORN MD, 2012, *An evaluation of the efficiency of self-organising versus fixed traffic signalling paradigms*, MSc Thesis, Stellenbosch University, Stellenbosch.
- [26] GHANEM B & AHUJA N, 2013, *Modeling dynamic swarms*, Computer Vision and Image Understanding, **117(1)**, pp. 1–11.
- [27] GOEBEL FR & WAY MJ, 2003, *Investigation of the impact of Eldana saccharina (Lepidoptera: Pyralidae) on sugarcane yield in field trials in Zululand*, Proceedings of the South African Sugar Technologists Association, **77**, pp. 256–265.

- [28] GOLDSIM TECHNOLOGY GROUP, 2014, *Simulation of ecological and biological systems*, [Online], [Cited November 2014], Available from <http://www.goldsim.com/Web/Solutions/EnvironmentalSystems/Ecological/>.
- [29] GRAS R, DEVAURS D, WOZNIAK A & ASPINALL A, 2009, *An individual-based evolving predator-prey ecosystem simulation using a fuzzy cognitive map as the behavior model*, *Artificial Life*, **15(4)**, pp. 423–463.
- [30] GU W & NOVAK RJ, 2009, *Agent-based modelling of mosquito foraging behaviour for malaria control*, *Transactions of the Royal Society of Tropical Medicine and Hygiene*, **103(11)**, pp. 1105–1112.
- [31] HEARNE JW, VAN COLLER LM & CONLONG DE, 1994, *Determining strategies for the biological control of a sugarcane stalk borer*, *Ecological Modelling*, **73(1)**, pp. 117–133.
- [32] HEATHCOTE RJ, 1984, *Insecticide testing against Eldana saccharina Walker*, *Proceedings of the South African Sugar Technologists Association*, **47(2)**, pp. 154–158.
- [33] HIEBELER D, 1994, *The swarm simulation system and individual-based modeling*, Santa Fe Institute, Working Paper, Available from <http://www.santafe.edu/media/workingpapers/94-11-065.pdf>.
- [34] HORTON PM, HEARNE JW, APALOO J, CONLONG DE, WAY MJ & UYS P, 2002, *Investigating strategies for minimising damage caused by the sugarcane pest Eldana saccharina*, *Agricultural Systems*, **74(2)**, pp. 271–286.
- [35] HORTON PM, 2008, *A simulation modelling approach to aid research into the control of a stalk-borer in the South African sugar industry*, PhD Thesis, University of KwaZulu-Natal, Pietermaritzburg.
- [36] HOWSTUFFWORKS, 2008, *Sugarcane*, [Online], [Cited October 2014], Available from <http://science.howstuffworks.com/life/botany/sugarcane-info.htm>.
- [37] ILLOVO PLANTERS GROUP, *Eston*, Contactable at ipg@estonclub.co.za.
- [38] INFORMS, 2015, *Award recipients: The South African National Defense Force*, [Online], [Cited September 2015], Available from <https://www.informs.org/Recognize-Excellence/Award-Recipients/South-African-National-Defense-Force2>.
- [39] INFORMS, 2015, *Inform Online: The Edelman Award*, [Online], [Cited September 2015], Available from <https://www.informs.org/Sites/Getting-Started-With-Analytics/The-Edelman-Award>.
- [40] INMAN-BAMBER NG, 1991, *A growth model for sugar-cane based on a simple carbon balance and the CERES-Maize water balance*, *South African Journal of Plant and Soil*, **8(2)**, pp. 93–99.
- [41] ITTMANN HW, 2015, IFORS Representative of the *Operations Research Society of South Africa*, Contactable at hittmann01@gmail.com.
- [42] JANSSEN MA, 2005, *Agent-based modelling*, [Online], [Cited April 2014], Available from http://isecoeco.org/pdf/agent_based_modeling.pdf.
- [43] KAREIVA PM, 1983, *Local movement in herbivorous insects: Applying a passive diffusion model to mark-recapture field experiments*, *Oecologia*, **57(3)**, pp. 322–327.
- [44] KAWAHARA AY & BARBER JR, 2015, *Tempo and mode of antibat ultrasound production and sonar jamming in the diverse hawkmoth radiation*, *Proceedings of the National Academy of Sciences*, **112(20)**, pp. 6407–6412.
- [45] KAZANCI C, 2007, *EcoNet: A new software for ecological modeling, simulation and network analysis*, *Ecological Modelling*, **208(1)**, pp. 3–8.

- [46] KEEPING MG & RUTHERFORD RS, 2004, *Resistance mechanisms of South African sugarcane to the stalk borer *Eldana saccharina* (Lepidoptera: Pyralidae): A review*, Proceedings of the African Sugarcane Technology Association, **78**, pp. 307–312.
- [47] KEIL S, GU H & DORN S, 2001, *Response of *Cydia pomonella* to selection on mobility: laboratory evaluation and field verification*, Ecological Entomology, **26(5)**, pp. 495–501.
- [48] KINDL J, KALINOVÁ B, ČERVENKA M, JÍLEK M & VALTEROVÁ I, 2011, *Male moth songs tempt females to accept mating: the role of acoustic and pheromonal communication in the reproductive behaviour of *Aphomia sociella**, PloS one, **6(10)**, pp. 264–276.
- [49] KLEYNHANS E, MITCHELL K, CONLONG D & TERBLANCHE J, 2014, *Evolved variation in cold tolerance among populations of *Eldana saccharina* (Lepidoptera: Pyralidae) in South Africa*, Journal of evolutionary biology, **27(6)**, pp. 1149–1159.
- [50] KUNESCH G, ZAGATTI P, LALLEMAND J, DEBAL A & VIGNERON J, 1981, *Structure and synthesis of the wing gland pheromone of the male African sugarcane borer: *Eldana saccharina* (Wlk.) (Lepidoptera, Pyralidae)*, Tetrahedron Letters, **22(52)**, pp. 5270–5274.
- [51] KUUSSAARI M, SACCHERI I, CAMARA M & HANSKI I, 1998, *Allee effect and population dynamics in the Glanville fritillary butterfly*, Oikos, pp. 384–392.
- [52] LAW AM, 2009, *How to build valid and credible simulation models*, Proceedings of the 2009 Winter Simulation Conference (WSC), pp. 24–33.
- [53] LAW AM, 1990, *How to build valid, credible and appropriately detailed simulation models*, Industrial Engineering, **22(4)**, pp. 16–17.
- [54] LAW AM & KELTON WD, 1991, *Simulation modeling and analysis*, McGraw-Hill, New York (NY).
- [55] LESLIE GW, 1988, *The identification and importance of predators of *Eldana saccharina* (Lepidoptera: Pyralidae)*, Proceedings of the South African Sugar Technologists' Association, pp. 122–126.
- [56] LESLIE GW, 2003, *Impact of repeated applications of alpha-cypermethrin on *Eldana saccharina* (Lepidoptera: Pyralidae) and on arthropods associated with sugarcane*, Sugar Cane International, pp. 16–22.
- [57] LINSTONE HA & TUROFF M, 1975, *The Delphi method: techniques and applications*, Addison-Wesley Reading, MA.
- [58] MAHAKRUSHI, 2011, *Growth morphology of sugarcane*, [Online], [Cited October 2014], Available from <http://www.mahakrushi.com/2011/07/>.
- [59] MINAR N, BURKHART R, LANGTON C & ASKENAZI M, 1996, *The swarm simulation system: A toolkit for building multi-agent simulations*, Santa Fe Institute, Santa Fe (CA).
- [60] MITROFF II & TUROFF M, 1975, *Philosophical and methodological foundations of Delphi*, pp. 17–35 in LINSTONE HA & TUROFF M (EDS), *The Delphi method: techniques and applications*, Addison-Wesley Reading, MA.
- [61] MUDAVANHU P, 2015–2016, Weed Biocontrol Researcher at ARC-PPRI, Vredenburg Campus, Stellenbosch, [Personal Communication], Contactable at Mudavanhup@arc.agric.za.
- [62] MUDAVANHU P, CONLONG D & ADDISON P, 2012, *Impact of mass-rearing and gamma radiation on the thermal tolerance of *Eldana saccharina* Walker (Lepidoptera: Pyralidae)*, Proceedings of the South African Sugar Technologists Association, **85**, pp. 139–143.
- [63] PFEIFFER J, 1968, *New Look at Education: Systems Analysis in our Schools and Colleges*, pp. 16–32, in *Decision Making in Action*, ERIC.

- [64] POTGIETER L, VAN VUUREN JH & CONLONG DE, 2013, *Modelling the effects of the sterile insect technique applied to Eldana saccharina Walker in sugarcane*, *ORION*, **28(2)**, pp. 59–84.
- [65] POTGIETER L, van VUUREN JH & CONLONG DE, 2013, *A reaction-diffusion model for the control of Eldana saccharina Walker in sugarcane using the sterile insect technique*, *Ecological Modelling*, **250**, pp. 319–328.
- [66] POTGIETER L, 2013, *A mathematical model for the control of Eldana saccharina Walker using the sterile insect technique*, PhD Thesis, Stellenbosch University, Stellenbosch.
- [67] PRATT SC, SUMPTER DJ, MALLON EB & FRANKS NR, 2005, *An agent-based model of collective nest choice by the ant Temnothorax albipennis*, *Animal Behaviour*, **70(5)**, pp. 1023–1036.
- [68] PRICE PW, 1984, *Insect ecology (2nd Ed)*, John Wiley & Sons, Inc, New York, (NY).
- [69] PRINGLE K, 2016, Senior Research Associate at *Stellenbosch University, Stellenbosch*, [Personal Communication], Contactable at k1p@sun.ac.za.
- [70] PYTHON SOFTWARE FOUNDATION, 2016, *Python applications*, [Online], [Cited May 2016], Available from <https://www.python.org/about/apps/>.
- [71] ROBINSON S, 2004, *Simulation: The practice of model development and use*, John Wiley & Sons, Chichester.
- [72] RUTHERFORD S, 2016, Senior Pathologist (Crop biology resource centre) at *the South African Sugarcane Research Institute, Mount Edgecombe*, [Personal Communication], Contactable at stuart.rutherford@sugar.org.za.
- [73] SARGENT RG, 2005, *Verification and validation of simulation models*, Proceedings of the 2005 Winter Simulation Conference (WSC), pp. 130–143.
- [74] SCHANK J, 2013, *Swarmfest 2013*, [Online], [Cited November 2014], Available from <http://www.agent-based-models.com>.
- [75] SCHANK JC, 2008, *The development of locomotor kinematics in neonatal rats: An agent-based modeling analysis in group and individual contexts*, *Journal of Theoretical Biology*, **254(4)**, pp. 826–842.
- [76] SCHUMACHER P, WEYENETH A, WEBER DC & DORN S, 1997, *Long flights in Cydia pomonella L. (Lepidoptera: Tortricidae) measured by a flight mill: Influence of sex, mated status and age*, *Physiological Entomology*, **22(2)**, pp. 149–160.
- [77] SCHUMANN B, 2016, Managing consultant and simulation expert at *decisionLab, United Kingdom*, [Personal Communication], Contactable at benjamin.schumann@gmail.co.za.
- [78] SCHUMANN B, 2016, *Marriage of agents and GIS shapefiles*, AnyLogic Conference 2014, [Online], [Cited March 2014], Available from <http://www.anylogic.com/alc-2014>.
- [79] SKUTSCH M & HALL D, 2016, *Delphi: potential uses in educational planning*, [Online], [Cited April 1973], Available from <http://files.eric.ed.gov/fulltext/ED084659.pdf>.
- [80] SNYMAN SJ, BAKER C, HUCKETT BI, MCFARLANE SA, VAN ANTWERPEN T, BERRY S, OMARJEE J, RUTHERFORD R & WATT DA, 2008, *South African Sugarcane Research Institute: Embracing biotechnology for crop improvement research*, *Sugar Tech* (2008), **10(1)**, pp. 1–13.
- [81] SOUTH AFRICAN SUGAR ASSOCIATION, 2014, *SASA: Facts and figures*, [Online], [Cited April 2013], Available from http://www.sasa.org.za/sugar_industry/FactsandFigures.aspx.

- [82] SOUTH AFRICAN SUGAR ASSOCIATION, 2014, *SASA: Industry overview*, [Online], [Cited April 2013], Available from http://www.sasa.org.za/sugar_industry/IndustryOverview.aspx.
- [83] STEPHENS PA, SUTHERLAND WJ & FRECKLETON RP, 1999, *What is the Allee effect?*, *Oikos*, pp. 185–190.
- [84] STRAY BJ, 2010, *Tactical sugarcane harvest scheduling*, PhD Thesis, Stellenbosch University, Stellenbosch.
- [85] SYSTEMS E & INSTITUTE R, 2016, *ESRI Shapefile Technical Description*, Santa Fe Institute, Working Paper, [Online], [Cited May 1998], Available from <https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.
- [86] Van COLLER L M, 1992, *Optimum biological control strategies for a problem in the sugar industry — A mathematical modelling approach*, MSc Thesis, University of Natal, Pietermaritzburg.
- [87] Van VUUREN BJ, POTGIETER L & van VUUREN JH, 2015, *An agent-based simulation model describing the lek mating process of Eldana saccharina Walker*, Proceedings of the 44th Annual Conference of the Operations Research Society of South Africa, pp. 1–11.
- [88] Van VUUREN BJ, POTGIETER L & van VUUREN JH, 2014, *Prerequisites for the design of an agent-based model for simulating the population dynamics of Eldana saccharina Walker*, Proceedings of the 43rd Annual Conference of the Operations Research Society of South Africa, pp. 62–70.
- [89] WALTON AJ, 2011, *Radiation biology of Eldana saccharina Walker (Lepidoptera: Pyralidae)*, MSc Thesis, Stellenbosch University, Stellenbosch.
- [90] WAY MJ, 2014–2016, Entomologist (Crop biology resource centre) at the *South African Sugarcane Research Institute, Mount Edgecombe*, [Personal Communication], Contactable at mike.way@sugar.org.za.
- [91] WAY MJ, 1995, *Developmental biology of the immature stages of Eldana saccharina Walker (Lepidoptera: Pyralidae)*, Proceedings of the South African Sugar Technologists Association, **69**, pp. 83–86.
- [92] WINSTON WL, 2004, *Operations research: Applications and algorithms*, Duxbury Press, Boston (MA).
- [93] WU K.-M, LU Y.-H, FENG H.-Q, JIANG Y.-Y & ZHAO J.-Z, 2008, *Suppression of cotton bollworm in multiple crops in China in areas with Bt toxin-containing cotton*, *Science*, **321(5896)**, pp. 1676–1678.
- [94] XJ TECHNOLOGIES COMPANY, 2010, *AnyLogic 7 user's guide*, [Online], [Cited October 2014], Available from <http://www.xjtek.com/>.
- [95] YADAV K, 2009, *Botanical description of sugarcane*, [Online], [Cited October 2014], Available from <http://agropedia.iitk.ac.in/content/botanical-description-sugarcane>.
- [96] YOUSUF MI, 2016, *The Delphi technique*, [Online], [Cited April 2006], Available from <http://drimran.articlealley.com/delphi-technique-112396.html>.
- [97] YOUSUF MI, 2007, *Using experts' opinions through Delphi technique*, Practical assessment, research & evaluation, **12(4)**, pp. 1–8.

APPENDIX A

Pilot sensitivity analysis data

This appendix contains the complete set of sensitivity analysis data for the pilot study described in §5.3.3. The data are presented in four tables. Tables A.1 and A.2 contain the damage indicator values, whilst Tables A.3 and A.4 represent the population level data collected during the set of experiments employing the **F** MovingWindow and **F** FindMatureCane oviposition site selection algorithms, respectively.

The measures used, as described in §5.3.3, are repeated here for convenience. In Tables A.1 and A.2, the mean infestation level (I_{μ_total}) is a measure of the average number of larvae which infested each stalk in the simulated space over the duration of each entire simulation run. This differs from the current infestation level ($I_{\mu_current}$) which represents the average number of larvae present in each stalk at the time instant when the simulation ended. The standard deviation of the total infestation over the duration of the simulation run (I_{σ_total}) is a comparison between the infestation of each individual stalk over the course of the simulation run to the (I_{μ_total}) of that run, and gives an indication to what degree the infestation was consistent throughout the simulated space. The current standard deviation, ($I_{\sigma_current}$), performs the same calculation, but exclusively for the infestation present at the time instant when the simulation is terminated. The ‘e/100s’ value represents the number of larvae and eggs currently present in the simulation as a measure per 100 simulated sugarcane stalks, whilst the ‘max. e/100s’ value represents the highest such infestation levels experienced over the course of the simulation.

Similarly, the record of the ‘Current eggs & larvae’ and ‘Max. eggs & larvae’ represent the total agents in those states at the termination of the simulation and the highest recorded level during the course of the simulation run, respectively. It was chosen to record the number of agents in the **Egg** and **Larvae** states because the greatest changes in the population numbers in the model were observed during these states. The carrying capacity, in particular, resulted in large numbers of larvae failing to mature to the **Pupae** state and, since the sensitivity analysis aimed to test which parameter had the most significant influence on aspects of the simulation model such as this, the number of agents in these states was deemed important to record.

Finally, a maximum total population level reached is recorded, as indicated by the ‘Max population’ column.

Varied parameter	$I_{\mu_{total}}$	$I_{\sigma_{total}}$	$I_{\mu_{current}}$	$I_{\sigma_{current}}$	e/100s	max e/100s
None	0.019	0.362	0.014	0.282	1.392	2.910
	0.054	0.644	0.038	0.477	3.818	10.125
	0.111	0.949	0.081	0.721	8.118	21.427
	0.044	0.593	0.032	0.462	3.202	7.239
	0.053	0.640	0.039	0.491	3.867	10.480
	0.082	0.794	0.059	0.595	5.887	13.787
	0.037	0.531	0.026	0.386	2.618	7.015
	0.068	0.729	0.049	0.545	4.862	12.395
	0.062	0.693	0.042	0.505	4.208	13.859
0.115	0.943	0.083	0.712	8.296	16.868	
SR = 10m	0.008	0.241	0.005	0.158	0.506	1.059
	0.029	0.470	0.020	0.347	1.982	5.811
	0.088	0.842	0.062	0.626	6.217	17.428
	0.118	0.991	0.081	0.708	8.042	26.298
	0.078	0.776	0.056	0.584	5.586	13.950
	0.056	0.647	0.042	0.509	4.225	4.397
	0.080	0.772	0.054	0.556	5.355	13.445
	0.085	0.818	0.061	0.612	6.094	11.958
	0.021	0.387	0.014	0.277	1.348	4.604
0.119	0.990	0.085	0.728	8.427	23.381	
SR = 15m	0.041	0.573	0.027	0.395	2.690	7.647
	0.030	0.477	0.020	0.338	2.009	5.473
	0.078	0.791	0.059	0.619	5.867	14.582
	0.109	0.942	0.079	0.705	7.842	22.618
	0.106	0.954	0.073	0.685	7.232	25.078
	0.086	0.803	0.063	0.613	6.320	13.265
	0.038	0.549	0.029	0.427	2.855	6.195
	0.082	0.793	0.056	0.573	5.566	15.259
	0.003	0.141	0.001	0.079	0.097	0.318
0.013	0.303	0.009	0.224	0.892	1.977	
Prox. = 0.05m	0.016	0.308	0.008	0.188	0.783	1.345
	0.014	0.308	0.009	0.222	0.909	1.550
	0.008	0.236	0.005	0.157	0.485	0.857
	0.012	0.278	0.007	0.187	0.700	1.029
	0.003	0.144	0.001	0.072	0.112	0.252
	0.007	0.209	0.003	0.123	0.305	0.711
	0.013	0.307	0.007	0.202	0.704	1.394
	0.019	0.353	0.011	0.244	1.132	1.591
	0.014	0.305	0.008	0.214	0.844	1.510
0.006	0.190	0.003	0.121	0.317	0.730	
Prox. = 0.25m	0.169	1.183	0.122	0.895	12.180	54.365
	0.245	1.438	0.171	1.058	17.080	65.793
	0.120	0.974	0.092	0.773	9.125	27.064
	0.162	1.165	0.123	0.909	12.296	37.967
	0.078	0.783	0.059	0.616	5.888	20.772
	0.194	1.251	0.142	0.949	14.136	42.633
	0.161	1.147	0.120	0.886	11.941	42.798

	0.058	0.690	0.045	0.555	4.438	14.164
	0.091	0.836	0.066	0.640	6.608	26.652
	0.094	0.861	0.070	0.666	6.983	20.547
PD = 2.75m	0.041	0.551	0.030	0.425	2.955	7.599
WI = 2.75%	0.057	0.646	0.041	0.487	4.097	6.788
	0.027	0.458	0.020	0.358	2.028	5.168
	0.026	0.416	0.018	0.310	1.789	2.945
	0.048	0.617	0.036	0.481	3.618	7.595
	0.056	0.645	0.040	0.491	3.972	5.251
	0.028	0.463	0.019	0.333	1.900	5.397
	0.042	0.574	0.029	0.428	2.870	7.084
	0.032	0.489	0.024	0.370	2.359	5.508
	0.048	0.620	0.035	0.479	3.448	9.146
PD = 2.25m	0.088	0.822	0.064	0.622	6.337	14.652
WI = 2.25%	0.054	0.642	0.039	0.485	3.840	12.222
	0.015	0.312	0.010	0.234	1.011	2.601
	0.094	0.867	0.071	0.679	7.052	14.088
	0.054	0.624	0.038	0.466	3.818	10.094
	0.143	1.057	0.103	0.796	10.249	17.702
	0.062	0.664	0.045	0.506	4.480	10.575
	0.057	0.649	0.041	0.492	4.123	8.868
	0.036	0.521	0.025	0.389	2.528	6.872
	0.014	0.311	0.009	0.221	0.915	1.974
CQT = 1.05	0.169	1.155	0.114	0.818	11.385	36.898
(Good)	0.046	0.600	0.032	0.444	3.234	8.618
	0.069	0.715	0.049	0.538	4.887	15.612
	0.025	0.417	0.018	0.314	1.757	3.342
	0.069	0.745	0.050	0.565	5.005	13.130
	0.068	0.738	0.047	0.546	4.733	10.025
	0.016	0.339	0.011	0.247	1.057	3.208
	0.087	0.825	0.062	0.617	6.198	16.276
	0.086	0.835	0.061	0.612	6.052	16.538
	0.041	0.578	0.031	0.440	3.075	8.123
CQT = 1.175	0.074	0.773	0.052	0.559	5.158	14.706
(Excellent)	0.045	0.585	0.033	0.443	3.247	6.853
	0.012	0.294	0.009	0.220	0.858	1.565
	0.086	0.821	0.062	0.616	6.161	16.902
	0.015	0.338	0.010	0.230	0.962	2.590
	0.148	1.085	0.105	0.807	10.465	23.972
	0.019	0.384	0.012	0.267	1.242	4.418
	0.070	0.746	0.049	0.556	4.914	11.242
	0.026	0.440	0.020	0.340	1.945	3.739
	0.021	0.386	0.015	0.299	1.525	2.881
<i>P_{mortality}</i>	0.117	0.950	0.083	0.695	8.252	22.811
Egg = 3.3%	0.018	0.355	0.010	0.223	1.025	2.212
LarvaeOut = 12.7%	0.018	0.341	0.013	0.257	1.248	2.261
LarvaeIn = 1%	0.032	0.483	0.022	0.360	2.239	5.364
Pupae = 0.8%	0.024	0.416	0.016	0.294	1.557	3.622
Adult = 22%	0.063	0.702	0.043	0.504	4.302	11.774
	0.035	0.498	0.023	0.352	2.268	4.703

	0.024	0.403	0.017	0.314	1.675	3.345
	0.021	0.407	0.013	0.272	1.295	5.227
	0.062	0.683	0.043	0.502	4.295	14.140
<i>P_{mortality}</i>	0.018	0.351	0.013	0.281	1.298	2.480
Egg = 2.7%	0.065	0.700	0.051	0.562	5.061	8.865
LarvaeOut = 10.4%	0.039	0.548	0.026	0.392	2.635	7.502
LarvaeIn = 0.8%	0.058	0.659	0.043	0.510	4.251	11.371
Pupae = 0.6%	0.029	0.449	0.022	0.357	2.178	4.481
Adult = 18%	0.134	1.054	0.096	0.778	9.528	28.236
	0.045	0.576	0.033	0.444	3.340	8.262
	0.010	0.255	0.005	0.152	0.485	0.955
	0.022	0.418	0.014	0.282	1.398	3.488
	0.053	0.640	0.040	0.493	3.942	8.674
$\lambda_{Erlang} = 1.25$	0.009	0.258	0.006	0.190	0.598	1.277
	0.068	0.724	0.048	0.538	4.766	12.232
	0.031	0.471	0.021	0.339	2.082	5.440
	0.044	0.567	0.031	0.431	3.077	7.587
	0.030	0.463	0.020	0.336	1.987	4.263
	0.012	0.287	0.008	0.223	0.806	1.762
	0.028	0.463	0.020	0.361	1.999	4.471
	0.064	0.708	0.045	0.528	4.513	13.075
	0.059	0.667	0.042	0.490	4.159	11.255
	0.023	0.428	0.015	0.308	1.522	3.870
$\lambda_{Erlang} = 1.75$	0.085	0.823	0.063	0.632	6.284	15.856
	0.009	0.233	0.006	0.189	0.608	0.860
	0.083	0.803	0.062	0.627	6.224	12.748
	0.073	0.730	0.053	0.556	5.321	7.747
	0.033	0.513	0.022	0.363	2.170	6.786
	0.110	0.939	0.078	0.696	7.822	19.225
	0.046	0.588	0.032	0.438	3.217	7.284
	0.070	0.736	0.053	0.583	5.325	11.183
	0.013	0.301	0.008	0.223	0.833	1.596
	0.061	0.690	0.042	0.499	4.211	11.282
<i>P_{scanning} = 0.65%</i>	0.022	0.398	0.013	0.258	1.326	3.338
<i>P_{displaying} = 0.65%</i>	0.021	0.393	0.013	0.278	1.318	3.006
	0.044	0.584	0.030	0.423	3.019	7.695
	0.017	0.339	0.012	0.245	1.157	2.562
	0.025	0.427	0.016	0.288	1.600	3.409
	0.035	0.513	0.023	0.368	2.321	5.365
	0.061	0.674	0.044	0.511	4.398	6.958
	0.016	0.326	0.011	0.250	1.098	2.061
	0.061	0.684	0.044	0.507	4.342	11.273
	0.034	0.502	0.024	0.370	2.353	4.495
<i>P_{scanning} = 0.85%</i>	0.183	1.194	0.131	0.894	13.068	30.252
<i>P_{displaying} = 0.85%</i>	0.105	0.923	0.070	0.650	6.989	25.991
	0.211	1.315	0.149	0.960	14.809	40.472
	0.106	0.945	0.075	0.702	7.487	24.178
	0.006	0.212	0.003	0.126	0.296	0.831
	0.089	0.838	0.066	0.646	6.630	16.093
	0.037	0.544	0.026	0.407	2.587	8.178

	0.152	1.125	0.110	0.845	10.995	34.503
	0.078	0.806	0.055	0.596	5.468	16.505
	0.049	0.628	0.033	0.445	3.288	11.013
$LD_{max} = 0.25\text{m}$	0.016	0.357	0.010	0.255	0.988	2.122
	0.015	0.319	0.008	0.232	0.801	1.822
	0.009	0.262	0.005	0.174	0.472	1.655
	0.005	0.180	0.002	0.121	0.231	0.554
	0.013	0.313	0.007	0.209	0.719	2.565
	0.015	0.330	0.010	0.257	0.966	2.270
	0.011	0.287	0.007	0.201	0.649	1.322
	0.021	0.401	0.013	0.295	1.320	3.772
	0.010	0.269	0.005	0.189	0.543	1.258
	0.010	0.278	0.006	0.203	0.582	1.562
$LD_{max} = 0.75\text{m}$	0.086	0.804	0.063	0.607	6.315	17.743
	0.090	0.837	0.065	0.625	6.528	18.952
	0.191	1.182	0.149	0.942	14.822	24.952
	0.032	0.488	0.023	0.371	2.325	6.779
	0.156	1.122	0.115	0.848	11.481	35.069
	0.195	1.210	0.146	0.922	14.534	36.219
	0.053	0.613	0.041	0.482	4.083	7.897
	0.152	1.111	0.111	0.831	11.064	39.644
	0.160	1.111	0.122	0.861	12.185	27.353
	0.192	1.214	0.144	0.929	14.351	28.028

TABLE A.1: Complete damage indication data collected during the sensitivity analysis using the 'Moving Window' oviposition site selection algorithm.

Varied parameter	$I_{\mu_{total}}$	$I_{\sigma_{total}}$	$I_{\mu_{current}}$	$I_{\sigma_{current}}$	e/100s	max e/100s
None	0.407	1.491	0.352	1.355	35.105	35.105
	0.171	0.930	0.147	0.845	14.615	15.547
	0.119	0.742	0.099	0.671	9.840	9.840
	0.161	0.880	0.137	0.804	13.597	14.221
	0.169	0.919	0.142	0.826	14.152	15.269
	0.112	0.753	0.092	0.666	9.196	9.812
	0.131	0.853	0.108	0.753	10.815	11.940
	0.726	2.006	0.634	1.831	63.212	63.212
	0.293	1.238	0.251	1.123	25.007	25.007
	0.149	0.866	0.124	0.778	12.400	12.415
SR = 10m	0.206	0.998	0.173	0.907	17.223	17.223
	0.811	2.072	0.701	1.889	69.877	69.877
	0.213	1.030	0.180	0.933	17.918	18.875
	0.193	0.983	0.165	0.893	16.401	16.401
	0.244	1.128	0.208	1.025	20.748	22.421
	0.166	0.903	0.141	0.828	14.098	14.098
	0.182	0.961	0.147	0.850	14.671	14.671
	0.240	1.098	0.203	0.990	20.287	20.903
	0.143	0.811	0.118	0.731	11.723	11.855
0.138	0.834	0.116	0.756	11.600	12.311	
SR = 15m	0.138	0.811	0.116	0.738	11.575	11.575
	0.312	1.247	0.266	1.133	26.474	26.474
	0.007	0.175	0.004	0.137	0.418	0.470
	0.308	1.230	0.260	1.118	25.912	25.912
	0.307	1.372	0.263	1.221	26.219	27.724
	0.216	1.034	0.187	0.947	18.615	18.984
	0.229	1.098	0.194	0.988	19.354	20.092
	0.175	0.924	0.146	0.834	14.558	14.558
	0.288	1.209	0.245	1.101	24.406	24.406
	0.286	1.219	0.243	1.103	24.220	24.220
Prox. = 0.05m	0.170	0.934	0.136	0.815	13.550	13.866
	0.083	0.614	0.061	0.526	6.098	6.098
	0.029	0.361	0.019	0.296	1.898	1.898
	0.181	0.927	0.141	0.807	14.015	14.015
	0.040	0.423	0.027	0.354	2.740	3.056
	0.058	0.518	0.046	0.456	4.574	4.574
	0.022	0.320	0.015	0.262	1.530	1.638
	0.017	0.277	0.011	0.223	1.083	1.083
	0.119	0.743	0.091	0.647	9.089	9.089
	0.031	0.378	0.022	0.318	2.188	2.429
Prox. = 0.25m	0.661	2.133	0.592	1.950	59.062	79.780
	0.701	1.957	0.627	1.813	62.482	62.482
	0.120	0.783	0.106	0.727	10.602	10.948
	0.483	1.579	0.426	1.454	42.419	42.419
	0.644	2.056	0.570	1.876	56.874	68.006
	0.341	1.314	0.302	1.222	30.102	31.242
	0.314	1.242	0.274	1.147	27.318	28.209

	0.397	1.459	0.351	1.346	35.002	36.275
	0.254	1.140	0.228	1.065	22.683	22.683
	0.277	1.161	0.244	1.073	24.307	25.144
CSR = 13.5m	0.339	1.326	0.301	1.223	30.053	33.879
WS = 0.26rad	0.284	1.179	0.251	1.092	25.003	25.902
	0.150	0.570	0.132	0.527	13.196	13.705
	0.400	1.464	0.354	1.347	35.297	38.832
	0.271	1.087	0.241	1.009	24.009	25.377
	0.219	0.896	0.192	0.828	19.157	19.814
	0.289	1.111	0.256	1.026	25.565	27.309
	0.578	2.339	0.512	2.166	51.070	53.616
	0.179	0.710	0.158	0.657	15.767	16.539
	0.384	1.506	0.340	1.391	33.894	36.535
CSR = 16.5m	0.054	0.215	0.048	0.199	4.743	4.994
WS = 0.79rad	0.387	1.402	0.343	1.295	34.184	36.129
	0.638	2.465	0.564	2.278	56.264	59.989
	0.696	2.605	0.616	2.405	61.392	65.493
	0.340	1.291	0.301	1.192	29.984	32.388
	0.241	1.082	0.213	1.003	21.228	21.746
	0.111	0.726	0.099	0.673	9.822	10.142
	0.612	2.455	0.542	2.271	54.068	57.252
	0.205	1.037	0.175	0.936	17.424	18.406
	0.108	0.725	0.090	0.653	9.022	9.167
<i>BatchProx_{final}</i> = 0.25m	0.082	0.635	0.069	0.573	6.835	7.114
	0.344	1.349	0.290	1.211	28.903	30.442
	0.213	1.020	0.181	0.928	18.058	18.058
	0.067	0.564	0.056	0.514	5.598	6.092
	0.277	1.235	0.240	1.120	23.896	26.022
	0.233	1.119	0.195	0.999	19.483	19.827
	0.196	1.008	0.165	0.913	16.458	16.458
	0.534	1.691	0.461	1.534	45.965	45.966
	0.536	1.680	0.457	1.518	45.544	46.895
	0.115	0.740	0.096	0.672	9.522	9.522
<i>BatchProx_{final}</i> = 0.75m	0.413	1.492	0.354	1.342	35.249	38.641
	0.240	1.095	0.202	0.989	20.165	20.165
	0.198	1.008	0.171	0.919	17.052	17.807
	0.279	1.174	0.238	1.069	23.733	24.149
	0.170	0.931	0.145	0.849	14.487	14.633
	0.173	0.924	0.147	0.840	14.642	15.274
	0.028	0.353	0.022	0.316	2.213	2.213
	0.245	1.097	0.206	0.996	20.502	20.633
	0.128	0.806	0.108	0.729	10.774	10.774
	0.497	1.600	0.426	1.456	42.421	42.421
<i>P_{mortality}</i>	0.333	1.286	0.283	1.165	28.178	29.120
Egg = 3.3%	0.511	1.600	0.437	1.453	43.553	43.553
LarvaeOut = 12.7%	0.418	1.580	0.356	1.401	35.530	39.503
LarvaeIn = 1%	0.171	0.894	0.144	0.810	14.278	14.356
Pupae = 0.8%	0.105	0.713	0.084	0.631	8.360	8.360
Adult = 22%	0.274	1.183	0.228	1.054	22.681	24.266
	0.204	1.033	0.171	0.924	17.012	18.562

	0.702	1.937	0.603	1.753	60.118	60.118
	0.287	1.256	0.247	1.135	24.635	27.394
	0.289	1.205	0.246	1.089	24.496	24.496
<i>P</i> _{mortality}	0.429	1.516	0.370	1.380	36.844	36.844
Egg = 2.7%	0.670	2.009	0.583	1.819	58.110	62.625
LarvaeOut = 10.4%	0.227	1.102	0.198	1.007	19.747	20.432
LarvaeIn = 0.8%	0.744	1.989	0.645	1.818	64.314	64.314
Pupae = 0.6%	0.070	0.571	0.058	0.516	5.735	6.258
Adult = 18%	0.360	1.411	0.313	1.284	31.155	31.604
	0.360	1.365	0.310	1.249	30.875	30.875
	0.166	0.959	0.141	0.853	14.082	14.799
	0.322	1.279	0.279	1.174	27.853	28.490
	0.932	2.225	0.812	2.027	80.931	80.931
$\lambda_{Erlang} = 1.25$	0.174	0.941	0.148	0.857	14.772	14.772
	0.037	0.428	0.031	0.384	3.059	3.355
	0.322	1.433	0.280	1.288	27.922	30.928
	0.318	1.381	0.271	1.228	27.006	27.365
	0.037	0.415	0.030	0.375	2.988	3.124
	0.379	1.579	0.324	1.406	32.305	38.888
	0.108	0.743	0.093	0.674	9.238	10.152
	0.245	1.165	0.207	1.041	20.661	22.015
	0.200	0.993	0.166	0.900	16.602	16.602
	0.460	1.659	0.401	1.500	40.011	44.845
$\lambda_{Erlang} = 1.75$	0.123	0.779	0.101	0.697	10.069	10.312
	0.246	1.159	0.213	1.050	21.254	22.901
	0.082	0.627	0.067	0.566	6.728	6.795
	0.081	0.617	0.067	0.554	6.686	7.060
	0.118	0.740	0.098	0.669	9.752	9.752
	0.295	1.214	0.253	1.108	25.239	26.779
	0.024	0.330	0.018	0.288	1.839	1.938
	0.249	1.104	0.213	1.011	21.226	21.226
	0.065	0.540	0.053	0.486	5.306	5.318
	0.152	0.887	0.128	0.800	12.806	12.806
<i>P</i> _{scanning} = 0.65%	0.088	0.659	0.072	0.591	7.152	7.619
<i>P</i> _{displaying} = 0.65%	0.221	1.069	0.186	0.959	18.528	19.265
	0.144	0.828	0.119	0.746	11.885	11.885
	0.106	0.698	0.087	0.626	8.665	8.878
	0.274	1.209	0.231	1.087	23.048	23.568
	0.274	1.198	0.230	1.074	22.948	22.948
	0.098	0.690	0.082	0.621	8.145	8.145
	0.177	0.950	0.146	0.852	14.522	14.522
	0.043	0.456	0.035	0.411	3.481	3.968
	0.164	0.908	0.139	0.820	13.825	13.825
<i>P</i> _{scanning} = 0.85%	0.088	0.650	0.074	0.591	7.371	7.472
<i>P</i> _{displaying} = 0.85%	0.440	1.538	0.382	1.403	38.038	38.038
	0.323	1.252	0.274	1.141	27.338	27.762
	0.553	1.798	0.478	1.631	47.649	47.649
	0.553	1.798	0.478	1.631	47.649	47.649
	0.245	1.141	0.208	1.029	20.725	22.712
	0.143	0.840	0.122	0.762	12.122	13.267

	0.292	1.175	0.250	1.080	24.968	24.968
	0.351	1.402	0.307	1.277	30.642	30.642
	0.148	1.009	0.123	0.923	12.248	13.917
$LD_{max} = 0.25\text{m}$	0.075	0.693	0.060	0.626	6.039	6.928
	0.353	1.523	0.295	1.393	29.360	29.360
	0.204	1.156	0.171	1.062	17.077	17.490
	0.356	1.538	0.296	1.400	29.595	29.605
	0.089	0.756	0.073	0.692	7.252	8.182
	0.144	0.969	0.118	0.882	11.772	12.583
	0.046	0.557	0.037	0.503	3.675	4.317
	0.085	0.756	0.069	0.683	6.909	7.807
	0.024	0.387	0.018	0.343	1.776	1.776
	0.465	1.419	0.400	1.268	39.848	39.848
$LD_{max} = 0.75\text{m}$	0.141	0.753	0.121	0.674	12.039	12.696
	0.252	0.984	0.216	0.885	21.564	21.564
	0.343	1.193	0.287	1.050	28.562	28.562
	0.196	0.944	0.164	0.828	16.342	16.342
	0.189	0.831	0.161	0.751	16.033	16.033
	0.131	0.676	0.109	0.604	10.844	11.106
	0.379	1.259	0.328	1.135	32.728	32.728
	0.207	0.957	0.174	0.843	17.362	19.404
	0.312	1.255	0.266	1.107	26.554	28.198
	0.196	0.944	0.172	0.833	17.167	17.280

TABLE A.2: Complete damage indication data collected during the sensitivity analysis using the 'Find Mature Cane' oviposition site selection algorithm.

Varied parameter	Current eggs & larvae	Max. eggs & larvae	Max population
None	1 809	3 783	4 147
	4 963	13 163	15 556
	10 553	27 855	33 931
	4 162	9 411	11 036
	5 027	13 624	15 351
	7 653	17 923	21 404
	3 404	9 119	11 114
	6 321	16 114	18 255
	5 470	18 017	20 327
	10 785	21 929	31 504
SR = 10m	658	1 377	1 538
	2 577	7 554	8 663
	8 082	22 656	26 178
	10 454	34 188	39 660
	7 262	18 135	21 938
	5 492	5 716	11 099
	6 962	17 479	21 344
	7 922	15 545	20 727
	1 752	5 985	6 598
10 955	30 395	39 392	
SR = 15m	3 497	9 941	12 598
	2 612	7 115	8 381
	7 627	18 956	21 845
	10 195	29 403	33 370
	9 402	32 602	39 467
	8 216	17 244	22 301
	3 712	8 053	9 947
	7 236	19 837	23 674
	126	413	437
	1 160	2 570	2 889
Prox. = 0.05m	1 018	1 748	2 243
	1 182	2 015	2 268
	631	1 114	1 281
	910	1 338	1 790
	146	327	370
	397	924	1 105
	915	1 812	2 089
	1 472	2 068	2 681
	1 097	1 963	2 199
	412	949	1 023
Prox. = 0.25m	15 834	70 675	80 673
	22 204	85 531	114 505
	11 863	35 183	41 582
	15 985	49 357	61 364
	7 655	27 003	30 465
	18 377	55 423	75 788
	15 523	55 637	71 433

	5 770	18 413	20 747
	8 591	34 648	38 898
	9 078	26 711	32 476
PD = 2.75m	3 841	9 879	10 903
WI = 2.75%	5 326	8 824	12 555
	2 637	6 718	7 444
	2 326	3 829	4 428
	4 704	9 874	11 959
	5 164	6 826	11 911
	2 470	7 016	7 915
	3 731	9 209	10 495
	3 067	7 160	8 148
	4 482	11 890	13 532
PD = 2.25m	8 238	19 047	24 072
WI = 2.25%	4 992	15 888	18 959
	1 314	3 381	3 639
	9 168	18 314	25 014
	4 963	13 122	14 662
	13 324	23 013	35 431
	5 824	13 748	16 166
	5 360	11 529	14 394
	3 287	8 934	10 000
	1 189	2 566	2 804
CQT = 1.05	14 801	47 967	64 359
(Good)	4 204	11 204	12 519
	6 353	20 295	23 885
	2 284	4 344	4 982
	6 507	17 069	21 293
	6 153	13 033	17 398
	1 374	4 171	4 451
	8 057	21 159	24 311
	7 867	21 499	25 700
	3 998	10 560	12 483
CQT = 1.175	6 705	19 118	23 320
(Excellent)	4 221	8 909	10 708
	1 115	2 034	2 254
	8 009	21 972	25 588
	1 250	3 367	3 640
	13 604	31 164	44 370
	1 615	5 744	6 274
	6 388	14 614	18 609
	2 529	4 861	5 648
	1 982	3 745	4 319
<i>P_{mortality}</i>	10 728	29 654	34 232
Egg = 3.3%	1 332	2 875	3 150
LarvaeOut = 12.7%	1 622	2 939	3 374
LarvaeIn = 1%	2 911	6 973	7 769
Pupae = 0.8%	2 024	4 709	5 436
Adult = 22%	5 593	15 306	17 272
	2 949	6 114	7 130

	2 178	4 348	4 965
	1 684	6 795	7 456
	5 583	18 382	20 408
<i>P_{mortality}</i>	1 687	3 224	3 872
Egg = 2.7%	6 579	11 524	14 106
LarvaeOut = 10.4%	3 426	9 753	11 775
LarvaeIn = 0.8%	5 526	14 782	17 061
Pupae = 0.6%	2 831	5 825	6 999
Adult = 18%	12 386	36 707	45 677
	4 342	10 740	12 181
	630	1 241	1 412
	1 818	4 535	5 452
	5 124	11 276	13 667
$\lambda_{Erlang} = 1.25$	778	1 660	1 887
	6 196	15 901	20 393
	2 707	7 072	8 153
	4 000	9 863	11 392
	2 583	5 542	6 448
	1 048	2 291	2 492
	2 599	5 812	7 088
	5 867	16 997	20 386
	5 407	14 632	17 042
	1 978	5 031	5 981
$\lambda_{Erlang} = 1.75$	8 169	20 613	24 193
	790	1 118	1 199
	8 091	16 573	20 111
	6 917	10 071	16 437
	2 821	8 822	9 829
	10 168	24 992	33 191
	4 182	9 469	12 179
	6 922	14 538	16 466
	1 083	2 075	2 263
	5 474	14 667	17 381
<i>P_{scanning} = 0.65%</i>	1 724	4 339	4 941
<i>P_{displaying} = 0.65%</i>	1 714	3 908	4 604
	3 925	10 004	11 939
	1 504	3 331	3 950
	2 080	4 432	5 400
	3 017	6 975	8 007
	5 718	9 045	14 236
	1 427	2 679	3 330
	5 645	14 655	17 161
	3 059	5 844	6 781
<i>P_{scanning} = 0.85%</i>	16 988	39 328	53 663
<i>P_{displaying} = 0.85%</i>	9 086	33 788	40 327
	19 252	52 614	69 477
	9 733	31 432	37 447
	385	1 080	1 321
	8 619	20 921	24 781
	3 363	10 631	11 906

	14 293	44 854	52 338
	7 109	21 457	24 428
	4 274	14 317	16 585
$LD_{max} = 0.25\text{m}$	1 285	2 758	3 711
	1 041	2 368	2 667
	613	2 152	2 434
	300	720	789
	935	3 335	3 839
	1 256	2 951	3 536
	844	1 719	2 069
	1 716	4 903	5 448
	706	1 636	1 947
	756	2 031	2 373
$LD_{max} = 0.75\text{m}$	8 210	23 066	25 706
	8 487	24 638	28 129
	19 269	32 437	53 489
	3 023	8 813	9 610
	14 925	45 590	52 717
	18 894	47 085	57 111
	5 308	10 266	11 844
	14 383	51 537	59 785
	15 841	35 559	44 996
	18 656	36 437	60 143

TABLE A.3: Complete population data collected during the sensitivity analysis using the ‘Moving Window’ oviposition site selection algorithm.

Varied parameter	Current eggs & larvae	Max. eggs & larvae	Max population
None	45 636	45 636	54 139
	18 999	20 211	22 869
	12 792	12 792	14 349
	17 676	18 487	20 975
	18 397	19 850	22 548
	11 955	12 755	14 478
	14 059	15 522	17 834
	82 175	82 175	98 534
	32 509	32 509	38 039
	16 120	16 139	19 359
SR = 10m	22 390	22 390	25 583
	90 840	90 840	107 746
	23 293	24 538	28 090
	21 321	21 321	25 745
	26 973	29 147	32 514
	18 327	18 327	21 594
	19 072	19 072	22 763
	26 373	27 174	31 842
	15 240	15 411	18 408
15 080	16 004	18 320	
SR = 15m	15 048	15 048	17 566
	34 416	34 416	38 577
	544	611	713
	33 685	33 685	40 423
	34 085	36 041	43 161
	24 200	24 679	28 913
	25 160	26 120	30 667
	18 926	18 926	21 263
	31 728	31 728	37 197
31 486	31 486	36 748	
Prox. = 0.05m	17 615	18 026	21 816
	7 927	7 927	9 732
	2 468	2 468	3 152
	18 219	18 219	21 702
	3 562	3 973	4 732
	5 946	5 946	7 201
	1 989	2 130	2 557
	1 408	1 408	1 799
	11 816	11 816	13 515
2 844	3 158	3 736	
Prox. = 0.25m	76 781	103 714	115 374
	81 227	81 227	96 975
	13 783	14 232	16 308
	55 145	55 145	62 917
	73 936	88 408	100 245
	39 132	40 614	46 490
	35 513	36 672	42 585

	45 503	47 158	54 623
	29 488	29 488	34 228
	31 600	32 687	37 839
CSR = 13.5m	39 069	44 043	50 260
WS = 0.26rad	32 504	33 673	38 815
	17 155	17 816	20 471
	45 886	50 482	57 880
	31 211	32 990	37 862
	24 904	25 758	29 810
	33 235	35 502	40 804
	66 391	69 700	80 366
	20 497	21 501	24 779
	44 063	47 496	54 579
CSR = 16.5m	6 166	6 492	7 469
WS = 0.79rad	44 440	46 967	54 247
	73 143	77 986	89 635
	79 809	85 141	97 695
	38 979	42 104	48 344
	27 596	28 270	32 740
	12 769	13 185	15 108
	70 289	74 428	85 667
	22 651	23 928	27 700
	11 729	11 917	13 828
<i>BatchProx_{final} = 0.25m</i>	8 886	9 248	10 631
	37 574	39 574	45 333
	23 475	23 475	26 539
	7 277	7 919	8 810
	31 065	33 829	38 186
	25 328	25 775	30 974
	21 396	21 396	25 710
	59 754	59 756	71 594
	59 207	60 963	72 193
	12 378	12 378	14 716
<i>BatchProx_{final} = 0.75m</i>	45 824	50 233	57 004
	26 215	26 215	31 128
	22 167	23 149	26 869
	30 853	31 394	37 126
	18 833	19 023	22 283
	19 035	19 856	22 741
	2 877	2 877	3 446
	26 653	26 823	31 737
	14 006	14 006	16 455
	55 147	55 147	64 407
<i>P_{mortality}</i>	36 632	37 856	44 401
Egg = 3.3%	56 619	56 619	65 557
LarvaeOut = 12.7%	46 189	51 354	60 739
LarvaeIn = 1%	18 562	18 663	21 940
Pupae = 0.8%	10 868	10 868	12 747
Adult = 22%	29 485	31 546	36 659
	22 115	24 131	27 664

	78 153	78 153	92 165
	32 026	35 612	39 899
	31 845	31 845	37 823
<i>P_{mortality}</i>	47 897	47 897	56 743
Egg = 2.7%	75 543	81 413	94 237
LarvaeOut = 10.4%	25 671	26 562	30 656
LarvaeIn = 0.8%	83 608	83 608	96 271
Pupae = 0.6%	7 455	8 136	9 046
Adult = 18%	40 502	41 085	48 652
	40 138	40 138	47 278
	18 307	19 239	22 499
	36 209	37 037	43 075
	105 210	105 210	127 679
$\lambda_{Erlang} = 1.25$	19 204	19 204	2 2203
	3 977	4 361	4 890
	36 299	40 207	46 237
	35 108	35 575	44 144
	3 885	4 061	4 581
	41 997	50 554	55 817
	12 009	13 198	14 618
	26 859	28 620	33 197
	21 582	21 582	23 883
	52 014	58 299	67 394
$\lambda_{Erlang} = 1.75$	13 090	13 406	15 632
	27 630	29 771	33 911
	8 747	8 833	10 491
	8 692	9 178	10 596
	12 678	12 678	14 877
	32 811	34 813	39 490
	2 391	2 519	2 973
	27 594	27 594	32 050
	6 898	6 914	8 081
	16 648	16 648	19 356
<i>P_{scanning} = 0.65%</i>	9 298	9 905	11 508
<i>P_{displaying} = 0.65%</i>	24 087	25 044	29 154
	15 450	15 450	18 006
	11 264	11 542	13 479
	29 962	30 638	36 278
	29 832	29 832	35 433
	10 589	10 589	12 366
	18 878	18 878	22 312
	4 525	5 158	5 631
	17 973	17 973	21 246
<i>P_{scanning} = 0.85%</i>	9 582	9 713	11 508
<i>P_{displaying} = 0.85%</i>	49 449	49 450	56 354
	35 539	36 090	42 599
	61 944	61 944	74 878
	61 944	61 944	74 878
	26 942	29 526	33 359
	15 758	17 247	19 400

	32 459	32 459	37 141
	39 835	39 835	47 770
	15 923	18 092	20 509
$LD_{max} = 0.25\text{m}$	7 851	9 006	10 319
	38 168	38 168	46 799
	22 200	22 737	27 451
	38 473	38 486	46 055
	9 427	10 637	11 905
	15 304	16 358	19 587
	4 777	5 612	6 299
	8 982	10 149	11 578
	2 309	2 309	2 795
	51 803	51 803	60 570
$LD_{max} = 0.75\text{m}$	15 651	16 505	18 809
	28 033	28 033	32 351
	37 130	37 130	42 735
	21 244	21 244	25 664
	20 843	20 843	24 594
	14 097	14 438	16 908
	42 547	42 547	48 976
	2 2571	25 225	27 932
	34 520	36 658	41 968
	22 317	22 464	26 683

TABLE A.4: Complete population data collected during the sensitivity analysis using the 'Find Mature Cane' oviposition site selection algorithm.

APPENDIX B

Contents of the accompanying compact disc

This appendix contains a brief description of the compact disc included with this dissertation. The compact disc contains an electronic version of the dissertation itself in “.pdf” format as well as the ANYLOGIC project file of the final, revised agent-based model, described in Chapters 4, 8 and 9. The model was created in ANYLOGIC version 7.3.1 and may be executed in this version of the software or later. There are two directories on the compact disc and their contents are described here by their directory names.

Dissertation. This directory contains an electronic copy of this dissertation in “.pdf” format.

Agent-based simulation model. This directory contains the complete agent-based simulation model described in Chapters 4, 8 and 9 as ANYLOGIC script files (“.alp” format). The simulation model is labelled “PhDSimulation.alp”. To execute this simulation model, the file should be opened from ANYLOGIC. Once opened, the user is required to run the model by either clicking the “Run” button in ANYLOGIC, or by pressing F5. Following this, the window shown in Figure 9.30 will appear from which the user can either proceed to the “Settings” tab in order to alter the values of certain parameters within the model, or run the model by clicking the “Run” button.