

A multi-objective approach towards geospatial facility location

Andries M Heyns



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

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: March , 2016

Abstract

Applications of the sub-discipline of location science within the larger discipline of operations research include the location of important facilities, such as hospitals, fire stations, libraries and depots. Existing facility location resolution techniques are generally based on commercial and transportation modelling criteria with little or no consideration given to the characteristics and influence of surrounding terrain. Recent technological advances have, however, resulted in a sub-field of location science rapidly increasing in popularity — the field of *geospatial facility location*. Geospatial facility location science places a strong emphasis on terrain and environmental factors in the search for suitable sites for facilities with complex location requirements. Examples of such facilities include radars, telecommunication towers, watchtowers and wind turbines.

The applications in which geospatial facility location solution approaches are relevant generally include only one type of facility. As a result, problem-specific solution approaches tailored to the specific facility type considered are usually adopted. On closer inspection, however, the solution approaches followed for these facility location problems reveal striking similarities which may be exploited in a generic manner in order to provide more efficient solution tradeoff alternatives.

By removing the problem-specific approach to facility location modelling and the design of the associated solution techniques, there is an opportunity to develop a generic, dynamically implementable geospatial facility location framework which may be applied to identify facility location trade-off alternatives in various applications. Moreover, networks of multiple facility types may be considered in such a framework — something that is a rarity in the current geospatial facility location literature. In view of the continual advancement of technology, a generic geospatial facility location framework may even prove to be applicable to future facility location problems in which the placement criteria are currently unknown.

The design of a generic geospatial facility location framework is therefore pursued in this dissertation. A dynamic mathematical foundation, which eliminates facility-specific solution methodologies, is established. A computerised concept demonstrator based on this framework is implemented within a generic solution paradigm. This concept demonstrator accommodates as special cases a variety of popular facility location problems from the literature. The geospatial facility location framework is demonstrated to be a dynamic and useful tool for solving complex facility location problems that have not previously been encountered due to the limitations of problem-specific solution search designs.

Uittreksel

Toepassings van die deel-dissipline van fasiliteitplasing binne die groter dissipline van operasionele navorsing sluit die plasing van belangrike fasiliteite soos hospitale, brandweerstasies, biblioteke en depots in. Bestaande oplossingstegnieke vir fasiliteitplasingsprobleme is gewoonlik gebaseer op kommersiële en vervoermodelleringskriteria met geen of weinige oorweging geskenk aan die eienskappe en invloed van die omliggende terrein. Onlangse tegnologiese vooruitgang het egter gelei na toenemende populariteit van 'n deel van die teorie van fasiliteitplasing — die veld van *geo-ruimtelike fasiliteitplasing*. In geo-ruimtelike fasiliteitplasing word daar sterk klem gelê op terrein- en omgewingsfaktore in die soeke na geskikte liggings vir die plasing van fasiliteite met komplekse plasingsvereistes. Voorbeelde van sulke fasiliteite sluit radar, telekommunikasietorings, uitkyktorings en windturbunes in.

Die toepassings waarin oplossingstegnieke vir geo-ruimtelike fasiliteitplasing relevant is, sluit gewoonlik net een tipe fasiliteit in. Gevolglik word probleem-spesifieke oplossingstegnieke gewoonlik aangewend wat gerig is op een spesifieke tipe fasiliteit. By nadere ondersoek blyk dit egter dat die oplossingstegnieke wat vir hierdie plasingsprobleme gebruik word, opvallende ooreenkomste openbaar wat uitgebuit kan word om meer doeltreffende oplossingsalternatiewe op 'n generiese manier daar te stel.

Deur die probleem-spesifieke benadering uit pogings tot modellering en die gepaardgaande ontwerp van oplossingstegnieke te verwyder, is daar die geleentheid om 'n generiese, dinamies-implementeerbare geo-ruimtelike fasiliteitplasingsraamwerk te ontwikkel waarvolgens afruilingsalternatiewe vir fasiliteitplasing in verskeie toepassings geïdentifiseer kan word. Netwerke van verskillende soorte fasiliteite kan verder in só 'n raamwerk oorweeg word — 'n rare verskynsel in die huidige literatuur oor geo-ruimtelike fasiliteitplasing. In die lig van die voortdurende vooruitgang in tegnologie, mag 'n generiese raamwerk vir geo-ruimtelike fasiliteitsplasing selfs toepasing vind in toekomstige fasiliteitplasingsprobleme waarvoor die plasingskriteria tans nog nie bekend is nie.

Die ontwerp van 'n generiese geo-ruimtelike fasiliteitplasingsraamwerk word gevolglik in hierdie proefskrif nagestreef. 'n Dinamiese wiskundige modelleringsgrondslag, wat fasiliteit-spesifieke oplossingstegnieke onnodig maak, word daargestel. 'n Gerekenariseerde konsepdemonstrator wat op hierdie raamwerk gebaseer is, word binne die paradigma van 'n generiese oplossingsbenadering geïmplementeer. Hierdie konsepdemonstrator akkommodeer 'n aantal populêre fasiliteitplasingsprobleme uit die literatuur as spesiale gevalle. Daar word getoon hoe dié raamwerk 'n nuttige, dinamiese hulpmiddel is waarmee komplekse fasiliteitplasingsprobleme opgelos kan word wat as gevolg van die beperkings van probleem-spesifieke oplossingsbenaderings nie vantevore oorweeg is nie.

Acknowledgements

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

- Prof JH van Vuuren for his extraordinary effort in overseeing my research over many years, starting way back in 2006 when I began my MSc under his supervision. Never would I have imagined that I would have a Masters degree, let alone a PhD, and I am only writing this as a result of Prof van Vuuren's belief in my abilities.
- The ARMSCOR LEDGER programme, the National Research Foundation and the Post-graduate Student Funding Office of Stellenbosch University for the financial assistance they provided.
- Mr Caxton Magazore, for his significant contribution in the writing of my first 'big' publication.
- Dr Jacques du Toit, for motivating me to persevere when times were at their absolute toughest. Next round is on me.
- My family and in particular my mother, Venesia, for always being there for me.

Table of Contents

List of Acronyms	xiii
List of Figures	xv
List of Tables	xix
List of Algorithms	xxi
1 Introduction	1
1.1 Background	1
1.2 Informal problem description	2
1.3 Dissertation aims and objectives	3
1.4 Dissertation scope	4
1.5 Research methodology	4
1.6 Dissertation organisation	5
2 Literature review	7
2.1 Introduction	7
2.2 Classical facility location problems in a GFLo context	8
2.2.1 Network-based problems	8
2.2.2 Covering problems	11
2.3 GFLo optimisation approach considerations	13
2.3.1 Point-based GFLo solution representation and evaluation	13
2.3.2 Exact versus approximate optimisation	15
2.3.3 Single-objective versus multi-objective optimisation approaches	16
2.3.4 Single-solution versus population-based optimisation	18
2.4 Approximate algorithms for solving GFLoPs	18
2.5 Chapter summary	20

3	GFLo data management and modelling	21
3.1	The GFLo database	21
3.1.1	The facility database	21
3.1.2	The geospatial database	22
3.1.3	Geospatial data operations	23
3.2	Practical data considerations	26
3.3	Classification of GFLo criteria	28
3.3.1	Location-specific criteria	28
3.3.2	Inter-site criteria	29
3.4	Mathematical modelling of the GFLo environment	31
3.4.1	Modelling search and interest zones	32
3.4.2	Modelling facilities	33
3.4.3	Modelling facility placement criteria	36
3.5	Chapter summary	39
4	Generic multi-objective problem formulation framework	41
4.1	Preliminary notation	41
4.2	Generic mathematical formulation of objectives	44
4.2.1	MaxiMin and MiniMax models	44
4.2.2	Covering models	45
4.2.3	Dispersion models	47
4.2.4	Centre models	49
4.3	Generic mathematical modelling of additional constraints	49
4.3.1	Location-specific constraints	51
4.3.2	Inter-site constraints	52
4.4	Generic GFLoP formulation process	52
4.5	Chapter summary	53
5	Development of MO GFLo algorithms	55
5.1	MOO concepts	56
5.1.1	Pareto-front approximation	56
5.1.2	Diversity preservation	57
5.1.3	Solution ranking	58
5.1.4	Constraint handling	59
5.1.5	Algorithm termination	61
5.1.6	Solution quality assessment	61

5.2	GFLo candidate solution representation	63
5.3	GFLo algorithms	63
5.3.1	Non-dominated sorting genetic algorithm-II	63
5.3.2	The novel AMOEBA algorithm	69
5.4	Worked example of the GFLo solution process	73
5.4.1	Terrain description	74
5.4.2	Facilities and zones	74
5.4.3	Objectives and constraints	74
5.4.4	Algorithm parameter selection	75
5.4.5	Mathematical model formulation	76
5.4.6	Algorithm results comparison	77
5.5	A preliminary appraisal of the algorithms	79
5.6	Chapter summary	81
6	GFLo-specific search improvements	83
6.1	A hybrid geospatial algorithm	83
6.1.1	Algorithm description	84
6.1.2	Implementation and results	85
6.2	Halving NSGA-II population size	87
6.3	A multi-resolution approach	91
6.3.1	Motivation	91
6.3.2	Method	94
6.3.3	Worked example	94
6.3.4	General observations and implementation notes	99
6.3.5	Important MRA considerations	100
6.4	Proximity-dependent cluster removal	101
6.5	Chapter summary	102
7	Practical implementation of the generic GFLo framework	105
7.1	Proposed GFLo decision support architecture	106
7.1.1	Architecture overview	106
7.1.2	An optimisation interface concept demonstrator	108
7.2	Practical use of the generic GFLo framework	110
7.2.1	Problem overview	110
7.2.2	Evaluation of existing camera locations	112
7.2.3	Solving the problem using the GFLo framework	113

7.2.4	Comparison of actual and proposed camera locations	114
7.2.5	Decision maker feedback and conclusion	115
7.3	Small-area problem instances	115
7.3.1	Low-complexity bi-objective GFLoP instance	116
7.3.2	Low-complexity tri-objective GFLoP instance	117
7.3.3	High-complexity tri-objective GFLoP instance	121
7.3.4	Algorithmic observations for the small-area problems	125
7.4	Large-area examples	128
7.4.1	High-complexity bi-objective GFLoP instance	129
7.4.2	Low-complexity tri-objective GFLoP instance	130
7.4.3	Algorithmic observations for the large-area problems	133
7.5	Chapter summary	135
8	Conclusion	137
8.1	Dissertation summary	137
8.2	Contributions of this Dissertation	138
8.3	Ideas for future work	140
	References	143
A	Mathematical formulations for the GFLoP instances of §7	153
A.1	Small-area problem instances	153
A.1.1	Low-complexity bi-objective GFLoP instance of §7.3.1	153
A.1.2	Low-complexity tri-objective GFLoP instance of §7.3.2	154
A.1.3	High-complexity tri-objective GFLoP instance of §7.3.3	155
A.2	Large-area examples	156
A.2.1	High-complexity bi-objective GFLoP instance of §7.4.1	156
A.2.2	Low-complexity tri-objective GFLoP instance of §7.4.2	158

List of Acronyms

AMOEBA: Asexual multi-objective evolutionary-based algorithm

BCLP: Backup covering location problem

DSS: Decision support system

EA: Evolutionary algorithm

FLo: Facility location

FLoP: Facility location problem

FNSA: Fast non-dominated sorting algorithm

GA: Genetic algorithm

GFLo: Geospatial facility location

GFLoP: Geospatial facility location problem

GIS: Geographic information system

GUI: Graphical user interface

HGA: Hybrid geospatial algorithm

IZ: Interest zone

LOS: Line-of-sight

LSCP: Location set covering problem

MCLP: Maximal covering location problem

MO: Multi-objective

MOEA: Multi-objective evolutionary algorithm

MOO: Multi-objective optimisation

MRA: Multi-resolution approach

NSGA-II: Non-dominated sorting genetic algorithm-II

RG: Reference grid

SA: Simulated annealing

SKA: Square kilometre array

SO: Single-objective

SRA: Single-resolution approach

SZ: Search zone

TBA: Teitz & Bart algorithm

TIN: Triangulated irregular network

UHF: Ultra high frequency

VHF: Very high frequency

List of Figures

2.1	The notion of facility dispersion	9
2.2	The notion of facility centring	11
2.3	Coverage provided by facilities in a GFLo context	13
2.4	Candidate GFLo solution representation on terrain	14
2.5	Exact versus metaheuristic solutions in objective space	16
3.1	Illustration of a typical set of GIS-based pre-processing operations	25
3.2	A GFLo database and associated operations	26
3.3	Terrain represented by raster data at two different resolution levels	27
3.4	The classification of GFLo criteria types	28
3.5	Location-specific criteria values extracted from a terrain area	30
3.6	An example of a viewshed analysis with respect to an observer gridpost	32
3.7	RG, SZs and IZs considered in Example 3.1	34
3.8	Placement of facilities in Example 3.2.	35
3.9	Location-specific criteria values for Example 3.3	37
3.10	Geospatial inter-site criteria values for Example 3.4	38
3.11	Facility-to-site inter-site criteria values for Example 3.4	39
3.12	Inter-facility inter-site criteria values for Example 3.4	40
4.1	An illustration of a set-pair and associated set-pair decision variables	43
4.2	An illustration of the set-pair dispersion model	48
4.3	An illustration of the set-pair centre model	50
4.4	A generic GFLoP formulation framework	54
5.1	Solution fronts in objective space for a bi-objective maximisation problem	57
5.2	Crowding distance for a solution	58
5.3	An illustration of the hypervolume performance measure	62
5.4	Chromosome string representation scheme employed for solving GFLoPs	63

5.5	The GFLo candidate solution representation scheme in Example 5.1	64
5.6	NSGA-II crossover performed on candidate solutions in Example 5.2	66
5.7	Mutation performed on an offspring solution	67
5.8	AMOEBA mutation performed on the parent solution in Example 5.3	71
5.9	The first three stages of an AMOEBA infection spread.	73
5.10	Terrain, SZs and average daily solar radiation in a worked example	75
5.11	Results obtained for the example of §5.4.1–5.4.5 by the NSGA-II and AMOEBA	78
5.12	Physical site location of the solutions in the attainment fronts of Figure 5.11	80
6.1	A flowchart of the HGA solution process	85
6.2	Results of the three algorithms for the problem instance of §5.4.1–5.4.5	86
6.3	Site locations of the solutions in the HGA attainment front of Figure 6.2 (a)	88
6.4	Attainment fronts of the NSGA-II with and without population halving	90
6.5	An illustration of excessive candidate location evaluation	92
6.6	Low-resolution and high-resolution observations that led to the MRA	94
6.7	An illustration of the process followed during execution of the MRA	95
6.8	SRA and MRA attainment fronts for the problem instance of §6.3.3	96
6.9	Comparisons of MRA and SRA run times and sites evaluated	98
6.10	Comparison of candidate locations evaluated by the SRA and MRA in §6.3.3	99
6.11	The explore-and-exploit search approach of the MRA.	100
6.12	Proximity-dependent cluster identification	102
7.1	Proposed GFLo decision support architecture	107
7.2	The optimisation interface concept demonstrator	109
7.3	Top view of the terrain elevation and antenna locations in the MeerKAT project	111
7.4	Locations of the actual observation cameras in the MeerKAT project	113
7.5	Search zones for camera placement in the MeerKAT study	114
7.6	Visibility results for a non-dominated solution of the MeerKAT study	115
7.7	Results of the three algorithms in the context of the problem instance of §7.3.1	118
7.8	Sites evaluated in the context of the example of §7.3.1	119
7.9	Results of the three algorithms in the context of the problem instance of §7.3.2	122
7.10	Sites evaluated by all three algorithms in the context of the example of §7.3.2	123
7.11	Results of the three algorithms in the context of the problem instance of §7.3.3	126
7.12	Sites evaluated by all three algorithms in the context of the example of §7.3.3	127
7.13	Terrain considered in the large-area problem instance of §7.4	129
7.14	Attainment fronts of the three algorithms for the GFLoP instance of §7.4.1	131

7.15	Sites corresponding to the fronts of Figure 7.14	131
7.16	Cover achieved by the solutions in the HGA attainment front in Figure 7.14 . . .	131
7.17	Attainment fronts and sites evaluated for the GFLoP instance of §7.4.2	134
7.18	Site locations of HGA solutions in the context of the GFLoP instance of §7.4.2 .	135

List of Tables

5.1	Hypervolume values for the example of §5.4.1–5.4.5	77
5.2	Algorithm statistics for the example of §5.4.1–5.4.5	79
6.1	Hypervolume values for problem instance §5.4.1–5.4.5	85
6.2	Algorithm statistics for problem instance of §5.4.1–5.4.5	87
6.3	Hypervolume values with and without the population halving technique	91
6.4	Average algorithm values with and without the population halving technique	91
6.5	Results for the SRA and MRA corresponding Figures 6.8 and 6.9	96
7.1	Hypervolume values of each of the algorithms for the GFLoP instance in §7.3.1	117
7.2	Algorithm statistics relevant to the GFLoP instance in §7.3.1	117
7.3	Hypervolume values of each of the algorithms for the GFLoP instance in §7.3.2	121
7.4	Algorithm statistics relevant to the GFLoP instance in §7.3.2	121
7.5	Hypervolume values of each of the algorithms in the GFLoP instance of §7.3.3	125
7.6	Algorithm statistics relevant to the GFLoP instance in §7.3.3	125
7.7	Hypervolume values of the algorithms for the GFLoP instance in §7.4.1	132
7.8	Algorithm statistics relevant to the GFLoP instance in §7.4.1	132
7.9	Hypervolume values of the algorithms for the GFLoP instance in §7.4.2	133
7.10	Algorithm statistics relevant to the GFLoP instance in §7.4.2	133

List of Algorithms

5.1	Crowding distance assignment algorithm	59
5.2	Fast non-dominated sorting algorithm	60
5.3	Non-dominated Sorting Genetic Algorithm-II	68
5.4	Asexual multi-objective evolutionary-based algorithm	74

CHAPTER 1

Introduction

Contents

1.1	Background	1
1.2	Informal problem description	2
1.3	Dissertation aims and objectives	3
1.4	Dissertation scope	4
1.5	Research methodology	4
1.6	Dissertation organisation	5

1.1 Background

The class of *facility location* problems is well-documented in the operations research literature [28, 34, 68, 98, 110]. The traditional aim in these problems, from an operations research perspective, is to find an optimal allocation of facilities to candidate locations in order to minimise some objective function — typically transportation costs or delivery times between the facilities and demand locations. As a result, these techniques are usually based on commercial and transportation modelling principles with little or no consideration given to the characteristics and influence of surrounding terrain and environmental criteria.

The research and application scope of traditional facility location problems have, however, expanded in recent times due to factors such as (a) an increase in the need for reliable and efficient telecommunication services, (b) the ever-increasing demand for innovation in respect of military/strategic operations and (c) the increasing need for renewable and sustainable energy sources. These developments have led to the steady rise in research related to the optimal placement of facilities according to terrain and environmental criteria — henceforth referred to as *geospatial facility location problems* (GFLoPs) — and are already well-documented and wide-ranging in solution methodology and practical application [1, 31, 53, 60, 64, 66, 93, 112, 114, 118, 133, 134].

A large portion of GFLoPs may be categorised as *region-based*. In this class of problems the aim is to find generally large, contiguous areas of terrain destined for the placement of a number of facilities within their boundaries, *e.g.* regions identified for the development of wind farms [60, 118, 134] or solar farms [112, 133]. The literature on this class of problems is abundant and the techniques developed for these purposes are largely generic, employing multi-criteria decision making techniques on geographic and spatial data layers [112, 118, 133, 134]. These

problems may be adapted with relative ease from one regional selection problem for one type of facility and applied to a problem related to a different facility.

This dissertation, however, is concerned with a second class of GFLoPs — namely those that are *point-based*. In point-based GFLoPs the aim is to find precise, discrete site locations for *networks* of facilities. The following are popular examples of the types of facilities that may be considered in this class of problems:

- Transmitters [2, 40, 73, 80, 83],
- Surveillance cameras [7, 53, 65, 93, 135],
- Wind turbines [31, 75, 130, 140],
- Radar, including weather and military radar [6, 27, 88, 125], and
- Weapon systems [6, 43, 125].

The identification of candidate locations to be considered in point-based analyses may often follow on region-based analyses in which suitable areas of terrain to be considered for the placement of the facilities have been identified, after the removal of undesirable areas, such as those that are prohibited or are considered inaccessible.

In point-based GFLoPs problems the aim is to find suitable facility site placement configurations at the candidate locations in such a manner that they provide (near-)optimal trade-off alternatives with respect to multiple (typically conflicting) objectives. Such a set of solution alternatives are typically presented to one or more decision makers, who desire a diverse set of solutions that offer alternatives in respect of *both* (a) the proposed site locations and (b) the expected performance with respect to a set of objectives. In order to solve these complex problems with the aim of providing a diverse set of solution alternatives satisfying these requirements, multi-objective combinatorial optimisation techniques are employed.

1.2 Informal problem description

Point-based multi-objective GFLoPs and their solution approaches from the literature are not uncommon [1, 2, 31, 37, 43, 65, 66, 114]. An investigation into these solution approaches has, however, revealed two significant limitations:

1. The networks for which the solution approaches have been developed generally include only one type of facility.
2. As opposed to region-based problems, these problems have been solved using problem-specific approaches and special-purpose algorithms designed for the specific problem instance and facility type considered in the model. As a result, these solution approaches include a limited set of criteria considered in the solution process and are solved according to a specific set of objectives and constraints — thereby limiting the application scope of the solution approach to the specific problem only.

Despite the above-mentioned observations in respect of traditional point-based multi-objective *geospatial facility location* (GFLo) solution approaches, there exist similarities between them in their pursuit of candidate solutions, irrespective of the type of facility to be placed. The

common denominators are the placement criteria included in these problems — to be included in the model either as objectives or constraints — in respect of their respective mathematical problem formulations and data representation structures.

The advancement of point-based facility networks as a result of requirements such as including more than one facility type in the network and the consideration of multiple terrain and environmental criteria in the problem means that point-based GFLoPs are destined to increase in complexity and practical application¹. As a result, this burgeoning field of research presents the opportunity to develop a generic, dynamically implementable multi-objective GFLo framework that exploits similarities between solution approaches developed for different facility location problems — removing dependency on facility type and allowing for the consideration of a diverse range of criteria types.

1.3 Dissertation aims and objectives

This dissertation is aimed at developing a generic multi-objective GFLo framework that avoids the redundant development of purpose-specific approaches and algorithms and, instead, offers a single point of reference that may provide entities such as telecommunication enterprises, military forces or renewable and sustainable energy developments the opportunity to solve complex problems that were previously not possible to solve — quickly, efficiently, and as accurately as possible. Towards realising this aim, seven detailed objectives are pursued:

- I To *perform* a literature survey on classical facility location models which may be considered for inclusion in a generic GFLo framework.
- II To *perform* a literature survey on the topic of multi-objective optimisation alternatives which may be employed to solve point-based multi-objective GFLoPs.
- III To *perform* a literature survey on popular geospatial criteria which may be considered for inclusion in a generic GFLo framework.
- IV To *develop* a general design framework for generic point-based multi-objective GFLo purposes. Within the context of this objective the following aims are pursued:
 - (a) To *provide* a self-contained introduction to the topic of geospatial analyses in respect of geospatial data.
 - (b) To *identify* and *exploit* similarities between popular criteria which may be considered during GFLo decision making processes.
 - (c) To *provide* generic mathematical formulations for objectives and constraints in respect of the above-mentioned geospatial criteria, so that they may easily be implemented for problems with wide-ranging practical applications.
 - (d) To *develop*, *implement*, and *evaluate* optimisation algorithms for inclusion in the framework.
- V To *conceptualise* and *implement* a concept demonstrator of a computerised decision support tool, based on the framework of Objective IV, which may be used to aid in solving multi-objective GFLoPs.

¹An example of a complex network is the placement of N watchtowers, each equipped with solar-powered devices, placed according to objectives related to terrain visibility and average solar irradiation values, with the additional requirement that each watchtower should be visible to one, or more, mutual communication device(s).

- VI To *demonstrate* the practical applicability of the generic framework of Objective IV by applying the decision support tool of Objective V in the context of realistic scenarios.
- VII To *identify* future directions of research that may improve upon the work presented in this dissertation.

1.4 Dissertation scope

The range of facilities, criteria and approaches that may be considered within a point-based multi-objective GFLo environment is extensive. In order to narrow down the scope of the problem, the following limitations are made in this dissertation:

1. The number of facilities to be considered for placement is pre-specified and fixed.
2. Only the multi-facility problem — *i.e.* the placement of *more than one* facility — is considered.
3. Each facility location problem is concerned with the placement of a new network of facilities only, *i.e.* any existing facilities are not considered.
4. Only one facility is permitted to be placed at any candidate location.
5. For facilities such as cameras and radar that have a specific operational range, the range is limited to a maximum of 15 kilometres.
6. As a result of (5) above, the effects of the curvature of the earth on computations such as *Line-of-Sight* (LOS) and radio wave propagation are considered negligible in the examples included in this dissertation.
7. The criteria related to (5) and (6) above are limited to binary values — for example, a point is either visible or not visible, or receives sufficient signal or not.

1.5 Research methodology

The methodological procedure followed in this dissertation to design and develop a generic GFLo framework is as follows:

1. *Consult* existing literature in respect of the current state of GFLo applications and solution methodologies.
2. *Develop* a proficiency in geospatial analysis within a geographic information system software environment.
3. *Identify* similarities between popular criteria encountered in the GFLo literature and, following this, *model* the geospatial environment mathematically, in a generic manner.
4. *Identify* classical facility location models which are suitable for implementation within a GFLo context by studying the facility location literature.
5. *Formulate* generically implementable versions of the classical facility location problems as well as typical GFLo placement problems, using generic criteria formulations, for dynamically interchangeable use in a GFLo environment.

6. *Design* and *implement* algorithms for solving GFLoP instances in a manner that incorporates the generic mathematical modelling of the geospatial environment and placement criteria.
7. *Design* purpose-specific software to facilitate the GFLo framework development and testing process.
8. *Develop* procedures to exploit problem-specific GFLo characteristics and employ this knowledge to improve search methods for solving GFLoP instances.
9. *Perform* a number of practical GFLoP instances to test and improve the performance of the GFLo framework as well as the algorithms embedded in it.
10. To *verify* the generic GFLo framework by means of at least one practical study involving a real-life facility placement problem and consulting with experts in the field.

1.6 Dissertation organisation

This introduction is followed by seven additional chapters. Chapter 2 serves the purpose of elucidating the field of facility location study, and GFLo in particular. A general background on facility location is provided, and this is followed by a literature review on classical facility location models and their prevalence and application potential in the context of GFLo. Thereafter, an investigation into solution approaches for solving GFLoPs is conducted, in addition to a high-level overview of algorithms that are generally employed to solve GFLoP instances.

The structure of a suitable GFLo database is proposed in Chapter 3. This includes descriptions of facility and geospatial databases that make up the GFLo database, and this is followed by a description of different types of geospatial analyses that are followed in order to process geospatial data for problem-specific purposes. Typical GFLo criteria are investigated and classified into generic classes according to similarities that they exhibit, which is followed by the mathematical modelling of the GFLo environment.

Generic mathematical formulations of objectives and constraints that may be considered in a GFLo framework are pursued in Chapter 4. The classical facility location models of dispersion and centring are considered, in addition to cover models which are prevalent in a number of GFLo studies. The chapter closes with a description of a generic GFLoP formulation framework, which serves the purpose of providing the necessary guidelines for formulating a GFLoP model independently of the type of facility included in the model.

Chapter 5 opens with an introduction to multi-objective optimisation concepts that are of relevance to the GFLo framework. These concepts are employed in the implementation of two different GFLo algorithms, namely the well-known NSGA-II and a novel algorithm called AMOEBA. The performance of these algorithms are compared to each other in the context of a realistically worked GFLoP instance.

A number of novel search improvements of specific relevance in the context of GFLo are introduced in Chapter 6. A hybrid implementation of the NSGA-II and AMOEBA is proposed. Additional search improvements that are of relevance to all three algorithms are also proposed. These improvements exploit problem-specific algorithmic and environmental characteristics, and return impressive results.

A variety of GFLoP instances are solved in Chapter 7 in order to demonstrate the workability of the GFLo framework proposed in this dissertation. The chapter opens the proposal of a

GFLo framework decision support architecture and a discussion on a concept demonstrator of this architecture which was employed to solve the GFLoP instances in this dissertation. A practical study performed in order to evaluate a real-life observation camera placement problem is also presented. This is followed by three GFLoP instances solved for small-area and large-area terrains.

Finally, a summary of the dissertation content is presented in Chapter 8, together with an overview of the contributions made. The chapter closes with a number of suggestions in respect of future work which may improve or build on the work conducted in this study.

CHAPTER 2

Literature review

Contents

2.1	Introduction	7
2.2	Classical facility location problems in a GFLo context	8
2.2.1	<i>Network-based problems</i>	8
2.2.2	<i>Covering problems</i>	11
2.3	GFLo optimisation approach considerations	13
2.3.1	<i>Point-based GFLo solution representation and evaluation</i>	13
2.3.2	<i>Exact versus approximate optimisation</i>	15
2.3.3	<i>Single-objective versus multi-objective optimisation approaches</i>	16
2.3.4	<i>Single-solution versus population-based optimisation</i>	18
2.4	Approximate algorithms for solving GFLoPs	18
2.5	Chapter summary	20

This chapter provides the material for understanding the fundamental concepts that form the basis of the mathematical formulation and algorithmic development which follow in later chapters. This chapter opens in §2.1 with an introduction to the facility location branch of operations research. A review of classical *facility location problems* (FLoPs) that may be considered for implementation in a GFLo framework is provided in §2.2. This is followed by a discussion in §2.3 on optimisation principles that are of importance when designing a GFLo framework — specifically with respect to the solution approaches involved in such a procedure. Algorithms that are typically used for solving GFLoPs are considered in §2.4.

2.1 Introduction

Facility location (FLo) is a branch of operations research related to the positioning of at least one facility in order to satisfy at least one objective as best as possible [34]. Location science covers a wide range of application areas which are concerned with the placement of public facilities, private facilities, and facilities in the military environment [34]. The development and acquisition of new facilities in this context is typically a costly and time-sensitive process [98]. Suitable locations must be identified *before* facilities can be purchased or constructed, as large amounts of capital are allocated to these projects [98]. The high costs associated with the facility location process make almost any location project a long-term investment and determining the best locations for new facilities is thus an important strategic challenge [98].

FLoPs may be categorised into two classes [4], namely static FLoPs and dynamic FLoPs. Static FLoPs take only space factors into account, while dynamic FLoPs take time factors into account as well. In the case of dynamic FLoPs, time factors are introduced in the sense of projected facility operational time, in addition to planned opening and closing times of facilities considered over the planning horizon [4]. Static FLoPs, on the other hand, simply take current factors and existing facility locations into account. The research covered in this dissertation is concerned with static FLoPs.

Static FLoPs may be further divided into the classes of continuous and discrete FLoPs [4, 110], depending on the space in which the facilities are located. According to Revelle and Eiselt [110], in the case of continuous FLoPs, the facilities to be sited can generally be placed anywhere on the plane (terrain). An example of such a continuous location problem in the plane might be the placement of a helicopter for trauma pickup [110]. In discrete problems the facilities can be placed only at a limited number of eligible points on the plane. Discrete location models have therefore gone through additional preprocessing phases that preselect candidate sites at which the facilities may be sited [110]. An example of a discrete location problem is the positioning of transmitter stations that are to be placed at some permissible points within a region, such as mountaintops [110]. Continuous location problems tend to be non-linear optimisation problems, while discrete location problems involve binary decision variables that result in integer programming or combinatorial optimisation problems. The research conducted in this dissertation considers the discrete class of FLoPs.

In the FLo literature, the most popular classes of FLoPs are consistently observed to be *median* and *centre* problems (typically related to the distance between facilities and demand points), *cover* problems (which typically aim to serve a set of demand points by a number of facilities), and *dispersion* problems (recently gaining in popularity and concerned with inter-facility distance) [4, 16, 68, 76, 78, 98, 110]. The problems mentioned here are elucidated on in the following sections in order to motivate their inclusion in a generic *multi-objective* (MO) GFLo framework, as well as determining suitable solution approaches in the context of the GFLo environment.

2.2 Classical facility location problems in a GFLo context

The review of classical FLoPs are divided into two categories in this chapter, namely *network-based problems* which are concerned with inter-facility interaction, in §2.2.1, and *covering problems* which are concerned with services that facilities provide to demand points, in §2.2.2. In a GFLo context, the majority of literature related to these problem categories are related to the covering problems.

While network-based FLoPs in this dissertation are aimed at the interaction between facilities, the problems that are considered in this section often involve interaction between facilities and demand points in terms of performance criteria such as distance or cost (or distance perceived as cost). In a GFLo context, however, service provided to demand points is mostly considered in covering problems for the purposes of transmission or visibility coverage. Therefore, FLoPs that are considered compatible with network-based GFLoPs are discussed in §2.2.1, while those that are considered to be members of the class of covering problems are discussed in §2.2.2.

2.2.1 Network-based problems

Literature related to network-based problems that involve inter-facility interaction is mostly found with respect to *dispersion* problems, while *centre* and *median* problems generally only

involve facility-demand interaction, yet merit inclusion in this literature review due to their similarity to dispersion problems and their potential for implementation in a GFLo framework.

Dispersion problems

Facility dispersion problems traditionally involve maximising the separation distance between facilities for various reasons that depend on the type of facility included in the problem — such facilities are those that may be considered for applications including military defense, franchise location, transportation of hazardous materials, layout planning for explosive chemicals and telecommunication network design [78]. An example of a typical dispersion problem in a military context is one where a decision maker may attempt to locate ammunition dumps in such a manner that the discovery and subsequent destruction of one such facility has as little impact as possible on the other facilities [110]. An illustration of the general notion of distance-based facility dispersion is provided in Figure 2.1.

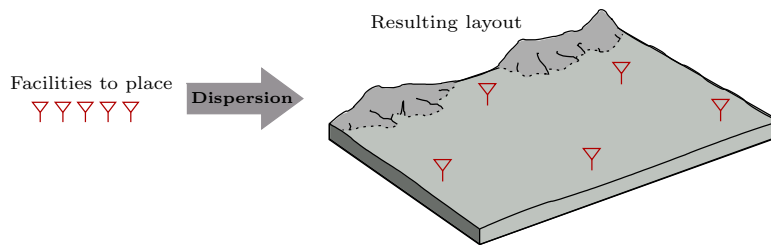


FIGURE 2.1: The general notion of distance-based facility dispersion, in which the separation distances between facilities are maximised.

There exist four basic dispersion facility location criteria in the literature [16, 33]:

1. the p -dispersion criterion, also known as the *MaxMinMin* criterion: First suggested by Shier [116], p facilities are to be located with the aim of maximising the minimum distance separating any two facilities.
2. the p -defense criterion, also known as the *MaxSumMin* criterion: Proposed by Moon and Chaudhry [91], the objective is to locate p facilities in order to maximise the sum of the minimum separation distances from each located facility to the other facilities.
3. the p -dispersion sum criterion, also known as the *MaxSumSum* criterion: Introduced by Kuby [74], this problem involves maximising the sum, over each facility, of the sum of all separation distances for that facility.
4. the *MaxMinSum* criterion: Suggested by Erkut and Neuman [33], this problem deals with the location of p facilities with the aim of maximising the smallest of the facility defined sums. A facility sum represents the sum of separation distances from a specific facility location to all other facilities.

The most recent advances in dispersion modelling come in the form of *unified* dispersion models by Lei and Church [76, 78]. They proposed a generalised formulation encapsulating the four basic dispersion criteria in a unified model, in [76], and proposed an improved formulation in [78]. The models include a facility and a system level. At the facility level, a partial-sum dispersion metric is incorporated, accounting for the sum of the smallest L distances to neighbouring facilities and weighted by a set of propulsion factors. These propulsion factors are used, for example, to

emphasise the interaction between closely located facilities. At the system level, the sum of the K smallest facility-defined partial sums are considered. The partial-sum dispersion metric provides an alternative to the *one-or-all* approach found in the four basic dispersion models described above and makes more sense in modelling many types of inter-facility interactions, according to Lei and Church [78]. One example that they provide to illustrate this fact is in a military defense context, where it is common to locate assets, such as missile silos, apart from one another in order to minimise the chance that two facilities are destroyed simultaneously. If, however, avoiding the simultaneous loss of two facilities leads to low average spacing between facilities, it may be desirable to devise a layout that minimises the chance of losing three or more facilities simultaneously¹.

Another recent and particularly useful advancement of p -dispersion problems is the modelling of the problems to include *multiple* facility types [16]. Here, a multi-type dispersion metric is conceptualised — this metric recognises that the extent to which two facilities ought to be dispersed should depend on their types. The strength of facility interaction is modelled using the traditional separation distance adjusted according to a type-specific ‘repulsion’ factor — a smaller repulsion measure reflects a stronger (friendlier) inter-facility interaction. Four dispersion models based on the four basic dispersion constructs were proposed, in addition to the multi-type metric where each multi-type dispersion model is a multi-type extension of a classic dispersion model.

In a generic GFLo solution framework, the inclusion of the unified models of Lei and Church merits consideration — as they themselves point out [76], however, their models are limited to include a single facility type only and incorporating multi-facility type functionality is an important future consideration for their work. The multiple facility type dispersion models of Curtin and Church [16], on the other hand, offer separate models for the four basic dispersion constructs and includes multi-facility functionality. Whereas a unified dispersion model is considered to be helpful, multi-facility dispersion models are considered *essential* in terms of practicality and for the purposes of future GFLo solution approaches. For these reasons the multi-facility models of Curtin and Church are considered in this dissertation. Furthermore, in order to narrow the scope of the work covered in this dissertation, only the p -dispersion (MaxMinMin) model is included in the framework proposed in this dissertation.

Centre and median problems

As opposed to dispersion problems, centre and median problems involve *minimising* the separation distance between facilities and *demand points*. In a traditional operations research context, problems that require such optimisation include the location of facilities such as libraries, schools, and emergency service centres, to which proximity from a demand point is desirable [98].

The centre and median facility location criteria in the literature include the following [98]:

1. The p -centre criterion, also known as the *minimax* criterion: First introduced by Hakimi [50], p facilities are to be located with the aim of minimising the maximum distance between any of the facilities and the demand points.
2. The p -median criterion, also known as the *minisum* criterion: Also introduced by Hakimi [50], the objective is to locate p facilities in order to minimise the sum of the maximum separation distances between each facility and the demand points.

¹See [76, 78] for further examples of partial-sum dispersion models.

As in the case of the unification models for dispersion problems by Lei and Church [76, 78], unification models that encapsulate and extend the p -centre and p -median problems have also been proposed [81, 96, 97], most recently also by Lei and Church [77].

In contrast to the dispersion models, however, a search in the literature for *inter-facility* p -centre or p -median models returned no results or references to such possibilities. For consistency and practical purposes — and in pursuit of a multi-type p -dispersion model in this dissertation as previously discussed — an endeavour of this dissertation will be the development of a multi-type, inter-facility p -centre model formulation for inclusion in a GFLo framework. The notion of distance-based *inter-facility* centring is illustrated in Figure 2.2.

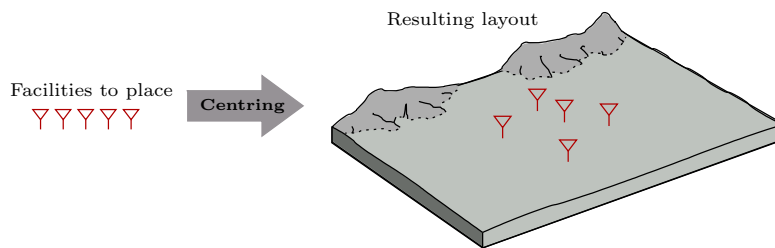


FIGURE 2.2: The notion of distance-based inter-facility centring, in which the separation distances between facilities are minimised.

2.2.2 Covering problems

Covering problems appear in a variety of problems in the operations research literature [35]. Their use in GFLoPs is not uncommon and is prevalent in problems involving the siting of facilities such as transmitter devices, surveillance devices, radar, weapon systems and wind turbines. The two most commonly found and practically implemented covering problems encountered in the GFLo literature, namely the *maximal covering location problem* (MCLP) and the *backup covering location problem* (BCLP) (which intrinsically includes the MCLP), are reviewed in this section.

Traditionally, covering problems aim to locate facilities in such a manner that customers are optimally served according to an objective function typically related to the distance between the facilities and the customers. In a GFLo context, the customers are considered points on the terrain surface that require ‘service’ from the facilities in terms of certain criteria — *e.g.* cellular network coverage required from transmitters in regions of high cellular demand, or visibility required by critical zones from surveillance cameras.

The maximal covering location problem

The MCLP is a natural extension of the *location set covering problem* (LSCP) initially proposed by Toregas *et al.* [129]. The objective in the LSCP is to minimise the number of facilities that are placed in order to service *all* customers in a set of customers that require service. Solving the LSCP involves an undetermined and possibly impractically large number of facilities to be placed in order to satisfy the objective of total demand coverage — this approach is often considered financially and logistically impractical, in addition to being considered unnecessary². For these reasons the LSCP is not considered in this dissertation. An extension of the LSCP

²See [7] for an interesting comparison of the LSCP and MSCP alternatives implemented in the context of a surveillance camera siting problem, in which these considerations are practically illustrated.

that makes more practical sense (and which is generally adopted in GFLoP contexts instead of the LSCP) is the MCLP proposed by Church *et al.* [12]. The aim of the MCLP is to maximise cover of demand given a fixed number of facilities that are available for placement.

The MCLP is most often included in GFLoPs concerned with the placement of transmission devices. In these problems, the aim is to locate transmitters in a wireless network, *e.g.* for cellular communication or television broadcast purposes. The objective is to provide optimal coverage with respect to a number of demand points. The standard MCLP formulation may be directly used in these applications — for VHF/UHF radio jammer systems in [40] and for transmitter towers in [73] — or interference may be added to the model. The aforementioned interference may be included in the model either as an additional term subtracted from the objective function [80, 83], or by the introduction of auxiliary variables included in the objective function [2].

GFLoPs involving facilities destined for surveillance purposes — such as watchtowers or cameras — also implement the MCLP in certain cases [7, 66], although the BCLP is more regularly encountered in this context. Kim *et al.* [66] conducted a well-cited study that considers the MCLP for general visibility coverage purposes with no specific facility type in mind [66], while Bao *et al.* solved the MCLP (in addition to the LSCP) for forest fire monitoring purposes [7].

The MCLP also appears in military and related studies. In addition to the VHF/UHF radio jammer system application previously mentioned in the context of transmission devices [40], radar and weapon systems also exhibit objectives that may be solved by the MCLP. Whereas the previously discussed literature on transmitter and surveillance facilities involved the coverage of terrain surface/points of interest on the terrain surface, the points that require service in a military context may be points along an enemy's possible approach path [43, 125] and the 'service' is provided by radar or weapon systems in the form of detection or engagement efficiency. In [125], the simultaneous siting of radar and weapon systems is solved in this context — although a closely related extension to the MCLP, the *maximal expected covering location problem* [18], is used instead.

The final category of applications of the MCLP is that of wind turbines — although, currently, wind turbine siting problems do not involve the MCLP directly. Wind turbine siting problems traditionally adopt SO optimisation, where the single (summed) objective function generally includes wind energy potential and wind turbine wake effects [31, 47, 92, 130]. The recent increase in the implementation of *MO optimisation* (MOO) algorithms in the solution processes of wind turbine micro-siting problems has, however, led to the possibility of additional objective functions, of which ones that include noise levels appear to be the first researched [75, 140]. The additional objective function aims to *minimise* predicted noise levels at certain 'receptors' — this may, however, be implemented as an MCLP that aims to maximise the *inverse* of predicted noise levels experienced at the receptors. Reducing the visibility of wind turbines from certain areas (see [90] for a related study) is another important consideration that may be implemented in future wind turbine siting problems as an additional MCLP objective, in a similar vein as the visibility problems previously discussed in this section. The objective in this case would be to maximise the number of points of interest that are *not* visible from the wind turbines.

The backup covering location problem

The BCLP is most commonly found in problems related to surveillance, since in surveillance monitoring the maximal coverage obtained by solving the MCLP is not always sufficient in the design of robust systems — ensuring coverage overlap for failsafe purposes is an additional, crucial objective. The BCLP is therefore an extension to the MCLP, originally proposed in

a traditional operations research context for the purpose of siting emergency services [19, 56]. An illustration of the two covering criteria which may be considered in MCLPs and BCLPs is provided in Figure 2.3.

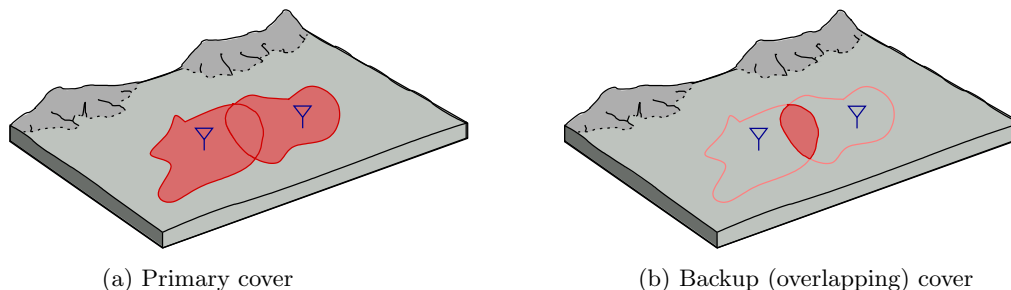


FIGURE 2.3: (a) Primary and (b) backup (overlapping) coverage (the shaded areas) considered in a geospatial context of the covering class of traditional FLoPs.

Inclusion of the BCLP criteria in GFLoPs is generally found in surveillance sensor siting problems. The most popular applications from the literature include the pioneering work by Murray *et al.* [93], utilising a SO BCLP implementation, while their research is furthered in [65] to consider an MO implementation of the BCLP. Another example of a SO implementation of the BCLP may be found in [135]. A weather radar-related study that indirectly implements the BCLP may be found in [88]. In this study there are three objectives: (1) to minimise the total cost of the network installation, and to maximise (2) primary and (3) overlapping cover of airspace relevant to weather prediction. The second and third objectives are essentially the BCLP, and are summed together with the first into a single weighted-sum objective in this case.

It is interesting to note that certain transmitter problems discussed in the previous section indirectly consider an interesting variation on the BCLP. In [80, 83], interference is considered in the MCLP formulation in the form of an additional interference term that is subtracted from the primary coverage received by the points of interest. This is essentially a SO version of the BCLP, with the aim of *minimising* backup cover. The aim to minimise backup cover in transmission siting problems stems from the fact that multiple coverage may often result in *weakened* signal quality, depending on the type of network [80, 83]. Solving similar problems by means of the traditional BCLP in an MO sense, with the aim of *minimising* overlapping cover (or maximising non-overlapping cover) as a second objective function may be an interesting and superior alternative to the SO approach discussed here.

2.3 GFLo optimisation approach considerations

The most important considerations pertaining to GFLo optimisation approaches in the literature are considered in this section. These considerations include GFLo solution representation and evaluation, exact versus approximate solution approaches, SO versus MO optimisation approaches and single-solution versus population-based optimisation.

2.3.1 Point-based GFLo solution representation and evaluation

The space of location decisions in point-based facility location problems may generally be categorised as continuous or discrete [110] (previously discussed in §2.1). In continuous problems, the points to be sited can generally be placed anywhere on the plane, while in discrete problems

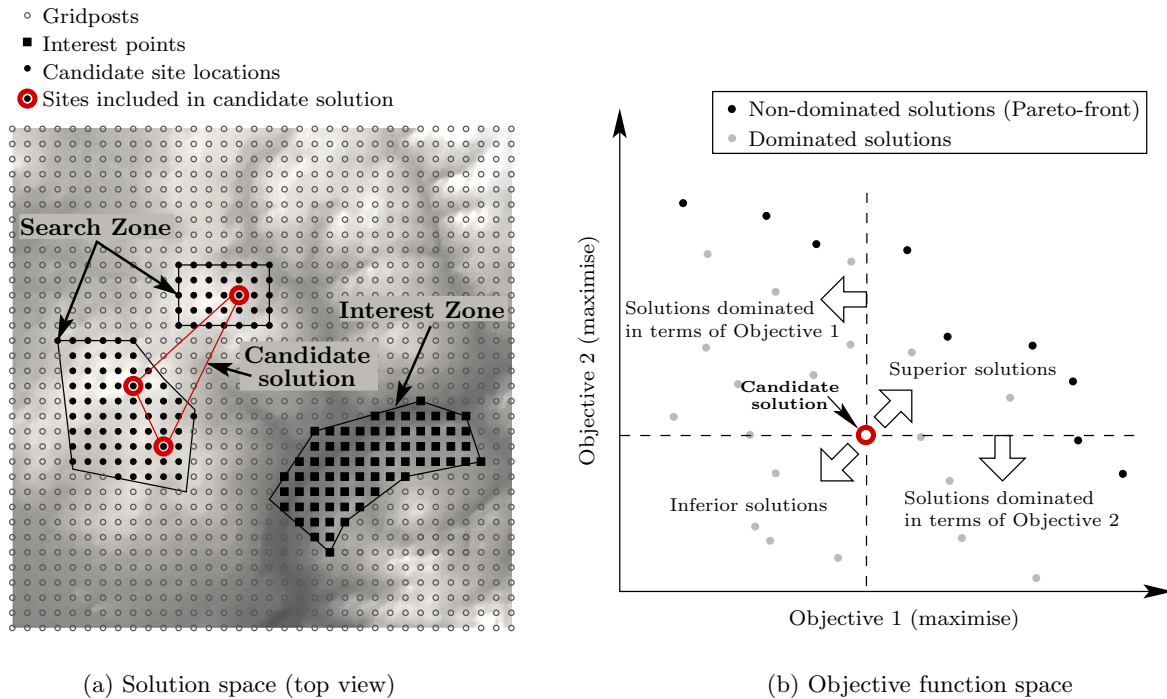


FIGURE 2.4: (a) Terrain represented by raster data and its application to GFLoPs in solution space. (b) The notions of solution domination by a candidate solution and of a Pareto-front of non-dominated solutions in objective function space.

the facilities can be placed only at a limited number of pre-selected candidate sites (eligible points) on the plane. GFLoPs are solved as discrete facility location problems in this dissertation — for which raster data are suited due to its ease of implementation and extensive use in the literature [37, 43, 53, 66, 75, 114, 125]. Raster data represent the earth’s surface and environmental information as uniformly spaced sample points, called *gridposts*, across the terrain surface [52, 54, 132]. These gridposts may be considered as candidate facility site locations. Examples of discrete GFLoPs that employ raster data include the placement of wind turbines [75, 114], radar and weapon systems [43, 125], and other *line-of-sight* (LOS)-dependent facilities [37, 53, 66].

An illustration of a raster data representation of terrain is provided in Figure 2.4 (a). The section of terrain surface that is shown in this figure is, in fact, a graphical representation of sampled elevation data at the gridposts (the empty dots). *Search zones* (SZs) are specified on the terrain surface and the gridposts enveloped by the designated SZ boundaries are candidate site locations that may be considered for facility placement (the solid dots within the SZ boundaries). A candidate solution is a specific configuration of a number of facilities (three for the example in the figure) at candidate sites in the SZ. When solving covering problems, candidate locations in the SZ may be evaluated with respect to their *influence* on gridposts enveloped within specified *interest zones* (IZs) — the solid squares within the IZ boundary.

The objectives of MO GFLoPs and classical FLoPs are often conflicting in nature. For example, the objectives of the BCLP (introduced in §2.2.2) are conflicting in the sense that gains in primary visibility coverage of a set of facilities with respect to some demand area can only be achieved by dispersing the facilities, resulting in a decrease in overlapping visibility coverage, and *vice versa*. As a result of these typically conflicting objectives, a set of trade-off solutions that perform well with respect to the objective functions are sought. The process of configuring a candidate solution takes place in the *solution space*, as illustrated in Figure 2.4 (a), and such

a candidate location combination corresponds to a single point in *objective function space* which measures the scores that a candidate solution's facility placement configuration achieves with respect to the placement criteria, as illustrated in (b).

The aim in MOO is to allow the user to examine trade-offs in objective function space by finding a set of compromised solutions — called the *Pareto-optimal* solutions [70, 141, 142]. These solutions are *non-dominated*, in the sense that no other Pareto-optimal solution is better than another with respect to *all* the objectives. Moving from one Pareto-optimal solution to another results in an improvement in at least one objective, but the degradation in at least one other. The inferior solutions that are not included in the Pareto-optimal set are said to be *dominated* by the Pareto-optimal solutions, because for each dominated solution there exists a Pareto-optimal solution that is better with respect to *all* the objectives. The representation of the set of Pareto-optimal solutions is commonly known as the *Pareto-optimal front*, or simply the *Pareto-front*, as they form a frontier in MO objective space as is visible in Figure 2.4 (b). These solutions may be explored by a decision maker and a solution that seems most appropriate may be chosen. In a GFLo context, this selection entails considering solutions from a subjective perspective both in terms of their proposed facility site locations on the physical terrain and their performance values with respect to the placement criteria.

2.3.2 Exact versus approximate optimisation

Two approaches exist to arrive at the Pareto-optimal front of non-dominated solutions, namely exact optimisation techniques and approximate optimisation techniques [127, 142]. Exact methods are those which guarantee locating the Pareto-optimal solutions to a class of problems, provided that a sufficient amount of processing time is available. One such method is explicit enumeration, whereby every possible facility layout configuration of a GFLoP is analysed [66]. Implicit enumeration methods such as the branch-and-bound method and gradient searches [142], on the other hand, arrive at the same true Pareto-optimal set of solutions with the benefit of not having to evaluate large subsets of candidate solutions that are implicitly proven to contain inferior solutions and, in doing so, reduces the computation time considerably. The advantages of solving a problem exactly is that the true Pareto-optimal set of solutions is found. The disadvantage is that a considerable amount of time may be required to arrive at these Pareto-optimal solutions in the case of a large solution space [66, 70]. This is an option very rarely available in real world problems, owing to the enormous number of candidate solution configurations, even when the problem is solved implicitly.

Heuristics and metaheuristics are search procedures that aim to determine the Pareto-front and are often capable of doing so, but more generally return *approximations* of the Pareto-front [21, 69, 127, 142]. Heuristics are highly problem-specific search procedures [45]. Metaheuristics, on the other hand, are algorithms that operate at a higher level than heuristics and may be defined as a high-level strategy for directing heuristics in the search for optimal solutions in domains where the task is hard [45]. Metaheuristics are often able to locate good solutions which may closely approach the global optimum and are frequently the only viable option solving difficult problems, owing to the large complexity or size of the problem. Metaheuristics are usually able to model problems more realistically and are generally less limited in terms of problem type restrictions (such as requirements of linearity or differentiable functions). Examples of metaheuristics include *evolutionary algorithms* (EAs), *Simulated Annealing* (SA), Tabu Search, and Ant Colony Optimisation [45]. The goal of metaheuristic optimisation is to find good approximations of the Pareto-optimal set and different algorithms may produce sets of differing quality in terms of their closeness to the true Pareto-optimal front, and their diversity along it.

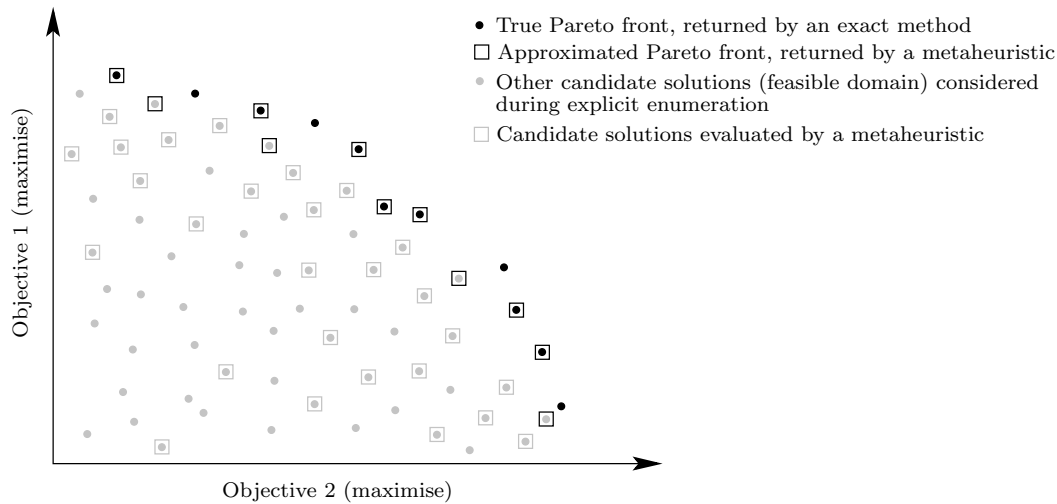


FIGURE 2.5: *Exact versus metaheuristic candidate solutions in objective space for a hypothetical problem with two objectives that are to be maximised.*

A hypothetical illustration in objective space of results returned by an explicit enumeration method and a (meta-)heuristic search technique is provided in Figure 2.5 for a problem with two maximisation objectives. The solutions evaluated by the metaheuristic are a subset of the solutions considered during explicit enumeration, but these solutions typically tend to be more highly concentrated towards an area which performs well with respect to at least one of the objectives later during the search. The final, approximated Pareto-front is similar in shape and location to the true Pareto front, but since metaheuristic procedures do not evaluate the entire solution space, some of the members of the true front do not appear in the approximated front. These omissions, however, come at the gain of a reduced computational burden.

Approximate algorithms are typically used to solve *NP-hard* problems — a problem is NP-hard when it is known that simple enumeration and search techniques, as well as general purpose algorithms, will likely have great difficulty in identifying any of the true Pareto-optimal solutions in the worst case [93]. The classical FLoPs that were discussed in §2.2 and which are included in the general framework proposed in this dissertation are NP-hard [3, 7, 28, 93, 95, 98, 99, 107]. For this reason, and because an MO GFLoP includes at least two (typically NP-hard) FLo objectives, only metaheuristic Pareto-front approximation approaches are followed in this dissertation.

2.3.3 Single-objective versus multi-objective optimisation approaches

Algorithms that aim to approximate the Pareto-optimal set of solutions may do so following two approaches: SO optimisation or MO optimisation. In SO optimisation, solutions are determined with respect to a single objective function only, although multiple criteria are typically considered in these problem instances [57, 122]. The single objective function is typically a weighted combination of multiple criteria. By varying the weighting coefficients of the single objective combination over multiple approximation runs, a set of near-optimal solutions may be arrived at — these solutions ‘trace out’ the Pareto-front [122]. Examples of GFLoPs that are inherently multi-objective, but are solved using SO approaches include wind turbine micro-siting [31, 75, 114] and the BCLP [56, 93].

In contrast to SO optimisation, MOO approaches are able to find multiple members of a (near-)optimal set of solutions in a single run [57]. A fundamental decision must therefore be made as to whether the problem is to be solved by multiple executions of a weighted SO

solution approach or by a single execution of an MO solution approach. The literature on this topic is diverse in opinion as to which of these approaches is better. From the literature it is, however, clear that at least the following has to be considered in such a decision:

- **Complicated assignment and distribution of SO weight schemes:** For the SO approach it is necessary to assign weights to the objectives with the aim of approximating the Pareto-optimal set of MO solutions closely, diversely and evenly [17, 62]. An example of a typical approach in assigning weights is the *linear weighted sum method* [122] alluded to above. Further discussions and examples of weight assignments and their implementations may be found in [57, 58, 59, 62, 101, 123]. The disadvantage of weighting procedures like these is that equally spaced weight sets (which are used often) may not result in good approximations of the Pareto-optimal set and any attempts at finding weight sets that perform well in this respect may be tedious and highly problem-specific. Further complexities associated with weighted sums of objectives are described by Das *et al.* in [17]. Compared to the weighted SO approach, the MO approach does not require weight assignments and the diversity of solutions in the objective space are intrinsically dealt with by the metaheuristics employed.
- **SO approaches return one solution at a time:** In order to approximate the Pareto front using a SO approach it is necessary to execute a SO metaheuristic multiple times — at least once per combination of weights over a pre-determined, balanced design of weight sets — whereas an MO approach requires a single execution in order to return a set of approximately Pareto-optimal solutions [57]. Depending on the algorithm parameters and the nature of the problem, executing the SO approach in such a manner for multiple weight sets may require significantly more time to carry out. Also, one SO metaheuristic execution for a specific weight combination does not guarantee that the solution returned is close to optimal; multiple executions are typically required per weight combination in order to achieve satisfactory solution quality. Compared to MO approaches, this may require significantly more floating point operations — even in instances where the MO approach is also executed multiple times.
- **MO approaches may struggle to converge with many objectives:** As the number of objectives increases, non-dominated solutions may become rare (as there are more objectives, the likelihood of domination with respect to *all* the objectives decreases). In such cases, MO methods may struggle to converge and may ultimately be outperformed by SO methods. Hughes [57] and Purshouse *et al.* [101] describe this phenomenon in detail and practical examples may be found in [57, 58].

Proponents of MO approaches often hold that weighting objectives together is bad practice which may result in poor objectivity and that multi-objectivising problems that are essentially single objective may even result in improved optimisation [69, 123, 138]. Of note is that hybrid approaches also exist, such as the one proposed in [58]. In the context of a visibility-related BCLP, Kim *et al.* [65] demonstrated how an MO approach offer substantial benefits over the weighted-sum method [93] and the true Pareto-front was, in fact, obtained by an approximation algorithm (for a problem of significantly lower complexity than the intended purposes of the GFLo framework in this dissertation).

Considering the above-mentioned trade-off between the advantages and disadvantages of SO and MO solution methodologies, only MO approaches are investigated in this dissertation.

2.3.4 Single-solution versus population-based optimisation

MO metaheuristics vary widely in type, yet may be classified into two principal categories. *Single-solution* algorithms focus on repetitively improving a single candidate solution through global and/or local search techniques [44, 67, 89]. Traditionally, single solution algorithms consider a single objective function only — it should, however, be noted that some essentially single-solution algorithms, such as Variable Neighbourhood Search and SA, have been adapted for use in an MO context [29, 119]. Single-solution methods have been proven to obtain excellent results for a variety of optimisation problems. They are, however, not considered appropriate for GFLoPs, since multiple solutions that provide trade-offs in solution performance with respect to a set of objective functions, in addition to varying alternatives with respect to the physical site locations that are proposed by the candidate solutions, are desired. This is because of the significant role that physical site selection plays in the GFLo process, so that a set of candidate solutions offering various configuration alternatives may be provided to a decision maker who, in turn, may study the alternatives and select a solution considered acceptable, both in terms of its physical site locations and associated objective function values.

As opposed to single-solution search methods, *population-based* algorithms aim to iteratively improve a set of candidate solutions, referred to as a population [8, 22]. Population-based algorithms start with a number of stochastically generated ‘guess’ solutions, after which the algorithm iteratively manipulates a number of the solutions in the population in the hope of steering it towards good solutions [22]. Deb [22] lists numerous advantages of using population-based algorithms, which are especially important in the context of GFLoPs. These are (1) a population of solutions provides the information of a good search *region*, instead of only a good point, (2) population-based algorithms offer *self-adaptive* possibilities, according to the convergence of the population towards the optimum³, and (3) the fact that the presence of multiple solutions allows for solution *diversity*, amongst others.

Considering the characteristics of population-based algorithms mentioned here and the solution requirements of a GFLoP, population-based algorithms are considered to be the most suitable for implementation in a GFLo framework and are investigated in the remainder of this dissertation.

2.4 Approximate algorithms for solving GFLoPs

The literature on practical MO GFLoPs solved in the literature is scarce. Typically, GFLoPs are solved using SO approaches [7, 31, 47, 66, 73, 83, 92, 130]. This is, however, not always due to a choice of using the weighted-sum method (discussed in §2.3.3), but simply because there is only one objective [7, 66, 73]. In SO approaches, the most popular algorithms are EAs employed for the purposes of visibility/transmission cover [3, 7, 66, 73, 105, 128] and for wind turbine siting [31, 47, 92], and SA employed for transmission and visibility cover purposes [66, 83]. MO GFLoPs are typically solved by *genetic algorithms* (GAs) which are members of the class of EAs — examples include radio network optimisation in terms of coverage achieved [86, 105], backup cover achieved for surveillance networks [65], and recently for the wind turbine micro-siting problem [75].

EAs are a popular sub-class of the class of population-based algorithms. In fact, almost all population-based algorithms are EAs [22]. EAs evolve a population of candidate solutions to an optimisation problem based on natural principles [8, 106]: the basic premise involves an initial, randomly generated population of candidate solutions which undergoes multiple generations of

³Such a self-adaptive method for a population-based algorithm is investigated in this dissertation.

carefully controlled evolution in order to arrive at a final generation of solutions that approximate the Pareto-optimal set of solutions [21, 106] of the problem. During each iteration, the current population generates an offspring population by mimicking natural selection and mutation processes. In GAs, the offspring population comprises solutions that are represented as *chromosome* strings of decision variables. The offspring population results from the application of a *crossover* operator between solutions selected from the current population and these solutions are, in turn, subject to *mutation*. Once the offspring population has been computed, the current and offspring populations are combined and the best solutions in this combination are carried over to the next generation [8, 21, 106].

In the context of MOEAs and GFLo, the *non-dominated sorting genetic algorithm-II* (NSGA-II) is popular and has been used for solving MO GFLoPs such as the BCLP [65], antenna placement [105] and wind turbine micro-siting [65, 75, 140]. In a comparative study concerning an antenna placement problem [105], the NSGA-II of Deb *et al.* [21] was compared to three other popular GAs, namely the *simple evolutionary algorithm for multi-objective optimisation* (SEAMO) [137], the *strength Pareto evolutionary algorithm 2* (SPEA 2) [143] and the *Pareto-envelope based selection algorithm* (PESA) [15]. Considering its consistent comparative quality of solutions obtained over a number of problem instances, it was concluded in their study that the NSGA-II is the strongest performing algorithm for the purposes of antenna placement planning.

MOEAs, and the NSGA-II in particular, offer the type of solution approaches that are desired for GFLoPs, in the sense that they (generally) provide a diverse set of trade-off solutions that approximate the Pareto-optimal front, and are known to achieve good results fast [3]. Due to the NSGA-II's good reputation and already proven success in solving MO GFLoPs in the literature [65, 75, 105, 140], along with the application of similar GAs to solve GFLoPs (whether SO or MO) [31, 47, 66, 86, 92, 130], the NSGA-II will be considered for inclusion in the GFLo framework of this dissertation.

In 2004, Kim *et al.* [66] performed an in-depth study into visibility-related facility placement applications involving the MCLP. In this study they considered three algorithms, namely a simple swap algorithm [126], the NSGA-II [21], and SA [67]. The MCLP was solved for observers ranging in number from two to ten, for a 20 km×20 km square section of terrain. The swap algorithm returned results similar to those of the NSGA-II and SA. Its results were, however, returned in significantly shorter computation times. This stems from the simplicity of the swap based algorithm used in the study. In the study SO algorithms were, however, considered. Nevertheless, even in an SO context the efficiency and speed of the swap-based algorithm compared favourably with those of the more complex algorithms. The swap algorithm used in this study is also known as the *interchange algorithm*, or the *Teitz & Bart algorithm* (TBA) [3, 66, 126].

The TBA is one of the most popular algorithms used in facility location [28, 109, 111] and is in essence a single-solution, SO swap algorithm [126]. The TBA starts by randomly generating a single starting solution comprising n_f facility placements. The typical approach is to identify a set of candidate locations that are not included in the current solution, after which a candidate location in this set is randomly selected and subsequently swapped with each location in the current solution. If any objective function value improvements are achieved as a result of candidate location swaps with the current locations, the swap that results in the best improvement in the (single) objective function value is accepted as the incumbent solution [126]. This procedure is repeated for the remaining locations in the candidate location set, which is continually updated relative to the incumbent solution, until some termination criterion has been met and the final incumbent solution is accepted as the best alternative.

The literature review in this section has mainly focussed on MOEAs and their applications in GFLoPs. This is because GFLoPs do not typically include the FLo models that are mostly

concerned with distance-related criteria, namely the p -dispersion, p -centre and p -median problems (previously discussed in §2.2.1), and are mostly concerned with terrain and environmental criteria related to covering problems and wind-energy projects [66, 75, 140]. The TBA and its implementation in other algorithms have, however, been shown to provide consistently good results for distance-related problems and its simplicity and speed makes it a very attractive heuristic for these purposes [109, 111, 136, 139]. The algorithm was, in fact, inspired by the p -median problem [126]. In addition to the good results achieved by the TBA in the study by Kim [66], two other interesting studies of importance to the GFLo framework put forward in this dissertation were performed in [3] and [100]. A GA implementation of the TBA achieved impressive results for the p -median problem in [3]. Alp *et al.* followed the general premise of population evolution of MOEAs and implemented a combination strategy instead of crossover. The solution space was also explored with TBA-inspired mutation. They report a worst-case result of only 0.41% away from the optimum and claim very fast computation times. The TBA has mostly been limited to implementation in the distance-based category of facility location problems. In a 2015 study of the MCLP for the purposes of sensor deployment in a military environment [100], however, the TBA consistently outperformed two population-based algorithms (one of which was a GA). Pond *et al.* come to the conclusion that the success of the TBA is due to its ability to efficiently explore a larger number of candidate locations than the other two algorithms.

Given the above-mentioned performance of the TBA for distance-based FLoPs earmarked for inclusion in the GFLo framework proposed in this dissertation, and the recently obtained results in [100] for an instance in the class of cover problems (which is probably the most prevalent class of classical FLoPs observed in a GFLo context), it may be concluded that the TBA — in a GA implementation that provides the advantages of a population-based approach similar to that in [3] — is suitable for inclusion in the GFLo framework and is pursued in this dissertation.

2.5 Chapter summary

This chapter opened in §2.1 with an introduction to the facility location branch of operations research. This was followed by an introduction to classical FLoPs whose constituent elements (objective functions and constraints) may be considered for inclusion in the GFLo framework proposed in this dissertation. These FLoPs included dispersion and centre problems, discussed in §2.2.1, and cover problems, elucidated in §2.2.2.

Fundamental solution approach considerations were discussed in §2.3. This included solution representation and evaluation, exact versus approximate solution, SO versus MO optimisation approaches and single-solution versus population-based optimisation, and was followed by a survey of MOO algorithms used to solve GFLoPs in the literature. It was found that the NSGA-II is the most popular algorithm and was selected for inclusion in the GFLo framework proposed in this dissertation. Impressive results returned by a swap-based algorithm compared to the NSGA-II in a visibility-related study was the inspiration for pursuing the development of an MO version of the algorithm in this dissertation.

CHAPTER 3

GFLo data management and modelling

Contents

3.1	The GFLo database	21
3.1.1	<i>The facility database</i>	21
3.1.2	<i>The geospatial database</i>	22
3.1.3	<i>Geospatial data operations</i>	23
3.2	Practical data considerations	26
3.3	Classification of GFLo criteria	28
3.3.1	<i>Location-specific criteria</i>	28
3.3.2	<i>Inter-site criteria</i>	29
3.4	Mathematical modelling of the GFLo environment	31
3.4.1	<i>Modelling search and interest zones</i>	32
3.4.2	<i>Modelling facilities</i>	33
3.4.3	<i>Modelling facility placement criteria</i>	36
3.5	Chapter summary	39

A comprehensive, generic GFLo framework requires data management in respect of the facilities and geospatial data that are considered in the problem — contained in a GFLo database — in addition to the modelling of the environment within which the facility and geospatial data are incorporated for decision-making purposes. This chapter opens in §3.1 with a description of the separate facility and geospatial databases that are proposed for this purpose. This is followed in §3.2 by a discussion on practical data considerations relevant to the geospatial database. A classification of GFLo criteria types according to similarities that they exhibit is proposed in §3.3, and this is followed in §3.4 by a mathematical modelling framework for GFLoPs.

3.1 The GFLo database

Two separate databases are proposed to make up the GFLo database. These are the *facility database* and the *geospatial database* and are introduced in §3.1.1 and §3.1.2, respectively.

3.1.1 The facility database

The facility database contains essential facility-related information that is specified during preliminary preparation of GFLoP instances. Care needs to be taken in specifying facility-related

data to be stored in the facility database, as these may in some instances be included in placement criteria according to which the facilities are evaluated and may ultimately affect the outcome of a GFLO study. The facility database may include, but is not limited to, the following information:

- **Facility types.** This category of information includes an identification of the types of facilities to be placed, *e.g.* radars, watchtowers or wind turbines. It may also be possible to consider multiple types of facilities in the same placement problem — a network of radars and watchtowers in a military environment, for example. Different modes of the same facility types may also be included. For example, long-range radars and short range radars may be included in the ‘radar’ facility type.
- **Facility availability.** The maximum number of each facility that is available for placement.
- **Facility costs.** The purchase, production or placement cost incurred when locating a facility.
- **Facility capabilities.** These data refer to the detection, generation or production capabilities of the facilities. Examples are the maximum detection range of a radar or the maximum power output per hour that a specific wind turbine or solar panel can generate under ideal conditions.
- **Facility requirements.** These data can be specified as network or functional requirements. Examples of network requirements are inter-visibility and proximity specifications — a radar in a network may, for example, be required to be visible to at least two watchtowers in the same network and be within a specified distance from both. Functional requirements are the various requirements of the facilities which ensure that they function properly or optimally — *e.g.* the level of solar energy required for a solar panel to generate a specified amount of electricity.
- **Facility dimensions.** The physical dimensions of each type of facility that is available for placement. It is possible that some larger facilities may occupy more than a single gridpost, which introduces orientation concerns and additional criteria-related computations per additional gridpost. In this dissertation, however, only facilities that are small enough to have their location represented by a single gridpost are considered.

3.1.2 The geospatial database

The geospatial database contains terrain and environmental data which must be managed efficiently and may require large storage volumes.

Geospatial layer data

In order to correctly represent the SZs and IZs for use in GFLOPs, appropriate geographical layers need to be collected and stored in the geospatial database. Examples of such layers include elevation data, municipal plot boundaries, natural park boundaries and various types of weather measurements. Depending on the data involved, such layers are readily available online, either free of charge or to purchase at rates determined by a subscription or per layer [38, 41, 42, 131]. Otherwise, custom layers may be created through the use of freely available *geographic information system* (GIS) software, such as gvSIG or QGIS [49, 103], or licensed GIS software, such as the popular ArcGIS [5]. For the purposes of this dissertation, geospatial data

were acquired from various online sources, such as those in [38, 41, 42, 131], and geospatial operations and data formatting were performed using ArcGIS and MATLAB [84].

Analysis zones

The SZs and IZs require prior analyses in order to be in the format that is required for implementation in GFLoPs. Before the zones are considered suitable for inclusion in a GFLoP, they are specified in generalised forms as *unrefined* zones. This generalised description provides the detail required to collect the necessary data for processing in order to arrive at the final SZs and IZs that are implemented in the GFLoP solution framework. Two types of unrefined forms of the GFLo zones are the following:

- **Unrefined Search Zone.** This type of zone includes the areas of terrain that are considered to contain potential facility location sites, before having performed detailed identification and filtering of suitable and unsuitable areas within it. The specification of an unrefined SZ may be vague (for example, a rectangularly specified latitude/longitude boundary) or slightly more detailed (for example, the terrain surface included within the boundaries of a province or municipal area). The unrefined SZ may also be a grouping of multiple zones. Multiple demarcated hilly areas in the surrounds of an urban area may, for example, be considered for telecommunication tower placement.
- **Unrefined Interest Zone.** When facilities are placed in the unrefined SZ, it is likely that their placement may directly or indirectly affect certain areas of interest in a multitude of ways, either positively or negatively. These affected areas are grouped together to form the unrefined IZ. The placement of facilities may be directly related to the unrefined IZ, such as the placement of telecommunication towers in the unrefined SZ so that they provide optimal coverage with respect to a specified unrefined IZ. Indirect effects include, for example, the potentially negative aesthetic or auditory impact of a wind turbine grid placement in the unrefined SZ on residential areas that make up the unrefined IZ.

3.1.3 Geospatial data operations

The data stored in the geospatial database may be used to analyse the unrefined SZ and unrefined IZ so that only suitable and authorised areas are considered for the purposes of facility placement. Reducing the unrefined SZ and unrefined IZ in this manner ensures that solutions to the location problem at hand are found more efficiently and are practical from an application point of view. In addition, data storage requirements may be significantly reduced if only the data related to feasible SZs and relevant IZs are extracted from source data and stored to disk. Reducing the data in such a manner requires layer pre-processing and additional data analyses.

Layer pre-processing

After the unrefined SZ and unrefined IZ reductions have been performed through layer processing, additional layered terrain zone reductions may be performed to extract data relevant to facility placement. As demonstrated in §2.3.1, raster data are employed in this dissertation. During preliminary geospatial data analyses processes, however, *vector* data are often employed. These data represent geographic features as vectors and typically define boundaries and spatial geometries. The vectors are represented by dots, lines and polygons [112] and are regularly used for the extraction of data that lie within feasible areas. An example of unrefined SZ to reduced

SZ processing is that of extracting elevation data from an elevation layer for the purposes of placing watchtowers in a national park, using a national park boundary layer, contained in vector data, as a mask to encapsulate the relevant terrain elevation data. This extraction is then stored as a new layer in the database, after which a visibility analysis may be performed on the reduced SZ elevation data with respect to the reduced IZ (or the unrefined IZ, depending on whether any IZ processing has been performed). Unrefined IZ to reduced IZ processing may be performed in the same manner. The reductions may be specialised even further by using purpose-made GIS analysis tools. A layer may, for example, be created by determining the degree of slope between consecutive gridposts within a SZ and saving the areas of acceptable slope (typically below some pre-specified maximum slope threshold) as a new layer. This new layer may then be used for additional extraction or analysis purposes. A practical example of reducing the SZ according to infeasible slope is available in [53].

It is important to consider the type of placement criteria included in a GFLoP in order to store the correct data. For example, a SZ may be reduced in such a manner that the feasible terrain surface includes intermittent areas that have been removed due to restrictions or infeasible terrain characteristics. Certain analyses performed on the terrain may, however, still require the inclusion of these areas in their evaluation process. Visibility-related analyses, for example, require consideration of the entire terrain surface area's interference between two points for which inter-visibility is computed, in which case an absence of data in areas that have been removed due to facility location restrictions will make this type of analysis impossible to perform. In these instances, feasible SZ data may be extracted for the location-specific analyses, but larger data sets have to remain in the geospatial database for the purposes of analyses such as those that are visibility-related.

Identifying likely suitable placement locations related to terrain characteristics and unique facility requirements related to the specific type of facility is possible [66]. Such identification may result in further reductions of the reduced SZ or reduced IZ, although these reductions may lead to the exclusion of good candidate solutions. The possible disregard of good solutions may, however, be compensated for by a reduction in the resulting time required to compute solutions. An example of reducing the search space in this manner is found in [66], where specific terrain analyses are performed to identify areas on the terrain surface that are likely to achieve good visibility of the surrounding terrain, such as ridges and peaks, for the purposes of a subsequent viewshed-dependent facility location. Such reductions are, however, not considered suitable in an MO environment, since the removal of areas that are considered to be 'weak' in respect of one criterion may result in the removal of areas that may be considered 'good' in respect of another. This approach is therefore not pursued in this dissertation.

Additional analyses

Following the determination of the reduced SZ and the reduced IZ, specific analyses related to the objectives of the GFLoP may be performed. Again, these analyses may be performed using GIS software or other specialised scientific software, such as MATLAB, depending on the type of analysis required and user preference.

In some cases, GIS-related analysis following SZ and IZ reductions, as discussed above, may not be necessary — for example, if the objective is to place a number of facilities as a function of the degree of slope at each location, performing custom layer filtering already provides the necessary data to solve the problem. For some computations, such as inter-visibility computation, however, further analysis of the reduced SZ is required with respect to the reduced IZ. This may be performed either *a priori* and stored to disk, or during a metaheuristic search process in which a

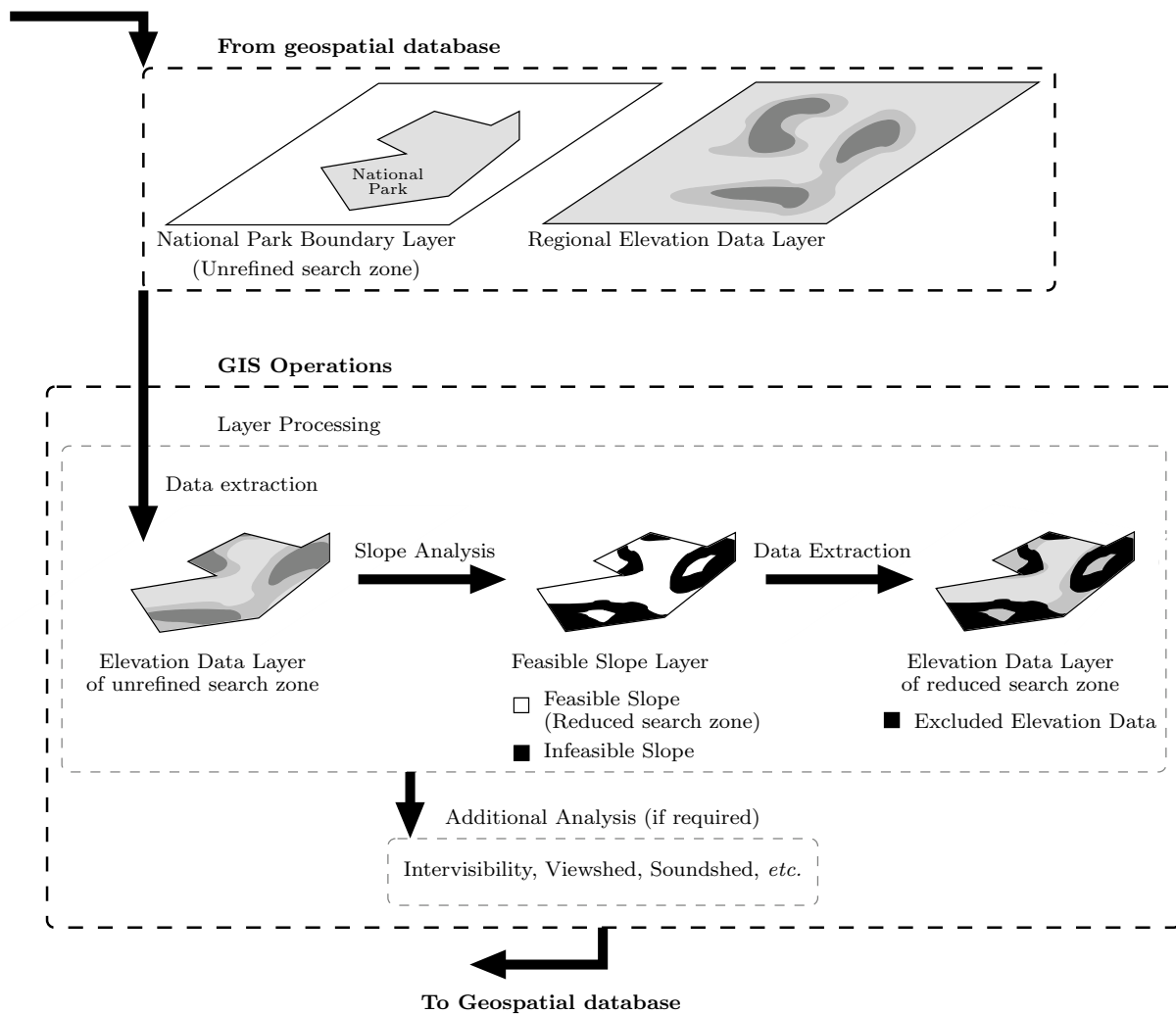


FIGURE 3.1: Illustration of a typical set of GIS-based pre-processing operations. Data are acquired from the geospatial database, after which layer processing and additional analyses are performed. The new data are then stored in the geospatial database again.

caching approach is followed. Such an approach involves storing data as they are required during the search process, making them available for fast retrieval in future analyses. The approach followed depends on the type of criteria evaluated in the GFLoP as well as the associated computational complexity, and is an important consideration for the user.

A typical GIS-based pre-processing operation is illustrated in Figure 3.1. The unrefined SZ and relevant elevation data are acquired from the geospatial database, after which layer processing may be performed. This includes the extraction of elevation data that fall within the unrefined SZ, after which slope analysis may be performed on the extracted elevation data in order to create a layer indicating areas within the unrefined SZ that satisfy some pre-determined slope constraint — this feasible area subsequently forms the reduced SZ. Further pre-processing may be performed, such as extracting elevation data that fall within the reduced SZ. The new data should then be stored in the geospatial database.

An illustration of the GFLo database framework proposed in §3.1.1–§3.1.3 is provided in Figure 3.2.

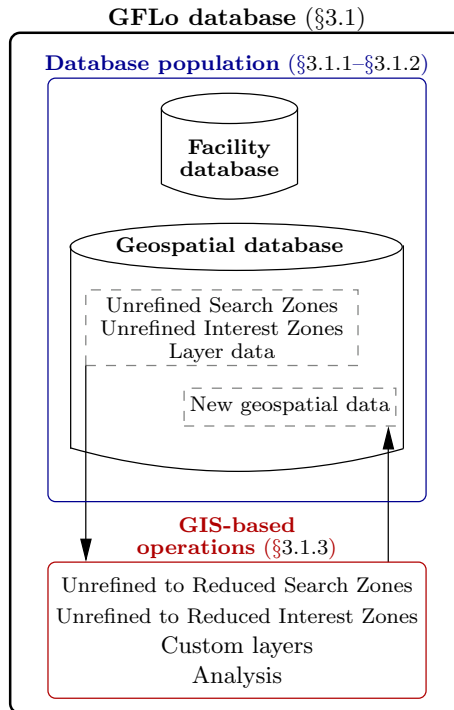


FIGURE 3.2: A GFLO database and associated operations, as proposed in §3.1.1–§3.1.3. Facility and geospatial databases are separately populated and layer processing is performed. Additional GIS analyses may be performed, after which the geospatial database is updated.

3.2 Practical data considerations

The raster data used in this dissertation are available at various resolutions [52, 54, 132]. Standard raster elevation data resolution levels are presented in denominations of arc seconds across latitude and longitude which, when converted to surface distance, translate to approximately 900 m, 90 m and 30 m spacings between the points on the terrain surface at the equator. Here, the notion of terrain data resolution corresponds to the density of gridposts in the representation — higher resolution data include more gridposts spaced closely together, whereas lower resolution data include fewer gridposts spaced further apart. An example of two different resolution representations for the same 5 km×5 km square section of terrain is provided in Figure 3.3. The gridposts in the high-resolution representation in part (a) of the figure are spaced approximately 28 metres apart and comprise a total of 32 617 gridposts, while the gridposts in the low-resolution representation in part (b) are spaced approximately 340 metres apart and comprise a total of 255 gridposts. It is important to consider which resolutions of the layer data are suitable for inclusion in the geospatial database for the following reasons:

- **Analysis accuracy.** An analysis involving criteria determined with respect to a single gridpost may depend on the surrounding gridposts. For example, slope is determined by considering the differences in height with respect to gridposts surrounding a particular gridpost, using techniques such as interpolation. It is therefore important to perform pre-processing analyses whose results depend heavily on locally surrounding gridposts at the highest available resolution so as to avoid the unnecessary loss of accuracy that results when the distances between gridposts become too large at lower resolutions¹.

¹An example of the importance of high-resolution analysis for the purposes of visibility-related computations may be found in [52].

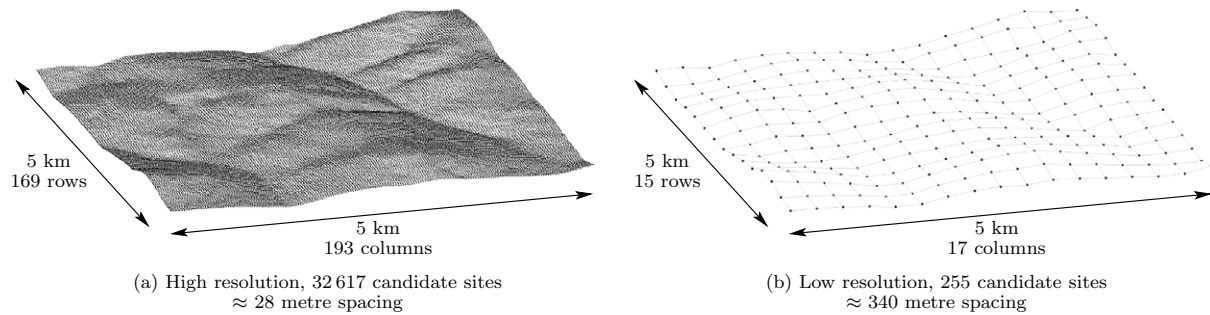


FIGURE 3.3: A $5\text{ km} \times 5\text{ km}$ square section of terrain represented by raster data at two different resolution levels. The high-resolution representation in (a) is associated with a high level of solution quality and accuracy, but involves a larger number of candidate sites, resulting in more time-consuming solution processes. The low-resolution representation in (b) involves a smaller number of candidate sites, resulting in faster solution processes, but is associated with a low level of solution quality and accuracy.

- Number of candidate solutions.** The use of higher resolution data results in a larger number of candidate locations within the SZ. The computational complexity of finding suitable location combinations for placement problems involving multiple types of facilities and multiple objectives may, as a result, be impractically high when the number of candidate locations is large. It may therefore sometimes be advantageous to perform pre-processing analyses for only a lower-resolution subset of candidate gridposts in the SZ, where the analyses themselves nevertheless take into account the highest resolution data available, so as to ensure a higher level of accuracy. Another possibility is to identify likely location regions by clustering good candidate locations at low resolution, after which the likely regions may be considered during a new round of higher-resolution heuristic analysis [54] — such an approach is considered later in this dissertation.
- Data storage requirements.** Location-specific analyses yield results that are simple to present and store — these results each consists of a single value calculated with respect to a certain placement criterion for each point in the SZ, which simply requires the storage of a number of values that is a linear function of the number of candidate locations in the SZ. For inter-site analyses (*i.e.* SZ with respect to IZ), however, the data storage requirements are greater. This is a result of each candidate location in the SZ requiring the storage of values with respect to each point of interest in the interest zone. If there are n_s candidate locations in the SZ and n_i points of interest in the IZ, the number of values that have to be stored is a linear function of $n_s \times n_i$. This requirement may become prohibitive at high resolutions.
- Computation time.** The use of high-resolution data ensures good solution quality and accuracy, but involves a larger number of candidate solutions and associated evaluations than lower-resolution representations, and are therefore more time-consuming [54, 55]. Lower-resolution representations of the data may be extracted from high-resolution data with the aim of reducing the number of candidate solutions and the computational complexity associated with solving the problem — ultimately resulting in shorter computation times. Shorter computation times, however, typically come at a loss of solution quality due to the loss of good candidate sites from the higher resolution data. A trade-off between solution quality and computation time requirements must therefore be achieved when choosing an appropriate resolution of terrain data to use in geospatial facility location models [54].

- Practical implementation.** The practical implementation of solutions should also be taken into account. Due to the discrete nature of the raster data used, a solution provides facility locations at very precise coordinates on the terrain surface. The actual terrain at the proposed sites may, however, not be perfectly suited for facility placement due to factors such as surface shape or soil type, and as a result the facility may preferably be placed some distance from the precise location instead (within a distance that keeps it closer to the proposed gridpost combination than to any other gridpost combination). As the resolution of the data decreases, the surface areas between gridposts increase and as a result the solutions embody a greater degree of uncertainty because of the larger distances over which placement combinations can be perturbed away from a gridpost combination before entering another combination's vicinity. Proposed solution sites should therefore always be analysed and presented at the highest possible resolution in order to obtain the best results with respect to the placement objectives, as long as this is within practical implementation reach. A good study on practical implementation *versus* 'optimal' solutions was conducted in [94].

3.3 Classification of GFLo criteria

Two main classes of criteria are identified for inclusion in the GFLo framework proposed in this dissertation: (a) *location-specific* criteria (§3.3.1) and (b) *inter-site* criteria (§3.3.2). Two sub-classes, *viz.* *geospatial* criteria and *facility-dependent* criteria, are also considered within each of these classes, as illustrated in Figure 3.4.

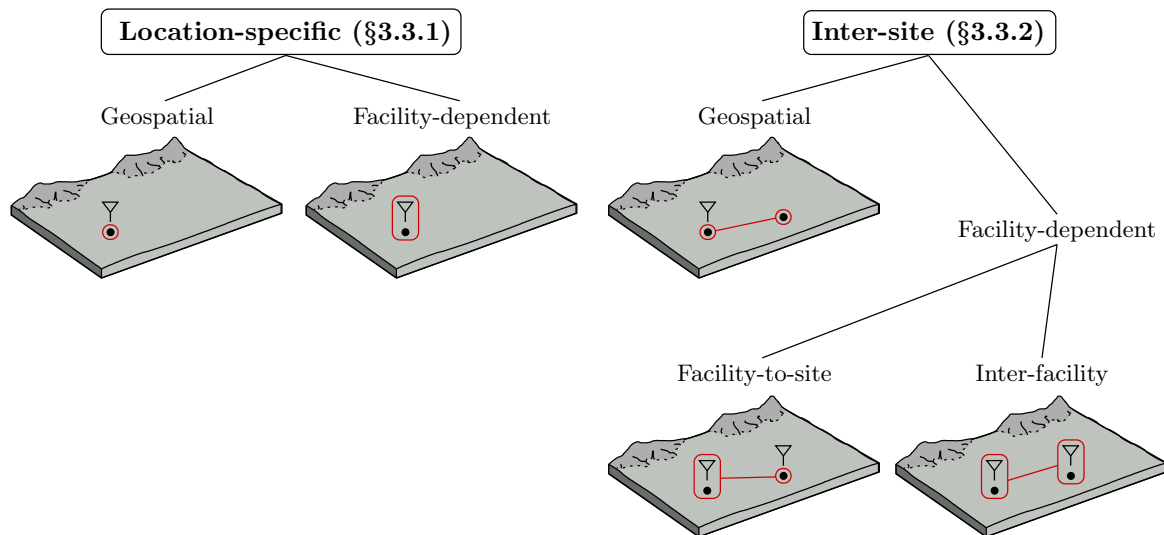


FIGURE 3.4: The classification of GFLo criteria types that may be considered in a generic GFLo framework.

3.3.1 Location-specific criteria

Location-specific criteria are those that are evaluated with consideration given to the candidate location at which the facility is placed — with no consideration of the influence that the placement may have with respect to any other gridpost or facility included in the GFLoP. Examples of geospatial location-specific criteria include elevation, slope, distance from a specific feature,

and average annual wind speed/direction. Location-specific criteria, such as average daily solar energy and wind speeds, may undergo additional computation in order to determine facility-dependent values related to the criteria at the specific site. For example, two different types of wind turbines may experience the same average daily wind energy at the same location, yet differ in their average daily wind energy generation potential due to factors such as the rotor size and other technical specifications.

Solar exposure and slope are important to consider in the planning of solar farms. Comprehensive solar farm planning requires the analysis of the slope of the terrain to determine areas that are easily accessible and suitable for photovoltaic panel installation. Areas with low slope angles are desirable as potential solar farm locations since they typically lead to easier access and cheaper installation costs [112, 133]. Fu *et al.* [39], for example, considered the effect of solar radiation on agriculture and forestry applications. While slope is not explicitly used as a criterion for agricultural planning in [39], the importance of slope in agricultural and forestry applications is nevertheless acknowledged. Despite the absence of these criteria in the solution of point-based GFLoPs in the literature, they are nevertheless considered for inclusion in the GFLo framework proposed in this dissertation. The most important criterion in the placement of wind turbines is the average wind speed on the terrain surface. This may be determined using historical weather station data or wind flow grid simulations [114, 120]. In this type of analysis, candidate sites for wind turbines are found with respect to the predicted wind speed (and possibly also wind direction, depending on the type of turbines). A recent development in the renewable energy domain is the development of *hybrid power systems*, which combine solar photovoltaic panels and wind turbines into a single power system [32]. These power systems may be attached to facilities such as watchtowers — the placement of the watchtower may then be determined according to solar and wind energy potential criteria, in addition to other criteria such as visibility.

Figure 3.5 contains examples of two sets of location-specific criteria values for a terrain area that has been extracted for facility location purposes. Location-specific criteria values are shown for feasible site locations with respect to (a) average daily solar irradiation, and (b) proximity from rivers — these values may be used to evaluate candidate facility placements. The small gaps in the values that are visible towards the top left of Figures 3.5 (a) and (b) are infeasible placement areas that were removed due to slope in excess of ten degrees. The solar irradiation values may be further analysed in order to determine average solar energy generation potential for facilities such as solar energy systems with different energy generation capabilities. These facility-dependent criteria values may then be used for the evaluation of the placement of facilities equipped with these devices.

3.3.2 Inter-site criteria

Inter-site placement criteria refer to criteria that are determined between a candidate facility location and another site which may be located in a SZ or IZ. In the geospatial sub-class, criteria such as distance, accessibility and difference in elevation/slope/temperature may be evaluated between two sites. The facility-dependent inter-site sub-class may further be divided into two categories. In *facility-to-site* criteria, consideration is given to the facility placed at the location from which the criterion is evaluated. In the event that a facility is located at the site with respect to which the criterion is evaluated, the interaction between facilities is not taken into account. In this sub-class, the criterion is typically destined for inclusion in covering problems (previously discussed in §2.2.2), involving criteria such as visibility/transmission coverage and noise energy propagation, or to evaluate the influence of source facilities of a specific type on other facilities in the network. In the *inter-facility* sub-class, consideration is given to the *interaction* between

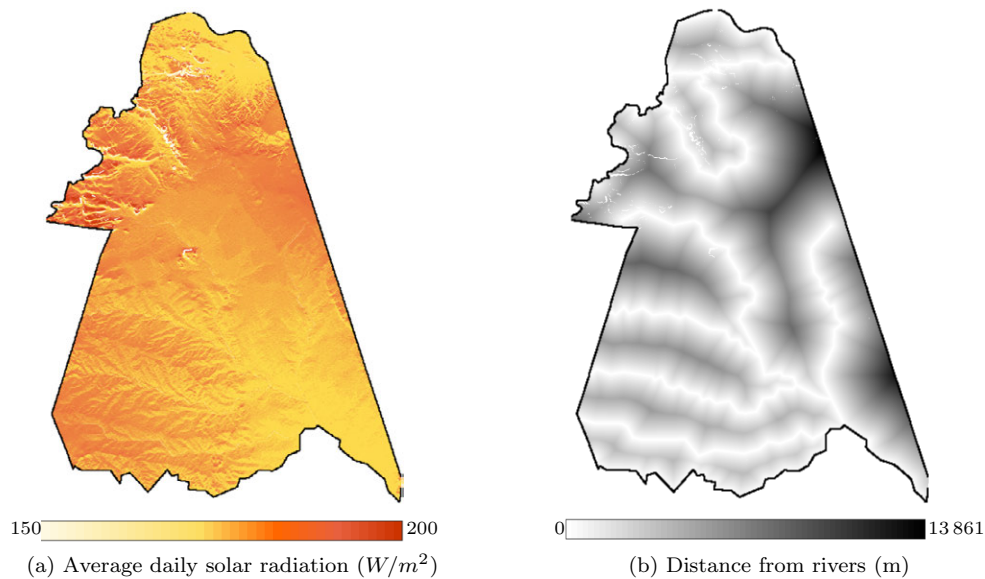


FIGURE 3.5: Location-specific criteria values for feasible site locations extracted from a terrain area for facility location purposes. Average daily solar irradiation values are indicated in (a), while proximity from rivers is indicated in (b).

two facilities placed at two candidate locations— these facilities may not necessarily be of the same type. Examples of inter-facility criteria may include inter-connectivity, inter-visibility, and inter-defence criteria. The notion of the prefix ‘inter’ in this context refers to the mutual ability of the facilities to ‘serve’ each other. For example, if an inter-defence criterion is evaluated between two weapon systems, the pair may achieve a score of zero if neither facility is able to defend the other. A score of one may be achieved if only one facility is able to defend the other, while a score of two may be achieved if each facility is able to defend the other. Binary value scoring may also be implemented. A score of zero may, for example, be assigned if the facilities cannot defend each other mutually, while a score of one may be assigned if each facility is able to defend the other. Values for inter-visibility and inter-connectivity criteria may be assigned in a similar manner. Due to the complexity involved with including inter-facility criteria in a GFLoP, this class of criteria is included in the mathematical formulation of GFLoPs performed in the following section, but is not considered in the practical development process and testing of worked examples.

A significant portion of facility-to-site criteria require LOS analyses and are visibility-related [7, 53, 66]. Visibility-related facility placement analyses take the form of multiple LOS inter-visibility computations. A single LOS query determines whether or not a *target* point on the terrain surface is visible from a source point on the terrain surface, referred to as the *observer* in [26, 120]. Visibility-related analyses in the literature generally include *viewshed* computations [7, 9, 24, 53, 66, 95, 124]. A *viewshed* is the portion of terrain that is visible from an observer gridpost, limited by terrain interference. Here the notion of visibility may have multiple interpretations, such as optical visibility, or transmitter tower or radar detection capability [26]. The determination of viewsheds simply entails a collection of LOS queries calculated from an observer gridpost with respect to all (or some significant portion) of the target gridposts within an IZ — within the visible or detection range of the facility in question. Figure 3.6 contains an example of what a viewshed may look like for an observer gridpost, within a 50 kilometre radius around it. The darker shaded areas within the circle are portions of the earth’s surface that are invisible to the observer.

The simplest and most commonly used algorithm for LOS queries is that of Bresenham [10], developed in 1965, although numerous techniques for decreasing the computation times of LOS queries have since been proposed [23, 24, 30]. LOS-related queries are generally performed with respect to two data structure types for representing the earth's terrain surface: raster data or *triangulated irregular networks* (TINs). TINs are derived from raster data by identifying important gridposts from the raster data model with respect to terrain relief characteristics, while other, lesser important gridposts are disregarded. The remaining important gridposts are then joined to each other by straight lines, creating triangular planes which approximate the terrain surface. Raster data and TINs have been compared extensively with respect to their effectiveness and accuracy [23, 24, 79, 117]. These comparisons indicate that no clear preferred data structure type is evident and, since the GFLo framework proposed in this dissertation is based on the use of raster data, this data type is therefore used in the determination of LOS and related analyses. Kim *et al.* [66] have provided a comprehensive list of viewshed applications and related literature, including the location of telecommunication relay towers and wind turbines, the protection of endangered species, the analysis of archaeological sites, urban planning and optimal path route planning. Numerous viewshed estimation procedures have been suggested for reducing overall computation times, such as those by Ben-Moshe [9], Heyns and van Vuuren [52] and Tabik *et al.* [124].

A *soundshed* is similar to the notion of a viewshed; it is the portion of terrain experiencing noise from a noise source, which is limited by terrain interference and acoustic directivity [26]. Soundshed analysis is important in the evaluation of noise pollution caused by existing or proposed facilities, such as airports and high-speed road links [64, 71, 108]. In the context of soundsheds determined with respect to discrete site locations, the placement of wind turbines may be evaluated according to noise pollution levels predicted for a specific configuration [75].

A comparison of figures from the literature that illustrate viewsheds [7, 52, 66], soundsheds [75, 108] and broadcast coverage [113] reveal striking similarities between the different applications. The data structures used in order to represent these criteria in cover problems (described in §2.2.2) may be developed in a generic manner: for each site from which the analysis is performed, criteria-related values are determined with respect to a set of points of interest included in IZs. The same gridded data structures in which to store the results when computed within a raster environment may be used for all the above-mentioned applications. As a result, the mathematical formulation of GFLoPs related to these criteria types may be achieved in a generic manner, as demonstrated in the following section.

3.4 Mathematical modelling of the GFLo environment

The basic mathematical infrastructure necessary for the formulation of GFLoP models is provided in this section. The infrastructure includes the modelling of SZs which comprise candidate sites for facility placement and IZs which comprise interest points that are influenced by facilities placed in the SZs, as discussed in §3.4.1. The establishment of the mathematical notation related to the available facilities and their placement in selected SZs follows in §3.4.2, and the mathematical modelling of the GFLo criteria classes identified in §3.3 is finally performed in §3.4.3.

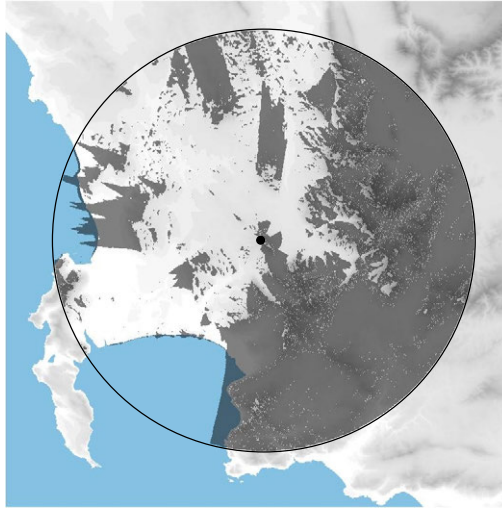


FIGURE 3.6: An example of a viewshed analysis with respect to an observer gridpost (the black dot) on the earth's surface within a specified radius of 50 kilometres. The darker shaded areas represent earth surface portions that are invisible to the observer as a result of terrain obstruction.

3.4.1 Modelling search and interest zones

SZs and IZs are modelled in this dissertation with respect to a *reference grid* (RG). The RG is a grid which envelops the entire terrain surface area considered in a GFLoP and, for simplicity, this grid is modelled as a rectangular area, with sides parallel with lines of latitude and longitude. The purpose of the RG is to provide a global referencing scheme that allows for a simple gridpost numbering convention aimed at simple gridpost identification. As a result of the rectangular modelling approach, the RG may encapsulate gridposts that are not considered during a GFLoP. The RG is governed by the following variables:

- The *global reference point*, denoted by \mathbf{G}_p , represents the latitude north and longitude east of the southwesternmost gridpost in the RG, denoted by G_n and G_e , respectively. It follows that $\mathbf{G}_p = (G_n, G_e)$, with $G_n, G_e \in \mathbb{R}$.
- The number of rows and columns in the grid are denoted by G_r and G_c , respectively.
- The resolution of the grid (*i.e.* the distance between gridposts along latitude and longitude) is measured in arc seconds (degrees), denoted by G_a .

The RG itself is denoted by $\mathbf{G}(\mathbf{G}_p, G_r, G_c, G_a)$. Furthermore, denote the number of gridposts in \mathbb{G} by $n_g = G_r \times G_c$. Then the set of gridposts included in the RG is denoted by $\mathbb{G} = \{1, \dots, n_g\}$. The gridposts are numbered according to the rows and columns in \mathbb{G} . The gridpost at the reference point is numbered 1 and the gridposts increase in number, northward along the rows of the reference point's column until the final row is reached, after which the numbering continues from the southernmost row in the next column.

SZs are areas within which optimal facility siting is pursued — the gridposts enveloped by the designated SZ boundaries are candidate site locations that may be considered for facility placement. SZs are modelled in such a manner that a specific GFLoP may include multiple SZs comprising overlapping sections (*i.e.* mutual gridposts) and a SZ may be considered for the placement of multiple facility types. Denote the number of SZs in a GFLoP by n_s and denote the set of gridposts considered as candidate locations in SZ $s \in \{1, \dots, n_s\}$ by \mathbb{S}_s , where $\mathbb{S}_s \subseteq \mathbb{G}$.

When solving covering problems, candidate locations in one or more SZs are evaluated with respect to their influence on point of interest gridposts enveloped within specified IZs. In the same manner as for SZs, IZs are modelled so that a specific GFLoP may include multiple IZs which may have overlapping sections. Denote the number of IZs in a GFLoP by n_i and denote the set of interest points that are included in IZ $i \in \{1, \dots, n_i\}$ by \mathbb{I}_i , where $\mathbb{I}_i \subseteq \mathbb{G}$.

The specification of an RG, SZs and IZs for a hypothetical GFLoP is illustrated by means of a small example.

Example 3.1 (GFLo zone modelling) *Suppose that the global reference point for the RG in a GFLoP is specified as $\mathbf{G}_p = (G_n^*, G_e^*)$, and that the RG is defined as $\mathbf{G}(\mathbf{G}_p, 8, 9, G_a^*)$. The resulting RG is shown in Figure 3.7 (a).*

Suppose the SZs

$$\begin{aligned} \mathbb{S}_1 &= \{22, 23, 24, 30, 31, 32, 39, 40\}, \\ \mathbb{S}_2 &= \{1, 2, 3, 4, 9, 10, 17, 18, 25\}, \\ \mathbb{S}_3 &= \{55, 56, 62, 63, 64, 70, 71, 72\}, \text{ and} \\ \mathbb{S}_4 &= \mathbb{S}_2 \cup \mathbb{S}_3 = \{1, 2, 3, 4, 9, 10, 17, 18, 25, 55, 56, 62, 63, 64, 70, 71, 72\} \end{aligned}$$

are specified, and suppose the IZs

$$\begin{aligned} \mathbb{I}_1 &= \{20, 27, 28, 35, 36, 43\}, \\ \mathbb{I}_2 &= \{36, 44, 45, 51, 52, 53, 59, 60, 61\}, \text{ and} \\ \mathbb{I}_3 &= \mathbb{I}_1 \cup \mathbb{I}_2 = \{20, 27, 28, 35, 36, 43, 44, 45, 51, 52, 53, 59, 60, 61\} \end{aligned}$$

are specified. These zones are illustrated graphically in Figure 3.7. ■

3.4.2 Modelling facilities

The assignment of facilities destined for placement within specified SZs and the associated facility placement decision variables are modelled in this section and demonstrated by means of an example.

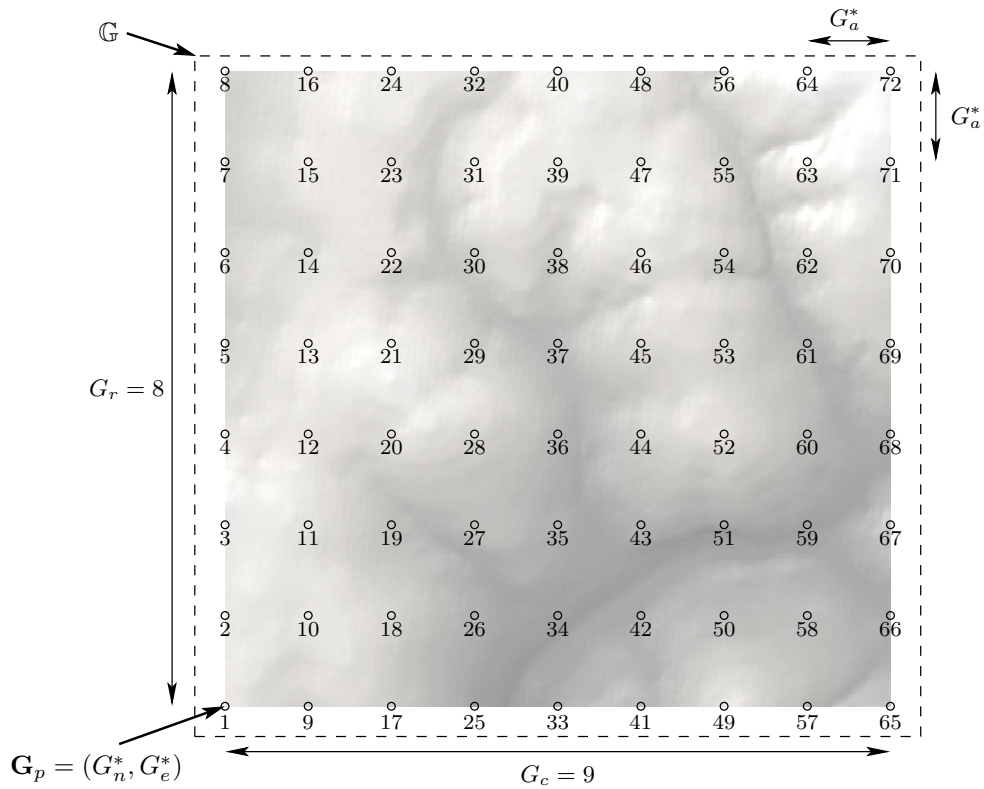
Suppose \mathbb{F} denotes the set of all facility types to be included in a GFLoP, of cardinality $|\mathbb{F}| = n_F$. Denote the number of facilities of type $f \in \mathbb{F}$ available for placement by n_f , and let the number of facilities of type f to be placed in \mathbb{S}_s be denoted by $n_{f,s}$. The values of $n_{f,s}$ may be stored in an $n_s \times n_F$ facility-SZ assignment matrix,

$$\mathbf{F} = \begin{bmatrix} n_{1,1} & n_{2,1} & \dots & n_{n_F,1} \\ n_{1,2} & n_{2,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ n_{1,n_s} & \dots & \dots & n_{n_F,n_s} \end{bmatrix}. \quad (3.1)$$

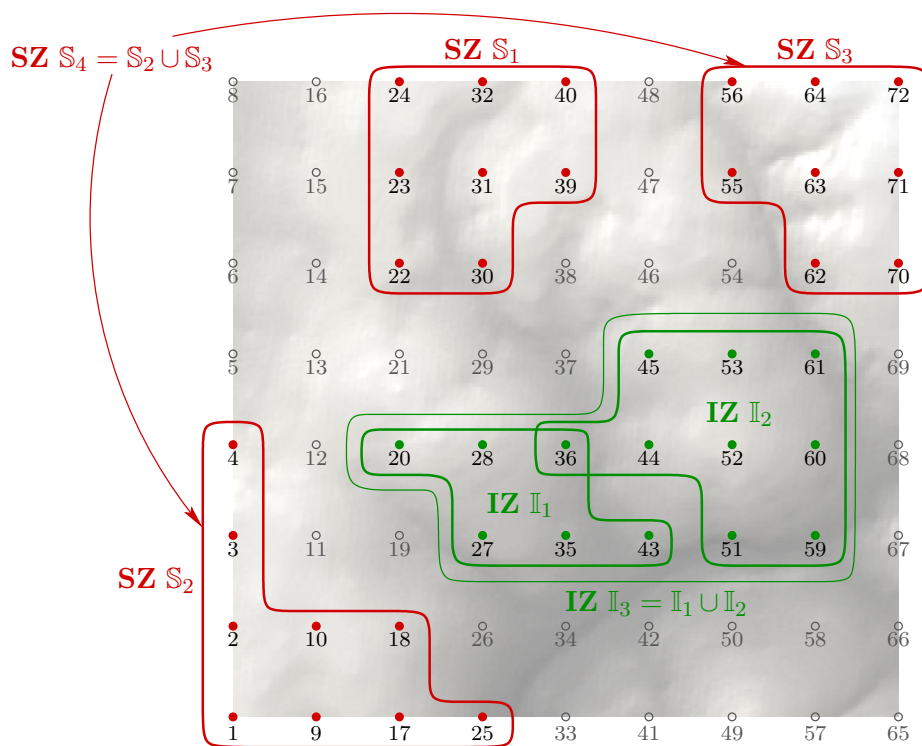
Facility placement

Denote the collection of candidate locations in all SZs by

$$\hat{\mathbb{S}} = \bigcup_{s=1}^{n_s} \mathbb{S}_s. \quad (3.2)$$



(a) Reference grid



(b) Search and interest zones

FIGURE 3.7: (a) RG , and (b) SZs as well as IZs considered in the hypothetical Example 3.1.

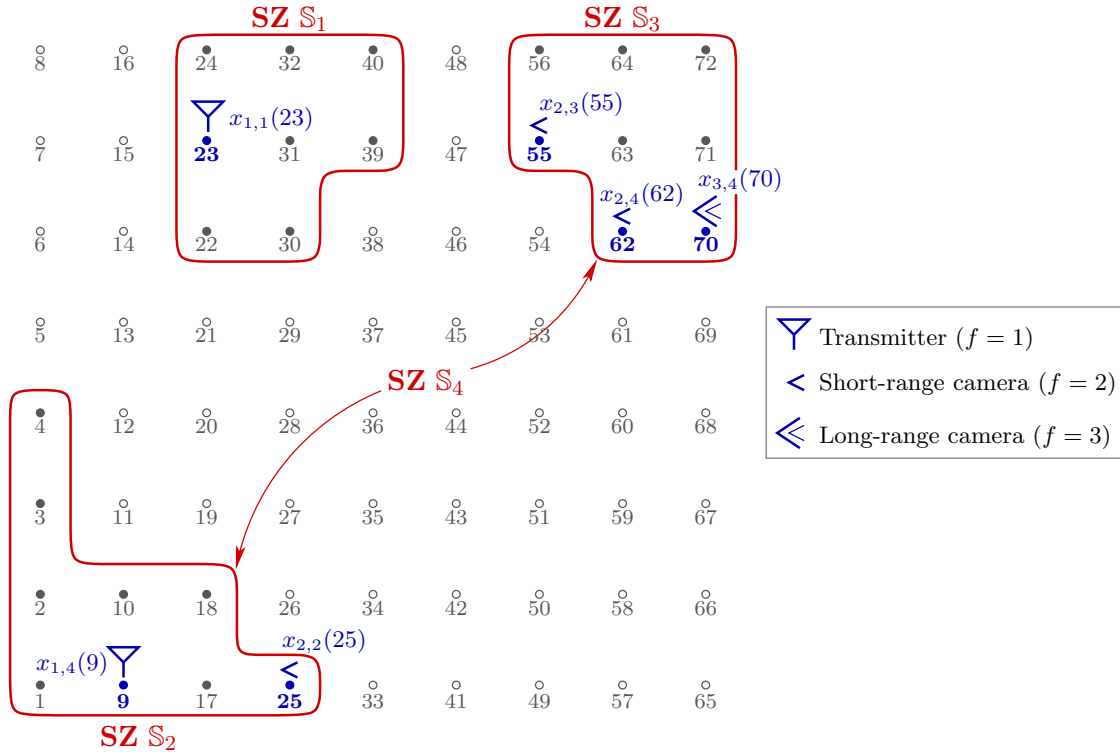


FIGURE 3.8: Placement of facilities in Example 3.2.

Define the facility location decision variable

$$x_{f,s}(\ell) = \begin{cases} 1, & \text{if a facility of type } f \in \mathbb{F} \text{ is placed at location } \ell \in \mathbb{S}_s, \\ 0, & \text{if no facility of type } f \in \mathbb{F} \text{ is placed at location } \ell \in \mathbb{S}_s, \\ & \text{or if } \ell \in \hat{\mathbb{S}}, \ell \notin \mathbb{S}_s. \end{cases} \quad (3.3)$$

Example 3.2 (GFLo facility modelling) Consider the RG and zones introduced in Example 3.1 together with the following facility types:

$$\mathbb{F} = \{1[\text{Transmitter}], 2[\text{Short-range camera}], 3[\text{Long-range camera}]\}.$$

Furthermore, suppose

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

An example of a feasible facility location layout according to \mathbf{F} is illustrated in Figure 3.8. The decision variables $x_{f,s}(\ell)$ that are equal to 1, indicating placement of each specific facility type at locations within specific SZs, are also illustrated in the figure. ■

Facility placement constraints

On closer inspection of Figure 3.8 in Example 3.2, some important observations may be made in respect of the placement decision variables related to the placement of the short-range cameras

($f = 2$). Since SZs \mathbb{S}_3 and \mathbb{S}_4 overlap, and since both SZs require the placement of one short-range camera, it is possible that both short-range cameras may be placed at candidate locations that are mutual to the SZs. This is the case for the decision variables $x_{2,3}(55)$ and $x_{2,4}(62)$ in the figure.

In such a candidate layout, the SZs to which the facilities belong may be interchanged to form a new facility-SZ assignment representing the same layout which remains feasible in respect of \mathbf{F} . Despite such a ‘swap’ resulting in the same layout, the new assignment may result in the candidate layout achieving different objective function values. For example, suppose the short-range camera in \mathbb{S}_3 is evaluated according to its distance from a certain point. If the ‘swap’ is made with the short-range camera in \mathbb{S}_4 , the layout remains the same, yet a different distance value is obtained for the ‘new’ camera in \mathbb{S}_3 , which may affect the objective function values achieved by the layout when evaluated with respect to the objective functions. The decision in respect of the SZ to which a facility should be assigned lies with the solution methodology employed to solve the problem. If both short-range cameras are assigned to only one of either SZs, the facility-SZ assignment will not be feasible according to the values in \mathbf{F} , although the facility layout will remain precisely the same.

For the above-mentioned reasons, the constraints

$$\sum_{\ell \in \mathbb{S}_s} \sum_{f=1}^{n_F} x_{f,s}(\ell) = n_{f,s}, \quad (3.4)$$

are imposed for all $s \in \{1, \dots, n_s\}$, in addition to requiring that

$$\sum_{s=1}^{n_s} \sum_{f=1}^{n_F} x_{f,s}(\ell) \leq 1, \quad (3.5)$$

for all $l \in \hat{\mathbb{S}}$, and that

$$x_{f,s}(\ell) \in \{0, 1\}, \quad (3.6)$$

for all $f \in \mathbb{F}$, for all $s \in \mathbb{S}$, and for all $l \in \hat{\mathbb{S}}$.

Constraint set (3.4) ensures that the number of facilities of each type placed in each SZ is equal to the value assigned by the facility-SZ assignment matrix. Constraint set (3.5) ensures that only one facility is placed at any of the candidate locations in $\hat{\mathbb{S}}$, while constraint set (3.6) specifies an integer requirement on $x_{f,s}(\ell)$.

3.4.3 Modelling facility placement criteria

Location-specific and inter-site placement criteria are modelled in this section. Denote the set of criteria considered in a GFLoP by \mathbb{C} . This set includes *all* the criteria considered in the problem, for all facility types in all SZs. A specific criterion, denoted by $c \in \mathbb{C}$, may be selected for evaluation according to the class of criteria to which it belongs.

Location-specific criteria

Location-specific criteria types were elucidated in §3.3.1. Let $v_c(\ell)$ denote the value of geospatial criterion $c \in \mathbb{C}$ at candidate location $\ell \in \hat{\mathbb{S}}$. Furthermore, let $v_{f,c}(\ell)$ denote the value of a facility-dependent criterion $c \in \mathbb{C}$ if a facility of type $f \in \mathbb{F}$ is placed at candidate location $\ell \in \hat{\mathbb{S}}$.

Example 3.3 (Location-specific criteria) Consider again the zones and facility placements in Examples 3.1–3.2. Suppose the location-specific criteria included in the problem are embodied in the set

$$\mathbb{C} = \{1[\text{Power source distance}], 2[\text{Number of } \mathbb{I}_3 \text{ gridposts visible}]\}.$$

Criterion $c = 1$ (a geospatial criterion) may be relevant to transmitters, whereas criterion $c = 2$ (a facility-dependent criterion) may be relevant to surveillance cameras. The resulting location-specific criteria values of $v_c(\ell)$ and $v_{f,c}(\ell)$, measured from the relevant facilities, are illustrated graphically in Figure 3.9. ■

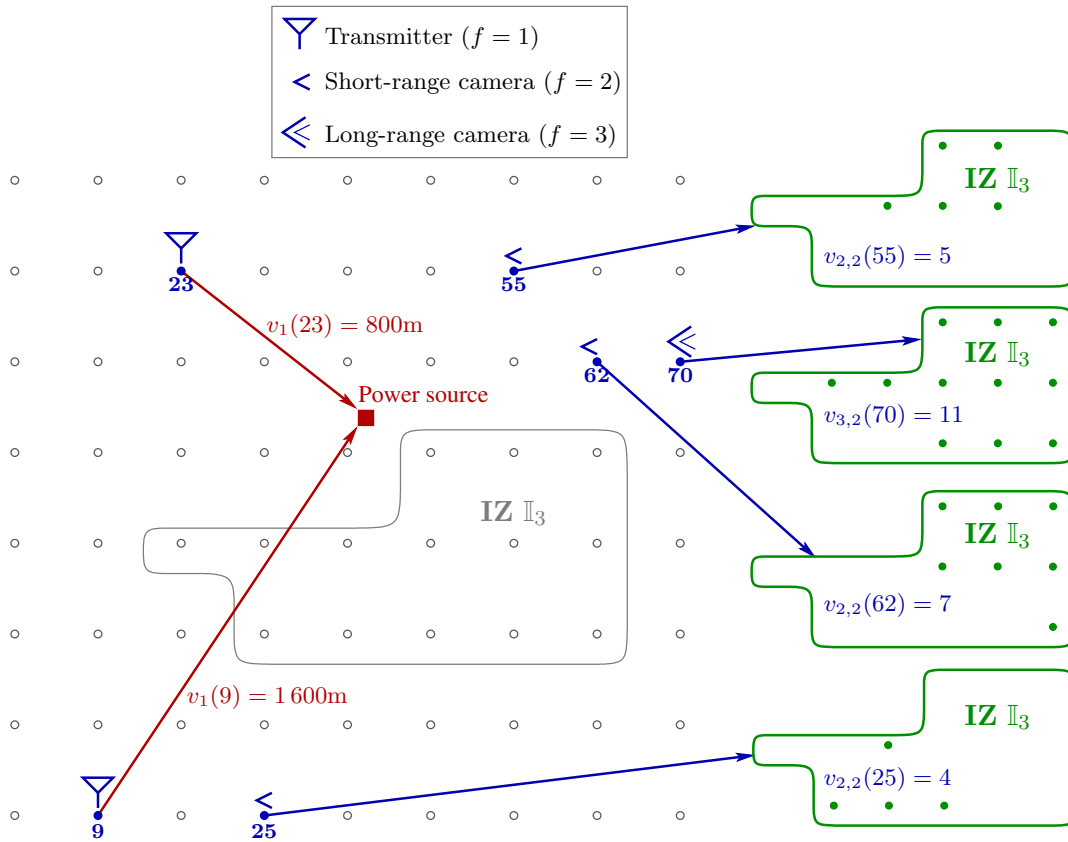


FIGURE 3.9: Location-specific criteria values for the hypothetical Example 3.3.

Inter-site criteria

Inter-site placement criteria types were discussed in §3.3.2. Define

$$\hat{\mathbb{G}} = \left(\bigcup_{s=1}^{n_s} \mathbb{S}_s \right) \cup \left(\bigcup_{i=1}^{n_i} \mathbb{I}_i \right). \quad (3.7)$$

Then the sites in $\hat{\mathbb{G}}$ are all those with respect to which inter-site criteria may be determined from the candidate facility locations.

Let $v_c(\ell, i^*)$ denote the value of a geospatial criterion $c \in \mathbb{C}$, measured from candidate location $\ell \in \hat{\mathbb{S}}$ to point of interest $i^* \in \hat{\mathbb{G}}$. Furthermore, let $v_{f,c}(\ell, i^*)$ denote the value of a geospatial

criterion $c \in \mathbb{C}$, if a facility of type $f \in \mathbb{F}$ is placed at candidate location $\ell \in \hat{\mathbb{S}}$ and evaluated according to its effect on point of interest $i^* \in \hat{\mathbb{G}}$.

The inter-facility performance value — an evaluation of the interaction between two facilities — for a facility of type $f \in \mathbb{F}$ placed at candidate location $\ell \in \hat{\mathbb{S}}$, according to criterion $c \in \mathbb{C}$, with respect to a facility of type $f^* \in \mathbb{F}$ placed at candidate location $\ell^* \in \hat{\mathbb{S}}$ is denoted by $v_{f,c}^{f^*}(\ell, \ell^*)$.

Example 3.4 (Inter-site criteria) Consider again the zones, facility placements and facility placement criteria of Example 3.3. If a geospatial inter-site criterion is added to \mathbb{C} ; then

$$\mathbb{C} = \{1, 2, 3[\text{Distance to long range cameras}]\}.$$

Suppose that criterion $c = 3$ is evaluated from transmitters ($f = 1$) in SZ 2 only. The determination of $v_c(\ell, i^*)$ from the relevant location is illustrated graphically in Figure 3.10.

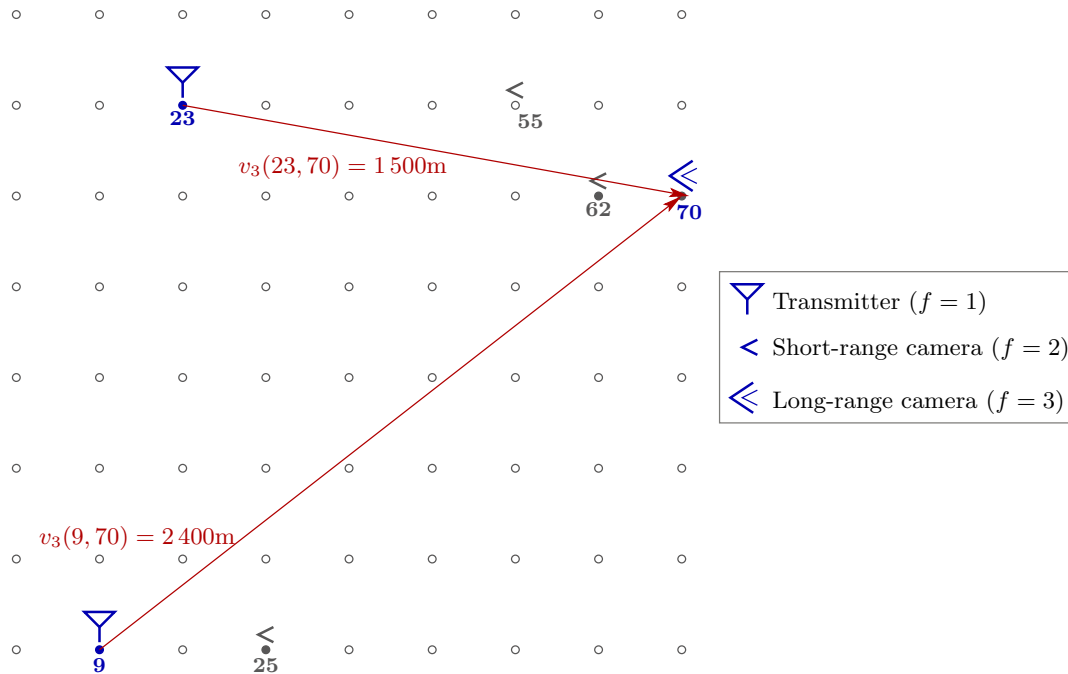


FIGURE 3.10: Geospatial inter-site criteria values for the hypothetical Example 3.4.

Suppose, furthermore, that three facility-to-site criteria are added to \mathbb{C} . Then

$$\mathbb{C} = \{1, 2, 3, 4[\mathbb{I}_3 \text{ gridpost cover}], 5[\mathbb{I}_2 \text{ gridpost visibility}], 6[\text{Transmitter visibility}]\}.$$

Criterion $c = 4$ may be relevant to transmitters ($f = 1$) in SZ 1, while criterion $c = 5$ may be relevant to short-range cameras ($f = 2$) in SZ 3, and criterion $c = 6$ may be relevant to the long-range camera ($f = 3$) in SZ 4. The determination of $v_{f,c}(\ell, i^*)$ from the relevant locations is illustrated graphically in Figure 3.11. Note that the values achieved are not provided, but that only an illustration of the sites between which the values are determined are shown.

Suppose finally that one inter-facility criterion is added to \mathbb{C} . Then

$$\mathbb{C} = \{1, 2, 3, 4, 5, 6, 7[\text{Mutual visibility}]\}.$$

Criterion $c = 7$ may be evaluated with respect to every pair of cameras of any type. Furthermore,

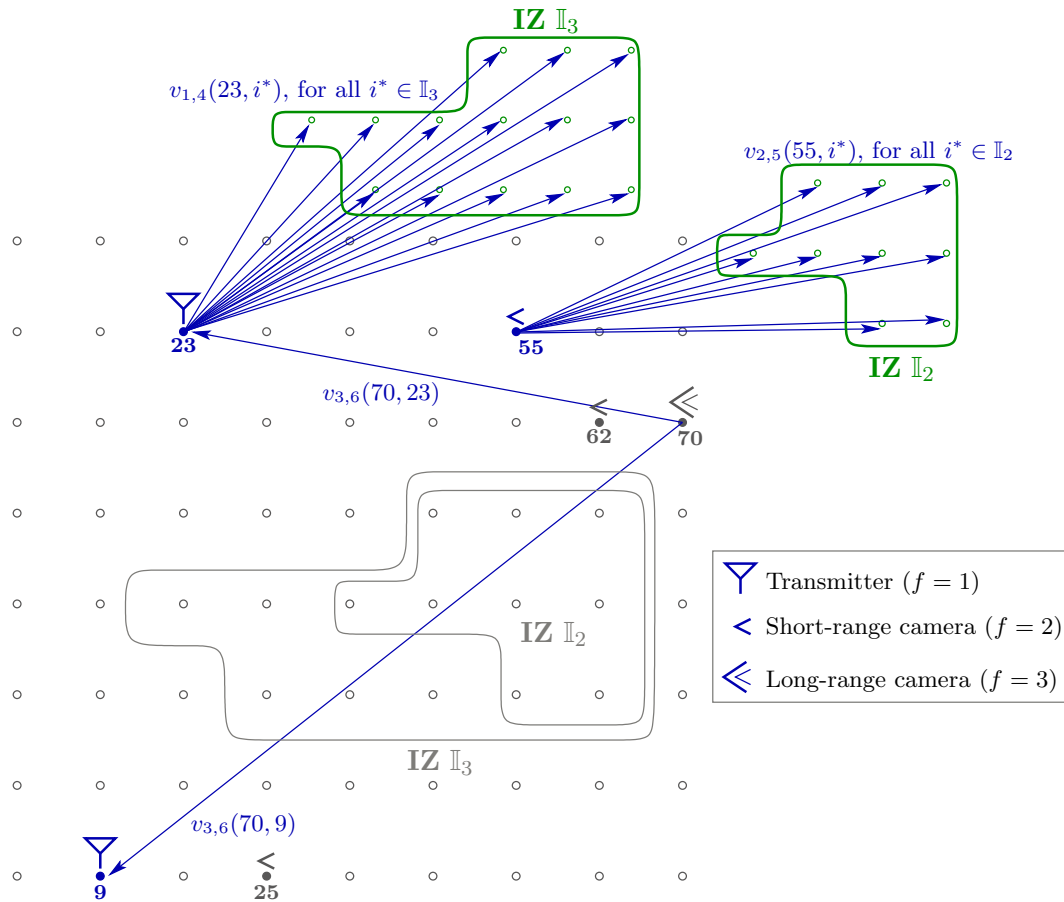


FIGURE 3.11: Facility-to-site inter-site criteria values for the hypothetical Example 3.4.

define

$$v_{f,7}^{f^*}(\ell, \ell^*) = \begin{cases} 2, & \text{if both facilities of type } f, f^* \in \mathbb{F}, \text{ placed at locations } \ell, \ell^* \in \hat{\mathbb{S}}, \\ & \text{are able to see each other,} \\ 1, & \text{if only one of the facilities of type } f, f^* \in \mathbb{F}, \text{ placed at locations } \ell, \ell^* \in \hat{\mathbb{S}}, \\ & \text{is able to see the other,} \\ 0, & \text{otherwise.} \end{cases}$$

Then the determination of $v_{f,7}^{f^*}(\ell, \ell^*)$ with respect to the relevant locations is illustrated in Figure 3.12. ■

3.5 Chapter summary

The structure of a suitable GFLo database was proposed in §3.1. This included descriptions of the separate facility and geospatial databases in §3.1.1–§3.1.2 that make up the GFLo database. This was followed by a description of different types of geospatial analyses in respect of geospatial data in §3.1.3, in partial fulfilment of Dissertation Objective IV, as stated in §1.3.

A discussion on practical data considerations, such as data resolution and analysis accuracy, was conducted in §3.2. The classification of GFLo criteria types into generic classes, according

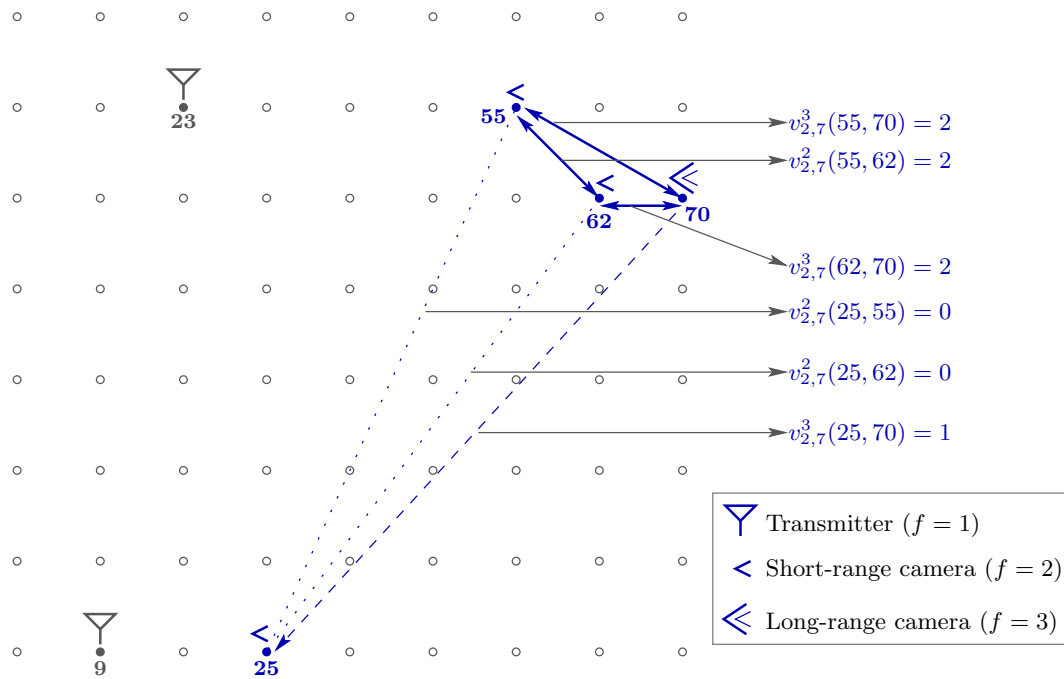


FIGURE 3.12: *Inter-facility inter-site criteria values for the hypothetical Example 3.4.*

to similarities that they exhibit, followed in §3.3, after which an approach to the mathematical modelling of the GFLO environment was proposed in §3.4. This was done in fulfilment of Dissertation Objective II and in partial fulfilment of Dissertation Objective IV, as stated in §1.3.

CHAPTER 4

Generic multi-objective problem formulation framework

Contents

4.1	Preliminary notation	41
4.2	Generic mathematical formulation of objectives	44
4.2.1	<i>MaxiMin and MiniMax models</i>	44
4.2.2	<i>Covering models</i>	45
4.2.3	<i>Dispersion models</i>	47
4.2.4	<i>Centre models</i>	49
4.3	Generic mathematical modelling of additional constraints	49
4.3.1	<i>Location-specific constraints</i>	51
4.3.2	<i>Inter-site constraints</i>	52
4.4	Generic GFLoP formulation process	52
4.5	Chapter summary	53

The objectives and constraints that may be considered in a comprehensive GFLo modelling framework are formulated in this chapter. The formulation of these objectives and constraints is performed in such a manner that different combinations of popular facility location problems from the literature, as well as additional user-specific constraints, may be included in a single GFLoP in a generic manner. The chapter opens with the introduction of the preliminary notation required for the formulation of GFLo objectives and constraints in §4.1, after which objectives and constraints are modelled in §4.2 and §4.3, respectively. The chapter closes with a summary of a generic mathematical GFLo formulation process that may be followed for GFLoPs intended for a wide variety of applications, using the mathematical foundation established in §3 and the formulations pertaining to objectives and constraints provided in this chapter.

4.1 Preliminary notation

As opposed to the traditional notion of formulating objectives and constraints according to facility types — as is generally the standard in the FLo literature — the mathematical models formulated in this dissertation are aimed at solving GFLoPs involving *pairs* of facilities and SZs, so that a specific facility type-SZ pair is viewed as a single entity. In a similar manner that

facilities are evaluated with respect their placement at a specific location or with respect to each other, or with respect to facilities of other types, the specification of sets of facility-SZ pairs is advocated here. In such a scheme, a pair is viewed as an entity, and may be evaluated with respect to its placement at a specific location, or with respect to other specific pairs.

The modelling of GFLoPs according to facility-SZ pairs offers more dynamic problem-solving possibilities. For example, a problem which includes multiple facility types and SZs may include an objective or constraint which requires the evaluation of facilities in specific SZs only or may require the evaluation of specific facility types in selected SZs only. Traditional FLo models do not accommodate this type of placement functionality, although the practicality of such an approach may certainly prove to be useful in future GFLoPs.

In order to facilitate paired facility placement, facility and SZ sets which are associated with each other may be specified. The facilities and SZ sets that are thus paired together are said to form part of a *set-pair*, and are of equal size. Suppose there exist n_p such set-pairs, and let \mathcal{F}_p denote the set of facilities in set-pair $p \in \{1, \dots, n_p\}$ which is paired with a set of SZs, denoted by \mathcal{S}_p . Facilities in \mathcal{F}_p and SZs in \mathcal{S}_p are paired according to their indices in the corresponding sets, so that facility $\mathcal{F}_p(1)$ forms a pair with SZ $\mathcal{S}_p(1)$, and, generally stated, facility $\mathcal{F}_p(i)$ forms a pair with SZ $\mathcal{S}_p(i)$.

Since objectives and constraints are proposed for inclusion in a generic GFLo modelling framework in this dissertation with the aim of providing set-pair functionality, it follows that the decision space is governed by the placement of facility-SZ pair entities. For simplicity and specificity in mathematical notation, set-pair-specific decision variables may be denoted as a special pair-specific instance of the regular facility placement decision variable, $x_{f,s}(\ell)$, as introduced in (3.3).

Let a *set-pair decision variable* be denoted by $\tilde{x}_p(\ell)$, where

$$\tilde{x}_p(\ell) = x_{f_p, s_p}(\ell), \quad (4.1)$$

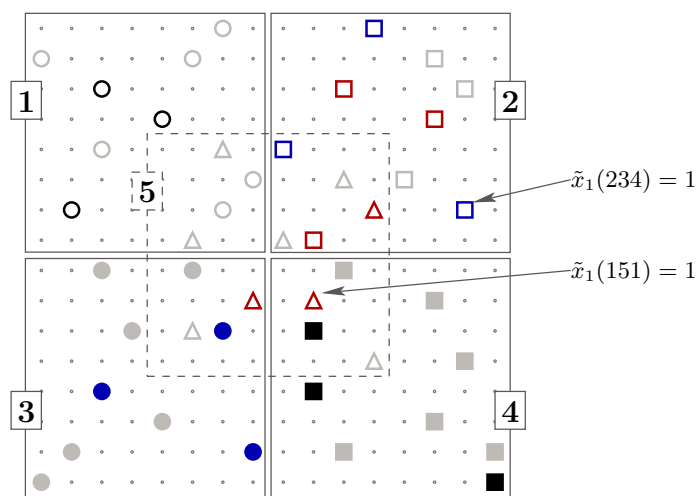
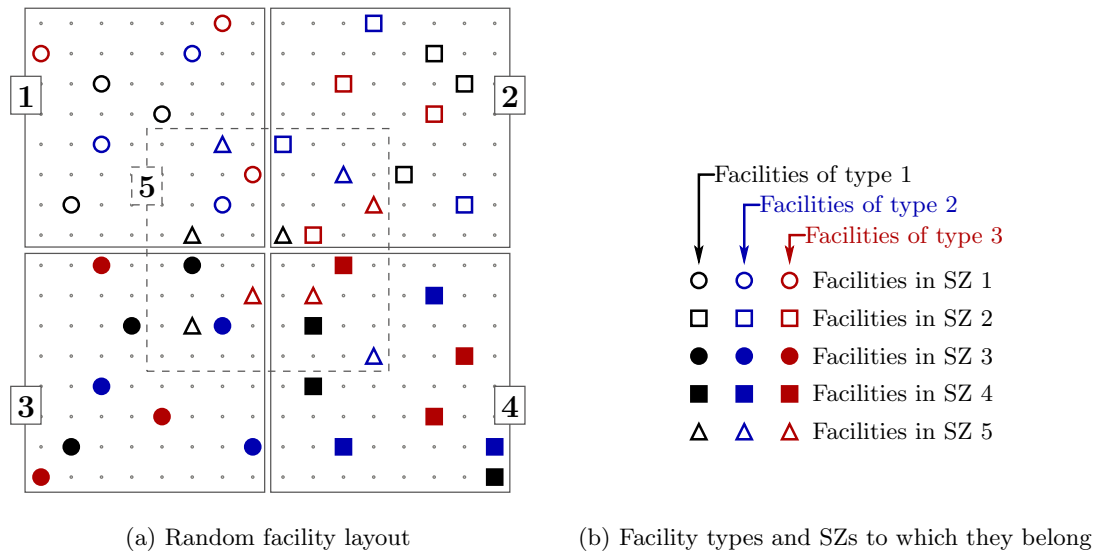
for all ordered facility-SZ pairs $f_p \in \mathcal{F}_p$ and $s_p \in \mathcal{S}_p$, and for all $\ell \in \mathbb{S}_{s_p}$ (\mathbb{S} was introduced in §3.4.1).

In general terms, this decision variable is interpreted as

$$\tilde{x}_p(\ell) = \begin{cases} 1, & \text{if a facility of type } f_p \in \mathcal{F}_p \text{ is placed at location} \\ & \ell \in \mathbb{S}_{s_p} \text{ within search zone } s_p \in \mathcal{S}_p, \\ 0, & \text{otherwise.} \end{cases}$$

Example 4.1 (Set-pairs and set-pair decision variables) *An illustration of a hypothetical GFLoP facility layout with five SZs and three facility types is provided in Figure 4.1 (a), where three instances of each facility type are placed in each SZ. The colours of the facility markers indicate the facility types to which each facility belongs, while the marker shapes and fills indicate to which SZ each facility belongs, as illustrated in Figure 4.1 (b).*

Suppose an ordered set-pair comprises the facility and SZ sets $\mathcal{F}_1 = \{1, 1, 2, 2, 3, 3\}$ and $\mathcal{S}_1 = \{1, 4, 2, 3, 2, 5\}$. In Figure 4.1 (c), the ordered facility-SZ pairs that result from the specified set-pair are illustrated, along with an illustration of the facilities in the search space which are evaluated according to the set-pair (the facilities that are excluded from the set-pair are indicated by grey markers). Also displayed in Figure 4.1 (c) are two examples of set-pair decision variables for two of the facilities that are included in the set-pair. ■



$$\mathcal{F}_1 = \left\{ \begin{array}{c} \circ \\ \blacksquare \\ \square \\ \bullet \\ \color{red}\square \\ \color{red}\triangle \end{array} \right\}$$

$$\mathcal{S}_1 = \left\{ \begin{array}{c} 1 \\ 4 \\ 2 \\ 3 \\ 2 \\ 5 \end{array} \right\}$$

(c) Pair selection

FIGURE 4.1: An illustration of a set-pair and associated set-pair decision variables, as considered in Example 4.1. A random facility layout is shown in (a) and the facility-SZ pairs included in the set pairs specified in the example are shown in (c).

4.2 Generic mathematical formulation of objectives

A number of generic GFLo models are formulated in this section. MaxiMin and MiniMax models are considered in §4.2.1, and this is followed by cover models in §4.2.2. The section closes with the formulation of dispersion and centre models in §4.2.3 and §4.2.4, respectively.

4.2.1 MaxiMin and MiniMax models

MaxiMin and MiniMax models are intended for implementation in the context of location-specific placement criteria, which were formulated in §3.4.3, and involve the location-specific criteria variables $v_c(\ell)$ and $v_{f,c}(\ell)$. These models are formulated for the purposes of maximising minimum values, or minimising maximum values, achieved by the separate facilities included in candidate solutions. These values are measured with respect to placement criteria such as distance from features, solar/wind energy exposure, elevation and slope, amongst many others.

MaxiMin models

The objective in a set-pair MaxiMin problem is to maximise the minimum location-specific placement criterion value V achieved by the ordered facility-SZ pairs in a specific set-pair p , comprising \mathcal{F}_p and \mathcal{S}_p . Formally stated, the objective is to

$$\text{maximise } V, \quad (4.2)$$

subject to the constraints

$$V \leq \begin{cases} v_c(\ell) + M(1 - \tilde{x}_p(\ell)) & \text{or} \\ v_{f_p,c}(\ell) + M(1 - \tilde{x}_p(\ell)), \end{cases} \quad (4.3a)$$

$$(4.3b)$$

for all ordered pairs of $f_p \in \mathcal{F}_p$ and $s_p \in \mathcal{S}_p$, and $\ell \in \mathbb{S}_{s_p}$, for geospatial (4.3a) or facility-dependent (4.3b) criteria, and also subject to constraints (3.4)–(3.6). In (4.3a)–(4.3b) M is a very large positive integer.

Constraints (4.3a)–(4.3b) force the value of the objective function V to be less than or equal to the minimum of the criteria values determined over all facility-SZ pair placements, depending on the type of MaxiMin problem solved (geospatial or facility-dependent). If the facility location evaluated by a constraint does not contain a facility-SZ pair in the set-pair under consideration, then the objective function value V has the simple requirement of being less than or equal to a very large number. In contrast, when the potential facility site under consideration is assigned a facility-SZ pair in the set-pair under consideration, V is constrained from above only by the criterion value.

MiniMax models

The objective in a set-pair MiniMax model is to minimise the maximum location-specific criterion value achieved by all ordered facility-SZ pairs in \mathcal{F}_p and \mathcal{S}_p . Therefore, the objective is to

$$\text{minimise } V' \quad (4.4)$$

subject to:

$$V' \geq \begin{cases} v_c(\ell)\tilde{x}_p(\ell) & \text{or} \\ v_{f_p,c}(\ell)\tilde{x}_p(\ell), \end{cases} \quad (4.5a)$$

$$(4.5b)$$

for all ordered pairs of $f_p \in \mathcal{F}_p$ and $s_p \in \mathcal{S}_p$, and for all $\ell \in \mathbb{S}_{s_p}$, for geospatial (4.5a) or facility-dependent (4.5b) criteria, and subject to constraints (3.4)–(3.6).

The constraints (4.5a)–(4.5b) force the value of the objective function V' to be greater than or equal to the maximum of the criterion values determined over all facility-SZ pair placements. If the facility location evaluated by a constraint does not contain a facility-SZ pair in the set-pair under consideration, then the objective function value V' has the simple requirement of being greater than or equal to zero. In contrast, when the potential facility site under consideration is assigned a facility-SZ pair in the set-pair under consideration, V is constrained only by the criterion value.

4.2.2 Covering models

In a traditional sense, covering problems aim to locate facilities in a such a manner that customers are optimally served according to an objective function typically related to the distance, time, or cost of travel between the facilities and the customers. These problems are modified for application in a GFLo context so that the customers are points on the terrain surface that require ‘service’ from the facilities in terms of certain coverage criteria involving LOS-analyses, such as surveillance and transmission coverage. Covering problems and their applications in a GFLo environment were considered in more detail in §2.2.2. Traditional covering models only involve placement of a single facility type in a single SZ. The models in this section, however, are formulated for implementation in the context of set-pairs.

The maximal covering location problem

The aim of the MCLP is to maximise coverage of service demand at points in the IZ, given a *fixed* number of facilities that are available for placement. The objective is to maximise coverage achieved by ordered facility-SZ pairs of $f_p \in \mathcal{F}_p$ and $s_p \in \mathcal{S}_p$, with respect to a selected IZ.

The mathematical formulation of the MCLP formulated for the purposes of this dissertation is based upon that presented by Church *et al.* [12]. Define

$$y_p(i^*) = \begin{cases} 1, & \text{if point of interest } i^* \in \mathbb{I}_i \text{ receives service from at least} \\ & \text{one facility-SZ pair in set-pair } p \text{ from which cover is demanded,} \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, the facility-dependent inter-site criteria value (§3.4.3) takes the following values

$$v_{f_p,c}(\ell, i^*) = \begin{cases} 1, & \text{if point of interest } i^* \in \mathbb{I}_i \text{ is serviced from a} \\ & \text{facility-SZ pair of } f_p \in \mathcal{F}_p \text{ and } s_p \in \mathcal{S}_p, \text{ and if } \ell \in \mathbb{S}_{s_p}, \\ 0, & \text{otherwise.} \end{cases}$$

For the formulation of the objective functions, sets that specify which locations provide cover to the points in \mathbb{I}_i are formulated. Let these sets be denoted by

$$\mathbb{N}_p(i^*) = \{\ell \mid v_{f_p,c}(\ell, i^*) = 1\}, \quad \ell \in \mathbb{S}_{s_p}, \quad (4.6a)$$

for all ordered pairs of $f_p \in \mathcal{F}_p$ and $s_p \in \mathcal{S}_p$. The objective is then to

$$\text{maximise } V = \sum_{i^* \in \mathbb{I}_i} y_p(i^*) \quad (4.7)$$

subject to the constraints

$$\sum_{\ell \in \mathbb{N}_p(i^*)} \tilde{x}_p(\ell) - y_p(i^*) \geq 0, \quad (4.8)$$

$$y_p(i^*) \in \{0, 1\}, \quad (4.9)$$

for all $i^* \in \mathbb{I}_i$ and subject to the constraints (3.4)–(3.6).

The objective in (4.7) is to maximise coverage. Constraint set (4.8) allows a point of interest i^* to be covered ($y_p(i^*) = 1$) only if one or more facilities are sited at locations in the set $\mathbb{N}_p(i^*)$. Constraint set (4.9) specifies a binary requirement on the auxiliary variable.

The backup covering location problem

The BCLP is an extension to the MCLP and, as a second objective, ensures coverage overlap for failsafe purposes. The objective is to maximise primary and overlapping coverage achieved by ordered pairs of $f_p \in \mathcal{F}_p$ and $s_p \in \mathcal{S}_p$, with respect to a selected IZ.

The mathematical formulation of the bi-objective BCLP, as suggested by Hogan and ReVelle [56], is followed here¹, although it should be noted that a weighted-sum SO implementation is also suggested [56], of which variations are implemented in certain problems [88, 93, 135]. Denote a new auxiliary variable, in addition to $y_p(i^*)$ of the MCLP, as

$$u_p(i^*) = \begin{cases} 1, & \text{if point of interest } i^* \in \mathbb{I}_i \text{ receives service from at least} \\ & \text{2 facility-SZ pairs in set-pair } p \text{ from which cover is demanded,} \\ 0, & \text{otherwise.} \end{cases}$$

The set $\mathbb{N}_p(i^*)$ from the MCLP is also determined for the BCLP, in the same manner. The objectives are then to

$$\text{maximise } V_1 = \sum_{i^* \in \mathbb{I}_i} y_p(i^*) \quad (4.10)$$

and

$$\text{maximise } V_2 = \sum_{i^* \in \mathbb{I}_i} u_p(i^*) \quad (4.11)$$

subject to the constraints

$$\sum_{\ell \in \mathbb{N}_p(i^*)} \tilde{x}_p(\ell) - y_p(i^*) - u_p(i^*) \geq 0, \quad (4.12)$$

$$u_p(i^*) - y_p(i^*) \leq 0, \quad (4.13)$$

$$y_p(i^*), u_p(i^*) \in \{0, 1\}, \quad (4.14)$$

for all $i^* \in \mathbb{I}_i$, in addition to constraints (3.4)–(3.6).

The first objective in (4.10) maximises primary coverage, while the second objective in (4.11) maximises overlapping coverage. Constraint sets (4.12) and (4.13) determine which points of interest have overlapping coverage. Constraint set (4.13) ensures that overlapping coverage is accounted for only after primary coverage has been received, and constraint set (4.14) specifies integer requirements on the auxiliary variables.

¹The formulation ‘BACOP2’ in [56] is the more generally used formulation, and is followed here.

4.2.3 Dispersion models

Facility dispersion problems traditionally involve maximising the separation distance between facilities and were introduced in §2.2.1. The models presented here, however, are reformulated for the dispersion of geospatial and inter-site criteria so that novel implementations, such as maximising the minimum number of certain facility types that are visible from another type (facility-to-site) or maximising the minimum inter-facility communication strength (inter-facility). Dispersion models are generally solved considering a single facility type and a single SZ — the multi-facility type model by Curtin *et al.* [16] is the only known exception. Dispersion models are formulated here to consider dispersion between facility-SZ pairs in separate set-pairs, using the multi-facility model in [16] as a guideline.

The objective in the set-pair dispersion model is to maximise the minimum criterion value determined between facility-SZ pairs in two separate set-pairs, p and p^* . The dispersion models formulated here are elucidated by an example.

Example 4.2 (Set-paired dispersion) *Consider the objective-specific sets for facility types and SZs and the pre-optimisation facility layout introduced in Example 4.1, presented again in Figures 4.2 (a) and (b). Suppose the following sets are additionally considered: $\mathcal{F}_2 = \{1, 2, 3\}$ and $\mathcal{S}_2 = \{3, 1, 4\}$.*

In Figure 4.2 (c) the facilities of $p = 1$ and $p^ = 2$ are dispersed according to the distances between the facility-SZ pairs in the set-pairs. A graphical explanation of the group selection performed according to the facility types and SZs are also provided in (c). ■*

Expressions for the objective functions are listed in the order of the criterion types according to which they may be computed, namely (a) geospatial, (b) facility-to-site and (c) inter-facility. The objective is to

$$\text{maximise } V \tag{4.15}$$

subject to:

$$V_o \leq \begin{cases} v_c(\ell, \ell^*) + M(2 - \tilde{x}_p(\ell) - \tilde{x}_{p^*}(\ell^*)), & \text{or} & (4.16a) \\ v_{f_p,c}(\ell, \ell^*) + M(2 - \tilde{x}_p(\ell) - \tilde{x}_{p^*}(\ell^*)), & \text{or} & (4.16b) \\ v_{f_p,c}^{f_{p^*}}(\ell, \ell^*) + M(2 - \tilde{x}_p(\ell) - \tilde{x}_{p^*}(\ell^*)) & & (4.16c) \end{cases}$$

for all ordered pairs of facilities and SZs in $f_p \in \mathcal{F}_p$ and $s_p \in \mathcal{S}_p$ and $f_{p^*} \in \mathcal{F}_{p^*}$ and $s_{p^*} \in \mathcal{S}_{p^*}$, and for all $\ell \in \mathbb{S}_{s_p}$ and $\ell^* \in \mathbb{S}_{s_{p^*}}$. The formulation is completed with the obligatory constraints (3.4)–(3.6).

Constraint sets (4.16a)–(4.16c) force the value of the objective function V to be less than or equal to the minimum of the criterion values determined between facility-SZ pairs in the respective set-pairs. If either (or both) of the two facility locations under consideration in a given constraint does not contain a facility in the set-pairs in question, then the objective function value V has the simple requirement of being less than or equal to a very large number M . In contrast, when both potential facility sites under consideration are assigned a facility in the particular set-pairs, the terms in (4.16a)–(4.16c) that contain the large number vanish and V is constrained only by the criterion value between the facility-SZ pairs. Since a constraint set exists for all pairs of potential facility locations in the set-pairs, V must be less than or equal to the minimum criterion value over any two facility-SZ pairs. The requirement of maximisation in the objective function (4.15) ensures that a solution which maximises the minimum criterion value is sought.

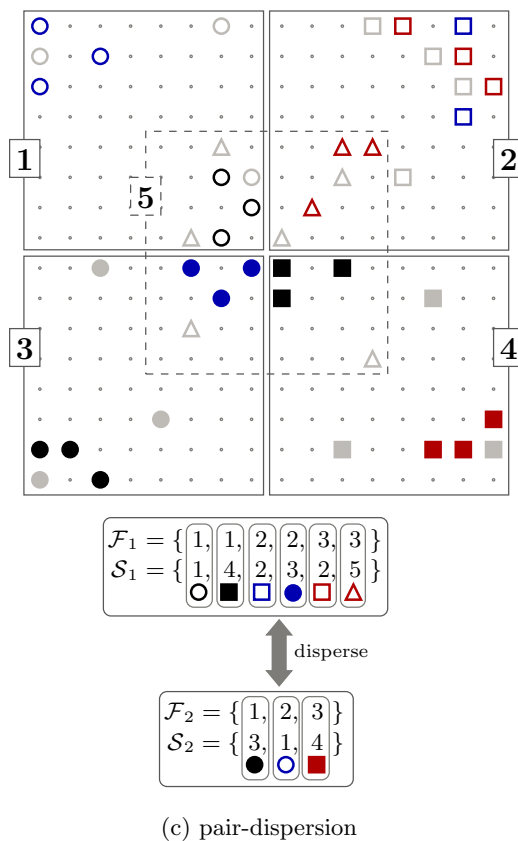
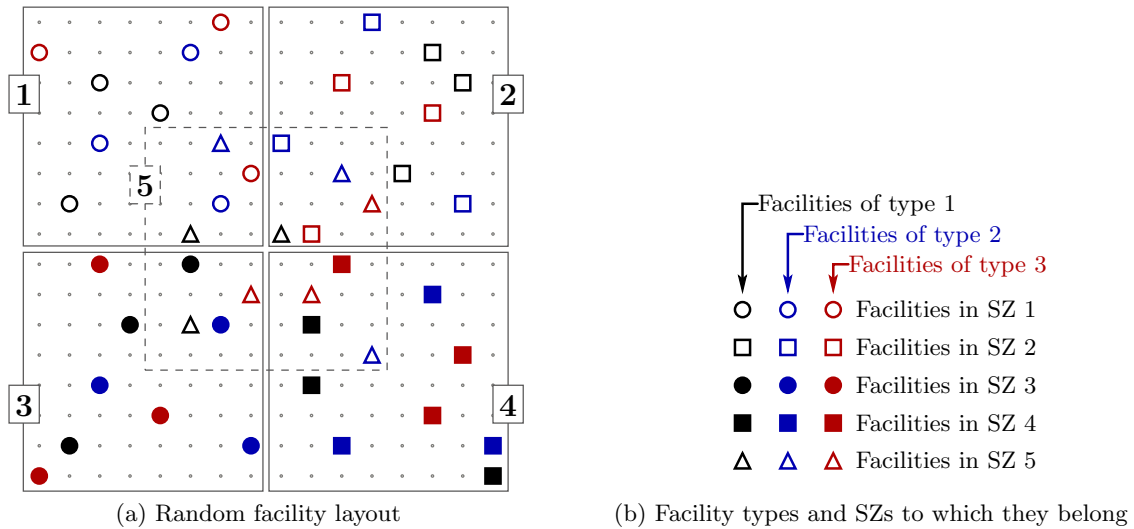


FIGURE 4.2: An illustration of the set-pair dispersion model formulated for the sets considered in Example 4.2. A random facility layout is provided in (a) and the facility layout after dispersion has been performed according to the facility-SZ set pairs specified in the example, is provided in (c).

4.2.4 Centre models

As opposed to the dispersion problem in which the objective is to *maximise* the criterion-related values determined between facilities, the centre problem aims to *minimise* these values. The models presented here are formulated for the possibility of new implementations of geospatial and inter-site criteria. An example of a (facility-to-site) centring application is that of minimising the maximum number of certain facility types that experience radio frequency interference from another facility type. Centre models were introduced in §2.2.1.

The set-pair centre model aims to minimise the maximum criterion value determined between facility-SZ pairs in two separate set-pairs, p and p^* .

Example 4.3 (Centring search sites) *Consider the same objective-specific sets and facility layout introduced in Examples 4.1 and 4.2.*

In Figure 4.3 (c) the facility-SZ pairs in the set-pairs $p = 1$ and $p^ = 2$ are centred according to inter-facility distance. A graphical explanation of the facility-SZ pairs in the set-pairs according to which the centring is performed is also provided in (c). ■*

The objective function formulation for the centring model is now presented — listed in the order of the criteria types according to which the function may be evaluated, namely (a) geospatial, (b) facility-to-site and (c) inter-facility. The objective is to

$$\text{minimise } V \tag{4.17}$$

subject to:

$$V \geq \begin{cases} v_c(\ell, \ell^*)(\tilde{x}_p(\ell) + \tilde{x}_{p^*}(\ell^*) - 1), & \text{or} & (4.18a) \\ v_{f_p, c}(\ell, \ell^*)(\tilde{x}_p(\ell) + \tilde{x}_{p^*}(\ell^*) - 1), & \text{or} & (4.18b) \\ v_{f_p, c}^{f_{p^*}}(\ell, \ell^*)(\tilde{x}_p(\ell) + \tilde{x}_{p^*}(\ell^*) - 1) & & (4.18c) \end{cases}$$

for all ordered pairs of facilities and SZs in $f_p \in \mathcal{F}_p$ and $s_p \in \mathcal{S}_p$ and $f_{p^*} \in \mathcal{F}_{p^*}$ and $s_{p^*} \in \mathcal{S}_{p^*}$, and for all $\ell \in \mathbb{S}_{s_p}$ and $\ell^* \in \mathbb{S}_{s_{p^*}}$. As is customary, the formulation is completed with the constraints (3.4)–(3.6).

Constraint sets (4.18a)–(4.18c) force the value of the objective function V to be greater than or equal to the maximum of the criterion values determined between facility-SZ pairs in the respective set-pairs. If both potential facility sites under consideration are assigned a facility in the respective set-pairs in question, the terms in (4.18a)–(4.18c) constrain V by the criterion value between the facilities. Additionally, if either (or both) of the two facility locations for a given constraint does not contain a facility-SZ pair in the respective set-pairs under consideration, then the objective function value V has the simple requirement of being greater than or equal to zero minus the criterion value — the assumption is made here that the criterion to be maximised is measured in positive values. Since a constraint exists for all logical pairings of potential facility locations in the set-pairs, V must be greater than or equal to the maximum criterion value over all facility-SZ pairs in the relevant set-pairs. The requirement of minimisation in (4.17) ensures that a solution is sought which minimises the maximum criterion value.

4.3 Generic mathematical modelling of additional constraints

A GFLoP may require constraints other than those included in the formulations for specific objectives formulated earlier in this chapter. These constraints may require that criterion values

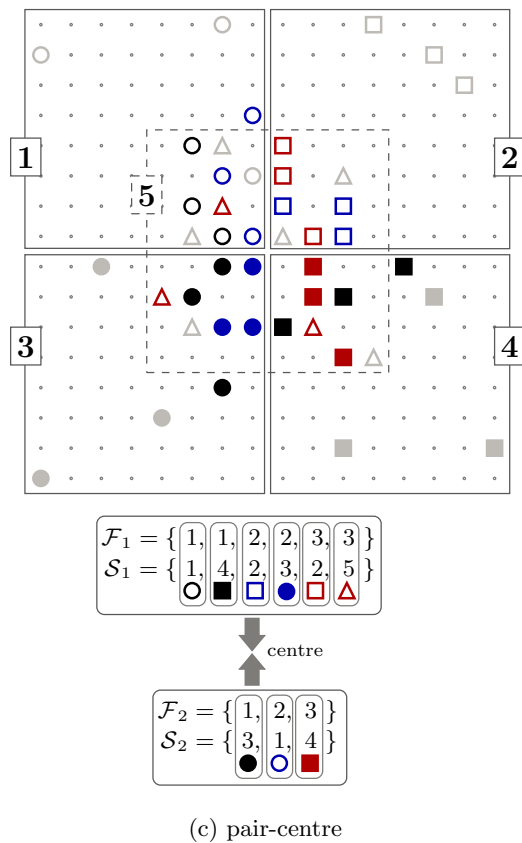
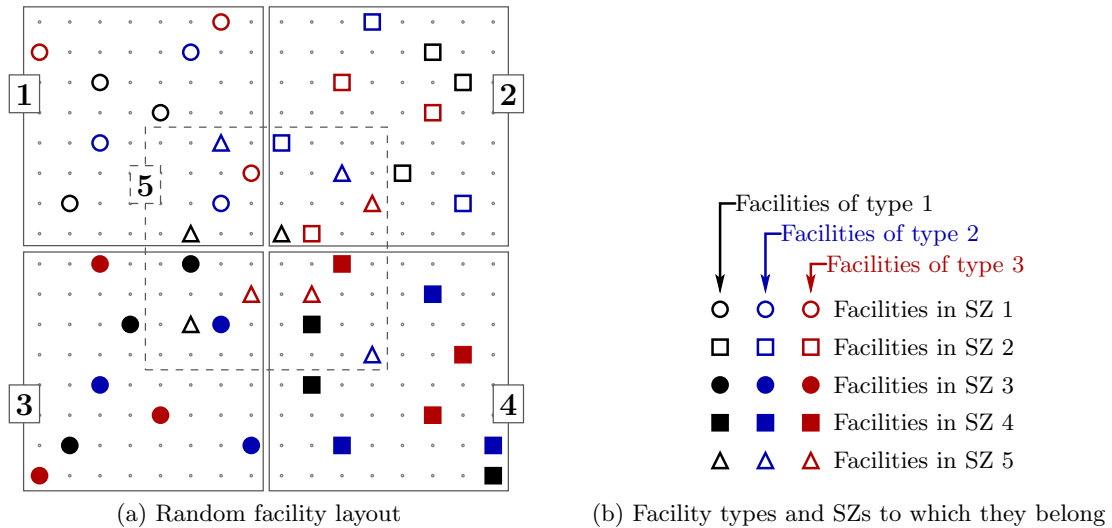


FIGURE 4.3: An illustration of the set-pair centre models discussed in Example 4.3. A random facility layout is provided in (a) and the facility layout after centring has been performed according to the facility-SZ set pairs specified in the example, is provided in (c).

must be above or below some pre-specified threshold value. It may also be possible that more than one upper or lower bound may be specified and evaluated between sets of facilities that are different to those in the objectives.

Bounds on criterion values are denoted by c_{\max} in the case of upper bounds or by c_{\min} for lower bounds. Constraints are formulated for location-specific criteria in §4.3.1, while constraints for criteria considered in dispersion and centre models are formulated in §4.3.2.

4.3.1 Location-specific constraints

Location-specific constraints enforce bounds on the values achieved by facilities when evaluated according to location-specific criteria. Practical examples of these constraints for geospatial criteria may be to limit the distances of certain facilities from specific features or to limit wind exposure depending on average annual wind measurements. A facility-dependent example may be to impose a minimum bound on the average annual solar energy generation potential achieved by a facility with a solar energy generation unit. A set-pair location-specific constraint enforces upper and/or lower bounds on criterion values determined for facility-SZ pairs in a specific set-pair.

The lower bounds on location-specific criteria values are enforced by including model constraints of the form

$$c_{\min} \leq \begin{cases} v_c(\ell) + M(1 - \tilde{x}_p(\ell)), & \text{or} \\ v_{f_p,c}(\ell) + M(1 - \tilde{x}_p(\ell)) \end{cases} \quad (4.19a)$$

$$(4.19b)$$

for all ordered pairs of facilities and SZs in $f_p \in \mathcal{F}_p$ and $s_p \in \mathcal{S}_p$, and for all $\ell \in \mathbb{S}_{s_p}$ in the case of geospatial (4.19a) or facility-dependent (4.19b) location-specific constraints, while upper bounds of the form

$$c_{\max} \geq \begin{cases} v_c(\ell)\tilde{x}_p(\ell), & \text{or} \\ v_{f_p,c}(\ell)\tilde{x}_p(\ell) \end{cases} \quad (4.20a)$$

$$(4.20b)$$

may be enforced for all ordered pairs of facilities and SZs in $f_p \in \mathcal{F}_p$ and $s_p \in \mathcal{S}_p$, and for all $\ell \in \mathbb{S}_{s_p}$ in the case of geospatial (4.20a) or facility-dependent (4.20b) criteria, respectively.

Constraint sets (4.19a)–(4.19b) force the minimum of the criterion values determined for a facility-SZ pair placement to be greater than or equal to the lower bound c_{\min} . If the facility location evaluated by a constraint does not contain a facility-SZ pair in the set-pair under consideration, then the constraint is satisfied since the second term on the right hand side of (4.19a)–(4.19b) will be a very large number M (larger than the largest criterion value possible and certainly larger than c_{\min}). In contrast, when the potential facility site under consideration is assigned a facility-SZ pair in the set-pair in question, the criterion value is constrained by the lower bound value.

The constraint sets (4.20a)–(4.20b) force the criterion values determined for a facility-SZ pair placement to be less than or equal to the upper bound c_{\max} . If the facility location evaluated by the constraint does not contain a facility-SZ pair in the set-pair in question, the constraint is satisfied since the term on the right hand side equals zero. In contrast, when the potential facility site under consideration is assigned a facility-SZ pair in the set-pair in question, the placement is constrained by the upper bound c_{\max} .

4.3.2 Inter-site constraints

Inter-site constraints enforce bounds on the values achieved *between* facilities when evaluated according to inter-site criteria. Practical examples of these constraints for geospatial criteria may be to limit the distances between facilities, while a facility-to-site constraint may be to limit the minimum number of certain facilities which are visible from a specific facility type. An inter-facility example may be to impose a lower bound on inter-facility communication capability required by a network.

Set-pair inter-site constraints enforce upper and/or lower bounds on criterion values determined between facility-SZ pairs in separate set-pairs p and p^* .

Inter-site constraint formulations are presented below in the order of (a) geospatial criteria, (b) facility-to-site criteria or (c) inter-facility criteria. The lower bound inter-site constraints may take the form

$$c_{\min} \leq \begin{cases} v_c(\ell, \ell^*) + M(2 - \tilde{x}_p(\ell) - \tilde{x}_{p^*}(\ell^*)), & \text{or} & (4.21a) \\ v_{f_p,c}(\ell, \ell^*) + M(2 - \tilde{x}_p(\ell) - \tilde{x}_{p^*}(\ell^*)), & \text{or} & (4.21b) \\ v_{f_p,c}^{f_{p^*}}(\ell, \ell^*) + M(2 - \tilde{x}_p(\ell) - \tilde{x}_{p^*}(\ell^*)), & & (4.21c) \end{cases}$$

while upper bound constraints may be formulated as

$$c_{\max} \geq \begin{cases} v_c(\ell, \ell^*)(\tilde{x}_p(\ell) + \tilde{x}_{p^*}(\ell^*) - 1), & \text{or} & (4.22a) \\ v_{f_p,c}(\ell, \ell^*)(\tilde{x}_p(\ell) + \tilde{x}_{p^*}(\ell^*) - 1), & \text{or} & (4.22b) \\ v_{f_p,c}^{f_{p^*}}(\ell, \ell^*)(\tilde{x}_p(\ell) + \tilde{x}_{p^*}(\ell^*) - 1) & & (4.22c) \end{cases}$$

for all ordered pairs of facilities and SZs in $f_p \in \mathcal{F}_p$ and $s_p \in \mathcal{S}_p$ and $f_{p^*} \in \mathcal{F}_{p^*}$ and $s_{p^*} \in \mathcal{S}_{p^*}$, and for all $\ell \in \mathbb{S}_{s_p}$ and $\ell^* \in \mathbb{S}_{s_{p^*}}$.

Constraint sets (4.21a)–(4.21c) force the criterion values determined between facility-SZ pairs in the respective set-pairs to be greater than or equal to the lower bound c_{\min} . If either (or both) of the two facility locations under consideration does not contain a facility in the set-pairs in question, then the constraint is satisfied since the right hand side of the inequality is a very large number M . In contrast, when both potential facility sites under consideration are assigned a facility in the set-pairs in question, the terms in (4.21a)–(4.21c) that contain the large number M vanish and the placement is constrained by the lower bound value.

Constraint sets (4.22a)–(4.22c) force the criteria values determined between facility-SZ pairs in the respective set-pairs to be less than or equal to the upper bound c_{\max} . If both potential facility sites under consideration are assigned a facility-SZ pair in the respective set-pairs in question, the terms in (4.22a)–(4.22c) are constrained by the upper bound value. If either (or both) of the two facility locations for a given constraint does not contain a facility in the respective set-pairs under consideration, the term on the right hand side equals zero minus the criterion value and the constraint is satisfied — under the assumption that the criterion on which the bound is enforced is measured in positive values.

4.4 Generic GFLoP formulation process

The purpose of this chapter was to provide generic formulations for objectives and constraints that may be included in GFLoP models. More specifically, the formulations were proposed in order to allow for different combinations of objectives and constraints to be included in a GFLoP

model based on various types of criteria. These formulations may be selected and used in a generic GFLo framework, included in the model as mathematical building blocks. In Figure 4.4, a generic GFLoP formulation framework is proposed, which serves the purpose of providing the necessary guidelines to formulate a GFLoP. The process begins with the specification of geospatial, facility and criteria sets to be considered and this is followed by the formulation of objectives and optional constraints for inclusion in the model. The process concludes with the inclusion of the mandatory constraints that should be present in all GFLoPs.

4.5 Chapter summary

A number of FLoPs from the operations research literature were formulated for inclusion in a generic GFLoP framework in this chapter, in partial fulfilment of Dissertation Objective IV, as stated in §1.3.

The preliminary notation required for the formulation of the framework was provided in §4.1, and included the formulation of set-pairs and set-pair decision variables. These set-pairs and set-pair variables were implemented in the generic mathematical modelling of objectives in §4.2. The problems modelled were MaxiMin and MiniMax problems in §4.2.1, covering models in §4.2.2, and dispersion and centre models in §4.2.3 and §4.2.4, respectively. Novel concepts were introduced, allowing for these problems — which typically involve only one type of facility and one SZ — to be implemented in combinations of facility types and SZs.

The mathematical modelling of optimal location constraints was performed in §4.3. The constraints allow for upper and lower bounds to be enforced on certain location-specific and inter-site criteria values determined between selected facility-SZ pairs.

The chapter closed with a description of a generic GFLoP formulation framework, illustrated in Figure 4.4, which serves the purpose of providing the necessary guidelines to formulate a GFLoP model which includes a combination of objectives and constraints. The framework is not limited to the customary procedures followed in the GFLo literature, where models are formulated according to specific problem and facility types.

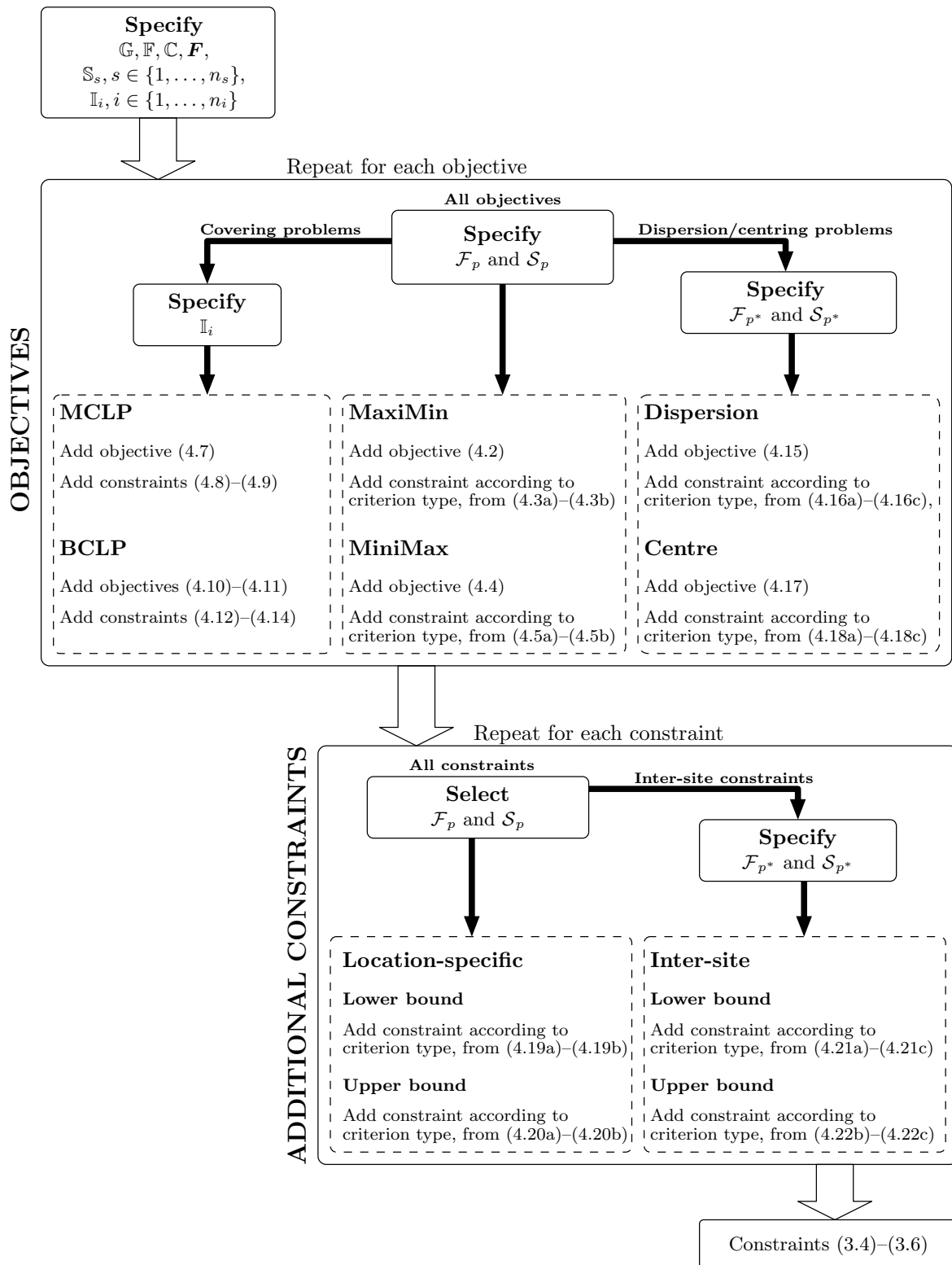


FIGURE 4.4: A generic GFLoP formulation framework which serves the purpose of providing the necessary guidelines to formulate a GFLoP model.

CHAPTER 5

Development of MO GFLo algorithms

Contents

5.1	MOO concepts	56
5.1.1	<i>Pareto-front approximation</i>	56
5.1.2	<i>Diversity preservation</i>	57
5.1.3	<i>Solution ranking</i>	58
5.1.4	<i>Constraint handling</i>	59
5.1.5	<i>Algorithm termination</i>	61
5.1.6	<i>Solution quality assessment</i>	61
5.2	GFLo candidate solution representation	63
5.3	GFLo algorithms	63
5.3.1	<i>Non-dominated sorting genetic algorithm-II</i>	63
5.3.2	<i>The novel AMOEBA algorithm</i>	69
5.4	Worked example of the GFLo solution process	73
5.4.1	<i>Terrain description</i>	74
5.4.2	<i>Facilities and zones</i>	74
5.4.3	<i>Objectives and constraints</i>	74
5.4.4	<i>Algorithm parameter selection</i>	75
5.4.5	<i>Mathematical model formulation</i>	76
5.4.6	<i>Algorithm results comparison</i>	77
5.5	A preliminary appraisal of the algorithms	79
5.6	Chapter summary	81

Two MOO algorithms are reviewed and developed anew for GFLo in this chapter. The first algorithm, the NSGA-II, is a popular member of the *multi-objective evolutionary algorithm* (MOEA) class of MOO algorithms. The NSGA-II is often implemented to solve GFLoPs, such as surveillance camera siting [53, 65, 66, 93] and wind turbine micro-siting [75, 130], and is elaborated upon in §5.3.1. The second algorithm, AMOEBA, is a novel algorithm developed specifically for the purposes of GFLo, in §5.3.2. AMOEBA employs many MOEA principles, but differs from traditional MOEAs in terms of its strategy of evolving solutions.

This chapter opens in §5.1 with an introduction to MOO concepts that are of importance with respect to both algorithms. This is followed in §5.3 by a description of the method of candidate solution representation adopted in the algorithms in §5.2, after which the NSGA-II and AMOEBA are described in some detail in the context of GFLo purposes. The method of

implementation of the algorithms is elucidated by means of a realistic worked example in §5.4, and this is followed by an appraisal of the algorithms in §5.5.

5.1 MOO concepts

The most important aspects of MO optimisation of GFLoPs and the algorithms reviewed and proposed in this chapter are elaborated upon in this section. These are the notions of Pareto-optimality, diversity preservation, solution ranking, constraint handling, algorithm termination criteria and solution quality assessment.

5.1.1 Pareto-front approximation

The notions of solution domination and Pareto-optimality in a GFLo environment were introduced in §2.3.1 and are discussed afresh in more detail in this section. As discussed previously in §2.3.1, MOO methods return a set of *non-dominated* solutions, also known as *Pareto-optimal* solutions, in objective space. These solutions may be examined by a decision maker in terms of the trade-offs that they offer between realising multiple objectives [70, 141, 142]. MO analysis is considered a more realistic approach than SO analysis in real engineering projects, especially in the context of GFLoPs. This is because MOO methods generally identify a wider range of alternative solutions and provide more information about these solutions, thereby empowering the decision maker.

The solutions that are included in the Pareto-optimal set are *non-dominated* in the sense that no other Pareto-optimal solution is better than another with respect to *all* the objectives [70, 141, 142]. Moving from one Pareto-optimal solution to another results in an improvement in at least one objective, but the degradation in at least one other. Inferior solutions that are not included in the Pareto-optimal set are said to be *dominated* by the Pareto-optimal solutions, because for each dominated solution there exists a solution in the Pareto-optimal set that is better with respect to *all* the objectives. The Pareto-optimal solutions are commonly known as the Pareto-optimal front, or simply the Pareto-front, as they form a frontier in MO objective space. The set of Pareto-optimal solutions may be explored by a decision maker and a solution that seems most appropriate may be chosen. In the context of GFLo, this entails considering solutions both in terms of their proposed facility site locations on the physical terrain and their performance values with respect to the placement criteria. Several sub-fronts may exist into which a set of solutions may be classified, with each lower front being dominated by the one above it [21, 115]. These fronts are typically ranked in order from best to worst, with all the solutions in the Pareto-front assigned *Pareto-rank* 1, and the solutions in the successive fronts assigned rank 2, then 3, and onwards until the final, ‘worst’ front has been ranked. An illustration of the ranking of solutions into fronts is provided in Figure 5.1. A sorting algorithm which may be employed to rank solutions is discussed later in this chapter.

Consider an optimisation problem with M maximisation objectives. If $\mathbf{d} = [d_1, d_2, \dots, d_n]$ represents a decision vector in respect of n decision variables in the decision space \mathbf{D} , then \mathbf{d} is said to represent a solution in this decision space. An objective function $\mathbf{f} : \mathbf{D} \mapsto \mathbf{Y}$ maps the decision vector to an objective function vector $\mathbf{y} = [y_1, y_2, \dots, y_M]$ in objective space, which represents the fitness of the solution. An objective function vector \mathbf{y}^1 dominates an objective function vector \mathbf{y}^2 (denoted by $\mathbf{y}^1 \prec \mathbf{y}^2$) if, for at least one $i \in \{1, \dots, M\}$, $y_i^1 > y_i^2$ and $y_j^1 \geq y_j^2$ for all $j \neq i$ [70]. It follows that a solution \mathbf{d}^1 may be considered better than a solution \mathbf{d}^2 when $\mathbf{f}(\mathbf{d}^1) \prec \mathbf{f}(\mathbf{d}^2)$, as illustrated in Figure 5.1 for the special case where $M = 2$. The non-

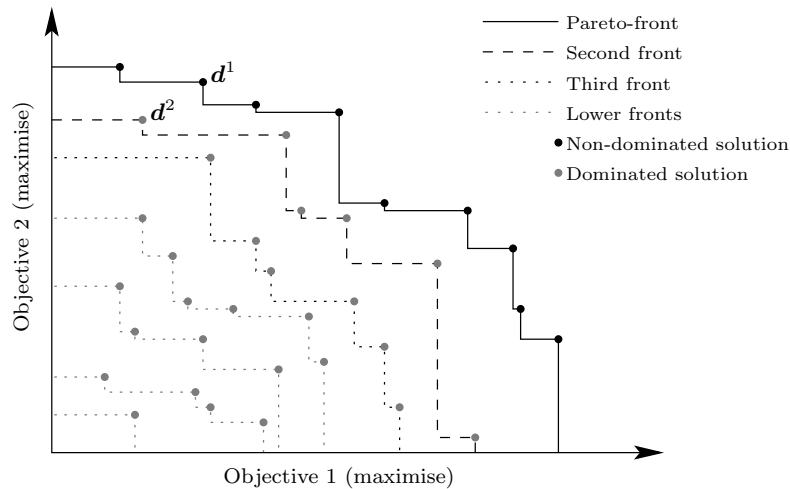


FIGURE 5.1: Solution fronts in objective space for a bi-objective maximisation problem. Solution d^2 is dominated by solution d^1 ($d^1 \prec d^2$).

dominated set which represents the best possible trade-offs between different objectives is the Pareto-optimal set, denoted by $D^* \subset D$.

Due to limited memory capacity and finite run time availability, as well as the fact that the true Pareto-front may be computationally impractical to find or (more generally) unknown, MOO algorithms merely aim to *approximate* the true Pareto-front [70, 142] instead of attempting to find it exactly. The goal in MOO is therefore to find a good approximation of the Pareto-optimal set, denoted by $A^* \subset D$. Since the true Pareto-optimal set is typically not available in real world problems [70], it is not possible to test whether the algorithm has attained any true Pareto-optimal solutions. One therefore strives to obtain a representative subset of the true Pareto-optimal set with the property of good spread along the Pareto-front. Many Pareto-front approximation algorithms exist, and different algorithms may produce sets of differing quality in terms of their closeness to the true Pareto-front, and their diversity along it. It is therefore necessary to develop performance assessment measures for the purposes of comparing approximation sets to each other. There exist various quality assessment measures which may be used to compare solution sets to one another [70, 142], which, in turn, may be used to compare the performance of different algorithms. Such measures are discussed later in this chapter.

5.1.2 Diversity preservation

In MOO, diversity preservation is the ideal of achieving a representative approximation of the Pareto-set comprising of solutions that are spread evenly along the Pareto-front in objective space [70]. This is generally achieved by using solution density information in respect of the crowding of solutions, which is most frequently calculated in terms of some measure of distance in objective space. Thorough exploration of the decision space is promoted by increasing the likelihood of selecting individuals with fewer neighbouring solutions [21].

The NSGA-II uses a so-called *crowding distance* measure to quantify solution density [21]. The quantity associated with a solution's crowding distance serves as an estimate of the perimeter of the cuboid formed around the solution, using the nearest neighbours as vertices [21]. A graphical illustration of the crowding distance density estimation scheme is provided in Figure 5.2 for a solution to a bi-objective maximisation problem. The crowding distance of solution i in the figure is the average side length of the cuboid (the dashed box) [21].

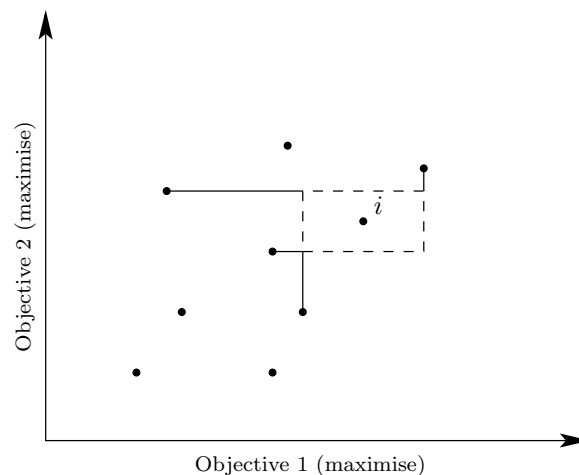


FIGURE 5.2: Crowding distance for solution i , indicated by the perimeter of the cuboid (the dashed rectangle) [21].

The computation of the crowding distance according to the method employed by the NSGA-II requires that solutions be sorted in ascending order of magnitude along each objective axis [21]. Let $V[i]|m$ denote the objective function value of the i^{th} candidate solution for the m^{th} objective (here i is in the sorted list). The crowding distance for the i^{th} solution of the m^{th} objective is denoted by $c_i|m$, while the overall crowding distance of solution i over all objectives is denoted by c_i . To ensure that the extremal solutions are selected (for the purposes of a well-spread distribution along the non-dominated front) an infinite crowding distance is assigned to them, that is $V[1]|m = \infty$ and $V[N]|m = \infty$. The crowding distances of the remaining intermediate solutions are determined as the normalised distance $(V[i+1]|m - V[i-1]|m) / (V[N]|m - V[1]|m)$. The overall crowding distance of each individual solution is taken as the accumulated value of its crowding distances over all of the objectives [21]. A solution with a small crowding distance relative to those of the other solutions effectively indicates that the solution is comparatively crowded — an undesirable feature. Similarly, a large crowding distance value indicates that the solution is relatively isolated, which is preferable. A pseudocode description of the method of crowding distance calculation appears in Algorithm 5.1.

5.1.3 Solution ranking

The process of sorting the solutions of a population into fronts in order to determine Pareto-ranks and to identify the approximated Pareto-front was introduced in §5.1.1. The algorithm that is used by the NSGA-II to achieve this is called the *Fast Non-dominated Sorting Algorithm* (FNSA) [21].

The FNSA computes, for each candidate solution i , a dominance count d_i of the number of solutions by which it is dominated. Similarly, a set S_i of all the solutions that are dominated by solution i is computed. Solutions achieving a dominance count value of $d_i = 0$ are placed in a separate set \mathcal{F}_1 , called the *first non-dominated front* (the approximate Pareto-front), and all these solutions are assigned dominance rank 1. To determine the dominance ranks of the remaining solutions, the algorithm cycles through each candidate solution j in S_i and decrements its d_j value by one for each of the solutions i in \mathcal{F}_1 . This results in the dominance effect of solution i on solution j being discounted from j 's dominance count. All of these remaining solutions achieving a dominance count value of $d_j = 0$ are assigned dominance rank 2, and are placed in a separate set \mathcal{F}_2 , called the *second non-dominated front*. This procedure is

Algorithm 5.1: Crowding distance assignment algorithm [21]

Input : A population of solutions \mathbf{P} , where each solution is a specific assignment of values to a vector of decision variables \mathbf{x} , and a vector \mathbf{y} of two or more computed objective function values for each solution.

Output: The crowding distance of population member, $\mathbf{P}[1]_{\text{dist}}, \dots, \mathbf{P}[N]_{\text{dist}}$.

```

1  $N = |\mathbf{P}|;$  //  $N$  is the number of solutions in  $\mathbf{P}$ 
2 for all  $i \in \mathbf{P}$  do
3    $\mathbf{P}[i]_{\text{dist}} \leftarrow 0;$  // initialise crowding distance
4 end
5 for all  $M$  objectives do
6    $\mathbf{P} = \text{sort}(\mathbf{P}, m);$  // sort population using value of objective  $m$ 
7    $\mathbf{P}[1]|_m \leftarrow \infty;$  // so that boundary points are always selected
8    $\mathbf{P}[N]|_m \leftarrow \infty;$ 
9   for  $i = 2$  to  $(N - 1)$  do
10     $\mathbf{P}[i]_{\text{dist}}|_m \leftarrow \mathbf{P}[i]_{\text{dist}}|_m + (\mathbf{P}[i + 1]|_m - \mathbf{P}[i - 1]|_m) / (m_{\text{max}} - m_{\text{min}});$ 
11   end
12 end

```

repeated until all the solutions have been partitioned into non-dominated fronts. The FNSA described here has a run time of at most $O(MN^2)$, where M denotes the number of objectives and N denotes the size of the population¹. A pseudocode description of the FNSA appears in Algorithm 5.2.

5.1.4 Constraint handling

Evolutionary optimisation algorithms are generally formulated for *unconstrained* optimisation — *i.e.* no consideration is given to candidate solutions' feasibility status, or a measure of their infeasibility, in the computation of objective function values. Constraint violations are typically accommodated by one of two methods, namely the use of *penalty functions* or the use of *repair algorithms* [13, 14, 20, 72, 87].

Penalty functions impose penalties on the objective function values associated with infeasible solutions — these penalties are typically based on the magnitude of the constraint violation multiplied by penalty factors which scale the violations to the same order of magnitude as the objective functions [13]. Ideally, this should result in an infeasible solution being slightly 'less feasible' than a feasible one, so that the population is encouraged to continue to evolve towards feasibility [87]. The penalty factors are problem-specific, and are typically found by thorough trial and error fine tuning [14]. Multiple constraints may be dealt with by weighted aggregation of such penalty functions, which further complicates matters by requiring weighting coefficients which specify the relative importance of constraints, in addition to requiring an additional level of fine tuning.

Repair algorithms attempt to 'fix' infeasible solutions. These algorithms may be seen as local search methods that direct their search towards areas which reduce constraint violations, ultimately generating feasible solutions from infeasible ones [13]. Designing a repair algorithm can, however, be as complex as solving the problem itself.

¹It should be noted that efforts have been made to reduce the computational complexity of the FNSA [63, 115], but are not considered in this dissertation.

Algorithm 5.2: Fast non-dominated sorting algorithm [21]

Input : A population of solutions P , where each solution is a specific assignment of values to a vector of decision variables \mathbf{x} , and a vector \mathbf{y} of two or more computed objective function values for each solution.

Output: The original population partitioned into successively dominating fronts, $\mathcal{F}_1, \dots, \mathcal{F}_n$.

```

1  $\mathcal{F}_1 \leftarrow \emptyset$ ;
2 for all  $i \in P$  do
3    $S_i \leftarrow \emptyset$ ;           //  $S_i$  is set of solutions which  $i$  dominates
4    $d_i \leftarrow 0$ ;           //  $d_i$  is count of how many solutions dominate  $i$ 
5   for all  $j \in P$  do
6     if  $i \prec j$  then
7        $S_i \leftarrow S_i \cup j$ ;           //  $i$  dominates  $j$ 
8     else if  $j \prec i$  then
9        $d_i = d_i + 1$ ;           // increment the domination counter of  $i$ 
10    end
11  end
12  if  $d_i = 0$  then
13     $i_{\text{rank}} \leftarrow 1$ ;           //  $i$  belongs to the first (non-dominated) front
14     $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup j$ ;
15  end
16 end
17  $n \leftarrow 1$ ;
18 while  $\mathcal{F}_n \neq \emptyset$  do
19    $Q \leftarrow \emptyset$ ;
20   for all  $i \in \mathcal{F}_n$  do
21     for all  $j \in S_i$  do
22        $d_j \leftarrow d_j - 1$ ; // discount effect of  $n^{\text{th}}$  front on  $j$ 's domination count
23       if  $d_j = 0$  then
24          $j_{\text{rank}} \leftarrow n + 1$ ;           //  $j$  is a member of the next front
25          $Q \leftarrow Q \cup j$ ;
26       end
27     end
28   end
29    $n \leftarrow n + 1$ ;
30    $\mathcal{F}_n \leftarrow Q$ ;
31 end

```

Whereas penalty functions and repair algorithms aim to incorporate infeasible solutions into the solution population, a simpler method of constraint handling will be used in this dissertation. The *rejection* method simply discards any infeasible solutions encountered during the search process. This may make it difficult to establish a feasible population and often results in premature convergence [72]. Approaches toward alleviation of the strict enforcement measure of the rejection method are, however, introduced for the respective algorithms later in this chapter.

5.1.5 Algorithm termination

MOO algorithms aim to *approximate* the true Pareto-front as accurately as possible. Since the true Pareto-front is not known in real world problems, however, it is necessary to establish a protocol for determining when a search should be terminated so as to prevent it from striving towards the true Pareto-front indefinitely. The most commonly employed methods, and by far the simplest, are to impose limits upon 1) the total number of iterations, 2) the total run-time, or 3) both, in which case whichever limit is reached first signals termination [61, 82].

Imposing a maximum iteration limit allows results from different studies to be compared independently of the computer software and hardware used. This method, however, becomes impractical when using algorithms which require different population sizes or in which the population sizes change dynamically. Since the algorithms put forward in this chapter make use of dynamic population sizing, this termination criterion is not considered in this dissertation.

A fair mechanism would seem to be to impose a limit on total run-time, such that an algorithm must do its best within an allotted time-frame. With the aim of promoting fairness in comparing the results of different algorithms, however, this method is only valid if all algorithms demonstrate similar behaviour with regards to their fitness improvement profiles and are implemented on the same computing platform. Algorithms with a linear average fitness improvement profile (*e.g.* greedy heuristics) may outperform ones with exponential improvement profiles (*e.g.* genetic algorithms) if the time budget is too brief. Estimating a satisfactory time limit to impose is a further consideration. Using average running times is one possibility — this may, however, be excessive and/or impractical to compute if the algorithms are very slow, as may be expected for GFLoPs with large SZs and/or a large number of facilities to place.

Considering the above-mentioned disadvantages to imposing limits on iterations and run-times for the purposes of signalling algorithm termination, alternative methods are employed in this dissertation. Termination is determined here according to *Pareto convergence measures*. Since the true Pareto-set is not available, these convergence measures are defined in terms of convergence towards a static population, in terms of either solution diversity or objective function values. Appropriate convergence measures are suggested later in this chapter with the aim of best suiting each algorithm's unique characteristics.

5.1.6 Solution quality assessment

There exist various methods to compare MOEAs qualitatively according to the solutions returned in their approximate Pareto-optimal sets [70] — these performance measures should be *Pareto-compliant*. Pareto-compliance means that an indicator should only give preference to one approximation set over another set, if the latter does not *weakly* dominate the former (§5.1.1). One approximate Pareto-set A is said to weakly dominate approximate set B ($A \prec B$) if for each $y \in B$, either $y \in A$ or there exists an $x \in A$, such that $x \prec y$. Two popular Pareto-compliant measures discussed in [70] are the *dominance rank* quantifier and the *hypervolume* measure.

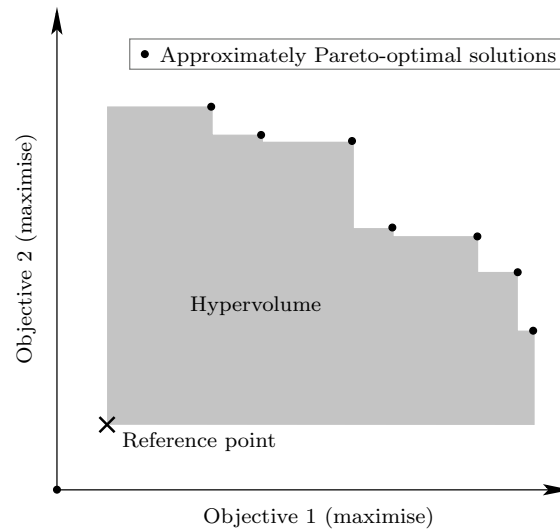


FIGURE 5.3: An illustration of the hypervolume performance measure.

The dominance rank quantifier uses a *binary weak-domination indicator* which compares approximation sets obtained by different algorithms to each other and determines a measure of dominance between the sets. This indicator compares approximation sets to each other and assigns ranks to the approximation sets, determined according to weak domination between these sets — in a manner similar to solution ranking performed according to solution domination principles (as described in §5.1.1). The approximation sets are sorted in ranks from lowest to highest and the lower the rank of an approximation set, the better the approximation set. Statistical ranking tests may then be used to determine whether the ranks assigned to one algorithm are significantly smaller than the ranks assigned to another — indicating that the former algorithm statistically outperforms the latter. As in the case with the solution ranking procedure — where solutions which are tied according to their ranks may be further distinguished according to their crowding distances — it is possible that algorithms may achieve identical dominance statistics, in which case alternatives, such as distinguishing the algorithms by average hypervolume values, may be considered.

The unary hypervolume metric proposed by Zitzler and Thiele [141] involves a fine-grained analysis in which the total hypervolume in objective space dominated by an approximation set is measured relative to a reference point which should be dominated by the entire solution set. More space is dominated by larger hypervolumes, hence these are desired. A graphical representation of the hypervolume measure in two dimensions is provided in Figure 5.3. The advantage of using this measure is that it simultaneously measures closeness to the true Pareto-front (if it is known), evenness in solution spread, and solution diversity [70].

These performance assessment measures may still be unsatisfactory, such as when algorithms are equally good at approximating solutions in certain regions of objective space, but vary dramatically in other regions. A graphical comparison of results remains the simplest assessment method, and may reveal additional information hidden by quantitative measures. It is therefore important to include graphics in addition to ordinary quality measures, although thus presenting approximation sets for optimisation problems with more than three objectives may become problematic.

The two algorithms considered in this chapter for integration in a GFLO solution framework offer novel solution approaches that require intensive empirical parameter selection studies and analyses in respect of their performance on different terrain types and with different types and

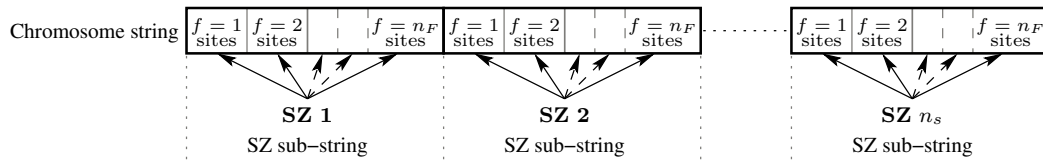


FIGURE 5.4: Chromosome string representation scheme employed for the purposes of solving GFLoPs, consisting of SZ sub-strings and their constituent facility assignments according to facility types.

numbers of facilities. Such an in-depth comparison and testing process falls outside the scope of this dissertation. The assessment of solution quality in this dissertation is merely considered an *exploratory* process which may provide *indications* of general algorithm performance for future consideration. Therefore, a graphical comparison of results in the form of the attainment sets of algorithms (the globally best set of non-dominated solutions from all optimisation runs) and the hypervolume measure are considered sufficient to evaluate algorithm performance (hypervolumes are also easily interpreted when their numerical values are analysed in conjunction with graphical result presentations). Hypervolumes are normalised to the range $[0, 1]$ for presentation purposes in this dissertation by scaling the hypervolume of each approximation set in relation to the largest hypervolume found by combining all approximation sets produced by the two algorithms.

5.2 GFLo candidate solution representation

A candidate solution is represented in this dissertation as a *chromosome* (string) of candidate facility location numbers. The chromosome lists the candidate facility sites grouped according to SZ sub-strings which are listed in numerical order from $SZ\ s = 1$ to $SZ\ s = n_s$. Furthermore, within each SZ's grouping, the candidate facility location numbers correspond to the facility types placed within each SZ, listed in order from facility type $f = 1$ to $f = n_F$, although not all facility types need to be included within a SZ grouping as a result of facilities of a specific type not being assigned to a specific SZ. It may also be possible for a SZ sub-string to include multiple entries of a facility type depending on the number of facilities of the specific type assigned to the SZ. An illustration of this chromosome representation scheme is provided in Figure 5.4. The chromosome representation of a candidate solution is illustrated by means of an example.

Example 5.1 (Chromosome representation of a GFLo solution) Suppose the candidate solution of Example 3.2 is represented as a chromosome. The site locations of the candidate solution are shown in Figure 5.5 (a), while the corresponding chromosome representation is provided in Figure 5.5 (b). ■

5.3 GFLo algorithms

The NSGA-II and novel AMOEBA algorithms are developed in this section.

5.3.1 Non-dominated sorting genetic algorithm-II

The NSGA-II performs evolution-inspired selection processes and modification operators on populations of solutions, until a termination criterion is met [21]. An initial population of candidate solutions of size N is stochastically generated according to a uniform distribution. Each of these

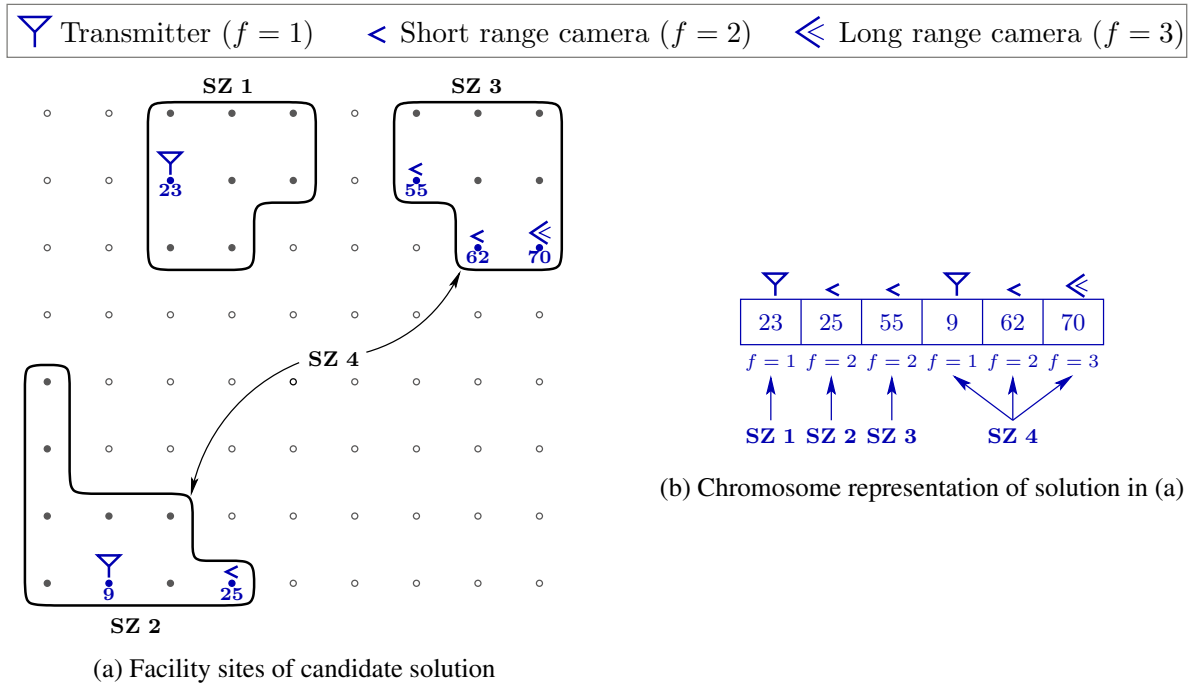


FIGURE 5.5: The GFLO candidate solution representation scheme for the solution in Example 5.1.

solutions represents a feasible facility location combination. The FNFA is then used to rank and sort solutions into fronts in objective space (as described in §5.1.3), after which a crowding distance density measure is computed for each candidate solution (as described in §5.1.2). The ranks and crowding density measures are used during the offspring solution generation. The process of offspring generation begins with the creation of a ‘mating pool’ of solutions of configurable size (a size of half the population size is not uncommon [21]). The solutions entered into the mating pool are selected stochastically from the current population. Tournament selection follows, whereby a number of solutions from the mating pool are selected stochastically and entered into the tournament — tournament sizes as small as 2 are not uncommon [21]. From this tournament the *fittest* solution is selected according to lowest dominance-rank and, in the event of a tie, according to the largest crowding density. This solution is the first *parent* solution. A second parent solution is selected in the same manner from a separate tournament, after which *crossover* and *mutation* operators — operators that mimic natural evolution — are applied to the two parents in order to generate two offspring solutions.

To perform crossover, one or more, say k , *crossover points* are chosen along the chromosome representations of the parent solutions, at the same k locations for both parents. The location of the k points along the chromosomes may be chosen arbitrarily for each parent pair, or fixed for all crossover operations [25]. Each of the parent strings are ‘cut’ at the crossover points, resulting in $k+1$ sub-strings from each chromosome’s cuts. The sub-strings are then interchanged alternately to create new chromosomes, so that the newly generated chromosome strings consist of ordered sub-strings that alternate in respect of the parent chromosome’s from which the sub-strings are sourced. This type of crossover creates new site combinations as solutions, but does not alter the constituent sites of the solutions. Since the parents that are selected for crossover typically perform well with respect to the objective functions, the offspring solutions inherit some of the strong properties of their parents, but at the same time also explore new solution combinations. Not all parent solutions chosen by tournament selection, however, undergo crossover. Instead, crossover is subject to a *crossover probability*, denoted by p_c .

After the crossover stage, a final measure of solution modification, namely mutation, is introduced which promotes solution diversity in the sense of introducing new sites into the location combinations of solutions, as opposed to merely exchanging these locations from an existing collection of sites. This is achieved by selecting a random site in the chromosome string and exchanging it for a randomly selected site from the SZ that corresponds to the selected point. Mutation occurs for each solution in the post-crossover offspring population with a *mutation probability*, denoted by p_m . The crossover and mutation operations are demonstrated by means of an example.

Example 5.2 (NSGA-II crossover and mutation operations) *Suppose the candidate solution of Example 5.1 and a second candidate solution are selected as parent solutions, to undergo crossover. The site locations and chromosome representations of the candidate solutions are shown in the top half of Figure 5.6. Suppose the crossover operator employs two-point crossover, randomly selected at the points indicated in the parent chromosome strings. The bottom part of Figure 5.6 contains the resulting offspring solutions with respect to their string representations and site locations.*

Suppose the first offspring solution of Figure 5.6 undergoes mutation at a randomly selected point in its chromosome string, as shown in the left half of Figure 5.7. The site locations and chromosome representation of the resulting mutated solution is shown in the right half of Figure 5.7. ■

The rejection constraint handling technique is followed in this dissertation (as discussed in §5.1.4), which means that any offspring solution that violates a constraint is immediately rejected. This approach may make it difficult to establish a feasible population and may result in premature convergence. In order to address this shortcoming, multiple crossovers at multiple randomly selected crossover points are attempted between two parents, until crossover points that result in a feasible offspring pair is found, or until a maximum number of crossover attempts have occurred — in which case crossover is considered to be impossible between the two parents and the crossover process between the two parents is abandoned. In this dissertation a maximum of ten crossover attempts per parent pair are permitted. This approach is not followed in the mutation operator — *i.e.* a solution selected to undergo mutation undergoes a single mutation attempt and the new candidate is accepted if it is feasible — although multiple mutation attempts is a consideration for future implementations.

Following the generation of the offspring population as a result of crossover and mutation, a larger intermediate population is formed by combining the current population's solutions with the offspring solutions. Since the offspring population is usually also of size N , an intermediate population of size $2N$ typically results. As a result of the strict constraint handling technique followed in this dissertation, the offspring population size may, however, vary at each generation, resulting in an intermediate population that also varies in size (at least N and at most $2N$). This larger population is then sorted and ranked into non-dominated fronts by the FNNSA. The next population is generated by selecting the solutions in the first non-dominated front, then the solutions in the second non-dominated front, and so forth, until a population size N is reached. In the event that a particular front consists of more solutions than is required to complete the new population, the solutions in that front are sorted with respect to their crowding distances and solutions are added to the new population in descending order of crowding distance until the population has size N .

This process is repeated until a pre-specified termination condition is met, as discussed in §5.1.5. A very simple *Pareto convergence measure* is adopted for this purpose in this dissertation.

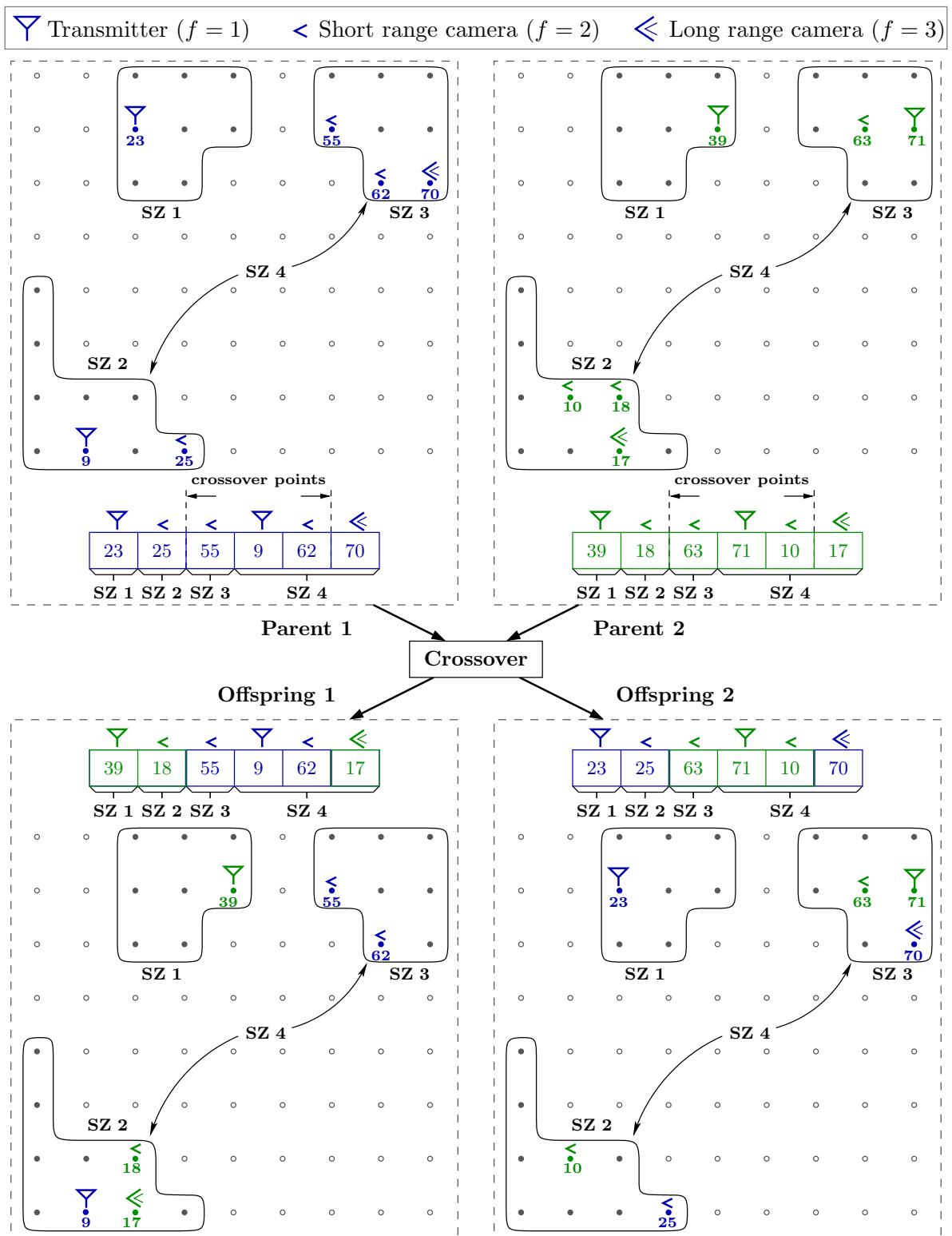


FIGURE 5.6: NSGA-II crossover performed on candidate solutions in Example 5.2.

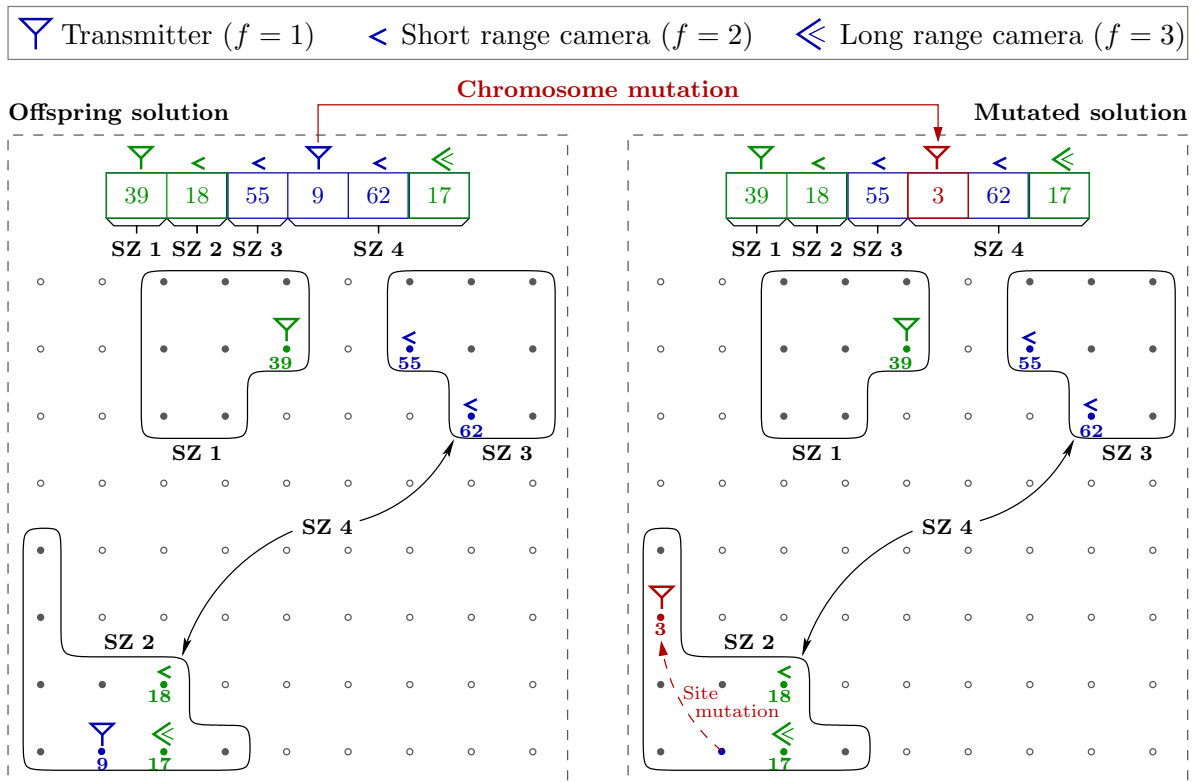


FIGURE 5.7: Mutation performed on the first offspring solution in Figure 5.6 of Example 5.2.

This measure is based on the average similarity of the non-dominated solutions per population when compared to those in the previous population — similarity is measured in this context as duplication in terms of chromosomes' site/facility-type assignments within SZ sub-strings. Convergence is considered to have occurred and the algorithm is terminated when the solutions in the populations of successive Pareto-fronts are above a pre-specified *Pareto similarity threshold percentage* over a fixed number of generations.

A pseudocode description of the main NSGA-II loop appears in Algorithm 5.3.

NSGA-II Parameter Selection

As with most optimisation algorithms, the NSGA-II has a number of parameters which may be tuned to adjust algorithm performance. GFLoPs may vary widely in their intended application and in the physical terrain that is considered (in terms of terrain surface type, features and size). The parameter values required for good algorithm performance are therefore problem-specific and require separate sensitivity analyses. The aim of each such sensitivity analysis is to find a parameter combination that results in approximation sets containing solutions that are *consistent* in terms of their objective function values and, more importantly, consistently yield *good* quality approximation sets (in terms of solution diversity and quality) compared to other parameter combinations.

After solving various GFLoPs, it was observed that the NSGA-II parameters which are the most influential in terms of approximation set quality are the population size and the crossover and mutation probabilities. The following general guidelines should be followed in respect of selecting these parameter values:

Algorithm 5.3: Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [21]

Input : A MOO problem where solutions are an assignment of values to a vector of decision variables \mathbf{x} , a population size N , a set of constraints, M objective functions, and a boolean algorithm termination variable T .

Output: An approximation of the Pareto-optimal solution set in multi-objective space, \mathbf{A}^* .

```

1 Randomly generate an initial population of solutions  $\mathbf{P}_0$  of size  $N$ .
2 Rank and sort  $\mathbf{P}_0$  using the FNSA [Algorithm 5.2].
3 Calculate the crowding distance of the solutions in  $\mathbf{P}_0$  [Algorithm 5.1].
4 Create offspring population  $\mathbf{Q}_0$  of size  $N$  using binary tournament selection from  $\mathbf{P}_0$ ,
  crossover and mutation.
5  $t \leftarrow -1$ ;
6 while  $T = 0$  do
7    $t \leftarrow t + 1$ ;
8    $\mathbf{R}_t \leftarrow \mathbf{P}_t \cup \mathbf{Q}_t$ ;
9   Partition  $\mathbf{R}_t$  into fronts  $\mathcal{F}_1, \mathcal{F}_2, \dots$  by means of the FNSA.
10   $\mathbf{P}_{t+1} \leftarrow \emptyset$  and  $i \leftarrow 1$ 
11  while  $|\mathbf{P}_{t+1}| < N$  do
12    if  $|\mathcal{F}_i| + |\mathbf{P}_{t+1}| \leq N$  then
13       $\mathbf{P}_{t+1} \leftarrow \mathbf{P}_{t+1} \cup \mathcal{F}_i$ ;
14    else if  $|\mathcal{F}_i| + |\mathbf{P}_{t+1}| > N$  then
15      Calculate crowding distance for all solutions in  $\mathcal{F}_i$ .
16      Sort  $\mathcal{F}_i$  members in order of decreasing crowding distance.
17       $\mathbf{P}_{t+1} \leftarrow \mathbf{P}_{t+1} \cup \{\text{the first } (N - |\mathbf{P}_{t+1}|) \text{ elements of } \mathcal{F}_i\}$ ;
18    end
19     $i \leftarrow i + 1$ ;
20  end
21  Calculate the crowding distance for each  $\mathbf{x} \in \mathbf{P}_{t+1}$ .
22  Create  $\mathbf{Q}_{t+1}$  of size  $N$  using crowded comparison selection, crossover and mutation.
23 end
24  $\mathbf{A}^* \leftarrow \mathbf{P}_t$ ;

```

1. **Population size:** As the number of facilities to be placed increases, the appropriate population size for the problem instance should also increase. An explanation for this guideline is that more facilities to be placed result in larger decision spaces, requiring larger populations to explore these spaces adequately.
2. **Crossover probability:** As the number of facilities to be placed increases, the crossover probability should increase. This guideline is attributed to the fact that the length of the chromosomes increase as the number of facilities to be placed increases, hence requiring more crossover operations to explore the decision space and find good combinations of the chromosomes' constituent facilities.
3. **Mutation probability:** As the number of facilities to be placed increases, the mutation probability should decrease. This guideline is a result of the larger population sizes and longer chromosomes automatically increasing the candidate sites that are explored in the SZs as the number of facilities to be placed increases.

In this dissertation, the NSGA-II employs the following parameters: Population sizes throughout are between 100 and 300. Crossover probabilities are between 0.7 and 0.9, while mutation

probabilities are between 0.15 and 0.25. These parameter value choices are the result of extensive numerical experimentation. The unusually high mutation rates employed result from the type of crossover operator adopted. It is standard practice for chromosomes to consist of binary representations of the decision variables and for crossover-points to be randomly chosen along these binary strings, resulting in the inherent mutation of offspring solutions [25]. The crossover employed for GFLo purposes does not, however, alter the sites in a solution combination (only the combination), and the inherent mutation is absent as a result. The absence of crossover-induced mutation is therefore countered by the high mutation rates after crossover, thus ensuring that the decision space is sufficiently explored.

Furthermore, the number of crossover points employed are either single or double — single when the number of facilities is no more than five, and double when the number of facilities is at least six. For the purposes of selecting parent solutions, a tournament size of four and a pool size of half the population size is used throughout. Convergence parameters are selected and motivated for each specific problem in later examples.

5.3.2 The novel AMOEBA algorithm

Asexual multi-objective evolutionary-based algorithm (AMOEBA) is a new algorithm that has been developed specifically for the purpose of GFLo — it may, however, also prove to be efficient in solving other optimisation problems. AMOEBA is, in many respects, an MOEA. This is due to it evolving populations of solutions in a strict elitist manner using *some* of the principles of evolution, until convergence is observed. The difference between AMOEBA and traditional MOEAs lies with the offspring generation process that is followed. Whereas traditional evolutionary algorithms incorporate interaction between solutions in the population in the process of generating offspring solutions, AMOEBA generates offspring solutions independent of any other solutions in the population by employing *asexual* reproduction instead.

The algorithm follows viral mutation principles in order to generate new *strains* of candidate solutions. Furthermore, similar to the start of an infection in real life, the search process requires a single initial candidate solution only, called the *source amoeba*, after which the infection spreads through the decision space as a growing population of evolving strains. In order to generate offspring solutions, each strain of the infection undergoes multiple mutations that result in new strains. Care is taken to control the size of the infection (*i.e.* the population size), since the *infection rate* may be rapid, depending on the mutation scheme employed.

The AMOEBA mutation scheme employed in this dissertation is an implementation of the popular *Teitz & Bart Algorithm* (TBA) [126], which is an algorithm for solving location-allocation problems. The TBA is in essence a single-solution, single-objective swap algorithm that starts by randomly generating a single starting solution comprising n_f facility placements. The typical approach is to identify a set of candidate locations that are not included in the current solution — in some instances *all* the remaining locations that are not in the initial solution are included in this set — after which a candidate location in this set is selected and consecutively swapped with each location in the current solution. If any objective function value improvements are achieved as a result of candidate location swaps with the current locations, the swap that results in the best improvement in the single objective function value is accepted as the incumbent solution [126]. This procedure is then repeated for the remaining locations in the candidate location set relative to a continually updated incumbent solution, until all locations in the search space have been swapped and the final incumbent solution is accepted as the best alternative.

The AMOEBA mutation operator utilises a similar swap procedure to that implemented by the TBA. In the context of GFLoPs, however, large SZs comprise large numbers of candidate

locations which may be impractical to swap exhaustively with each facility in a candidate solution's chromosome string, especially when this procedure is repeated for each of the candidate solutions in a large infection with numerous solutions. The computational burden of the swap process is reduced by imposing pre-specified swap limits per candidate solution in the infection.

For each SZ included in the chromosome string a candidate location containing no facility is arbitrarily selected from the same SZ for swaps. Then, for each facility in each SZ sub-string in the chromosome, the chromosome site location is swapped with the arbitrarily selected swap location for the SZ in question. Each swap that is performed results in a new strain, and each strain is the result of the single swap which does not 'remember' previous swaps. Therefore, strains resulting from swaps performed upon a single parent strain differ only in terms of the single facility location that was swapped and, as a result, inherits the majority of the parent strain's characteristics. It is possible that future implementations may enforce swaps involving more than one arbitrarily selected swap locations per SZ². It should also be noted that each strain's mutation process is performed independently — the new sites introduced are therefore specific to each strain's mutation. This allows for rapid diversification of candidate sites included in the candidate solutions that make up the entire infection. An example of the AMOEBA mutation procedure follows.

Example 5.3 (AMOEBA mutation) *Suppose the candidate solution in Examples 5.1 and 5.2 undergoes AMOEBA mutation. The mutation process is demonstrated graphically in Figure 5.8. For each SZ sub-string in the parent chromosome strain, a single location in the SZ containing no facility is selected for substitution with the facilities in its SZ's sub-string. Each facility in the sub-string is swapped, in turn, with the new location, thereby generating a new offspring solution that differs from the parent solution only at the facility location that has been swapped.* ■

Each strain's mutation process results in a number of mutated offspring strains equal to the number of facilities to be placed (since each facility in the strain is swapped once, resulting in an offspring strain). This is followed by each of the offspring strains further developing another set of offspring strains, resulting in an exponential growth in the size of the infection. The size of the infection therefore requires control, and this is achieved by the following considerations:

1. **Constraint handling by means of rejection:** New strains that violate constraints are rejected. The fact that multiple offspring strains result from a single strain's mutation procedure means that it is reasonable to expect some infeasible offspring rejections, while it should also be expected that a fair number of feasible offspring are likely to survive. Multiple mutation attempts, as suggested for the NSGA-II in §5.3.1, is therefore not considered for AMOEBA in this dissertation since the number of feasible offspring strains emanating from each parent strain is expected to be sufficient for adequate infection spread.
2. **Controlled acceptance of feasible strains:** Compared to the TBA — in which candidate solutions are evaluated with respect to a single objective — new strains are evaluated with respect to multiple objectives. Whereas the TBA selects the single best solution that resulted from the swaps (if any improvement is found), AMOEBA accepts *all* offspring strains emanating from mutation that exhibit improvements over the parent strain with respect to *any* of the objective functions (in spite of any weakening in objective function values with respect to the other objectives). The reason for this design choice is that, by

²This has, in fact, been tested with success, yet the advantages and disadvantages of this approach over the single swap method followed in this dissertation requires further empirical study.

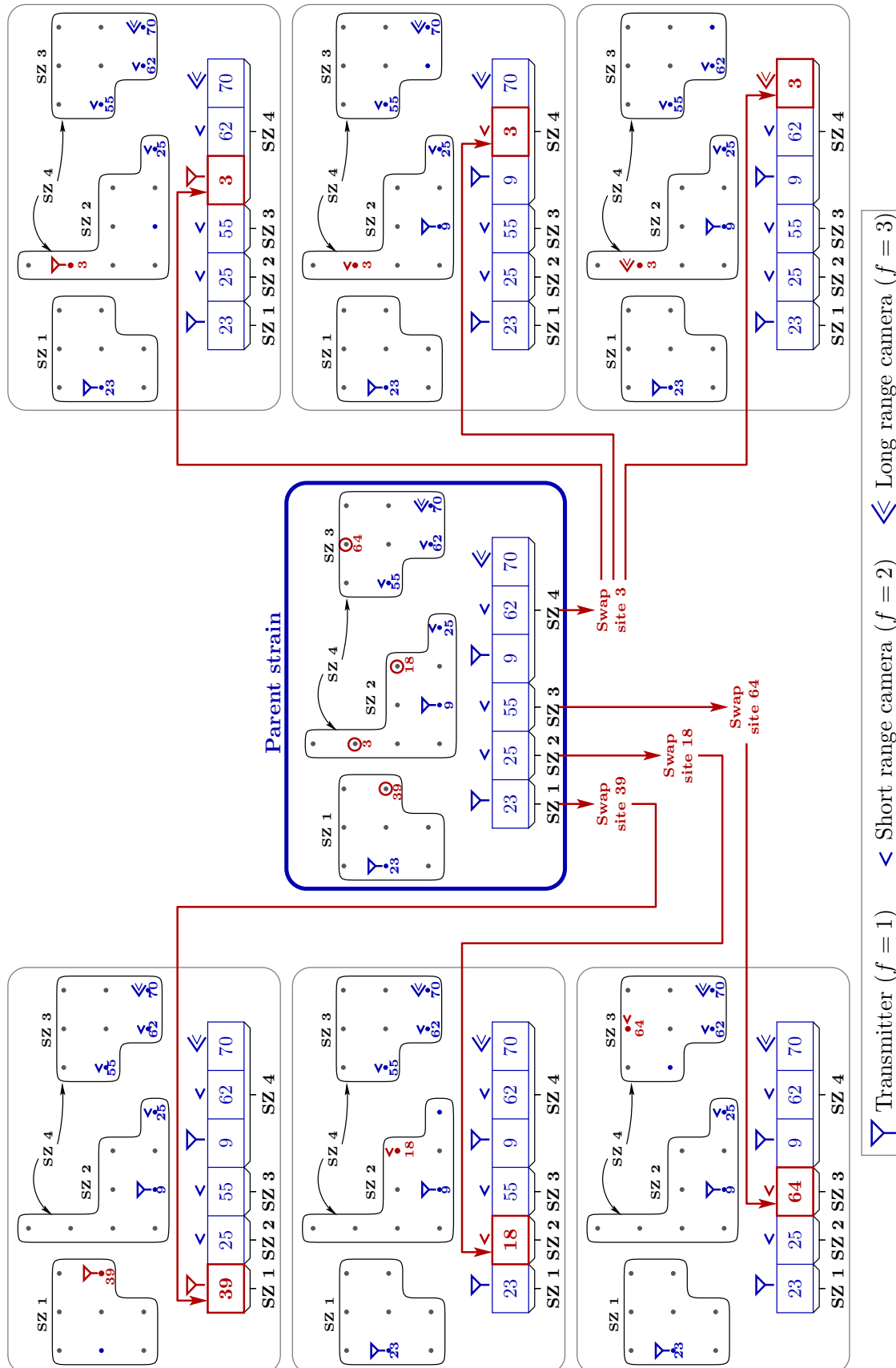


FIGURE 5.8: AMOEBA mutation performed on the parent solution in Example 5.3.

the rules of MOO solution dominance discussed in §5.1.1, the accepted offspring strains are considered to be at least as good as their parent solution and some may even dominate it.

3. **Strict elitist survival:** Offspring strains are a result of mutation performed upon all strains in the current infection. The above considerations ensure that only feasible strains and strains that are at least as good as their parent strains are added to the infection. However, an offspring strain of one parent strain may not necessarily be as good as another parent strain or its offspring. Similarly, some, or all, of one parent strain's offspring may dominate another parent strain or its offspring. In order to limit the size of the infection based upon the comparative quality of the strains, a very strict elitist strategy is followed. The parent and offspring strains are sorted by the FNSEA, after which *only* the front of non-dominated solutions proceed to the next infection stage — *i.e.* the infection spreads via non-dominated solutions of successive iterations only. As mentioned in §5.1.3, the FNSEA has an $O(MN^2)$ run time, where M denotes the number of objectives and N denotes the population size, respectively. Since only the first non-dominated front is desired by AMOEBA, the run time of the FNSEA is reduced (potentially significantly) since only the first front needs to be ranked and the ranking of successive fronts that normally follows is skipped.
4. **Limited infection size:** Although the size of the infection is managed to some extent by allowing only the non-dominated solutions of successive infections to proceed, the infection may nevertheless grow rapidly. This is because the number of solutions in the true Pareto-front may be very large and any good approximation of this front may therefore also grow to a large size³. This is likely when the solution space is large or when a large number of objective functions are present— large solution spaces are, in fact, typically observed with an increase in the number of objectives, when the SZs are large, or when a large number of facilities are to be placed. In order to address this problem, a *pruning* mechanism is employed to limit the infection size. This is achieved by imposing a *maximum infection size*; if the most recent approximate Pareto-front is larger than this size, a number of strains that is equal to the maximum infection size is selected from the front in descending order of crowding distance in objective space, ensuring solution diversity and even distribution in objective function space in the process.

An illustration of the first three stages of an infection spread, following the above infection control techniques, is provided in Figure 5.9. The infection process is repeated until a pre-specified termination condition, such as those discussed in §5.1.5, is met. In this dissertation, the AMOEBA stopping criterion is based on the average quality of the approximately Pareto-optimal solutions per infection stage. Convergence is considered to have occurred when these averages improve by less than an *improvement threshold percentage* t_p over a fixed number of *improvement threshold stages* t_s . This evaluation is performed with respect to the average of each objective function value over all approximately Pareto-optimal solutions. The average values with respect to *all* the objectives are required to improve by less than the improvement threshold percentage over the number of improvement threshold generations — these parameters are selected and motivated for each specific problem in later examples.

This type of convergence evaluation is implemented instead of the Pareto similarity thresholds employed by the NSGA-II due to unique characteristics exhibited by AMOEBA as it approaches

³During the development process of AMOEBA, population sizes of between 1000 and 2000 were regularly observed for large problems — sometimes even larger than 2000. Such large population sizes are prohibitive for problems with intensive, time-consuming file read/write requirements.

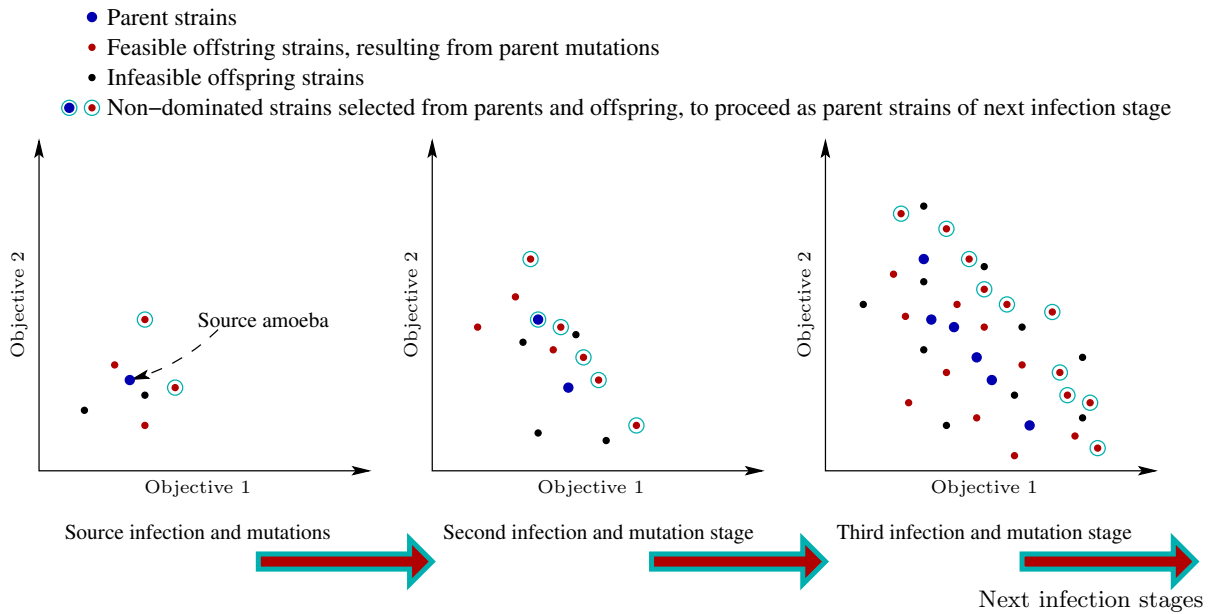


FIGURE 5.9: *The first three stages of an AMOEBA infection spread.*

convergence. Whereas the solutions in the NSGA-II move towards a static non-dominated front of increasingly similar solutions with similar objective function values, the non-dominated front achieved by AMOEBA moves towards a non-dominated front that is static in its relative objective function values only, but may yet continue to exhibit constant changes in the actual solutions that are included in the front. This is due to the algorithm reaching convergence as a whole — thereby reaching a boundary in the progression that is achieved by the front — while the mutation procedure generates many offspring strains that may result in significant numbers of new non-dominated solutions, even at the time of algorithm convergence. These solutions may exhibit only very slight, negligible improvements over their source strains, or merely fill ‘gaps’ in the spread of the Pareto-front and, when evaluated according to average objective function values or hypervolume measures, may result in values that are practically identical⁴. While algorithm convergence may then be considered to have occurred in terms of the approximate Pareto-front’s average objective function values, the front may, in the case that a solution-based Pareto similarity measure is employed, be considered to be far from convergence and may continue to run for a significant time longer without producing any practical improvements in the approximation front’s general quality. For these reasons the objective function value-based Pareto similarity measure is employed by AMOEBA.

A pseudocode description of the working of AMOEBA appears in Algorithm 5.4.

5.4 Worked example of the GFLo solution process

In this section, the working of the NSGA-II and AMOEBA is illustrated by means of an example. The purpose is to provide preliminary validation that the algorithms are suitable for solving GFLoPs, and to provide some indication of the performance of the algorithms.

⁴Comparing hypervolumes of successive generations is, in fact, another approach towards determining population convergence [57, 58], although significantly more computationally expensive to determine than the simple average value measure.

Algorithm 5.4: Asexual multi-objective evolutionary-based algorithm (AMOEBA)

Input : A MOO problem where solutions are an assignment of values to a vector of decision variables \mathbf{x} , a maximum infection size N_{\max} , a set of constraints, M objective functions, and a boolean algorithm termination variable T .

Output: An approximation of the Pareto-optimal solution set in multi-objective space, \mathbf{A}^* .

```

1 Randomly generate an initial infection of solutions  $\mathbf{I}_0$  of size 1 (source amoeba).
2 Create offspring infection  $\mathbf{Q}_0$  using AMOEBA mutation on source amoeba.
3  $t \leftarrow -1$ ;
4 while  $T = 0$  do
5    $t \leftarrow t + 1$ ;
6    $\mathbf{R}_t \leftarrow \mathbf{I}_t \cup \mathbf{Q}_t$ ;
7   Partition  $\mathbf{R}_t$  into front  $\mathcal{F}_1$  by means of the FNSA [Algorithm 5.2].
8    $\mathbf{I}_{t+1} \leftarrow \emptyset$ 
9   if  $|\mathcal{F}_1| \leq N_{\max}$  then
10     $\mathbf{I}_{t+1} \leftarrow \mathcal{F}_1$ ;
11  else if  $|\mathcal{F}_1| > N_{\max}$  then
12    Calculate crowding distance for all solutions in  $\mathcal{F}_1$  [Algorithm 5.1].
13    Sort  $\mathcal{F}_1$  members in order of decreasing crowding distance.
14     $\mathbf{I}_{t+1} \leftarrow \{\text{the first } N_{\max} \text{ elements of } \mathcal{F}_1\}$ ;
15  end
16  Create  $\mathbf{Q}_{t+1}$  using AMOEBA mutation on all strains in  $\mathbf{I}_{t+1}$ .
17 end
18  $\mathbf{A}^* \leftarrow \mathbf{I}_t$ ;

```

5.4.1 Terrain description

A section of terrain exhibiting gentle hills and multiple peaks was chosen in an area of the South African Western Cape, surrounding the town of Malmesbury. The terrain is square and measures 10 kilometres south to north and 10 kilometres west to east. Figure 5.10 (a) contains a relief representation of the terrain elevation. The highest available resolution of terrain data, namely 1 arc second or approximately a 28 metre spacing between gridposts, was obtained, resulting in a total of 125 125 gridposts over the entire terrain surface area.

5.4.2 Facilities and zones

In this example, facilities of a single type are considered. Six facilities are to be placed in two SZs, as indicated in Figure 5.10 (a). SZ 1 is located in the centre of the terrain area, while SZ 2 surrounds SZ 1, and the SZs are disjoint. Three facilities are to be placed in each SZ. The number of gridposts in these SZs are 30 400 in SZ 1 and 94 725 in SZ 2. The facilities are solar-powered and hence the desirability of their placement is based on average daily solar irradiation values.

5.4.3 Objectives and constraints

In this example, two criteria are considered. The first criterion is geospatial location-specific and involves average daily solar irradiation values, while the second is geospatial inter-site and is formulated in terms of inter-facility distance. Average daily solar radiation values for the terrain

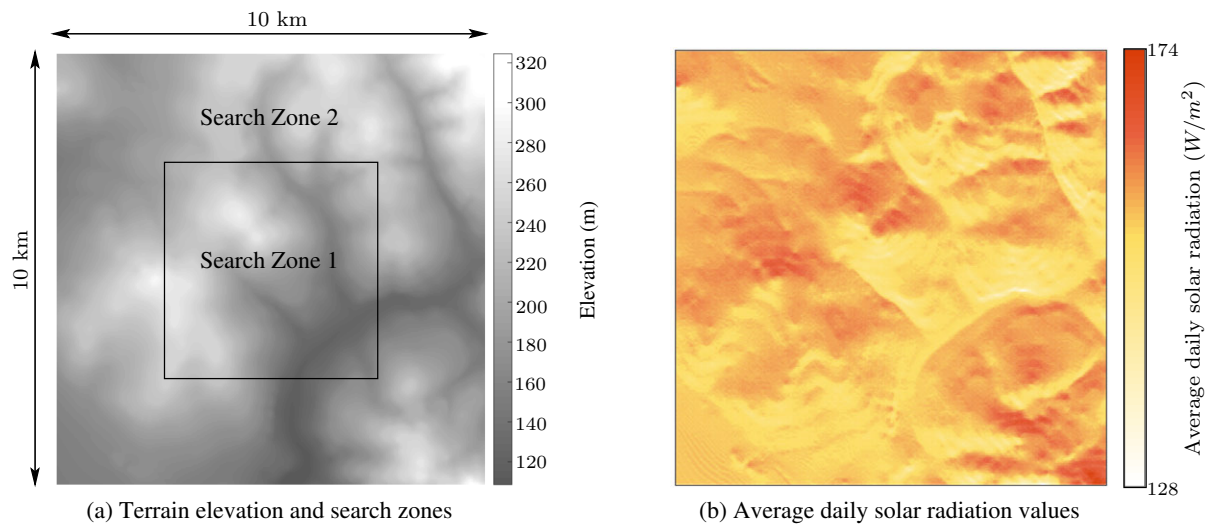


FIGURE 5.10: (a) Terrain and SZs considered and (b) average daily solar radiation in the worked example.

are provided in Figure 5.10 (b). These values were determined by use of the ArcGIS software suite. The problem is solved according to MaxiMin and dispersion objectives.

The objectives are to:

1. **Maximise** the minimum average daily solar radiation value of the six facilities (MaxiMin), and to
2. **Maximise** the minimum distance measured between facilities in SZ 1 and SZ 2 (dispersion).

Furthermore, a lower bound of 500 metres is enforced on the distance between any facility pair, in addition to the obligatory placement constraints (3.4)–(3.6). The constraints are therefore that:

1. exactly six facilities should be placed,
2. facilities should be placed according to their facility-SZ assignments,
3. at most one facility is allowed per gridpost, and
4. no facility may be within 500 metres of another.

5.4.4 Algorithm parameter selection

For the implementation of the NSGA-II in the context of the above example, separate sensitivity analyses were performed with the aim of finding parameter combinations that result in Pareto-front approximation sets containing solutions which are consistent in terms of returning good objective function values and an even spread of solutions along the non-dominated front. As previously discussed in §5.3.1, the NSGA-II parameters that are often the most influential in this respect are the population size and the crossover and mutation probabilities.

A suitable combination of NSGA-II parameters for this specific example was found to be a population size of 200, a crossover probability of 0.8 — employing randomly selected two-point

crossover — and a mutation probability of 0.2. In terms of algorithm convergence (as discussed in §5.3.1), a good combination of Pareto similarity threshold percentage and number of generations was found to be 90% over ten generations.

Due to the simplicity of AMOEBA, its parameter selection process was significantly simpler. This required the specification of an infection size limit and convergence parameters only. The infection size limit was set to 100 — although this limit was not reached during any of the runs. During convergence parameter evaluation it was found that a good combination of improvement threshold percentage and number of generations (see §5.3.2) was found to be 2% over ten generations.

5.4.5 Mathematical model formulation

The mathematical model formulation follows the generic framework proposed in Figure 4.4 of §4.4. There are a total number of 125 125 gridposts in the RG (which envelops the two SZs); therefore $\mathbb{G} = \{1, \dots, 125\ 125\}$. The facility and criteria sets are

$$\mathbb{F} = \{1[\text{Solar-powered facility}]\}$$

and

$$\mathbb{C} = \{1[\text{Average daily solar radiation}], 2[\text{Inter-facility distance}]\}.$$

The gridposts in SZs \mathbb{S}_1 and \mathbb{S}_2 are all those enveloped by the zones indicated in Figure 5.10, while the facility-SZ assignment matrix is given by

$$\mathbf{F} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}.$$

In the first objective, the minimum average daily solar irradiation value over all facilities is to be maximised. The set-pair p_1 for the first objective therefore corresponds to the facility and SZ sets $\mathcal{F}_1 = \{1, 1\}$ and $\mathcal{S}_1 = \{1, 2\}$. In the second objective the facilities are dispersed according to the SZs in which they are placed. The set-pairs p_2 and p_2^* for the second objective correspond to the facility and SZ sets $\mathcal{F}_2 = \{1\}$ and $\mathcal{S}_2 = \{1\}$ and $\mathcal{F}_{2^*} = \{1\}$ and $\mathcal{S}_{2^*} = \{2\}$, respectively. Finally, since there is only one type of facility, the objective-specific type set is simply \mathbb{F} , which serves both objectives.

The first objective, according to (4.2), is to

$$\text{maximise } V_1 \tag{5.1}$$

subject to (4.3a); that is

$$V_1 \leq v_{c_1}(\ell) + M(1 - \tilde{x}_{p_1}(\ell)) \tag{5.2}$$

for all ordered pairs $f_{p_1} \in \mathcal{F}_{p_1}$ and $s_{p_1} \in \mathcal{S}_{p_1}$, and for all $\ell \in \mathbb{S}_{s_{p_1}}$.

The second objective, according to (4.15), is to

$$\text{maximise } V_2 \tag{5.3}$$

subject to (4.16a); that is

$$V_2 \leq v_{c_2}(\ell, \ell^*) + M(2 - \tilde{x}_{p_2}(\ell) - \tilde{x}_{p_2^*}(\ell^*)) \tag{5.4}$$

for all ordered pairs of facilities and SZs $f_{p_2} \in \mathcal{F}_{p_2}$ and $s_{p_2} \in \mathcal{S}_{p_2}$, all $f_{p_2^*} \in \mathcal{F}_{p_2^*}$ and $s_{p_2^*} \in \mathcal{S}_{p_2^*}$, and for all $\ell \in \mathbb{S}_{s_{p_2}}$ and $\ell^* \in \mathbb{S}_{s_{p_2^*}}$.

The set-pair for the constraint is p_1 , and the constraint is enforced between all the facilities in the same set-pair, therefore $p_1^* = p_1$. The lower bound constraint is $c_{\min}^2 = 500$ metres. From (4.21), the constraint is therefore

$$500 \leq v_{c_2}(\ell, \ell^*) + M(2 - \tilde{x}_{p_1}(\ell) - \tilde{x}_{p_1^*}(\ell^*)) \quad (5.5)$$

for all ordered pairs of facilities and SZs $f_{p_1} \in \mathcal{F}_{p_1}$ and $s_{p_1} \in \mathcal{S}_{p_1}$, all $f_{p_1^*} \in \mathcal{F}_{p_1^*}$ and $s_{p_1^*} \in \mathcal{S}_{p_1^*}$, and for all $\ell \in \mathbb{S}_{s_{p_1}}$ and $\ell^* \in \mathbb{S}_{s_{p_1^*}}$. The formulation is completed with the inclusion of the mandatory constraints (3.4)–(3.6).

5.4.6 Algorithm results comparison

This section contains a summary of the results obtained from *thirty* separate runs of each of the NSGA-II and AMOEBA in the context of the example described in §5.4.1–5.4.5. The results of the respective algorithms are compared to each other with respect to approximate Pareto-front quality, computational performance, and the facility site locations of non-dominated solutions.

Solution quality

The thirty approximated fronts and the attainment fronts obtained after thirty runs of the NSGA-II and AMOEBA are shown in objective function space in Figures 5.11 (a) and (b), respectively. From these results it is clear that the AMOEBA results are superior to those returned by the NSGA-II. As may be seen in Figure 5.11 (b), the NSGA-II attainment front is strictly dominated⁵ by that of AMOEBA (all the NSGA-II solutions are dominated by at least one AMOEBA solution).

Values of the hypervolume measure for the final attainment fronts obtained by the thirty separate runs of the respective algorithms are shown in Table 5.1 and the normalised hypervolume values per run are provided in Figure 5.11 (c). The reference point with respect to which the hypervolumes were calculated was located at the minimum values obtained with respect to both objectives over all runs of both algorithms. This reference point was at 147.5 W/m² for objective 1 and 1 213 metres for objective 2 — near the origin in Figure 5.11 (a). All hypervolume measures are normalised by the AMOEBA attainment hypervolume, since it was the largest.

	Mean (Std dev)	Minimum	Maximum	Attainment
NSGA-II	0.818 (0.072)	0.666	0.972	1
AMOEBA	0.794 (0.087)	0.529	0.957	0.996

TABLE 5.1: Normalised hypervolume values over thirty runs each of the NSGA-II and AMOEBA for the example of §5.4.1–5.4.5. The values are normalised by the largest attainment hypervolume obtained, which was that of AMOEBA.

It may be seen from the normalised hypervolume indicators in Table 5.1 and Figure 5.11 (c) that AMOEBA consistently obtains larger hypervolumes than the NSGA-II. Furthermore, AMOEBA obtains the largest mean hypervolume value, albeit with a slightly larger standard deviation.

⁵One approximate Pareto-set A is said to strictly dominate another set B ($A \prec B$) if for each $y \in B$, there exists an $x \in A$, such that $x \prec y$.

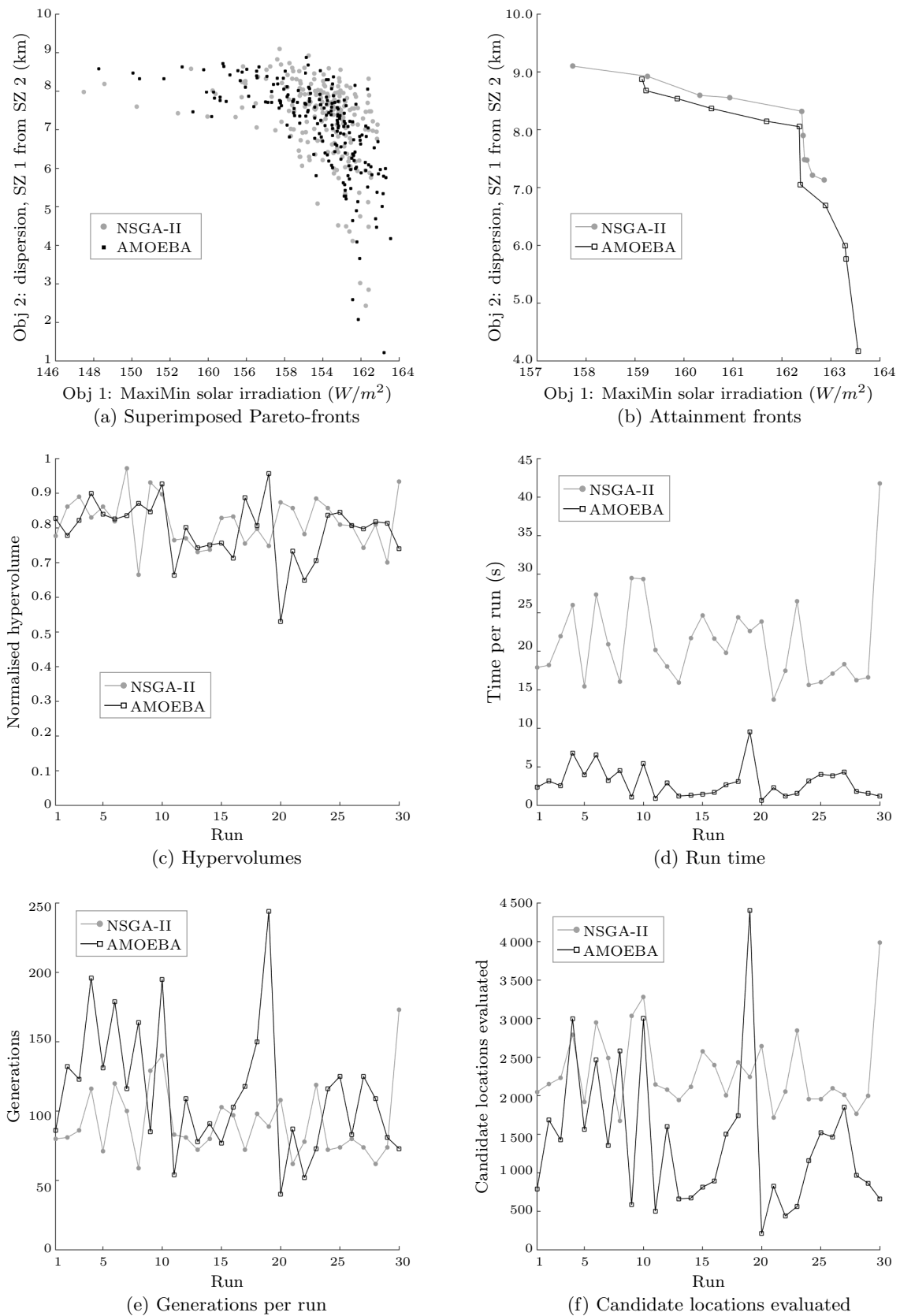


FIGURE 5.11: Results obtained for the example of §5.4.1–5.4.5 by thirty runs each of the NSGA-II and AMOEBA.

	Run time (s)	Generations	Sites evaluated
NSGA-II	21.2	88	2 318
AMOEBa	3.0	110	1 391

TABLE 5.2: Average values per run for thirty runs each of the NSGA-II and AMOEBA obtained for the example of §5.4.1–5.4.5.

Computational performance

Figure 5.11 (d) contains the mean run time per run over the thirty runs of the respective algorithms. It may be observed that AMOEBA is significantly faster than the NSGA-II.

When comparing the average number of generations per run in Figure 5.11 (e), it may be observed that the NSGA-II requires much fewer generations before termination occurs than is the case for AMOEBA. This is a result of the NSGA-II initialising with a relatively large population of solutions, therefore covering a reasonably large area in objective function space from which resulting populations are created, compared to the single solutions at which every run of AMOEBA starts — necessitating a number of early AMOEBA iterations to produce populations of effective size from which the algorithm may proceed to find promising solutions. Although AMOEBA requires considerably more generations before termination than the NSGA-II, it terminates within shorter run times. A reason for this is the smaller initial population sizes associated with AMOEBA, in addition to AMOEBA swap procedure being considerably simpler (in theory and computational complexity) than the parent selection processes and the crossover and mutation operators of the NSGA-II.

When considering the average number of site locations evaluated per run, Figure 5.11 (f) suggests that AMOEBA exhibits superior search efficiency compared to the NSGA-II. This is because although AMOEBA typically evaluates significantly fewer candidate sites than the NSGA-II, it nevertheless consistently returns solutions of similar quality. It is acknowledged that this may, however, simply be the case for the specific example solved.

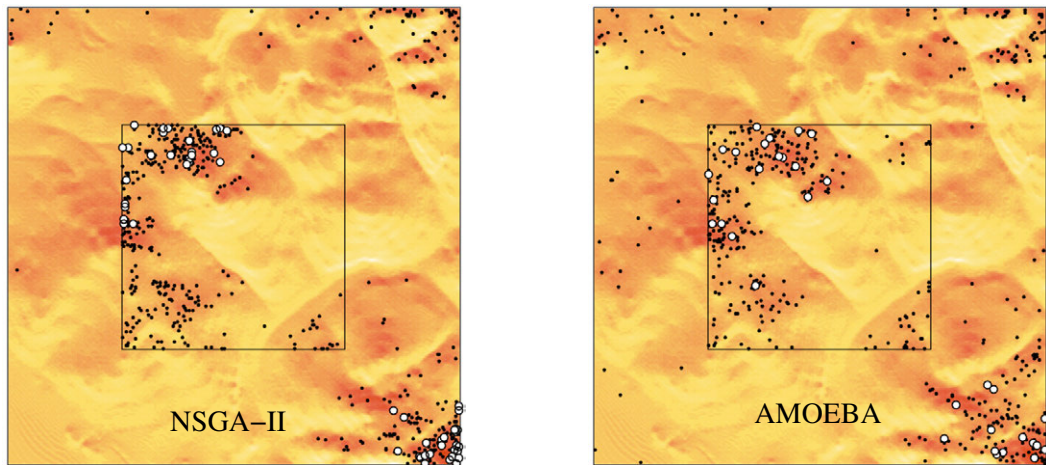
Facility site locations of non-dominated solutions

The locations of the constituent sites of all the solutions uncovered in the thirty separate approximation sets of Figure 5.11 (b) are shown in Figure 5.12 (a) (indicated here by the black markers), in addition to the sites of the solutions in the attainment fronts of Figure 5.11 (b) (the white markers). It is interesting to note that the sites in Figure 5.12 (a) are very similar for both algorithms.

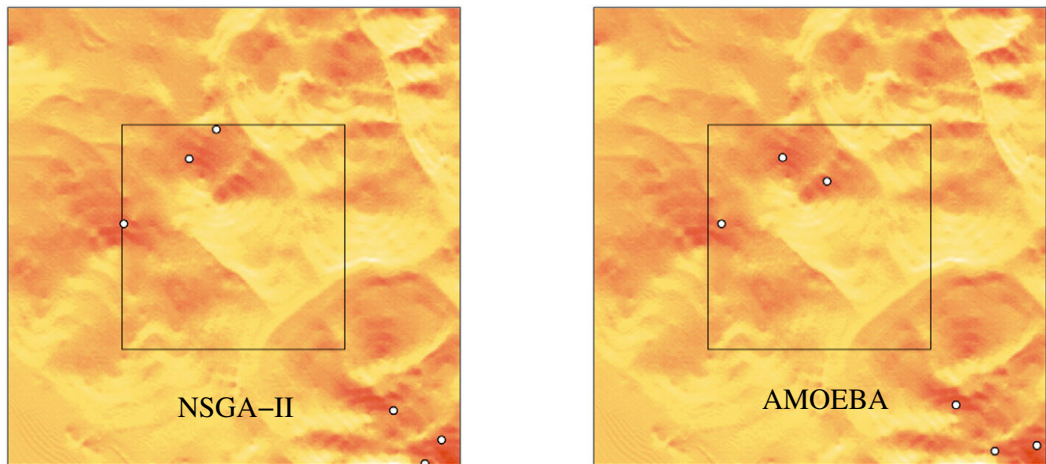
Figure 5.12 (b) contains the physical site locations for the solutions in the respective attainment fronts of the algorithms of Figure 5.11 (b) corresponding to the highest value with respect to the solar irradiation objective. Furthermore, Figure 5.12 (c) contains the physical site locations for the solutions in Figure 5.11 (b) corresponding to the highest value with respect to the distance objective. These figures encapsulate the type of information that would typically be provided to decision makers in order to facilitate a final decision with respect to the placement of facilities.

5.5 A preliminary appraisal of the algorithms

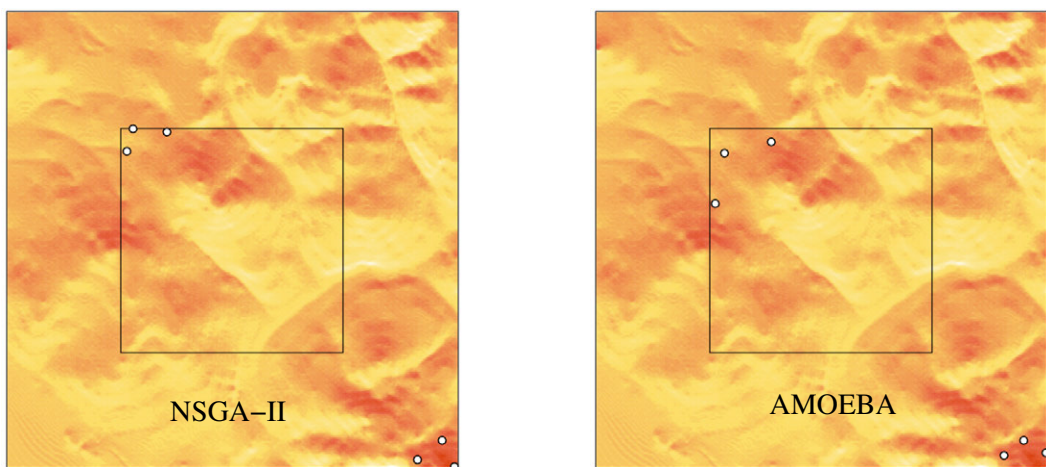
It was shown in §5.4 that thirty runs of AMOEBA consistently exhibited similar performance in terms of average and attainment hypervolumes compared to the same number of runs of the



(a) Physical site locations of solutions in approximated Pareto-fronts of Figure 5.11 (a) (the black markers) and physical site locations of attainment fronts of Figure 5.11 (b) (the white markers)



(b) Physical site locations of solutions in attainment fronts of Figure 5.11 (b) that return the highest values with respect to the MaxiMin solar irradiation objective



(c) Physical site locations of solutions in attainment fronts of Figure 5.11 (b) that return the highest values with respect to the dispersion distance objective

FIGURE 5.12: (a) Physical site location of the solutions in the attainment fronts of Figure 5.11 (b), and the locations of sites in the solutions that return the highest values with respect to (b) the MaxiMin solar irradiation and (c) the dispersion objectives.

NSGA-II for a specific GFLo problem instance, within substantially shorter computation times. The reason for this is AMOEBA's efficient exploratory properties due to its random, strictly controlled swap procedure. This may, however, become a weakness as the number of facilities to be placed increases in the sense that the efficiency of the combinatorial explorative strength of the NSGA-II could outperform AMOEBA in terms of computation time and solution quality.

When comparing the NSGA-II and AMOEBA purely in terms of computational complexity, AMOEBA is a much simpler algorithm — both in terms of its search processes and in terms of its ease of implementation. In terms of parameter specification, AMOEBA also only requires the specification of one parameter, compared to the multiple parameters that have to be specified for the NSGA-II. The choice between which algorithm to use in real-life scenarios of larger sizes may depend on the expected computation times of the algorithms and requires thorough testing. More scenarios of varying complexity and purpose are solved in the following chapters of this dissertation in order to provide a preliminary analysis in this regard.

5.6 Chapter summary

Two MOO algorithms were reviewed and developed for solving GFLoP instances in this chapter. The chapter opened in §5.1 with an introduction to MOO concepts that are of importance with respect to these algorithms. These were the notion of Pareto-optimality, diversity preservation, solution ranking, constraint handling, algorithm termination and solution quality assessment.

The first algorithm considered in this chapter, the NSGA-II, was developed in §5.3.1 and its crossover and mutation operations were explained by means of a graphical example. The second algorithm, AMOEBA, is a novel algorithm that was developed specifically for the purposes of GFLo in §5.3.2 and its viral-like mutation process was illustrated by means of a graphical example.

Implementation of the algorithms was performed by means of a realistic GFLoP instance in §5.4. Over thirty runs of each of the algorithms, AMOEBA returned solutions of slightly weaker quality than the NSGA-II, but within substantially shorter run times. Computational results and physical site locations of non-dominated solutions were also discussed, accompanied by illustrative figures.

CHAPTER 6

GFLo-specific search improvements

Contents

6.1	A hybrid geospatial algorithm	83
6.1.1	<i>Algorithm description</i>	84
6.1.2	<i>Implementation and results</i>	85
6.2	Halving NSGA-II population size	87
6.3	A multi-resolution approach	91
6.3.1	<i>Motivation</i>	91
6.3.2	<i>Method</i>	94
6.3.3	<i>Worked example</i>	94
6.3.4	<i>General observations and implementation notes</i>	99
6.3.5	<i>Important MRA considerations</i>	100
6.4	Proximity-dependent cluster removal	101
6.5	Chapter summary	102

Two algorithms that may be used to solve GFLoPs were described in the previous chapter and were demonstrated in the context of a simple GFLoP involving two objectives. In this chapter, a hybridised version of these algorithms is proposed which exploits certain characteristics of these algorithms and their behaviour with respect to the geospatial environment. A population convergence-based method which results in significant reductions in the run times of the NSGA-II is introduced in §6.2. A multi-resolution approach that aims to concentrate the search process towards promising areas, resulting in superior solution quality for all algorithms, is finally proposed in §6.3. Post-optimisation approximation set simplification considerations are discussed in §6.4.

6.1 A hybrid geospatial algorithm

Observations made with respect to the results returned by the NSGA-II and AMOEBA algorithms for the problem instance of §5.4 reveal advantageous properties of each algorithm that may be exploited in the design of a hybrid algorithm.

In the example of §5.4, AMOEBA outperformed the NSGA-II in terms of computation time and returned solutions of similar quality, despite the fact that it consistently evaluated fewer candidate locations during its solution process. This indicates that AMOEBA exhibits strong

and efficient explorative properties. The smaller number of candidate locations considered stems from the fact that AMOEBA initiates with a single solution only, after which strict control over the spread of the non-dominated front is enforced. Under these conditions, however, AMOEBA may perform poorly in large decision spaces when the chromosome is longer (more facilities). The NSGA-II, on the other hand, required the evaluation of more candidate sites than AMOEBA to arrive at its solutions in the example of §5.4. This is a result of its traditional population-based approach which requires a larger number of sites to be included in its starting population.

Both algorithms perform well compared to each other, despite the fact that their search processes are balanced towards either of the basic concepts of combinatorial searches — the NSGA-II exhibits strong combinatorial efficiency as a result of its crossover operations, whereas AMOEBA is essentially a mutation-based algorithm. These observations and the comparative performance of the algorithms were general occurrences observed during multiple runs of the algorithms to solve GFLoPs of varying complexity, and not only the single example in §5.4¹. Combining the mechanisms of these algorithms in order to exploit their strengths presents an interesting alternative to choosing between the NSGA-II and AMOEBA. Such a hybridisation is the topic of this section.

6.1.1 Algorithm description

The solution process of a newly proposed algorithmic hybridisation between the NSGA-II and AMOEBA, called the *hybrid geospatial algorithm* (HGA), is illustrated graphically in Figure 6.1. An initial population of candidate solutions of size N is randomly generated. Following this, the mating procedure of the NSGA-II is followed to generate an offspring generation of solutions. This process entails the selection of solutions into a mating pool, followed by parent selection via tournament selection, and crossover. The offspring generation process followed here differs from the NSGA-II in the sense that the solutions resulting from crossover do not undergo the typical NSGA-II mutation process, but instead proceed directly to the offspring generation. The parent and offspring generations are combined and the N best solutions are selected by means of the FNSA, according to a combination of solution rank and crowding distance. This new population now undergoes AMOEBA mutation for *all* the solutions in the population, as opposed to the typical AMOEBA process of mutating non-dominated solutions only. The population of offspring solutions are combined with its parent population and the FNSA is once again employed to select the N best solutions from this combined population. This process is repeated until a pre-specified termination criterion is met, which, for the purposes of this dissertation, is the same criterion used for AMOEBA.

Due to the strong explorative properties of AMOEBA despite its small population sizes, the population size of the HGA may be significantly smaller than those typically employed by the NSGA-II. The population benefits from both the combinatorial strength of the NSGA-II and the explorative strength of AMOEBA, and so good results may be expected despite the small population size. In addition, the smaller population size is expected to result in faster run times, since fewer crossover operations are required than in a typical NSGA-II implementation, and also because the FNSA executes faster with smaller population sizes (a principal reason for AMOEBA's fast run times).

¹The example does, however, provide a good illustration of the results that were typically returned over a variety of GFLoPs solved by the two algorithms.

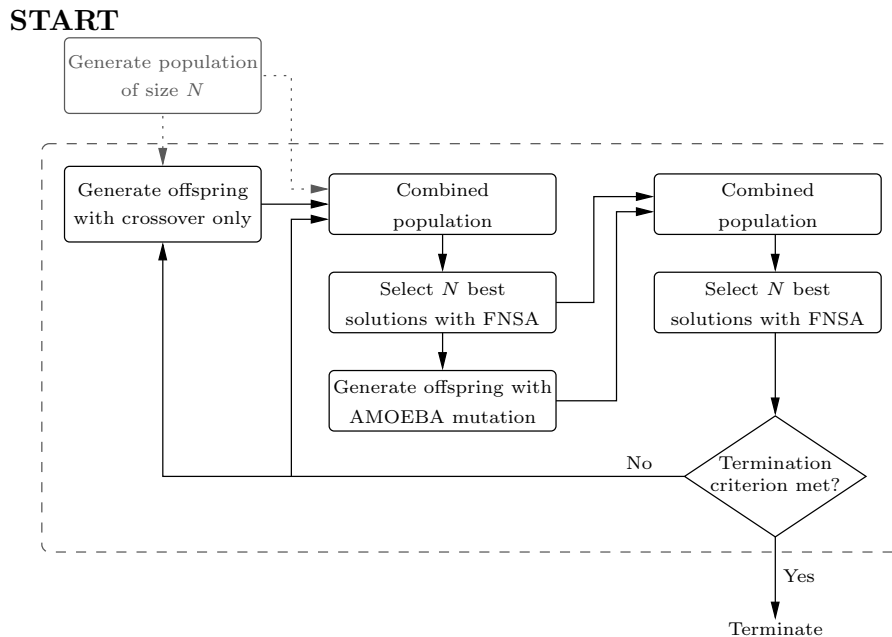


FIGURE 6.1: A flowchart of the HGA solution process.

6.1.2 Implementation and results

The HGA was used to solve the same problem instance as in the example of §5.4. The same parameter set used by the NSGA-II was adopted during the crossover stage, while the same convergence parameters of the AMOEBA implementation were used to signal algorithm termination. A small population of twenty candidate solutions was used.

A comparison of the first two algorithms' results and the results of thirty runs of the HGA is presented in Figure 6.2. The attainment fronts in Figure 6.2 (a) indicate a good level of attainment achieved by the HGA, with a uniform spread and good solution diversity along the Pareto approximation set.

Values of the hypervolume measure for the final attainment fronts obtained by the thirty separate runs of the respective algorithms are shown in Table 6.1 and the normalised hypervolume values per run are provided in Figure 6.2 (b). All hypervolume measures are normalised by the hybrid attainment hypervolume, since it was the largest.

	Mean	Minimum	Maximum	Attainment
NSGA-II	0.815	0.671	0.962	0.986
AMOEBEA	0.792	0.537	0.943	0.979
HGA	0.869	0.738	0.987	1

TABLE 6.1: Normalised hypervolume values for thirty runs each of the HGA, NSGA-II and AMOEBA obtained in the context of the problem instance of §5.4.1–5.4.5. The values are normalised according to the HGA attainment hypervolume obtained.

It may be seen from the normalised hypervolume indicators in Table 6.2 and Figure 6.2 (b) that the HGA consistently achieves larger hypervolumes than the NSGA-II and AMOEBA. Furthermore, the HGA obtains the largest mean hypervolume value, with an impressive minimum hypervolume value when compared to those of the other algorithms — indicating that its approximation fronts are consistently of superior quality. An interesting observation is that the

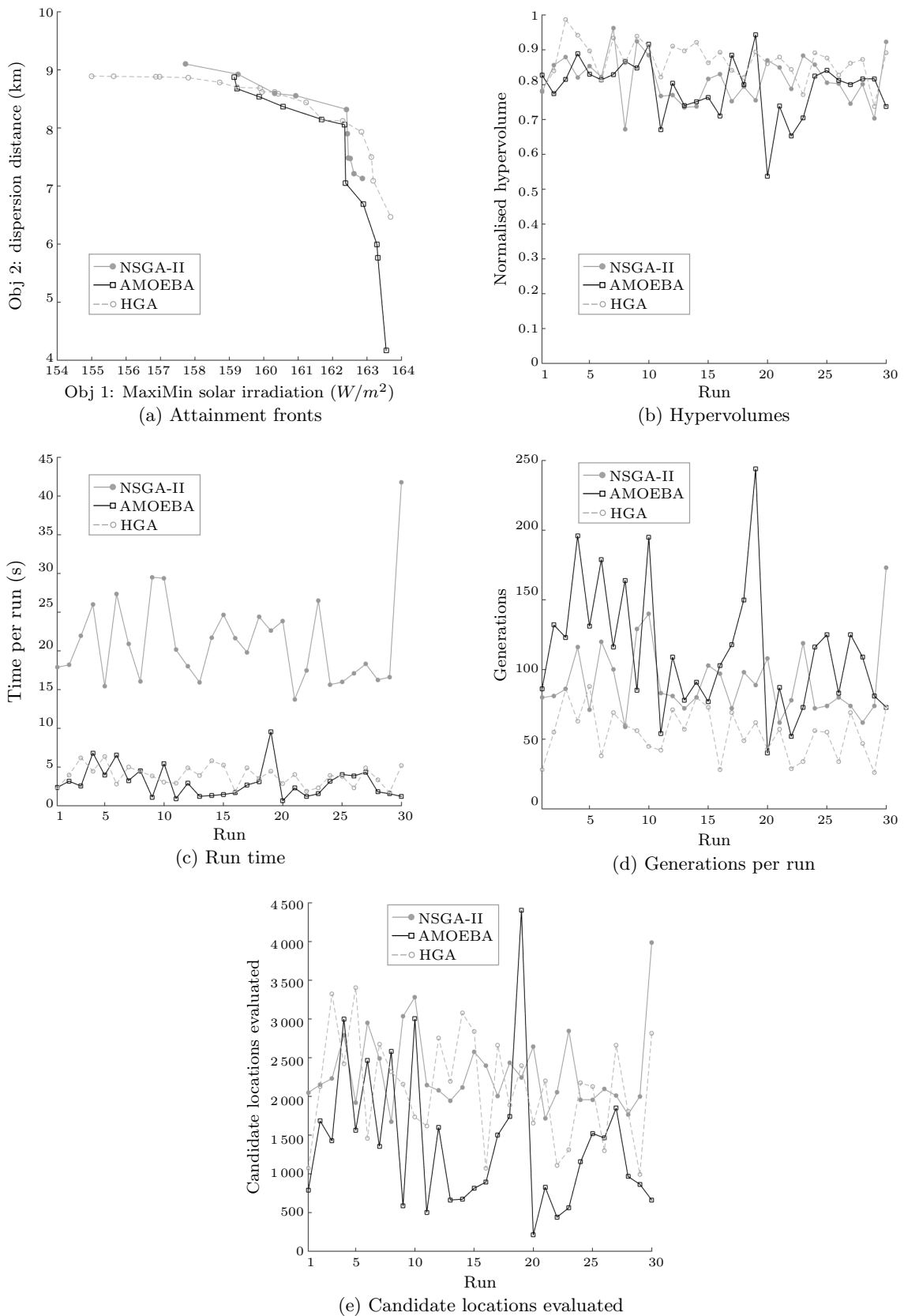


FIGURE 6.2: Results obtained by thirty runs each of the NSGA-II, AMOEBA and the HGA in the context of the problem instance of §5.4.1–5.4.5.

	Run time (s)	Generations	Sites evaluated
NSGA-II	21.2	88	2 318
AMOEBa	3.0	110	1 391
HGA	3.9	58	2 111

TABLE 6.2: Average values per run for thirty runs each of the HGA, NSGA-II and AMOEBA in the context of the problem instance of §5.4.1–5.4.5

maximum hypervolume value achieved by a single run of the HGA is larger than the hypervolumes of the attainment fronts achieved by the NSGA-II and AMOEBA.

From Table 6.2 and Figure 6.2 (c) it may be seen that the HGA is not much slower than AMOEBA, and significantly faster than the NSGA-II. A comparison of the number of generations required per run is also provided in the table and in Figure 6.2 (d). The HGA is observed to terminate within a number of generations that is generally smaller than the number of generations of the NSGA-II and AMOEBA. These values for the HGA are, however, not as simple to interpret as those of the NSGA-II and AMOEBA, since each generation includes a crossover-FNSA stage and mutation-FNSA stage — *i.e.* two offspring generations per generation.

When considering the average number of site locations included in the feasible solutions evaluated per run, Table 6.2 and Figure 6.2 (e) reveal that the HGA considers a number of candidate locations that is slightly less than the NSGA-II and considerably more than AMOEBA. The superior solution quality of the HGA solutions may be attributed to this observation, since a larger portion of the solution space is explored — more efficiently. The HGA exhibits fast run times despite the large number of candidate locations evaluated, which may be attributed to its small population size.

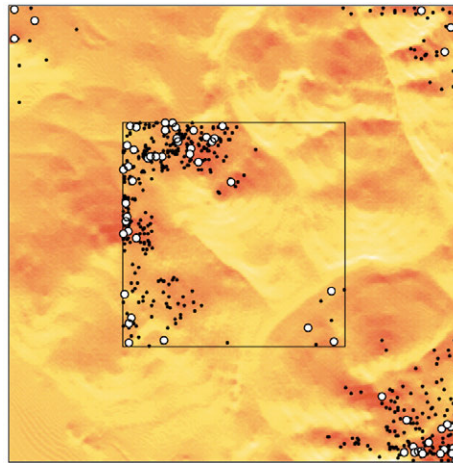
Figure 6.3 (a) shows the locations of the constituent sites of all of the solutions in the thirty separate approximation fronts of the HGA (indicated by the black markers), in addition to the sites of the solutions in the HGA attainment front in Figure 6.2 (a) (the white markers). The solutions in this figure are similarly located to those of the NSGA-II and AMOEBA in Figure 5.12 (a) and (b), although the HGA sites seem more clearly grouped together which may point to more consistent convergence across its thirty separate runs (in terms of the site locations included in the approximation fronts at algorithm termination).

Figure 6.3 (b) contains the physical site locations for the solution in the HGA attainment front that returns the highest value with respect to the MaxiMin solar irradiation objective. Figure 6.3 (c), on the other hand, contains the physical site locations for the solution in Figure 6.2 (b) that returns the highest value with respect to the dispersion distance objective.

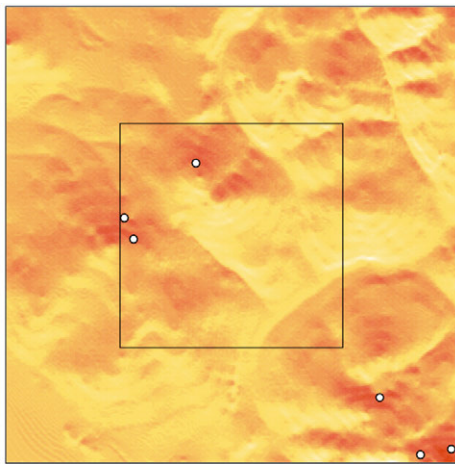
From the observed solution quality and run times of the HGA, it may be concluded that the HGA may be a valuable addition to the clan of algorithms that may be considered for solving GFLoPs.

6.2 Halving NSGA-II population size

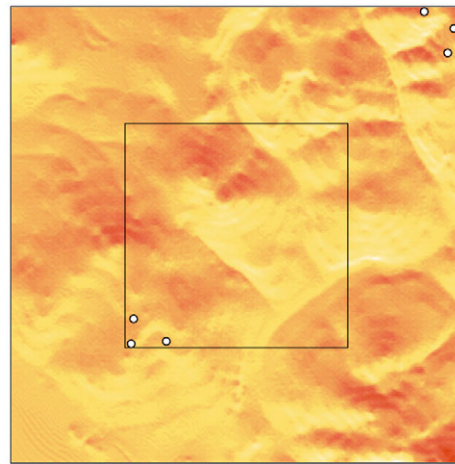
The search for approximately Pareto-optimal solutions should be considered a dynamic process. This is because the search progresses from an initial state of random solution generation with minimal knowledge of the decision space, towards a state where the search has uncovered information about this space that may be exploited. In general, genetic algorithms start with the generation of a random initial population of a specific size that is expected to provide (at least)



(a) Site locations in solutions of the HGA attainment front



(b) Maximum MaxiMin solar sites



(c) Maximum distance dispersion sites

FIGURE 6.3: (a) Physical site locations of the solutions in the HGA attainment front of Figure 6.2 (a), and the locations of sites in the solutions that return the largest values with respect to (b) the MaxiMin solar and (c) the distance dispersion objectives.

satisfactory results [142]. As the search progresses, the constituent elements of the chromosomes (the facility sites in GFLo) systematically converge to a group of better performing sites. When successive populations do not differ significantly from preceding ones, the algorithm has reached a point where its search improvement potential (mostly governed by the algorithm parameter selection and sites in the current population of solutions) begins to stagnate. At this point, the solutions or constituent sites that will eventually appear in the final approximation set may already be uncovered. Moreover, weaker solutions in the ‘trailing’ end of the population, at the time of population convergence, typically consist of weaker combinations of the best sites that are already included in locations in the better group of solutions in the populations, or at the least include weaker sites.

Consistently observing the above-mentioned population convergence characteristics during evaluation of the NSGA-II in the context of GFLoPs led to an investigation into the possibility of ‘trimming’ the population in the event that it was determined to be converging. This results in reduced computational complexity of generating offspring generations as a result of the smaller number of crossover and mutation operations associated with the trimmed population

size, in addition to resulting in faster solution ranking because of fewer solutions that have to be compared to each other. In order to decrease the computation time of the NSGA-II in such a manner, a *population convergence measure* that is similar to the Pareto convergence measure (of §5.3.1) is introduced. The population convergence measure is based on the average similarity of the solutions in the current population when compared to those in the previous population. Population convergence is considered to occur when the solutions in successive populations are above a pre-specified *population similarity threshold percentage* over a fixed number of *population similarity threshold generations*. When population convergence is observed, the population size is halved using the FNSA to select the better half of solutions from the current population to be carried over to the next population. The procedure is performed once only, should population convergence be observed.

Example 6.1 (NSGA-II population halving) *In this example the NSGA-II is used to solve a GFLoP with and without the population halving technique. Suppose the GFLoP involves the same terrain and facility-SZ assignment matrix as used in §5.4. As before, the facilities are solar-powered and their placements are done according to average daily solar irradiation values. Additionally, suppose the facilities are either ‘repulsed’ or ‘attracted’ by an existing feature located at the geographical centre-point of the physical terrain and so distance evaluated from the geographical centre-point is another criterion included in the problem.*

The objectives are therefore to:

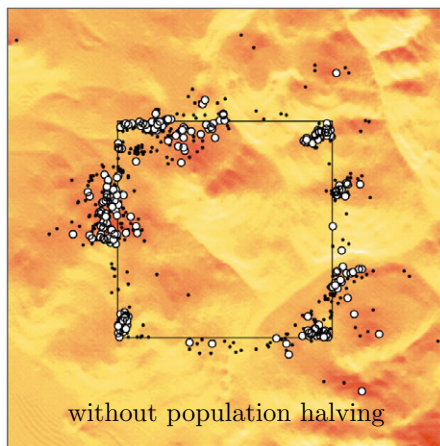
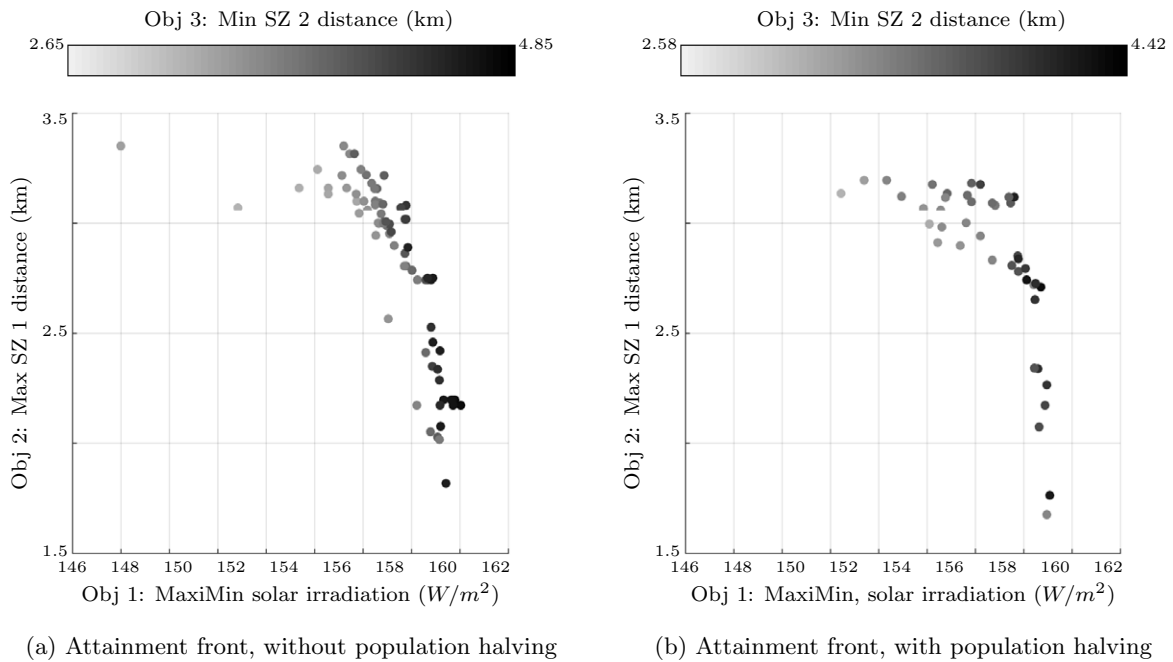
1. **Maximise** *the minimum average daily solar radiation value of the six facilities,*
2. **Maximise** *the minimum distance measured between facilities in SZ 1 and the geographical centre-point of the terrain, and to*
3. **Minimise** *the maximum distance measured between facilities in SZ 2 and the geographical centre-point of the terrain.*

As in the previous example GFLoPs, a lower bound of 500 metres is enforced on the inter-facility distance between any facility pair, in addition to the obligatory placement constraints (3.4)–(3.6).

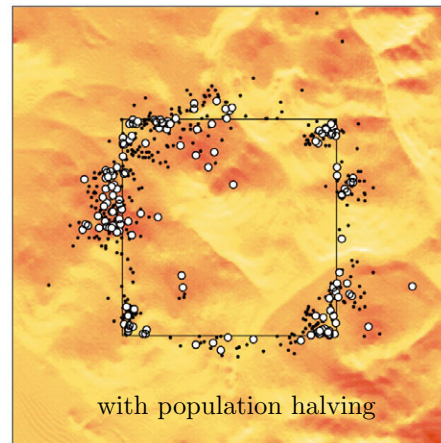
The NSGA-II parameters used for this specific example was an initial population size of 100 for both implementations, a crossover probability of 0.8 employing randomly selected two-point crossover, and a mutation probability of 0.2. In terms of population halving, a good combination of population similarity threshold percentage and generations was found to be 80% over ten generations. Both implementations were terminated with a Pareto similarity threshold percentage of 90% over ten generations.

The attainment fronts of the two implementations of the NSGA-II are provided in Figure 6.4 (a) and (b). The fronts achieve solutions of similar quality, as confirmed by the normalised hypervolume values contained in Table 6.3. A slight worsening in solution quality is, however, observed in terms of the hypervolumes achieved by the NSGA-II with population halving. Of particular significance is that although the solution quality with halving remains similar to solutions obtained without halving, the average run time is halved, as may be seen in Table 6.4. This is attributed to the smaller number of generations and the smaller number of sites that have to be evaluated by the NSGA-II with halving, as seen in Table 6.4. ■

The halving procedure discussed above is aimed towards reduced computation times. Nevertheless, results returned by NSGA-II analyses using the halving procedure consistently return solutions of similar quality to those returned by the NSGA-II without population halving. In



(c) Site locations of solutions obtained by the NSGA-II without population halving



(d) Site locations of solutions obtained by the NSGA-II with population halving

FIGURE 6.4: Attainment fronts of the NSGA-II (a) without population halving, and (b) with population halving for the GFLOP instance in Example 6.1. In (c) and (d) the sites included in the solutions of the approximation sets (the black markers) and the sites included in solutions of the attainment fronts (the white markers) are presented for the NSGA-II without and with population halving, respectively.

the instances where the solutions are weaker, the difference in solution quality is negligible. The magnitude of the reduction in computation times is problem-specific and not easy to generalise. This depends on the number of facilities to be placed and the complexity of the problem in terms of the physical size of the SZs, the physical terrain type, the number of objectives and the types of the objectives, in addition to whether data are accessed from memory (faster) or read from disk storage (slower). The halving technique results in at least a slight reduction in computation time, with the reduction in run times increasing with an increase in problem complexity — especially when data are accessed from files instead of memory. The halving procedure is implemented in all NSGA-II solution processes from here onwards in this dissertation.

	Mean	Minimum	Maximum	Attainment
No halving	0.786	0.544	0.904	1
Halving	0.718	0.582	0.853	0.948

TABLE 6.3: Normalised hypervolume values for thirty runs each of the NSGA-II with and without the population halving technique for the GFLoP instance in Example 6.1. The values are normalised by the NSGA-II attainment front obtained without the halving technique.

	Run time (s)	Generations	Sites evaluated
No halving	20.8	95	2451
Halving	10.4	66	1576

TABLE 6.4: Average values per run for thirty runs each of the NSGA-II with and without the population halving technique for the GFLoP instance in Example 6.1.

6.3 A multi-resolution approach

A *multi-resolution approach* (MRA) that aims to concentrate the search process towards well-performing areas of the search space is proposed in this section. The main purpose of the MRA is to improve solution quality. A second advantage of the MRA, however, is that it reduces overall computation time for problems requiring resource-intensive analyses and data storage (although it is also capable of reducing computation times for the other problems as well).

The MRA has been proven to return superior results for a variety of GFLoPs involving varying objectives and terrains of different types (in terms of ‘roughness’) and sizes. The MRA is one of the major contributions of this dissertation and a variety of examples of its application and the superior solution quality that it returns are available in research outputs [54, 55]. A general discussion on its motivation and working are provided in §6.3.1 and §6.3.2, respectively. This is followed in §6.3.3 by an example instance of typical results achieved by the MRA, as compared to the standard *single-resolution approach* (SRA) used until now.

6.3.1 Motivation

During multiple runs of the three algorithms of §5.3 and §6.1 developed for identical problem instances, the sites included in the solutions making up the final populations and Pareto-front approximation sets are typically observed to emanate from the same regions. Analogously, large poorly performing areas, which often envelop large numbers of gridposts, are typically absent from the final sets of solutions and these absent regions typically exhibit much larger sizes than the areas covered by those containing final solution sites. If the initial solution populations and randomly chosen mutation sites are selected stochastically over entire SZs, gridposts that eventually do not form part of the final population of solutions are therefore needlessly analysed. To illustrate this occurrence, examples of typical collections of all the sites evaluated, in addition to the final sites that are included in the non-dominated solution combinations, are presented for three different GFLoPs in Figure 6.5. In Figure 6.5 (a), all the sites that were evaluated over the thirty runs of the NSGA-II for the GFLoP instance of §5.4 are indicated by grey markers, while all the sites that are included in the non-dominated solutions of all thirty runs are indicated by red markers. Similarly, results for the thirty runs of the NSGA-II without population halving for the GFLoP instance of Example 6.1 are provided in Figure 6.5 (b), while the results of ten runs of the NSGA-II are provided in Figure 6.5 (c) for a large-area terrain (approximately 100×70 km), bi-objective GFLoP involving the placement of five facilities. The number of sites

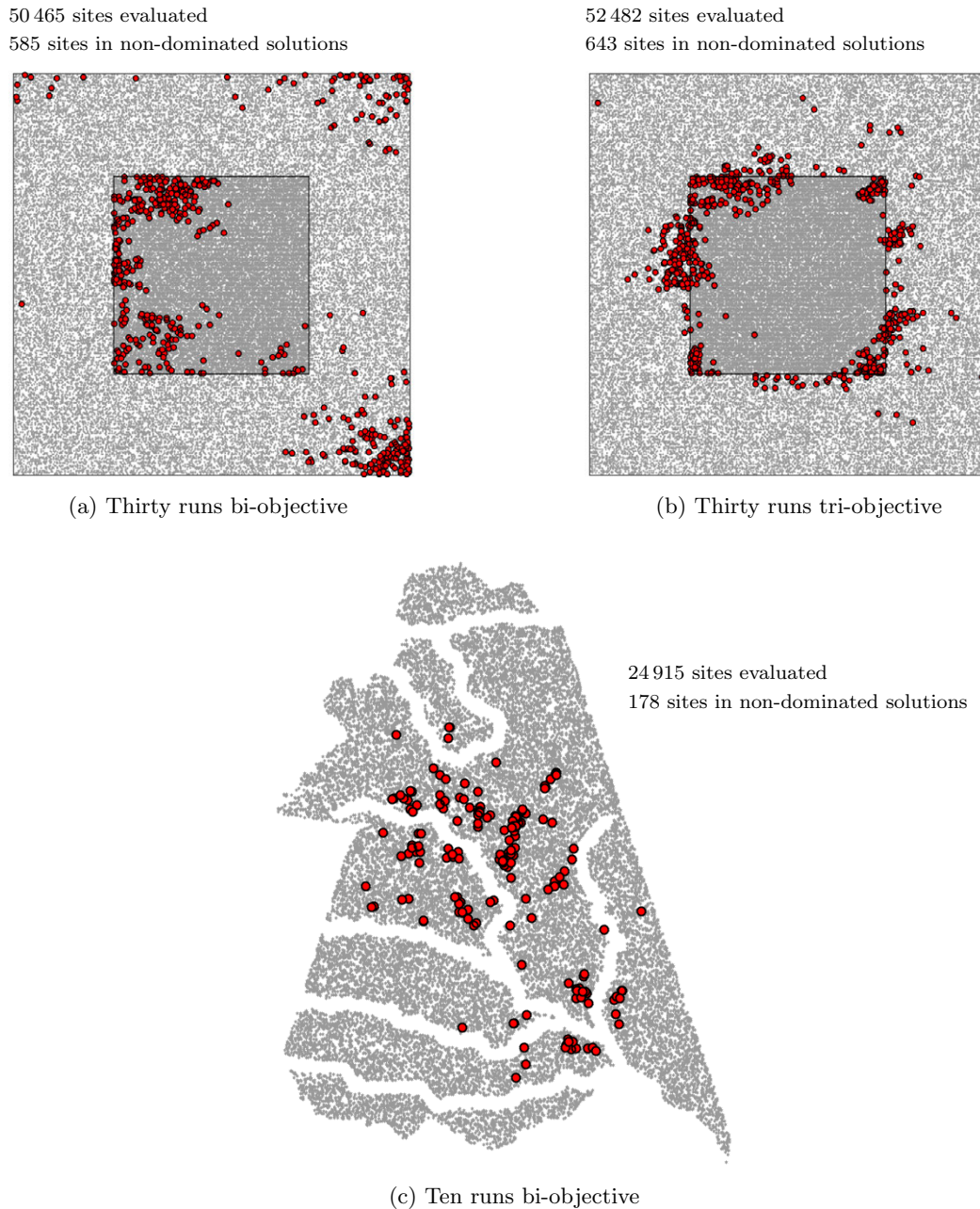


FIGURE 6.5: Typical candidate sites evaluated and resulting sites included in the non-dominated solutions returned by (a) thirty runs of the NSGA-II for the problem instance of §5.4, (b) thirty runs of the NSGA-II for the problem instance of Example 6.1, and (c) ten runs of the NSGA-II for a five-facility, bi-objective problem instance on a large-area terrain of approximately 100×70 km. Sites included in the final solutions uncovered by multiple runs continue to appear in the same, well-performing regions (the red markers), but involve the costly evaluation of a large number of sites that are not included in the final solutions (the grey markers).

evaluated and sites included in non-dominated solutions differ substantially — approximately only 1% of all evaluated sites are included in non-dominated solutions.

It is clear from Figure 6.5 that a large number of gridposts are evaluated which are not included in the final solutions. Accumulatively, consideration of these sites can be very costly in terms of the computation times required to calculate their performance values with respect to the

objectives. This is especially significant in analyses that require time-consuming computations for each location that is considered, depending on the type of criteria considered. For example, the computation of a single gridpost's viewshed in respect of *all* the gridposts in a 10×10 km area of interest (the size of the terrain used in the GFLoP examples until now), using a data resolution of approximately 28 metres between gridposts, takes on average 2.3 seconds². Additionally, these data are often stored to disk for fast access later³, in which case the number of locations analysed and associated file reads required should ideally be minimised. Required data storage volumes are also reduced as a result of the evaluating fewer candidate sites.

If the search for solutions can be focussed in the well-performing areas surrounding the sites that are included in the non-dominated solutions — thereby avoiding the evaluation of the remaining solutions — a considerable amount of computational burden can potentially be avoided. One manner of achieving this is to perform criteria-specific analyses in order to identify gridposts that are likely suitable placement locations with respect to terrain characteristics and unique facility requirements. An example of a visibility-related identification of likely placement locations is provided in [66] where analyses are performed to identify areas on the terrain surface (typically areas of higher elevation such as ridges and peaks) that are likely to achieve good visibility of the surrounding terrain for the purposes of a subsequent viewshed-dependent facility location. Reducing the SZ in such a manner may, however, lead to the exclusion of some good candidate solution sites. It has, in fact, been shown in the literature that this type of terrain classification does not necessarily result in the selection of the best potential sites [66, 104]. Moreover, it may take a large amount of time to perform specific geographical analyses for the identification of good areas. The inclusion of objectives that are not related to visibility (*e.g.* proximity) would further render such a procedure meaningless, since the filtering of areas that perform well with respect to visibility-related objectives has no relation to proximity-related criteria. This type of criteria-specific filtering is therefore not considered in this dissertation.

Recurring observations arise in respect of solution quality at different resolution levels. To perform lower resolution analyses, gridposts representing candidate site locations at the highest available data resolution may simply be reduced to uniformly spaced low-resolution representations, after which the algorithms may be applied to find separate sets of Pareto-front approximation sets based on candidate solutions at the respective resolutions⁴. Results of the different resolution data analyses may then be compared in decision and objective spaces — Figure 6.6 (a) and (b) provides illustrative examples of typical results. As may be seen in Figure 6.6 (b), low-resolution Pareto-fronts are observed to be inferior to their high-resolution counterparts due to 'good' gridposts being overlooked at low resolution and are therefore not included in candidate solutions. The inferior fronts may, however, be determined in a shorter amount of time, due to fewer gridposts being considered, which may ultimately lead to faster approximate Pareto-front convergence. With respect to the solutions in Figure 6.6 (a), the low-resolution solution sites are often observed to correspond with — or at least be in the vicinity of — some of those returned by higher resolution analyses. This observation leads to the conjecture that, due to the essentially continuous nature of geospatial criteria values, lower resolution data may be explored in order to quickly identify regions in decision space that perform well with respect to the objectives, after which higher resolution data may be exploited within the vicinity of these regions in order to improve on solution quality.

²Determined using MATLAB on an Intel Core i7-4770 CPU, utilising one of eight 3.40GHz cores, 8 GB RAM, and running in the Ubuntu 12.04 LTS 64-bit operating system.

³The file reads themselves may be time-consuming, especially for large files. However, even large file reads are generally faster than re-computing the data.

⁴It is important to note that a lower resolution representation merely involve fewer candidate sites within the SZs on the terrain surface, while the analysis data computed with respect to the IZs and objective function

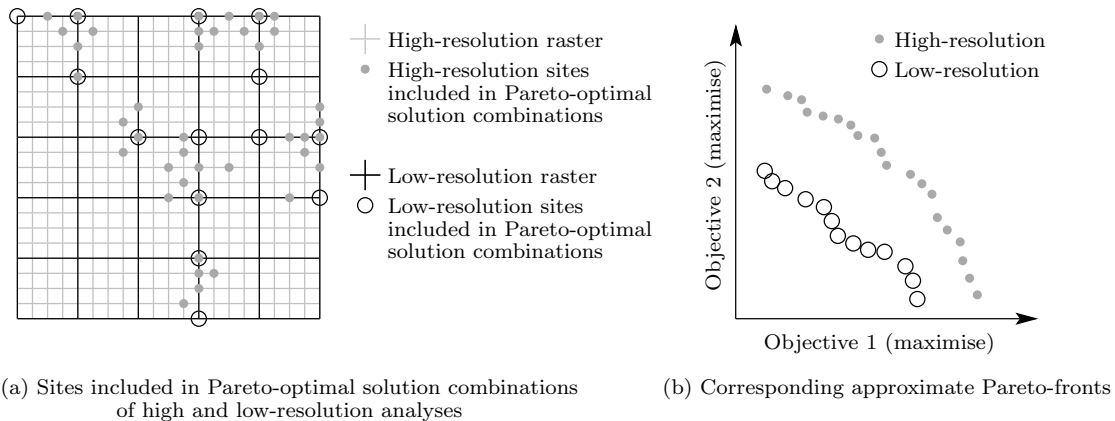


FIGURE 6.6: Sites included in approximately Pareto-optimal solutions determined during low-resolution analyses generally occur in the vicinity of those obtained during high resolutions, but may be calculated in a fraction of the corresponding time. Low-resolution sites may therefore be used to quickly identify well-performing regions that merit further exploration.

6.3.2 Method

The newly proposed MRA starts by accepting as input the SZs at low resolution, after which it enters a loop during which the problem instance is repeatedly solved (approximately) at gradually increasing resolutions (*exploration*), as illustrated in Figure 6.7 for an MRA process solved at three pre-specified resolution levels⁵. All the sites that are included in the solutions of the final population — or only those from selected fronts, depending on user preference — are carried over to the next resolution. At the next resolution the problem is solved for new SZs which include the inherited solution sites and a selection of their higher-resolution neighbours (*exploitation*). The process of selecting which higher resolution gridposts in the vicinity of each transferred site is user-defined. The steps in this loop are repeated until the problem instance is solved at the maximum resolution.

6.3.3 Worked example

A typical example of the solution process of the MRA is provided in this section. In this example, ten runs of each of the SRA and MRA implementations of the NSGA-II are executed to solve a GFLoP involving the same terrain as considered in §5.4. Suppose that two facility types (three of each type and six facilities in total) with visibility/detection requirements (*e.g.* watchtowers, radars or telecommunication towers) are to be placed on the terrain and that they have the following characteristics:

- The facilities of type $f = 1$ have visible distances that extend past the boundaries of the portion of terrain considered. Viewsheds for these facilities are therefore determined with respect to the entire surface area. The heights of these facilities are such that the viewpoints are 5 metres above the ground. This is important for viewshed computation since an elevated point of view results in improved terrain visibility.
- Facilities of type $f = 2$ each has a visible distance of 2 500 metres and a viewpoint height of 3 metres above the ground.

criteria may always be determined using the highest resolution data.

⁵A pseudocode description of the MRA is provided in [55].

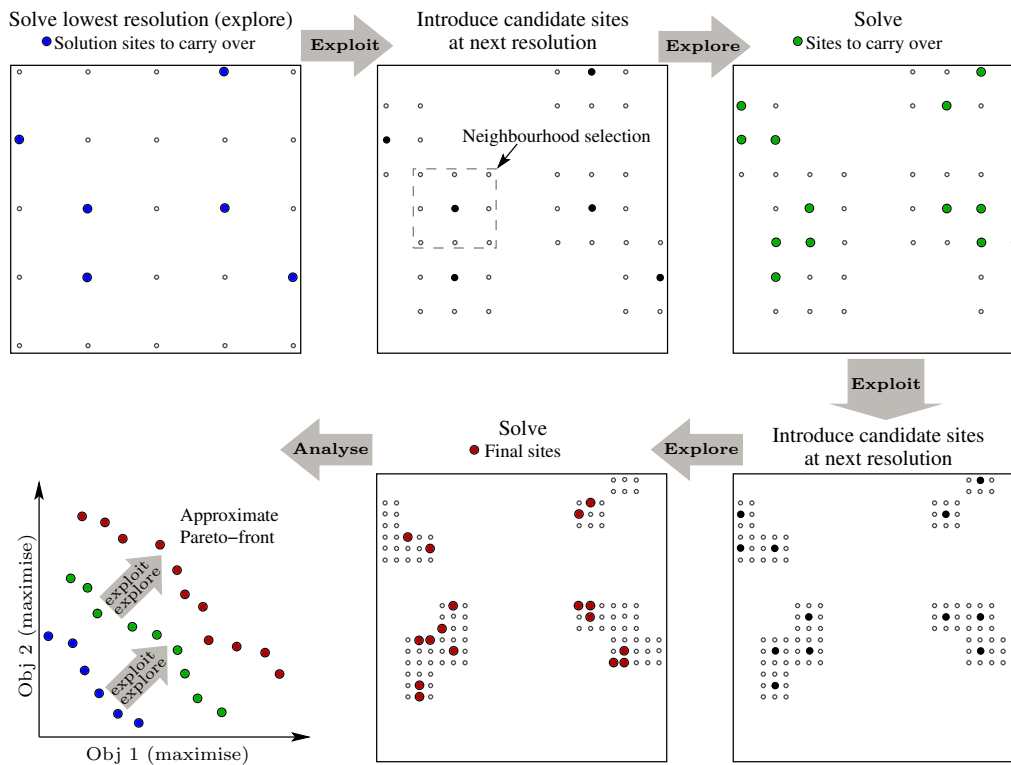


FIGURE 6.7: An illustration of the process followed during execution of the MRA.

The aim is (a) to place two facilities of type 1 and one facility of type 2 in SZ 1, and (b) to place one facility of type 1 and two facilities of type 2 in SZ 2. Suppose the problem to be solved is a BCLP and that the visibility requirements of the entire terrain surface are considered to be equally important and so the IZ comprises all the gridposts in the terrain. The objectives are:

- to place the six facilities in such a manner that the number of gridposts which are visible from *at least one* of the six facilities is maximised, and
- to place the six facilities in such a manner that the number of gridposts in the IZ which are visible from *at least two* of the six facilities is maximised.

As in the previous GFLoP examples, a lower bound constraint of 500 metres is enforced on inter-facility distance between any facility pair, in addition to the obligatory placement constraints (3.4)–(3.6).

Suppose that four resolution levels are selected for use during the MRA solution process. The starting resolution level is chosen as 12 arc second spacing, which roughly translates to 340 metres between successive gridposts (as opposed to the maximum available resolution of 1 arc second, or approximately 28 metres between successive gridposts). At this resolution, the total number of gridposts in both SZs representing the terrain is 924 (as opposed to 125 125 at the maximum resolution). After the initial resolution level, the subsequent levels are chosen as 6, 3 and the maximum level of 1 arc seconds⁶. When entering a new resolution level, the new gridposts are selected as a 5×5 gridpost span across longitude and latitude, centred around each gridpost carried over. The maximum data resolution level is used to solve the problem for the SRA.

⁶Graphical illustrations of the candidate site locations at the four resolution levels are available in [55].

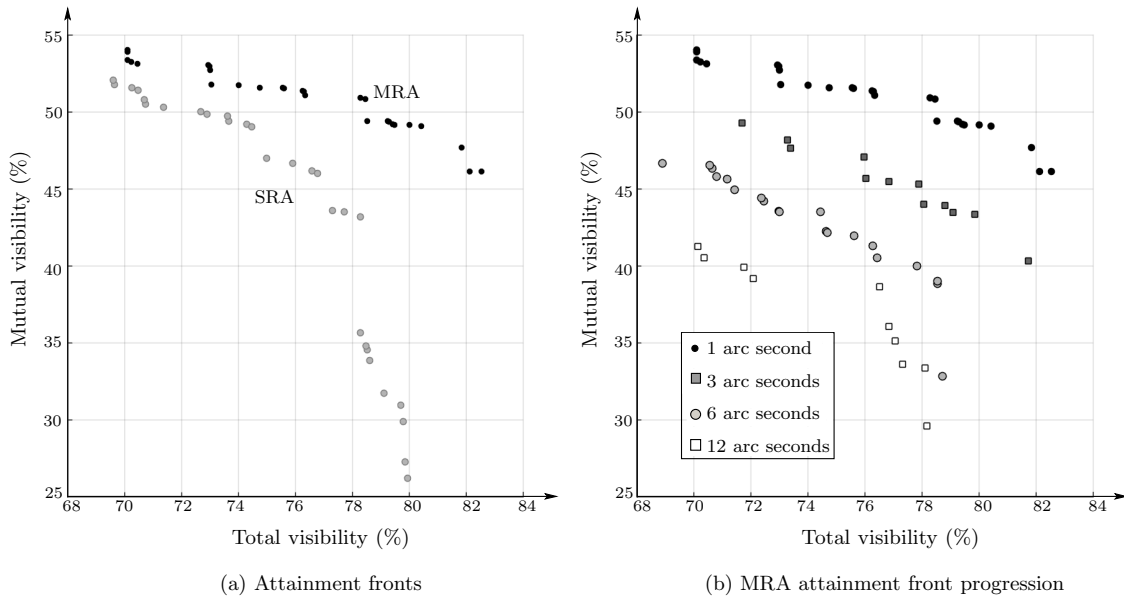


FIGURE 6.8: (a) Attainment fronts obtained by solving the problem instance of §6.3.3 according to the SRA and MRA. In (b), the attainment fronts obtained by the MRA are shown at its separate resolution levels.

Population sizes of 100 and 50 are used for the SRA and MRA, respectively. Both implementations employ a crossover probability of 0.8 and mutation rates of 0.2 and 0.1 for the SRA and MRA, respectively. The population sizes and mutation probabilities are selected according to the sizes of the search spaces encountered in the respective approaches.

Solution quality

The optimisation results are presented graphically in objective space (with total *versus* mutual visibility expressed as a percentage of all the gridposts included in the terrain) in Figure 6.8 (a), in the form of the two attainment fronts of the respective approaches. As may be seen in the figure, all SRA solutions are dominated by solutions in the MRA front. The progression of the approximate Pareto-fronts obtained at the different resolution levels of the MRA is illustrated in Figure 6.8 (b) in the form of the attainment fronts of the approximated sets at each level. The values of the hypervolume measure (normalised by the hypervolume of the attainment front of the MRA) for the two approaches are shown in Table 6.5.

	Hypervolumes				Mean run time (minutes)		Mean number of generations
	Mean	Min	Max	Attainment	Metaheuristic	Complete	
SRA	0.424	0.184	0.742	0.764	1.5	31.26	62
MRA	0.794	0.635	0.950	1	2.0	20.47	181

TABLE 6.5: Results for ten runs of each of the SRA and MRA corresponding to the results of Figures 6.8 and 6.9. The hypervolume values are normalised according to the MRA attainment front hypervolume.

From the fronts in Figure 6.8 and the normalised hypervolume values in Table 6.5 it may be seen that the MRA Pareto-front approximation sets clearly outperform those of the SRA. The mean hypervolume value of the MRA is larger than that of the attainment front for the SRA. Based on these values (and other empirical evaluations, including those in [54, 55]) it is anticipated that any number of runs of the MRA will be superior to the same number of runs of the SRA.

Computation times and locations evaluated

In §6.3.1, the advantages of evaluating fewer locations that are concentrated within well-performing areas of the search space were outlined in order to arrive at high-quality approximate Pareto-fronts within reduced computational times. It is important to note that each of the ten runs for each respective approach commenced with most of the viewshed data being pre-computed from former analyses, resulting in drastically reduced computation times when compared to solving the problem with no pre-computed data. A cacheing approach was followed, whereby only the viewsheds of previously unevaluated locations are computed and stored to disk. These viewsheds are recalled in the event that a location is again considered (in combination with other locations) in subsequent analyses — hence avoiding the requirement of costly re-computation of viewshed data for a previously considered location. The effects of viewshed computations are nevertheless important during the process of solving GFLoPs, since not all problems may have pre-computed data available at the start of the solution process and a more efficient approach, such as the MRA, may have a drastic effect on the overall computation time if it reduces the number of locations analysed, as expected. For these reasons the computation times were monitored in terms of (a) *metaheuristic computation time*, which is the computation time of the solution process excluding viewshed computation and file read times, and (b) simulated *complete computation times*, which is file read and viewshed computation times added to the metaheuristic computation times. By monitoring the new locations introduced per facility type per run it was possible to simulate the complete computation times of the approaches using average viewshed computation times, in the event that both started with no pre-computed data and ran separately, *i.e.* files were not shared between the two approaches.

During the analysis of the solution data of both approaches, a high correlation was observed between the number of locations considered and the corresponding computation time, as expected. These data are presented visually in Figure 6.9, employing dual-scale vertical axes in Figure 6.9 (a)–(c) which represent the number of locations evaluated as well as computation times. The number of locations in the figure refer to single locations only and do not consider whether the location was evaluated twice for separate facility types. The viewshed computation times of separate facility types were, however, accounted for in simulated computation times. In Figure 6.9 (a) it may be seen that the SRA introduces a substantial number of previously unevaluated candidate locations per run. The MRA introduces considerably fewer such locations per run and a notably smaller initial number of locations compared to that of the SRA results in a considerably smaller computation time for the smaller number of viewsheds. As may be seen in both approaches, the computation time per run strongly correlates with the number of locations evaluated. Since, for this particular scenario terrain, a viewshed took, on average, 2.32 seconds to compute per candidate location for the placement of a facility of type $f = 1$, and 0.23 seconds for a facility of type $f = 2$, the total viewshed computation time is a significant contribution to the total computation time per run.

In Figure 6.9 (b) it may be seen how the cumulative run time of the SRA continues to rise steeply, while that of the MRA gradually begins to settle down. This is a result of the MRA continually focussing its search in the same promising reasons and, as a result, viewshed computations required for new locations decrease for each run. This is particularly significant considering the fact that the MRA returns superior Pareto-front approximations — a clear indication of its ability to focus its search in well-performing areas only. The cumulative computation time of the runs is related to the cumulative number of gridposts considered.

Figure 6.9 (c) is of particular significance to GFLoPs in which data is accessed from memory and do not require the computation of resource-intensive analyses (such as viewsheds), or problems

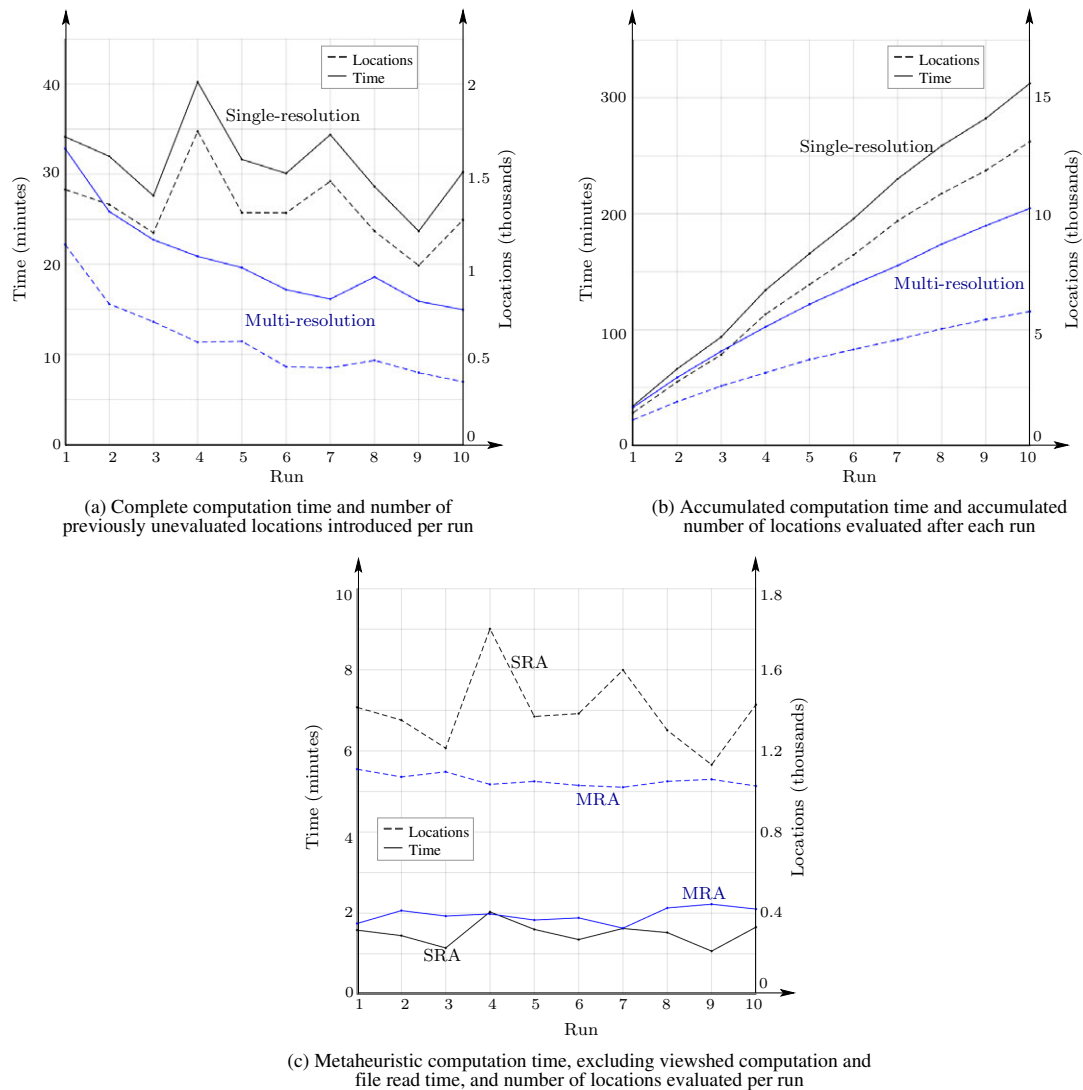


FIGURE 6.9: (a)–(b) Computation times and numbers of locations evaluated for ten runs of each solution approach in §6.3.3 (for separate runs and accumulatively). In (c), viewshed computation times are excluded and therefore represent only metaheuristic computation times related to the NSGA-II.

for which such analyses are already pre-computed (in which file read time remains an important consideration). The graph in (c) shows the computation times of the two approaches when excluding viewshed computation and file read times. The computation times presented in these figures therefore represent the time spent on metaheuristic analysis only. The MRA in this case executes within slightly slower average run times — attributed to the increase in the number of generations over the four resolution levels, negating the gains achieved in terms of reduced computational complexity at each resolution level. This type of observation is problem-specific and problems in which the MRA executes faster than the SRA in terms metaheuristic computation times are not uncommon (see [55], for example), while in instances where the MRA metaheuristic computation times are slower than those of the SRA, the increases in these times are generally negligible. The mean number of generations, as well as the mean metaheuristic and complete computation times for the two approaches are presented in Table 6.5.

In Figure 6.9 (c), it may also be seen that the SRA evaluates a considerably larger number of locations per run than does the MRA (a mean of 1 400 compared to 1 055), since it requires

larger population sizes than those of the MRA in order to explore the terrain surface adequately. Figure 6.10 shows, in grey, all the locations considered during all ten runs of the SRA, a total of 13 115, compared to the 5 784 locations considered during all ten runs of the MRA. It may be seen for the MRA how certain ‘weaker’ areas are evaluated at lower resolutions only and identified as such, while some areas are specifically identified and analysed more intensely at higher resolutions. Sites included in the non-dominated solutions of all ten runs of each approach are also indicated in red for facilities of type $f = 1$ and in blue for facilities of type $f = 2$. The progression of candidate sites and resulting final population sites (around which the candidate sites of each new resolution level were centred) are shown in Figure 6.11 for all ten runs of the MRA at the four resolution levels, so as to visually illustrate the manner in which the explore-and-exploit approach seeks out the locations that make up the best solutions.

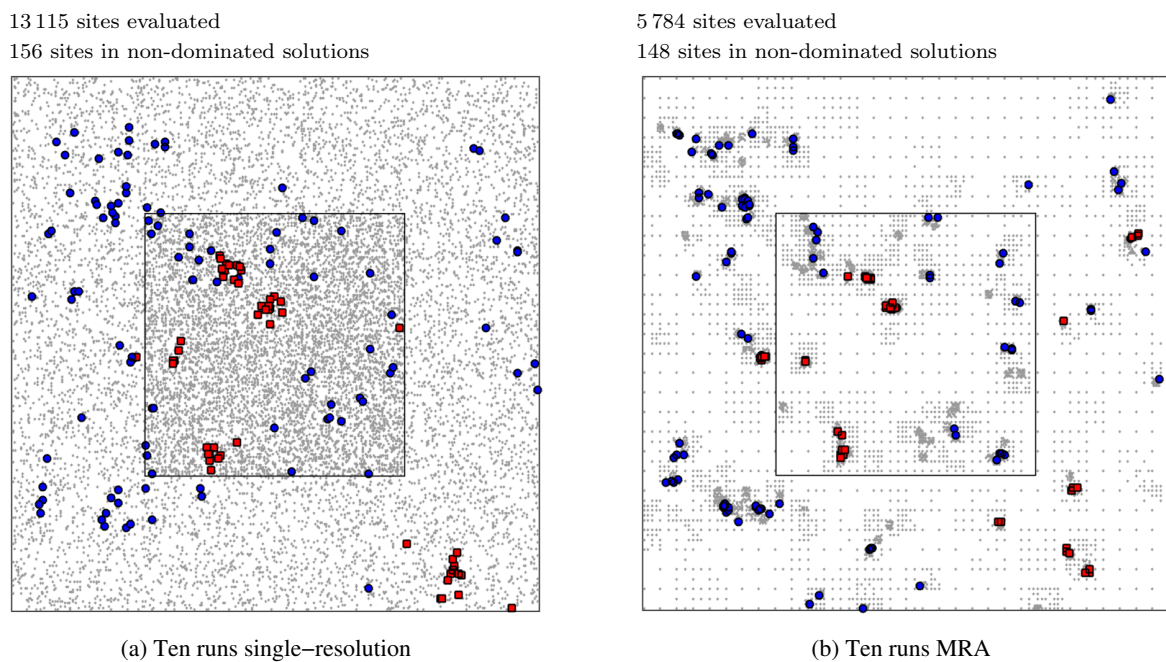


FIGURE 6.10: (a) Candidate locations (indicated by grey markers) evaluated during ten runs of the SRA compared to (b) candidate locations evaluated during ten runs of the MRA for the problem instance solved in §6.3.3. Sites included in the non-dominated solutions of all ten runs of each approach are also indicated in red for facilities of type $f = 1$, and blue for facilities of type $f = 2$.

6.3.4 General observations and implementation notes

The effectiveness of the newly proposed MRA was demonstrated in §6.3.3 and additional examples are available in [54, 55]. The MRA returns varying results in terms of computation times for the three different algorithms. The MRA nevertheless consistently returns superior quality solutions for all three algorithms as a result of its explore-and-exploit approach. Observations in terms of computation times, however, vary as a result of the population sizes and solution approaches followed by the respective algorithms. In all analyses that require resource-intensive analyses and file reads, the algorithms execute within reduced computation times as a result of the fewer candidate locations evaluated and associated files stored to disk, in addition to the data being accessed by a caching method, so as to ensure faster lookups in later analyses. In terms of metaheuristic computation times, the MRA may have a negative effect on AMOEBA and the HGA, compared to the NSGA-II. This is because AMOEBA and the HGA in general

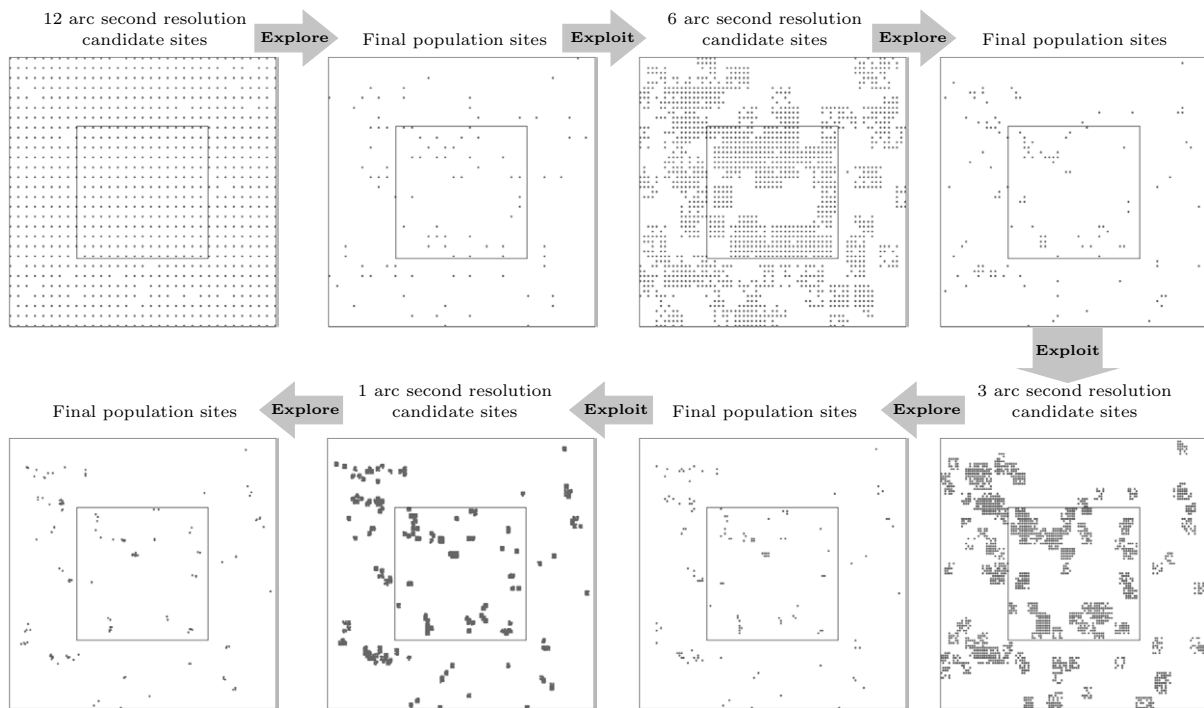


FIGURE 6.11: The progression of candidate sites and sites included in the final populations of solutions in all ten runs of the MRA at the four resolution levels considered in the scenario, illustrating the explore-and-exploit search approach.

evaluate fewer candidate solutions per run than does the NSGA-II, as observed in the problem instances of §5.4 and §6.1.2. AMOEBA and the HGA therefore gain less than the NSGA-II in terms of the number of locations reduced per run and, in some instances for AMOEBA, the total number of locations per run may even increase. The increase in run times that are observed (if any are observed) are nevertheless acceptably small and considering the already fast computation times of AMOEBA and the HGA, are deemed to be an acceptable trade-off in terms of the consistent and significant improvement in solution quality observed for all three algorithms. In the pursuit of the best quality solutions and to ensure fair algorithm comparisons, the MRA is employed in the contexts of all three algorithms for all GFLoPs solved in the remainder of this dissertation.

6.3.5 Important MRA considerations

The choice of which resolution levels to include in the model — in terms of the number of levels and each level’s spacing distances — may be a parameter pre-specified by the user, or it may be computed dynamically during the solution process. The quality of the final solutions are highly dependent on the locations of the solution sites identified during analyses performed at the first (lowest) resolution level, since subsequent analyses at higher resolutions will be limited to candidate sites in their vicinity — as may be observed in Figure 6.11. If this resolution level is chosen too low, there is a risk of the solution space being overly simplified, resulting in potentially well-performing candidate sites and their sub-regions being overlooked and lost. It is therefore critical to select the initial resolution level so that it contains sufficient information for the identification of candidate sites that lie within well-performing sub-regions. An appropriate initial resolution level is expected to depend primarily on terrain roughness, features and size — ‘rough’ terrains are expected to require higher initial resolution levels than undulating terrains.

The range of neighbourhood exploitation when moving into higher resolution levels is another important consideration. In this dissertation (and as in the example of §6.3.3) a simple $n \times n$ site span, centred around each candidate site carried over, is used to introduce new candidate sites when moving into higher resolutions⁷. If the range of neighbourhood exploitation is too small, there is a risk of reducing the solution space too quickly, resulting in weak solution diversity and potentially weak solution quality due to the premature loss of good candidate sites. If this range is too large, however, more candidate sites, including more weaker ones, may be introduced into the population — resulting in increased computation times and potentially weaker solution quality.

6.4 Proximity-dependent cluster removal

An observation made during the solution processes of numerous GFLoPs is that the number of approximately Pareto-optimal solutions generally increase with an increase in the number of facilities to place. This is to be expected, because an increase in the number of facilities to place results in an increase in the size of the solution space. A large number of candidate solutions may, however, be undesirable from a decision making perspective, as the decision maker may become inundated with solutions and trade-offs to consider. This becomes even more problematic when the solutions are very similar in terms of their candidate facility sites and in objective function space — meaning that a large number of ‘redundant’ solutions may clutter the solution set.

To illustrate the similarity of solutions that may result in an undesirably large number of solutions, consider Figure 6.12. Figure 6.12 (a) contains an attainment front (in grey) achieved by AMOEBA for a tri-objective problem involving the placement of six facilities with a BCLP and a proximity-related objective, for a GFLoP solved in the context of the terrain in §6.3.3. The attainment front consists of 762 solutions — an impractically large number for a decision maker to consider. There exist methods in the literature to reduce approximation sets of considerable size, such as the epsilon grid method [85]. Such a method is, however, not considered suitable in a GFLo environment, given the role of the physical solution space and the locations of facilities in terms of the decision making process. Instead, a proximity-based cluster filtering technique is proposed. A cluster may be defined as a set of more than three solutions for which the constituent facility sites are either identical or within a specified distance of a site included in another solution. Using this approach and a clustering distance of 1 000 metres, fifty clusters were identified within the attainment front of Figure 6.12. The largest of the clusters, consisting of 42 solutions, is presented in Figure 6.12 (a) as black markers. The physical site locations that are included in the cluster’s solutions are shown in Figure 6.12 (b).

It is clear that such a cluster does not provide any significant trade-offs in objective function space, while the physical sites that are included in the solutions also do not provide any remarkable trade-offs in terms of facility placement options. In fact, all forty two of the solutions include combinations of the four sites that are isolated and on the right-hand side in Figure 6.12 (b), while the cluster solutions’ two remaining constituent sites are merely different combinations of one facility from each of the two groups of remaining sites. Such a cluster may, for example, be reduced in number by simply selecting the solutions from the cluster that achieve the maximum values with respect to the three objectives, and omitting the remaining solutions. Following this approach, the cluster in question reduced from forty two to four solutions (two solutions achieved the maximum proximity value within the cluster). When this type of clustering and

⁷Selecting new candidate sites according to a neighbourhood radius instead of a site span is another approach that may be considered. Dynamic neighbourhood site spans or radii, determined according to each resolution level, are alternative approaches that may be considered.

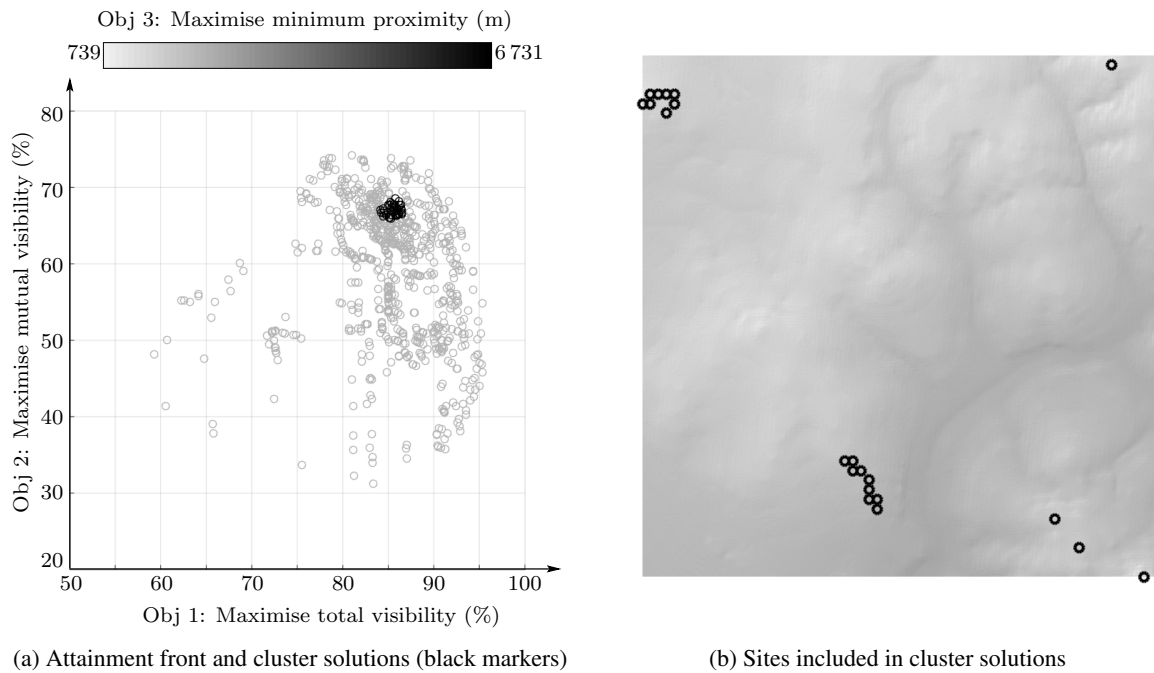


FIGURE 6.12: (a) A cluster of 42 solutions identified within an attainment front of 763 solutions. (b) 21 sites included in the 42 solutions that form the cluster in (a). The figures illustrate the occurrence of negligible trade-off alternatives offered in respect of physical site locations and objective function scores for such a cluster.

filtering was performed on all fifty clusters in the attainment set of solutions, the number of solutions in the attainment front reduced from 762 to 305, and resulted in only a small decrease in the normalised hypervolume value, from 1 to 0.986. This type of clustering and filtering may drastically reduce the number of redundant solutions in large solutions sets, while facilitating a simpler post-optimisation decision-making process for the decision-maker in terms of selecting a final solution for implementation.

The removal of clusters does not form part of the main scope of this dissertation and was performed for investigative purposes. It is therefore not considered in the remainder of this dissertation, but was mentioned here because of its importance in the decision making process of GFLo. The declustering of GFLo results is expected to be an important decision making component in a comprehensive GFLo framework interface that offers GFLo decision making capabilities to a user in a simplified, interactive manner. Such an interface is discussed in the following chapter.

6.5 Chapter summary

This chapter opened with the development of a hybrid algorithm in §6.1, inspired by the results returned by the NSGA-II and AMOEBA for the problem instance of §5.4 which revealed advantageous properties of each algorithm that may be exploited in a hybrid solution approach. In the example of §6.1.2, the HGA outperformed both AMOEBA and the NSGA-II in terms of solution quality, while its computation times were observed to be similar to that of AMOEBA, which are significantly faster than those of the NSGA-II. The advantage of the HGA lies in its combinatorial search strength gained from NSGA-II crossovers, while it exhibits strong explorative properties due to its implementation of the AMOEBA mutation process.

An NSGA-II population halving procedure, aimed at reducing NSGA-II computation times, was introduced in §6.2. This procedure is based on population convergence, and in Example 6.1 returned solutions of similar quality to that returned by the NSGA-II without the halving procedure — this was, however, achieved within half the time.

The MRA, which aims to concentrate the search process towards well-performing areas of the search space, was introduced in §6.3. The method was shown in §6.3.3 to return superior results for a BCLP solved by the NSGA-II for both the SRA and MRA methodologies. The approach was demonstrated to be especially valuable in terms of its possible inclusion in GFLo solution approaches that require resource-intensive computations. The MRA is one of the major contributions of this dissertation and further proof of its effectiveness is available in the literature [54, 55].

A proximity-dependent attainment set declustering technique, aimed at simplifying the decision making process, was finally proposed in §6.4.

CHAPTER 7

Practical implementation of the generic GFLo framework

Contents

7.1	Proposed GFLo decision support architecture	106
7.1.1	<i>Architecture overview</i>	106
7.1.2	<i>An optimisation interface concept demonstrator</i>	108
7.2	Practical use of the generic GFLo framework	110
7.2.1	<i>Problem overview</i>	110
7.2.2	<i>Evaluation of existing camera locations</i>	112
7.2.3	<i>Solving the problem using the GFLo framework</i>	113
7.2.4	<i>Comparison of actual and proposed camera locations</i>	114
7.2.5	<i>Decision maker feedback and conclusion</i>	115
7.3	Small-area problem instances	115
7.3.1	<i>Low-complexity bi-objective GFLoP instance</i>	116
7.3.2	<i>Low-complexity tri-objective GFLoP instance</i>	117
7.3.3	<i>High-complexity tri-objective GFLoP instance</i>	121
7.3.4	<i>Algorithmic observations for the small-area problems</i>	125
7.4	Large-area examples	128
7.4.1	<i>High-complexity bi-objective GFLoP instance</i>	129
7.4.2	<i>Low-complexity tri-objective GFLoP instance</i>	130
7.4.3	<i>Algorithmic observations for the large-area problems</i>	133
7.5	Chapter summary	135

The purpose of this chapter is to provide proof of the workability of the components that form the generic GFLo framework proposed in this dissertation. This includes the mathematical models developed in §4 and the algorithms and GFLo-specific search improvements introduced in §5 and §6. In order to achieve this, a GFLo decision support architecture is proposed in §7.1. The GFLoP instances in this chapter were solved by an optimisation interface concept demonstrator of this architecture.

A practical study involving observation camera placement at a large radio telescope project [121] was performed as a method of comparing the results of the GFLo framework to actual facility sites. This study is performed in §7.2 in order to compare the results of the GFLo framework

to actual facility sites. This study was performed using the concept demonstrator of §7.1 and provides preliminary proof of the GFL0 framework's usefulness.

A selection of realistic GFL0P instances involving a variety of criteria and facility-SZ assignments which combine objectives and constraints that are absent from the traditional GFL0 literature are solved in §7.3 and §7.4 so as to further illustrate the workability of the framework — these GFL0P instances were all also solved using the concept demonstrator of §7.1. Moreover, these additional examples serve the purpose of shedding light on the performance of the algorithms when used to solve different GFL0Ps. An in-depth comparison of the algorithms in order to establish an algorithm selection framework is, however, not pursued, since the performance of the algorithms is expected to be problem-specific and such a detailed study falls outside the scope of this dissertation. As such, the results that are presented in respect of algorithmic performance and solution quality for the separate problem instances are mostly informative. Since the GFL0P instances presented here involve novel concepts in terms of the objectives, constraints and facility-SZ assignments, it is anticipated that the examples presented here may spawn new avenues of research in the GFL0 domain¹.

One bi-objective and two tri-objective GFL0P instances are solved in §7.3 on the same portion of terrain as that encountered in the GFL0Ps of previous chapters, together with multiple facility-SZ assignments and considering varying placement criteria. A bi-objective GFL0P instance and a tri-objective GFL0P instance are solved on a large terrain area in §7.4. The mathematical formulations for the problem instances are provided in Appendix A for the purpose of improved readability of the main body of the chapter. Since the performance of the algorithms are problem-specific and are expected to vary when used to solve problem instances other than those considered here, an analysis and evaluation of the algorithms for each problem instance is not pursued. Instead, for each terrain surface area the problems are first solved and their results presented for the separate problem instances, after which overarching observations in respect of the algorithms are made in the specific contexts of the problem instances considered.

7.1 Proposed GFL0 decision support architecture

The generic GFL0 framework proposed in this dissertation is a synthesis of the theoretical, mathematical and MOO concepts put forward in Chapters 3–6. In order to implement the framework and its components in an efficient and practically manageable system, a generic GFL0 *decision support system* (DSS) architecture is proposed in this section.

7.1.1 Architecture overview

An overview of the GFL0 DSS architecture is provided in Figure 7.1. The pre-optimisation, optimisation and analysis components should, ideally, be made available to a user in the form of a *graphical user interface* (GUI) in order to facilitate ease of use for the user. Otherwise, these components may be tedious to use and render the GFL0 process extremely complicated.

The purpose of each component of the DSS in Figure 7.1 is listed below.

- **Pre-optimisation input.** This component provides an interface to perform pre-optimisation analyses, such as those discussed in §3.1.3. The interface communicates with the GIS component, which performs the requested geospatial operations.

¹All GFL0P instances in this chapter were solved using MATLAB on an Intel Core i7-4770 CPU, utilising one of eight 3.40GHz cores, 8 GB RAM, and running in the Ubuntu 12.04 LTS 64-bit operating system.

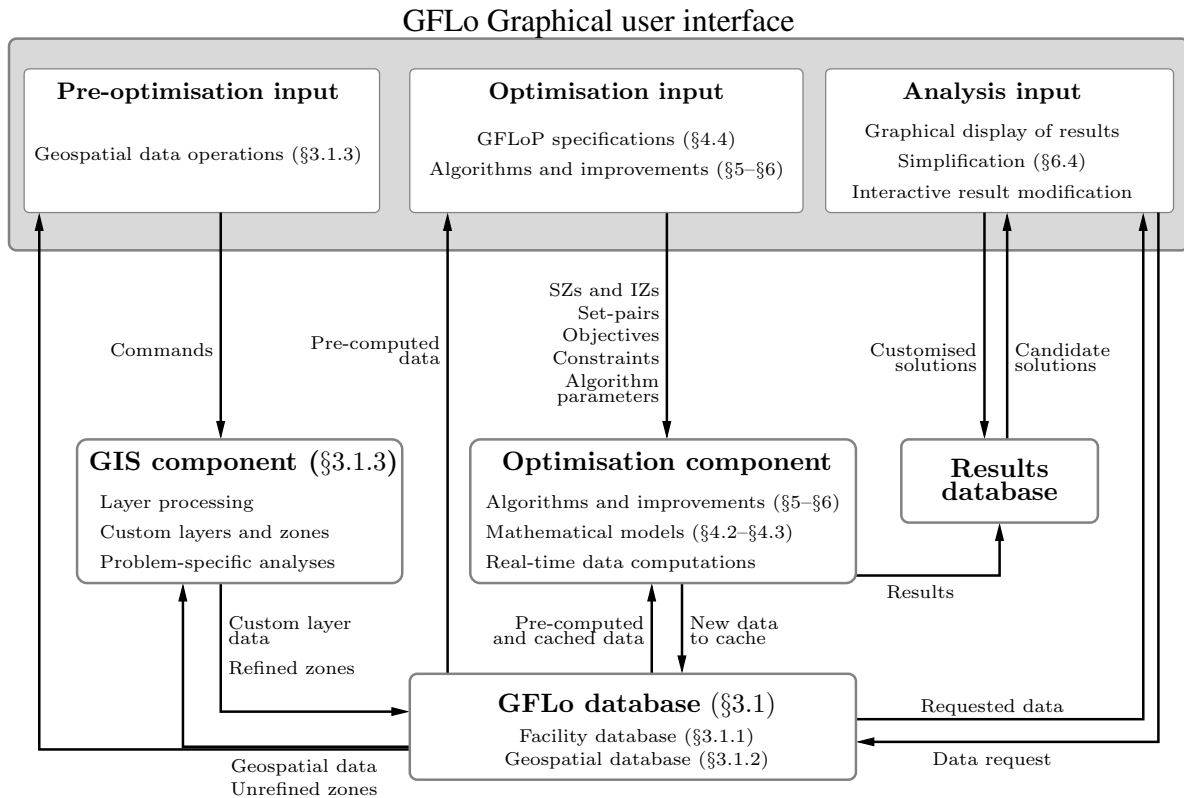


FIGURE 7.1: Proposed GFLo decision support architecture and the interaction between its components.

- GIS component.** The GIS component provides the software link between the GFLo database and the pre-optimisation input component. It may be a collection of user-created code, implemented in the same software from which it receives its user commands, or it may alternatively be external software packages, such as ArcGIS [5] or GRASS [48], which provide for external software packages to access their problem-specific analysis tools. The results of the GIS component's analysis are stored in the GFLo database for optimisation purposes later.
- Optimisation input.** This component receives the specification of the GFLoP to be solved, in the form of the zones, objectives and constraints, facility-SZ pair sets and the algorithms and algorithm parameters to include in the GFLoP. The component accesses pre-computed data in the GFLo database which may aid in the process of specifying the GFLoP in the form of visual presentations of zones and criteria values, if possible. The specified GFLoP instance data are relayed to the optimisation component.
- Optimisation component.** The optimisation component receives the information that describes a GFLoP and performs the solution search processes required to obtain the approximate set of Pareto-optimal solutions. The software implementations of the algorithms and the associated GFLo-specific search improvements discussed in §6 are executed here. The algorithms implement the mathematical models developed in §4 according to the objectives, constraints and data input values received from the optimisation input component and, furthermore, performs real-time computations related to data such as viewsheds. These data are then stored to the GFLo database for fast lookup purposes in later analyses. The optimisation component also accesses the GFLo database in order to acquire the data related to the objective and constraint criteria, as well as pre-computed or cached data related to data such as viewsheds.

- **Results database.** The results of the optimisation component are stored in a results database for analysis and decision making purposes. The results database may be updated with customised solutions tested by the user.
- **Analysis input.** Results stored in the results database are retrieved by the analysis input component, so that the solutions may be compared to each other in order to facilitate the user to make a decision in terms of a facility configuration that is considered the most suitable for implementation of a GFLoP instance. In the GFLo development process, the component may also access the results in the results database, to be used in preliminary analyses for the purposes of determining good algorithmic parameters. Furthermore, the user may perform reductions of large solution sets and perform filtering techniques — such as the proximity-based declustering technique proposed in §6.4 — with the aim of simplifying the decision making process. Finally, the component may also offer interactive result modification options, in the event that the decision maker views a specific solution and would like to view the objective function values of the solution if one of the facilities were perturbed to a neighbouring candidate location. In such an analysis, the relevant data are accessed from the GFLo database and the new solution is generated for evaluation — and may be stored in the results database, if desired.
- **GFLo database.** The GFLo database serves as described in §3.1 and interacts with all the components of the GFLo DSS.

7.1.2 An optimisation interface concept demonstrator

In order to demonstrate the type of development required for the implementation of the GFLo DSS architecture proposed in Figure 7.1, a concept demonstrator was designed for the optimisation input component of the DSS. The concept demonstrator has a GUI that allows the user to select a number of objectives and constraints in a generic manner, in addition to other GFLoP parameters, and was used in the worked examples presented in this dissertation. The GUI of this concept demonstrator is shown in Figure 7.1 in the context of solving the GFLoP instance of Example 6.1.

The concept demonstrator is an early development version of what is expected from a fully functional optimisation input GUI. As such, its development was not approached in a fully dynamic manner, but was instead focussed on satisfying the minimum requirements deemed acceptable to demonstrate the generic GFLoP solving potential of the GFLo framework proposed in this dissertation. The GUI is therefore limited to two SZs, two IZs and two facility types per problem. The number of objectives and constraints that may be added are unlimited in a programming sense, but this is nevertheless constrained by the practical solution capabilities of the algorithms since an increase in the number of objectives and constraints increases the complexity of the problem to a level that is potentially out of practical reach. The data were pre-processed using MATLAB and ArcGIS software, and stored in a GFLo database which contains two sets of location-specific criteria values (such as average daily solar irradiation levels and proximity from a selected feature) per facility type, determined for each candidate location, for consideration in MaxiMin or MiniMax objectives. Dispersion and centre objective function values only involve inter-facility distance, since this criterion type is the easiest to demonstrate in terms of practical results when displaying the facility layouts proposed for a specific GFLoP involving inter-facility objectives or constraints (such as the facility locations in Figures 6.5 (a)) and was an instrumental visual aid during debugging and functional testing processes. The programming structure nevertheless implements the generic formulations proposed in §4 and

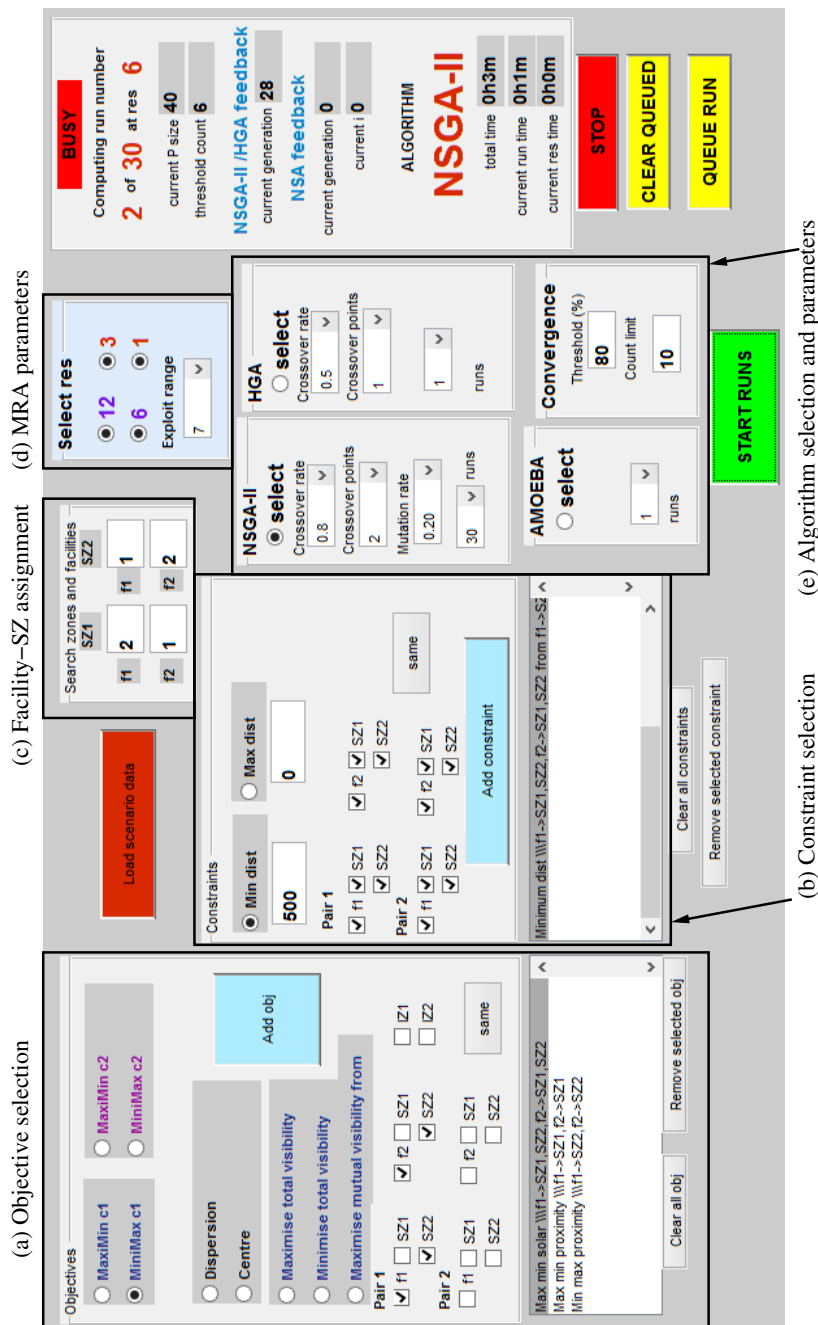


FIGURE 7.2: The optimisation interface concept demonstrator.

the implementation of these models in conjunction with other criteria is expected to be a simple process in future applications.

As may be seen in Figure 7.1, five user input panels are displayed in (a)–(e). In the objective selection panel in Figure 7.1 (a), the user selects a paradigm from MaxiMin, MiniMax, dispersion and centre, and specifies covering objectives. The set-pairs and IZs (for the covering objectives) are selected prior to the objective selection and is added to the model when the ‘Add obj’ button is pressed. The objective is then displayed in the objective list in the bottom of the panel, where objectives may be removed if desired. Constraints are similarly added to the model in a constraint selection panel in Figure 7.1 (b), and offers the same inclusion and removal functionality as the objective selection panel. Facility-SZ assignments are specified in Figure 7.1 (c), while the resolutions that are selected for consideration by the MRA, as well as its exploitation range, are selected in Figure 7.1 (d). The solution procedure of the GFLoP is prepared in the algorithm and parameter selection panel in Figure 7.1 (e). In this panel, a choice between the NSGA-II, AMOEBA and HGA is made and their parameters are specified². The user may specify the number of runs of the algorithm to be performed and the results are stored in a pre-specified directory.

The GUI provides algorithm-specific feedback in the form of a panel on the right of Figure 7.1. This was a particularly useful tool during preliminary algorithm development for the purposes of monitoring the speed of the algorithms, problems with convergence (in the event that the number of generations reached impractical numbers), amongst other observations. The GUI was also used in a practical evaluation which lead to a research publication [53]. The results of this paper and other implementations of the GUI in the context of demonstrating the generic GFLo framework proposed in this dissertation are the subject of the remainder of this chapter.

7.2 Practical use of the generic GFLo framework

A practical implementation of a real-life GFLoP was pursued in order to gain insight into the practicality and potential for future implementation of the generic GFLo framework proposed in this dissertation. For this purpose, system engineers from the *Square Kilometre Array* (SKA) radio telescope project [121] were approached in order to evaluate the effectiveness of observation camera sites within the MeerKAT sub-project of the SKA. This section serves as a concise presentation of the findings of this evaluation, which was published in [53]. A video of an early version of the concept demonstrator GUI in the process of solving the GFLoP instance solved here is available [51].

7.2.1 Problem overview

The locations of three observation cameras within the MeerKAT project were finalised in June 2011, before any of the SKA radio telescope antennas had been erected. At the time of writing this section, only two of the sixty-four antennas had been erected, and the cameras were yet to be mounted at the selected locations. Their foundations had, however, already been laid and their selected locations were considered final [53]. In 2012, the author was approached by an SKA systems engineer with a request to provide visual illustrations of the coverage that the cameras would provide once they were installed. These illustrations are reported and discussed

²It should be noted that the NSGA-II and HGA population sizes were directly specified in the code. This is why these parameters are absent from the selection panel.

in this section, with consideration given to the antennas' predicted interference in visibility computations.

The MeerKAT observation cameras are used for commercial and informative display purposes, as well as for surveillance (mostly of antenna construction) [53]. The MeerKAT system consists of a core collection of forty eight radio telescope antennas that are closely clustered around a central point, together with sixteen further antennas that are located outside of the core. A top view of the locations of the antennas is provided in Figure 7.3. All the antennas peak at 19.5 metres above ground [53].

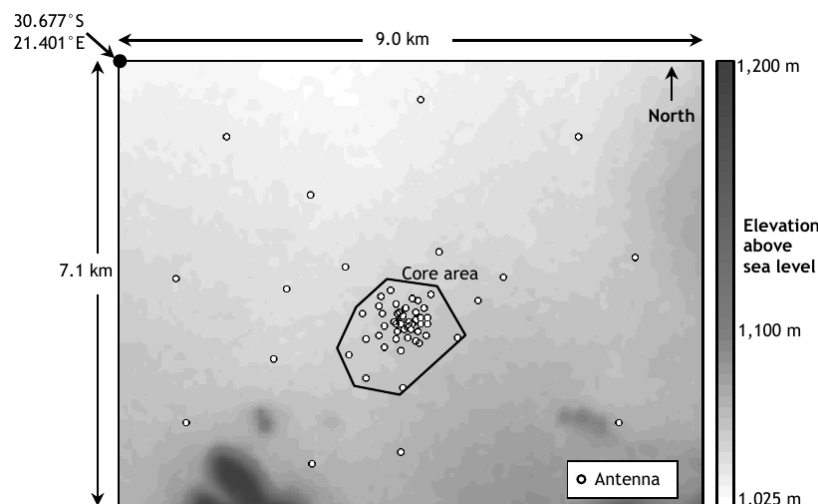


FIGURE 7.3: Top view of the terrain elevation and antenna locations of the sixty four antennas that make up the MeerKAT sub-project of the SKA [53].

The facility location aspect of the task was to place two short-range cameras in the core area, while one long-range camera was to be placed further away, preferably at a high elevation [53]. The main purpose of the core cameras is to provide visibility coverage of the core antennas, while the purpose of the high-elevation camera is to provide coverage of all the antennas, as well as a good general overview of the MeerKAT area [53].

Each of the two short-range cameras has a visibility range of one kilometre and can be rotated to provide a 360-degree horizontal field of view [53]. The long-range camera has a visibility range of seven kilometres and a horizontal field of view of 140 degrees. All three cameras are mounted on structures that elevate them three metres above the terrain surface, thus providing good vantage points. The SKA infrastructure team proposed precise camera placement coordinates for each camera to the SKA systems engineer for investigation; ultimately, these precise coordinates were used [53]. These coordinates were provided to the SKA systems engineer by the system engineering manager of the SKA, who used the viewshed tool of Google Earth Pro [46] to analyse potential sites. He related that multiple points were analysed arbitrarily within specific pre-determined regions. For the high-elevation camera, the search region was restricted to the hill in the southwest of Figure 7.3. Potential core camera sites were analysed within the core area shown in Figure 7.3. The system engineering manager's approach involved aligning the two core cameras in such a manner that a line drawn between them would cross the centrepoint of the core area, thus ensuring that the cameras would provide good visibility with respect to the core antennas when aligned to face each other [53] (this was decided on without a 360-degree rotation being taken into consideration). For all the cameras, consideration was given to terrain characteristics such as peaks, valleys, and slopes, and their influence on the suitability for camera

placement, although the core area is not particularly rough. When analysing specific candidate sites, the visibility from only single points was considered, and no particular attention was given to the performance of the camera configuration in respect of visibility objectives when analysed as a network. In total, the process took nearly two days to complete [53].

The SKA systems engineer then had to visit the sites in order to decide whether they were suitable from a construction point of view (*i.e.*, for the laying of foundations), and whether their practical visibility capabilities were indeed as good as the preliminary analysis suggested. The systems engineer also had to take into account that the cameras would be elevated, and proposed an appropriate elevation height. No visibility performance measures, such as the terrain surface area or the number of antennas visible to the cameras, either individually or as a network, were provided to aid in the process. With regard to the physical suitability of the investigated regions, the appropriateness of the physical terrain for the laying of foundations was also evaluated. The SKA systems engineer recalled that his site visit and associated evaluation process took quite long: 14 hours in total of driving from Cape Town to Carnarvon and back (since then, a three-hour direct return flight has become available), in addition to considerable physical activity, which amounted to roughly an entire day's work. In the end, the sites initially proposed by the SKA infrastructure team were indeed selected for implementation.

Although no specific visibility objectives were communicated to the SKA systems engineer, the two that he mainly considered were (1) to maximise the number of antennas visible to the network, and (2) to maximise the visible terrain surface area. It was not entirely possible to evaluate the extent to which the first objective was achieved by the camera locations, since none of the antennas had been erected at the time, although their future locations were known. With respect to the high-elevation camera, the SKA systems engineer aimed to direct the centre of its field of view towards the centre of the core area [53].

7.2.2 Evaluation of existing camera locations

The camera locations were subsequently evaluated by the analysis tools included in the development of the GFLo framework of this dissertation and the suitability of the actual camera site locations was estimated by performing a visibility analysis of these sites. To simulate visibility interference caused by the antennas, terrain elevation was raised by 19.5 metres at each antenna's future location. Antenna visibility was calculated to the peak of each antenna, thus ensuring at least partial visibility of an antenna from the peak downwards, if the peak is visible. Terrain surface visibility was determined within a boundary that surrounds the sixty four antennas, as indicated in the visibility results of the actual network shown in Figure 7.4. In the visibility computations, terrain elevation is presented as a grid of gridposts that are spaced equally along longitude and latitude. Terrain surface visibility is then measured as the percentage of gridposts in the designated area that are visible from a specific location or a network of locations.

The author's computations revealed that the proposed network of camera locations will be able to observe sixty one of the sixty four antennas' peaks. A visibility percentage of 85.2% is achieved with respect to the terrain surface within the boundary area. The two invisible antennas in the northeast lie outside the visible range of the cameras (an invisibility arc shadow at 7 km is clearly visible from the high-elevation camera). The invisible antenna in the southwest is outside of the core cameras' visible ranges and outside of the high-elevation camera's field of view. The 'shadow lines' that are visible in the core region, radiating in a north-easterly direction, are sections of terrain that are invisible from the high-elevation camera, due to antenna interference.

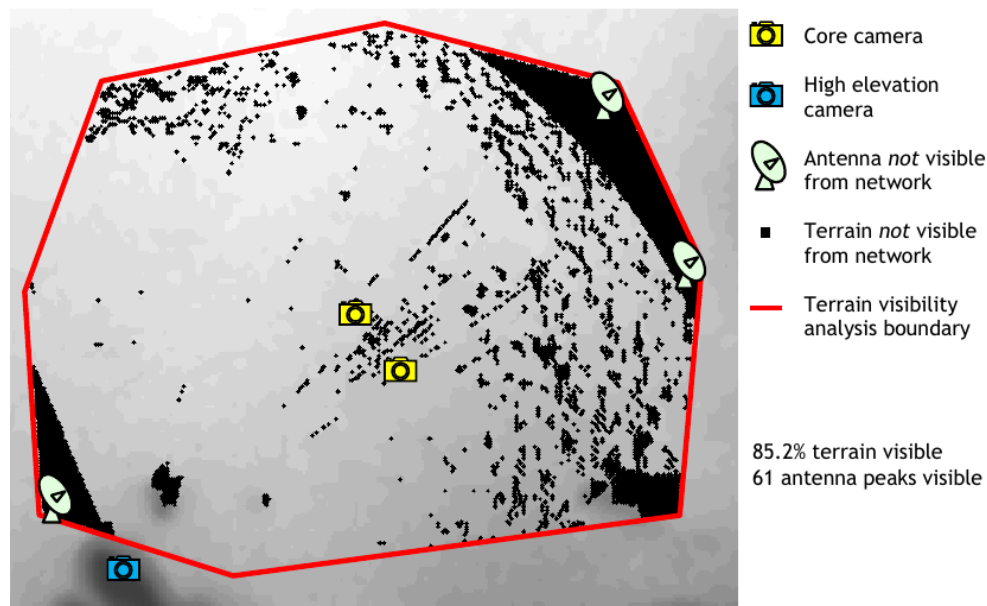


FIGURE 7.4: Locations of the actual observation cameras, and their computed visibility results [53].

7.2.3 Solving the problem using the GFLo framework

Two search zones were identified for this study. The core search zone is the zone around the core antennas earmarked for the placement of the two short-range cameras, and is illustrated in Figure 7.5. The high-elevation search zone is the zone outside of the core camera search zone earmarked for the placement of the high-elevation camera. These two search zones are disjoint. In addition to the hill where the high-elevation camera is to be located, the high-elevation search zone was selected to include higher-lying areas located to the east and southeast of the antennas, as may be seen in the illustration of the search zones in Figure 7.5. These hills were not originally considered for placement by the decision-makers because of proximity concerns, but these possibilities were included in this investigation in a bid to ascertain whether the framework proposed in this dissertation would propose any interesting solutions involving placements in these regions. Terrain where the surface slope exceeded ten degrees was removed from the search zones (only affecting the high-elevation search zone) because areas of excessive slope lead to accessibility and construction complications. The slope of the terrain was determined using the popular ArcGIS software [5]. The areas removed due to excessive slope are visible in Figure 7.5, recognisable as the white areas around the higher-lying areas in the high-elevation search zone. The high-elevation search zone was selected to include lower-lying areas for two reasons. First, the removal of lower-lying areas may result in the exclusion of alternatives that perform well with respect to some visibility objective. Such exclusion may reduce the alternatives (both in terms of solution quality and diversity of camera site locations) presented to the decision-maker. Secondly, if the search zone were reduced by excluding lower-lying areas, additional GIS-based operations and a motivation of the reasons for the selection of certain areas would be required. It is also expected that the algorithms should automatically identify higher-lying areas as candidate site locations if they are indeed included in the best solutions, while simultaneously omitting lower-lying alternatives.

There are 1 545 feasible sites in the core search zone and 97 268 in the high-elevation search zone. Therefore, there are $1\,160 \times 10^9$ possible camera layout combinations, which is an impractical number to evaluate exhaustively. Visibility of each feasible site with respect to the designated

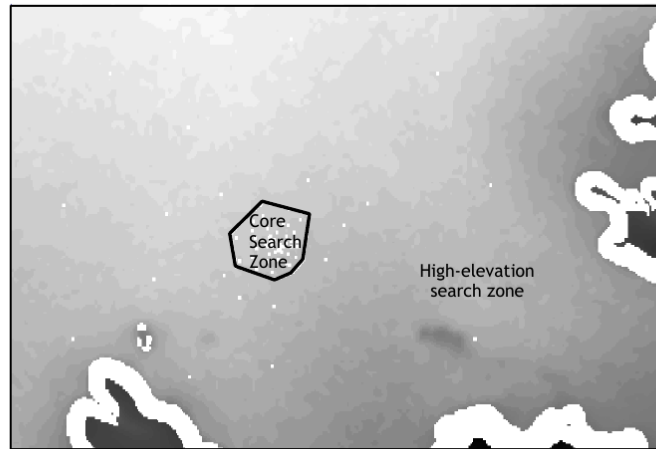


FIGURE 7.5: Search zones identified and explored for camera placement. The small white ‘dots’ indicate antenna locations excluded from the search zones, while the large white areas adjacent to hills represent terrain that was removed due to excessive slope [53].

area of interest was pre-computed, and the visibility results of the sites in the core search zone were sufficiently small to store in memory for fast lookup purposes during the optimisation phase. The acquisition of the necessary data, GIS-based analyses (including visibility), and data formatting for appropriate use within the framework proposed here took roughly two days.

Twenty runs of the NSGA-II were performed for the GFLoP instance. The twenty runs took one hour and 21 minutes to complete, at an average of four minutes and three seconds per run. The multi-resolution approach was employed at resolution levels of three and one arc seconds.

7.2.4 Comparison of actual and proposed camera locations

The non-dominated solutions obtained by twenty runs of the NSGA-II provide interesting camera placement alternatives. A striking result was that the actual camera coordinates proposed by the infrastructure team and verified by the SKA systems engineer (previously shown in Figure 7.4) were remarkably similar to a non-dominated solution returned by the NSGA-II, which provides visibility to sixty one antennas in terms of both camera locations and performance with respect to the visibility objectives. The actual high-elevation camera was located just 31 metres (one gridpost) north of the one found in the non-dominated solution in question, while the core cameras were also very similarly placed. Whereas the number of visible antenna peaks was the same for both placements, the non-dominated solution achieved 1.3% more visible terrain (86.5% compared to 85.2%). It may therefore be concluded that the actual camera network is a very effective solution, and its choice, without any software assistance, is impressive (a conclusion that the MeerKAT team were very pleased with in retrospect).

An interesting alternative to the actual camera sites and those proposed by the non-dominated solution discussed above is one where the high-elevation camera is located 1 029 metres to the southeast of its actual location, in addition to a slightly different orientation of core cameras. This alternative provides visibility to 62 antennas, while 86.0% of terrain is visible, as shown in Figure 7.6. Compared with the actual network, this solution provides slightly better results with respect to both objectives (therefore strongly dominating the actual camera network locations) without requiring much distance between actual and proposed camera locations.

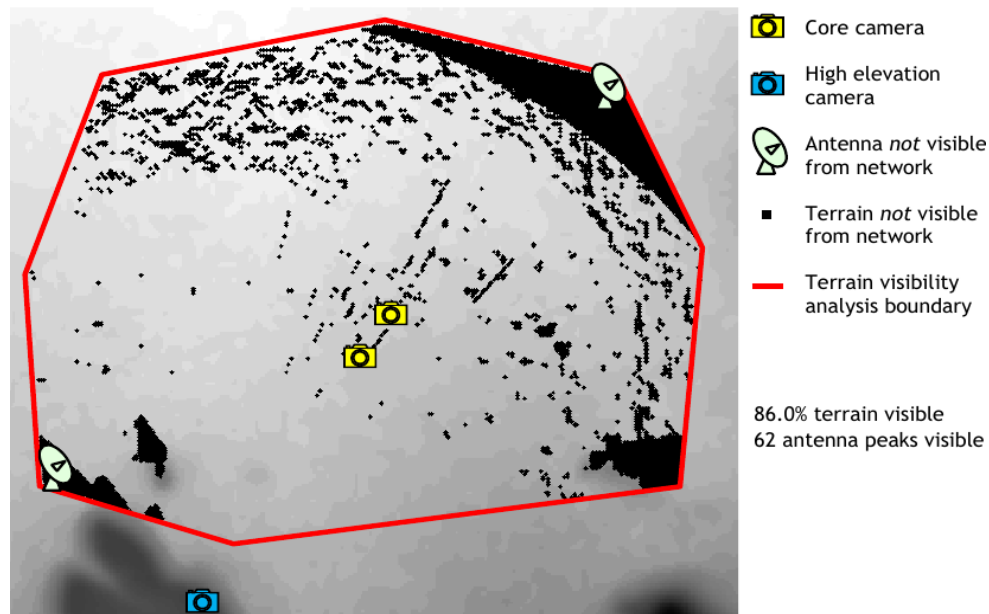


FIGURE 7.6: Visibility results for a non-dominated solution of the attainment front achieved after twenty runs of the NSGA-II, for a solution associated with sixty two visible antenna peaks [53].

The final of the three non-dominated solutions was remarkably different from the actual network and the two previously-discussed non-dominated solutions. In this solution, the high-elevation camera is located in a low-lying region (albeit at a higher location than the terrain in the area of interest due to the terrain sloping upwards towards the south). It is interesting that this solution provided the best visibility with respect to the number of visible antenna peaks: sixty three in total. This is attributed to the fact that visibility was only computed in respect of the peaks of the antennas, and that all the antennas lie within the visibility range of the long-range camera. It was concluded that very few of the antennas will be significantly visible from the cameras in this solution, especially since only 30.5% of the terrain surface is visible.

7.2.5 Decision maker feedback and conclusion

Whereas the actual camera sites were selected without consideration of possible interference of SKA antennas in respect of visibility computations, an advantage of the research in this dissertation is the ability to evaluate placement solutions before the construction of any physical structures that may affect facility performance with respect to the visibility objectives. This advantage follows from including the estimated heights of the SKA antennas in the visibility analyses.

Feedback by the decision-maker tasked with finalising the camera sites on the work reported in this section was positive, and has led to the possibility of further collaboration in respect of SKA facility location sub-projects.

7.3 Small-area problem instances

For the small-area problem instances in this section the same terrain and SZs introduced in §5.4 are considered again. Thirty runs of each of the NSGA-II, AMOEBA and HGA were recorded for each problem instance considered in this section.

For all the problems, the resolution levels employed by the MRA were an initial resolution of 12 arc seconds, with subsequent levels at 6, 3 and the maximum level of 1 arc seconds. When entering a new resolution level, the exploitation range employed by the MRA was selected as a 7×7 square gridpost span centred around each gridpost carried over.

For each of the three GFLoP instances in this section solved over a small-area terrain, three facility-SZ assignment matrices are considered, with the purpose of providing an indication of algorithmic performance in terms of the number of facilities that require placement.

7.3.1 Low-complexity bi-objective GFLoP instance

The GFLoP instance considered in this section is classified as a low-complexity problem, since the evaluation of candidate solutions require memory reads only. The objectives and constraints included in the problem instance are the same as those in §5.4 — the problem differs here in terms of the facility-SZ assignments considered and the fact that the problem is solved using the NSGA-II population halving and MRA search improvements, introduced in §6, for the first time. The objectives are:

1. to **maximise** the minimum average daily solar radiation value of the facilities, and
2. to **maximise** the minimum distance measured between the facilities in SZ 1 from those in SZ 2.

The constraint is that a lower bound of 500 metres is enforced on the distance between any facility pair, in addition to the obligatory placement constraints (3.4)–(3.6).

The facility-SZ assignment matrices for which the problem is solved are:

$$\mathbf{F}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \quad \mathbf{F}_2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \quad \text{and} \quad \mathbf{F}_3 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}.$$

Parameter selection

Based on the parameters used in previous GFLoP instances in this dissertation and a number of sensitivity analyses, the following parameters were selected for each algorithm:

- **NSGA-II:** A population size of 50 before halving was used throughout, with the population similarity threshold percentage and generations chosen as 80% over five generations. The algorithm was terminated when the Pareto similarity threshold percentage reached 90% over ten generations. In respect of the three respective facility-SZ assignment matrices, the crossover rates employed were 0.7, 0.8 and 0.9, together with mutation rates of 0.25, 0.2 and 0.15. Randomly selected single-point crossover was used for \mathbf{F}_1 , while randomly selected two-point crossover was employed for \mathbf{F}_2 and \mathbf{F}_3 .
- **AMOEBa:** The population size limit was set to 50 and the algorithm was terminated after an improvement threshold percentage of 0.5% was observed over ten generations.
- **HGA:** A population size of 20 was employed, together with the same crossover parameters employed by the NSGA-II for the respective facility-SZ assignments. The algorithm was terminated after an improvement threshold percentage of 2% was observed over ten generations.

Presentation of results

The results of the algorithms are presented in Figure 7.7, in the form of their attainment fronts (top) and computation times (bottom) for each of the three respective facility-SZ problem instances. Figure 7.8 (a)–(c) contains the sites evaluated over all thirty runs of each algorithm (the grey markers) and the sites included in the solutions of the attainment fronts of Figure 7.7 (blue markers for SZ 1 facilities and red markers for SZ 2 facilities), for the problem instance corresponding to F_2 . Figure 7.8 (d)–(e) contains the constituent site locations of the solutions in the HGA attainment front in Figure 7.7 (b) that returned the highest value with respect to each of the two objectives, for the F_2 problem instance.

The normalised hypervolume values over thirty runs of each of the algorithms are presented in Table 7.1. The values are normalised by the largest attainment hypervolume obtained, separately for each of the three respective facility-SZ assignment matrices considered. Average values per run in terms of computation time, the number of generations until algorithm termination (over all resolutions), and the number of sites evaluated are contained in Table 7.2.

		Mean	Minimum	Maximum	Attainment
F_1	NSGA-II	0.697	0.457	0.878	0.948
	AMOEBa	0.705	0.478	0.886	0.950
	HGA	0.794	0.630	0.910	1
F_2	NSGA-II	0.631	0.222	0.837	1
	AMOEBa	0.644	0.416	0.827	0.959
	HGA	0.801	0.616	0.944	0.993
F_3	NSGA-II	0.644	0.324	0.898	0.966
	AMOEBa	0.759	0.519	0.923	0.994
	HGA	0.857	0.699	0.976	1

TABLE 7.1: Normalised hypervolume values over thirty runs each of the algorithms for the GFLoP instance in §7.3.1. The values are normalised by the largest attainment hypervolume obtained for each of the three respective facility-SZ assignment matrices considered.

		Run time (s)	Generations	Sites evaluated
F_1	NSGA-II	11.0	201	1 350
	AMOEBa	2.6	210	1 101
	HGA	4.2	75	1 353
F_2	NSGA-II	15.2	272	1 603
	AMOEBa	4.2	258	1 293
	HGA	8.0	115	1 741
F_3	NSGA-II	17.2	293	1 668
	AMOEBa	5.3	287	1 443
	HGA	10.9	140	2 093

TABLE 7.2: Average statistics per run over thirty runs of each of the algorithms applied to the GFLoP instance in §7.3.1.

7.3.2 Low-complexity tri-objective GFLoP instance

The GFLoP instance considered in this section is the same as that considered in Example 6.1 (for the purposes of demonstrating NSGA-II population halving). It is solved here for the three

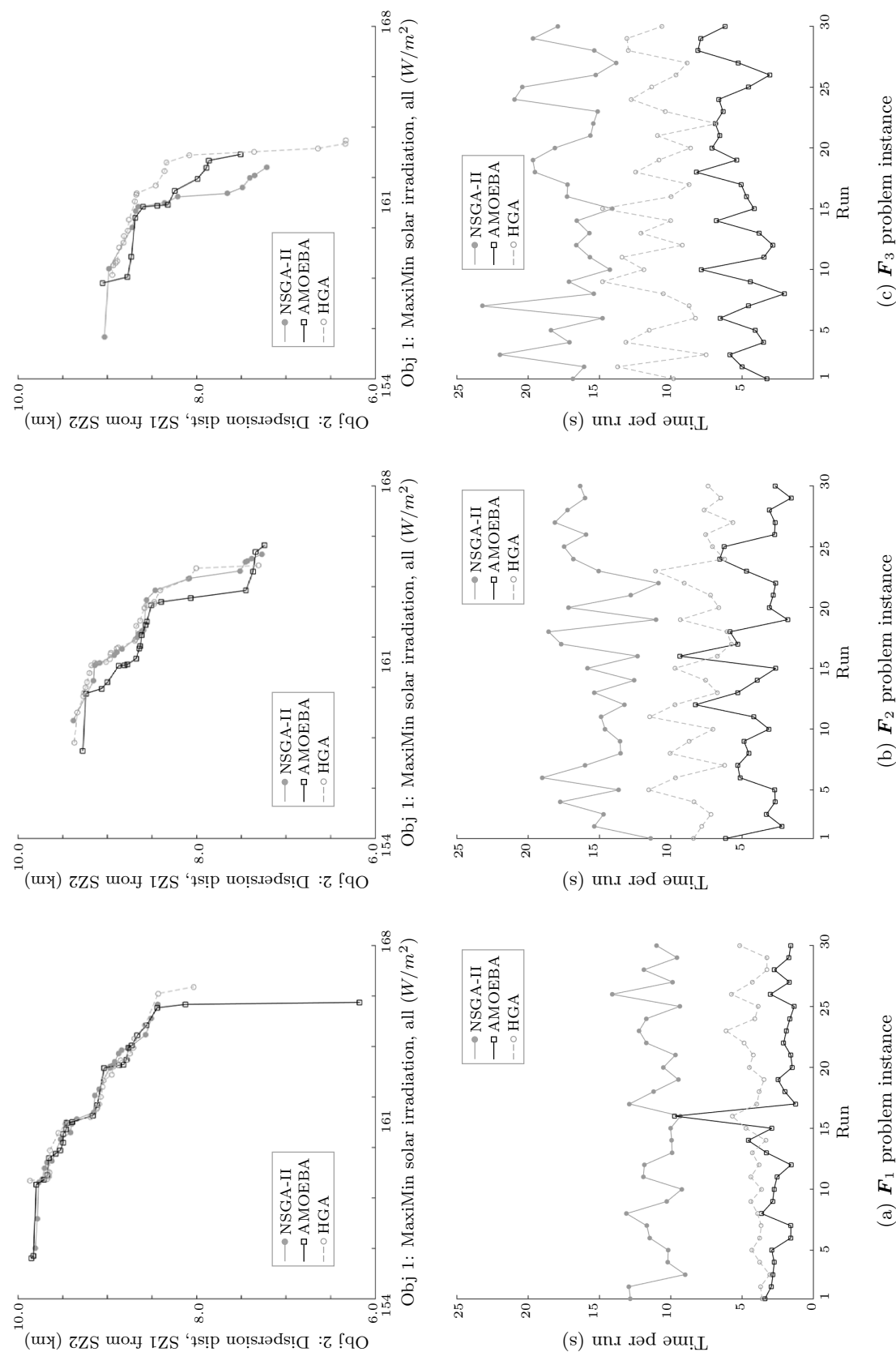


FIGURE 7.7: Results of the three algorithms in the context of the problem instance of §7.3.1, in the form of their attainment fronts (top) and computation times (bottom) for each of the three respective facility-SZ assignment matrices in (a)–(c).

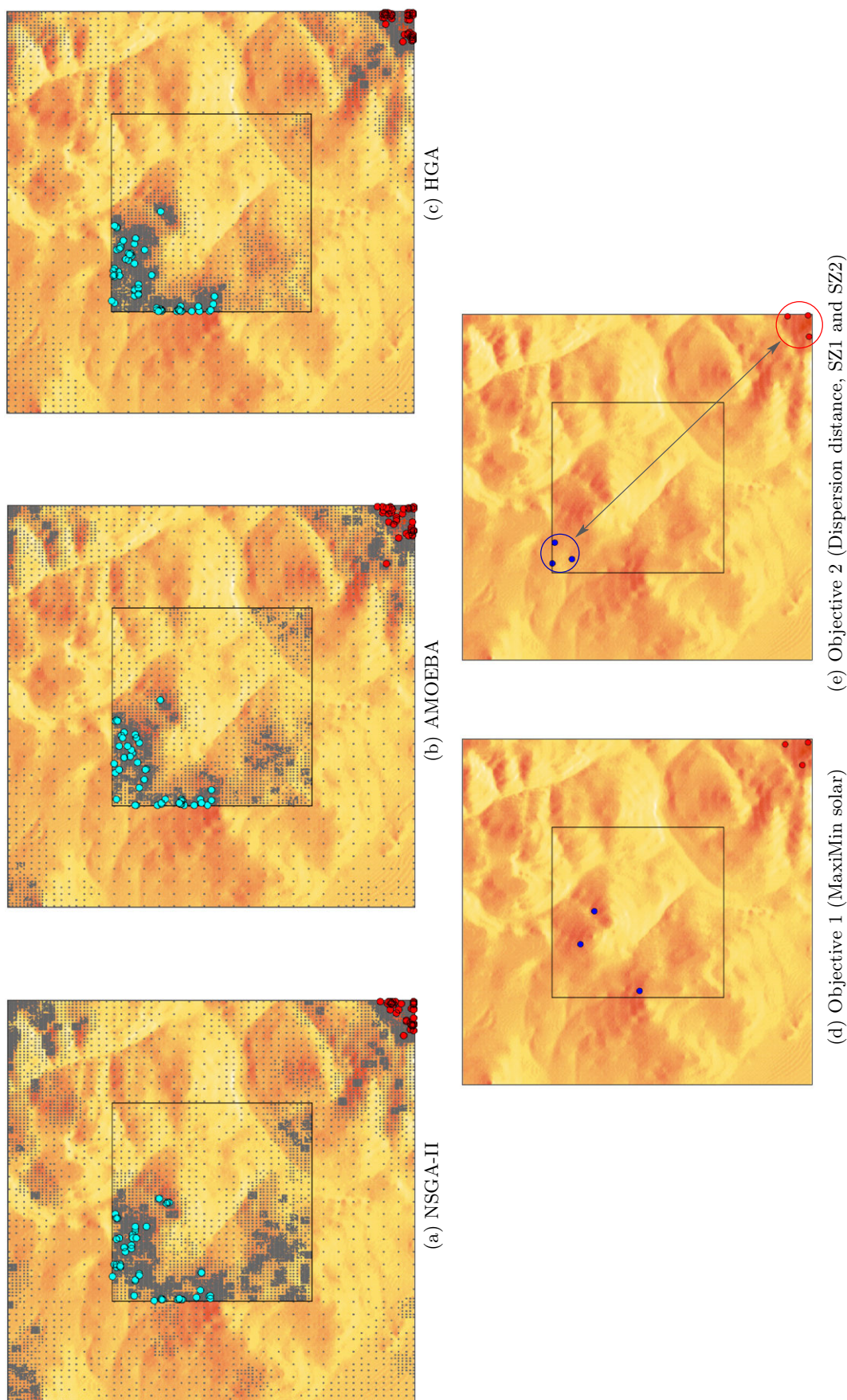


FIGURE 7.8: (a)–(c) Sites evaluated over all thirty runs of each algorithm (the grey markers) and sites included in the solutions of the attainment fronts (blue and red markers), for the problem instance corresponding to \mathbf{F}_2 in the context of §7.3.1. In (d)–(e) the physical site locations of the solutions in the HGA attainment front for the same problem instance and that returned the highest value with respect to each of the two objectives are presented.

assignment matrices and employing the search improvements of the previous chapters. The problem is classified as low-complexity since all data required for the evaluation of solutions are accessed from memory. The addition of a third objective, however, increases the complexity of the problem when compared to the bi-objective instance of the previous section.

The problem is solved for the same three facility-SZ assignment matrices of the previous instance of §7.3.1. The objectives are:

1. to **maximise** the minimum average daily solar radiation value of the six facilities,
2. to **maximise** the minimum distance measured between facilities in SZ 1 and the geographical centre-point of the terrain, and
3. to **minimise** the maximum distance measured between facilities in SZ 2 and the geographical centre-point of the terrain.

Again, a lower bound of 500 metres is enforced on the inter-facility distance between any facility pair, in addition to the obligatory placement constraints (3.4)–(3.6).

Parameter selection

The following parameters were selected for each algorithm:

- **NSGA-II:** A population size of 80 before halving was used throughout, with the population similarity threshold percentage and generations chosen as 70% over five generations, after which the algorithm was terminated when a Pareto similarity threshold percentage of 80% was observed over ten generations. The same crossover and mutation parameters that were used in the GFLoP instance of §7.3.1 were also used here.
- **AMOEBa:** The population size limit was set to 100 and the algorithm was terminated after an improvement threshold percentage of 3% was observed over ten generations.
- **HGA:** A population size of 30 was employed, together with the same crossover parameters employed by the NSGA-II. The algorithm was terminated after an improvement threshold percentage of 5% was observed over ten generations.

Presentation of results

Figures 7.9 (a)–(c) contain the results of the three algorithms for each of the three respective facility-SZ problem instances. Only the attainment front of the NSGA-II is shown for each of the problem instances, because showing all three tri-objective fronts on the same set of axes in an easily distinguishable manner is problematic. Moreover, the same problem was solved by the NSGA-II to illustrate the effect of population halving in the GFLoP instance of Example §6.1 for F_2 , and the results achieved here may be compared to those in Figure 6.4, indicating a clear improvement in solution quality as a result of the additional implementation of the MRA. Figures 7.10 (a)–(c) contain the sites evaluated over all thirty runs of each algorithm (the grey markers) and the sites included in the solutions of the attainment fronts of the three algorithms (blue markers for SZ 1 facilities and red markers for SZ 2 facilities), for the problem instance corresponding to F_3 . In Figure 7.10 (d)–(f), the constituent site locations of the solutions in the HGA attainment front that returned the highest value with respect to each of the three objectives, for the F_3 problem instance, are shown.

Hypervolume values of the algorithms, normalised by the largest attainment hypervolume obtained for each of the three respective facility-SZ assignment matrices considered in the problem, are presented in Table 7.3. Average values per run are contained in Table 7.4.

		Mean	Minimum	Maximum	Attainment
F_1	NSGA-II	0.846	0.783	0.892	0.950
	AMOEBA	0.847	0.762	0.915	0.982
	HGA	0.905	0.836	0.964	1
F_2	NSGA-II	0.827	0.727	0.905	0.975
	AMOEBA	0.779	0.666	0.866	0.943
	HGA	0.879	0.781	0.949	1
F_3	NSGA-II	0.725	0.583	0.896	0.976
	AMOEBA	0.641	0.468	0.809	0.863
	HGA	0.825	0.69	0.918	1

TABLE 7.3: Normalised hypervolume values over thirty runs of each of the algorithms for the GFLoP instance in §7.3.2. The values are normalised by the largest attainment hypervolume obtained for each of the three respective facility-SZ assignment matrices considered.

		Run time (s)	Generations	Sites evaluated
F_1	NSGA-II	12.0	119	1 716
	AMOEBA	3.8	152	1 714
	HGA	10.1	91	2 225
F_2	NSGA-II	21.1	152	1 929
	AMOEBA	4.3	134	1 667
	HGA	12.1	89	2 481
F_3	NSGA-II	19.7	162	2 079
	AMOEBA	5.8	126	1 662
	HGA	13.7	88	2 631

TABLE 7.4: Average statistics per run over thirty runs of each of the algorithms applied to the GFLoP instance in §7.3.2.

7.3.3 High-complexity tri-objective GFLoP instance

In this section, the GFLoP instance of §7.3.1 is re-solved upon the incorporation of an additional maximal covering objective. This problem is classified as a high-complexity problem, since it has three objectives and requires file reads and LOS computations. The evaluation of the solutions also involve memory-intensive computations in respect of the covering objective criteria.

In terms of the MCLP which forms the third objective, the same two facility types that were introduced in the BCLP solved in the MRA example of §6.3.3 are again considered here. The facility types have the following characteristics:

- Facilities of type $f = 1$ have visible distances that extend past the boundaries of the terrain and the heights of these facilities are such that the viewpoints are 5 metres above the ground.
- Facilities of type $f = 2$ have a visible distance of 2 500 metres and a viewpoint height of 3 metres above the ground.

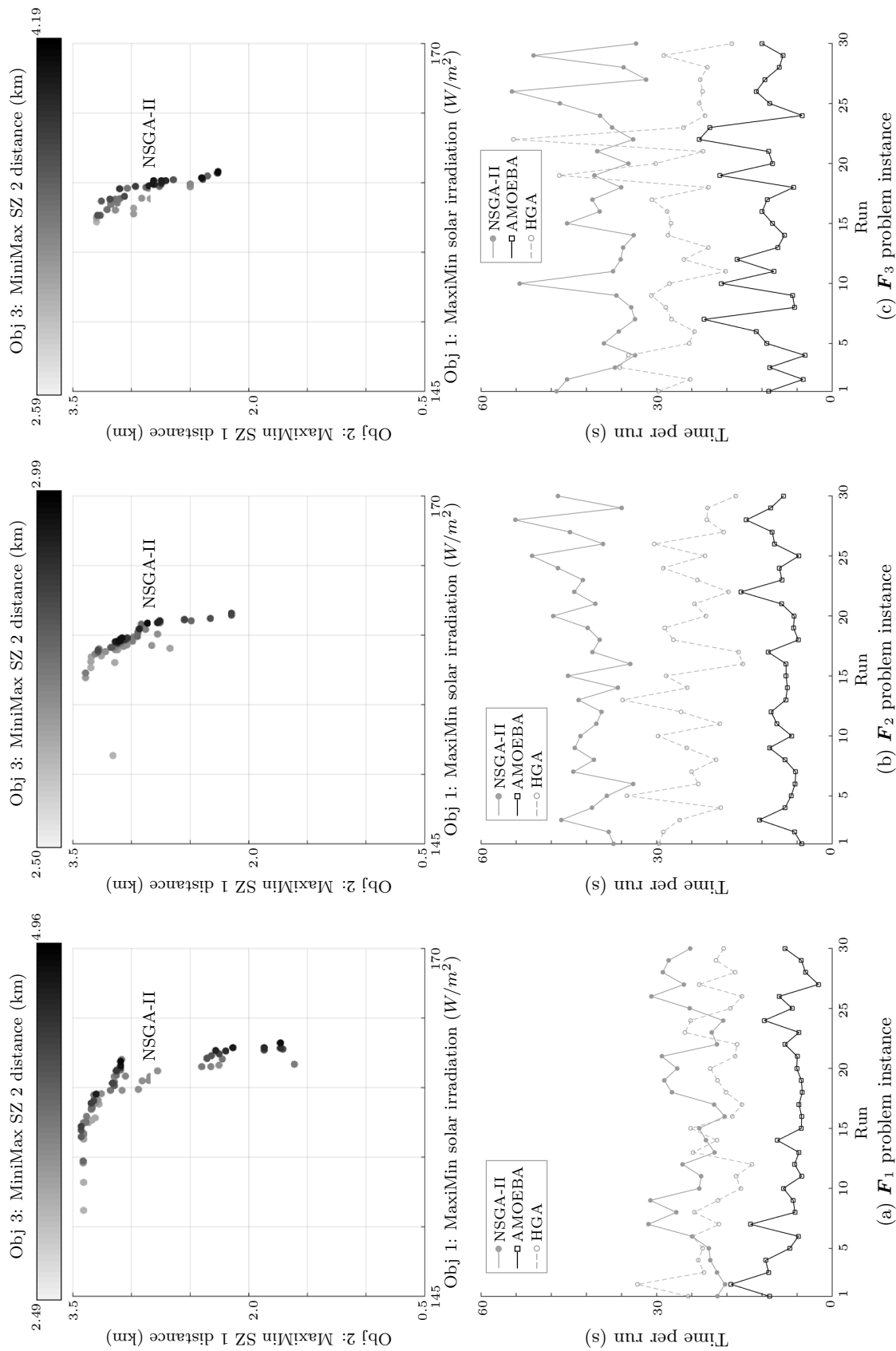


FIGURE 7.9: Results of the three algorithms in the context of the problem instance of §7.3.2, in the form of the NSGA-II attainment fronts (top) and computation times (bottom) for each of the three respective facility-SZ assignment matrices in (a)–(c).

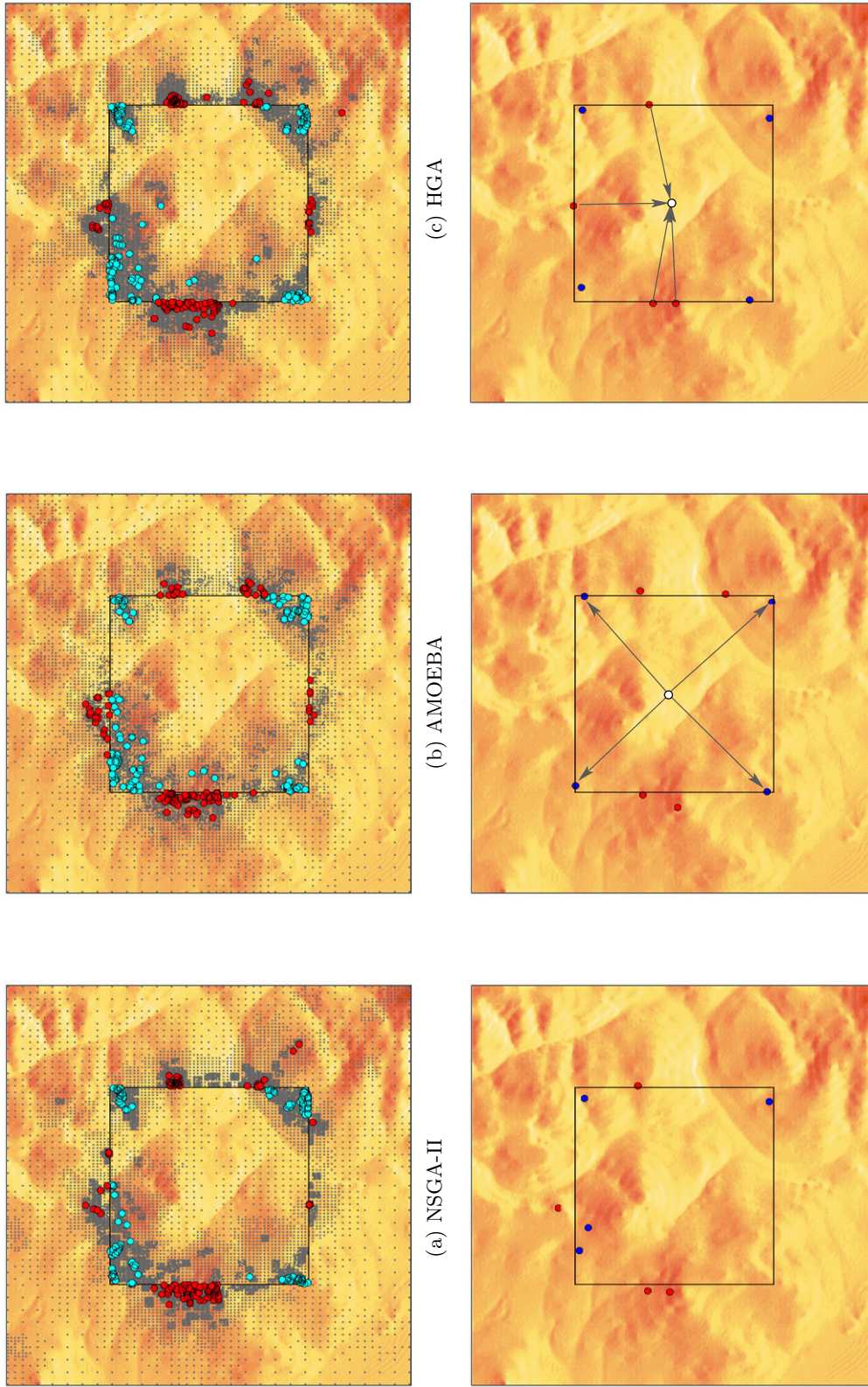


FIGURE 7.10: (a)–(c) Sites evaluated over all thirty runs of each algorithm (the grey markers) and sites included in the solutions of the attainment fronts (blue and red markers), for the problem instance corresponding to \mathbf{F}_3 in the context of §7.3.2. The physical site locations of the solutions in the HGA attainment front for the same problem instance and that returned the highest value with respect to each of the objectives are presented in (d)–(f).

The problem is solved for the following three facility-SZ assignment matrices:

$$\mathbf{F}_4 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{F}_5 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad \text{and} \quad \mathbf{F}_6 = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}.$$

The entire terrain surface is considered equally important and so the IZ comprises all the gridposts in the terrain. The objectives are:

1. to **maximise** the minimum average daily solar radiation value of the facilities,
2. to **maximise** the minimum distance measured between facilities in SZ 1 and SZ 2, and
3. to place all the facilities in such a manner that the number of gridposts visible from *at least one* of the facilities is **maximised**.

The lower bound of 500 metres is again enforced on the inter-facility distance between any facility pair, along with the obligatory placement constraints (3.4)–(3.6).

Parameter selection

The following parameters were used for each algorithm:

- **NSGA-II:** A population size of 80 before halving was used with a population similarity threshold percentage and generations chosen as 70% over five generations. The algorithm was terminated when a Pareto similarity threshold percentage of 80% was observed over ten generations. The same crossover and mutation parameters used in the GFLoP in §7.3.1 and §7.3.2 were also used here.
- **AMOEBA:** A population size limit of 100 was used and the algorithm was terminated when a threshold percentage of 4% was observed over ten generations.
- **HGA:** A population size of 30 was employed, with the same crossover parameters employed by the NSGA-II. Termination occurred after an improvement threshold percentage of 5% was observed over ten generations.

Presentation of results

Figures 7.11 (a)–(c) contain the results of the three algorithms for each of the three respective facility-SZ problem instances. The attainment fronts of the AMOEBA are shown for each of the problem instances for illustrative purposes. The run times shown are metaheuristic computation times only — excluding viewshed computation and file read times. This is because the effect of viewshed computation and file read times were previously discussed in §6.3.3, where it was shown that a high correlation exists between the number of gridposts analysed and the complete run times. The run times of AMOEBA and the HGA are longer than that of the NSGA-II for all three assignment matrices and given the larger number of gridposts that they evaluate per run (indicated in Table 7.6), it may be concluded that their complete run times will be longer than that of the NSGA-II. Figures 7.12 (a)–(c) contain the sites evaluated over all thirty runs of each algorithm (indicated by the grey markers) and the sites included in the solutions of the attainment fronts of the three algorithms for the problem instance corresponding to \mathbf{F}_6 (blue and red markers for facilities of type $f = 1$ and $f = 2$, respectively). In Figure 7.12 (d)–(f), the site locations of the solutions in the HGA attainment front corresponding to the highest

value with respect to each of the three objectives are shown for the F_5 problem instance. Also indicated in black in (d)–(f) are the areas that are *invisible* from the facilities, corresponding to the third, maximal cover objective.

Hypervolume values, normalised by the largest attainment hypervolume obtained for each of the three respective facility-SZ assignment matrices considered, are presented for each algorithm in Table 7.5. Other average statistic values observed per algorithmic run are contained in Table 7.6.

		Mean	Minimum	Maximum	Attainment
F_4	NSGA-II	0.731	0.587	0.855	0.926
	AMOEBA	0.873	0.729	0.975	1
	HGA	0.870	0.741	0.943	0.998
F_5	NSGA-II	0.683	0.499	0.812	0.862
	AMOEBA	0.837	0.763	0.923	0.938
	HGA	0.835	0.710	0.932	1
F_6	NSGA-II	0.645	0.548	0.763	0.846
	AMOEBA	0.817	0.715	0.933	0.965
	HGA	0.814	0.677	0.886	1

TABLE 7.5: Hypervolume values determined over thirty runs of each of the algorithms in the GFLoP instance of §7.3.3, normalised by the largest attainment hypervolume obtained for each of the three respective facility-SZ assignment matrices considered.

		Run time (m)	Generations	Sites evaluated
F_4	NSGA-II	1.8	96	1 759
	AMOEBA	3.8	143	4 611
	HGA	3.7	163	3 354
F_5	NSGA-II	3.3	108	2 067
	AMOEBA	9.5	143	5 939
	HGA	8.0	238	4 317
F_6	NSGA-II	2.2	122	2 304
	AMOEBA	8.7	139	6 085
	HGA	9.9	216	4 786

TABLE 7.6: Average statistics per run over thirty runs of each of the algorithms applied to the GFLoP instance in §7.3.3. The run times are metaheuristic run times and exclude viewshed computation and file read times.

7.3.4 Algorithmic observations for the small-area problems

The results presented for all three GFLoPs instances solved in this section for the small-area terrain returned good results. No unusual results were observed and all three algorithms returned similar results, within similar run times. Following these observations, all three algorithms may be considered acceptable for inclusion in a GFLo framework.

In all the problem instances the HGA achieved the largest hypervolume attainment value, except for one instance in each of §7.3.1 and §7.3.3, where it was marginally outperformed by the NSGA-II and AMOEBA, respectively. From Figures 7.8, 7.10 and 7.12 it is clear that the algorithms find site locations included in the non-dominated solutions of their respective runs in the same

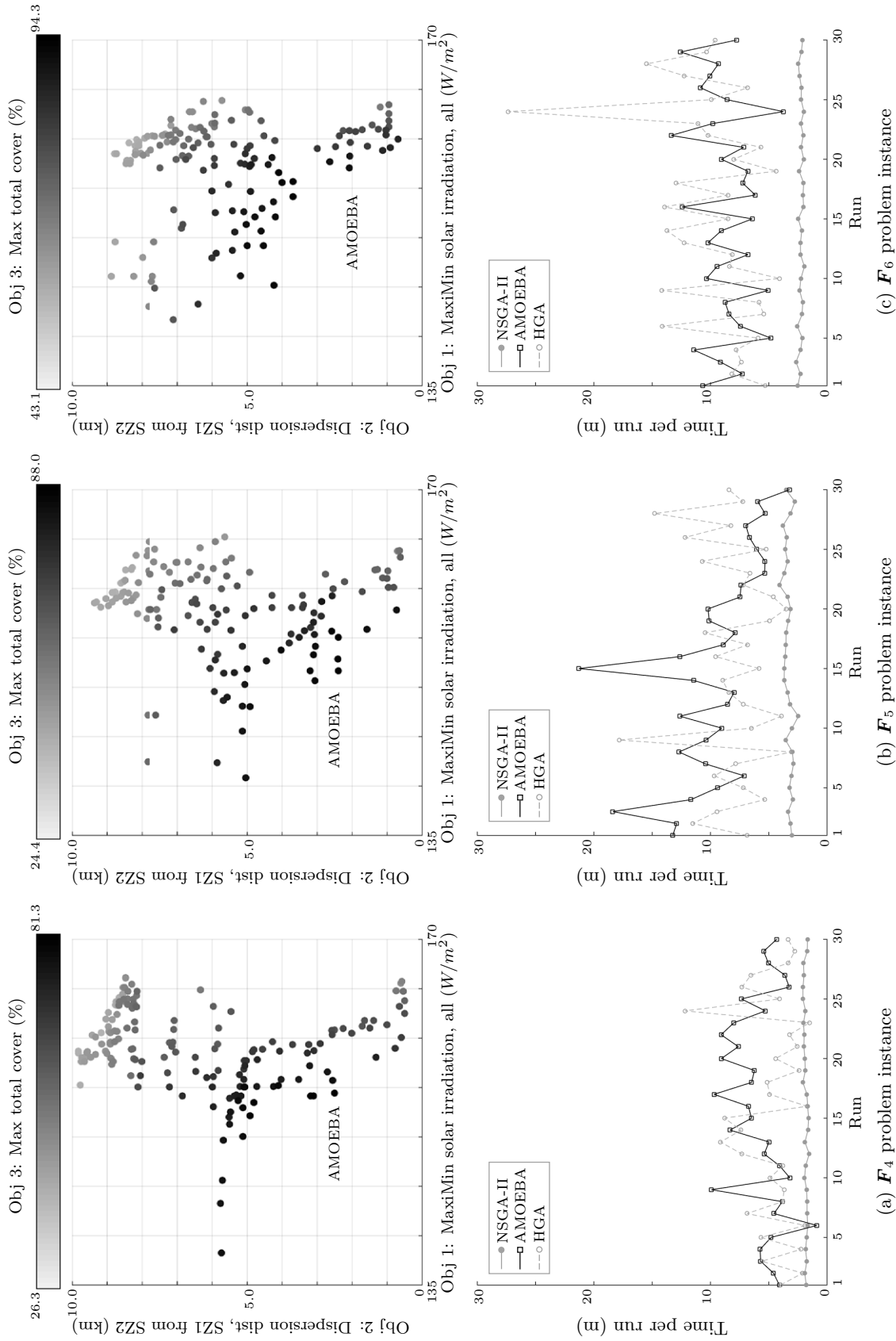


FIGURE 7.11: Results of the three algorithms in the context of the problem instance of §7.3.3, in the form of AMOEBA attainment fronts (top) and computation times (bottom) for each of the three respective facility-SZ assignment matrices in (a)–(c).

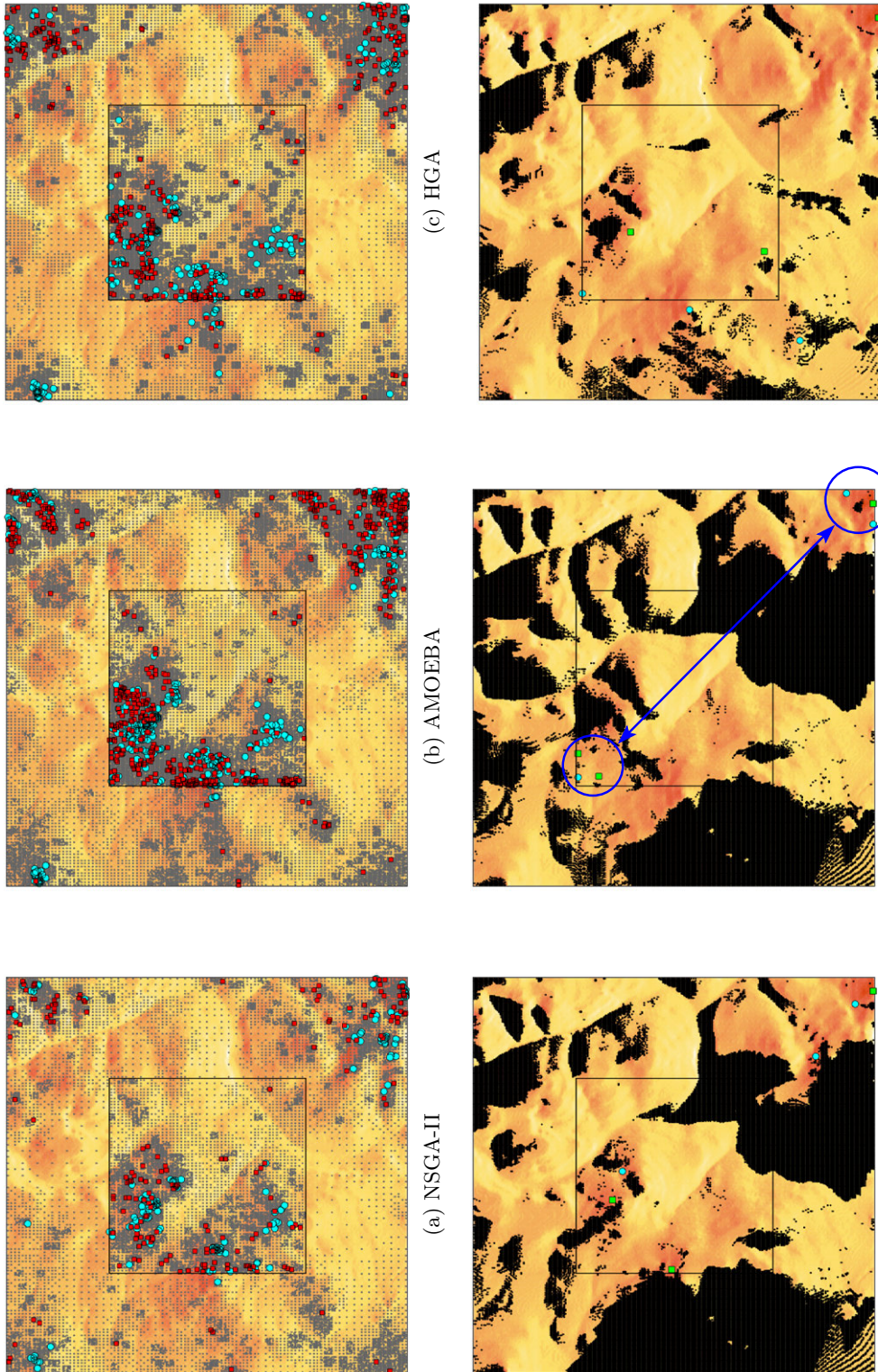


FIGURE 7.12: (a)–(c) Sites evaluated over all thirty runs of each algorithm (the grey markers) and sites included in the solutions of the attainment fronts (blue and red markers for facilities of type $f = 1$ and $f = 2$, respectively), for the problem instance corresponding to \mathbf{F}_6 in the context of the example of §7.3.3. Physical site locations of the solutions in the HGA attainment front for the \mathbf{F}_5 problem instance that returned the highest values with respect to the objectives are presented in (d)–(f). Also indicated in (d)–(f), in black, are the areas that are invisible from the facilities, corresponding to the third, maximal cover objective.

areas, and the efficiency of the MRA is visible from the concentration of the sites evaluated towards these areas. As expected, the run times increase with an increase in the number of facilities placed, although no unusual increases were observed.

In terms of the GFL0P instances of §7.3.1–7.3.2 in which data were only accessed from memory, AMOEBA is consistently the fastest of the three algorithms, while returning solutions of very good quality compared to the NSGA-II and the HGA. The HGA is also faster than the NSGA-II despite the fact that it evaluates a larger number of sites — a result of its combination of search methodologies from the NSGA-II and AMOEBA and also the reason for its superior results.

In terms of the GFL0P instance of §7.3.3, the NSGA-II returned within the shortest computation times, yet yielded solutions of weaker quality than those of AMOEBA and the HGA. This is attributed to its lenient convergence parameters compared to those that were implemented in the other problem instances. Comparing AMOEBA and the HGA, they return solutions of similar quality and within similar run times — although AMOEBA evaluates a larger number of sites in its solution process which, when adding viewshed computation and file read times, would result in it requiring longer computation times than the HGA in the event that both algorithms started with no pre-computed viewshed data.

7.4 Large-area examples

In order to gain more insight into the performance of the algorithms when employed to solve GFL0P instances which involve the placement of facilities over vast terrains, the northern section of South Africa’s Kruger National Park (an area that experiences serious rhino poaching) was selected. The terrain is shown in Figure 7.13, is relatively triangular and measures approximately 100 kilometres from south to north and 68 kilometres from east to west. The elevation ranges from a minimum of 191 metres (darker areas in the figure) to a maximum of 645 metres (lighter areas) and is mostly smooth. Two GFL0P instances are solved in this section. The first, a bi-objective GFL0P with two maximal covering objectives, is solved in §7.4.1. The second example is a tri-objective problem involving both solar and distance-related objectives, and is solved in §7.4.2.

In both problem instances there are two SZs, separated by the black curves in Figure 7.13. In fact, the black curves indicate a proximity of 1 kilometre from rivers. SZ 1 is the terrain area that lies *outside* of 1 kilometre from rivers in the park (the wider areas of terrain between the curves), while SZ 2 is the area of terrain that lies *within* 1 kilometre from rivers (the narrow bands of terrain between the curves). The motivation behind the SZ area selections is that wildlife often gather in herds near water sources and remote observation of the river zone may therefore reveal interesting animal behaviour or events. Gridposts where the ground slope exceeds 10 degrees were removed from both SZs, while the areas of terrain covered by rivers were removed from SZ 2 (the removed areas are not shown in the figure).

In terms of the implementation of the MRA, when entering a new resolution level the exploitation range employed was selected as a 21×21 square gridpost span centred around each gridpost carried over. Different MRA resolution levels were employed in the two problems — both started at a level of 12 arc seconds and ended at a level of 3 arc seconds. At the final resolution there are 415 197 and 104 023 candidate locations in SZ 1 and SZ 2, respectively (compared to a combined total of 125 125 at a resolution of 1 arc second for the SZs in the preceding examples).

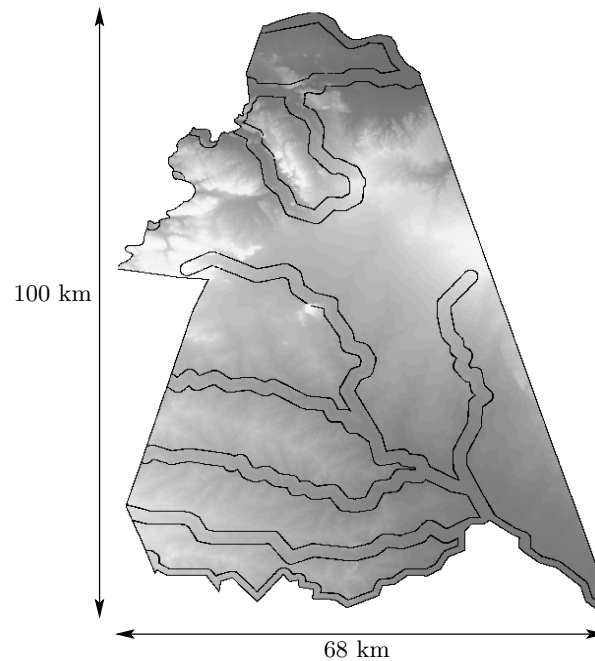


FIGURE 7.13: Terrain considered in the large-area problem instance of §7.4. The elevation increases from darker to lighter areas.

7.4.1 High-complexity bi-objective GFLoP instance

In this section ten runs of each of the algorithms of §6 are used to solve a GFLoP instance with two maximal covering objectives, each determined with respect to one of two different IZs. The first IZ is the entire terrain surface in Figure 7.13, while IZ 2 is all terrain enveloped between the river boundary lines — *i.e.* terrain within 1 kilometre from rivers. The gridposts in IZ 2 are a subset of the gridposts in IZ 1. The two facility types considered in the problem have visibility/detection requirements, with the following specifications:

- Facilities of type $f = 1$ have a visible distance of 10 000 metres and a viewpoint at 10 metres above the ground — these are to be placed in SZ 1.
- Facilities of type $f = 2$ have a visible distance of 3 000 metres and a viewpoint height of 3 metres above the ground and are to be placed in SZ 2.

The facility-SZ assignment matrix is specified as follows:

$$\mathbf{F}_7 = \begin{bmatrix} 3 & 0 \\ 0 & 8 \end{bmatrix}.$$

The objectives in this GFLoP instance are:

- to place the facilities in such a manner that the number of gridposts in IZ 1 which are visible from *at least one* of either type of facility is **maximised**, and
- to place the facilities so that the number of gridposts in IZ 2 which are visible from *at least one* of the facilities of type $f = 2$ is **maximised**.

Two lower bounds are enforced on inter-facility distance, along with the obligatory placement constraints (3.4)–(3.6). The first bound is enforced between the two different facility types in

the respective SZs, while the second bound is enforced between the facilities of type $f = 2$ which are located in the river area. The constraints are:

- that a lower bound of 3 000 metres must be maintained between facilities of type $f = 1$ and facilities of type $f = 2$ (*i.e.* the coverage of a facility of type $f = 2$ does not reach a facility of type $f = 1$), and
- that a lower bound of 6 000 metres must be maintained between pairs of facilities of type $f = 2$ (*i.e.* the coverage of facilities of type $f = 2$ do not overlap).

Parameter selection

The MRA was implemented at two resolutions, namely with a starting resolution of 12 arc seconds and followed by a resolution of 3 arc seconds. The following parameters were selected for the algorithms:

- **NSGA-II:** A population size of 80 before halving was used. The population similarity threshold percentage and generations were chosen as 60% over five generations. The algorithm was terminated when a Pareto similarity threshold percentage of 80% was observed over five generations. The crossover probability was set at 0.9 (randomly selected two-point), with a mutation rate set at 0.25.
- **AMOEA:** A population size limit of 50 was used and the algorithm was terminated when a threshold percentage of 1.5% was observed over ten generations.
- **HGA:** The population size was set at 20, with the same crossover parameters of the NSGA-II. The algorithm terminated after an improvement threshold percentage of 5% was observed over ten generations.

Presentation of results

The attainment fronts of the algorithms are illustrated in Figure 7.14. Figure 7.15 contains the sites evaluated during the ten runs of each algorithm (the grey markers) and sites included in the solutions of the attainment fronts of Figure 7.14, indicated by blue markers for facilities of type $f = 1$ placed in SZ 1 and red markers for facilities of type $f = 2$ placed in SZ 2. Figure 7.16 is a graphical illustration of cover achieved by the solutions in the HGA attainment front in Figure 7.14 which returned the highest value with respect to each of the two objectives. The cover achieved with respect to IZ 1 by all the facilities in respect of the first objective is indicated by blue and red areas combined, while the cover achieved by facilities of type $f = 2$ with respect to IZ 2 in respect of the second objective is indicated by the red areas.

The normalised hypervolume values achieved over ten runs of each of the algorithms are presented in Table 7.7. The values are normalised by the attainment hypervolume of the HGA, since it was the largest. Average statistic values per run are contained in Table 7.8. As in the preceding example, the computation times shown in the table are metaheuristic computation times only.

7.4.2 Low-complexity tri-objective GFLoP instance

Twenty runs of each algorithm was performed when solving another GFLoP instance involving two MaxiMin objectives and one centring objective. Only one type of facility is considered in

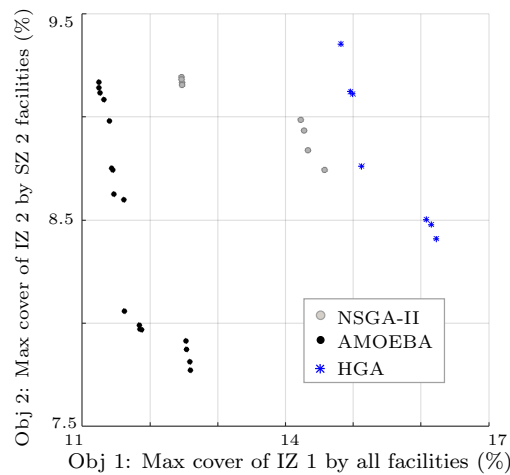


FIGURE 7.14: Attainment fronts achieved after ten runs of each of the three algorithms in the context of the GFLoP instance of §7.4.1.

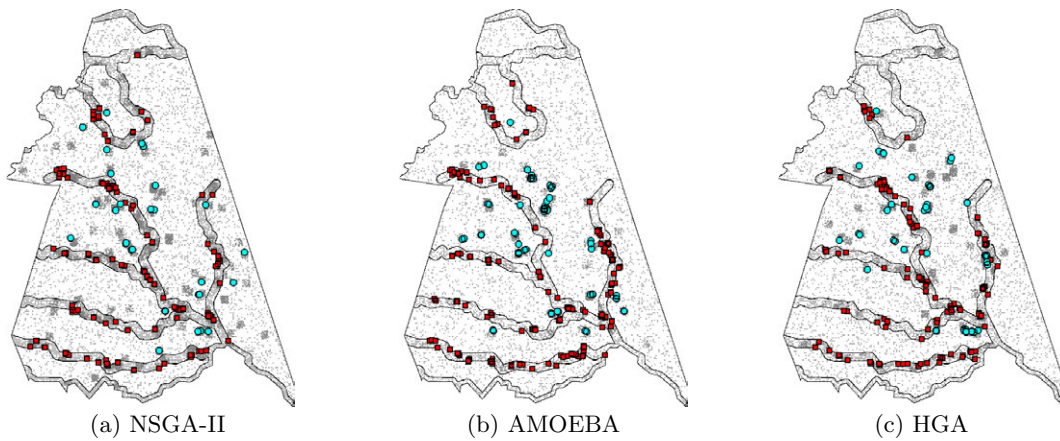


FIGURE 7.15: Sites evaluated during the ten runs of each algorithm (the grey markers) and sites included in the solutions of the attainment fronts of Figure 7.14, indicated by blue markers for facilities of type $f = 1$ placed in SZ 1, and red markers for facilities of type $f = 2$ placed in SZ 2.

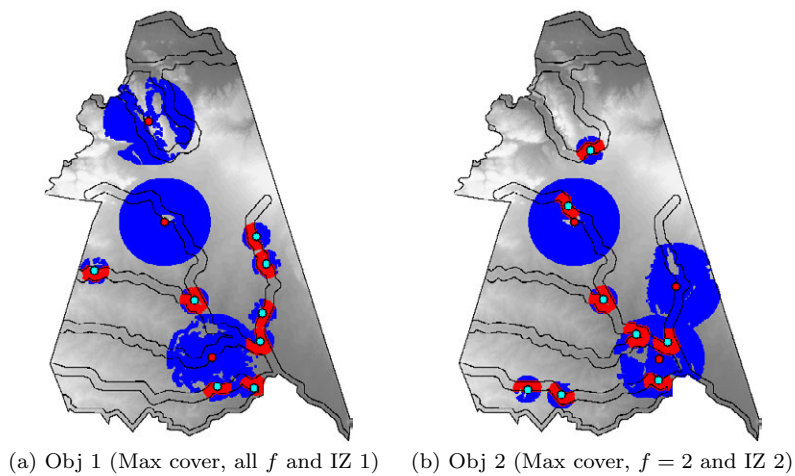


FIGURE 7.16: Cover achieved by the solutions in the HGA attainment front in Figure 7.14 that returned the highest value with respect to each of the two objectives in the context of the GFLoP instance of §7.4.1. Cover achieved with respect to IZ 1 by all the facilities is indicated by blue and red areas combined, while cover achieved by facilities of type $f = 2$ with respect to IZ 2 is indicated by the red areas.

	Mean	Minimum	Maximum	Attainment
NSGA-II	0.303	0.119	0.684	0.713
AMOEBa	0.139	0.008	0.290	0.348
HGA	0.324	0.103	0.868	1

TABLE 7.7: Normalised hypervolume values over ten runs each of the algorithms for the GFLoP instance in §7.4.1. The values are normalised by the attainment hypervolume of the HGA, since it was the largest.

	Run time (m)	Generations	Sites evaluated
NSGA-II	73	86	2716
AMOEBa	38	108	1085
HGA	55	38	1684

TABLE 7.8: Average statistics per run for each of the algorithms applied to the GFLoP instance in §7.4.1. Run times shown are metaheuristic run times, and exclude viewshed computation and file read times.

this instance, placed according to the following facility-SZ assignment matrix:

$$\mathbf{F}_8 = \begin{bmatrix} 3 \\ 8 \end{bmatrix}.$$

The objectives are:

- to place the facilities so that the minimum distance measured between the three facilities in SZ 1 and rivers is **maximised**,
- to place the facilities in such a manner that the minimum average daily solar irradiation value of the eight facilities in SZ 2 is **maximised**, and
- to place the facilities so that the maximum distance measured between the facilities in SZ 1 from those in SZ 2 is **minimised**.

The same lower bounds that were enforced in §7.4.1 are also enforced for this GFLoP instance.

Parameter selection

The MRA was implemented at three resolutions levels, starting at 12 arc seconds, and followed by 6 and then 3 arc seconds. The following parameters were selected for the algorithms:

- **NSGA-II:** The population size was set at 100 before halving, with a population similarity threshold percentage and generations of 60% over five generations. The algorithm was terminated when a Pareto similarity threshold percentage of 90% was observed over five generations. Randomly selected two-point crossover, with a probability of 0.9, and a mutation rate of 0.25 was chosen.
- **AMOEBa:** A population size limit of 50 was used and the algorithm was terminated when a threshold percentage of 4% was observed over ten generations.
- **HGA:** A population size of 30 was used, employing the crossover parameters of the NSGA-II. The algorithm terminated following an improvement threshold percentage of 5% over ten generations.

Presentation of results

Figures 7.17 (a)–(c) present the attainment fronts (top) and sites evaluated (bottom) for the three respective algorithms. The grey markers indicate all the sites evaluated over the twenty runs of each algorithm, while the blue and red markers indicate the locations of the sites in SZ 1 and SZ 2, respectively, included in the final attainment fronts achieved by the algorithms. In Figure 7.18 (a)–(c), the site locations of the solutions in the HGA attainment front that returned the highest value with respect to each of the three objectives are shown, indicated by blue markers for facilities in SZ 1 and green markers for facilities in SZ 2. Distance from rivers in SZ 1, of relevance to the first objective, is indicated increasing in value from grey to black — *i.e.* black areas are more desirable. The average daily solar irradiation values of SZ 2, of relevance to the second objective, are indicated increasing in value from yellow to red — *i.e.* red areas are more desirable.

Normalised hypervolume values of the algorithms are presented in Table 7.9. The values are normalised by the HGA attainment hypervolume value, the largest achieved. Average statistics per run are presented in Table 7.8 for the three algorithms.

	Mean	Minimum	Maximum	Attainment
NSGA-II	0.508	0.355	0.723	0.857
AMOEBa	0.621	0.369	0.800	0.864
HGA	0.696	0.561	0.863	1

TABLE 7.9: Normalised hypervolume values over twenty runs each of the algorithms for the GFLoP instance in §7.4.2. The values are normalised by the attainment hypervolume of the HGA, since it was the largest.

	Run time (s)	Generations	Sites evaluated
NSGA-II	37.3	233	5 843
AMOEBa	33.1	147	7 315
HGA	37.8	133	7 551

TABLE 7.10: Average statistics per run for each of the algorithms applied to the GFLoP instance in §7.4.2.

7.4.3 Algorithmic observations for the large-area problems

The algorithms returned satisfactory results for the two GFLoP instances solved in this section over large portions of terrain. No unusual results were observed and the algorithms returned similar results, within similar run times. The only result that stood out was that of AMOEBA for the bi-objective problem, returning weak results, but within short computation times compared to those of the other two algorithms (see Tables 7.7 and 7.8), suggesting weak convergence parameters. Following these results, the three algorithms may also be considered acceptable for inclusion in a GFLo framework for large-area problems.

In both problems the HGA achieved the largest hypervolume attainment value. From Figures 7.15 and 7.17 it is clear that the algorithms find site locations included in the non-dominated solutions of their respective runs in similar areas, and the efficiency of the MRA is once again visible from the concentration of the sites evaluated towards these areas. The attainment fronts of AMOEBA and the HGA exhibit much better diversity and spread in solutions than that of

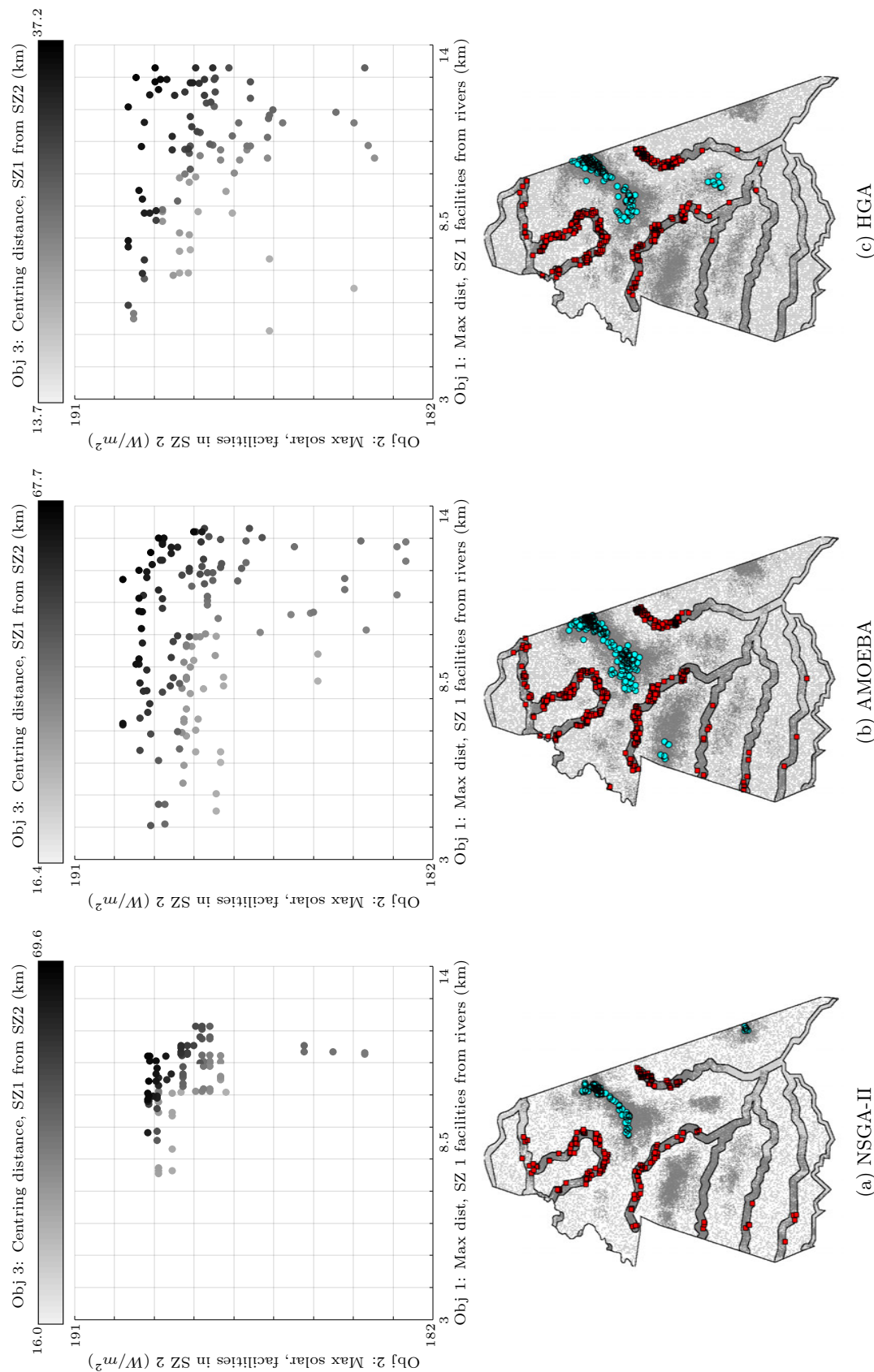


FIGURE 7.17: Attainment fronts (top) and sites evaluated (bottom) for the three respective algorithms in the context of the GFLoP instance of §7.4.2. Grey markers indicate all the sites evaluated over the twenty runs of each algorithm, while blue and red markers indicate locations of the sites in respectively SZ 1 and SZ 2 included in the attainment fronts achieved by the algorithms.

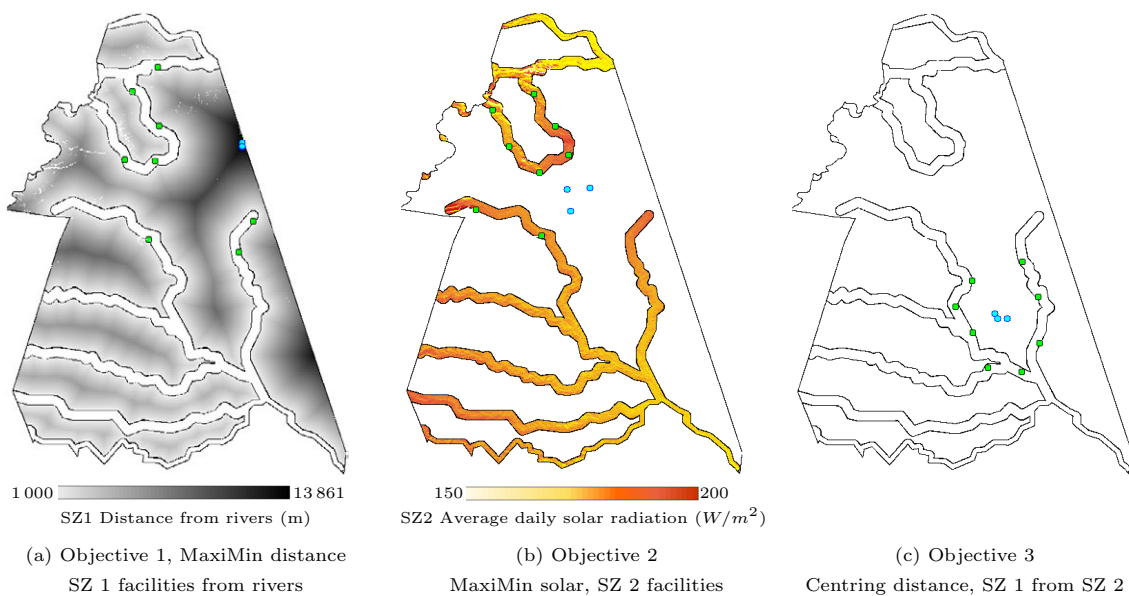


FIGURE 7.18: Site locations of the solutions in the HGA attainment front that returned the highest value with respect to each of the three objectives in the context of the GFLoP instance of §7.4.2. Blue markers indicate facilities located in SZ 1, while green markers indicate facilities placed in SZ 2. The average daily solar irradiation values of SZ 2 are indicated increasing in value from yellow to red.

the NSGA-II. This is likely due to the fact that they evaluate more candidate sites, yet this is achieved in similar run times (faster by AMOEBA). This is important since the data were accessed from memory only, and so the HGA and AMOEBA exhibit superior search efficiency in these large problem instances.

7.5 Chapter summary

A generic GFLo DSS architecture and its components were proposed in §7.1. An overview of the architecture components was given in §7.1.1 and a concept demonstrator GUI of the optimisation component of this architecture was described in §7.1.2.

The implementation of the GFLo framework put forward in this dissertation, including the use of the concept demonstrator of §7.1.2, was discussed in §7.2. The study provided preliminary proof of the GFLo framework's usefulness and the decision makers involved in the project were impressed with the results.

A variety of additional GFLoP instances were solved in this chapter in order to showcase the workability of the GFLo framework put forward in this dissertation. The first set of these GFLoP instances were encountered in §7.3 with the solution of three problems on small portions of terrain. This comprised one bi-objective and two tri-objective problems, with three facility-SZ assignment matrices and varying placement criteria considered for each problem instance. The results returned by the algorithms were presented graphically in terms of a selection of attainment fronts achieved, in addition to the sites that were evaluated and included in attainment front solutions. Statistics in terms of hypervolume values and average run times, number of generations and locations evaluated per run for each of the algorithms were provided in tabular form for the three respective GFLoP instances. Observations in respect of the performance of the algorithms in terms of solution quality and computation times were made in §7.3.4.

Bi-objective and tri-objective GFLoP instances were solved on large portions of terrain in §7.4. The two problems involved the placement of eleven facilities within two SZs, and the bi-objective problem involved the covering of two IZs. As for the small-area problem instances, the results obtained by the algorithms were presented graphically in terms of attainment fronts achieved, and sites evaluated. Statistics in terms of hypervolume values and average run times were also provided in tabular form for the two respective GFLoPs. The performance of the algorithms were discussed in §7.4.3 in terms of solution quality and computation times.

CHAPTER 8

Conclusion

Contents

8.1 Dissertation summary	137
8.2 Contributions of this Dissertation	138
8.3 Ideas for future work	140

A summary of the research conducted in this dissertation is presented in the first section of this chapter. This is followed by an appraisal of the dissertation contributions in §8.2, and suggestions for future work are made in §8.3.

8.1 Dissertation summary

A brief introduction to the current scope and applications of GFLoPs and their solution methodologies was provided in Chapter 1. Limitations of GFLo solution approaches were described and the most notable of these limitations were identified to be (1) the fact that only one type of facility is generally included in the solution approach, and (2) that the solution approaches in the literature to various GFLoPs exhibit various similarities, yet are designed in a purpose-specific manner. This provided the motivation behind the contributions made in this dissertation. The objectives and scope of this study were then outlined, and this was followed by a brief description of the dissertation organisation.

Chapter 2 opened with an introduction to the facility location branch of operations research, followed by a literature review on classical facility location models and their prevalence and application potential in the context of GFLo, in accordance with Dissertation Objective I. This was followed by an investigation into MOO approach considerations for solving GFLoPs, in addition to a survey of algorithms that are generally employed in GFLo solution frameworks, in fulfilment of Dissertation Objective II.

The structure of a generic GFLo database was proposed in Chapter 3. This included descriptions of the facility and geospatial databases that make up the GFLo database, and this was followed by a description of different types of geospatial analyses in respect of geospatial data (in fulfilment of Dissertation Objective IV(a)). Typical GFLo criteria were considered and classified into generic classes according to similarities that they exhibit (in accordance with Dissertation Objectives III and IV(b)), and this was followed by a suggestion as to how the GFLo environment may be modelled mathematically in a generic manner, in partial fulfilment of Dissertation Objective IV(c).

Dissertation Objective IV(c) was achieved in Chapter 4 where generic mathematical formulations of objectives and constraints that may be considered as building blocks in GFLo models were presented. The chapter closed with the proposal of a generic GFLoP formulation framework which serves the purpose of providing the necessary guidelines to formulate a GFLoP model which is not limited to the customary procedure in the GFLo literature, where models are formulated according to specific problem and facility types.

Chapter 5 opened with an introduction to MOO concepts that are of relevance to the GFLo framework, in partial fulfilment of Dissertation Objective II. These concepts were employed in the following section, in which two distinctly different GFLo algorithms, namely the well-known NSGA-II and the novel AMOEBA, were respectively reviewed and proposed, in partial fulfilment of Dissertation Objective IV(d).

A number of search improvements of specific relevance in the context of GFLo were introduced in Chapter 6. A hybrid implementation of the NSGA-II and AMOEBA was proposed, and additional search improvements that are of relevance to all three algorithms were then proposed (in partial fulfilment of Dissertation Objective IV(d)).

A concept demonstrator of a computerised decision support tool, based on the framework of Chapter 4 and designed for the purposes of this study, was described in Chapter 7, in fulfilment of Dissertation Objective V. Using the decision support concept demonstrator, a variety of GFLoP instances were solved in order to demonstrate the workability of the GFLo framework proposed in this dissertation, in accordance with Dissertation Objective VI and in partial fulfilment of Dissertation Objective IV(d). These GFLoP instances included three problems instances on a small-area terrain (one bi-objective and two tri-objective instances). This was followed by bi-objective and tri-objective GFLoP instances solved on a large-area terrain.

8.2 Contributions of this Dissertation

An attempt is made in this section to clarify the contributions made in this dissertation with respect to advancement of the field of GFLo.

Contribution 8.1 *The design of a generic MO GFLo framework.*

The main contribution of this dissertation was the proposal of a MO GFLo framework which is able to solve various existing GFLoPs in a generic manner, in addition to presenting novel facility location solving opportunities. This was achieved through the generic classification of typical GFLo criteria, which provided the theoretical foundation required for the generic mathematical modelling of the geospatial environment. Prior to the GFLo mathematical model building blocks proposed in this dissertation, the GFLoPs considered for inclusion in the GFLo framework only involved one type of facility, and a single SZ and IZ included in the model, with the only exception being the multi-facility dispersion model [16]. The GFLo criteria considered in the framework design allows for GFLoPs to be modelled in a manner that is independent of the type of facility involved in the problem, and allows for a combination of multiple facility types, SZs and IZs to be included in the model — for all the classical FLoPs considered. Furthermore, classical FLoP constructs were modelled in such a manner that multiple objectives and constraints may be easily added to these models, thus facilitating the solution of GFLoPs that were not previously possible, or conceptualised.

Contribution 8.2 *The design and implementation of the novel AMOEBA algorithm.*

The AMOEBA algorithm is novel in concept, since it combines a fast and simple single-solution, swap-based search strategy, but has been adapted for implementation within a MO, population-

based environment. The algorithm requires minimal parameter testing and tuning: a maximum population size limit is required, along with convergence parameters which are standard parameters for most MOEAs. The algorithm starts with a single, randomly generated solution only, and rapidly spreads into a population of solutions, much like a virus. In multiple test studies, the algorithm consistently outperformed the NSGA-II and HGA in terms of computation speed for problem instances in which data are accessed from memory — this was achieved while obtaining results of consistently similar quality. Furthermore, the algorithm is simple in theory and also in its practical implementation compared to the NSGA-II and other evolutionary algorithms, and may easily be integrated with existing NSGA-II applications to form the HGA (which was consistently shown to achieve superior results to those of the NSGA-II and AMOEBA in the worked examples).

Contribution 8.3 *The design, implementation and testing of the novel MRA.*

The newly proposed MRA exhibits superior performance compared to the standard approach of solving a GFLoP at a single resolution (in which only the highest available resolution data are typically considered). The MRA yields consistently superior solutions — both in terms of solution quality and also regularly in terms of computation time. The reason for the superior results of the MRA is its ability to identify potentially well-performing areas in the SZs and its avoidance of unnecessarily evaluating solutions in poorer areas. As a result, fewer candidate locations are evaluated by the MRA, which typically results in shorter computation times. Supplementary results in supporting documentation [55] has illustrated that the MRA is effective and superior to the SRA in the context of problems involving ‘rough’ terrain, terrain that covers very large areas, and for problems in which data are accessed by file reads. A paper describing the MRA and illustrating its superior search efficiency has been published [54].

Contribution 8.4 *A demonstration of the practical implementation potential of the GFLo framework in the form of an evaluation of observation camera placement within the MeerKAT radio telescope project.*

In order to demonstrate the practical usability of the GFLo framework, an evaluation of observation camera placement within the MeerKAT radio telescope project was pursued and resulted in a research publication [53]. This study involved collaboration with system engineers who were originally tasked with placing the existing cameras (whose locations are now fixed). The process that they followed in choosing the actual sites proved to be very laborious and time-consuming, compared to the approach followed in solving the problem using the GFLo framework proposed in this dissertation. The NSGA-II and the MRA were used in conjunction to solve the problem of siting three observation cameras (two short range and one long range camera) with the two objectives of monitoring 64 MeerKAT radio telescope antennas and a large terrain surface area. The suitability of the actual camera sites compared very favourably with the solutions obtained *via* the framework of this dissertation, although the framework’s solutions outperformed the actual configuration (very slightly). Whereas the actual camera sites were selected without consideration of possible interference of antennas in respect of visibility computations, however, an advantage of using the framework was the ability to evaluate placement solutions before the construction of any physical structures that may affect facility performance with respect to the visibility objectives. This advantage follows from including the estimated heights of the antennas in the visibility analyses. Feedback on the results of this study by the decision maker tasked with finalising the camera sites was positive, and has led to the possibility of further collaboration in respect of their future facility location sub-projects.

Contribution 8.5 *The establishment of a diverse range of realistic worked GFLoP instances to demonstrate the workability of the GFLo framework and to compare the algorithms designed in this dissertation.*

The GFLo framework proposed in this dissertation was designed in such a manner that GFLoP instances may be solved generically by including combinations of objectives and constraints that were not previously possible or considered. The accomplishment of this aim was demonstrated by means of a collection of realistic worked GFLoP instances throughout the dissertation chapters. These problem instances included different combinations of covering, dispersion, centre, MaxiMin and MiniMax objectives and constraints. The GFLoP instances also served the purpose of shedding light on the performance of the algorithms compared to each other and it was concluded that all three algorithms merit inclusion in the GFLo framework. Additional examples of GFLoPs solved by the GFLo framework are available in supplementary documentation [55].

8.3 Ideas for future work

Due to the large scope of the GFLo domain, there was not sufficient time available to investigate all of the detailed aspects that were identified as potential research directions. The framework presented in this study, while achieving its intended purpose, is by no means complete. This section contains a number of suggestions with respect to possible future research.

Proposal 8.1 *Additional practical evaluations performed in a similar vein as the MeerKAT study of §7.*

Solving alternative facility location solutions for real-world problems, as determined by the GFLo framework, should be performed in order to illustrate that problems solved by alternative approaches in practice may have been solved within significantly shorter computation times and with relative ease using the framework proposed in this dissertation, compared to the procedures that were actually followed.

Proposal 8.2 *Additional evaluation of the algorithms developed in this dissertation.*

Although various GFLoPs were solved by the three algorithms in this dissertation, the terrain areas over which they were solved were limited to only two. The performance of the algorithms should be evaluated over a variety of terrain types in order to better ascertain their effectiveness in solving problems involving different terrain types and numbers of facilities to be placed. Random landscape generators may be used for these purposes, in order to provide a variety of terrain types considered for facility placement. Comparing the algorithms over these diverse landscapes will provide better insight into the overall performance of the algorithms compared to each other — and may also provide insight into which algorithms may be better suited to specific terrain types/facility placement requirements.

Proposal 8.3 *The development of a fully functional generic GFLo DSS.*

The optimisation concept demonstrator GUI proposed in §7.1.2 is a useful and powerful tool that has been used to solve complex FLoPs in this dissertation. The GUI is, however, limited in modularity in terms of the number of facility types, SZs and IZs that may be included in the model — this is not a practically feasible implementation for the purposes of the research that is advocated in this dissertation. The modularity of the GUI is therefore a potential future development and, in order to complete the user interface, pre-optimisation, optimisation and analysis components should be combined into a comprehensive GFLo user interface.

Proposal 8.4 *Inclusion of other algorithms in the framework.*

While the NSGA-II, the HGA and AMOEBa were all considered to be suitable algorithms for a diverse range of GFLoP instances, there remain strong alternatives, such as simulated annealing

[67] and strength-pareto evolutionary algorithm 2 [143], that may yield favourable results in some instances.

Proposal 8.5 *More detailed design and testing of the novel MRA.*

The MRA proposed in §6.3 returned excellent results and is a major research output of this dissertation. It remains, however, a very new approach that requires intensive testing and development in order to gain a better sense of understanding of its practical performance when implemented in the context of different problems. In order to achieve this, the MRA will have to be tested in respect of terrains of various sizes and roughness, and its performance is also expected to vary according to different placement criteria types. The range of neighbourhood exploitation when moving into higher resolution levels and the choice of resolution levels to consider in the MRA is also an important problem-specific consideration that is expected to depend primarily on terrain roughness, features and size, and requires further investigation.

Proposal 8.6 *Further development of AMOEBA and the HGA.*

AMOEBA returned very good results within fast computation times, compared to those achieved by the popular and highly rated NSGA-II. AMOEBA is, however, a recent research output and requires further testing in order to gauge its effectiveness according to different types of GFLoPs. It was observed that AMOEBA is an excellent alternative for GFLoP instances in which data are accessed from memory only, yet its results for problem instances in which data are accessed from disk storage were average and may therefore benefit from further analyses. The HGA consistently performed the best of the algorithms in terms of solution quality, within good computation times. The superiority of its results stem partly from the success of AMOEBA, and any improvement in AMOEBA is therefore expected to yield improvements in the performance of the HGA.

Proposal 8.7 *A search efficiency metric.*

In §6.3.1 it was noted that only a very small percentage of all evaluated sites were included in the final non-dominated solutions of a particular study. This, however, is a general occurrence for all the GFLoPs solved in this dissertation. Techniques such as the MRA may reduce the number of sites that are evaluated in order to arrive at a set of non-dominated solutions of similar (or better) quality — an indication of the MRA’s superior search efficiency. A *search efficiency metric* that measures a solution approach’s effectiveness should be investigated. Such a search efficiency metric should, at least, consider the total number of candidate sites evaluated and the quality of the non-dominated front returned by the approach (the hypervolume measure, for example). Total computation time is another factor that may be included in such a metric. The metric may then be used to compare the efficiency of different algorithms and search approaches for various GFLoPs involving different terrain types, terrain size, facility types and facility numbers, which may ultimately provide problem solvers with information that may aid in the selection of algorithms and search strategies to employ when solving new GFLoPs.

Proposal 8.8 *Further development of a novel viewshed estimation technique.*

The author proposed a fast viewshed estimation technique in a research publication [52]. This technique was, however, not implemented in the GFLo framework due to the requirement that viewsheds had to be determined with respect to *all* the gridposts in the IZs, as opposed to the viewshed estimation technique’s method of evaluating visibility with respect to only *some* gridposts. By implementing techniques such as distance-based clustering analysis, the method may yet be developed to be compatible with the GFLo framework. The viewshed estimation technique nevertheless warrants further research for use in studies that merely require visibility percentages with respect to specific areas.

References

- [1] AGARWAL PK, BEREG S, DAESCU O, KAPLAN H, NTAFO S, SHRIR M & ZHU B, 2005. *Guarding a terrain by two watchtowers*, Proceedings of the Twenty-First Annual Symposium on Computational Geometry, pp. 346–355.
- [2] AKELLA MR, DELMELLA R, BATA R, ROGERSON P & BLATT A, 2010. *Adaptive cell tower location using geostatistics*, Geographical Analysis, **42**, pp. 227–244.
- [3] ALP O, ERKUT E & DREZNER Z, 2003. *An efficient genetic algorithm for the p-median problem*, Annals of Operations Research, **122**, pp. 21–42.
- [4] ARABANI AB & FARAHANI RZ, 2012. *Facility location dynamics: An overview of classifications and applications*, Computers and Industrial Engineering, **62**, pp. 408–420.
- [5] ARCGIS, 2013. *ArcGIS*, [Online], [Cited January 6th, 2014], Available from <http://www.arcgis.com>.
- [6] ANONYMOUS, 1994. *Intelligence preparation of the battlefield*, [Online], [Cited January 6th, 2015], Available from www.fas.org/irp/doddir/army/fm34-130.pdf.
- [7] BAO S, XIAO N, LAI Z, ZHANG H & KIM C, 2015. *Optimizing watchtower locations for forest fire monitoring using location models*, Fire Safety Journal, **71**, pp. 100–109.
- [8] BEHESHTI Z & SHAMSUDDIN S, 2013. *A review of population-based meta-heuristic algorithms*, International Journal of Advances in Soft Computing and its Applications, **5**, pp. 1–35.
- [9] BEN-MOSHE B, 2005. *Geometric facility location optimization*, PhD Dissertation, Ben-Gurion University, Beer-Sheva.
- [10] BRESENHAM JE, 1965. *Algorithm for computer control of a digital plotter*, IBM Systems Journal, **4**, pp. 25–30.
- [11] CHAUDHARI PM, DHARASKAR PV & THAKARE VM, 2010. *Computing the most significant solution from Pareto-front obtained in multi-objective evolutionary*, International Journal of Advanced Computer Science and Applications, **1**, pp. 63–68.
- [12] CHURCH R & REVELLE C, 1974. *The maximal covering location problem*, Papers of the Regional Science Association, **32**, pp. 101–118.
- [13] COELLO COELLO CA, 2002. *Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art*, Computer Methods in Applied Mechanics and Engineering, **191**, pp. 1245–1287.

- [14] COELLO COELLO CA & MEZURA MONTES E, 2002. *Constraint-handling in genetic algorithms through the use of dominance-based tournament selection*, Advanced Engineering Informatics, **16**, pp. 193–203.
- [15] CORNE DW, KNOWLES JD & OATES MJ, 200. *The Pareto envelope-based selection algorithm for multiobjective optimization*, Proceedings of the Parallel Problem Solving from Nature VI Conference, **16**, pp. 839–848.
- [16] CURTIN KM & CHURCH RL, 2006. *A family of location models for multiple-type discrete dispersion*, Geographical Analysis, **38**, pp. 248–270.
- [17] DAS I & DENNIS JE, 1997. *A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems*, Structural and Multidisciplinary Optimization, **14**, pp. 63–69.
- [18] DASKIN MS, 1983. *A maximum expected covering location model: Formulation, properties and heuristic solutions*, Transportation Science, **17**, pp. 48–69.
- [19] DASKIN MS & STERN EH, 1981. *A hierarchical objective set covering model for emergency medical service vehicle deployment*, Transportation Science, **15**, pp. 137–152.
- [20] DEB K, 2000. *An efficient constraint handling method for genetic algorithms*, Computer Methods in Applied Mechanics and Engineering, **186**, pp. 311–338.
- [21] DEB K, PRATAP A, AGARWAL S & MEYARIVAN T, 2002. *A fast and elitist multi-objective genetic algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation, **6**, pp. 182–197.
- [22] DEB K, 2004. *A population-based algorithm-generator for real-parameter optimization*, Soft Computing, **9**, pp. 236–253.
- [23] DE FLORIANI L & MAGILLO P, 1999. *Intervisibility on terrains*, Chapter 38. In MAGUIRE D & RHIND D, *Geographic Information Systems: Principles, Techniques, Management and Applications*, Wiley & Sons, New York, pp. 543–556.
- [24] DE FLORIANI L & MAGILLO P, 2003. *Algorithms for visibility computation on terrains: A survey*, Environment and Planning B: Planning and Design, **30**, pp. 709–728.
- [25] DE JONG KA & SPEARS WM, 1992. *A formal analysis of the role of multi-point crossover in genetic algorithms*, Annals of Mathematics and Artificial Intelligence Journal, **5**, pp. 1–6.
- [26] DEMERS MN, 2009. *Fundamentals of geographic information systems*, Fourth Edition, Wiley & Sons, Hoboken (NJ).
- [27] DOMENIKIOTIS C, DALEZIOS NR & FARASLIS I, 2010. *GIS-based weather radar siting procedure in mountainous terrain*, Physics and Chemistry of the Earth, **35**, pp. 35–42.
- [28] DREZNER Z, 2012. *Facility location: A survey of applications and methods*, Springer, New York (NY).
- [29] DUARTE A, PANTRIGO J, PARDO E & MLADENOVIC N, 2015. *Multi-objective variable neighborhood search: An application to combinatorial optimization problems*, Journal of Global Optimization, **30**, pp. 515–536.

- [30] DUVENHAGE B, 2009. *Using an implicit min/max KD-tree for doing efficient terrain line of sight*, (Unpublished) Technical Report, The Council for Scientific and Industrial Research, Pretoria.
- [31] EMAMI A & NOGHREH P, 2010. *New approach on optimization in placement of wind turbines within wind farm by genetic algorithms*, *Renewable Energy*, **35**, pp. 1159–1564.
- [32] ENERGY.GOV, 2015. *Hybrid wind and solar electric systems*, [Online], [Cited December 1st, 2015], Available from www.energy.gov/energysaver/hybrid-wind-and-solar-electric-systems.
- [33] ERKUT E & NEUMAN S, 1991. *Comparison of four models for dispersing facilities*, *Information Systems and Operational Research*, **29**, pp. 68–86.
- [34] FARAHANI RZ, STEADIESEIFI M & ASGARI N, 2010. *Multiple criteria facility location problems: A survey*, *Applied Mathematical Modelling*, **49**, pp. 1689–1709.
- [35] FARAHANI RZ, ASGARI N, HEIDARI N, 2012. *Covering problems in facility location: A review*, *Computers and Industrial Engineering*, **62**, pp. 368–407.
- [36] FONSECA CM & FLEMING PJ, 1993. *Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization*, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416–423.
- [37] FRANKLIN WR, 2002. *Siting observers on terrain*, *Proceedings of the 10th International Symposium on Advances in Spatial Data Handling*, pp. 109–120.
- [38] FREE GIS DATA, 2015. *Free GIS Data*, [Online], [Cited December 1st, 2015], Available from <http://freegisdata.rtwilson.com>.
- [39] FU P & RICH P, 2002. *A geometric solar radiation model with applications in agriculture and forestry*, *Computers and Electronics in Agriculture*, **37**, pp. 25–35.
- [40] GENCER C, KIZILKAYA AYDOGAN E & CELIK C, 2008. *A decision support system for locating VHF/UHF radio jammer systems on the terrain*, *Information System Frontiers*, **10**, pp. 111–124.
- [41] GEOCOMMONS, 2015. *Geocommons*, [Online], [Cited December 1st, 2015], Available from <http://geocommons.com>.
- [42] GEOCOMMUNITY, 2015. *Geocommunity — The premier portal for geospatial technology professionals*, [Online], [Cited December 1st, 2015], Available from <http://www.geocomm.com>.
- [43] GHOSE D, PRASAD UR & GURUPRASAD K, 1993. *Missile battery placement for air defense: A dynamic programming approach*, *Applied Mathematical Modelling*, **17**, pp. 450–458.
- [44] GLOVER F, 1989. *Tabu search — Part I*, *ORSA Journal of Computing*, **1**, pp. 190–206.
- [45] GOGNA A & TAYAL A, 2013. *Metaheuristics: Review and application*, *Journal of Experimental & Theoretical Artificial Intelligence*, **25**, pp. 503–526.
- [46] GOOGLE, 2015. *Google earth*, [Online], [Cited December 1st, 2015], Available from <http://www.google.com/earth>.

- [47] GRADY SA, HUSSAINI MY & ABDULLAH MM, 2005. *Placement of wind turbines using genetic algorithms*, *Renewable Energy*, **30**, pp. 259–270.
- [48] GRASS DEVELOPMENT TEAM, 2015. *Geographic resources analysis support system (GRASS GIS) software*, [Online], [Cited December 1st, 2015], Available from <http://grass.osgeo.org>.
- [49] GVSIG, 2013. *gvSIG Portal — gvSIG*, [Online], [Cited January 6th, 2014], Available from <http://www.gvsig.org>.
- [50] HAKIMI SL, 1965. *Optimum distribution of switching centers in a communication network and some related graph theoretic problems*, *Operations Research*, **13**, pp. 462–475.
- [51] HEYNS AM, 2015. *MO GIS-based facility location GUI demonstration video*, [Video file], Retrievable from www.vuuren.co.za/photos/L0CVideo.ogv.
- [52] HEYNS AM & VAN VUUREN JH, 2013. *Terrain visibility-dependent facility location through fast dynamic step-distance viewshed estimation within a raster environment*, *Proceedings of the 42nd Annual Conference of the Operations Research Society of South Africa*, pp. 112–121.
- [53] HEYNS AM & VAN VUUREN JH, 2015. *An evaluation of the effectiveness of observation camera placement within the MeerKAT radio telescope project*, *South African Journal of Industrial Engineering*, **26**, pp. 10–25.
- [54] HEYNS AM & VAN VUUREN JH, 2016. *A multi-resolution approach towards point-based multi-objective geospatial facility location*, *Computers, Environment and Urban Systems*, **57**, pp. 80–92.
- [55] HEYNS AM & VAN VUUREN JH, 2016. *Supporting documentation for the multi-resolution approach*, [Online]. [Cited January 6th, 2016], Available from <http://www.vuuren.co.za/papers/mradoc.pdf>.
- [56] HOGAN K & REVELLE C, 1986. *Concepts and applications of backup coverage*, *Management Science*, **32**, pp. 1434–1444.
- [57] HUGHES EJ, 2005. *Evolutionary many-objective optimisation: Many once or one many?*, *The 2005 Congress on Evolutionary Computation*, pp. 222–227.
- [58] ISHIBUCHI H, NOJIMA Y & DOI T, 2006. *Comparison between single-objective and multi-objective genetic algorithms: Performance comparison and performance measures*, *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pp. 1143–1150.
- [59] ISHIBUCHI H, MURATA T & TURKSEN IB, 1997. *Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems*, *Fuzzy Sets and Systems*, **89**, pp. 135–150.
- [60] JANKE J, 2010. *Multicriteria GIS modeling of wind and solar farms in Colorado*, *Renewable Energy*, **35**, pp. 2228–2234.
- [61] JANSEN T, 2013. *Analyzing evolutionary algorithms*, First Edition, Springer, New York (NY).
- [62] JASZKIEWICZ A, 2003. *On the computational efficiency of multiple objective metaheuristics: The knapsack problem case study*, *European Journal of Operational Research*, **158**, pp. 418–433.

- [63] JENSEN MT, 2003. *Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms*, IEEE Transactions on Evolutionary Computation, **7**, pp. 503–515.
- [64] KIANI SADRA M, NASSIRI P, HOSSEINI M, MONAVARI M & GHARAGOZLOU A, 2014. *Assessment of land use compatibility and noise pollution at Imam Khomeini International Airport*, Journal of Air Transport Management, **34**, pp. 49–56.
- [65] KIM K, MURRAY AT & XIAO N, 2008. *A multiobjective evolutionary algorithm for surveillance sensor placement*, Environment and Planning B: Planning and Design, **35**, pp. 935–948.
- [66] KIM Y, RANA S & WISE S, 2004. *Exploring multiple viewshed analysis using terrain features and optimisation techniques*, Computers and Geosciences, **30**, pp. 1019–1032.
- [67] KIRKPATRICK S, GELATT C & VECCHI M, 1983. *Optimization by simulated annealing*, Science, **220**, pp. 671–680.
- [68] KLOSE A & DREXL A, 2005. *Facility location models for distribution system design*, European Journal of Operational Research, **162**, pp. 4–29.
- [69] KNOWLES JD, WATSON RA & CORNE DW, 2001. *Reducing local optima in single-objective problems by multi-objectivization*, Proceedings of the First International Conference on Evolutionary Multi-criterion Optimization, pp. 269–283.
- [70] KNOWLES JD, THIELE L & ZITZLER E, 2006. *A tutorial on the performance assessment of stochastic multiobjective optimizers*, TIK Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich.
- [71] KO J, CHANG S & LEE B, 2011. *Noise impact assessment by utilizing noise map and GIS: A case study in the city of Chungju, Republic of Korea*, Applied Acoustics, **72**, pp. 544–550.
- [72] KRAMER O, 2010. *A review of constraint-handling techniques for evolution strategies*, Applied Computational Intelligence and Soft Computing, **2010**, pp. 1–11.
- [73] KRZANOWSKI RM & RAPER J, 1999. *Hybrid genetic algorithm for transmitter location in wireless networks*, Computers, Environment and Urban Systems, **23**, pp. 359–382.
- [74] KUBY MJ, 1987. *Programming models for facility dispersion: The p-dispersion and maximum dispersion problems*, Geographical Analysis, **19**, pp. 315–329.
- [75] KWONG WY, ZHANG PY, ROMERO D, MORAN J, MORGENROTH M & AMON C, 2014. *Multi-objective wind farm layout optimization considering energy generation and noise propagation with NSGA-II*, Journal of Mechanical Design, **136**, pp. 1–10.
- [76] LEI TL & CHURCH RL, 2013. *A unified model for dispersing facilities*, Geographical Analysis, **45**, pp. 401–418.
- [77] LEI TL & CHURCH RL, 2014. *Vector assignment ordered median problem: A unified median problem*, International Regional Science Review, **37**, pp. 194–224.
- [78] LEI TL & CHURCH RL, 2015. *On the unified dispersion problem: Efficient formulations and exact algorithms*, European Journal of Operational Research, **241**, pp. 622–630.

- [79] MALOY MA & DEAN DJ, 2001. *An accuracy assessment of various GIS-based viewshed delineation techniques*, Photogrammetric Engineering and Remote Sensing, **67**, pp. 1293–1298.
- [80] MARIANOV V & EISELT HA, 2012. *Transmitter location for maximum coverage and constructive-destructive interference management*, Computers and Operations Research, **39**, pp. 1441–1449.
- [81] MARÍN AS, NICKEL JP, PUERTO J & VELTEN S, 2009. *A flexible model and efficient solution strategies for discrete location problems*, Discrete Applied Mathematics, **157**, pp. 1128–1145.
- [82] MARTÍ L, GARCÍA J, BERLANGA A & MOLINA JM, 2009. *An approach to stopping criteria for multi-objective optimization evolutionary algorithms: The MGBM Criterion*, Proceedings of the 2009 IEEE Congress on Evolutionary Computation, pp. 1263–2009.
- [83] MATHAR R & NIESSEN T, 2000. *Optimum positioning of base stations for cellular radio networks*, Wireless Networks, **6**, pp. 421–428.
- [84] MATHWORKS, 2014. *MATLAB — The language of technical computing*, [Online], [Cited May 8th, 2014], Available from <http://www.mathworks.com/products/matlab/>.
- [85] MAVROTAS G, 2009. *Effective implementation of the ϵ -constraint method in multi-objective mathematical programming problems*, Applied Mathematics and Computation, **213**, pp. 455–465.
- [86] MEUNIER H, TALBI E & REININGER P, 2000. *A multiobjective Genetic algorithm for radio network optimization*, Proceedings of the 2000 Congress on Evolutionary Computation, pp. 317–324.
- [87] MEZURA MONTES E & COELLO COELLO CA, 2011. *Constraint-handling in nature-inspired numerical optimization: Past, present and future*, Swarm and Evolutionary Computation, **1**, pp. 173–194.
- [88] MINCIARDI R, SACILE R & SICCARDI F, 2003. *Optimal planning of a weather radar network*, Journal of Atmospheric and Oceanic Technology, **20**, pp. 1251–1263.
- [89] MLADENović N & HANSEN P, 1997. *Variable neighborhood search*, Computers and Operations Research, **24**, pp. 1097–1100.
- [90] MOLNOROVA K, SKLENICKA P, STIBOREK J, SVOBODOVA K & SALEK M, 2007. *Visual preferences for wind turbines: Location, numbers and respondent characteristics*, Applied Energy, **92**, pp. 269–278.
- [91] MOON ID & CHAUDHRY SS, 1984. *An analysis of network location problems with distance constraints*, Management Science, **30**, pp. 290–307.
- [92] MOSETTI G, POLONI C & DIVIACCO B, 1994. *Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm*, Journal of Wind Engineering and Industrial Aerodynamics, **51**, pp. 105–116.
- [93] MURRAY AT, KIM K, DAVIS JW, MACHIRAJU R & PARENT R, 2007. *Coverage optimization to support security monitoring*, Computers, Environment and Urban Systems, **31**, pp. 133–147.

- [94] MURRAY AT, 2003. *Site placement uncertainty in location analysis*, Computers, Environment and Urban Systems, **27**, pp. 205–221.
- [95] NAGY G, 1994. *Terrain visibility*, Computers and Graphics, **18**, pp. 763–773.
- [96] NICKEL JP & PUERTO J, 1999. *A unified approach to network location problems*, Networks, **34**, pp. 283–290.
- [97] NICKEL JP & PUERTO J, 2005. *Location theory: A unified approach*, Springer Verlag, Berlin.
- [98] OWEN SH & DASKIN MS, 1998. *Strategic facility location: A review*, European Journal of Operational Research, **111**, pp. 423–447.
- [99] PISINGER D, 2006. *Upper bounds and exact algorithms for p-dispersion problems*, Computers and Operations Research, **33**, pp. 1380–1398.
- [100] POND GT, BRIMBERG J, WANG Y & SIMMS B, 2015. *A comparison of heuristics applied to the sensor deployment problem in two dimensions*, Journal of Defense Modeling and Simulation: Applications, Methodology, Technology **12**, pp. 343–352.
- [101] PURSHOUSE RC & FLEMING PJ, 2003. *Evolutionary many-objective optimisation: An exploratory analysis*, The 2003 Congress on Evolutionary Computation, pp. 8–12.
- [102] PUZICHA J, HOFMANN T & BUHMANN JM, 1999. *A theory of proximity based clustering: Structure detection by optimization*, Pattern Recognition, **33**, pp. 617–634.
- [103] QGIS, 2015. *Welcome to the QGIS project!*, [Online], [Cited December 1st, 2015], Available from <http://qgis.org>.
- [104] RANA S, 2003. *Fast approximation of visibility dominance using topographic features as targets and the associated uncertainty*, Photogrammetric Engineering and Remote Sensing, **59**, pp. 1149–1160.
- [105] RAISANEN L & WHITAKER RW, 2005. *Comparison and evaluation of multiple objective genetic algorithms for the antenna placement problem*, Mobile Networks and Applications, **10**, pp. 79–88.
- [106] RAJABALIPOUR CHESHMEHGAZ H, HARON H & SHARIFI A, 2015. *The review of multiple evolutionary searches and multi-objective evolutionary algorithms*, Artificial Intelligence Review, **43**, pp. 311–343.
- [107] RAVI SS, ROSENKRANTZ DJ & TAYI GK, 1994. *Heuristics and special case algorithms for dispersion problems*, Operations Research, **42**, pp. 299–310.
- [108] REED SE, BOGGS JL & MANN JP, 2012. *A GIS tool for modeling anthropogenic noise propagation in natural ecosystems*, Environmental Modelling and Software, **37**, pp. 1–5.
- [109] RESENDE MGC & WERNECK RF, 2007. *A fast swap-based local search procedure for location problems*, Annals of Operations Research, **150**, pp. 205–230.
- [110] REVELLE CS & EISELT HA, 2005. *Location analysis: A synthesis and survey*, European Journal of Operational Research, **165**, pp. 1–19.
- [111] ROSING KE, HILLSMAN EL & ROSING-VOGEHR H, 1929. *A note comparing optimal and heuristic solutions to the p-median problem*, Geographical Analysis, **11**, pp. 86–89.

- [112] SANCHEZ-LOZANO JM, TERUEL-SOLANO J, SOTO-ELVIRA PL & SOCORRO GARCIA-CASCALES M, 2013. *Geographical Information Systems (GIS) and Multi-Criteria Decision Making (MCDM) methods for the evaluation of solar farms locations: Case study in south-eastern Spain*, Renewable and Sustainable Energy Reviews, **24**, pp. 544–556.
- [113] SCHEIBE KP, CARSTENSEN LW, RAKES R & REES LP, 2006. *Going the last mile: A spatial decision support system for wireless broadband communications*, Decision Support Systems, **42**, pp. 557–570.
- [114] SERRANO-GONZÁLEZ J, BURGOS-PAYÀN M & GONZÁLEZ-LONGATT F, 2013. *A review and recent developments in the optimal wind-turbine micro-siting problem*, Renewable and Sustainable Energy Reviews, **30**, pp. 133–144.
- [115] SHI C, CHEN M & Z SHI Z, 2005. *A fast nondominated sorting algorithm*, 2005 International Conference on Neural Networks and Brain, **3**, pp. 1605–160.
- [116] SHIER DR, 1977. *A min-max theorem for p-center problems on a tree*, Transportation Science, **11**, pp. 1696–1707.
- [117] SIMS BR, 2010. *A viewshed accuracy assessment: Comparisons of field-derived and computer-derived viewsheds*, MA Thesis, California State University, Long Beach (CA).
- [118] SLIZ-SZKLINIARZ B & VOGT J, 2011. *GIS-based approach for the evaluation of wind energy potential: A case study for the Kujawsko-Pomorskie Voivodeship*, Renewable and Sustainable Energy Reviews, **15**, pp. 243–252.
- [119] SMITH K, EVERSON R, FIELDSEND J, MURPHY C & MISRA R, 2008. *Dominance-based multiobjective simulated annealing*, IEEE Transactions on Evolutionary Computation, **12**, pp. 323–342.
- [120] SMITH MJ, GOODCHILD MF & LONGLEY PA, 2009. *Geospatial analysis: A comprehensive guide to principles, techniques and software tools*, Third Edition, Matador, Leicester.
- [121] SQUARE KILOMETRE ARRAY AFRICA, 2015. *Digital elevation data*, [Online], [Cited December 5th, 2015], Available from <http://www.ska.ac.za>.
- [122] STANIMIROVIĆ IP, ZLATANOVIĆ ML & PETKOVIĆ MD, 2011. *On the linear weighted sum method for multi-objective optimization*, Facta Universitatis (Niš Series Mathematics and Informatics), **26**, pp. 49–63.
- [123] STEWART TJ, 2007. *The essential multiobjectivity of linear programming*, ORION, **23**, pp. 1–15.
- [124] TABIK S, ZAPATA EL & ROMERO LF, 2013. *Simultaneous computation of total viewshed on large high resolution grids*, International Journal of Geographical Information Science, **27**, pp. 804–814.
- [125] TANERGÜCLÜ T, MARAŞ H, GENCER S & AYGÜNEŞ H, 2010. *A decision support system for locating weapon and radar positions in stationary point air defence*, Information Systems Frontiers, **14**, pp. 423–444.
- [126] TEITZ MB & BART P, 1968. *Heuristic methods for estimating the generalized vertex median of a weighted graph*, Operations Research, **16**, 955–961.

- [127] THIELE L, LAUMANN S, FONSECA CM & DA FONSECA VG, 2003. *Performance assessment of multiobjective optimizers: An analysis and review*, IEEE Transactions on Evolutionary Computation, **7**, pp. 117–131.
- [128] TONG D, MURRAY A & XIAO N, 1968. *Heuristics in spatial analysis: A genetic algorithm for coverage maximization*, Annals of the Association of American Geographers, **99**, pp. 698–711.
- [129] TOREGAS C, SWAIN R, REVELLE C & BERGMAN L, 1971. *The location of emergency service facilities*, Operations Research, **19**, pp. 1363–1373.
- [130] TRAN R, WU J, DENISON C, ACKLING T, WAGNER M & NEUMANN F, 2013. *Fast and effective multi-objective optimisation of wind turbine placement*, Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, pp. 1381–1388.
- [131] USGS, 2015. *US geological survey*, [Online], [Cited December 1st, 2015], Available from <http://www.usgs.gov>.
- [132] VIEWFINDERPANORAMAS, 2015. *Digital elevation data*, [Online], [Cited December 2nd, 2015], Available from <http://www.viewfinderpanoramas.org/dem3.html>.
- [133] UYAN M, 2013. *GIS-based solar farms site selection using Analytic Hierarchy Process (AHP) in Karapinar region, Konya/Turkey*, Renewable and Sustainable Energy Reviews, **28**, pp. 11–17.
- [134] VAN HAAREN R & FTHENAKIS V, 2011. *GIS-based wind farm site selection using Spatial Multi-Criteria Analysis (SMCA): Evaluating the case for New York State*, Renewable and Sustainable Energy Reviews, **15**, pp. 3332–3340.
- [135] VAN DEN HENGEL A, HILL R, WARD B, CICHOWSKI A, DETMOLD H, MADDEN C, DICK A & BASTIAN J, 2009. *Automatic camera placement for large scale surveillance networks*, 2009 Workshop on Applications of Computer Vision, pp. 1–6.
- [136] JALALI VARNAMKHAHI M, 2012. *Overview of the algorithms for solving the p-median facility location problems*, Advanced Studies in Biology, **4**, pp. 49–55.
- [137] VALENZUELA CL, 2002. *A simple evolutionary algorithm for multi-objective optimization (SEAMO)*, Proceedings of the 2002 Congress of Evolutionary Computation, pp. 717–722.
- [138] WATANABE S & SAKAKIBARA K, 2005. *Multi-objective approaches in a single-objective optimization environment*, The 2005 IEEE Congress on Evolutionary Computation, pp. 1714–1721.
- [139] WHITE J & CASE K, 1974. *On covering problems and the central facilities location problem*, Geographical Analysis, **6**, pp. 281–293.
- [140] YAMANI DOUZI SORKHABI S, ROMERO DA, KAI YAN G, DAO GU M, MORAN J, MORGENROTH M & AMON CH, 2016. *The impact of land use constraints in multi-objective energy-noise wind farm layout optimization*, Renewable Energy, **85**, pp. 359–370.
- [141] ZITZLER E & THIELE L, 1999. *Multiobjective evolutionary algorithms: A comparative case study and the Strength Pareto Evolutionary Algorithm*, IEEE Transactions on Evolutionary Computation, **3**, pp. 257–271.
- [142] ZITZLER E, DEB K & THIELE L, 2000. *Comparison of multiobjective evolutionary algorithms: Empirical results*, Evolutionary Computation, **8**, pp. 173–195.

- [143] ZITZLER E, LAUMANN M & THIELE L, 2001. *SPEA2: Improving the strength Pareto evolutionary algorithm*, (Unpublished) Technical Report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich.

APPENDIX A

Mathematical formulations for the GFLoP instances of Chapter 7

This appendix contains the mathematical formulations of the GFLoP instances of Chapter 7.

A.1 Small-area problem instances

This section is devoted to the formulation of the objectives and constraints for the small-area GFLoP instances of §7.3.

A.1.1 Low-complexity bi-objective GFLoP instance of §7.3.1

The facility and criteria sets are

$$\mathbb{F} = \{1[\text{Solar-powered facility}]\}$$

and

$$\mathbb{C} = \{1[\text{Average daily solar radiation}], 2[\text{Inter-facility distance}]\},$$

respectively.

The gridposts in SZs \mathbb{S}_1 and \mathbb{S}_2 are all those enveloped by the zones indicated in Figure 5.10, while the facility-SZ assignment matrices are given by

$$\mathbf{F}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \quad \mathbf{F}_2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \quad \text{and} \quad \mathbf{F}_3 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}.$$

The set-pair p_1 for the first objective corresponds to the facility and SZ sets $\mathcal{F}_1 = \{1, 1\}$ and $\mathcal{S}_1 = \{1, 2\}$. The set-pairs p_2 and p_2^* for the second objective correspond to the facility and SZ sets $\mathcal{F}_2 = \{1\}$ and $\mathcal{S}_2 = \{1\}$ and $\mathcal{F}_{2^*} = \{1\}$ and $\mathcal{S}_{2^*} = \{2\}$, respectively.

The first objective, according to (4.2), is to

$$\text{maximise } V_1 \tag{A.1}$$

subject to (4.3a); that is

$$V_1 \leq v_{c_1}(\ell) + M(1 - \tilde{x}_{p_1}(\ell)) \tag{A.2}$$

for all ordered pairs $f_{p_1} \in \mathcal{F}_{p_1}$ and $s_{p_1} \in \mathcal{S}_{p_1}$, and for all $\ell \in \mathbb{S}_{s_{p_1}}$.

The second objective, according to (4.15), is to

$$\text{maximise } V_2 \quad (\text{A.3})$$

subject to (4.16a); that is

$$V_2 \leq v_{c_2}(\ell, \ell^*) + M(2 - \tilde{x}_{p_2}(\ell) - \tilde{x}_{p_2^*}(\ell^*)) \quad (\text{A.4})$$

for all ordered pairs of facilities and SZs $f_{p_2} \in \mathcal{F}_{p_2}$ and $s_{p_2} \in \mathcal{S}_{p_2}$, all $f_{p_2^*} \in \mathcal{F}_{p_2^*}$ and $s_{p_2^*} \in \mathcal{S}_{p_2^*}$, and for all $\ell \in \mathbb{S}_{s_{p_2}}$ and $\ell^* \in \mathbb{S}_{s_{p_2^*}}$.

The set-pair for the constraint is p_1 , and the constraint is enforced between all the facilities in the same set-pair, therefore $p_1^* = p_1$. The lower bound constraint is $c_{\min}^2 = 500$ metres. From (4.21), the constraint is therefore

$$500 \leq v_{c_2}(\ell, \ell^*) + M(2 - \tilde{x}_{p_1}(\ell) - \tilde{x}_{p_1^*}(\ell^*)) \quad (\text{A.5})$$

for all ordered pairs of facilities and SZs $f_{p_1} \in \mathcal{F}_{p_1}$ and $s_{p_1} \in \mathcal{S}_{p_1}$, all $f_{p_1^*} \in \mathcal{F}_{p_1^*}$ and $s_{p_1^*} \in \mathcal{S}_{p_1^*}$, and for all $\ell \in \mathbb{S}_{s_{p_1}}$ and $\ell^* \in \mathbb{S}_{s_{p_1^*}}$. The formulation is completed with the inclusion of the mandatory constraints (3.4)–(3.6).

A.1.2 Low-complexity tri-objective GFLoP instance of §7.3.2

The same facility types, SZs and facility-SZ assignment matrices of §A.1.1 are considered in this GFLoP instance. The criteria set is

$$\mathbb{C} = \{1[\text{Average daily solar radiation}], 2[\text{Distance from centre-point}]\}.$$

The set-pairs p_1 and p_2 for the first two objectives are the same as in §A.1.1, while set-pair p_3 is the same as set pair p_2^* in §A.1.1.

The first objective, according to (4.2), is to

$$\text{maximise } V_1 \quad (\text{A.6})$$

subject to (4.3a); that is

$$V_1 \leq v_{c_1}(\ell) + M(1 - \tilde{x}_{p_1}(\ell)) \quad (\text{A.7})$$

for all ordered pairs $f_{p_1} \in \mathcal{F}_{p_1}$ and $s_{p_1} \in \mathcal{S}_{p_1}$, and for all $\ell \in \mathbb{S}_{s_{p_1}}$.

The second objective, according to (4.2), is to

$$\text{maximise } V_2 \quad (\text{A.8})$$

subject to (4.3a); that is

$$V_2 \leq v_{c_2}(\ell) + M(1 - \tilde{x}_{p_2}(\ell)) \quad (\text{A.9})$$

for all ordered pairs $f_{p_2} \in \mathcal{F}_{p_2}$ and $s_{p_2} \in \mathcal{S}_{p_2}$, and for all $\ell \in \mathbb{S}_{s_{p_2}}$.

The third objective, according to (4.4), is to

$$\text{minimise } V_3 \quad (\text{A.10})$$

subject to (4.5a); that is

$$V_3 \geq v_{c_2}(\ell) \tilde{x}_{p_3}(\ell) \quad (\text{A.11})$$

for all ordered pairs $f_{p_3} \in \mathcal{F}_{p_3}$ and $s_{p_3} \in \mathcal{S}_{p_3}$, and for all $\ell \in \mathbb{S}_{s_{p_3}}$.

The same constraints as in §A.1.1 are also included in this GFLoP instance.

A.1.3 High-complexity tri-objective GFLoP instance of §7.3.3

The facility and criteria sets are

$$\mathbb{F} = \{1[\text{Long-range solar-powered cameras}], 2[\text{Short-range solar-powered cameras}]\}$$

and

$$\mathbb{C} = \{1[\text{Average daily solar radiation}], 2[\text{Inter-facility distance}], 3[\text{Gridposts visible in } \mathbb{I}_1]\},$$

respectively.

The facility-SZ assignment matrices are given by

$$\mathbf{F}_4 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{F}_5 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad \text{and} \quad \mathbf{F}_6 = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}.$$

The set-pairs are the following: p_4 (all facilities) for the first and third objectives corresponds to the facility and SZ sets $\mathcal{F}_4 = \{1, 1, 2, 2\}$ and $\mathcal{S}_4 = \{1, 2, 1, 2\}$. Furthermore, p_5 (SZ 1 facilities) and p_5^* (SZ 2 facilities) for the second objective correspond to the sets $\mathcal{F}_5 = \{1, 2\}$ and $\mathcal{S}_5 = \{1, 1\}$, and to the sets $\mathcal{F}_{5^*} = \{1, 2\}$ and $\mathcal{S}_{5^*} = \{2, 2\}$, respectively.

The first objective, according to (4.2), is to

$$\text{maximise } V_1 \tag{A.12}$$

subject to (4.3a); that is

$$V_1 \leq v_{c_1}(\ell) + M(1 - \tilde{x}_{p_4}(\ell)) \tag{A.13}$$

for all ordered pairs $f_{p_4} \in \mathcal{F}_{p_4}$ and $s_{p_4} \in \mathcal{S}_{p_4}$, and for all $\ell \in \mathbb{S}_{s_{p_4}}$.

The second objective, according to (4.15), is to

$$\text{maximise } V_2 \tag{A.14}$$

subject to (4.16a); that is

$$V_2 \leq v_{c_2}(\ell, \ell^*) + M(2 - \tilde{x}_{p_5}(\ell) - \tilde{x}_{p_5^*}(\ell^*)) \tag{A.15}$$

for all ordered pairs of facilities and SZs $f_{p_5} \in \mathcal{F}_{p_5}$ and $s_{p_5} \in \mathcal{S}_{p_5}$, all $f_{p_5^*} \in \mathcal{F}_{p_5^*}$ and $s_{p_5^*} \in \mathcal{S}_{p_5^*}$, and for all $\ell \in \mathbb{S}_{s_{p_5}}$ and $\ell^* \in \mathbb{S}_{s_{p_5^*}}$.

For the specification of the MCLP of the third objective, the IZ is the entire terrain surface, and visibility is determined with respect to \mathbb{I}_1 , and so

$$y_{p_4}(i^*) = \begin{cases} 1, & \text{if point of interest } i^* \in \mathbb{I}_1 \text{ receives service from at least} \\ & \text{one facility-SZ pair in set-pair } p_4 \text{ from which cover is demanded,} \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, the facility-dependent inter-site criteria value (§3.4.3) takes the values

$$v_{f_{p_4}, c}(\ell, i^*) = \begin{cases} 1, & \text{if point of interest } i^* \in \mathbb{I}_1 \text{ is serviced from a} \\ & \text{facility-SZ pair of } f_{p_4} \in \mathcal{F}_{p_4} \text{ and } s_{p_4} \in \mathcal{S}_{p_4}, \text{ and if } \ell \in \mathbb{S}_{s_{p_4}}, \\ 0, & \text{otherwise.} \end{cases}$$

The set that specifies which locations provide cover to the points in \mathbb{I}_1 is specified as

$$\mathbb{N}_{p_4}(i^*) = \{\ell \mid v_{f_{p_4},c}(\ell, i^*) = 1\}, \ell \in \mathbb{S}_{s_{p_4}}, \quad (\text{A.16a})$$

for all ordered pairs $f_{p_4} \in \mathcal{F}_{p_4}$ and $s_{p_4} \in \mathcal{S}_{p_4}$. The objective, according to (4.7) is to

$$\text{maximise } V_3 = \sum_{i^* \in \mathbb{I}_1} y_{p_4}(i^*) \quad (\text{A.17})$$

subject to the constraints (4.8)–(4.9), that is

$$\sum_{\ell \in \mathbb{N}_{p_4}(i^*)} \tilde{x}_{p_4}(\ell) - y_{p_4}(i^*) \geq 0, \quad (\text{A.18})$$

$$y_{p_4}(i^*) \in \{0, 1\} \quad (\text{A.19})$$

for all $i^* \in \mathbb{I}_1$, and subject to the same constraints as in §A.1.1 (p_1 is substituted by p_4).

A.2 Large-area examples

This section is devoted to the formulation of the objectives and constraints for the large-area GFLoP instances of §7.4.

A.2.1 High-complexity bi-objective GFLoP instance of §7.4.1

The facility and criteria sets are

$$\mathbb{F} = \{1[\text{Long-range cameras}], 2[\text{Short-range cameras}]\}$$

and

$$\mathbb{C} = \{1[\text{Gridposts visible in } \mathbb{I}_1], 2[\text{Gridposts visible in } \mathbb{I}_2], 3[\text{Inter-facility distance}]\},$$

respectively. The facility-SZ assignment matrix is given by

$$\mathbf{F}_7 = \begin{bmatrix} 3 & 0 \\ 0 & 8 \end{bmatrix}.$$

The set-pairs are the following: p_6 (all facilities) for the first objective corresponds to the facility and SZ sets $\mathcal{F}_6 = \{1, 2\}$ and $\mathcal{S}_6 = \{1, 2\}$. Furthermore, p_7 (facilities of type $f = 2$) for the second objective corresponds to the sets $\mathcal{F}_7 = \{2\}$ and $\mathcal{S}_7 = \{2\}$.

For the specification of the MCLP of the first objective, the IZ is the entire terrain surface, and visibility is determined with respect to \mathbb{I}_1 , and so

$$y_{p_6}(i_1^*) = \begin{cases} 1, & \text{if point of interest } i_1^* \in \mathbb{I}_1 \text{ receives service from at least} \\ & \text{one facility-SZ pair in set-pair } p_6 \text{ from which cover is demanded,} \\ 0, & \text{otherwise.} \end{cases}$$

For the specification of the MCLP of the second objective, the IZ is the river areas, and visibility is determined with respect to \mathbb{I}_2 , and

$$y_{p_7}(i_2^*) = \begin{cases} 1, & \text{if point of interest } i_2^* \in \mathbb{I}_2 \text{ receives service from at least} \\ & \text{one facility-SZ pair in set-pair } p_7 \text{ from which cover is demanded,} \\ 0, & \text{otherwise.} \end{cases}$$

The facility-dependent inter-site criteria values (§3.4.3) take the values

$$v_{f_{p_6},c}(\ell, i_1^*) = \begin{cases} 1, & \text{if point of interest } i_1^* \in \mathbb{I}_1 \text{ is serviced from a} \\ & \text{facility-SZ pair of } f_{p_6} \in \mathcal{F}_{p_6} \text{ and } s_{p_6} \in \mathcal{S}_{p_6}, \text{ and if } \ell \in \mathbb{S}_{s_{p_6}}, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$v_{f_{p_7},c}(\ell, i_2^*) = \begin{cases} 1, & \text{if point of interest } i_2^* \in \mathbb{I}_2 \text{ is serviced from a} \\ & \text{facility-SZ pair of } f_{p_7} \in \mathcal{F}_{p_7} \text{ and } s_{p_7} \in \mathcal{S}_{p_7}, \text{ and if } \ell \in \mathbb{S}_{s_{p_7}}, \\ 0, & \text{otherwise,} \end{cases}$$

The set that specifies which locations provide cover to the points in \mathbb{I}_1 is specified as

$$\mathbb{N}_{p_6}(i_1^*) = \{\ell \mid v_{f_{p_6},c}(\ell, i_1^*) = 1\}, \ell \in \mathbb{S}_{s_{p_6}}, \quad (\text{A.20a})$$

for all ordered pairs $f_{p_6} \in \mathcal{F}_{p_6}$ and $s_{p_6} \in \mathcal{S}_{p_6}$, while set that specifies which locations provide cover to the points in \mathbb{I}_2 is specified as

$$\mathbb{N}_{p_7}(i_2^*) = \{\ell \mid v_{f_{p_7},c}(\ell, i_2^*) = 1\}, \ell \in \mathbb{S}_{s_{p_7}}, \quad (\text{A.21a})$$

for all ordered pairs $f_{p_7} \in \mathcal{F}_{p_7}$ and $s_{p_7} \in \mathcal{S}_{p_7}$.

The objectives, according to (4.7) are to

$$\text{maximise } V_1 = \sum_{i_1^* \in \mathbb{I}_1} y_{p_6}(i_1^*) \quad (\text{A.22})$$

and to

$$\text{maximise } V_2 = \sum_{i_2^* \in \mathbb{I}_2} y_{p_7}(i_2^*) \quad (\text{A.23})$$

subject to the constraints (4.8)–(4.9), that is

$$\sum_{\ell \in \mathbb{N}_{p_6}(i_1^*)} \tilde{x}_{p_6}(\ell) - y_{p_6}(i_1^*) \geq 0, \quad (\text{A.24})$$

$$\sum_{\ell \in \mathbb{N}_{p_7}(i_2^*)} \tilde{x}_{p_7}(\ell) - y_{p_7}(i_2^*) \geq 0, \quad (\text{A.25})$$

$$y_{p_6}(i_1^*), y_{p_7}(i_2^*) \in \{0, 1\} \quad (\text{A.26})$$

for all $i_1^* \in \mathbb{I}_1$ and $i_2^* \in \mathbb{I}_2$.

The set-pair for the first additional constraint is p_8 (SZ 1) which corresponds to $\mathcal{F}_8 = \{1\}$ and $\mathcal{S}_8 = \{1\}$, while $p_8^* = p_7$. The lower bound constraint is $c_{\min}^3 = 3\,000$ metres. From (4.21), the constraint is therefore

$$3\,000 \leq v_{c_3}(\ell, \ell^*) + M(2 - \tilde{x}_{p_8}(\ell) - \tilde{x}_{p_8^*}(\ell^*)) \quad (\text{A.27})$$

for all ordered pairs of facilities and SZs $f_{p_8} \in \mathcal{F}_{p_8}$ and $s_{p_8} \in \mathcal{S}_{p_8}$, all $f_{p_8^*} \in \mathcal{F}_{p_8^*}$ and $s_{p_8^*} \in \mathcal{S}_{p_8^*}$, and for all $\ell \in \mathbb{S}_{s_{p_8}}$ and $\ell^* \in \mathbb{S}_{s_{p_8^*}}$.

The lower bound constraint is $c_{\min}^3 = 6\,000$ metres. From (4.21), the constraint is therefore

$$6\,000 \leq v_{c_3}(\ell, \ell^*) + M(2 - \tilde{x}_{p_7}(\ell) - \tilde{x}_{p_7^*}(\ell^*)) \quad (\text{A.28})$$

for all ordered pairs of facilities and SZs $f_{p_7} \in \mathcal{F}_{p_7}$ and $s_{p_7} \in \mathcal{S}_{p_7}$, all $f_{p_7^*} \in \mathcal{F}_{p_7^*}$ and $s_{p_7^*} \in \mathcal{S}_{p_7^*}$, and for all $\ell \in \mathbb{S}_{s_{p_7}}$ and $\ell^* \in \mathbb{S}_{s_{p_7^*}}$. The formulation is completed with the inclusion of the mandatory constraints (3.4)–(3.6).

A.2.2 Low-complexity tri-objective GFLoP instance of §7.4.2

The facility and criteria sets are

$$\mathbb{F} = \{1[\text{Solar-powered facility}]\}$$

and

$$\mathbb{C} = \{1[\text{Distance from rivers}], 2[\text{Average daily solar radiation}], 3[\text{Inter-facility distance}]\},$$

respectively. The facility-SZ assignment matrix is given by

$$\mathbf{F}_8 = \begin{bmatrix} 3 \\ 8 \end{bmatrix}.$$

The set-pairs are the following: p_9 (SZ 1) for the first objective corresponds to the facility and SZ sets $\mathcal{F}_9 = \{1\}$ and $\mathcal{S}_9 = \{1\}$. Furthermore, p_{10} (facilities of type $f = 2$) for the second objective corresponds to the sets $\mathcal{F}_{10} = \{1\}$ and $\mathcal{S}_{10} = \{2\}$.

The first objective, according to (4.2), is to

$$\text{maximise } V_1 \tag{A.29}$$

subject to (4.3a); that is

$$V_1 \leq v_{c_1}(\ell) + M(1 - \tilde{x}_{p_9}(\ell)) \tag{A.30}$$

for all ordered pairs $f_{p_9} \in \mathcal{F}_{p_9}$ and $s_{p_9} \in \mathcal{S}_{p_9}$, and for all $\ell \in \mathbb{S}_{s_{p_9}}$.

The second objective, according to (4.2), is to

$$\text{maximise } V_2 \tag{A.31}$$

subject to (4.3a); that is

$$V_2 \leq v_{c_2}(\ell) + M(1 - \tilde{x}_{p_{10}}(\ell)) \tag{A.32}$$

for all ordered pairs $f_{p_{10}} \in \mathcal{F}_{p_{10}}$ and $s_{p_{10}} \in \mathcal{S}_{p_{10}}$, and for all $\ell \in \mathbb{S}_{s_{p_{10}}}$.

The third objective, according to (4.17), is to

$$\text{minimise } V_3 \tag{A.33}$$

subject to (4.18a); that is

$$V_3 \geq v_{c_3}(\ell, \ell^*)(\tilde{x}_{p_9}(\ell) + \tilde{x}_{p_{10}^*}(\ell^*) - 1) \tag{A.34}$$

for all ordered pairs of facilities and SZs $f_{p_9} \in \mathcal{F}_{p_9}$ and $s_{p_9} \in \mathcal{S}_{p_9}$, all $f_{p_{10}^*} \in \mathcal{F}_{p_{10}^*}$ and $s_{p_{10}^*} \in \mathcal{S}_{p_{10}^*}$, and for all $\ell \in \mathbb{S}_{s_{p_9}}$ and $\ell^* \in \mathbb{S}_{s_{p_{10}^*}}$.

The set-pair for the first additional constraint is p_9 (SZ 1) which corresponds to $\mathcal{F}_9 = \{1\}$ and $\mathcal{S}_9 = \{1\}$, while $p_9^* = p_{10}$. The lower bound constraint is $c_{\min}^3 = 3000$ metres. From (4.21), the constraint is therefore

$$3000 \leq v_{c_3}(\ell, \ell^*) + M(2 - \tilde{x}_{p_9}(\ell) - \tilde{x}_{p_9^*}(\ell^*)) \tag{A.35}$$

for all ordered pairs of facilities and SZs $f_{p_9} \in \mathcal{F}_{p_9}$ and $s_{p_9} \in \mathcal{S}_{p_9}$, all $f_{p_9^*} \in \mathcal{F}_{p_9^*}$ and $s_{p_9^*} \in \mathcal{S}_{p_9^*}$, and for all $\ell \in \mathbb{S}_{s_{p_9}}$ and $\ell^* \in \mathbb{S}_{s_{p_9^*}}$.

The lower bound constraint is $c_{\min}^3 = 6\,000$ metres. From (4.21), the constraint is therefore

$$6\,000 \leq v_{c_3}(\ell, \ell^*) + M(2 - \tilde{x}_{p_{10}}(\ell) - \tilde{x}_{p_{10}^*}(\ell^*)) \quad (\text{A.36})$$

for all ordered pairs of facilities and SZs $f_{p_{10}} \in \mathcal{F}_{p_{10}}$ and $s_{p_{10}} \in \mathcal{S}_{p_{10}}$, all $f_{p_{10}^*} \in \mathcal{F}_{p_{10}^*}$ and $s_{p_{10}^*} \in \mathcal{S}_{p_{10}^*}$, and for all $\ell \in \mathbb{S}_{s_{p_{10}}}$ and $\ell^* \in \mathbb{S}_{s_{p_{10}^*}}$. The formulation is completed with the inclusion of the mandatory constraints (3.4)–(3.6).