

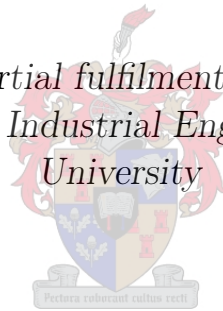
# Determining optimal primary sawing and ripping machine settings in the wood manufacturing chain

by

Berndt Gerald Lindner

15150526

*Thesis presented in partial fulfilment of the requirements for  
the degree of Master of Industrial Engineering at Stellenbosch  
University*



Department of Industrial Engineering,  
University of Stellenbosch,  
Private Bag X1, Matieland 7602, South Africa.

Supervisors:

Supervisor: Prof. P.J. Vlok

Co-supervisor: Mr. C.B. Wessels

March 2014

# Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Signature: .....

Date: .....

Copyright © 2014 Stellenbosch University  
All rights reserved.

# Abstract

## Determining optimal primary sawing and ripping machine settings in the wood manufacturing chain

B.G. Lindner (15150526)

*Department of Industrial Engineering,  
University of Stellenbosch,  
Private Bag X1, Matieland 7602, South Africa.*

Thesis: MEng (Industrial Engineering)

March 2014

For wood manufacturers around the world, the single biggest cost factor is known to be its raw material. Thus maximum utilisation, specifically volume recovery of this raw material, is of key importance for the industry. The wood products industry consists of several interrelated manufacturing steps for converting trees into logs and logs into finished lumber. At most primary and secondary wood processors the different manufacturing steps are optimised in isolation or based on operator experience. This can lead to suboptimal decisions and a substantial waste of raw material. The objective of this study was to determine the optimal machine settings for two interrelated operations, namely the sawing and ripping operations which have traditionally been optimised individually.

A model, having two decision variables, was developed which aims to satisfy market demand at a minimal cost. The first decision was how to saw the log supply into different thicknesses by choosing specific sawing patterns. The second was to decide on a rip saw's settings, namely part priority values, which determines how the products from the primary sawing operation are ripped into products of a certain thickness and width.

The techniques used to determine the machine settings included static simulation with the SIMSAW software to represent the sawing operation and mixed integer programming to model the ripping operation. A metaheuristic, namely the Population Based Incremental Learning algorithm, was the link between the two operations and determined the optimal settings for the combined pro-

cess.

The model's objective function was formulated to minimise the cost of production. This cost included the raw material waste cost and the over or under production cost. The over production cost was estimated to include the stock keeping costs. The under production cost was estimated as the buy-in cost of purchasing the under supplied products from another wood supplier.

The model performed well against current decision software available in South Africa, namely the Sawmill Production Planning System package, which combines simulation (SIMSAW) and mixed integer programming techniques to maximise profit. The model added further value in modelling and determining the ripping priority settings in addition to the primary sawing patterns.

# Uittreksel

## Bepaling van optimale primêre -en kloofsaag masjienstellings in die houtvervaardigingsketting

*(“Determining optimal primary sawing and ripping machine settings in the wood manufacturing chain”)*

B.G. Lindner (15150526)

*Departement van Bedryfs Ingenieurswese,  
Universiteit van Stellenbosch,  
Privaatsak X1, Matieland 7602, Suid Afrika.*

Tesis: MIng (Bedryfs Ingenieurswese)

Maart 2014

Die grootste enkele koste vir houtprodukvervaardigers wêreldwyd is dié van hulle roumateriaal. Die maksimale gebruik van rou materiaal, of volume herwinning, is dus van primêre belang vir hierdie industrie. Die vervaardigingsproses in die houtprodukte-industrie bestaan uit ‘n verskeidenheid interafhanklike stappe om bome na stompe te verwerk en stompe na eindprodukte. By meeste primêre -en sekondêre houtvervaardigers word die verskillende vervaardigingsstappe in isolasie ge-optimeer. Hierdie praktyk lei tot sub-optimale besluite en ‘n vermorsing van roumateriale. Die doelwit van hierdie studie was om die optimale masjienverstellings vir twee interafhanklike prosesse, die primêre -en kloofsaag prosesse, te bepaal. Tradisioneel word hierdie twee prosesse individueel optimeer.

‘n Model met twee besluitnemingsveranderlikes is ontwikkel wat poog om die markaanvraag te bevredig teen ‘n minimum koste. Die eerste besluit was watter saagpatroon gekies moet word om die stompe in die regte dikte produkte te saag. Die tweede besluit was wat die kloofsaagstellings, ook bekend as prioriteitswaardes, moet wees sodat die regte wydte produkte gesaag word.

Die tegnieke wat gebruik is sluit statiese simulاسie met SIMSAW sagteware in om die primêre saagproses te modelleer en gemengde heelgetalprogrammering (“*mixed integer programming*”) om die kloofsaagproses te modelleer. ‘n Metaheuristiek genaamd die “*Population Based Incremental Learning*” algoritme,

was die skakel tussen die twee operasies om die optimale masjienstellings vir die proses te bepaal.

Die model se doelfunksie was geformuleer om die koste van produksie te minimeer. Hierdie koste sluit die roumateriaal afvalkoste en die kostes van oor -en onderproduksie in. Die oorproduksiekoste was 'n skatting van die voorraadkoste. Die onderproduksiekoste was 'n skatting van die koste om voorraad van 'n ander verskaffer aan te koop.

Die model het goed opgeweeg teen die beskikbare besluitnemingssagteware in Suid Afrika, die "*Sawmill Production Planning System*", wat 'n kombinasie van SIMSAW en 'n gemengde heelgetalprogrammeringstegniek is. Die model het verder waarde toegevoeg deur die kloofsaag se prioriteitswaardes te modelleer saam met die primêre saagpatrone.

# Acknowledgements

I would like to thank God for blessing me with this opportunity and helping me along this difficult but rewarding path.

I would like to express my sincere gratitude to the following people and organisations ...

- Brand Wessels who has provided outstanding input, editing and guidance throughout the whole project. This was above expectations and was greatly needed.
- Tanya Visser who initially guided the project and helped further my knowledge in the exciting field of Operations Research.
- P.J. Vlok who steered the project to completion and provided much needed guidance on content, structure, writing tools and handing in of drafts.
- Case Company for its valuable effort, time, data and sponsorship.
- Paul for his valuable insight, time and discussions.
- Gerrie for his time, data and explanations.

# Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Uittreksel</b>	<b>iv</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>Contents</b>	<b>vii</b>
<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xiii</b>
<b>List of Acronyms</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	3
1.2 Problem Statement . . . . .	8
1.3 Objective . . . . .	9
1.4 Thesis Overview . . . . .	10
<b>2 Literature Study</b>	<b>11</b>
2.1 Operations Research in the Sawmilling Industry . . . . .	11
2.1.1 Linear and Integer Programming . . . . .	13
2.1.2 Dynamic Programming . . . . .	14
2.1.3 Simulation . . . . .	22
2.2 Decision Support System (DSS) in the Sawmilling Industry . . . . .	24
2.2.1 WOODCIM . . . . .	25
2.2.2 SAWSIM and SAWSIM-LP . . . . .	25
2.2.3 AUTOSAW . . . . .	25
2.2.4 SIMSAW and Sawmill Production Planning System (SPPS) . . . . .	27
2.2.5 Metaheuristics for the Log Positioning Problem . . . . .	29
2.3 Rip and Chop Saw Operations . . . . .	31
2.3.1 Simulators and DSS . . . . .	32



2.3.2	Optimisation . . . . .	34
2.3.3	Priority Values . . . . .	36
2.4	Conclusion . . . . .	37
<b>3</b>	<b>Materials and Methods</b>	<b>39</b>
3.1	Modelling Components . . . . .	39
3.1.1	Objective Function(s) . . . . .	39
3.1.2	Decision Variables . . . . .	40
3.1.3	Constraints . . . . .	41
3.1.4	Operations Research Techniques . . . . .	42
3.2	Logs Shape Characteristics . . . . .	43
3.2.1	Calculating the Log Volume . . . . .	44
3.2.2	Logs Generated . . . . .	45
3.3	Sawing Operation: Simulation Data . . . . .	45
3.4	Ripping Operation: Integer Programming . . . . .	47
3.5	Metaheuristics for Optimising the Decision Variables . . . . .	48
3.6	Population Based Incremental Learning (PBIL) Algorithm . . . . .	50
3.6.1	Initialisation . . . . .	50
3.6.2	Learning . . . . .	51
3.6.3	Mutation . . . . .	51
3.6.4	Termination . . . . .	51
3.6.5	PBIL Steps . . . . .	52
3.6.6	PBIL Parameters . . . . .	53
3.7	Proposed Model . . . . .	53
3.8	Software Used . . . . .	54
3.9	Comparison Strategy . . . . .	54
3.10	Conclusion . . . . .	55
<b>4</b>	<b>Model Proposed</b>	<b>56</b>
4.1	Formulating the Model . . . . .	56
4.1.1	Operation 1: Sawing the Logs . . . . .	56
4.1.2	Operation 2: Optimally Ripping the Flitches . . . . .	59
4.1.3	Fitness Cost Functions . . . . .	61
4.2	PBIL Algorithm . . . . .	63
4.2.1	Formulation of Decision Variables as Bits . . . . .	63
4.2.2	Model's Constraints within PBIL . . . . .	65
4.3	Simulation of Flitches' Maximum Edged Board Width . . . . .	65
4.4	Modelling Software Used . . . . .	66
4.5	Conclusion . . . . .	66
<b>5</b>	<b>Parameter Estimation from Case Data</b>	<b>68</b>
5.1	Possible Sawing Patterns . . . . .	68
5.2	Financial Parameters . . . . .	68
5.2.1	Raw Material Waste Cost . . . . .	69

<i>CONTENTS</i>	<b>ix</b>
5.2.2 Under Production Cost . . . . .	70
5.2.3 Over Production Cost . . . . .	71
5.3 Machine Parameters . . . . .	72
5.4 Supply and Demand Scenarios . . . . .	73
5.5 Conclusion . . . . .	81
<b>6 Results and Discussion</b>	<b>82</b>
6.1 Supply and Demand Scenarios . . . . .	82
6.2 Model's Best Solutions Found . . . . .	83
6.3 PBIL's Best Parameters Found . . . . .	86
6.4 PBIL's Parameter Values Re-run . . . . .	87
6.5 Supply versus Demand Results . . . . .	95
6.6 Model's Optimum Decision Variables . . . . .	95
6.7 Conclusion . . . . .	102
<b>7 Conclusions and Recommendations</b>	<b>104</b>
7.1 Improved upon SPPS . . . . .	104
7.2 Implementing Solutions and a Possible DSS . . . . .	105
7.3 Further Modelling Improvements . . . . .	106
7.3.1 Supply of Logs as another Decision Variable . . . . .	107
7.3.2 Stochastic Model . . . . .	107
7.3.3 Dynamic Model . . . . .	107
7.3.4 Operation 3: Optimally Chopping the Lumber . . . . .	107
7.3.5 Recovering Side Edged Products . . . . .	109
7.3.6 Quality Tolerances: Bark/Wane and Knots . . . . .	109
7.3.7 Objective Function(s) . . . . .	110
7.4 Final Remarks . . . . .	111
<b>List of References</b>	<b>112</b>
<b>Appendices</b>	<b>119</b>
<b>A Volume of Logs Supplied for Scenarios</b>	<b>120</b>
<b>B Ripping Machine's Optimisation Strategy for Case Company</b>	<b>124</b>
<b>C Decision Making for Case Company</b>	<b>125</b>
<b>D Flitches' Maximum Board Width Simulated</b>	<b>126</b>
<b>E Model's Code in AIMMS 3.13 Environment</b>	<b>148</b>

# List of Figures

1.1	Illustration of important wood manufacturing operations. . . . .	1
1.2	SA lumber recovery and costs . . . . .	2
1.3	Illustrations to highlight the many ways to saw a log . . . . .	4
1.4	Log sawing methods . . . . .	5
1.5	Two possible sawing patterns for a certain sized log . . . . .	6
1.6	Screen shot of SIMSAW 6 software package . . . . .	7
1.7	Two ways to rip a flitch at operation 2 . . . . .	8
1.8	Proposed wood sawing operations to model in this study. . . . .	10
2.1	Optimising the sawing pattern of a log by Geerts (1984) . . . . .	15
2.2	Lumber manufacturing modelled by Faaland and Briggs (1984) . . .	16
2.3	Conversion processes modelled by Reinders (1992) . . . . .	16
2.4	Possible outputs of algorithm by Reinders and Hendriks (1989) . .	18
2.5	Sawing operations modelled by Todoroki and Rönqvist (1999) . .	18
2.6	Secondary processing modelled by Todoroki and Rönqvist (1999) .	19
2.7	Example flitch consisting of two edged pieces (Todoroki and Rön- qvist, 2002) . . . . .	20
2.8	SAWSIM: Cant breakdown . . . . .	26
2.9	SAWSIM 3-D view . . . . .	26
2.10	Screen shot of SIMSAW 6 software package . . . . .	27
2.11	Screen shot of SPPS software . . . . .	28
2.12	An integrated DSS for the entire lumber production chain in SA . .	28
2.13	Log shape with sweep and taper . . . . .	30
2.14	Comparison of metaheuristics for SA sawmill processing . . . . .	30
2.15	Conventional fixed blade set-up for rip saws . . . . .	32
2.16	Fixed blade gang rip saw operation . . . . .	33
2.17	Gang rip saws-movable blade and arbour set-ups . . . . .	33
2.18	Main ROMI-3 interface window . . . . .	34
2.19	Patterns of channel widths for the optimal ripping solution . . . .	35
2.20	Optimising chop saws found in the industry . . . . .	36
3.1	Log shape with sweep and taper . . . . .	43
3.2	Log volume calculation . . . . .	44
3.3	Log sawing methods . . . . .	46

3.4	Framesaw . . . . .	46
3.5	Principle of simulation and optimisation with metaheuristics . . . . .	48
3.6	Comparison of metaheuristics for SA sawmill processing . . . . .	49
3.7	Populations and probability vectors in the PBIL . . . . .	50
3.8	PBIL algorithm in pseudo code . . . . .	52
3.9	PBIL steps in model . . . . .	53
3.10	Tools and techniques used in this project . . . . .	54
4.1	Two possible sawing patterns for a certain log at sawing operation . . . . .	57
4.2	Fitch's maximum board width with sweep and taper . . . . .	58
4.3	Two ways to rip a fitch at operation 2 . . . . .	60
4.4	The proposed model at a glance with variables and techniques used . . . . .	67
5.1	Four possible sawing patterns . . . . .	69
5.2	South African (SA) Lumber Price Index (LPI) September 2013 . . . . .	71
5.3	Framesaw's blade and kerf . . . . .	72
5.4	Box plot explained . . . . .	74
5.5	Logs supplied: Scenario 1 . . . . .	75
5.6	Lumber demanded: Scenario 1 . . . . .	76
5.7	Logs supplied: Scenario 2 . . . . .	77
5.8	Lumber demanded: Scenario 2 . . . . .	78
5.9	Logs supplied: Scenario 3 . . . . .	79
5.10	Lumber demanded: Scenario 3 . . . . .	80
6.1	The model's total cost under different PBIL parameters: Scenario 1 . . . . .	88
6.2	The model's best cost breakdowns after re-run: Scenario 1 . . . . .	89
6.3	The model's total cost re-run and compared to SPPS: Scenario 1 . . . . .	90
6.4	Model randomly searching compared to SPPS's solution: Scenario 2 . . . . .	91
6.5	The model's total costs under different PBIL parameters: Scenario 2 . . . . .	92
6.6	The model's best PBIL parameters re-run: Scenario 2 . . . . .	93
6.7	The model's total costs under different PBIL parameters: Scenario 3 . . . . .	94
6.8	Supply versus demand results: Scenario 1 . . . . .	96
6.9	Supply versus demand results: Scenario 2 . . . . .	97
6.10	Supply versus demand results: Scenario 3 . . . . .	98
6.11	Proposed model versus SPPS's decisions . . . . .	103
7.1	Two possible length sawing decisions at operation 3 . . . . .	108
7.2	Side edged lumber recovered . . . . .	109
7.3	Partial board product demonstrating wane . . . . .	110
A.1	Log supply volume: Scenario 1 . . . . .	121
A.2	Log supply volume: Scenario 2 . . . . .	122
A.3	Log supply volume: Scenario 3 . . . . .	123
B.1	Ripping machine's optimisation strategy for case company. Email. . . . .	124

*LIST OF FIGURES*

**xii**

C.1 Decision making for case company. Email correspondence . . . . . 125

# List of Tables

2.1	Number of logs to satisfy demand in Todoroki and Rönnqvist (2002) study . . . . .	22
3.1	PBIL Parameters suggested and tested . . . . .	53
4.1	Decisions encoded into PBIL’s binary Solution Vector (SV) . . . . .	64
4.2	Decisions encoded into PBIL’s binary SV . . . . .	64
4.3	SV generated in PBIL with sawing decision constraint . . . . .	64
5.1	Financial fitness measurements . . . . .	68
5.2	Traditional sawlog classes . . . . .	70
5.3	Weighted average pine saw log prices at roadside 2013 Q2 . . . . .	70
5.4	Kerf of blades for machines . . . . .	72
5.5	Different scenarios run through the model . . . . .	73
6.1	Different scenarios run through the model . . . . .	83
6.2	Results of model (PBIL) compared to SPPS (MIP): Scenario 1 . . . . .	84
6.3	Results of model (PBIL) compared to SPPS (MIP): Scenario 2 . . . . .	84
6.4	Results of model (PBIL) compared to SPPS (MIP): Scenario 3 . . . . .	84
6.5	SPPS’s optimal decision variables: Scenario 1 . . . . .	99
6.6	Proposed model’s near optimal decision variables: Scenario 1 . . . . .	99
6.7	SPPS’s optimal decision variables: Scenario 2 . . . . .	100
6.8	Proposed model’s near optimal decision variables: Scenario 2 . . . . .	100
6.9	SPPS’s optimal decision variables: Scenario 3 . . . . .	101
6.10	Proposed model’s near optimal decision variables: Scenario 3 . . . . .	101

# List of Acronyms

**AIMMS** Advanced Interactive Multidimensional Modelling System

**DP** Dynamic Programming

**DSS** Decision Support System

**ERP** Enterprise Resource Planner

**GA** Genetic Algorithm

**IP** Integer Programming

**LP** Linear Programming

**LPI** Lumber Price Index

**LR** Learning Rate

**MC** Monte Carlo

**MIP** Mixed Integer Programming

**MP** Mutation Probability

**MS** Mutation Shift

**PBIL** Population Based Incremental Learning

**PV** Probability Vector

**OR** Operations Research

**ORSSA** Operations Research Society of South Africa

**ROMI-RIP** ROugh Mill RIP-first simulator

**RSA** Republic of South Africa

**SA** South Africa

**SP** Sawing Pattern

*LIST OF ACRONYMS*

**xv**

**SPPS** Sawmill Production Planning System

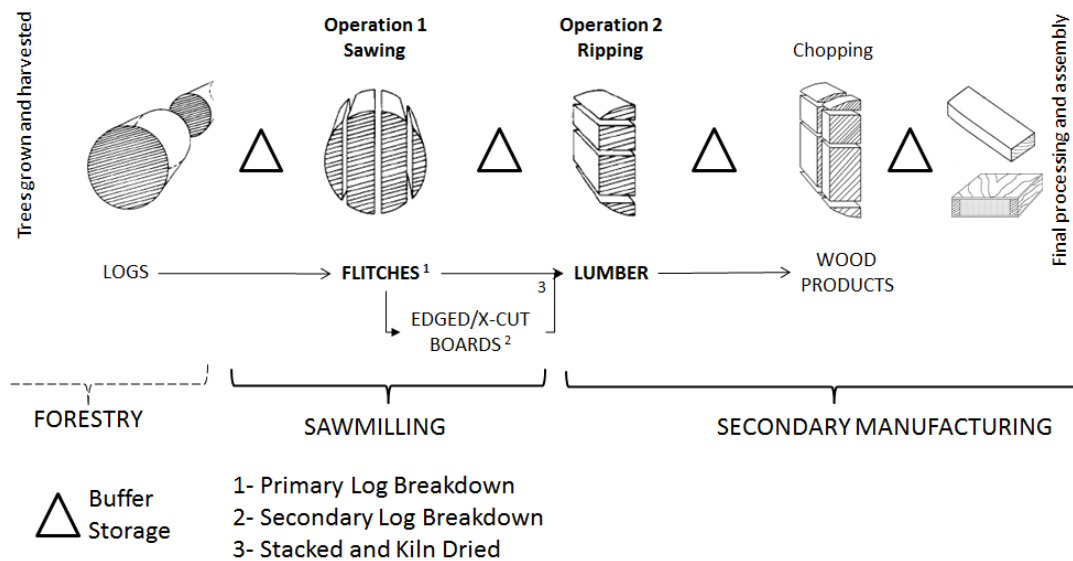
**SV** Solution Vector



# Chapter 1

## Introduction

For wood manufacturers around the world, the single biggest cost factor is known to be its raw material (Lundahl, 2007; Buehlmann *et al.*, 2011; Crickmay and Associates, 2013a). Thus maximum utilisation, specifically volume recovery of this raw material, is of key importance for the industry. The wood product industry consists of several interrelated manufacturing steps for converting trees into logs and logs into finished lumber (Faaland and Briggs, 1984; Todoroki and Rönnqvist, 1999; Wessels *et al.*, 2006). A generalised manufacturing environment is illustrated in Figure 1.1.



**Figure 1.1:** Illustration of important manufacturing operations in the wood supply industry. Operations one and two are the focus areas in this study.

The wood product industry can be separated into primary and secondary wood products. Primary products include products that are produced directly from

logs, including paper, lumber, plywood and composite materials such as particleboard and orientated strand board. Secondary products are produced from primary products and include flooring, cabinets, furniture, doors and windows (Buck, 2009).

As mentioned the single biggest cost for wood manufacturers is known to be its raw material. However the wood industry wastes a lot of its raw material. South Africa (SA)'s national volume recovery for the formal sawmilling sector is shown in Figure 1.2. Also included is the estimated cost breakdowns of the formal sawmilling sector. Saw logs (raw material) is the biggest cost factor by far.

	2012	2011	2010	2009	2008
Volumetric recovery (%)	<b>47.4</b>	47.5	47.8	46.8	47.5
Cost Item	Percentage of total cost of production (incl. overheads) (%)				
Sawlog cost/m <sup>3</sup> produced	<b>54.6</b>	54.3	54.3	52.5	54.8
Variable (excl. sawlog)	<b>2.6</b>	2.5	2.3	2.3	2.2
Depreciation and insurance	<b>3.6</b>	3.7	4.5	5.2	4.0
Power and water	<b>3.2</b>	3.1	2.8	2.5	1.8
Rent and rates	<b>0.6</b>	0.5	0.4	0.4	0.4
People costs	<b>17.8</b>	17.6	18.1	18.8	17.7
Kiln drying	<b>2.4</b>	2.8	2.7	2.9	2.7
Maintenance	<b>7.1</b>	7.2	7.2	7.5	8.6
Vehicles	<b>2.3</b>	2.3	2.0	1.8	2.0
Overheads	<b>5.4</b>	5.6	5.0	5.4	4.9
Other fixed costs	<b>0.4</b>	0.4	0.7	0.7	0.8
Total (%)	<b>100.0</b>	100.0	100.0	100.0	100.0

**Figure 1.2:** SA sawmilling volumetric recovery and cost breakdown (Crickmay and Associates, 2013a)

As seen in Figure 1.2 less than half of the log volume is converted into timber products, the rest being chips, shavings, and shrinkage. Chips and sawdust produced by the saw blades are commonly referred to as “waste”. This is because its selling price is extremely low in comparison to that of solid wood.

For wood manufacturers the economic benefits of increasing volume recovery of its raw material can be substantial. Van Zyl (2011) estimated that for a medium-sized South African sawmill, having an annual log intake of 100 000 m<sup>3</sup>, a 1% increase in volume recovery will result in additional profit of about R2.2 million annually. Prior to the sawmilling process there is volume loss with trees being converted into logs and after the sawmilling process further volume losses occur during secondary manufacturing. In the secondary manufacturing environment raw material costs can comprise 40 to 70 % of total

manufacturing expenses (Buck, 2009). It is estimated that for a secondary manufacturer increasing volume yield by 1% can potentially save 2% of total production costs (Buehlmann *et al.*, 2003). Koch *et al.* (2009) estimated this 1% waste represents a loss of € 45 000 per year for an international secondary manufacturing firm.

Apart from the economic advantages there are also environmental benefits linked to volume recovery improvements. If there is more raw material recovered, i.e. less waste from the tree, then there is in turn less consumption of trees to meet the proposed demand (Van Zyl, 2011).

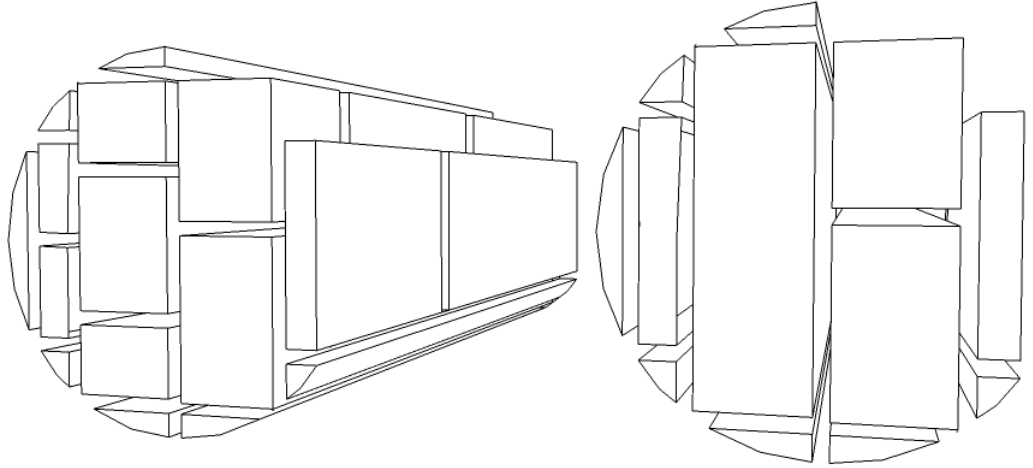
The important question is what is preventing manufacturers from increasing their volume recovery by as little as 1%? One influence is the saw kerf (blade thickness) size. Studies have shown a difference in theoretical yield lying between 2.4% and 3.7% when a 4.8 mm saw kerf size is replaced with a 2.8 mm saw (Lundahl, 2007, 112). Another influence is shape characteristics of the logs supplied, such as diameter, taper and sweep. However the most challenging problem is the complexity inherent in optimally matching the naturally varying log supply to the fluctuating wood product demand.

Sawmills constantly struggle to maximise value yield and reduce the effects of the “sawmill paradox”. This is the problem that only about half of the logs’ volume will be sawn into wood products, the rest being chips or sawdust and that half of the sawn wood products will become low priced products due to low demand (Lundahl, 2007). Figure 1.3 illustrates this more graphically. In addition wood products are produced within a hybrid batch/flow process environment from a highly variable natural resource with often unpredictable processing characteristics (Kapp, 1997, 8).

In this study “optimisation” or Operations Research (OR) techniques are explored to assist with processing decisions for the linked operations scoped, namely the primary sawing and ripping of wood material. The aim is to satisfy market demand optimally, namely at the maximum volume recovery.

## 1.1 Background

At sawmills logs are converted, or broken down, into boards by a series of sawing operations (Todoroki and Rönqvist, 1999). As illustrated in Figure 1.1 the log breakdown can be viewed as a two stage process. Logs are sawn into slabs of wood known as flitches during the primary stage and flitches are further processed to produce edged (sawn length-wise) and trimmed (sawn width-wise) pieces (Todoroki and Rönqvist, 1997). The term flitch refers to a piece primarily sawn from a log having two sawn faces and two waney/bark



**Figure 1.3:** There are many ways to convert a circular log into demanded rectangular wooden products of certain dimensions. Each of these is described as a different Sawing Pattern (SP). Each SP will produce different products and volume output.

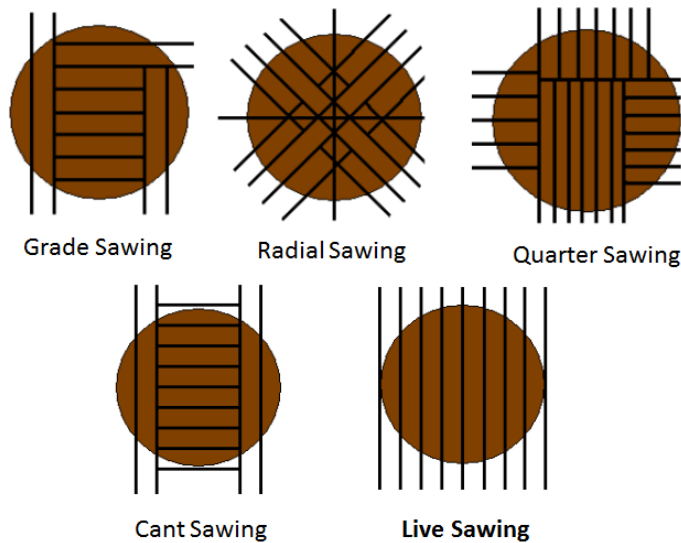
(outer layer of tree) edges. The sawing operations in the wood manufacturing industry are of a sequential nature and they need to be optimised simultaneously to attain a global optimal recovery (Todoroki and Rönqvist, 1999).

Secondary manufacturers of solid wood products, saw kiln-dried flitches or lumber boards into dimension parts of specified sizes, qualities and quantities according to customer orders (Buehlmann *et al.*, 2011). Secondary manufacturers are also referred to as rough millers in the United States (Buehlmann *et al.*, 2011). It is important to note that the differentiation of sawmillers and secondary manufacturers in Figure 1.1 is not always so clear. Sometimes the sawmill will do the ripping operation as well. Also the forestry or forest products industry, is seen to include the sawmill process (Rönqvist, 2003).

This project focusses on the South African (SA) pine manufacturing chain. The South African sawmilling industry is unique in that it obtains virtually 100% of its raw material supply from exotic plantations. New Zealand is second with 97% (Bredenkamp *et al.*, 2012, Ch. 8.7). South Africa's total forest plantation area constitute 1.3 million hectares. These areas are either intensely managed softwood (51%) and/or hardwood plantations (Van Zyl, 2011). The hardwoods used in South Africa include eucalyptus and wattle species, and are used mostly for pulp and board mills (Du Plessis, 2010). Softwoods are mostly used for dimensionally sawn timber processed into standard sizes, which is in turn commonly used in the construction industry and the manufacturing of solid wood products (Du Plessis, 2010). The plantation softwoods processed in South Africa is collectively known as SA Pine, which includes five different species. These plantations are managed by the forestry industry, which

are usually separate profit centres from the sawmillers, or separate companies (Wessels *et al.*, 2006).

There are a few different sawing techniques used at the primary breakdown stations of sawmills. The most common patterns include cant, live, quarter, grade and radial sawing amongst others. These sawing operations are illustrated in Figure 1.4.

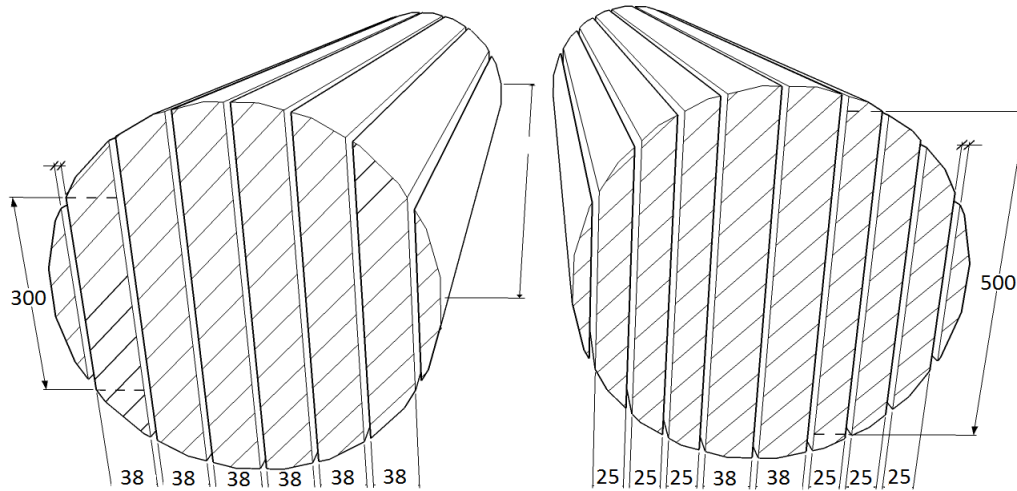


**Figure 1.4:** Log sawing methods (Van Zyl, 2011).

As mentioned this project focusses on the South African (SA) pine manufacturing chain. Specifically manufacturers utilising live sawing patterns at the log breakdown operation. This study might also find application for hardwood manufacturers (Wessels *et al.*, 2006) and/or those utilising other sawing patterns and subsequent ripping operations in other countries. For example New Zealand uses live sawing for log breakdown of pine material (Todoroki and Rönnqvist, 1999).

As illustrated in Figure 1.3 and Figure 1.5 the way in which the saws and their spaces relative to each other are set is described as a Sawing Pattern (SP). This is an important decision, machine setting, to be optimised at the primary sawing station. Determining the optimum sawing patterns can be seen as a three-dimensional knapsack problem. The primary breakdown of the log can be formulated as the well known one-dimensional knapsack problem. Further breakdown of boards and/or flitch assortments into boards can be formulated as a two-dimensional knapsack problem. The latter can be thought of as the problem of fitting rectangles of various dimensions and values into a circle

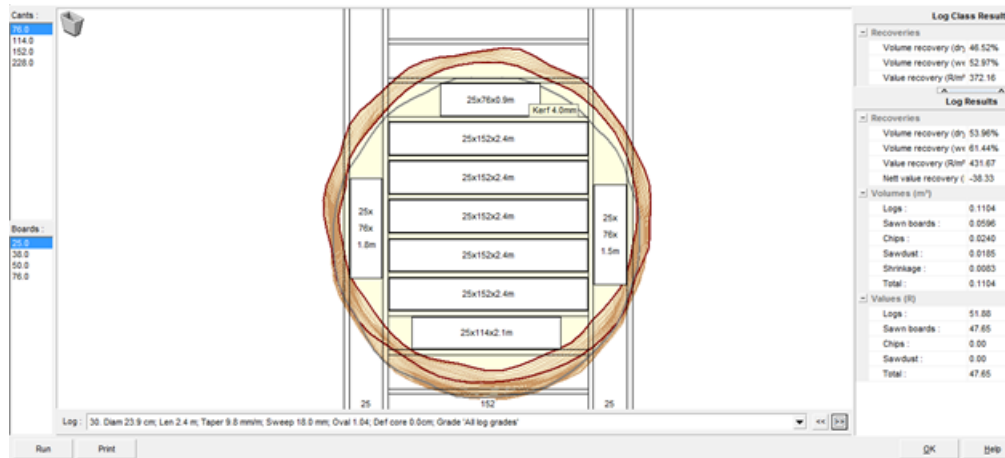
in such a way that the total value of the circle is maximised (Reinders and Hendriks, 1989). Optimising the sawing patterns has been of interest to various researchers for decades (Faaland and Briggs, 1984; Geerts, 1984; Hallock *et al.*, 1979).



**Figure 1.5:** Example of two possible sawing patterns for a certain sized log at the primary sawing operation. Each pattern will produce different dimensioned products and volume outputs.

There are many models and software programs available to help improve the industry in optimising its decisions as described in the literature review. Most notably in South Africa is the SIMSAW and Sawmill Production Planning System (SPPS) packages, with similar packages existing worldwide. SIMSAW 6 is a sawing simulation tool which predicts the sawn product (board) recovery from logs, given certain inputs (Wessels *et al.*, 2001). User inputs include the possible sawing patterns, log shape characteristics and some machine settings for the primary and secondary log breakdown machines. The SPPS package uses simulation data from SIMSAW and with the use of linear and mixed integer programming techniques, maximises the total profit of a sawmill, subject to constraints set by the user (Wessels *et al.*, 2006). A screen-shot of the SIMSAW package is shown in Figure 1.6.

Secondary manufacturers, sometimes referred to as rough millers in America, saw solid wood parts from kiln-dried lumber boards bought from sawmills and other wholesalers. These lumber boards are further processed so that the specified part sizes are obtained and defects are sawn out. The processed products are then used for final assembly of furniture, cabinets, doors, windows and other products (Buehlmann and Thomas, 2001).

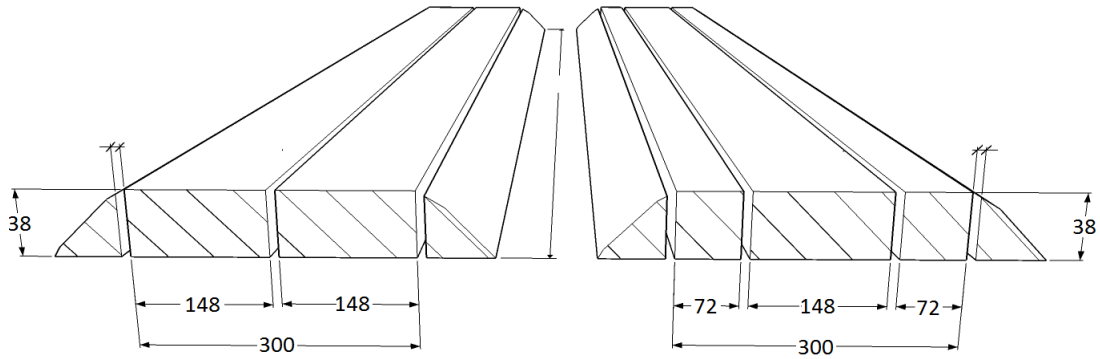


**Figure 1.6:** Screen shot of SIMSAW 6 software package.

The industry's rip and chop saw machines have changed from using extensive manual decision sawing systems to machines capable of automated decision making in an effort to increase yield, amongst other objectives (Zuo, 2003). These sophisticated scanning and computerised systems for the rip (and chop) saw machines have been developed to optimise the value of lumber for secondary manufacturers (Cothrell and Higgins, 2003). These machines are commonly referred to as “optimisers” in the industry. One of the most important decisions that these “optimising” machines try to solve is illustrated in Figure 1.7.

Although secondary manufacturers have started to incorporate a large amount of automatic and computerised equipment, especially in gang-rip first processes, many of the process settings still rely on human decision making. Examples include: choosing the appropriate grade of lumber for processing; designing the optimal arbour for the gang-rip saw; and defining suitable part priority values for the chop saw (Zuo, 2003). Currently industrial optimising chop saws use a part-priority approach to optimise for a production run. With this approach the operator manually sets priorities for certain difficult-to-cut parts (Maness *et al.*, 2009). Recently these priority approaches have been incorporated into modern ripping machines (Cothrell and Higgins, 2003).

One of the methods used to optimally rip (or chop) an incoming board/fitch, is to have each product possible to be ripped have an assigned priority value, sometimes called the dollar or decision value. Depending on this value the machine's computer will determine which combination of parts can be ripped out of the board and the combination with the highest value is chosen (Cothrell and Higgins, 2003). These part priority values thus force the machine to rip more of one product than the other. These values are usually determined according to the supply of products needed to fulfil expected demand.



**Figure 1.7:** Example of two ways to rip a flitch at the ripping operation. Each with different volume and product outputs. Priority settings/values usually determine which products to rip out.

Wood product parts that are in higher demand are given a considerably higher priority value and less desirable parts are given a lower value in comparison. However, the values are usually assigned arbitrarily, since in reality it is difficult to determine the actual value of any part flowing through the mill. Thus value strategies are very dependent on the assumptions of the person running the ripping simulation. In addition these static part valuation strategies lack the ability to consider diminishing need as parts are sawn (Thomas, 1996).

Determining what the priority value settings should be at the ripping operation is one of the two key decision variables to be optimised in this study. The other being the combined optimal sawing patterns at the primary sawing operation.

## 1.2 Problem Statement

Modern lumber manufacturers are becoming increasingly like other processing industries, characterised by large, continuous flows of raw materials through its supply chain (Lundahl, 2007). Sustainable usage of raw material, its rising cost, and increasing international competition are some of the reasons why sawmills must improve their production practices (Vuorilehto, 2001). Most modern sawmills use optimisation techniques to maximise yield from logs (Todoroki and Rönnqvist, 2002). However process optimisation tools in sawmills have not followed the rapid development of automation and increased production speed. Each machine and monitoring system is commonly viewed as a detached system rather than as part of the same process (Lundahl, 2007) with each operation optimising its functionality in isolation from the preceding and subsequent operations (Wessels *et al.*, 2006).



Modern machine centres with their sophisticated optical scanners and powerful computers performing optimisation techniques, usually only find an optimum solution in isolation. They thus tend to maximise value for a single processing piece and do not consider how the solutions fit into the overall production plan (Turner, 2010). A series of local optimum processes will rarely lead to a global optimum for a system.

As highlighted in the literature there is substantial research and development on modelling wood processing operations. Most notably is the software available for SA sawmillers to compute optimal sawing patterns, namely SIMSAW and SPPS. These programs are tailored more specifically to the primary sawing operations (Figure 1.1). The biggest lack in the existing work is that of integrating the operations performed by sawmillers and secondary manufacturers, specifically the ripping operation.

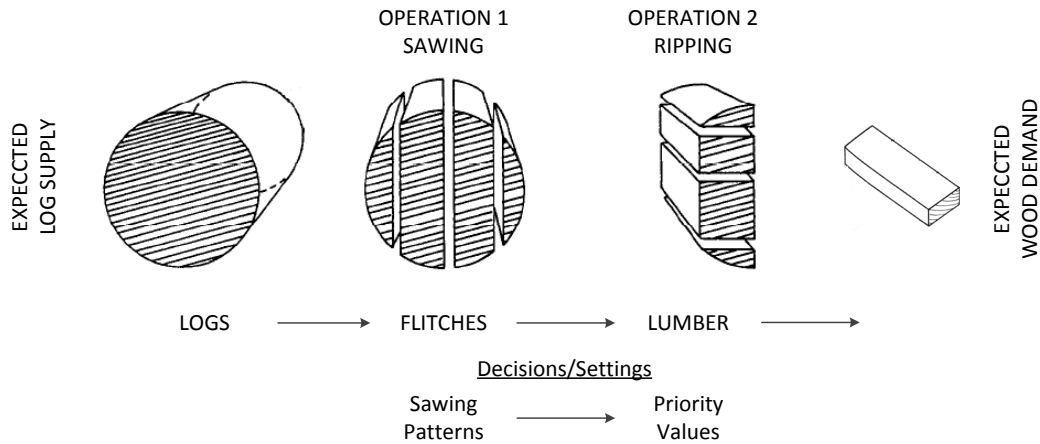
Although SIMSAW's edging simulation can be manipulated somewhat to represent the ripping operation, with simulated results then exported to SPPS, which in turn finds the optimal assignment of sawing patterns (sawing decisions/settings). This edging operation is simulated to only maximise volume (or length) of lumber products. This optimisation strategy fails to take the market demand of specific sized products into account.

Ripping machines within the industry aim to maximise recovery whilst trying to meet the market demand by assigning priority (or decision) values to lumber products (Turner, 2010). This operation and its priority values are not incorporated in SIMSAW and subsequently the SPPS software package. There is thus a need to determine the sawing patterns in combination with rip machine settings, namely the part priority values.

### 1.3 Objective

There is a need to determine the appropriate settings at the sawing and rip machine centre optimisers that are used extensively within the industry. The main decisions for the sawing and ripping machines are the sawing patterns and the priority values respectively. The decision variables or settings to be optimised in this study are illustrated in Figure 1.8.

The purpose of this thesis is to develop a method to find optimal sawing pattern and ripping priority value choices, given a specific log supply and product demand scenario. This is expected to improve upon traditional methods which usually determine these decision variables in isolation. These results will be compared to that of the SPPS software package, which determines optimal sawing patterns but does not cater for the ripping priority value's decision



**Figure 1.8:** Proposed wood sawing operations to model in this study.

variables as well. This being because the ripping (“edging” in SIMSAW) operation’s optimisation strategy is to maximise volume, leaving no setting available to adjust what products should be ripped according to demand.

## 1.4 Thesis Overview

In order to solve the problem stated and determine the mentioned objectives, the rest of this study is structured in the following way:

- Chapter 2 reviews the current research and development applied to the sawing and ripping operations for the wood industry.
- Chapter 3 highlights the materials and methods used in this study. Specifically the approach taken to solve the problem stated and the optimisation techniques that were used.
- Chapter 4 shows how the model solved the problem that was formulated and the modelling software tools used.
- Chapter 5 describes how the important parameters of the model were estimated from real case company data.
- Chapter 6 discusses the results.
- Chapter 7 draws conclusions and proposes future work.

# Chapter 2

## Literature Study

The wood products industry is a unique processing environment. The production of square dimensional lumber from round uneven logs and further production of dimensional wood products from lumber is a manufacturing step unknown to other industries (Buehlmann *et al.*, 2003). Therefore the literature reviewed focusses on modelling techniques specifically for the wood manufacturing industry, although there are many similar modelling techniques applied to other industries, for example the cutting stock problem for paper manufacturers.

This chapter provides background on existing modelling research of the wood sawing operations as shown in Figure 1.1. This includes sawing the logs into flitches/boards, as performed by the sawmill and ripping these flitches/boards into lumber, typically performed by a secondary manufacturer. The process of optimally chopping the lumber into desired length wood products is also included, even though it was not modelled in the study. This is because this process faces the same problem as the ripping process, namely that of choosing what products to saw that will fit into the incoming material. Also this is where priority optimisation strategies were first created and have only recently been incorporated into modern ripping machines (Cothrell and Higgins, 2003). It is also a subsequent process after ripping for most secondary manufacturers and is usually modelled in conjunction with ripping operations (Ye and Maness, 2006).

### 2.1 Operations Research in the Sawmilling Industry

OR, sometimes referred to as management science, is a scientific approach to decision making that seeks to best design and operate a system. This is usually under conditions requiring the allocation of scarce resources (Winston and Goldberg, 2004). Another description defines OR as *the art of giving bad*

*answers to problems to which otherwise worse answers are given* (Saaty, 2004). The objective of OR is to provide a scientific basis to the decision maker for solving the problems involving interaction of various components of a system to find a solution which is in the best interest of the system as a whole. The best solution obtained is known as the optimal decision (Murhty, 2007).

The term Operations Research was coined when British military leaders asked scientists and engineers to analyse several military problems such as the deployment of radar and the management of convoy, bombing, antisubmarine, and mining operations during World War II. This scientific approach to decision making usually involves the use of one or more mathematical models. A mathematical model is a mathematical representation of an actual situation that may be used to make better decisions or simply to understand the actual situation better (Winston and Goldberg, 2004).

Applying OR techniques to the forest and wood product industry has been utilised extensively. D'Amours *et al.* (2008) highlights that the literature dealing with OR in the forest and forest products industry can be divided into two categories:

- Forestry, particularly forest management, harvesting and transportation.
- Supply chain planning for the different products/ markets, such as pulp and paper, bio-fuel, lumber, and engineered wood products.

This project will fall under the supply chain planning of lumber and engineered wood products. Engineered wood products are a range of derivative wood products bounded together with adhesives to form composite materials.

Since the 1960's researchers have looked at the sawn timber recovery optimisation problem from different angles (Du Plessis, 2010). The research methods to date can be crudely categorised under the following headings (Du Plessis, 2010):

- Empirical, where real logs are sawn using a number of processes and where the processes are evaluated.
- Theoretical, where the recovery is calculated mathematically. This includes linear and dynamic programming amongst others.
- Simulation studies, where the sawing process is simulated using artificially created or scanned images of real logs and recovery is calculated based on different input variables set by the user. Metaheuristics are usually combined with simulation to optimise the input variables.

This study will follow the theoretical and simulation route. Different OR techniques applied to the sawn timber recovery problem are described in the following chapters, specifically those for the sawmilling industry and those for secondary manufactures, specifically the ripping operations.

### 2.1.1 Linear and Integer Programming

Linear Programming (LP) is a tool for solving optimisation problems (Winston and Goldberg, 2004). Linear programming uses a mathematical model to describe a theoretical problem. Linear means that all the mathematical functions in this model are required to be linear functions. The word programming does not refer to computer programming; rather, it is essentially a synonym for planning (Hiller and Lieberman, 2010). In 1947, George Dantzig developed an efficient method, the simplex algorithm, for solving linear programming problems (Winston and Goldberg, 2004). The development of linear programming has been ranked among the most important scientific advances of the mid-20th century Hiller and Lieberman (2010).

It is interesting to note that while linear programming has been employed most heavily in industries other than wood products, most notably in the petrochemical industry, the industry in which it was first applied was in fact the wood products industry. Leonid Kantorovich, a Russian mathematician, was given the task of optimising production in the Russian plywood industry and in the course of this work effectively invented what we today call linear programming. For this he was awarded the Nobel Prize (Turner, 2010).

As early as 1972, Laurens developed a linear programming approach for the normative production planning of a South African Pine sawmill. By using OR methods, particularly linear programming, he developed production planning techniques for a Natal sawmill with an annual intake of 75 000 m<sup>3</sup>. Results obtained were useful in developing operational and sawmill design plans. It was however stated that only larger, more efficient mills would benefit from these techniques. It was also recommended that Monte Carlo (MC) simulation methods be used to simulate production and further improve the production planning technique.

An Integer Programming (IP) problem is an LP in which some or all of the variables are required to be non-negative integers. A pure IP problem is one where all the variables are required to be integers. When only some of the variables are required to be integers it is specifically called a Mixed Integer Programming (MIP) problem (Winston and Goldberg, 2004).

Linear and integer programming techniques are typically used in conjunction with simulation data. For example the Sawmill Production Planning System (SPPS) package is a decision making system that uses static simulation data from SIMSAW and with the use of linear and mixed integer programming techniques maximises the total profit of a sawmill, subject to constraints set by the user (Wessels *et al.*, 2006). The SPPS package and other related packages are described in more detail under Section 2.2.

### 2.1.2 Dynamic Programming

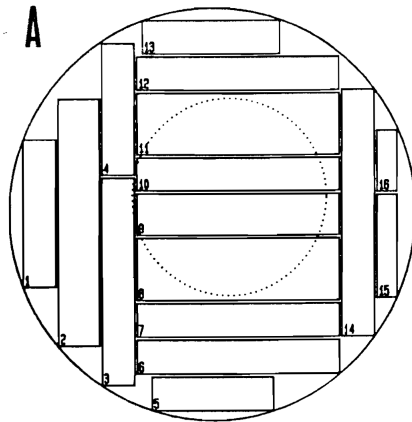
Dynamic Programming (DP) is an OR technique for making a sequence of interrelated decisions. Basically it provides a systematic procedure for determining the optimal combination of decisions (Hiller and Lieberman, 2010). In most applications, DP finds solutions by working backward from the end of a problem toward the beginning. It breaks up a large, unyielding problem into a series of smaller, more traceable problems (Winston and Goldberg, 2004).

In 1984 Geerts developed a mathematical solution for optimising the sawing pattern of a log given its dimensions and its defect core. Geerts constructed the model as a two-stage guillotine type. He modelled the first breakdown of the log into flitches as a one-dimensional dynamic programming algorithm. The next breakdown of the flitches into timber assortments formed the second level of optimisation. Here the value of the timber assortments was the variable to be maximised. Figure 2.1 illustrates this graphically. Option A maximises the total value (determined by fixed price for a certain assortment) whilst option B maximises the volume (no grade prices). Optimisation was done by nesting the two levels which means the output of level two becomes the input to level one and backtracking at each level until the optimal solution is found.

Faaland and Briggs (1984) similarly used dynamic programming techniques to integrate the traditionally separated production steps of producing shorter logs out of tree stems (“log bucking”) and the conversions of these logs into lumber (Figure 2.2).

Faaland and Briggs (1984) also modelled the ripping operation, referred to as the composite board problem (Figure 2.2). This was formulated as a standard integer knapsack with general integer variables and solved by DP. A static value, typically the price, was assigned to each product ripped. This project is different in that it tries to determine what this value should be, a priority value instead of a priced value, to produce parts that meet market demand at a minimal cost.

Reinders followed heavily on the work done by Geerts (1984) and Faaland and Briggs (1984). In Reinders’ PhD (1989) he developed a decision support



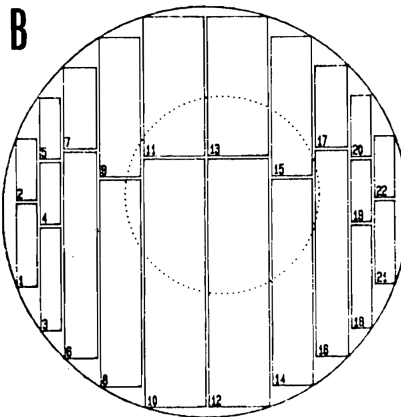
LOG DIAMETER : 500 mm  
 DEFECT CORE DIAMETER : 240 mm  
 RECOVERY PERCENTAGE : 80.7 %

SAW KERFS : vert. 3 mm horz. 4 mm  
 DEFECT CORE DISPL. : 20 mm 20 mm  
 SAWCUT COSTS : 0.00 \$ 0.00 \$

CPU TIME (PDP 11/34) : 245.3 min.

TOTAL NUMBER OF ASSORTMENTS IN THE SET : 42

ASSORTMENT	SELECTED :	
	CLEAR VALUE	ACTUAL VALUE
1) 40 * 180	1.066	1.066
2) 50 * 300	2.445	2.445
3) 40 * 250	1.600	1.600
4) 40 * 160	0.947	0.947
5) 150 * 40	0.798	0.798
6) 250 * 40	1.600	1.600
7) 250 * 40	1.600	1.600
8) 250 * 75	3.000	2.000
9) 250 * 50	2.000	0.667
10) 250 * 40	1.600	0.533
11) 250 * 75	3.000	2.000
12) 250 * 40	1.600	1.600
13) 170 * 40	1.006	1.006
14) 40 * 300	1.956	1.956
15) 25 * 125	0.369	0.369
16) 25 * 75	0.221	0.221
TOTAL :	24.81	20.41



LOG DIAMETER : 500 mm  
 DEFECT CORE DIAMETER : 240 mm  
 RECOVERY PERCENTAGE : 83.1 %

SAW KERFS : vert. 4 mm horz. 4 mm  
 DEFECT CORE DISPL. : 20 mm 20 mm

CPU TIME (PDP 11/34) : 42.8 min.

TOTAL NUMBER OF ASSORTMENTS IN THE SET : 45

ASSORTMENT	SELECTED :	
	CLEAR VALUE	ACTUAL VALUE
1) 25 * 100	0.295	0.295
2) 25 * 75	0.221	0.221
3) 25 * 125	0.369	0.369
4) 25 * 75	0.221	0.221
5) 25 * 75	0.221	0.221
6) 40 * 250	1.600	1.600
7) 40 * 100	0.472	0.472
8) 50 * 250	2.000	1.333
9) 50 * 170	1.258	0.838
10) 75 * 300	3.668	1.222
11) 75 * 170	1.887	0.629
12) 75 * 300	3.668	1.222
13) 75 * 170	1.887	0.629
14) 50 * 250	2.000	0.667
15) 50 * 170	1.258	0.419
16) 40 * 250	1.600	1.600
17) 40 * 100	0.472	0.472
18) 25 * 125	0.369	0.369
19) 25 * 75	0.221	0.221
20) 25 * 75	0.221	0.221
21) 25 * 100	0.295	0.295
22) 25 * 75	0.221	0.221
TOTAL :	24.42	13.76

Figure 2.1: Optimising the sawing pattern of a log given its dimensions and its defect core (Geerts, 1984). Option A - Optimisation by value (price); Option B - Optimisation by volume.

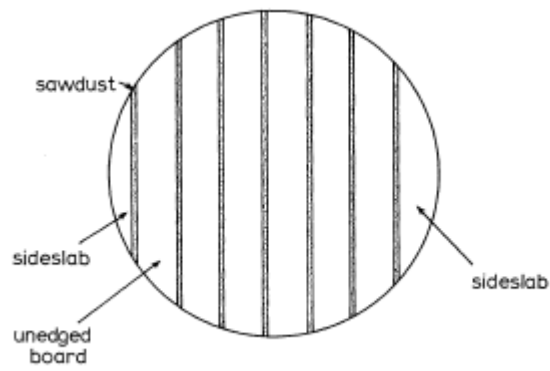


FIGURE 2. Live Sawing Pattern.

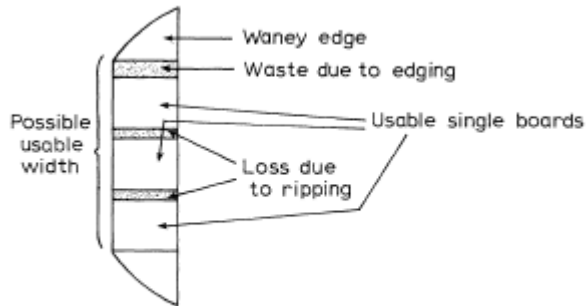


FIGURE 3. A Composite Board.

Figure 2.2: Lumber manufacturing modelled by Faaland and Briggs (1984).

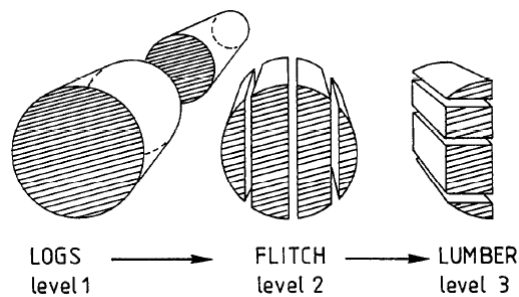


Figure 2.3: Conversion processes modelled by Reinders (1992) from tree trunk to boards.

system for wood processing entitled IDEAS (Integral Decision Effect Analysis System) for integral logistics in centralized wood processing. Reinders and Hendriks (1989) described their study as *Lumber production optimisation*. This optimisation focussed specifically on volume maximisation in conjunction with production planning (Reinders, 1992).



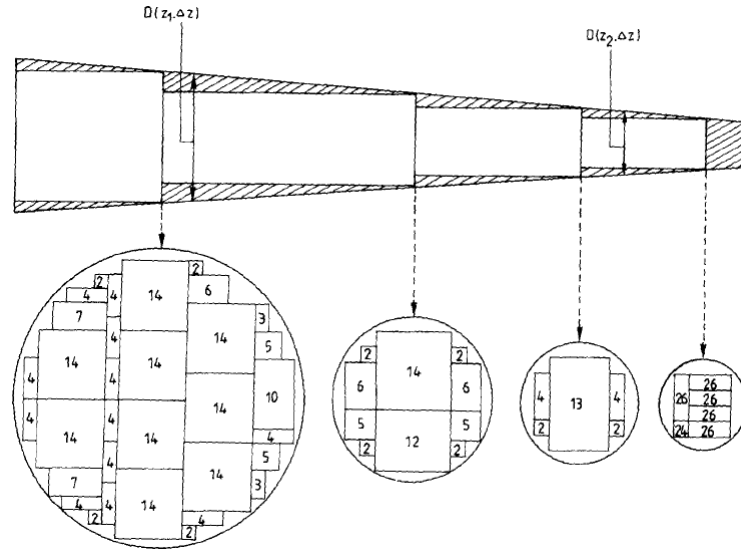
Similarly to Geerts (1984), Reinders and Hendriks (1989) used nested dynamic programming sub algorithms for the different processing levels (Figure 2.3). The criterion used by Reinders and Hendriks (1989) to select a typical sawing pattern was maximization of the value of the timber on the basis of a given list of prices and dimensions of boards to be sawn. This project differs in that it tries to determine what these value should be, a priority value instead of a priced value, to produce parts that meet market demand at a minimal cost. In a simplified manner Reinders and Hendriks (1989) described the problem as:

- Function: How many assortments of logs out of which trees should be made and purchased? And how many board types (dimension and grades) should be made, in what week during which days? This according to a number of objectives. In such a way that value recovery is maximum, service rate is maximum, production cost, inventory cost and cost of labour are minimum. This all anticipating future demand.
- Goal: Integrated optimisation
- Input: Cross-cutting and sawing patterns as defined by company and scenario data
- Output: a multi period production plan

Similarly this project's function is to maximise value recovery by minimising raw material waste cost, maximising service rate by minimising under production cost, and minimising inventory cost with the anticipated future demand. The input includes possible sawing patterns to choose from and the ripping "patterns" possible.

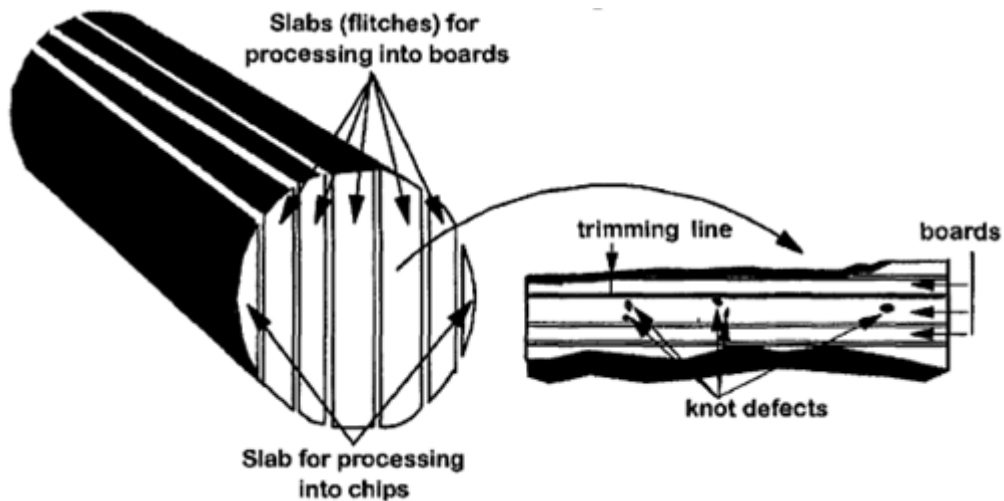
Todoroki and Ronnqvist (1997) viewed the conversion of logs into lumber as a two-stage process. Logs supplied by the forestry industry are first sawn into slabs of wood, known as flitches (operation one in Figure 1.1). This operation is commonly referred to as the primary breakdown (Todoroki and Ronnqvist, 1997). In their research they addressed the secondary breakdown problem, which further processes flitches into edged (sawn length-wise) and trimmed (sawn width-wise) wood pieces. They modelled the secondary breakdown as a set packing problem where the objective was to maximise value. The problems was solved using DP techniques.

Following this Todoroki and Rönqvist (1999) used linked dynamic programming to combine the primary and secondary log breakdown (Figure 2.5). This again is similar to the way Geerts (1984), Faaland and Briggs (1984) and Reinders and Hendriks (1989) solved the log breakdown steps. As seen in Figure 2.6 either volume or value can be maximised, where the value coefficients are



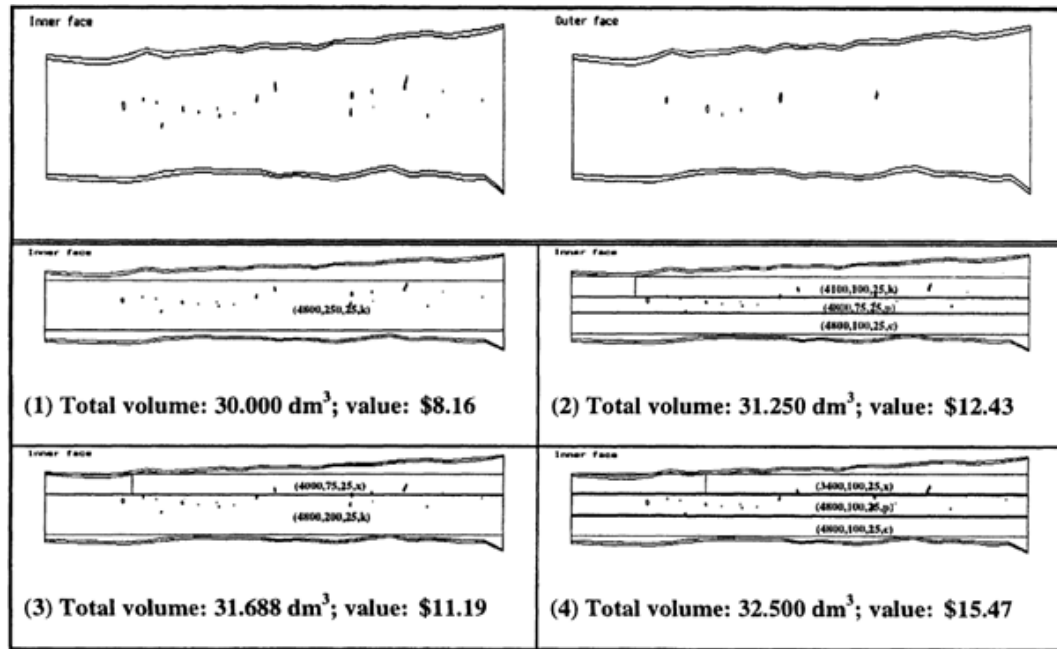
**Figure 2.4:** Example of possible outputs of algorithm developed by Reinders and Hendriks (1989).

set as the fixed price of lumber assortments. This project mimics these optimisation approaches but it differs in that it tries to determine what these static values should be (priority values) to ensure market demand is optimally met at minimal cost.



**Figure 2.5:** Sawing operations modelled and analysed by Todoroki and Rönqvist (1999).

Todoroki and Rönqvist (2001) and Todoroki and Rönqvist (2002) further improved their previous models by having these value coefficients updated



**Figure 2.6:** Example of flitch and secondary log breakdown modelled and analysed by Todoroki and Rönnqvist (1999).

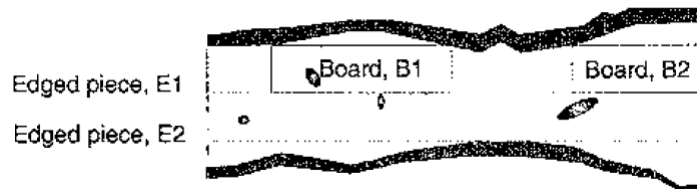
whilst sawing, causing the model to learn and adapt its current supply and demand. Thus these values become dynamic in that they change over time. As they stated that although log throughput is an important consideration for sawmills, the “effectiveness” of the sawmill as a whole is equally or even more important. However the term “effectiveness” is itself subjective. For example they state that volume optimised solutions try to maximise utilisation of each log, value optimised solutions try to maximised value recovery from logs, and a hybrid value optimised solution is a compromise between the two (Todoroki and Rönnqvist, 2002). However each of these approaches does not taken into account the most important aspect of a sawmill, customer demand, since they assume a perfectly elastic demand for each and every timber product. Following this they developed a new optimisation approach which has the value change over time (dynamic), so as to meet market demand, they compare this with previous approaches namely volume and value optimisation as well. These three optimisation approaches are summarised below:

1. Volume Optimisation
2. Value/Grade Optimisation
3. Product Optimisation

Methods 1 and 2 above are similar to the previous optimisation methods by the same author (Todoroki and Rönnqvist, 1999). They found that method

3 reduced overproduction and eliminated underproduction of target products. The optimisation approach was formulated as a set packing problem and solved using dynamic programming. It was also implemented in a log sawing simulation system called AUTOSAW. This system is explained in more detail in Section 2.2.3.

To explain the 3 different methods an example is explained courtesy of Todoroki and Rönqvist (2002). Imagine that at some stage through the optimisation process the ones of thousands combination given by the two edged pieces in Figure 2.7 is evaluated. The full length board E2 is shown as the lower edged piece. E1 on the other represents the upper edged piece that requires trimming to remove excess wand (depicted by the shaded areas of the flitch denoting the presence of the original under bark surface). The boards B1 (containing one know) and B2 (free of defects) which will be the trimmed products from E1 and will be evaluated separately.



**Figure 2.7:** Example flitch consisting of two edged pieces, E1 and E2, with E1 trimmed to produce two boards, B1 and B2 (Todoroki and Rönqvist, 2002).

#### Volume Optimisation:

Continuing with the the optimisation problem example above, if the objective is to maximize volume then the value or cost-coefficient associated with edged piece E1 is equal to the sum of the two board volumes:

$$\text{coeff}_{E1} = \text{volume}_{B1} + \text{volume}_{B2} \text{ (m}^3\text{)} \quad (2.1.1)$$

This is how SIMSAW and subsequently the SPPS package model the edging/ripping operation. This project differs in that it assigns priority values, similar to the optimisation technique explained below, where static values are assigned to each lumber product and maximised.

#### Value/Grade optimization:

With value or grade maximisation, first the grade and dimensions of B1 and B2 are determined, then its value is found according to a timber price list for that product. Typically these prices are subject to premiums where more is paid for wider boards. Suppose B1 is graded as “Knotty” and B2 as “Clear”.

Further suppose that timber prices are given per cubic metre of sawn timber. For value grade maximization the coefficient then becomes:

$$\text{coeff}_{E1} = \frac{\text{volume}_{B1} \times \text{timberprice}_{\text{Knotty}} + \text{volume}_{B2} \times \text{timberprice}_{\text{Clear}}}{\text{volume}_{B2} \times \text{timberprice}_{\text{Clear}}} (\$) \quad (2.1.2)$$

Product Optimisation:

In the second method described above the cost coefficient's value  $\text{coeff}_{E1}$  remains static, it does not change over production time. However for the product optimisation method, previous production of boards influences this coefficient and it thus becomes dynamic:

$$\begin{aligned} \text{coeff}_{E1}^k &= \frac{\text{remaining demand}_{\text{Knotty}} + \text{remaining demand}_{\text{Clear}}}{\text{volume}_{B2} \times \text{timberprice}_{\text{Clear}}} \\ &= \text{coeff}_{\text{Knotty}}^{k-1} - \text{production}_{\text{Knotty}}^{k-1} + \text{coeff}_{\text{Clear}}^{k-1} - \text{production}_{\text{Clear}}^{k-1} \end{aligned} \quad (2.1.3)$$

Here  $\text{production}_{\text{Grade}}^{k-1}$  represents the volume already sawn of a product of a given grade. These values are usually initialised as zero production, i.e.  $\text{production}_{\text{Grade}}^{k-1} = 0$ . However in cases where stockpiles of timber are already present, these can be set as the initial values. Initial values for the cost coefficients have been shown to influence results when applied to a 1D cross-cutting problem (Ronnqvist & Gustafsson 1999). In this application, coefficients for each of the grades,  $\text{coeff}_{\text{Clear}}^{k-1} = 0$ , are initialized with the corresponding order given by the order book. Once demand has been met the value of a product is set to zero.

The formulation for the flitch edging, trimming and grading sub problem was written as a set packing problem in the following way:

$$\text{Maximize } \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (2.1.4)$$

where:

$m$  = the number of width classes

$c_{ij}$  = the value of an edged piece of width  $w_i$

The decision variables are given by:

$$x_{ij} = \begin{cases} 1 & \text{if an edged piece of width } w_i \text{ is cut at level } j \\ 0 & \text{Otherwise} \end{cases} \quad (2.1.5)$$

These cost coefficients values can either be updated on a piece-by-piece basis, or flitch by flitch, or log by log, or even after processing a given number of logs. The log-by-log option was selected since the procedure was required not only to be effective, but also to provide solutions in real time. Also it seemed an appropriate time to check production and update tallies for timber of differing

qualities.

Simulations were repeated for each of the three sawing strategies, namely volume, value and product optimization, and the system was tested against three case studies given by order books 1, 2, and 3 (Table 2.1). The product optimisation strategy outperformed the others for all three order book scenarios. For Order book 2 both volume and value optimisation strategies failed to fulfil all orders with the 430 logs that were available for the simulations.

**Table 2.1:** Number of logs processed to satisfy product demands in each of the different supply scenarios, called order books (Todoroki and Rönnqvist, 2002).

Sawing Strategy	Orderbook 1	Orderbook 2	Orderbook 3
Volume Optimisation	375	>430	424
Value Optimisation	378	>430	409
Product Optimisation	321	332	373

Unlike the dynamic values, this project attempts to meet market demand by assigning static values for a certain run (batch of logs). Similar to the value/grade optimisation approach above (Method 2). But instead of having these values be the price of products, they are any priority values necessary to obtain a good production mix which reflects the market demand.

### 2.1.3 Simulation

Simulation is defined as a technique that imitates the operation of a real-world system as it evolves over time (Winston and Goldberg, 2004). This usually involves using a computer to imitate (simulate) the operation of an entire process or system (Hiller and Lieberman, 2010).

Simulation models can be either static or dynamic. Static simulation models represent a system at particular point in time, this is usually referred to as a Monte Carlo (MC) simulation. In contrast dynamic simulation models represent a system as it evolves over time. Within these two classifications, simulation models may be either deterministic or stochastic. Deterministic simulation models are ones that contain no random variables. Stochastic simulation models contain one or more random variables (Winston and Goldberg, 2004).

Simulation has been used to design and analyse sawmills for over two decades (Dogan *et al.*, 1997). The primary benefit being that there are substantial cost savings compared to experimenting with the real system. Other benefits include the ability to compare suggested systems, to observe circumstances

rarely available, and to experiment with alternative characteristics of material (Dogan *et al.*, 1997). Dogan *et al.* 1997 identified the data requirements for a sawmill simulation to be log population characteristics, process times, buffer capacities, and product output characteristics.

Determining the optimum sawing patterns and positioning methods for logs using simulation software has been studied extensively since the 1960's by several researchers (Peter and Bamping, 1962; McAdoo, 1969; Hallock and Lewis, 1971). In 1976 Hallock *et al.* took their Best Opening Face (BOF) procedure further in a simulation study with the title *is there a "best" sawing method*. Further research has followed through the decades (Lewis, 1985; Steele and Wengert, 1987; Maness and Donald, 1994; Chang *et al.*, 2005).

Since early 1970 the VTT (Technical Research Centre of Finland, Wood Technology) has developed simulation and optimizing software systems for the mechanical forest industry. These initiatives are to increase the value yield and profitability for wood conversion companies (Usenius, 1996; Usenius *et al.*, 1996).

Usenius (1996) developed an optimising software system to "create new profitable operation activities for the wood conversion chain, from forest to the end products". The system aimed to optimally control the number and volume of extra timber products inherently produced from converting the non-homogeneous tree. Additional problems that the optimising system could solve include: how to select wood raw materials into various processes; how to cross cut the stems; how to split and convert the wood in the optimum way into various products according to the demand profile; how to sort and grade the wood raw material and products. The mathematical models were found to simulate real situations and applying their results was estimated to increase the economic yield by 10 to 20%. An advantage of these models are that they provided an overall view of the processes, even if there was no physical production. The model aimed to calculate optimal values but most importantly provide management with a sound basis on which to make decisions, such that "man always makes the final decisions" (Usenius, 1996).

Following extensive research described on simulating sawing operations, there have been many software programs developed for practitioners in the industry around the world. This includes SIMSAW in South Africa and SAWSIM in America amongst others. These software packages also sometimes include support features that try and optimise decisions for the user. This includes Sawmill Production Planning System (SPPS) for SIMSAW and SAWSIM-LP for SAWSIM. These packages are thus by definition Decision Support System (DSS) and are further expanded upon in the following Section 2.2.

## 2.2 DSS in the Sawmilling Industry

A Decision Support System (DSS) is an interactive computer-based system to help managers use data and models to support (rather than replace) their decision making (Hiller and Lieberman, 2010, Ch. 13). For the wood product industry the function of a DSS, specifically for production planning, is to help the decision maker to match the resources to the demanded products (Reinders, 1993).

In 1992 Erasmus developed a *decision support system for the management of lumber mill production* in South Africa. As he states, the number of influencing variables and processing alternatives makes production planning a formidable task and it is practically impossible to find the optimal solution without computer support. Erasmus implemented this DSS for an existing mill. This included a linear programming model, data base software to generate data for the LP model and a spreadsheet aid for detailed production scheduling. During the first two months of the system's use a foremen reported the following improvements: the number of backlog orders were reduced; the production of undesirable products were reduced; there was a marked improvement in the volume recovery. In addition Erasmus showed through sensitivity analysis, specifically interpreting the dual values (Winston and Goldberg, 2004), that by increasing one facility's production capacity and by lowering a certain product's minimum demand constraint profits could be substantially increased.

(Kapp *et al.*, 1999) presented a feasibility study on the development of an integrated manufacturing planning and control system for the South African sawmilling industry. (Kapp, 1997, 33) states that the lack of an integrated manufacturing planning and control system is a major obstacle to improving the competitiveness of the industry. The central theme of his thesis is that of "virtual supply chain integration", and he proposed that this concept will provide improvement opportunities for all supply chain participants. Kapp *et al.* argued that the sawmill supply chain must operate in an integrated fashion, namely for the sake of gaining competitive advantage through cost and value differentiation. The system was designed generically for the average South African sawmilling environment and was intended to provide a sound basis for further investigations of a more detailed nature. The most relevant development recommended by Kapp is described in the next subsections, namely SIMSAW and other related software.

Price *et al.* (2002) looked at using modelling and integrated forestry and sawmill software systems to value the pruned log resource. It was concluded that modelling gives a relatively accurate result of industry's actions. However by virtue of the assumptions made in creating the model, the results should not be regarded as absolute.



### 2.2.1 WOODCIM

Since the early 1970s the WOODCIM (Usenius, 1999) package, which is an integrated optimising software system, has been under development at the Technical Research Centre of Finland (VTT). It consists of several software modules for the sawmill that support research including (Pinto, 2004, 24):

1. Software for optimum selection of stands and bucking of tree stems
2. Program for optimising the number of sawlog classes needed
3. Simulation program for predicting the sawn product yield in sawmilling
4. Software for optimising the manufacturing of components
5. Sawing model based upon linear programming techniques

This project compares its results to a similar software program (SPPS) which uses linear and integer programming (point 5 above) based on Simulation data from the SIMSAW program which also predicts the value yield in sawmilling (point 3). This project also tries to optimise the manufacturing (ripping) of wood products (point 4).

### 2.2.2 SAWSIM and SAWSIM-LP

SAWSIM is a flexible computer program that models the breakdown of logs in any sawmill (HALCO Software Systems Ltd., 2006). This sawing simulation program is capable of producing logs and lumber of various grades depending on the quality of trees generated (Middleton and Zhang, 2009). These trees can be generated and combined with inputs with the Tree and Stand Simulator (TASS) (Middleton and Zhang, 2009).

The SAWSIM-LP optimisation systems were developed to help sawmillers determine “sawing pattern recipes”, which is the optimum mix of sawing patterns to convert a known supply of logs to a specific production mix. In addition to assisting the sawmiller with these shorter term production planning decisions, the system can also be used to evaluate longer term marketing and log procurement questions, such as: at what price does a new product opportunity become profitable for us and at what price premium should we pay for ‘premium’ log diameter/length classes? (HALCO Software Systems Ltd., 2006)

### 2.2.3 AUTOSAW

At the New Zealand Forest Research Institute in 1981, Dr. Oscar Garcia conceived the idea of a glass or see-through log displayed on a computer screen.

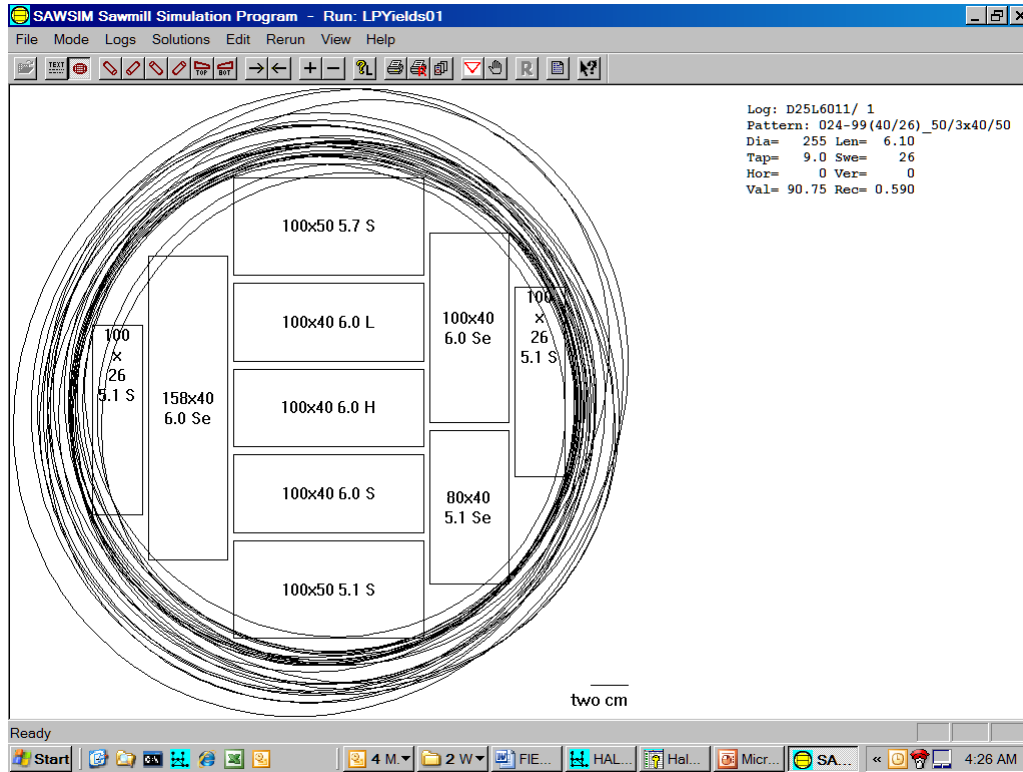


Figure 2.8: SAWSIM: Cant breakdown (Turner, 2010).

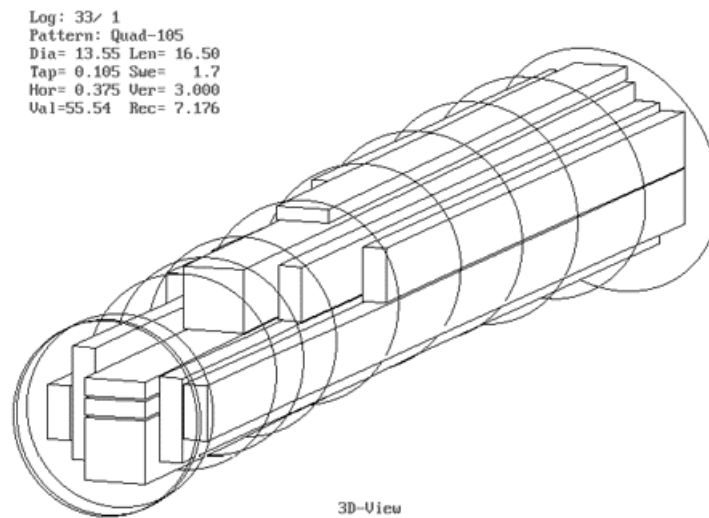


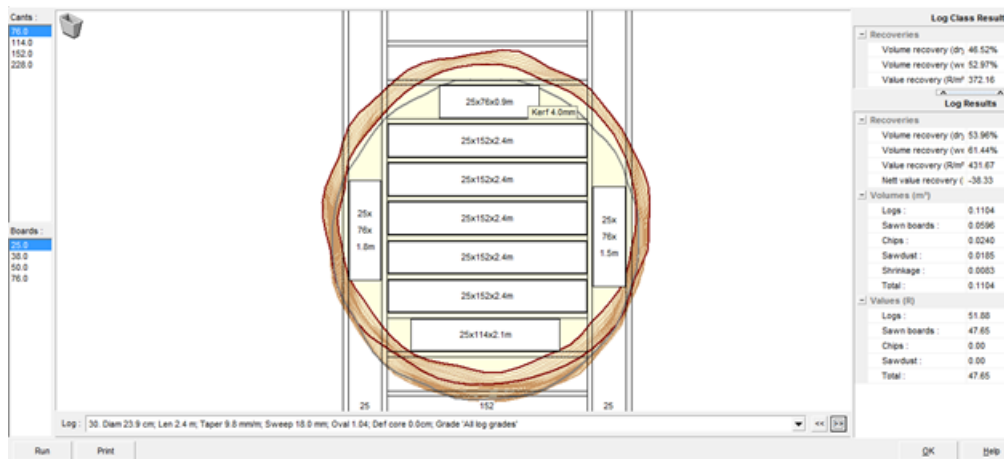
Figure 2.9: SAWSIM 3-D view (HALCO Software Systems Ltd., 2006).

He further developed this, using data and specifications from Jim Park, to produce the prototype pruned log sawing simulator, SEESAW (Todoroki, 1996). In 1990 SEESAW became part of the much more powerful AUTOSAW sim-

ulation system and has been studied and applied abundantly by researchers (Todoroki, 1996).

## 2.2.4 SIMSAW and SPSS

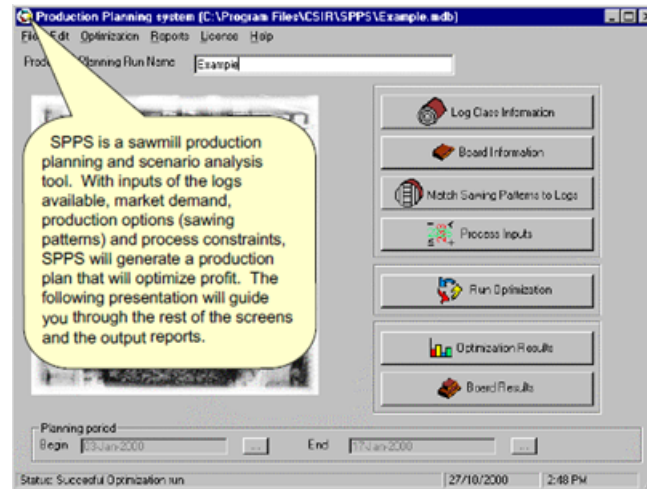
As mentioned there are countless options open to a sawmiller, making it difficult for his decision to be optimal (Singmin, 1978). This caused Reynolds (1970) to propose the principle of how to saw one log in different ways in his simulation study. This further led Dirk van Niekerk in South Africa to develop SIMSAW, a sawing simulation tool (Singmin, 1978). The first version of the SIMSAW program was developed by the CSIR in South Africa in 1975, and the latest versions have the ability to simulate three-dimensional log images (Wessels, 2009b). SIMSAW 6 is the latest free-ware version available to the industry. It is a simulation tool which predicts the end-product (board) recovery from logs given certain inputs (Wessels *et al.*, 2006).



**Figure 2.10:** Screen shot of SIMSAW 6 software package, freely available to the industry.

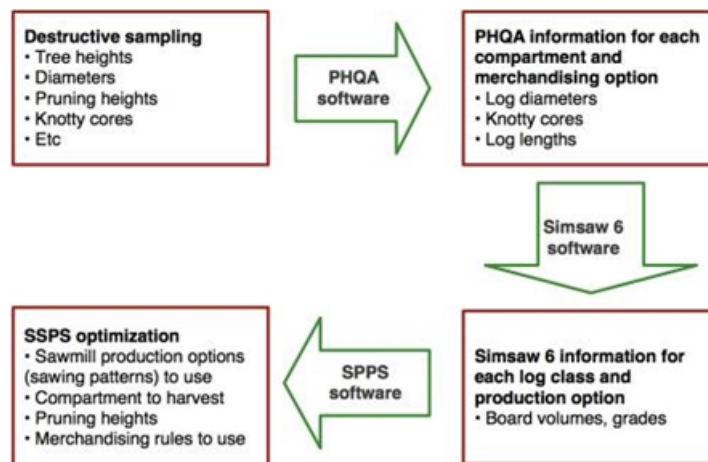
The Sawmill Production Planning System (SPPS) is a decision making system to maximise the total profit, product value less log costs, fixed costs and the cost of buying in timber, subject to constraints set by the user (Wessels *et al.*, 2006). These constraints include kiln drying capacities and fixed market demand. Simulation data from SIMSAW is exported to SPPS which uses linear and mixed integer programming techniques to generate an optimal solution.

Currently a system is being developed and tested to optimally integrate tree harvesting and sawmill operations (Wessels *et al.*, 2006). This consists of using three integrated software models: Pre-Harvest Quality Assessment (PHQA)



**Figure 2.11:** Screen shot of SPPS software. Downloaded from <http://ffp.csir.co.za/products/spps.html>

software, a sawmill simulation package (SIMSAW 6) and a Sawmill Production Planning System (SPPS) (Figure 2.12).



**Figure 2.12:** Information flow in an integrated decision support system for the entire lumber production chain in South Africa (Wessels *et al.*, 2006).

However results generated from SIMSAW and SPPS are typically reserved for sawmilling operations and it is difficult to interpret these results to the separate optimising sawing machines found in secondary manufacturing industry. Specifically the machine's setting regarding the priority of certain products. Priority settings are value/cost coefficients typically found on these optimising sawing machines which increase or lower production of certain products (Ye and Maness, 2006; Maness *et al.*, 2009). These values will force the machine

to cut more or less of the valued products even though this does not produce the optimum volume recovery. This is to minimise over or under production of certain products which will occur if the machine's only objective is to maximise volume recovery of products. This is one of the key objectives of this study as shown in Figure 1.8.

### 2.2.5 Metaheuristics for the Log Positioning Problem

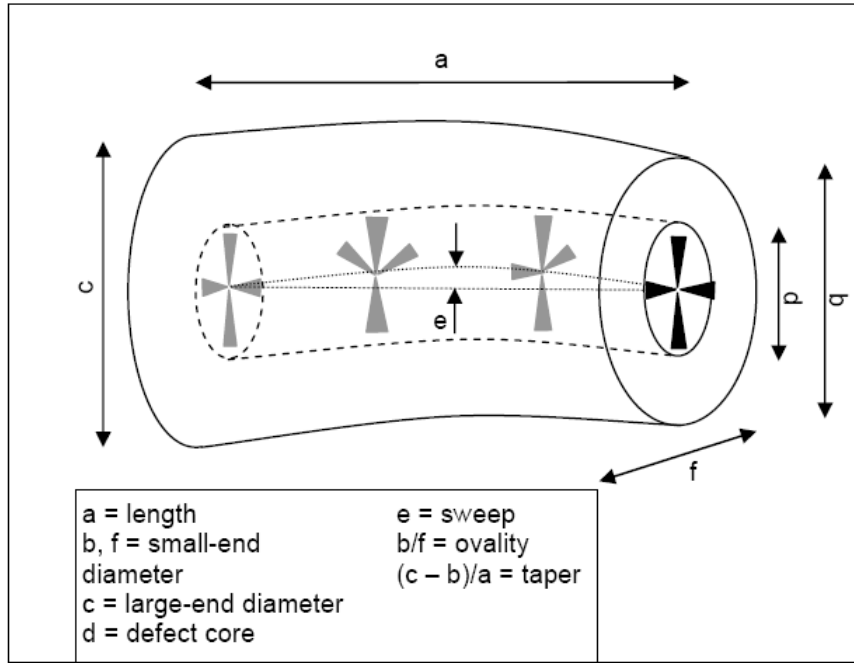
A heuristic method is a procedure that is likely to find a very good feasible solution, but not necessarily an optimal solution, for the specific problem being considered. A metaheuristic is a more specific solution method that provides both general structure and strategy guidelines for developing a specific heuristic method to fit a particular kind of problem. For OR practitioners metaheuristics have become one of the most important techniques to date Hiller and Lieberman (2010).

There have not been many metaheuristics applied specifically to the problem of finding optimal decisions for sawing and rip machine settings to the best of the author's knowledge. The most relevant work found was done by Fathi and Aksakalli (2004) where heuristics methods, based on the principles of local improvement, simulated annealing, and genetic algorithms were used for the ripping operation. This is described further in Section 2.3.2.

However there has been studies conducted on using metaheuristics in the optimisation of log position during sawmill processing. A real log is not a perfect cylinder and has certain asymmetrical shape characteristics influenced by the tree's growth as illustrated in Figure 2.13. Thus depending on which way a log is positioned whilst being sawn, the yield of dimensional wood products will differ.

In his thesis Du Plessis (2010) investigated the use of meta-heuristics in the optimisation of log position during sawmill processing in SA. This includes well known algorithms such as the Particle Swarm Optimisation, Simulated Annealing, Genetic Algorithm (GA), and the PBIL. Wessels *et al.* (2011) developed a search (tentacle) algorithm to determine the optimal log position before being sawn (Operation 1 in Figure 1.1) and it to these various algorithms. This is summarised in Figure 2.14.

Van Zyl (2011) followed by determining the optimal log position during primary breakdown using internal wood scanning techniques and metaheuristic algorithms. Van Zyl (2011) found that a maximum increase in product value of 8.23% was possible when internal knot, branches of the tree, data was considered compared to conventional log positioning rules. When only external shape was considered a maximum increase in product value of 5% was possible com-



**Figure 2.13:** Typical log shape characteristics (Du Plessis, 2010). Sweep and taper are of concern in this study.

Description	Parameters	Abbreviation	Average maximum volume recovery obtained (%)	
			After 90 simulations <sup>a</sup>	After 350 simulations <sup>a</sup>
Population-based incremental learning (PBIL)	Learning rate			
	Mutation probability			
	Mutation shift			
Particle swarm optimisation (PSO)	Velocity = 0.5	PBIL_a	51.20	51.78
	Velocity = 1.0	PBIL_b	51.20	51.85
	Velocity = 1.5	PBIL_c	51.33	51.57
Simulated annealing (SA)	Starting temperature = 5 000; linear cooling	PBIL_d	51.31	51.71
	Starting temperature = 50 000; 10% cooling	PSO_a	51.06	51.06
	Starting temperature = 5 000; 10% cooling	PSO_b	51.43	51.45
Tentacle	Flexible tentacles; constant growth	PSO_c	51.61	51.61
	Linear tentacles; constant growth	SA_a	51.15	51.80
Random selection of positions within search space		SA_b	51.13	51.81
		SA_c	51.24	51.82
Exhaustive search	Range 1	Ten_a	51.71	51.89
	Range 2	Ten_b	51.64	51.80
Global optimum in search space (4 056 simulations)		Ran	51.24	51.79
Recovery at centred, 'horns-up' position (one simulation)		Rng1	51.74	
		Rng2		52.07
		Glob max	52.38	
		Normal	49.87	

<sup>a</sup> Or less if convergence was reached before stated number of simulations

**Figure 2.14:** Comparison of metaheuristics for the cant sawing log position optimisation by Wessels *et al.* (2011).

pared to using conventional log positioning rules. The algorithms applied in the study include the GA, Simulated Annealing, PBIL, Cross-Entropy Method and an alternative self developed algorithm. He also evaluated the investment

decision to purchase an internal log scanning and positioning system.

## 2.3 Rip and Chop Saw Operations

Most of the literature described above focusses on the sawing operations and their machine centres, namely the sawmill's primary and sometimes secondary log breakdown operations. Secondary breakdown refers to the process of breaking down flitches (outputs from the primary log breakdown) into square boards (Todoroki and Ronnqvist, 1997). This is done by the operation referred to as edging and cross-cutting/trimming. SIMSAW and subsequently SPPS do have edging, which can be seen as ripping operations, however it solely tries to maximise volume for its edging operation. Todoroki and Rönqvist (1999), who combined primary and secondary log breakdown does maximises value, where a cost is assigned to each product edged. However this project treats this value as a decision variable to meet market demand, similar to Todoroki and Rönqvist (2002) method as described in Equation 2.1.2. These are commonly referred to as priority values. How this decision affects how the machine saws parts and research regarding this process is describe further in this section.

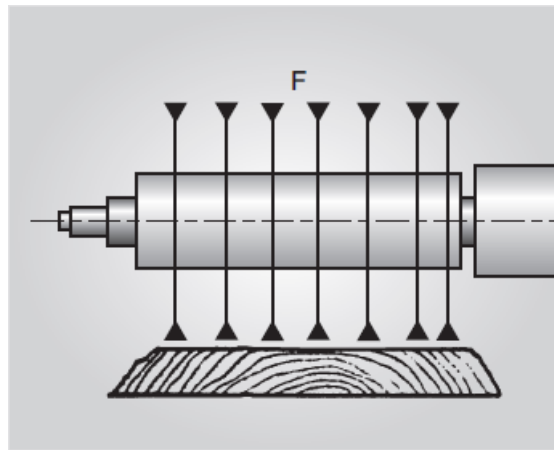
Secondary manufacturing, sometimes referred to as rough milling, includes ripping and cross-cutting operations which have changed from using extensive manual decision sawing systems to machines capable of automated decision making in an effort to increase yield, amongst other objectives Zuo (2003). Secondary manufacturers' operations have received unabated attention from researchers and industry over the last several decades. This is primarily due to the high percentage of total product cost incurred in such operations (Thomas and Buehlmann, 2007).

Secondary manufacturers have to decide whether to rip first or cross-cut (chop) first (Figure 1.1). Determining whether to rip or chop first has been considered one of the most important decisions in the rough mill and has been an ongoing debate since the 1970's (Zuo, 2003, 7). There are many good reasons why the question "Should we be chopping or ripping first in our rough mill?" still exists for many mills. The manufacturer's layout depends on many factors and some of the more important ones are subject to change from day to day or at least year to year (Wiedenbeck, 2001). For many years boards were first chopped to length, then ripped to appropriate widths. This method was efficient and practical whilst yield was not terribly important and automation options were limited for the time. However as lumber became more expensive and automated material-handling systems were developed, rip first systems gained more prominence.

A gang rip saws boards into strips, which are then sawn to required lengths.

Numerous studies have shown that this strategy produces superior results in comparison to the cross-cut first system (Ferrar and King, 1999). This project focusses on secondary manufacturers that rip first since the rip-first process has become the dominant layout in new rough mills during the 1990's (Zuo, 2003, 9).

Today's rip-first rough mills include highly technical operations using scanners and computers to control important steps in the process (Zuo, 2003). The ripping machines are commonly referred to as gang saws (Buehlmann, 1998). There are many types of gang saws but the gang saws having multiple moving blades produce the greatest recovery, assuming that they are equipped with an accurate scanning system (Zuo, 2003). The difference between conventional fixed blade, fixed blade gang rip, and moveable blade gang rip saws are illustrated in Figure 2.15, 2.16 and 2.17 respectively. An important component of rip saws is the arbour. An arbour consists of a series of saws and saw spacings that overlap and interact (Gatchell, 1996).



**Figure 2.15:** Conventional fixed blade set-up for rip saws (Weining Group)

Rising raw lumber costs have initiated the development and use of advanced techniques and equipment to increase rough mill conversion efficiency. The typical technologies used to enhance gang-rip saw operations are: lumber scanning systems; movable fences or moving blade saws; and simulation capability (Zuo, 2003, 20). This study focusses on the simulation capabilities and subsequent optimisation techniques. Current simulation capabilities and optimisation of the ripping operation are described further in the following subsections.

### 2.3.1 Simulators and DSS

Before the use of computers, manufacturers of dimension wood parts relied on traditional methods and their own experience to estimate yield and to find the



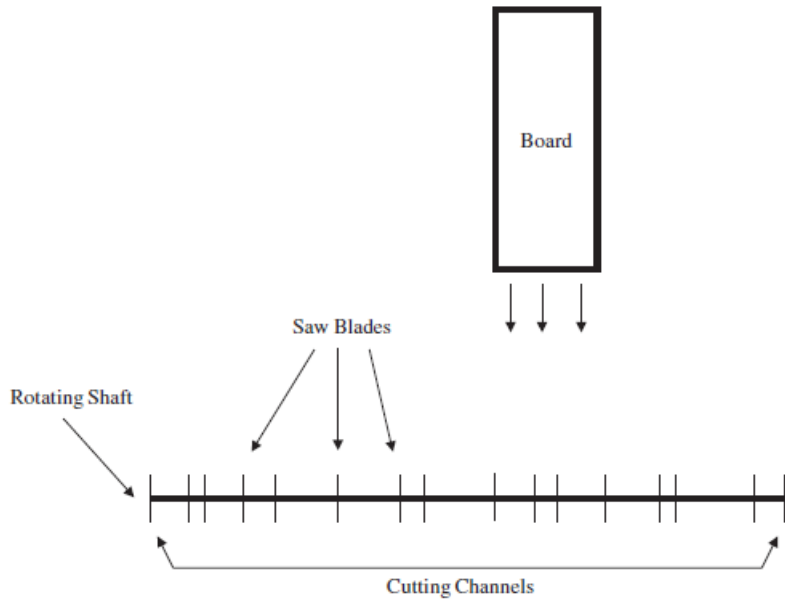


Figure 2.16: Fixed blade gang rip saw operation (Fathi and Aksakalli, 2004)

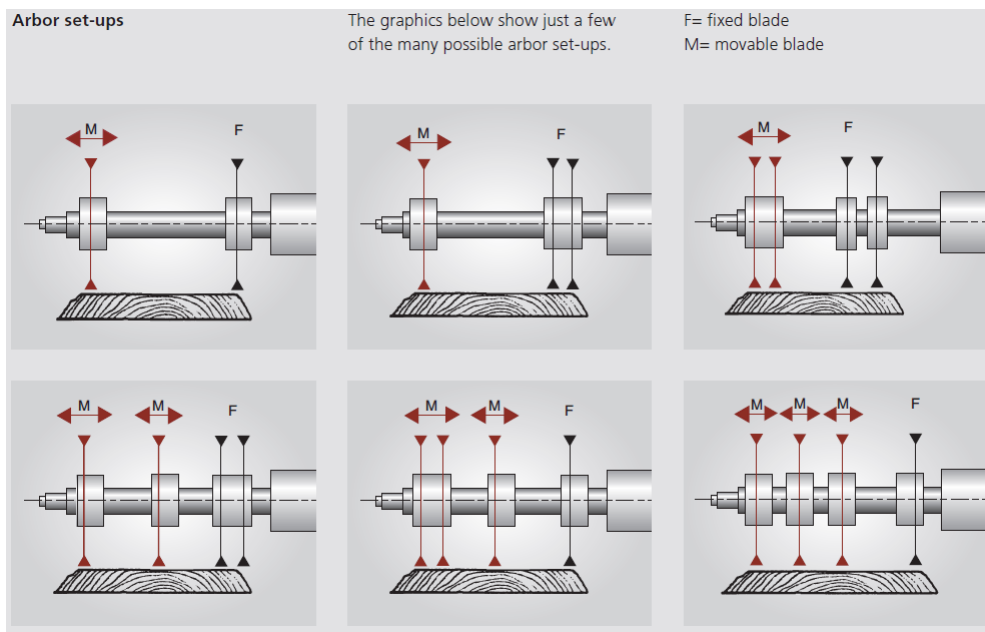


Figure 2.17: Gang rip saws-movable blade and arbour set-ups (Weining Group)

best cut-up solutions for particular boards for a specific order. Today, due to the lack of general applicable pure mathematical models to solve the problem of cutting lumber, computer simulation techniques are widely used to simulate rough-mill processing and lumber yield (Buehlmann and Thomas, 2001). As early as the 1960's (in North Carolina State University) work on com-

puter based solutions to the cutting-pattern problem has been developed. The most significant algorithms that followed from then were the Yield, MULRIP and RIPPYLD programs. However none of these early programs simulated the rough milling process exactly and had other limitations. The CORY program (published in 1984) was the first program that contained a part prioritisation strategy for the selection of parts to be cut (Buehlmann and Thomas, 2001).

Today, the two most significant programs that simulate the lumber cut-up are the Rip-X program by Harding and Steele and the ROugh Mill RIP-first simulator (ROMI-RIP) by Thomas Buehlmann and Thomas (2001). ROMI-RIP is the first rough mill simulation tool that reflects most accurately the operations of an actual mill. It allows the user to select from among six prioritization strategies, two cutting sequences, and seven arbour setups, arbours are the axis on which the saw blades that rip the boards into smaller strips are attached (Buehlmann and Thomas, 2001). In summary the Rough Mill Simulator (ROMI Version 3.0) is a computer software package that simulates industry's current practices for rip-first and chop-first lumber processing (Weiss and Thomas, 2005).

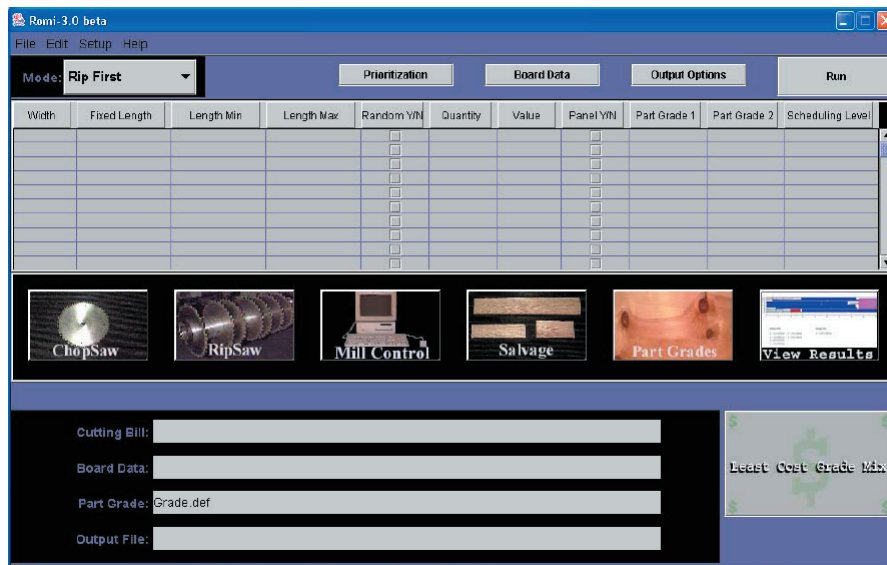
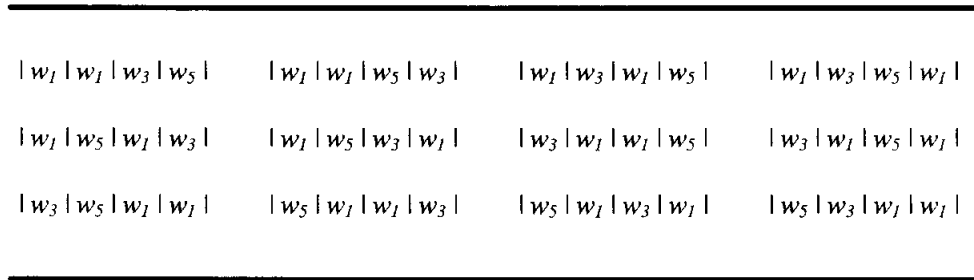


Figure 2.18: Main ROMI-3 interface window (Weiss and Thomas, 2005)

### 2.3.2 Optimisation

Fathi *et al.* (1996) used linear programming for gang-rip saw arbour design and scheduling. This is for fixed-blade arbour gang rip-saws (Figure 2.16). The optimal combinations of widths for each possible incoming board of a certain width was solved, which can be referred to as a cutting pattern. Fathi and

Aksakalli (2004) further used heuristics methods, based on the principles of local improvement, simulated annealing, and genetic algorithms for the gang-rip saw arbour design and scheduling. Fathi and Aksakalli's (2004) study on fixed saw blade saws found that: the local improvement methods produce fast and efficient heuristics; GA is also an efficient alternative to local improvement methods but the results obtained are not as good; Simulated Annealing finds high quality solutions, its computational requirements are somewhat excessive.

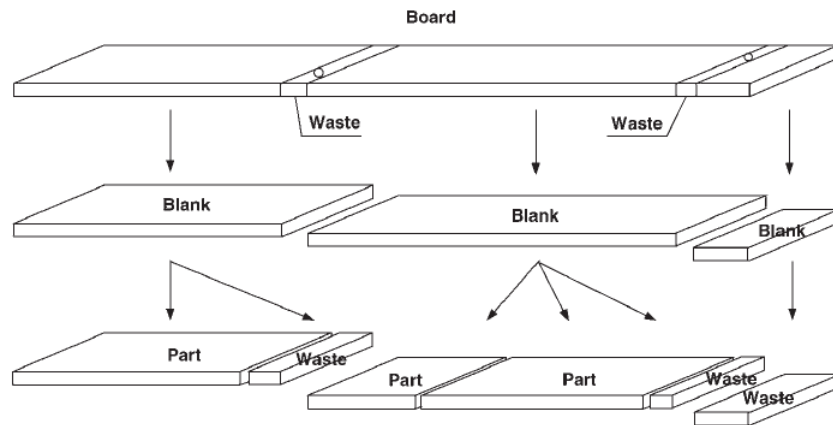


**Figure 2.19:** Patterns of channel widths, if embedded in an arbour, that would realise the optimal ripping solution,  $w_i$  is the  $i$ th distinct width (inches).

Although this study focusses on moveable blade gang-rip saws (Figure 2.17 as opposed to these fixed blades (Figure 2.15 and 2.16), Fathi and Aksakalli's (2004) study could help for choosing an appropriate optimisation algorithm for the moveable rip saw model.

For the chop saw operation, which is a similar cutting stock problem to the ripping operation, Maness *et al.* (2009) used linear and dynamic programming to determine the real-time optimal chopping of boards into dimension parts in a production optimisation system. Typically in a rough mill, the boards, or rough lumber, is firstly marked by a grader before being sent to the chop saw. The grader draws a line on the boards with a reflective crayon to identify defects and separate the waste from the blanks, which are clear length sections marked out of the boards as illustrated in Figure 2.20. The task of these optimising chop saws are then to detect marks on the incoming boards through the use of a scanning unit and chopping the blanks in a such a way as to obtain all the wood product parts listed in a cutting list, with the overall objective of minimising cost. To achieve all these task, chop saw systems typically use an optimiser and automated optimization algorithm. There are different approaches and algorithms to achieve this, i.e. increase the part yield or decrease the waste to fill a given order (Maness *et al.*, 2009). Most of these

optimising saws assign some sort priority value to each product to be sawn (ripped or chipped).



**Figure 2.20:** Optimising chop saws found in the industry (Maness *et al.*, 2009).

### 2.3.3 Priority Values

Currently industrial optimising chop saws use a part-priority approach to optimise for a production run. With this approach the operator manually sets priorities for certain difficult-to-cut parts (Maness *et al.*, 2009). Recently these priority approaches have been incorporated into modern ripping machines (Cothrell and Higgins, 2003).

There are two basic priority value set up modes: static and dynamic. The static priority mode uses only one value for each product during the ripping process, unless changed manually by the operator (Ye and Maness, 2006; Maness *et al.*, 2009). Static values are the conventional value setup system and are split up into two criteria: value and yield (Zuo, 2003, 59). The yield method maximises the part yield, based on the surface area without considering the demand requirements and does thus not employ any value system. Such yield-based methods tend to rip or chop products that best fit into the sawing decision, making it difficult to meet a specific market demanded production mix. Various static priority value setup algorithms have been proposed. The simplest rule is to apply the surface area of each part size as its priority value; e.g., length x width where a constant thickness means this just represents the volume. This is similar to the volume optimisation method modelled by Todoroki and Rönnqvist (2002), Equation 2.1.1. This is also how SIMSAW and subsequently SPPS models the ripping operation.

Value methods, on the other hand, maximise total part value from each strip, based on the part values assigned to each part size. This method is more commonly used in rough mills when the goal of the cutting is to satisfy demand (Zuo, 2003, 59). This is similar to the value optimisation method modelled by Todoroki and Rönqvist (2002), Equation 2.1.2. However Equation 2.1.2 assigns a value based on the product's price, this project attempts to include these values as a decision variable, as to meet market demand more accurately (priority values), in conjunction with the optimal sawing patterns (Figure 1.8).

Some programs prioritise parts based on a priority (sometimes called a dollar or decision) value. Wood product parts that are more desirable are given a considerably higher value. However, the values assigned to these parts are often arbitrary. It is often difficult to determine what the actual value of any single part flowing through the mill should be. This makes value strategies very dependent on the assumptions of the person performing the simulation. In addition, the part valuation strategy also lacks the ability to consider diminishing need as parts are sawn (Thomas, 1996). This is why dynamic value strategies have been developed. See for example Todoroki and Rönqvist's 2002 product optimisation strategy (Equation 2.1.3). However this study focusses on static priority values for a production run since they are the conventional value setup system (Zuo, 2003, 59).

## 2.4 Conclusion

As highlighted in the literature there have been many models developed for the wood sawing operations. However they are usually focussed on sawmilling or secondary manufacturing in isolation. Most of the mathematical models and simulation software developed for the sawmilling industry do include edging/ripping operations. However, for secondary manufacturers their ripping machines and optimisation strategies differ in that a priority value is manually assigned to each product as to somehow meet the market's products demanded.

The reason sawmills don't necessarily use the priority value optimisation approach is because they traditionally produce lumber products, that will stand in stock and be sold when demanded by the market (i.e. open market conditions). Thus value optimisation will be more based on the price the timber can be sold. This is how most of models developed by researchers under Section 2.1 (Figure 2.6) and DSS such as SPPS optimise. "If open market conditions are assumed when maximising volume or value yields, overproduction of some products and underproduction of others may result" (Todoroki and Rönqvist, 2002).

Secondary manufacturers do not have the luxury of producing (sometimes over-producing) the products that will have the highest selling price. They produce different sized dimensional stock that are crucial for assembly of final products (cabinet, furniture, doors, window, etc.). Thus their ripping machines have priority (dollar or decision) values, which force the machines to saw certain products more than others. This is done mainly to ensure that there is no excess of one product to another in relation to demand needed for subsequent assembly.

There are many models and research on modelling the sawing and ripping operations. However there was not any found specifically having the decision variables as set out in this project's objectives (Figure 1.8), namely the sawing patterns in conjunction with rip part priority values.

# Chapter 3

## Materials and Methods

This chapter describes the way the project's problem was approached, whilst Chapter 4 describes the formulation of the model in detail. The key components of any OR problem is described, namely the objective function(s), decision variables and constraints. The techniques chosen to solve the problem are expanded upon, namely log volume calculations, simulation data from SIMSAW, Mixed Integer Programming (MIP) and an appropriate metaheuristic, namely the PBIL algorithm. The chapter finally provides a summarised proposed model with important tools, software and techniques used. Also a strategy is developed to compare the model's results.

### 3.1 Modelling Components

*The scientific approach to decision making usually involves the use of one or more mathematical models. A mathematical model is a mathematical representation of an actual situation that may be used to make better decisions or simply to understand the actual situation(s) better (Winston and Goldberg, 2004).*

A prescriptive model includes the following components (Winston and Goldberg, 2004), which are key components in the optimisation model proposed by this project:

- objective function(s)
- decision variables
- constraints

#### 3.1.1 Objective Function(s)

In most models, there will be a function to be maximised or minimised. This function is called the model's objective function (Winston and Goldberg, 2004).

There can be many different ways to represent the objective function of a real world problem, which the model will try to minimise or maximise. There are also instances where there are more than one objective, this is referred to as multiple objective decision making (Winston and Goldberg, 2004). This thesis's model uses a single objective cost or profit function, since this was route taken by related studies, two of which are described below.

The objective of the SPPS model is to maximize the total profit, i.e. the sum of product value less log costs, fixed costs and the cost of buying in timber (Wessels *et al.*, 2006).

Maness *et al.* (2009) states the overall cost associated for a secondary manufacturer (specifically the chop saw operation) is mainly composed of the following factors:

1. cost of raw materials
2. cost of holding inventory for overproduced parts
3. replacement cost for under-produced parts
4. cost of disposing the waste
5. processing cost

The cost objective function to be minimised in this model includes the first three items above. The cost of disposing the waste and associated processing cost are outside the scope of this study, but should not be ignored and hopefully can be incorporated into future work.

### 3.1.2 Decision Variables

Decision variables in a model are the variables whose values which can be controlled and influence the performance of the system (Winston and Goldberg, 2004, Ch. 1).

Important decision variables for the sawing operations include the optimum production options, production line sawing patterns, to use in the sawmill (operational planning) (Wessels *et al.*, 2006). In a typical sawmill, logs are sorted in the yard to many diameter and/or length sorts (classes). The mill is then given the task of satisfying sales commitments. The question then becomes: what sawing pattern (or patterns) should be used for each log class, to produce the required production mix in the most efficient way? (Turner, 2010).

For the ripping operation the most important parameters which drive the optimisation decisions are the “decision” or priority values, which are the product



values specified for the different lumber products defined to the optimizer (an optimising rip or chop saw system) (Turner, 2010).

Thus the decision variables to be solved for this study are:

1. The sawing patterns per log class
2. The linked priority values of the different ripped lumber widths

### 3.1.3 Constraints

In most situations only certain values are possible for the decision variables in a system. These restrictions on the values on the decision variables are called constraints (Winston and Goldberg, 2004).

For the SPPS package there are several types of constraints namely (Wessels *et al.*, 2006):

- Log volume constraint: This limits the volume of logs available per log sorting class in the sawmill
- Timber demand constraint- this is the minimum and maximum volumes of each board product (dimension and grade) that are required to satisfy the market demand.
- Available processing time constraint: Each production option has a throughput ( $\text{m}^3/\text{hr}$ ) associated with it. Each production line has limited production time available and the sum of the time needed to complete the optimum production options must be less than or equal to the total processing time available.
- Integer shift volume constraint: Most of the machines used in the sawmill industry have very long set-up times and therefore production options (i.e. sawing patterns) can only be changed between shift changes. This constraint is necessary to forces the volume of logs to be sawn to be a multiple of the log input volume that can be processed during a single production shift with a specific production option.
- Drying constraint: This constraint ensures that the products manufactured stay within the limits of the (kiln) drying capacity of the sawmill.

The model proposed in this study focusses on two of these constraints, namely the log volume constraint and integer shift volume constraint. Namely that there are a certain amount of logs available (a parameter in the model) and a log diameter class can only be assigned to one sawing pattern.

For the ripping process typical constraints to be enforced would be making sure that number of many width products that may be ripped from a flitch do not exceed the usable width as illustrated in Figure 2.2 (Faaland and Briggs, 1984). Another constraint of the ripping process is the limit on how many widths may be ripped, this is dependent on the machine's design, namely the number blades and spacers and the length of the arbour.

### 3.1.4 Operations Research Techniques

There are a full range of OR methods that have been proposed to support planning problems in the forest products industry. Rönnqvist (2003) presented a series of typical planning problems found in the forest products industry, with comments about the time available for solving each of these problems. Operational planning problems usually need to be solved rapidly, within seconds or minutes, and thus heuristics, meta-heuristics and easy-to-solve network methods are generally used for such problems. Strategic planning problems can be solved over a longer period of time, sometimes taking many hours, thus MIP and stochastic programming methods are better for tactical and strategic planning problems. Many of the OR models are usually implemented as industrial DSS, integrated with application specific databases holding all the information needed for the models (D'Amours *et al.*, 2008).

What further limits the possible pool of OR techniques available is that when one is trying to model the sawing operations it is very difficult to develop equations that would predict the expected volume of each lumber item as a function of lumber values, machine penalties and other parameters. Firstly this is because of the continuous nature of lumber values, there are an infinite number of lumber-value and machine parameters. Second, the non-linearity described above makes it impossible to write an equation that could realistically be solved using standard mathematical tools. When one is trying to predict the outcome of a system with the non-linear nature of a sawmill, simulation is often the best alternative. The simulator needs to be able to predict what products will be recovered when using the same set of parameters as the mill (Turner and Rapoport, 2010)

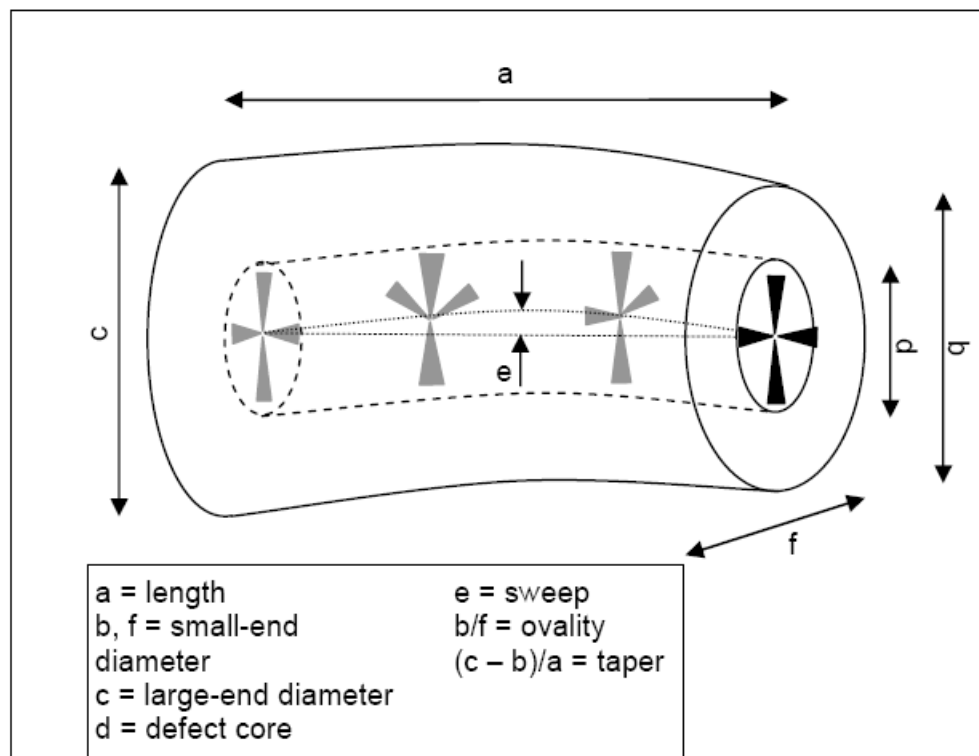
The ripping process is easier to model since the shape of flitch is much less complex than that of a log. It can be modelled as a packing problem, and solved using integer programming (described further in Section 3.4) instead of using simulation techniques.

Thus to summarise, instead of trying to solve the problem through analytical techniques, the strategy was to use available simulation methods (namely SIMSAW) for the primary sawing operation and linear/integer programming techniques for the ripping operation. The decision variables for both opera-

tions however need to interact with each other producing dependent output variables. With this in mind one of the best ways to find the optimal decision variables for the two operations is through the use of an appropriate metaheuristic. The metaheuristic is thus the link between the simulation of the sawing operation (SIMSAW data) and the ripping operation (MIP).

## 3.2 Logs Shape Characteristics

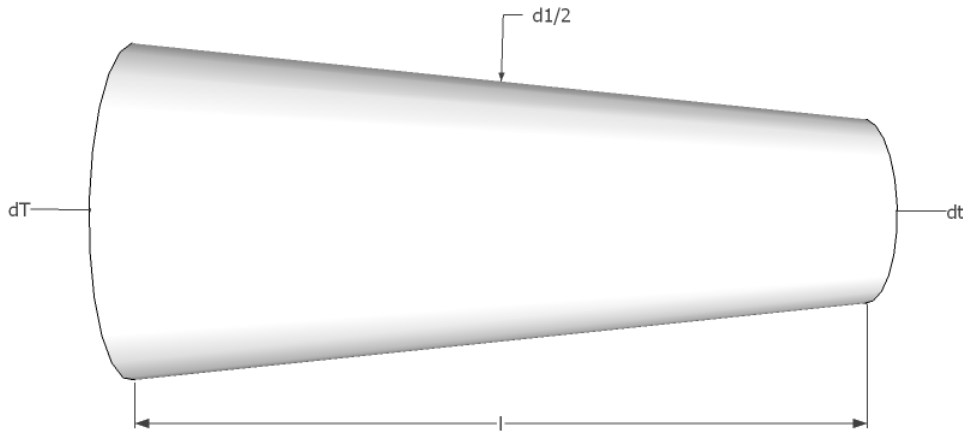
One of the most important aspects influencing the volume recovery of wood products from a sawmill is the irregular shape characteristics of its raw material, its logs. Logs are produced from harvested trees, cross-cut by the foresters. They are transported from the plantations to a sawmill's log deck where they are typically sorted according to log classes (Du Plessis, 2010, 16). These classes are mostly dependent on the log's diameter but characteristics such as sweep, length and absence of certain defects can also be incorporated (Du Plessis, 2010, 16). These shape and knot (which stem from the branches of a tree) characteristics are illustrated in Figure 3.1).



**Figure 3.1:** Typical log shape and knot characteristics (Du Plessis, 2010). Sweep and taper are of concern in this study.

### 3.2.1 Calculating the Log Volume

In the SA forestry industry a Log's Volume (LV) is calculated to represent that of a cylinder as shown in Figure 3.2.



**Figure 3.2:** Log volume calculation (Huber's method)

The cross-sectional area of a log can be measured at the mid-length of the log ( $b_{1/2}$ ) or alternatively as the average of the small/thin end ( $b_t$ ) and the large/thick end ( $b_T$ ). Another method is to measure all three positions and determine a weighted average, as  $(b_t + 4b_{1/2} + b_T)/6$ . The first method is most commonly used and is known as Huber's method (Bredenkamp *et al.*, 2012, 239).

$$\text{Log Volume (Huber)} = \frac{d_{1/2}^2 \pi l}{4} \quad (3.2.1)$$

However Huber's formula above cannot be exactly determined when logs are stacked on top of each other as is usually the case in industry, since it is difficult to measure the mid-length diameter ( $d_{1/2}$ ). Because of this a compromise is made where the under-bark diameter is measured at the thin end ( $d_t$ ). Log diameters are usually measured in centimetres at the thin-end and rounded down to the closest uneven number. Assuming constant, uniform taper, the diameter at mid-length can be estimated and thus volume calculation becomes:

$$\text{Log Volume (Huber)} = \frac{(d_t + \frac{l}{2 \times \text{taper}})^2 \pi l}{4} \quad (3.2.2)$$

Nationally the taper for softwoods (pines) is normalised to 1 cm/m (hardwoods are at 0.8 cm/m), the volume ( $m^3$ ) of logs for softwood then becomes:

$$\text{Log Volume (Huber)} = \frac{(d_t + \frac{l}{2})^2 \pi l}{40000} \quad (3.2.3)$$

### 3.2.2 Logs Generated

Logs were virtually generated in SIMSAW, where a user can generate a population of logs, following a certain distribution for appropriate log characteristics, namely the thin end diameter, length, taper and sweep.

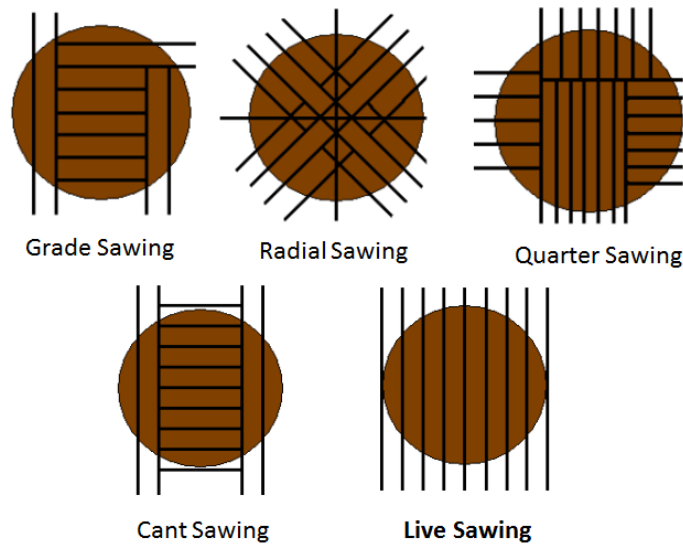
The thin end diameters were generated either normally or uniformly within the log classes. The number of logs to generate for a production time period was according to data collected from the project's case company (following Section 5). Logs were generated having a fixed length, since the case company only receives fixed 3 meter length logs. However the model does cater for varying length classes (explained later, Equation 4.1.2).

The taper of logs was distributed normally, 95% within 8-11 mm/m limits. The sweep of logs was distributed normally, 95% within 0-15 mm/m limits. These values were chosen since these are the default values in SIMSAW and South Africa's national taper is normalised to 1 cm/m for softwoods (Bredenkamp *et al.*, 2012, 240).

## 3.3 Sawing Operation: Simulation Data

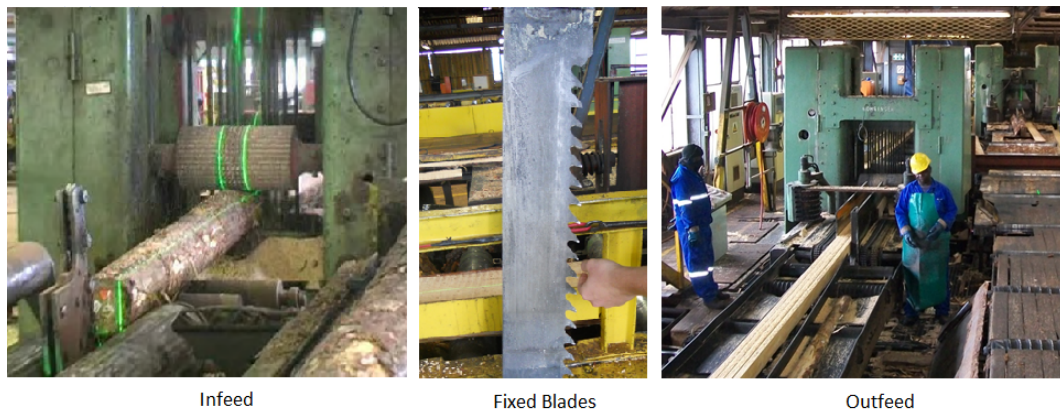
Live sawing patterns saw logs through-and-through, meaning that all saw cuts are parallel (Figure 3.3). They have the advantage that they are fast and minimize log turning; however, every flitch sawn from the log requires more edging. Other sawing methods are quarter sawing, grade sawing and cant sawing. Each of these sawing patterns are depicted in Figure 3.3. The quarter sawing pattern is a specialized sawing pattern that finds application with hardwoods, since it shows off the grain of decorative timbers. The log is primarily sawn into quarters and each quarter is further broken down to produce boards. However it is a slow procedure. Grade sawing, whilst also characterized by relatively slow throughputs due to log turning, involves sawing around the log, with the aim of maximizing the grade yield of high-quality boards. It is specifically suited to the sawing of pruned logs. The fourth cutting pattern, cant sawing, is applicable to both pruned and un-pruned logs and suits high-production mills, especially those with a gang-saw downstream (Todoroki and Rönnqvist, 2002).

For the primary breakdown of logs, most pine sawmills in South Africa use a framesaw as the preferred sawing machine (Figure 3.4). These framesaws make use of either the cant or the live sawing method. Cant sawing is the most common sawing pattern used in softwood (pine) sawmills in South Africa (Du



**Figure 3.3:** Log sawing methods (Van Zyl, 2011).

Plessis, 2010). The model can cater for both cant and live sawing methods, but the case company uses live sawing methods and was thus the method chosen for this project.



**Figure 3.4:** Framesaw

A framesaw consists of a number of fixed blades that move up and down in a frame at a high speed (Figure 3.4). The log is fed through the framesaw, which then produces various boards, referred to as flitches (Figure 1.1). One major issue of a framesaw is that the dimensions of the boards sawn cannot be changed during production. This is because to change these dimensions, a whole new blade setup on the framesaw has to be done (Van Zyl, 2011, 24). A saw's spacer setup is referred to as a specific sawing pattern which are the

decision variables for this operation in this project.

It is important to note that this study focusses on live sawing, and subsequent ripping. There is not much research on modelling or optimising these specific operations and some of their decisions. Live sawing is the preferred method researched in New Zealand (Todoroki and Rönnqvist, 1999).

### 3.4 Ripping Operation: Integer Programming

Secondary manufacturing (referred to as rough milling in America) includes ripping and chopping operations. These operations have changed from using extensive manual decision sawing systems to machines capable of automated decision making in an effort to increase yield, amongst other objectives (Zuo, 2003). The ripping machines are commonly referred to as gang saws (Buehlmann, 1998). There are many types of gang saws but the ones having multiple moving blades saws produce the greatest recovery, assuming they are equipped with an accurate scanning systems (Zuo, 2003).

The operating principle of scanning and optimisation systems in the industry is generally the same, regardless of the machine or machine centres. The solution decision process usually follow the following principles (Turner and Rapoport, 2010):

1. Scan the incoming flitch/board piece optically to determine the geometric, and in some cases biological characteristics.
2. Consider all the possible ripping solutions for the flitch/board piece (this usually takes place in several hundred milliseconds).
3. Select the solution with maximum 'value' for the given flitch/board piece.

Instead of determining all the possible combinations of products possible to be ripped, and then selecting the one with the maximum value yield, integer programming was used to find this maximum yield combination.

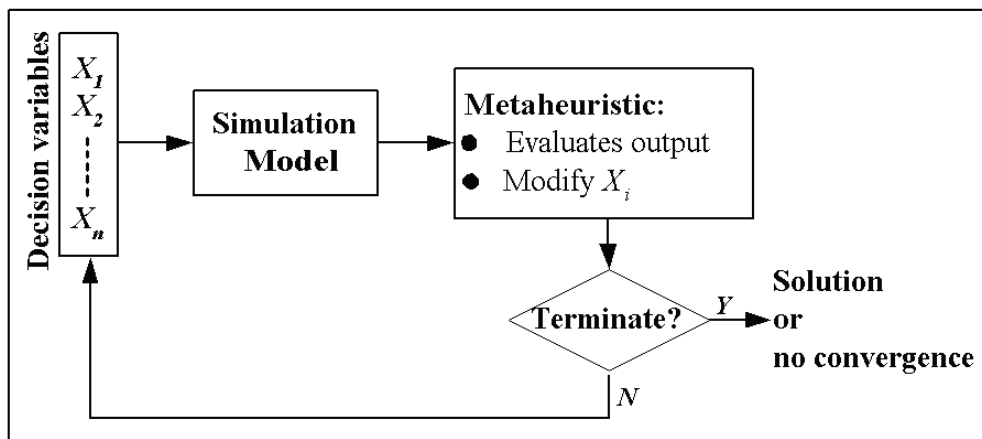
Most of the optimising chop saws use priority rule calculations to determine which products to cut, which have recently been incorporated into modern ripping machines as well (Cothrell and Higgins, 2003). There are two basic priority value set up modes: static and dynamic. The static priority mode uses only one value for each product during the ripping process, unless changed manually by the operator (Ye and Maness, 2006; Maness *et al.*, 2009). Static values are the conventional value setup system and are split up into two criteria: value and yield. The yield method maximizes the part yield, based on the surface area without considering the demand requirements and does thus not

employ any value system. Such yield-based methods tend to rip products that best fit into the cutting decision, making it difficult to meet specific market demanded products. This is, however, how SIMSAW and subsequently SPSS models ripping decisions are made. Value methods, on the other hand, maximize total part value from each strip, based on the part values assigned to each part size. This method is more commonly used in rough mills when the goal of the cutting is to satisfy demand (Zuo, 2003).

The optimising machines have different strategies for how they should rip based on priority of parts. The case company's ripping machine's optimisation strategy is shown in Appendix B. Other types of ripping machine optimisation strategies could be of further interest but fall outside the scope of this study.

### 3.5 Metaheuristics for Optimising the Decision Variables

Metaheuristics are usually used when a model cannot be clearly stated in mathematical terms, and thus OR techniques such as linear, non-linear, dynamic techniques cannot be used. Metaheuristics are especially useful in conjunction with simulation, where the parameters of the model can be varied and through evaluating the outputs for each change, try and find better solutions (Figure 3.5).



**Figure 3.5:** Principle of simulation and optimisation with metaheuristics (Bekker and Olivier, 2008).

As highlighted in the literature there have been many heuristics and metaheuristics applied to the wood manufacturing industry. For the ripping operation Fathi and Aksakalli (2004) used heuristic methods for gang-rip saw arbor



design and scheduling. These heuristic methods were based on the principles of local improvement, simulated annealing, and genetic algorithms. Although this study focusses on moveable blade rip saws, Fathi and Aksakalli's (2004) study on fixed saw blade saws concluded that: the local improvement methods produce fast and efficient heuristics; GA is also an efficient alternative to local improvement methods but the results obtained are not as good; simulated annealing finds high quality solutions, but its computational requirements are somewhat excessive.

Figure 3.6 shows the results of different metaheuristics and other algorithms for the optimisation of cant sawing log positioning in a sawmill (Wessels *et al.*, 2011).

Description	Parameters	Abbreviation	Average maximum volume recovery obtained (%)			
			After 90 simulations <sup>a</sup>	After 350 simulations <sup>a</sup>		
Population-based incremental learning (PBIL)	Learning rate					
		Mutation probability				
		Mutation shift				
	0.1	0.03	0.05	PBIL_a	51.20	51.78
	0.1	0.1	0.1	PBIL_b	51.20	51.85
	0.2	0.03	0.05	PBIL_c	51.33	51.57
	0.2	0.1	0.1	PBIL_d	51.31	51.71
Particle swarm optimisation (PSO)	Velocity = 0.5			PSO_a	51.06	51.06
	Velocity = 1.0			PSO_b	51.43	51.45
	Velocity = 1.5			PSO_c	51.61	51.61
Simulated annealing (SA)	Starting temperature = 5 000; linear cooling			SA_a	51.15	51.80
	Starting temperature = 50 000; 10% cooling			SA_b	51.13	51.81
	Starting temperature = 5 000; 10% cooling			SA_c	51.24	51.82
Tentacle	Flexible tentacles; constant growth			Ten_a	51.71	51.89
	Linear tentacles; constant growth			Ten_b	51.64	51.80
Random selection of positions within search space				Ran	51.24	51.79
Exhaustive search	Range 1			Rng1	51.74	
	Range 2			Rng2		52.07
Global optimum in search space (4 056 simulations)				Glob max	52.38	
Recovery at centred, 'horns-up' position (one simulation)				Normal	49.87	

<sup>a</sup> Or less if convergence was reached before stated number of simulations

**Figure 3.6:** Comparison of metaheuristics for the cant sawing log position optimisation by Wessels *et al.* (2011).

Although this project's problem does not involve the log positioning problem it is related in that it involves simulating a sawmill's sawing process. The results from the log positioning problem by Wessels *et al.* (2011) could possibly help in determining the appropriate metaheuristic. Besides the developed Tentacle algorithm, the standard PBIL faired well and because it is based upon the GA which faired well in Fathi and Aksakalli (2004)'s fixed blade gang rip study, it was chosen as the appropriate metaheuristic. Other metaheuristics could be of further interest but are outside the scope of this study.

## 3.6 PBIL Algorithm

The Population Based Incremental Learning (PBIL) algorithm was developed by Shameet Baluja (1996) and was based upon the genetic algorithm. It uses an updated Probability Vector (PV) as a basis for generating a population of solutions. The PV is updated by it learning or striving toward the fittest solution in the population.

	Population 1				Population 2				Population 3			
SV=1	O	0	1	1	1	0	0	0	1	0	1	0
2	1	1	0	0	1	1	0	0	0	1	0	1
3	1	1	1	0	1	1	0	0	1	0	1	0
4	O	0	0	1	1	1	0	1	0	1	0	1
	2/4	2/4	2/4	2/4	4/4	3/4	0/4	1/4	2/4	2/4	2/4	2/4
	Probability Vector 1				Probability Vector 2				Probability Vector 3			
	0.5	0.5	0.5	0.5	1	0.75	0.0	0.25	0.5	0.5	0.5	0.5

**Figure 3.7:** Populations and Probability Vectors (PV) in PBIL. Baluja (1994). SV = Solution Vector, where the number of vectors is equal to the population size.

The PBIL algorithm operates very similarly to the GA algorithm in terms of how solutions are structured. The decision variables of the problem are encoded as binary values to form a complete binary string, which is called a SV (Figure 3.7). The SV's string length is mostly determined by the nature of the decision variables it represents. For example for decision variables having a decimal range of 0-100, a seven binary digit string (0-128) is sufficient. The values in the solution vector's string are randomly assigned binary values (0 or 1). The probability that a bit in the string is assigned a 1 depends on the population's PV (Figure 3.7). For example suppose that the PV for the second bit in population 2 in Figure 3.7 is 0.75, then the probability that a SV's second bit will be assigned a 1 is 75%. A finite set of SVs forms a population of possible solutions which are evaluated individually to determine the fittest/best SVs. The size of the population is an important parameter that the analyst must select (Bekker and Olivier, 2008).

For this project's model a SV containing a string of binary digits will represent one or more decision variables to be run through the model.

### 3.6.1 Initialisation

The initial population generated is distributed according to a certain value, usually 50% (Bekker and Olivier, 2008).

### 3.6.2 Learning

Once an initial population of SV has been generated, the fitness of each solution vector is evaluated. For this project it is the cost associated with implementing the sawing patterns and priority values for manufacturers. The PBIL algorithm generates the next population based upon the fittest individual found in the previous population. It does this by making it PV (a representation of the population's solution vector) strive towards the fittest SV as shown in Equation 3.6.1.

$$PV_{gb} \leftarrow P_{gb}(1 - LR) + BSV_{gb}(LR) \quad (3.6.1)$$

where:

$PV_{gb}$  = Probability Vector's value of the b-th cell in the g-th generation

LR = Learning rate

$BSV_{gb}$  = Best Solution Vector's value of the b-th digit in the solution vector (binary-0 or 1) yielding the maximum/minimum evaluation (the current "best"), in the g-th generation

### 3.6.3 Mutation

While the PBIL algorithm is searching for more optimum solutions, the role of mutation becomes more important at later stages of the process. This is because as the algorithm is striving to converge on upon better solutions, diversity is inversely lost in the population. Many optimisation algorithms use a mutation operator to ensure that an algorithm does not get stuck in a local optimum solution. However in the PBIL algorithm mutation does not play such a crucial role as it does in GAs, but it does aid the PBIL algorithm in finding better solutions (Bekker, 2012). On each of the cells in the probability vector, mutation is done as follows (Bekker, 2012):

$$PV_{gb} \leftarrow \begin{cases} PV_{gb}(1 - MS) + (Rand_1)(MS) & Rand_2 \leq MP \\ PV_{gb} & \text{Otherwise} \end{cases} \quad (3.6.2)$$

where:

MS = Mutation Shift

MP = Mutation Probability

$Rand_1$  = A random number generated from a uniform distribution (0-1)

$Rand_2$  = An independent random number generated from a uniform distribution (0-1)

### 3.6.4 Termination

The PBIL can be terminated in two ways. The first method stops the algorithm when it starts to converge, meaning that the values in the PV converge to either

0 or 1 values (Bekker and Olivier, 2008). Usually threshold values are set so that when the values in the PV are either less than a set low threshold or higher than a set high threshold the algorithm is terminated. These threshold values are chosen beforehand and are typically 0.05 (low threshold) and 0.095 (high threshold) (Du Plessis, 2010; Bekker and Olivier, 2008). The other method is used when the algorithm has been running for too long. After a set number of generations, specified by the user (e.g. 300), the algorithm is terminated. With this method there is of course a risk that the algorithm can be stopped prematurely while it is still converging to better solutions (Bekker and Olivier, 2008).

### 3.6.5 PBIL Steps

Summarising the above descriptions, the PBIL algorithm in pseudo code is shown in Figure 3.8.

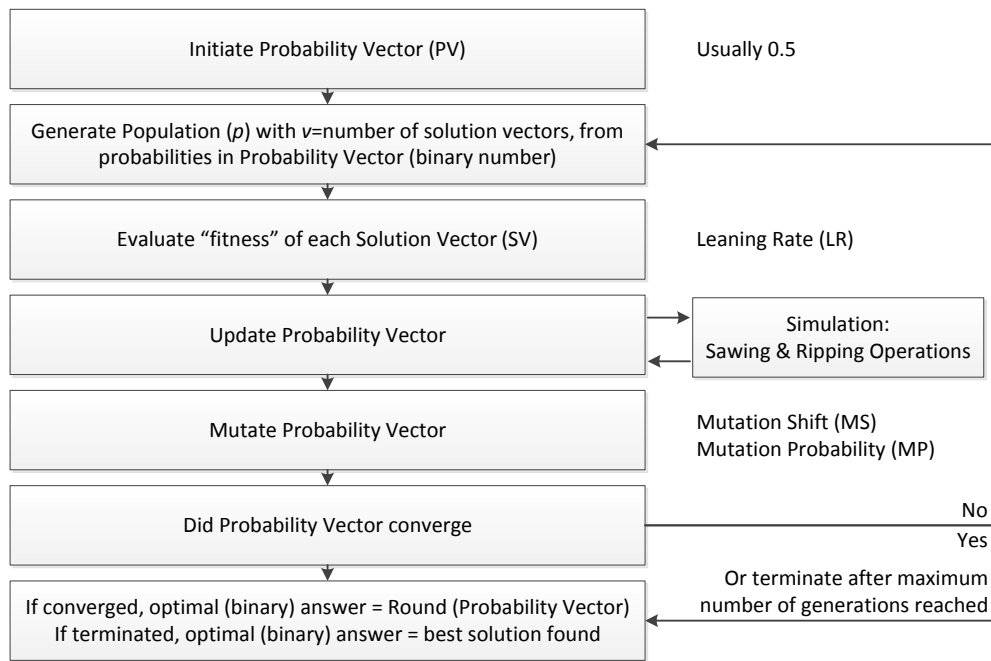
```

//Start Algorithm
1: Assign LR, Pr(Mutation), MS,
   MaxGenerations, PopulationSize
2: Initialise Probability Vector P
3: k←1
4: Converged←False
5: While Not Converged AND k ≤ MaxGenerations
6:   For j = 1 To PopulationSize
7:     Generate solution vectors in population
8:     Evaluate the solution vectors
9:   Next j
10:  Find the best solution vector
   //Update probability vector P:
11:  For i = 1 To Number of Elements of P
12:     $P^k(i) \leftarrow P^k(i) \times (1 - LR) + SV_B^k(i) \times LR$ 
13:  Next i
   //Do mutation
14:  For j = 1 To Number of Elements of P
15:    If (Random(0,1) < Pr(Mutation)) Then
16:       $P^k(j) \leftarrow P^k(j) \times (1 - MS) + \text{Random}(0 \text{ or } 1) \times MS$ 
17:    End If
18:  Next j
19:  k←k + 1
20:  Converged ← Call Function CheckConvergence
21: End While
22: Return Result
//End Algorithm

```

**Figure 3.8:** PBIL algorithm in pseudo code (Bekker, 2012)

How the PBIL algorithm was applied to the model is shown in Figure 3.9.



**Figure 3.9:** Diagram of PBIL steps executed within the model, built upon notes courtesy of Bekker (2012).

### 3.6.6 PBIL Parameters

There are four parameters that can be adjusted in the PBIL algorithm and the typical values suggested by some authors are illustrated in Table 3.1.

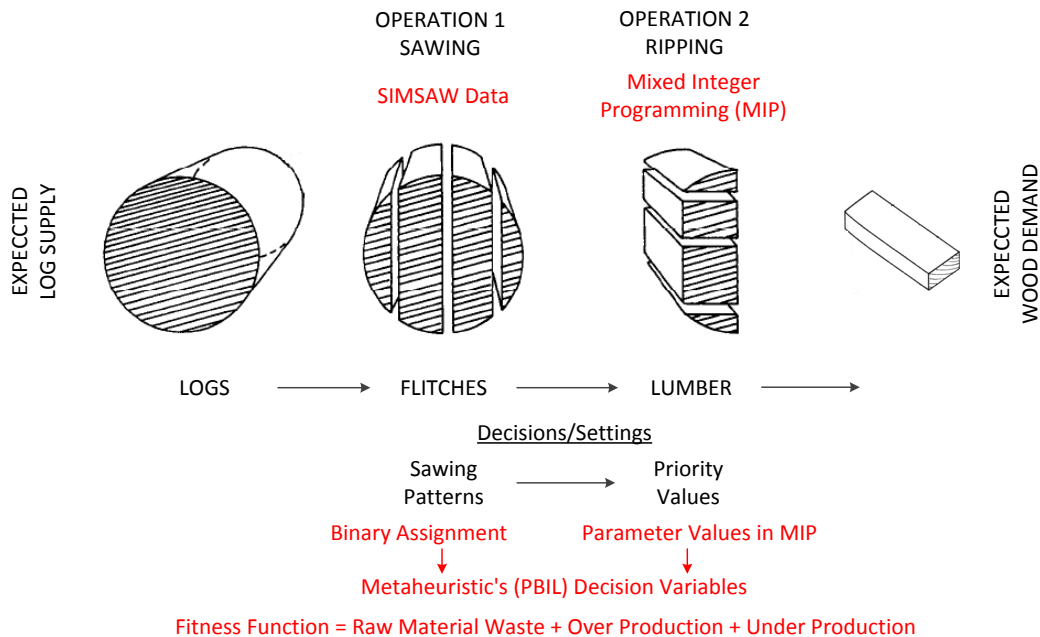
**Table 3.1:** PBIL Parameters suggested and tested in different studies: 1-(Baluja, 1996); 2-(Van Zyl, 2011); 3-(Wessels *et al.*, 2011); 4- (Bekker, 2012).

Parameter	1	2	3	4
Population Size	100	10-20	10	30-100
Learning Rate	0.1	0.1-0.4	0.1 or 0.2	
Mutation Shift	0.05	Not applied	0.05 or 0.1	0.05
Mutation Probability	0.02	Not applied	0.03 or 0.1	0.02

## 3.7 Proposed Model

The proposed model to solve this project’s optimisation problem is summarised in Figure 3.10. SIMSAW data was used as parameter values in the model. The output of this operation becomes the input for the ripping operation, which models the operation as a MIP. The decision variables for these two operations,

the sawing patterns and part priority values, are encoded as binary strings for the PBIL which strives to find the optimal solution (near optimal solution). The fitness function with which the PBIL algorithm strives to minimise is the cost associated with wasting raw material (recovery), over production of certain parts, and under production of certain parts. Various constraints are also of importance and are explained fully in Section 4.



**Figure 3.10:** Tools and techniques used in this project.

### 3.8 Software Used

The model, including the decision variables of the metaheuristic was run in the Advanced Interactive Multidimensional Modelling System (AIMMS) 3.13 modelling environment. AIMMS is an *advanced integrated multidimensional modelling software for building decision support and optimization applications* ([www.aimms.com](http://www.aimms.com)). However the parameter representing the output of flitches after the sawing operation was determined via the SIMSAW 6 simulation package.

### 3.9 Comparison Strategy

The model was first developed after consulting the literature and industry (case company) on its objectives and its operating procedures. Also the neces-

sary parameters were estimated after data collection from the case company.

The model was tested under different supply and demand scenarios and compared to SPPS's answers. The model is formulated in more details in the following Chapter.

### 3.10 Conclusion

This chapter provided the methods used to solve the project's problem stated. The artificial logs generated in SIMSAW will have their volume calculated according to Huber's formula. The subsequent sawing operation's output will be formulated as a proportion, this proportion will be determined via simulation data. The subsequent ripping operation will be formulated as a MIP problem. The PBIL algorithm was chosen as an appropriate metaheuristic to link and optimise the decision variables (machine settings) of the sawing and ripping operation. The cost objective to be minimised consists of the raw material waste cost, over production cost and under production cost. The software used to develop the model include SIMSAW 6 and AIMMS 3.13. The proposed model and its results was compared to the SPPS package's similar results under the same conditions.

# Chapter 4

## Model Proposed

This chapter describes how an appropriate model was formulated that links the two decision variables of wood processing operations, namely the primary sawing of logs and subsequent ripping of flitches/boards. Simulation data, from SIMSAW, was used to represent the sawing operation. Mixed Integer Programming (MIP) was used to model and solve the ripping operation optimisation problem. An appropriate metaheuristic, namely the PBIL algorithm was used to link the models and optimise the decision variables, namely the sawing patterns and the rip part priority values. The objective function or the fitness function for the PBIL algorithm was defined as a single cost objective, composed of the three important aspects, namely raw material waste cost, under production cost and over production cost.

### 4.1 Formulating the Model

This section describes how the two operations were formulated with relevant equations.

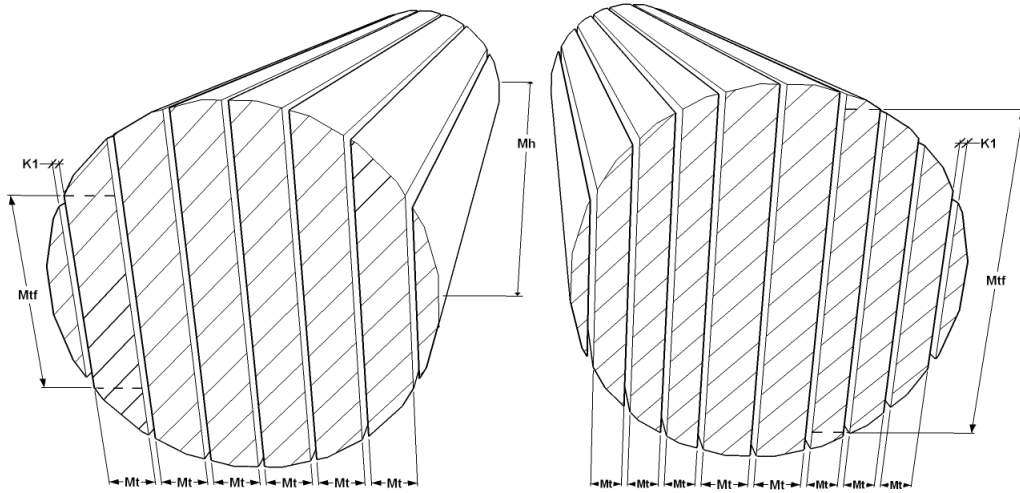
#### 4.1.1 Operation 1: Sawing the Logs

Logs are trees that have been cross-cut into appropriate lengths, usually specified by the sawmill. The inputs for this operation are the number of logs supplied by foresters falling within specified log diameter classes.

Logs are usually classified in such a way that specific sawing patterns can be assigned to them, yielding better results for that specific class. For example a log class having 35-36.9 cm diameter logs will be sawn (or broken down) differently than a log class containing 19-20.9 cm diameter logs (Du Plessis, 2010, 17).



Thus an appropriate decision variable for this operations is the sawing patterns assigned to the specified log classes, similar to the SPSS package. A parameter of this operation was the possible sawing patterns pre-determined by management. Usually sawing patterns yielding the highest recovery rates and an appropriate mixture of products demanded are selected based on experience. The decision variables represent which sawing pattern will be assigned to which of the set log diameter classes, Equation 4.1.1.



**Figure 4.1:** Two possible sawing patterns for a certain log class at the sawing operation. Each pattern will produce different dimensioned products and volume outputs. The labels are introduced in the equations below.

The outputs from this operation are a number of unedged boards, called fitches (Equation 4.1.8). For this project the fitches were classified by their maximum edged board widths (Equation 4.1.6), meaning its edges were straight and there is no bark or wane. This value is dependent on the logs taper and sweep as shown in Figure 4.2. Instead of determining each log's sawn output mathematically and, so to speak, re-invent the wheel, existing simulation software (SIMSAW 6) was used as an estimation of the fitches' maximum edged board width. With SIMSAW the user can generate logs with varying taper and sweep (Figure 3.1), which will affect the fitches' maximum edged board width as shown in Figure 4.2. By specifying possible sawing patterns it then simulates the board output for each log class and sawing pattern. The logs were simulated through the set sawing patterns and the number of fitches with a set maximum edged board width (Equation 4.1.7), were used as a data parameter in the optimisation model.

Let:

$$Y_{dc} = \begin{cases} 1 & \text{Log diameter class } d \text{ is assigned to sawing pattern } c \\ 0 & \text{Otherwise} \end{cases} \quad (4.1.1)$$

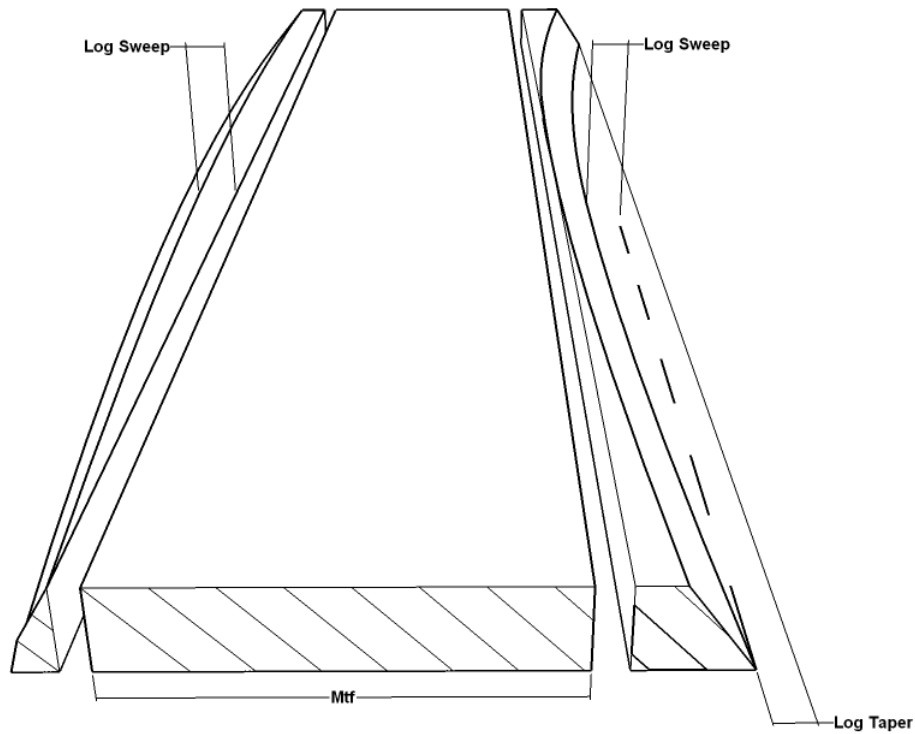
$$M_h = \text{Length of log within set length } h \quad (4.1.2)$$

$$M_d = \text{Diameter of log within set log diameter class } d \quad (4.1.3)$$

$$S_{dh} = \text{Number of logs supplied having diameters that are within set log class } d \text{ and having set log length } h \quad (4.1.4)$$

$$M_t = \text{Thickness of flitch within set thickness } t \quad (4.1.5)$$

$$M_{tf} = \text{Flitch's maximum edged board width within set thickness } t \text{ and flitch board width } f \quad (4.1.6)$$



**Figure 4.2:** Flitch's maximum edged board width ( $M_{tf}$ ) dependent on the log's sweep and taper. This was determined via SIMSAW.

$$\begin{aligned} \text{SIM}_{dctf} = & \text{Simulated proportion of flitches, having set thickness } t \\ & \text{and flitch board width } f \text{ sawn if} \\ & \text{log class } d \text{ is assigned to sawing pattern } c \end{aligned} \quad (4.1.7)$$

The proportion of flitches (Equation 4.1.7) was determined through simulating a 1 000 logs, having a normal taper distribution 95 % within 8-11 mm/m limits and a normal sweep distribution 95 % within 0-15 mm/m limit within each log class  $d$  through each possible sawing pattern  $c$ . This proportion of flitches having a set thickness  $t$ , board width  $f$  produced for each log class  $d$  and sawing pattern  $c$  was determined from SIMSAW data and used as a parameter value (data) in the model. This is further explained in Section 4.3

$$S_{tfh} = \sum_d \sum_c Y_{dc} \text{SIM}_{dctf} S_{dh} \quad (4.1.8)$$

Equation 4.1.8 then calculates the supply ( $S_{tfh}$ ) or number of flitches having thickness  $t$ , board width  $f$  and length  $h$  by multiplying the proportion of flitches expected ( $\text{SIM}_{dctf}$ ) times the number of logs supplied ( $S_{dh}$ ) and the binary variable determining if the logs are assigned to the set sawing pattern  $c$  ( $Y_{dc}$ ). This becomes the input material for the next operation (ripping).

Constraint 4.1.9 is required to ensure that all logs falling within the set log class  $d$  are assigned to only one sawing pattern  $c$ . This constraint was forced into the model by manipulating the PBIL's probability vector. This is explained further in Section 4.2.2

$$\sum_c Y_{dc} = 1 \quad \forall d \quad (4.1.9)$$

## 4.1.2 Operation 2: Optimally Ripping the Flitches

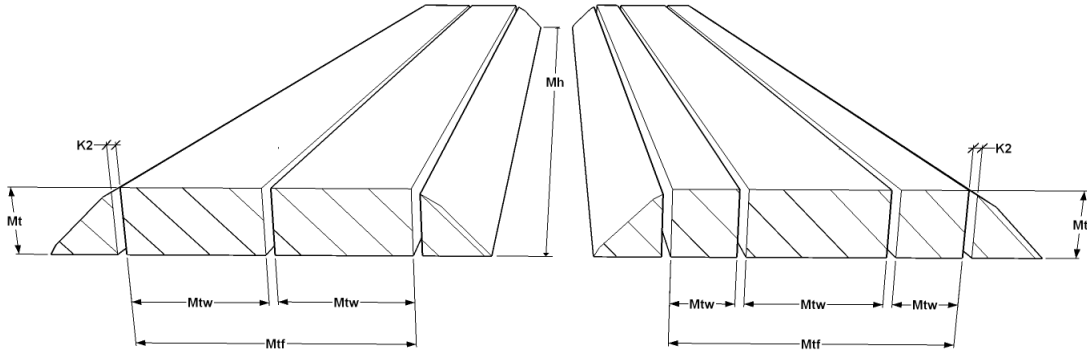
The second operation was formulated as a Mixed Integer Programming (MIP) problem. This formulation is similar to Todoroki and Rönnqvist's (1999) method which uses dynamic programming (Todoroki and Ronnqvist, 1997) to solve the problem.

Let:

$$M_{tw} = \text{Width of lumber within set thickness } t \text{ and width } w \quad (4.1.10)$$

$$P_{tw} = \text{Priority value of lumber ripped with thickness } t \\ \text{and width } w \quad (4.1.11)$$

The priority values are usually percentage values so they typically have a range between 0-100.



**Figure 4.3:** Two ways to rip a flitch at operation 2. Each with different volume and product outputs. Priority values (settings on the machine) usually determine which products to rip.

$$X_{tfw} = \text{Number of components having set width } w \text{ ripped from} \quad (4.1.12)$$

$$\text{flitch having set thickness } t \text{ and flitch board width } f$$

$$X_{tfw} \text{ is integer} \quad (4.1.13)$$

$$K_2 = \text{Kerf, thickness of saw blades, at operation 2} \quad (4.1.14)$$

As mentioned in the literature there are many different machines with many different saw blade and arbor setups and further combinations of algorithms used for ripping optimisation strategies. For this project the type of machine and its algorithm was specifically modelled for the case company. The machine's optimisation strategy and settings for the case company is shown in Appendix B. Thus the objective function for operation 2 was modelled as:

$$\text{Max } V = \sum_t \sum_f \sum_w M_{tw} P_{tw} X_{tfw} (M_t M_h) \quad (4.1.15)$$

For the project's ripping formulation the thickness is predetermined by the sawing operation and is thus constant in the equation above ( $M_t$ ). Also the length of the flitch stays the same during the ripping operations in this project's model formulation ( $M_h$ ). Thus these two constants do not necessarily need to be included in the maximisation problem above. However they are added in Equation 4.1.15 to show that  $V$  in fact represents the volume ( $M_{tw} M_t M_h$ ) of the product ripped ( $X_{tfw}$ ) times the products priority value ( $P_{tw}$ ). This is similar to Todoroki and Rönnqvist's (2002) value optimisation method for secondary log breakdown (Equation 2.1.2) where the timber price in Equation 2.1.2 is similar to the priority value ( $P_{tw}$ ). For this project, priority values are however not set by the timber price but can be changed (decision variable) by

the operator as to meet market demand more accurately.

Subject to the following two constraints:

$$\sum_w X_{tfw} \leq 3 \quad \forall(t, f) \quad (4.1.16)$$

$$\sum_w X_{tfw} P_{tw} + (\sum_w X_{tfw} - 1) K_2 \leq M_{tf} \quad \forall(t, f) \quad (4.1.17)$$

Constraint 4.1.16 is to ensure that a maximum of three products are ripped per flitch. Please note this value can change depending on the machine's design and settings. Constraint 4.1.17 is to ensure that the sum of all the products ripped out ( $\sum_w X_{tfw} P_{tw}$ ) and the sum of all the blades' ( $\sum_w X_{tfw} - 1$ ) kerfs ( $K_2$ ) fit into the maximum edged board width of the flitch ( $M_{tf}$ ).

The optimal ripping pattern (mixture of product widths) for each flitch  $f$  is thus found and after all the flitch boards  $f$  have been virtually ripped, the total supply ( $S_{twh}$ ) of lumber having thickness  $t$ , width  $w$  and log length  $h$  is determined as:

$$S_{twh} = \sum_f X_{tfw} S_{tfh} \quad (4.1.18)$$

$$S_{tw} = \sum_h S_{twh} \quad (4.1.19)$$

The Total Lumber Supply Volume (TLSV) is calculated as:

$$\text{TLSV} = \sum_t \sum_w S_{tw} \quad (4.1.20)$$

### 4.1.3 Fitness Cost Functions

As mentioned earlier in Section 3.1.1, the objective function for a wood manufacturer can consist of three expected costs amongst others to be minimised:

- Raw Material Waste Cost
- Over Production Cost
- Under Production Cost

The Total Raw Material Volume (TRMV) is the volume of all logs supplied. This is done by calculating each log diameter class's ( $d$ ) associated volume per log and multiplying this log volume by the number of logs ( $S_{dh}$ ) supplied. A Log Volume Conversion ( $LVC_{dh}$ ) variable from Huber's formula (Equation 3.2.3) is calculated:

$$\text{LVC}_{dh} = \frac{(M_d + \frac{M_h}{2})^2 \pi M_h}{40\,000} \quad (4.1.21)$$

The Log Volume (LV) per log class  $d$  and length  $h$  is then:

$$\text{LV}_{dh} = \sum_d \sum_h S_{dh} \text{LVC}_{dh} \quad (4.1.22)$$

$$\text{TRMV} = \sum_d \sum_h \text{LV}_{dh} \quad (4.1.23)$$

The Total Raw Material Wasted (TRMW) is then:

$$\text{TRMW} = \text{TRMV} - \text{TLVS} \quad (4.1.24)$$

The Raw Material Cost Rate (RMCR) is the estimated value of the raw material for a wood manufacturer. How this value was estimated is further explained in Section 5.2.1.

The Total Raw Material Waste Cost (TRMWC) is then:

$$\text{TRMWC} = (\text{TRMW})(\text{RMCR}) \quad (4.1.25)$$

To determine over and under production cost we need to first know what the expected demand is, so we define:

$$D_{tw} = \text{Demand of lumber having set thickness } t \text{ and width } w \quad (4.1.26)$$

Note that the length of the product demanded is not included in the equation above since chopping the lumber material into demanded lengths occurs at the next operation (which was not modelled in this study).

The Over Production Volume ( $\text{OPV}_{tw}$ ) of each lumber product having set thickness  $t$  and width  $w$  is:

$$\text{OPV}_{tw} = \begin{cases} S_{tw} - D_{tw} & S_{tw} > D_{tw} \\ 0 & \text{Otherwise} \end{cases} \quad (4.1.27)$$

The Over Production Cost Rate (OPCR) is the estimated value of overstocking, this includes stock keeping and warehousing cost amongst other. How this value is estimated is further explained in Section 5.2.3.

The Over Production Cost ( $\text{OPC}_{tw}$ ) is then:

$$\text{OPC}_{tw} = (\text{OPV}_{tw})(\text{OPCR}) \quad (4.1.28)$$

The Total Over Production Cost (TOPC) is then:

$$\text{TOPC} = \sum_t \sum_w \text{OPC}_{tw} \quad (4.1.29)$$

The Under Production Volume ( $\text{UPV}_{tw}$ ) of each lumber product having set thickness  $t$  and width  $w$  is:

$$\text{UPV}_{tw} = \begin{cases} D_{tw} - S_{tw} & S_{tw} < D_{tw} \\ 0 & \text{Otherwise} \end{cases} \quad (4.1.30)$$

The Under Production Cost Rate (UPCR) is the estimated value of under stocking certain parts. This is usually equal to the cost of buying in these products from another supplier. How this value was estimated is further explained in Section 5.2.2.

The Under Production Cost ( $\text{UPC}_{tw}$ ) is then:

$$\text{UPC}_{tw} = (\text{UPV}_{tw})(\text{UPCR}) \quad (4.1.31)$$

The Total Under Production Cost (TUPC) is then:

$$\text{TUPC} = \sum_t \sum_w \text{UPC}_{tw} \quad (4.1.32)$$

The Total Fitness Cost (TFC) function to which the metaheuristic is improved upon is then the summation of the above 3 costs:

$$\text{TFC} = \text{TRMWC} + \text{TOPC} + \text{TUPC} \quad (4.1.33)$$

## 4.2 PBIL Algorithm

This section describes how the PBIL algorithm was encoded to link and optimise the sawing and ripping operations as formulated above.

### 4.2.1 Formulation of Decision Variables as Bits

With the PBIL the identified decision variables of the problem are encoded as adjacent binary values to form a complete binary string, which is called a Solution Vector (SV) (Figure 3.7). The decision variables of the model, namely the binary assignment of log classes to sawing patterns (Equation 4.1.1) and the priority values of lumber products to be ripped (Equation 4.1.11) were coded into binary format. To illustrate how this was done consider the example shown in Table 4.1 where there are 21 log classes ( $d$ ), 4 sawing patterns ( $c$ ), 2 thickness classes ( $t$ ) and 12 width classes ( $w$ ).

**Table 4.1:** Example of how sawing and ripping decisions were encoded in the PBIL's binary solution vector

	SAWING DECISIONS ( $Y_{dc}$ )				RIPPING DECISIONS ( $P_{rw}$ )				
	d=1	2		21	t=1	t=1		t=2	
c=1	2	3	4	1	2	3	4	...	...
Bit <sub>6</sub>	1	2	3	4	5	6	7	8	...
Solution Vector	1	0	0	0	0	1	1	1	1

**Table 4.2:** Example of how sawing and ripping decisions were encoded in the PBIL's binary solution vector.

	SAWING DECISIONS ( $Y_{dc}$ )				RIPPING DECISIONS ( $P_{rw}$ )			
	d=1	2		21	t=1	t=1		t=2
c=1	2	3	4	1	2	3	4	...
Bit <sub>6</sub>	1	2	3	4	5	6	7	8
SV	1	0	0	0	0	1	1	1
Conversion	1	1	1	1	1	1	1	1
SV (decimal)	1	0	0	0	0	1	0	0

**Table 4.3:** Illustration of how one solution vector (SV) is generated. Note how the constraint in Equation 4.1.9 was enforced for the sawing decisions so that each log class  $d$  is ONLY assigned (=1) to one of the possible sawing patterns  $c$ .

	SAWING DECISIONS ( $Y_{dc}$ )				RIPPING DECISIONS ( $P_{rw}$ )			
	d=1	2		21	t=1	t=1		t=2
c=1	2	3	4	1	2	3	4	...
Bit <sub>6</sub>	1	2	3	4	5	6	7	8
PV	0.3	0.8	0.2	0.1	0.7	0.1	0	0.1
Cumulative PV <sub>6</sub>	0.3	1.1	1.3	1.4	0.7	0.8	0.8	0.9
Rand()	0.62	0.32			0.203	0.796	0.213	0.942
Rand() × CPV <sub>c=4</sub>	0.62 × 1.4 = 0.868	0.288			0	0	0	0
SV	0	1	0	0	1	0	0	0



The sawing operation's decisions is represented in binary format (0 or 1), where a 1 means the bit's associated log class ( $d$ ) is assigned to the bit's associated sawing pattern ( $c$ ). Alternatively a 0 means the log class is not assigned to the sawing pattern.

For the ripping operation's decisions each priority value for all the lumber products ripped having set thickness  $t$  and width  $w$  were encoded as a 7 bit/string, which represent decimal values ranging from 0-128.

Matrix multiplication was used to convert each solution vector (containing  $b$  bits) into the associated binary solution for the sawing operation's decisions (straight 1 to 1 conversion) and the decimal solutions for the ripping decisions (7 bit binary conversion), this is explained with example given in Table 4.2.

### 4.2.2 Model's Constraints within PBIL

The Constraint 4.1.9 ( $\sum_c Y_{dc} = 1$ ), to ensure that all the logs falling within the set log class  $d$  are assigned to only one sawing pattern  $c$ , was enforced into the model by manipulating the PBIL probability vector for the bits representing the sawing decisions (Table 4.3).

To explain how this was done look at the example of log class  $d = 1$  in Table 4.3. The probability vectors (chances that the solution vectors will contain a 1 or 0) for  $c = 1, 2, 3, 4$  are 0.3, 0.8, 0.2 and 0.1 respectively (shaded in Table 4.3). Accumulating these values gives the regions 0-0.3, 0.3-1.1, 1.1-1.3 and 1.3-1.4 respectively. Only one of these four sawing patterns ( $c$ ) must be assigned for the log class ( $d$ ). A random number is chosen (0.62) which is normalised to the total region (1.4) described above to 0.868 (0.62x1.4). This falls within the range 0.3-1.1 ( $c = 2$ ), so sawing pattern 2 is assigned (=1).

## 4.3 Simulation of Fitches' Maximum Edged Board Width

The value in Equation 4.1.7 ( $SIM_{dctf}$ ) was determined through static simulation techniques.

The number of fitches in Equation 4.1.7 was converted to represent a proportion of fitches. Within each log diameter class  $d$  a 1 000 logs, with distributed taper and sweep values, were virtually generated and sawed within SIMSAW through each of the 4 possible sawing patterns  $c$ . The proportion ( $SIM_{dctf}$ ), sometimes resembling a distribution of some sort, of boards (fitches maximum edged board width) produced per log class was calculated by dividing the sum

of boards produced (output) by the 1 000 logs sawed (input). Appendix D contains the various data.

## 4.4 Modelling Software Used

Except for SIMSAW 6 determining the simulated proportion of flitch boards variable in Equation 4.1.7 (data parameter in model) all equations were modelled in the AIMMS 3.13 modelling environment. The MIP problem for the ripping was solved, through the commercial CPLEX 12.4 solver licensed within AIMMS, within the PBIL's execution code. The model's complete code was executed in AIMMS and is provided in Appendix E.

## 4.5 Conclusion

The whole model is summarised in Figure 4.4. As shown the metaheuristic (PBIL) strives to find better (fitter) solutions in terms of low raw material waste, under and over production costs. The decision variables for the sawing operations is the assigned sawing patterns to log classes ( $Y_{dc}$ ), which are binary (0 or 1) values. The sawing operation was represented by SIMSAW data. The decision variables for the ripping operation are the part priority values for lumber able to be ripped ( $P_{tw}$ ). The ripping operation is modelled and solved through a MIP solver, namely the CPLEX 12.4 solver found in AIMMS.

One of the key achievements of this chapter was having the parameter values ( $P_{tw}$ ) of the MIP becoming decision variables for the metaheuristic. Thus every time the metaheuristic produces a new solution vector the MIP's parameter values are changed and what products to be ripped is solved for the different priority values ( $P_{tw}$ ). Another achievement was incorporating the constraint that only one sawing pattern may be assigned to a log class (Constraint 4.1.9) in the PBIL's code. This was described in detail under Section 4.2.2.

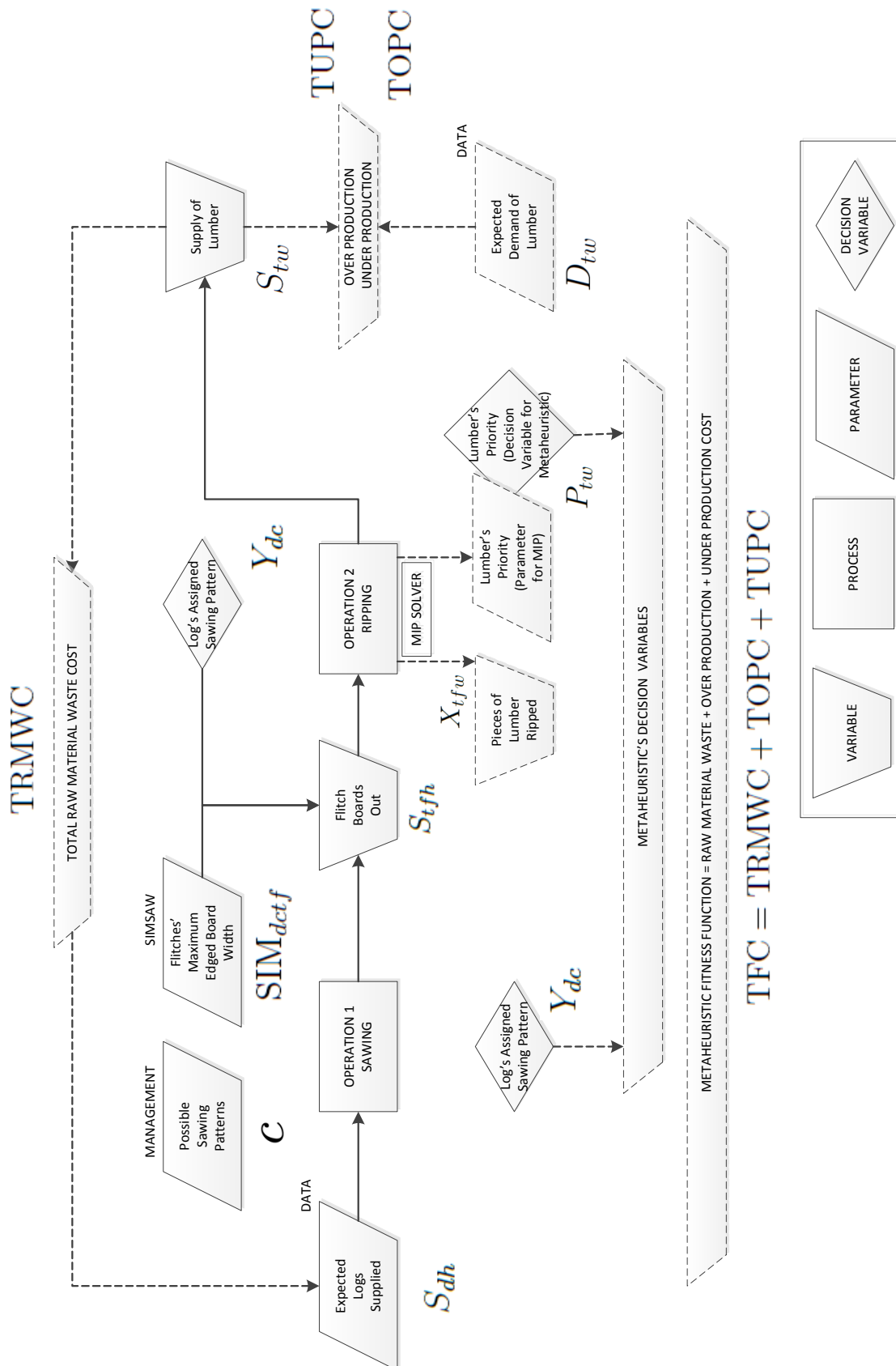


Figure 4.4: The proposed model at a glance with variables and methods used.

## Chapter 5

# Parameter Estimation from Case Data

The case company is a pine and hardwood wood product manufacturer in South Africa. Data for the model was captured with the help of management and the company's Enterprise Resource Planner (ERP).

### 5.1 Possible Sawing Patterns

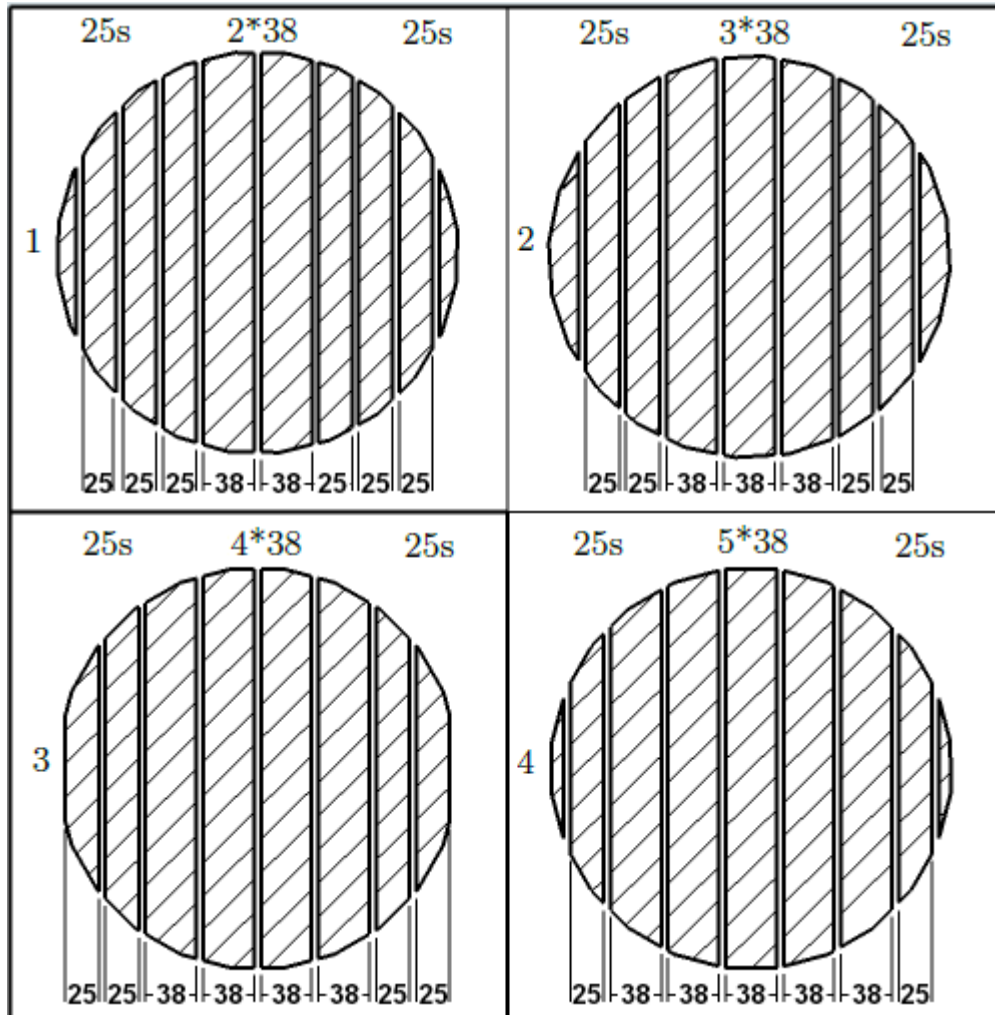
Management usually has four sawing patterns setups available to choose from, Figure 5.1 illustrate the four possible live sawing patterns management chooses from.

### 5.2 Financial Parameters

Table 5.1 shows the parameters that were estimated and used in this model. The South African (SA) Lumber Price Index (LPI) is an industry standard estimation of what the nominal price of lumber is. How and why these values were estimated are explained in the following subsections.

**Table 5.1:** Financial monthly rates used by the model, costs were estimated by company's management accountants from ERP data and the SA LPI.

Measurement	Value	Unit	Source
Raw Material Cost Rate (RMCR)	2 175	R/m <sup>3</sup>	ERP: May 2013
Over Production Cost Rate (OPCR)	138	R/m <sup>3</sup>	ERP: May 2013
Under Production Cost Rate (UPCR)	2 367	R/m <sup>3</sup>	LPI 2013



**Figure 5.1:** Four possible sawing patterns. The dimensions are in millimetres (mm) and represent the dry thickness the lumber product will be after being kiln dried.

### 5.2.1 Raw Material Waste Cost

As mentioned throughout the study raw material cost is by far the most significant cost for sawmillers (Figure 1.2) and secondary manufacturers (Maness *et al.*, 2009). For secondary manufacturers the raw material costs can comprise 40 to 70 percent of the total manufacturing expenses (Buck, 2009). For sawmills the raw material is the logs supplied by the foresters. The average prices for South Africa's logs differs depending on the industry's traditional log classes as shown in Table 5.2 (Bredenkamp *et al.*, 2012, Ch. 8.7). The average prices calculated by Crickmay & Associates for these logs are shown in Table 5.3. Weighting these values accordingly the average roadside price of logs was calculated as 511.06 R/m<sup>3</sup>.

**Table 5.2:** Traditional sawlog classes in SA. (Bredenkamp *et al.*, 2012, Ch. 8.7)

Log Class	Length	Thin-end diameter
A	1.8 m to under 3.6 m	130-179 mm
B1	1.8 m to under 3.6 m	180-259 mm
B2	3.6 m and longer	180-259 mm
C1	1.8 m to under 3.6 m	260-339 mm
C2	3.6 m and longer	260-339 mm
D1	1.8 m to under 3.6 m	260-339 mm
D2	3.6 m and longer	340 mm

**Table 5.3:** Weighted average pine sawlog prices at roadside, second quarter 2013 (Allpass, 2013).

Log Class	Weighted Ave. of 3 Lowest Prices	Industry Weighted Average	Weighted Ave. of 3 Lowest Prices
A	170	382	481
B1	183	410	563
B2	293	455	559
C1	246	484	687
C2	440	569	698
D1	324	551	732
D2	499	658	832

Average Roadside prices: **511.06 R/m<sup>3</sup>**

However in this study the raw material cost was rather taken as the cost at which the sawmill sells its lumber to its secondary manufacturers. This is seen as the lost opportunity cost if the manufacturer wastes lumber it could have recovered. This was estimated from the ERP system of the case company as being 2 175 R/m<sup>3</sup> (Table 5.1).

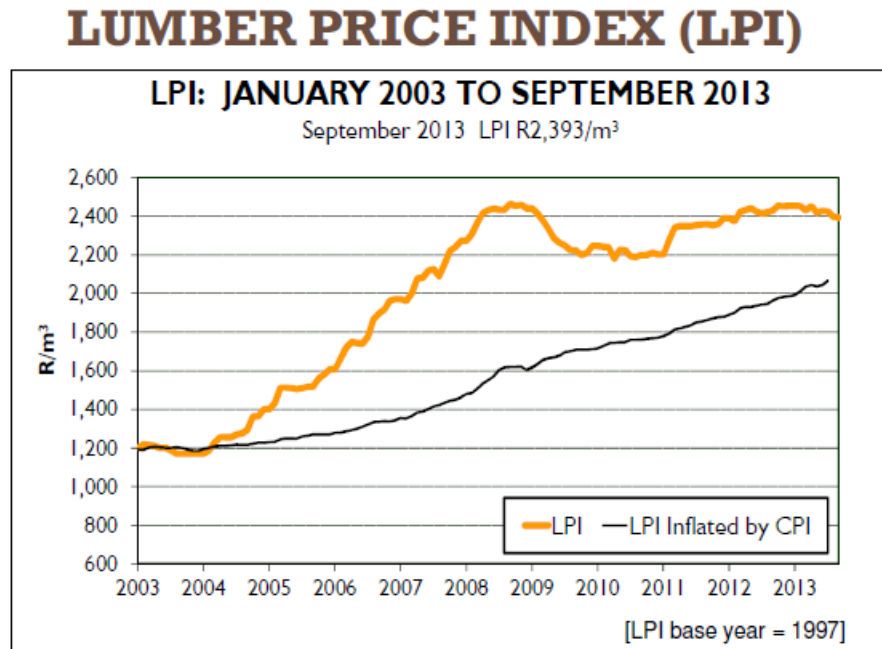
### 5.2.2 Under Production Cost

Under production of wood product parts can delay assembly or delivery of the final product (Maness *et al.*, 2009). Some companies incur an extra cost to fill their orders by purchasing the under produced parts from alternative sources (Maness *et al.*, 2009).

In the model the under production cost was this estimated buy-in cost of lumber for the case company. The SA LPI was used as the under production cost rate. The LPI is an industry standard estimation of what the nominal price of lumber is. It is estimated by Crickmay & Associates who conduct bench-

marking comparisons amongst forestry contractors and sawmills in a monthly publication (Crickmay *et al.*, 2004).

A lumber buy in price of 2367 R/m<sup>3</sup> was used by the company at the time data was collected, which was very similar to the national lumber price index as shown in Figure 5.2.



**Figure 5.2:** SA LPI September 2013 (Crickmay and Associates, 2013b).

The SPPS similarly has a buy-in cost parameter that the user can change for each dimensional wood product.

### 5.2.3 Over Production Cost

Over production of wood product parts incur handling and damage cost amongst others (Maness *et al.*, 2009). The over production cost estimate includes warehousing and related costs necessary for overproduced lumber.

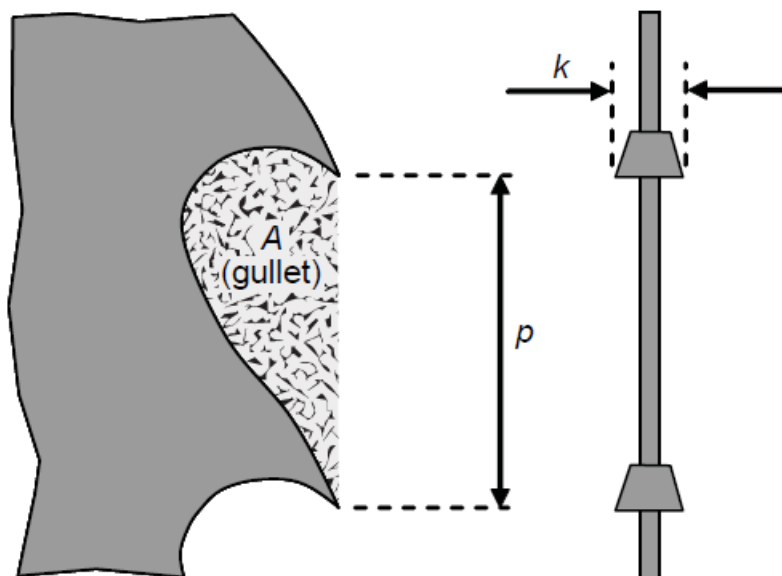
The SPPS package does not have a cost factor for overproducing, instead it has a constraint option, where the user can specify the maximum amount of products allowed to be sawn. This was however left as unconstrained when this project's model was compared to the SPPS, since determining what this maximum value of production should be would be another study of its own, unless a manufacturing company has a mandate or policy on the allowable

overproduction.

For the case company it was estimated that its over production cost, which included its stock keeping and warehousing costs was 138 R/m<sup>3</sup> per month. This was calculated by dividing all the costs (Rand) at the company's warehouse by the volume (m<sup>3</sup>) of lumber it holds on average for a month.

### 5.3 Machine Parameters

The machine parameters of interest included the blade's kerf measurements for the saws at operation one and two. Saw kerfs are the width of the path cut by the saw teeth as the saw blade moves through a log or piece of wood material Vuorilehto (2001).



**Figure 5.3:** Side and front views the framesaw's teeth. For this project only the kerf ( $k$ ) is of interest. (Wessels, 2009a)

**Table 5.4:** Kerf of blades for machines at operation 1 and 2

Measurement	Value	Unit
Kerf for saw blades at sawing operation ( $K_1$ )	5.0	mm
Kerf for saw blades at ripping operation ( $K_2$ )	4.7	mm



The values in Table 5.4 were determined after consulting the manufacturer's technicians for the mentioned sawing and ripping machine. A 4.7 mm kerf is also very similar to the common kerf thickness simulated by the ROMI-3 rough mill simulator, which are 0.125 inches (3.17 mm) and 0.1875 inches (4.76 mm) (Weiss and Thomas, 2005, 74). In some other studies it is stated that the edger's kerf width is typically 2.0-3.0 mm wide (Todoroki and Rönqvist, 2002). The kerf of the saws of the ripping machine was a parameter in the model, which can be changed by the user. It might be of further interest to analyse how much improvement can be expected if this value is decreased, since this will translate into significantly less wood chip waste.

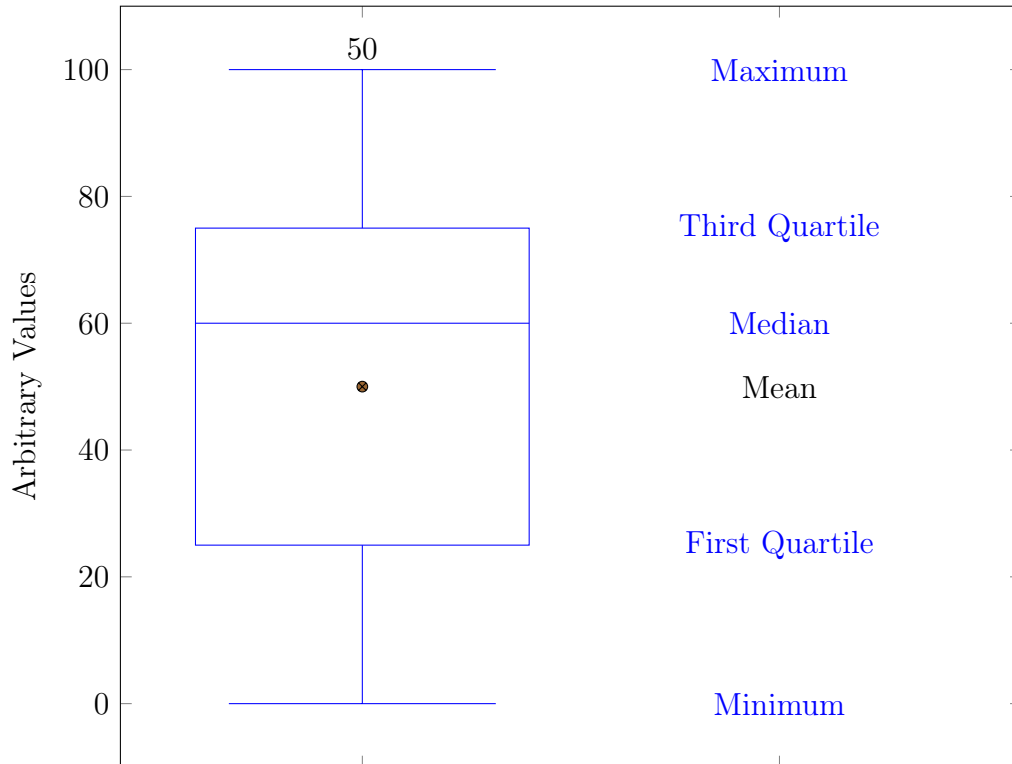
## 5.4 Supply and Demand Scenarios

For the varying supply and demand parameter values of the model, three scenarios (time periods) were chosen to be run through the model and compared to SPSS.

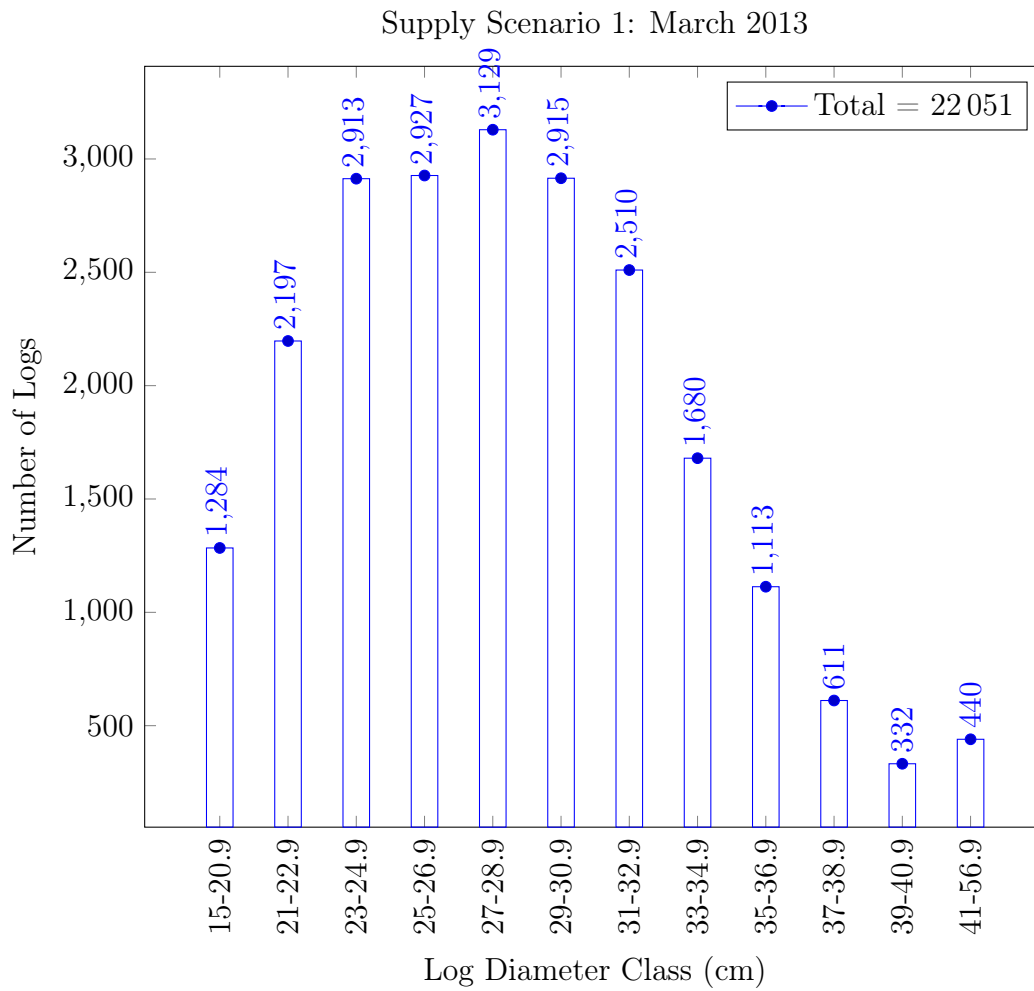
**Table 5.5:** Different supply and demand scenarios run through the model.

	<b>Scenario 1</b>	<b>Scenario 2</b>	<b>Scenario 3</b>
Log Supply	Mar. 2013	Feb.-Jun. 2013	Jul. 2012-Jul. 2013
Avg. Monthly Vol.	4 499 m <sup>3</sup>	4 815 m <sup>3</sup>	4 794 m <sup>3</sup>
Lumber Demand	Jan.-Dec. 2012	Feb.-Jul. 2013	Jul. 2012-Jun. 2013
Avg. Monthly Vol.	2 451 m <sup>3</sup>	3 071 m <sup>3</sup>	2 329 m <sup>3</sup>
Avg. Vol. Recovery	54.48%	63.78%	48.58%

The different supply and demand scenarios are shown in Figures 5.5 to 5.10. All the logs were supplied at a length of 3 meters for the case company. The average (mean) monthly values were the deterministic values run through the model. Box plots are used to represent the variance of the data and is explained in the example figure below:

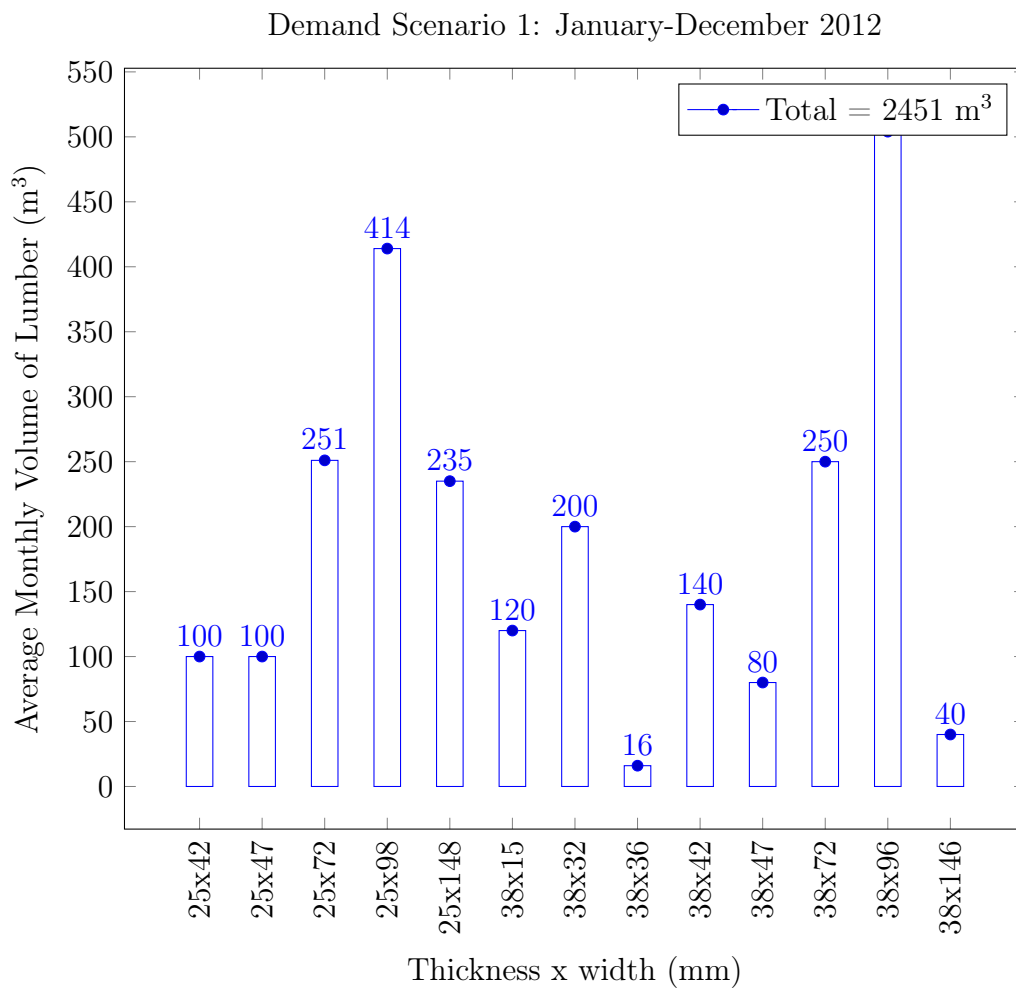


**Figure 5.4:** Example to show the different values indicated in the box plots used in Figures 5.5 to 5.10.

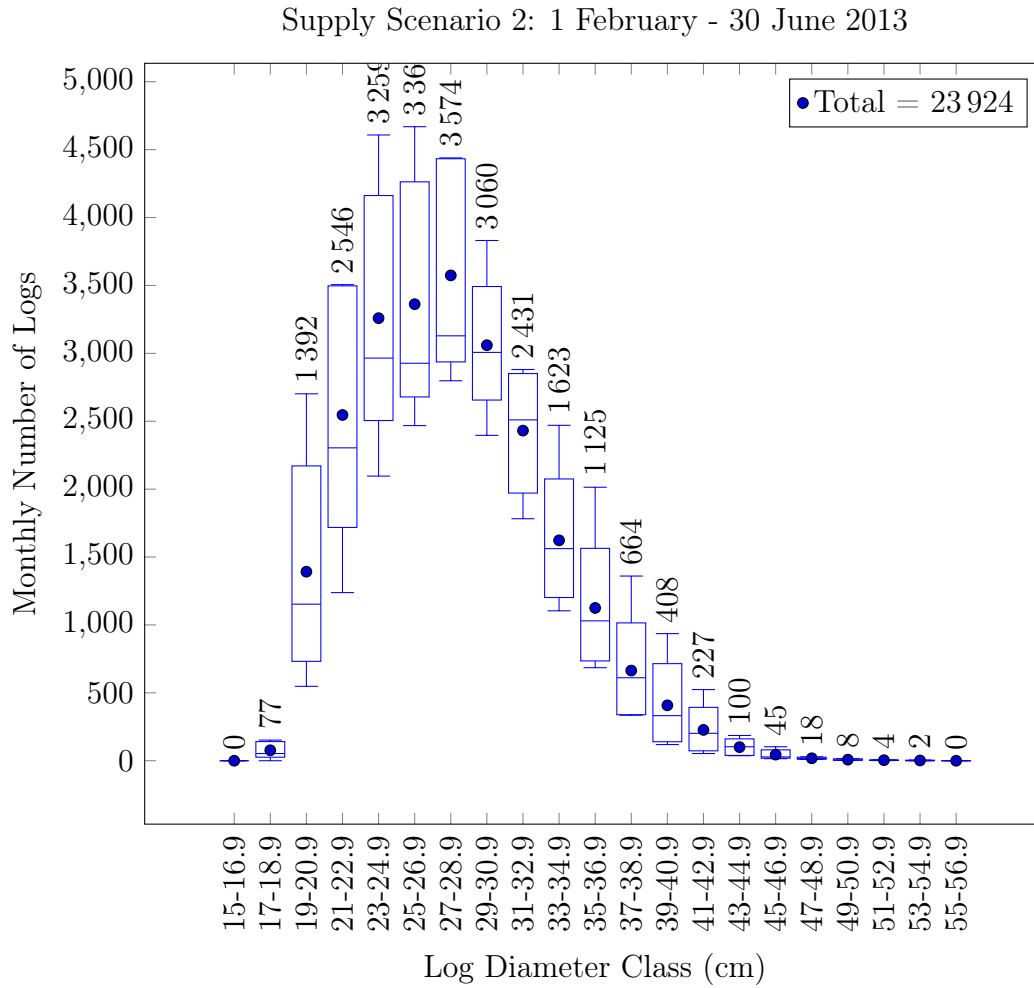


**Figure 5.5:** The number of logs supplied to the case company's sawmill. Total volume = 4 499 m<sup>3</sup>.

The first scenario's data was the initial data used to test the model. This was the number of logs supplied for March 2013. The demand of lumber represents the expected demand the ripping operation was to supply on average per month for the 2012 year. This scenario and its results was published as a paper in the 42nd Annual Conference of the Operations Research Society of South Africa (ORSSA) (Lindner *et al.*, 2013), which should be published by the 20th of December.

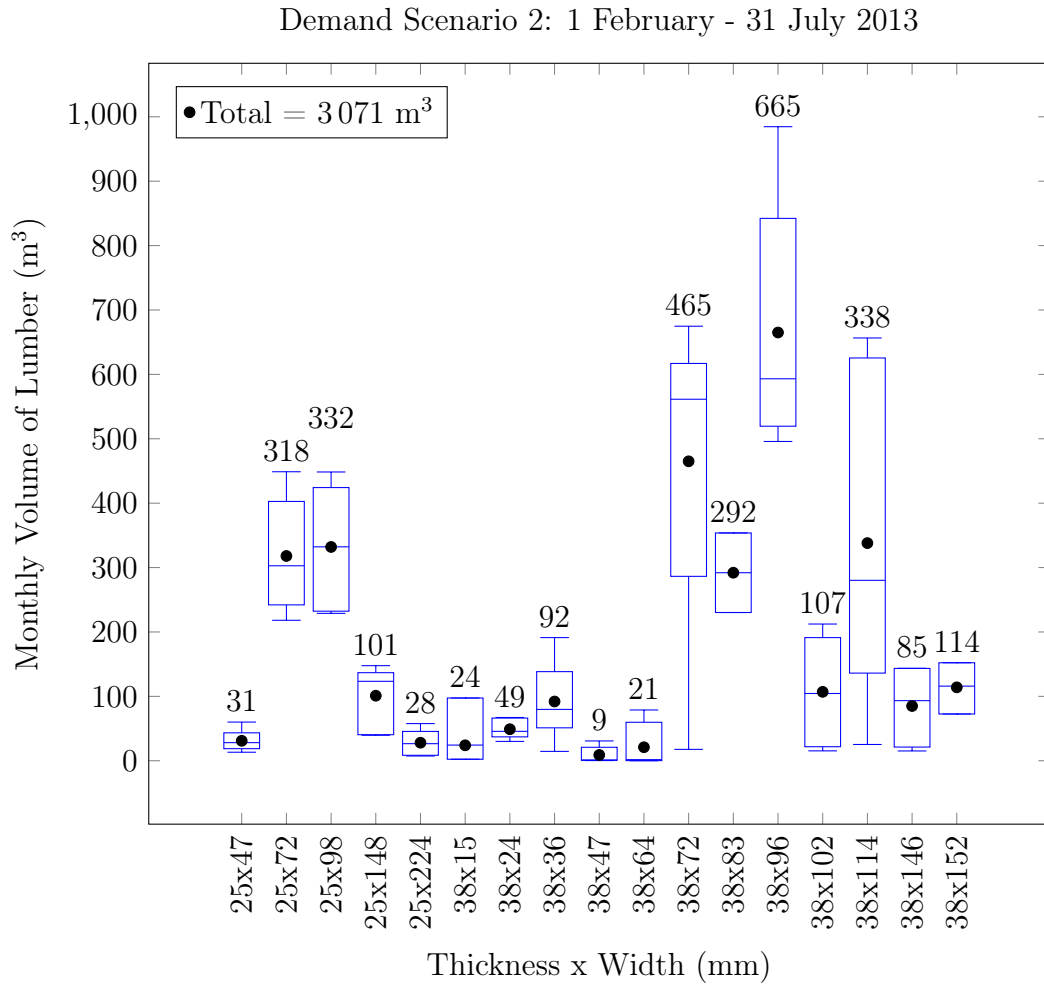


**Figure 5.6:** Average monthly volume of lumber products supplied by the secondary manufacturer.



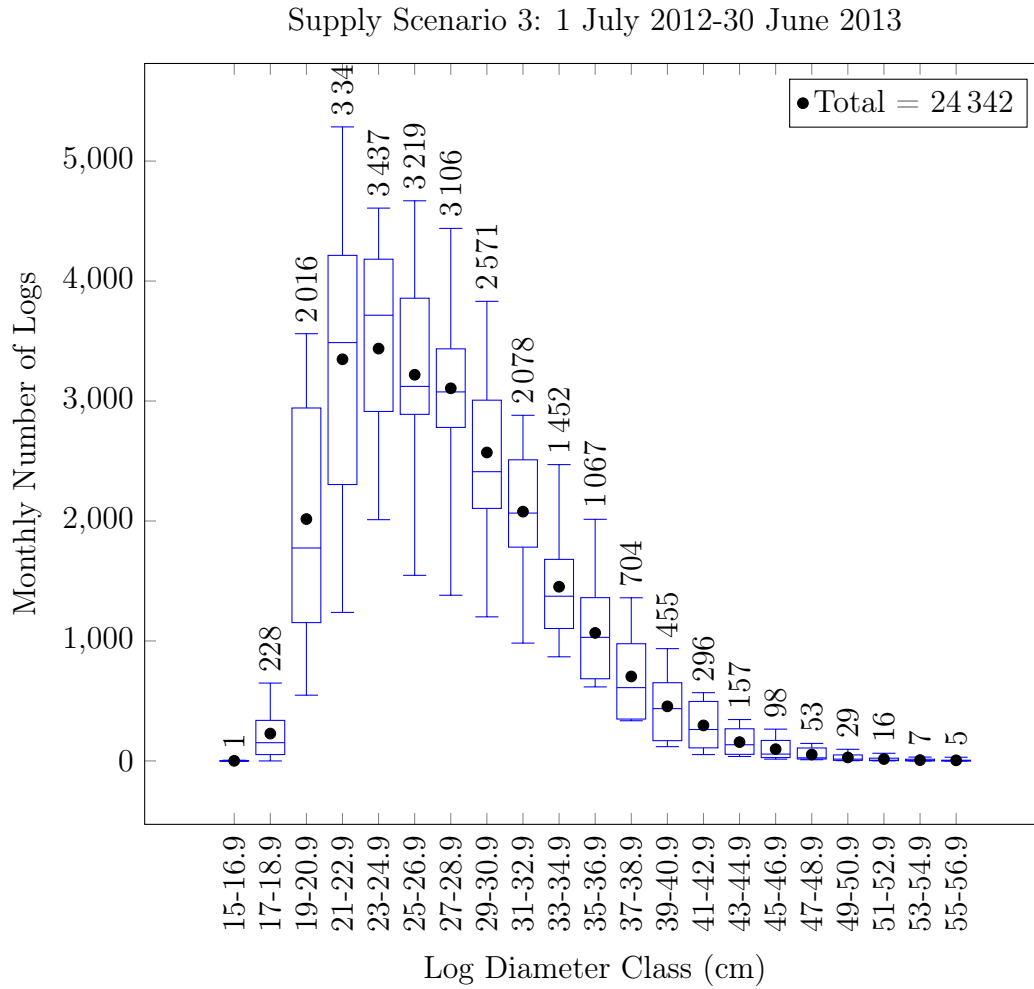
**Figure 5.7:** Box plots of monthly number of log classes supplied to the case company’s sawmill. Total volume = 4 815 m<sup>3</sup>.

For Scenario 2, the number of logs supplied were the average values for 1 February - 30 June 2013, which represents the case company’s last 5 months of operation for its 2012 financial year (July 2012 - June 2013). The demand of lumber represents the ripping operation’s supply for the months ranging from 1 February 2013 - 31 July 2013.

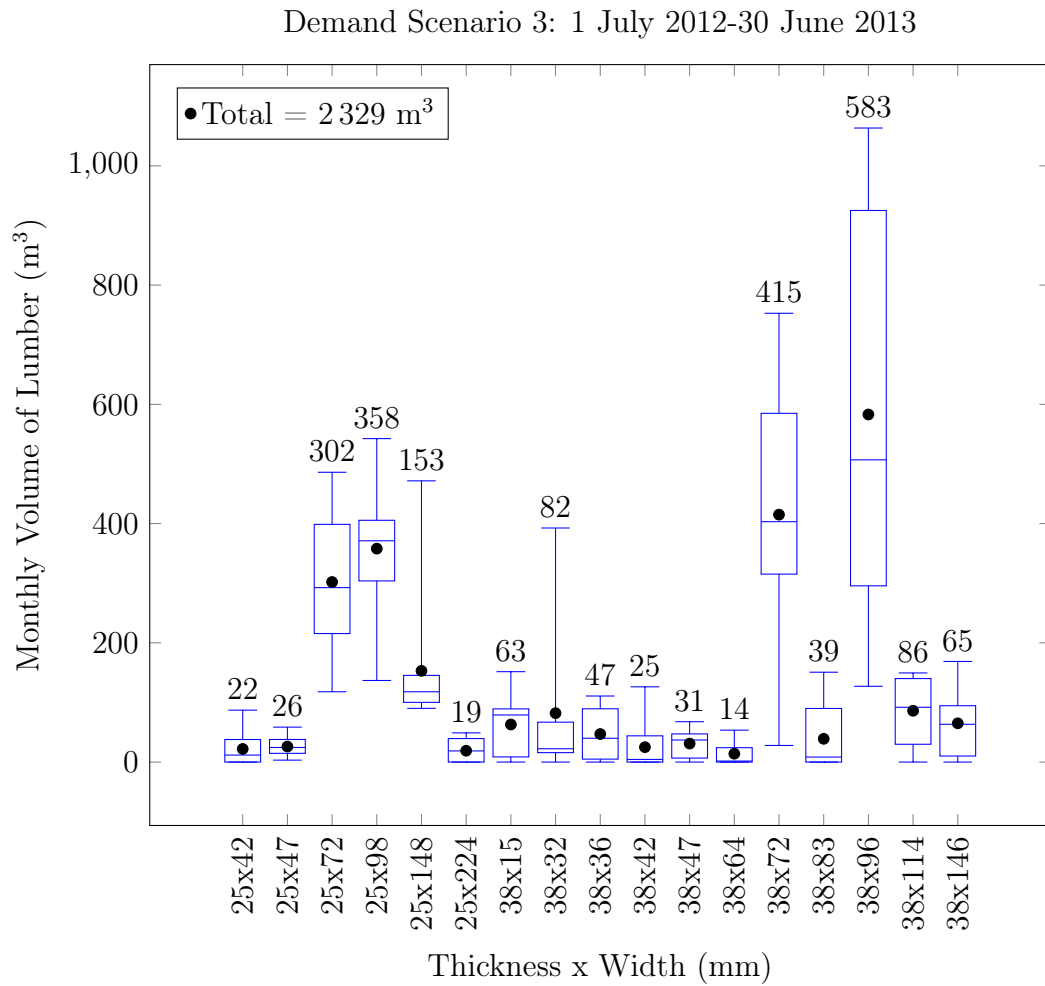


**Figure 5.8:** Box plot of monthly volume of lumber supplied by the secondary manufacturer.

For Scenario 3, the number of logs supplied were the average values for 1 July 2012 - 30 June 2013, which represents the case company's 2012 financial year. Similarly the demand of lumber represents the ripping operation's supply for months ranging from 1 July-30 June 2013.



**Figure 5.9:** Box plots of monthly number of logs classes supplied to the case company's sawmill. Total volume = 4 794 m<sup>3</sup>.



**Figure 5.10:** Box plot of monthly volume of lumber supplied by the secondary manufacturer.



## 5.5 Conclusion

This chapter determined what the important parameter values of the model proposed in Chapter 4 should be. This included the financial parameters, namely the raw material waste, over and under production cost rates and the machine parameters, namely the kerf (saw blade teeth's thickness) for the sawing and ripping operations. Three different supply and demand scenarios were proposed to evaluate the model. Since the model is deterministic, where there was variance, the average values were used.

# Chapter 6

## Results and Discussion

This chapter discusses the most important results found after running the model proposed in Chapter 4 with the case data and parameters estimated in Chapter 5. Most of the discussion revolves around the results provided in the tables and figures throughout this chapter. Also further concluding remarks can be found in Chapter 7.

The model's results were compared to the SPPS package since it is the most similar modelling tool or technique used in South Africa. However, it differs somewhat in its problem formulation. It determines optimal sawing patterns assuming that edging/ripping maximises volume. In this project's model the machine's part priority value settings were included as decision variables, where value was maximised instead of volume in an effort to satisfy the market demand optimally.

### 6.1 Supply and Demand Scenarios

Different supply and demand scenarios were run through the model as summarised in Table 6.1.

For the different scenarios the sawing and ripping machine's saw kerf ( $K_1$  and  $K_2$ ) were kept constant throughout as 5 and 4.7 mm respectively. Similarly the financial cost rates, namely the raw material waste cost rate, the over production cost rate and the under production cost rate were kept constant throughout the three scenarios as 2 175, 138, and 2 367 R/m<sup>3</sup> respectively.

For scenario 2 and 3, the four live sawing patterns as shown in Figure 5.1 were used. For Scenario 1 slightly different patterns were used, where the 1st pattern saws 1 x 38 mm thick flitches in the middle of the log with subsequent increases until the 4th pattern which saws 4 x 38 mm thick flitches from log. The reason for this difference is because these were the possible sawing

**Table 6.1:** Different supply and demand scenarios run through the model and compared to the SPPS package results

Parameters (unit)	Scenario 1	Scenario 2	Scenario 3
Machine saw kerfs (mm)	Sawing blades ( $K_1$ ) - 5 & Ripping blades ( $K_2$ ) - 4.7 Table 5.4		
Financial Rates (R/m <sup>3</sup> )	RMCR - 2 175; OPCR - 138; UPCR - 2 367 Table 5.1		
Sawing Patterns	# 38s - 1,2,3,4	# of 38 mm products sawn - 2,3,4,5 Figure 5.1	
Log Supply (Avg. #/month)	Mar 2013 <sup>n</sup> Figure 5.5	Feb-Jun 2013 <sup>u</sup> Figure 5.7	Jul 2012-Jul 2013 <sup>u</sup> Figure 5.9
Lumber Demand (Avg. m <sup>3</sup> /month)	2012 Figure 5.6	Feb-Jul 2013 Figure 5.8	Jul 2012-Jun 2013 Figure 5.10

<sup>n</sup> Normally generated (in SIMSAW) within each log diameter class  $d$

<sup>u</sup> Uniformly generated (in SIMSAW) within each log diameter class  $d$

patterns used by management at the time the data was collected. The 1st Scenario's data was collected and run through the model slightly earlier to the subsequent scenarios and thus had slightly different sawing patterns.

The average number (#) of logs supplied and the average volume (m<sup>3</sup>) of lumber products demanded varied according to time periods chosen for Scenarios 1 to 3 as illustrated in Figures 5.5 to 5.10. The actual log diameters were distributed normally within the log diameter classes (e.g. 27-28.9 cm) for Scenario 1. Whilst for Scenarios 2 and 3 the actual diameters were distributed uniformly within the log diameter classes ( $d$ ).

The 1st Scenario's data and its results was accepted as a paper in the 42nd Annual Conference of the ORSSA (Lindner *et al.*, 2013), which should be published by the 20th of December.

## 6.2 Model's Best Solutions Found

Firstly, the different PBIL's parameter settings mentioned in Table 3.1, namely the Learning Rate (LR), Mutation Shift (MS) and Mutation Probability (MP) were tested and evaluated under different user defined combinations. The best parameter combination found, after the maximum number of generations (set at 300 in study) was reached or the probability vector converged (termination criteria described in Section 3.6.4), was then re-run a further 500 generations to see if a better solution would still be found by the algorithm.

The absolute best solution found was then compared to the SPPS package's

solution as indicated in Tables 6.2 to 6.4. Also included is the best solution found by having the model randomly search within the decision space. This is equivalent to letting the PBIL algorithm run with 0% LR, MS and MP values respectively (0-0-0). This was done to see what would happen if the model was left to explore the decision space at random (Figure 6.4). More results on the PBIL's different parameter values and associated results are explained in Section 6.3.

**Table 6.2:** Results found by the model's best PBIL parameters (30-5-5) and the model randomly searching (PBIL<sub>0-0-0</sub>), compared to SPPS's solutions. Scenario 1

Measurement	Model's Best	Model Random	SPPS
	PBIL <sub>30-5-5</sub>	PBIL <sub>0-0-0</sub>	MIP Solver
Total Fitness Cost	R5 303 046	R6 065 040	R6 445 488
Raw Material Waste Cost	R4 372 138	R4 422 040	R4 574 444
Under Production Cost	R874 626	R1 550 065	R1 775 162
Over Production Cost	R56 281	R92 517	R95 882
Lumber Recovery	55.30%	54.8%	53.23%
Run time	±1 hour	±1 hour	±1 sec.

**Table 6.3:** Results found by the model's best PBIL parameters (30-5-5) and the model randomly searching (PBIL<sub>0-0-0</sub>), compared to SPPS's solutions. Scenario 2

Measurement	Model's Best	Model Random	SPPS
	PBIL <sub>30-5-5</sub>	PBIL <sub>0-0-0</sub>	MIP Solver
Total Fitness Cost	R6 041 017	R7 052 429	R8 520 947
Raw Material Waste Cost	R4 645 099	R4 724 771	R4 921 419
Under Production Cost	R1 370 912	R2 256 102	R3 469 006
Over Production Cost	R25 005	R71 555	R 130 521
Lumber Recovery	55.58%	54.81%	52.99%
Run time	±1 hour	±1 hour	±1 sec.

**Table 6.4:** Results found by the model's best PBIL parameters (50-5-5) and the model randomly searching (PBIL<sub>0-0-0</sub>), compared to SPPS's solutions. Scenario 3

Measurement	Model's Best	Model Random	SPPS
	PBIL <sub>50-5-5</sub>	PBIL <sub>0-0-0</sub>	MIP Solver
Total Fitness Cost	R5 394 696	R5 947 627	R7 277 559
Raw Material Waste Cost	R4 624 468	R4 708 305	R4 898 814
Under Production Cost	R684 098	R1 132 379	R2 220 234
Over Production cost	R86 129	R106 943	R158 511
Lumber Recovery	55.61%	54.81%	53.00%
Run time	±1 hour	±1 hour	±1 sec.

As can be seen for all three scenarios the model's best solution found was much better (lower) for all the cost functions proposed, where it produced a 18%, 29% and 26% lowering of the Total Fitness Cost (TFC) for Scenarios 1, 2 and 3 compared to SPPS respectively (calculated from Tables 6.2 to 6.4).

The model outperformed the SPPS most significantly for the Total Under Production Cost (TUPC), where it produced a 51%, 60% and 69% reduction in this cost for Scenarios 1, 2 and 3 respectively. This improvement was expected since the model tries to satisfy market demand better by having priority values for each ripped product, which the SPPS package does not have. The SPPS package maximises volume at the ripping operation (i.e. no priority part values).

The model saved a lot of costs by still increasing the volume recovery, inversely lowering the Total Raw Material Waste Cost (TRMWC), where it produced a 4%, 6% and 6% reduction in this cost for Scenarios 1, 2 and 3 respectively. The volume recovered stayed quite similar for each of three methods for all three scenarios. For example the SPPS usually recovered  $\pm 53\%$  lumber volume, with the best solutions found by the model randomly searching being  $\pm 1.7\%$  higher and model's best (optimal) solution found was usually  $\pm 2.5\%$  higher.

Although the cost savings aren't that much in comparison, the model also always improved on the Total Over Production Cost (TOPC). Where it produced a 41%, 80% and 46% reduction in this cost compared to SPPS for Scenarios 1, 2 and 3 respectively. The reason the second scenario's improvement value was much higher in comparison (80%) was because for the second scenario there was a considerable increase of lumber volume demanded, namely a 63.78% volume recovery (Table 5.5) was expected from the log material supplied. Thus the model rarely overproduced on any products. However the SPPS did unnecessarily oversupply on certain products, specifically wider products since its ripping strategy tries to just maximise volume (this is explained more in Section 6.5).

As can be seen in Figure 6.2, the Under Production Cost (UPC) affected the metaheuristic's "fitness" function the most. This was especially evident early on in the generated populations (Figure 6.2). However the raw material cost was by far the highest, but it did not increase or decrease that much in comparison. The over production cost was minimal in comparison, since its cost rate was low (138 R/m<sup>3</sup> compared to 2 174 and 2 367 R/m<sup>3</sup>).

Tables 6.2 to 6.4 also include the estimated time it took for the two programs to find their "optimal" (best) solutions. These values are not entirely accurate as they were run under slightly different computing powers, however the esti-

mated or average time it took is indicated as to show that the better solutions found by the model's PBIL algorithm took much longer than the SPPS. However this substantially longer solving time was not of that much importance since the production plan was for a month's time period and an hour or two should be sufficient for month to month planning. However if shorter planning production time periods are desired (week to week or day to day), then this relatively longer solving time might become a problem.

### 6.3 PBIL's Best Parameters Found

Firstly, the different PBIL's parameter settings mentioned in Table 3.1, namely the LR, MS and MP were tested and evaluated under different user defined combinations. The best parameter combination found was then re-run a further 500 generations to see if a better solution would still be found by the algorithm. The absolute best (optimal) solution found was then compared to the SPPS package's solution as already discussed and indicated in Tables 6.2 to 6.4.

As can be seen in Figures 6.1, 6.5 and 6.7 the higher the learning rate the sooner the algorithm converged to lower cost functions. However there is a tipping point, e.g. 30% in Scenario 2 (Figure 6.5) and 50% in Scenario 3 (Figure 6.7), where higher learning rates, actually find inferior (higher) cost solutions. This is probably because there was not enough diversity (exploring slightly mutated/different solutions) for each population generated.

As can be seen in Figures 6.1, 6.5 and 6.7 higher MS and MP values (e.g. 10%) resulted in the algorithm finding inferior solutions. This is because higher MS and MP forces the probability vector, which produces a population of solution vectors, to randomly mutate and thus choose other values as opposed to those that usually produce good solutions (converging). This is the common trade-off that most algorithms face, where exploitation, meaning more convergence to best solutions found (i.e. a higher learning rate in the PBIL algorithm), means less exploration of the search space and vice versa. This argument is inversely substantiated in Figure 6.5 where the PBIL algorithm had a LR=50%, MP and MS = 0% (50-0-0) which caused the algorithm to converge on a good solution vector already within the 20th generation and was subsequently terminated from exploring the search space further. The PBIL algorithm was thus terminated because the Probability Vector (PV) converged onto values that were either less than 0.05 or greater than 0.95 (for more on this termination rule refer back to Section 3.6.4).

The model even found better results when it randomly searched within the

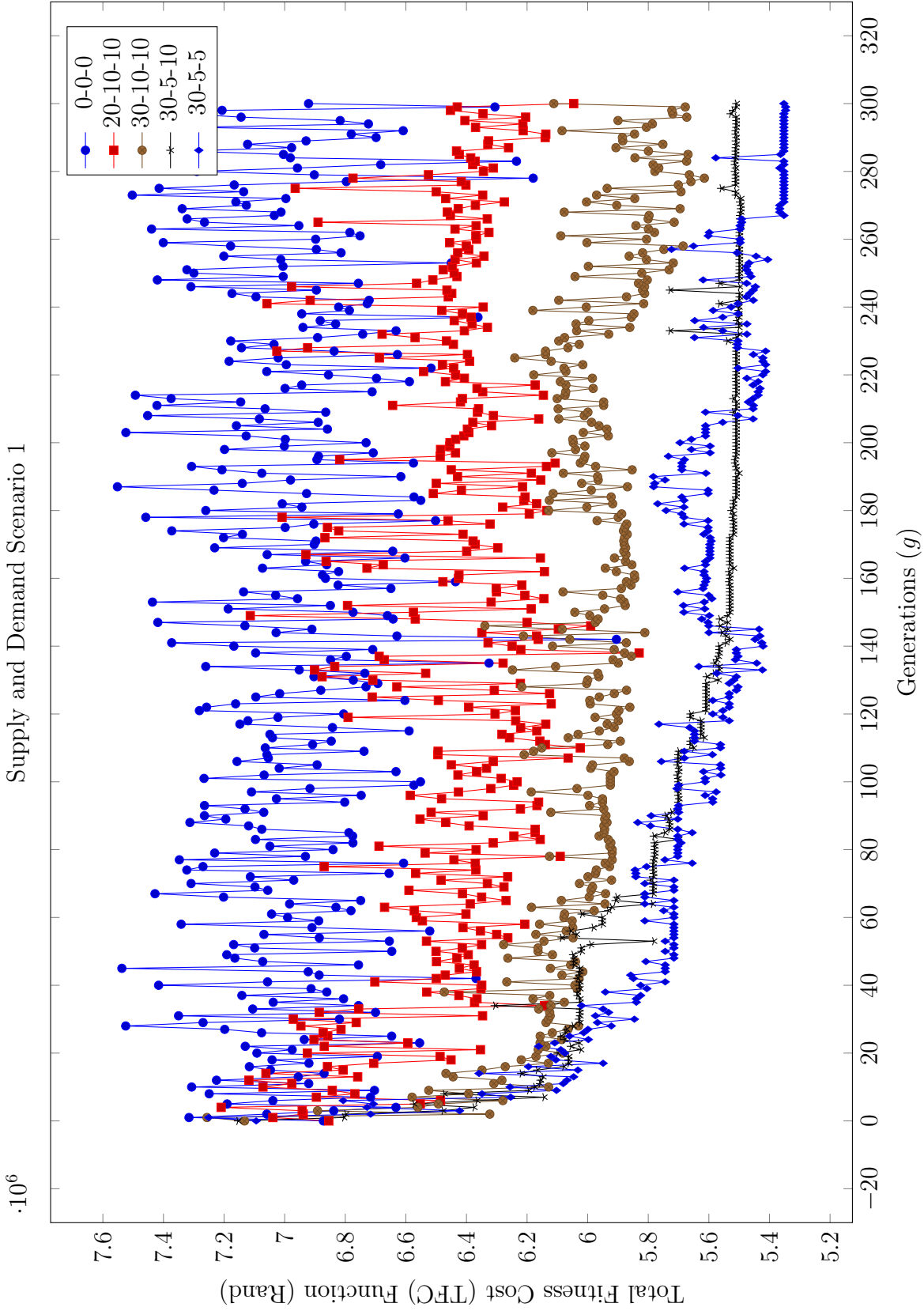
decision space (0-0-0 LR, MS and MP). As seen in Figures 6.4 and 6.7 when the model randomly searched it produced results that hovered above and below the SPPS package's solution (specifically highlighted in Figure 6.4). This was because there was always one sawing pattern selected (constraint 4.1.9) and thus sometimes a relatively good volume was selected for the 30 solution vectors within every population generated. Also because there is a maximising value problem formulated at the ripping operation (Equation 4.1.15), relatively good volume recovery was expected, only slightly affected by the priority values chosen. Because the SPPS package does not cater for priority values, even just choosing random priority values sometimes improved upon the over and under production costs. All of these of factors then lead to the model sometimes "randomly" selecting good solutions as can be seen in Figure 6.4 and 6.7.

In summary the best combination of PBIL parameters found for Scenario 1 was: LR = 30%; MS = 5% and MP = 5% (Figure 6.1). The best combination of parameters found for Scenario 2 was: LR = 30%; MS = 5% and MP = 5% (Figure 6.5). The best combination of parameters found for Scenario 3 was: LR = 50%; MS = 5% and MP = 5% (Figure 6.7). In summary the best PBIL parameter values found across all three scenarios was a LR of 30-50%, a MS and MP of 5%.

## 6.4 PBIL's Parameter Values Re-run

Figure 6.6 shows results for re-running the best parameter values for Scenario 2 (30-50-5). As can be seen the first time the model's PBIL algorithm ran these values it "luckily" found a very good solution early on, which it converged upon and produced populations having similar decision variables. Re-running the exact same parameter values twice produced results that unfortunately did not find such a good solution as in the previous run.

The last two re-runs did follow or converge to similar results (Figure 6.6), which offers some validation to the model developed.



**Figure 6.1:** The model's best (min) cost function found within each generation containing 30 solution vectors (population size in PBIL). This is under the different combinations of the PBIL's parameters proposed. 30-10-5 indicates the combination with LR = 30%, MS = 10% and MP = 5%. The different combinations are ordered from worst (highest) to best (lowest) in the legend.



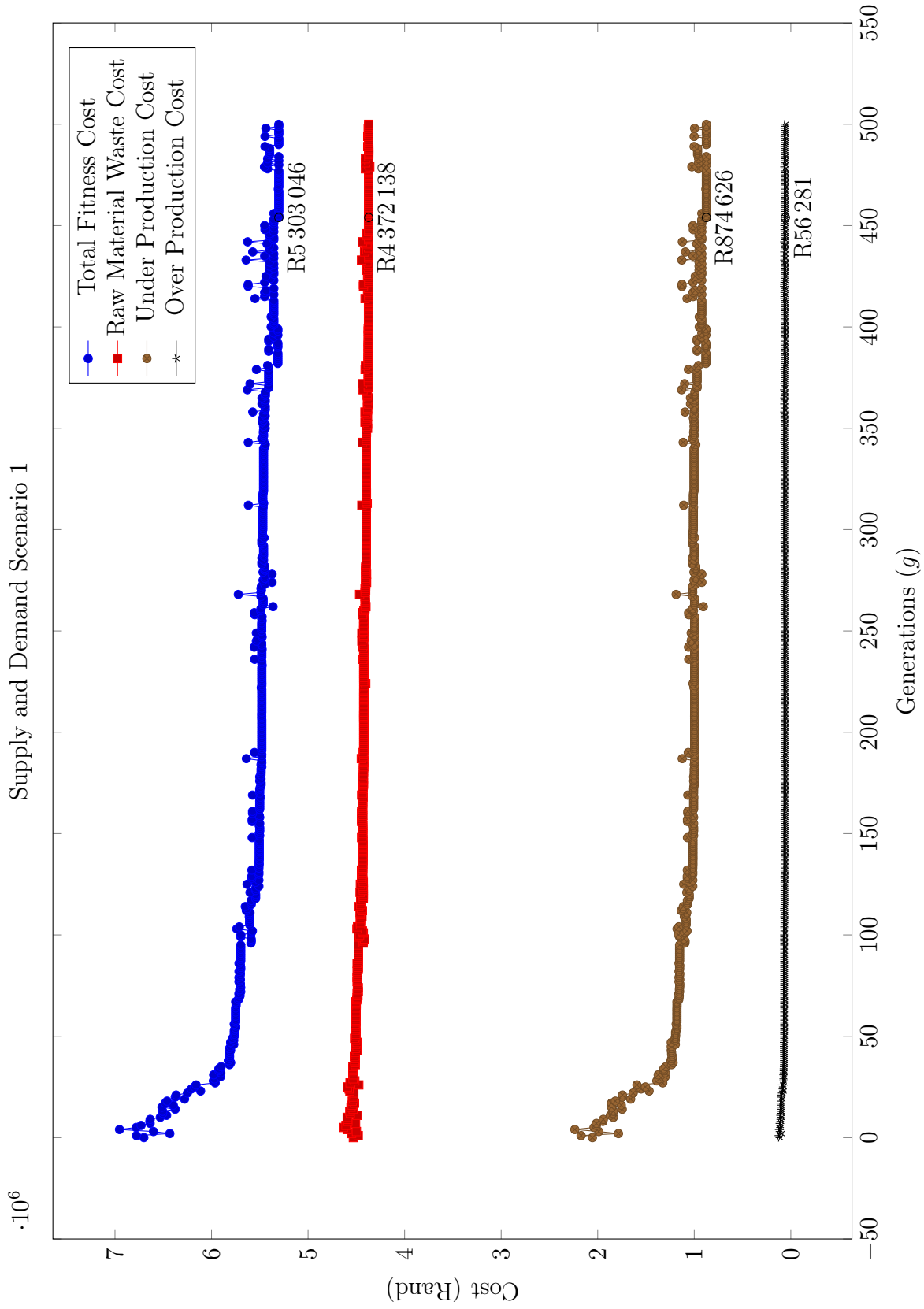
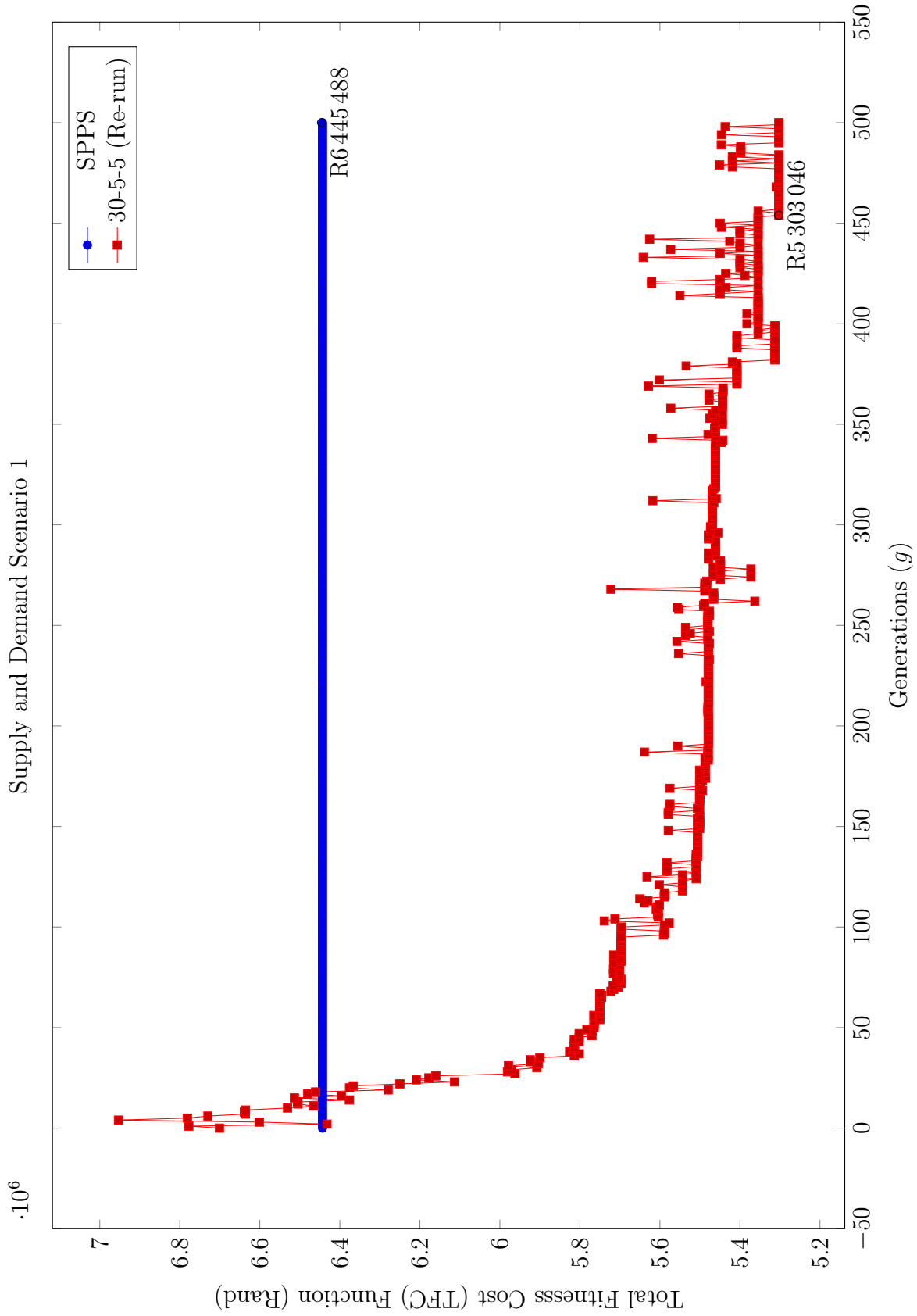
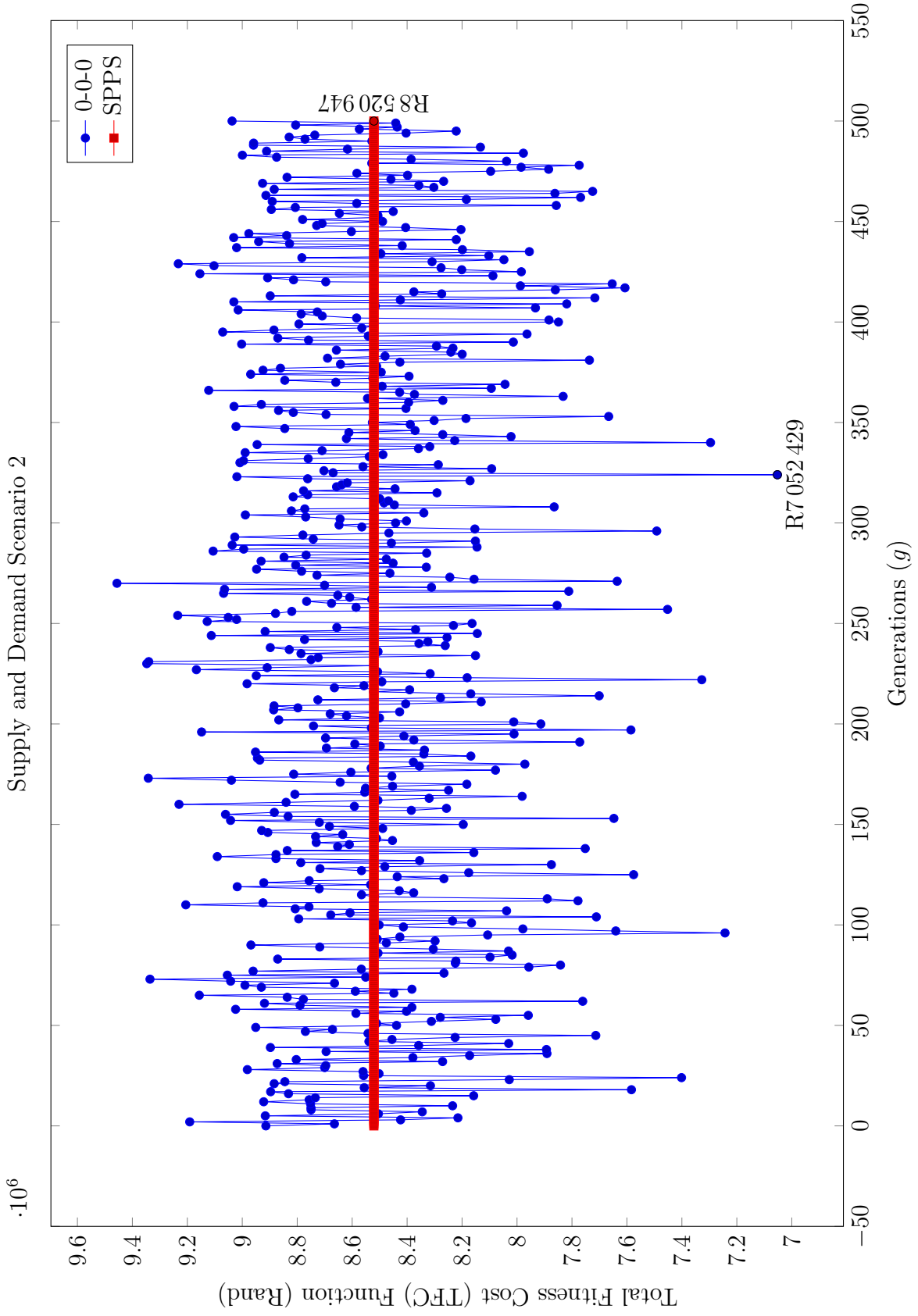


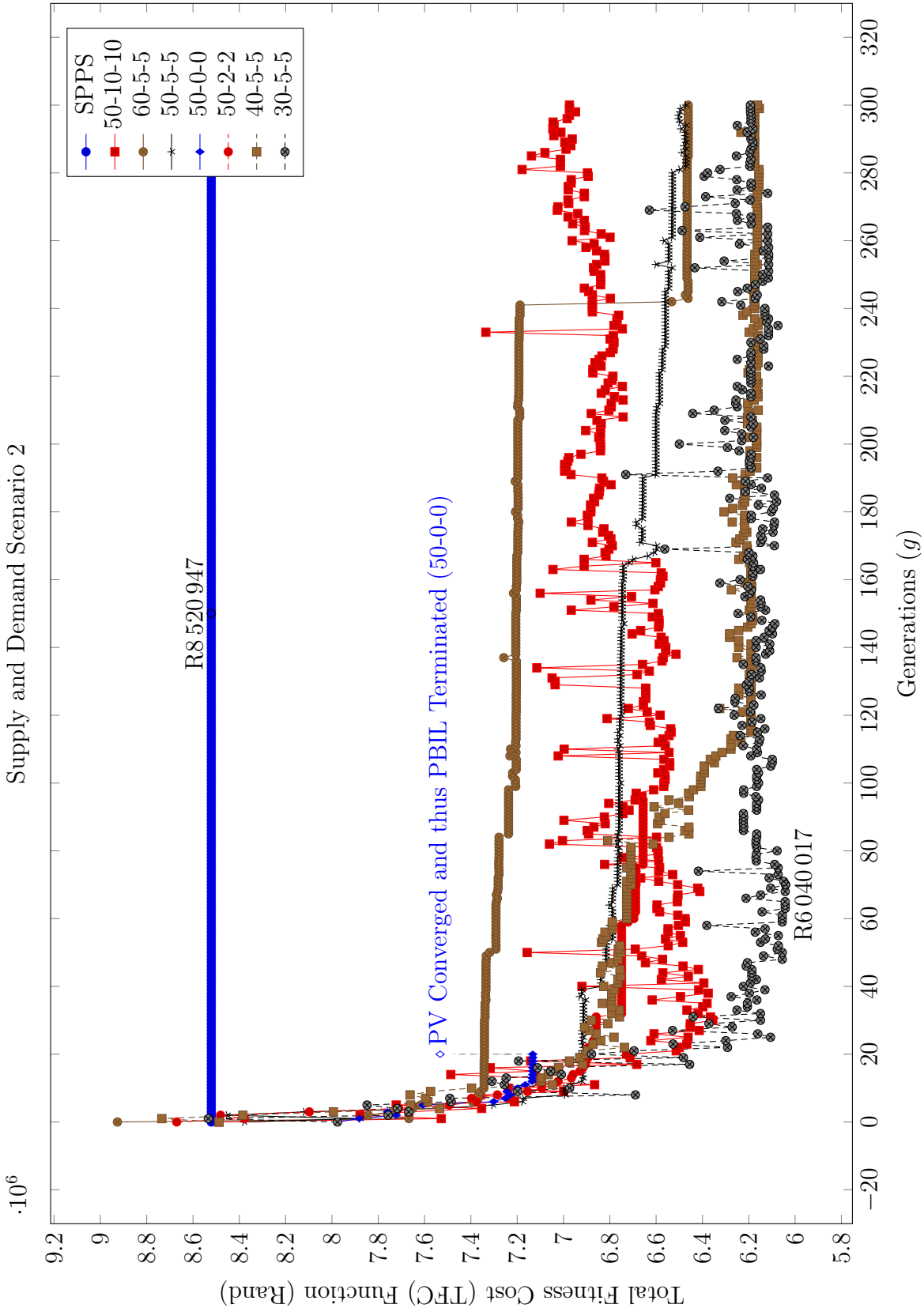
Figure 6.2: The model's best PBIL parameters (30-5-5) re-run with cost breakdown.



**Figure 6.3:** The model's best cost function found compared to SPPS's optimal (constant) solution.



**Figure 6.4:** Model randomly searching (PBI<sub>0-0-0</sub>) compared to SPPS's optimal (constant) solution.



**Figure 6.5:** The model's best (min) cost function found within each generation (300) containing 30 solution vectors (population size in PBIL). This is under the different combinations of the PBIL's parameters proposed. 30-10-5 indicates the combination with LR=30%, MS = 10% and MP = 5%. The different combinations are ordered from worst (highest) to best (lowest) in the legend

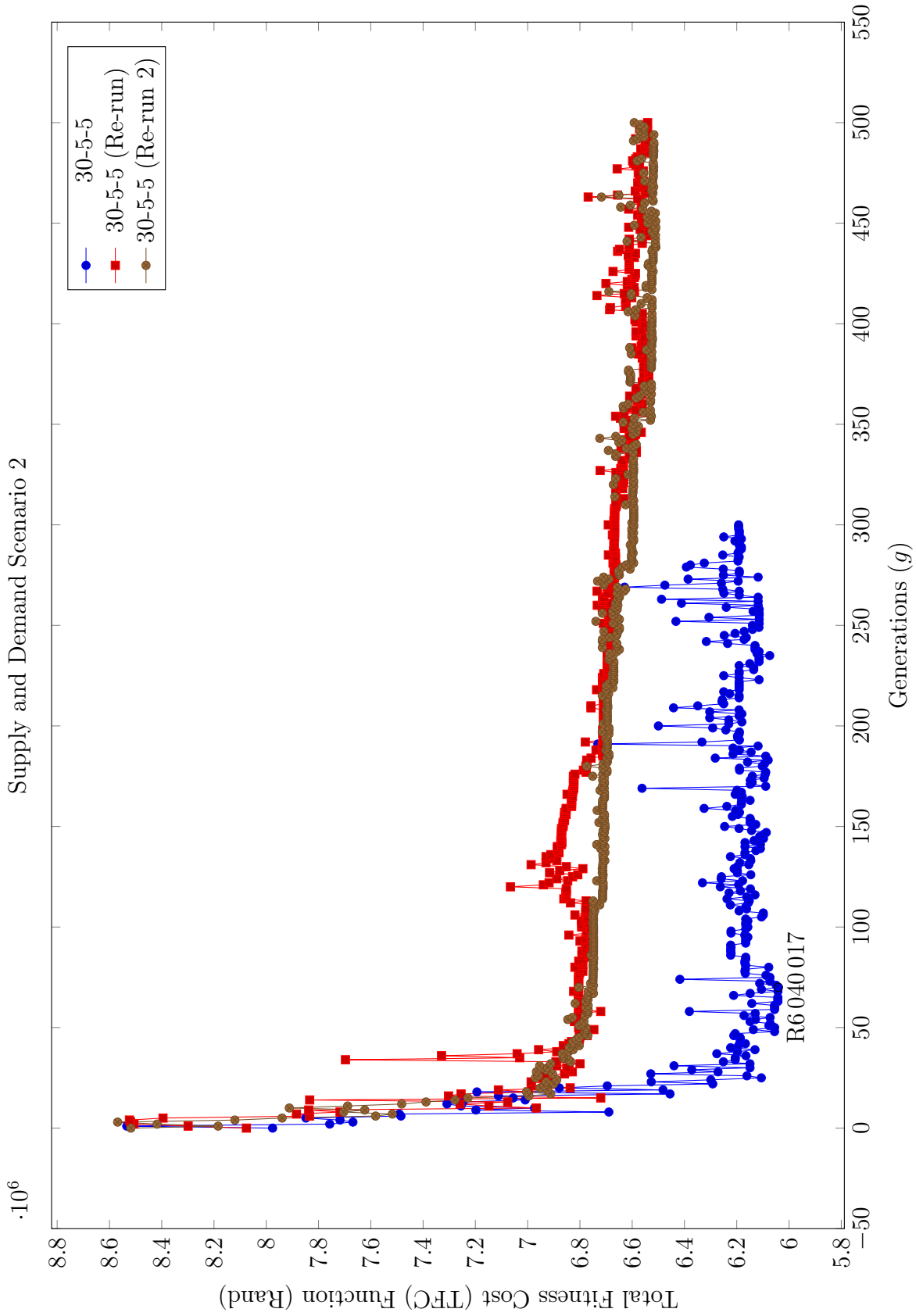
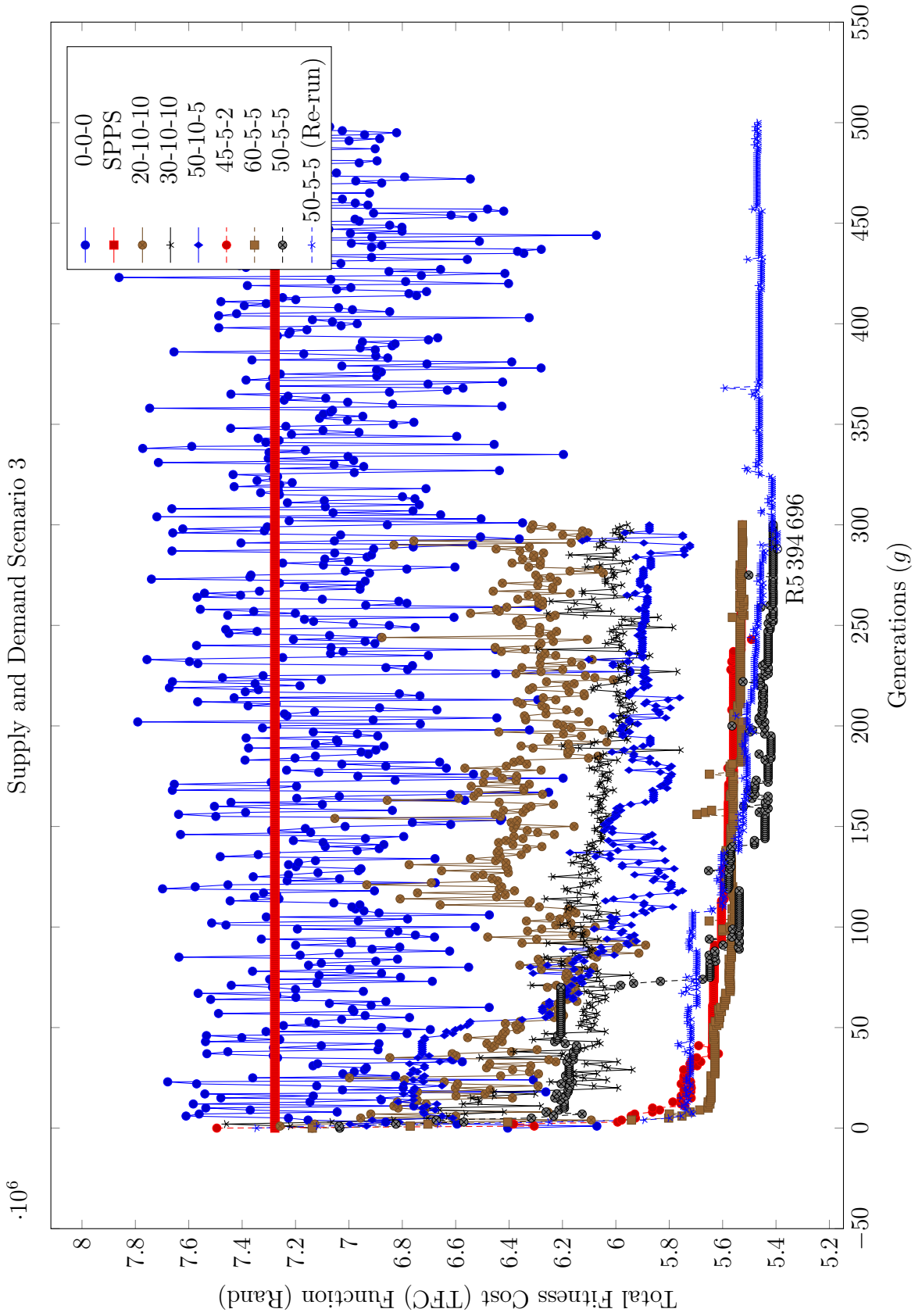


Figure 6.6: The model's best PBIL parameters re-run twice.



**Figure 6.7:** The model's best (min) cost function found within each generation (300) containing 30 solution vectors (population size in PBIL). The different combinations are ordered from worst (highest) to best (lowest) in the legend.

## 6.5 Supply versus Demand Results

Figures 6.8 to 6.10 compare the supply results for the model's best solution found to the expected demand and the SPPS package's supply. The obvious result to note in the figures is that the SPPS package usually over supplied on the wider widths (25x224 mm, 38x148). This is because its ripping optimising strategy is to maximise volume.

The model proposed in this study has the added benefit of assigning priority values to the ripped products and in contrast its ripping strategy is to maximise value. This then translated into satisfying market demand more evenly as is evident in Figures 6.8 to 6.10. This is because the model tries to lower the associated under and over production costs in Tables 6.2 to 6.4, i.e. a solution's "fitness" within the PBIL algorithm is evaluated on meeting demand as best as possible (over and under production "penalty" costs) and at a minimal raw material waste cost.

## 6.6 Model's Optimum Decision Variables

Tables 6.5 to 6.9 show the optimum decision variables for the minimum cost solutions found. As can be seen the model's PBIL algorithm fulfilled the constraint (4.1.9) that only one sawing pattern is allowed to be assigned to a log class, i.e. summing across the log diameter classes ( $d$ ) always equals 1 in Tables 6.6, 6.8 and 6.10.

The proposed model satisfied market demand better than the SPPS package and still produced a higher recovery (inversely a lower raw material waste cost). This is because it has a value optimisation strategy for the ripping operation with added priority value decision variables. For example for all three scenarios the proposed model assigned more logs (classes) to the last sawing pattern ( $d = 4$ ). Note that the latter sawing patterns have a higher volume recovery because less blades are used for the increased number of thicker flitches sawn from the centre of the log (Figure 5.1). Thus the proposed model could choose sawing patterns with higher recoveries and manipulate the ripping operation subsequently to satisfy market demand. The SPPS package on the other hand, rips for maximum volume, and thus it only has the sawing patterns (with set ripping decisions to follow maximum volume recovery) as a decision variable. This concept is further illustrated in Figure 6.11.

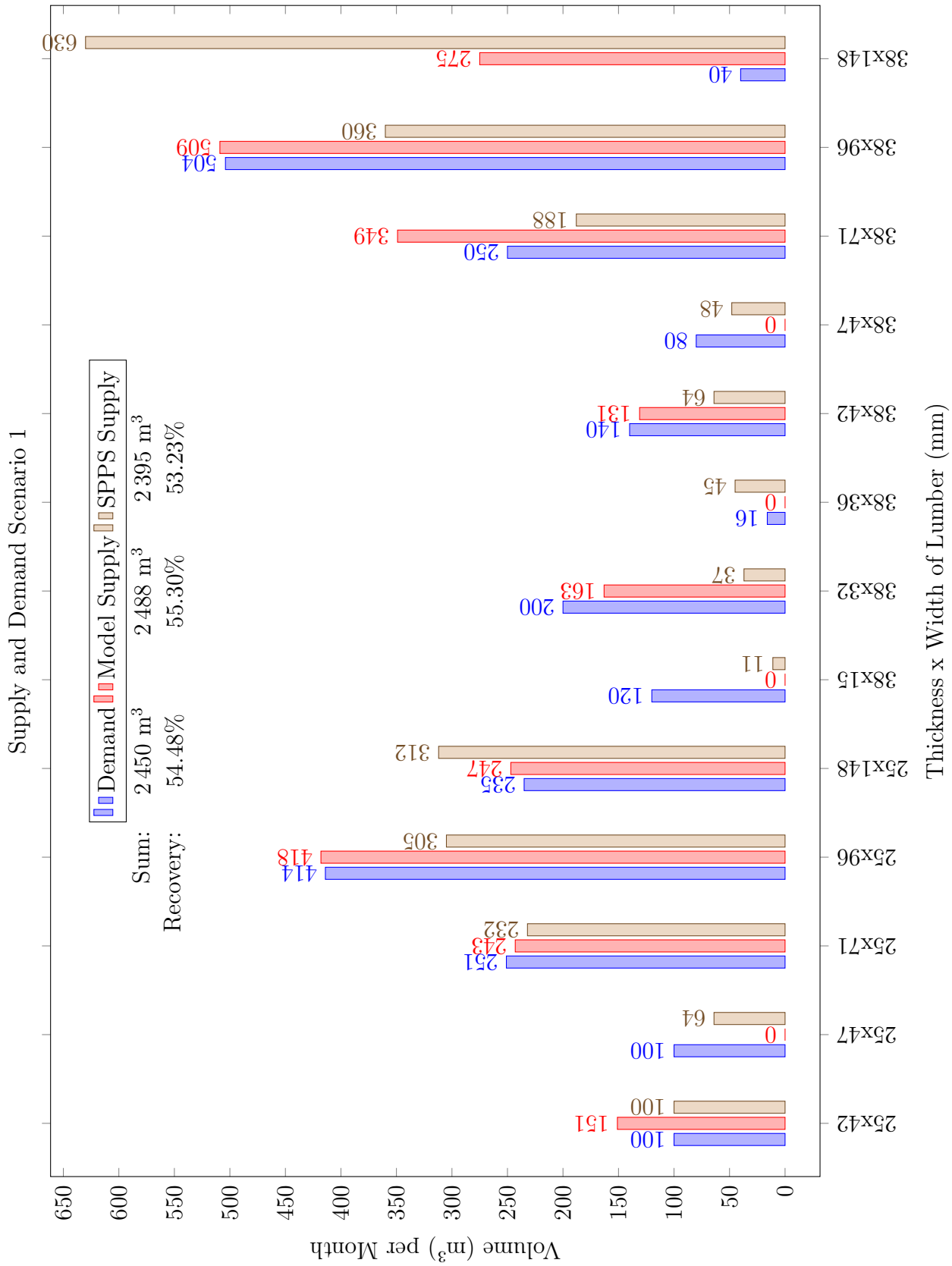


Figure 6.8: Supply versus demand results the proposed model and the SPPS package.



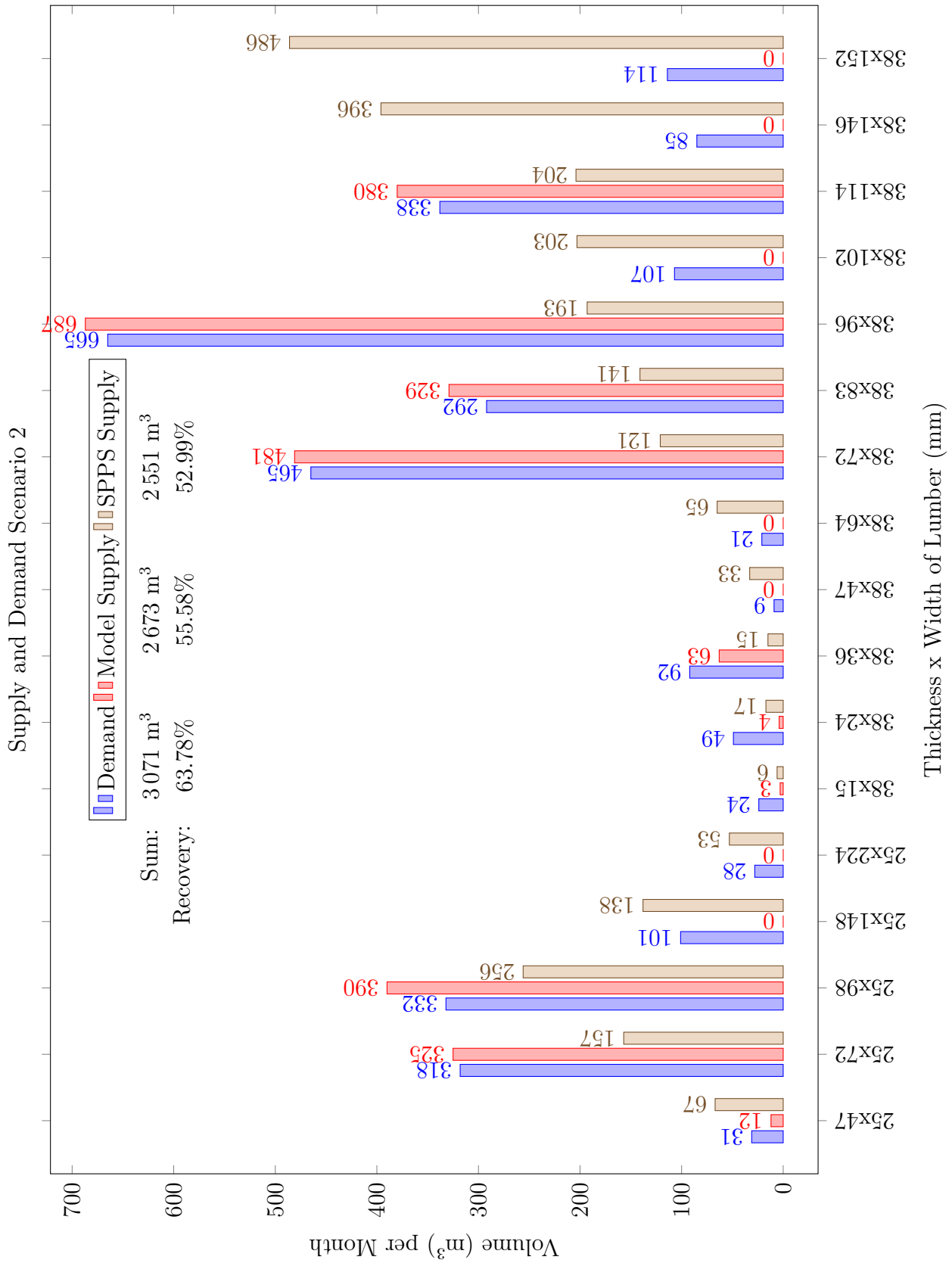


Figure 6.9: Supply versus demand results for the proposed model and the SPSS package.

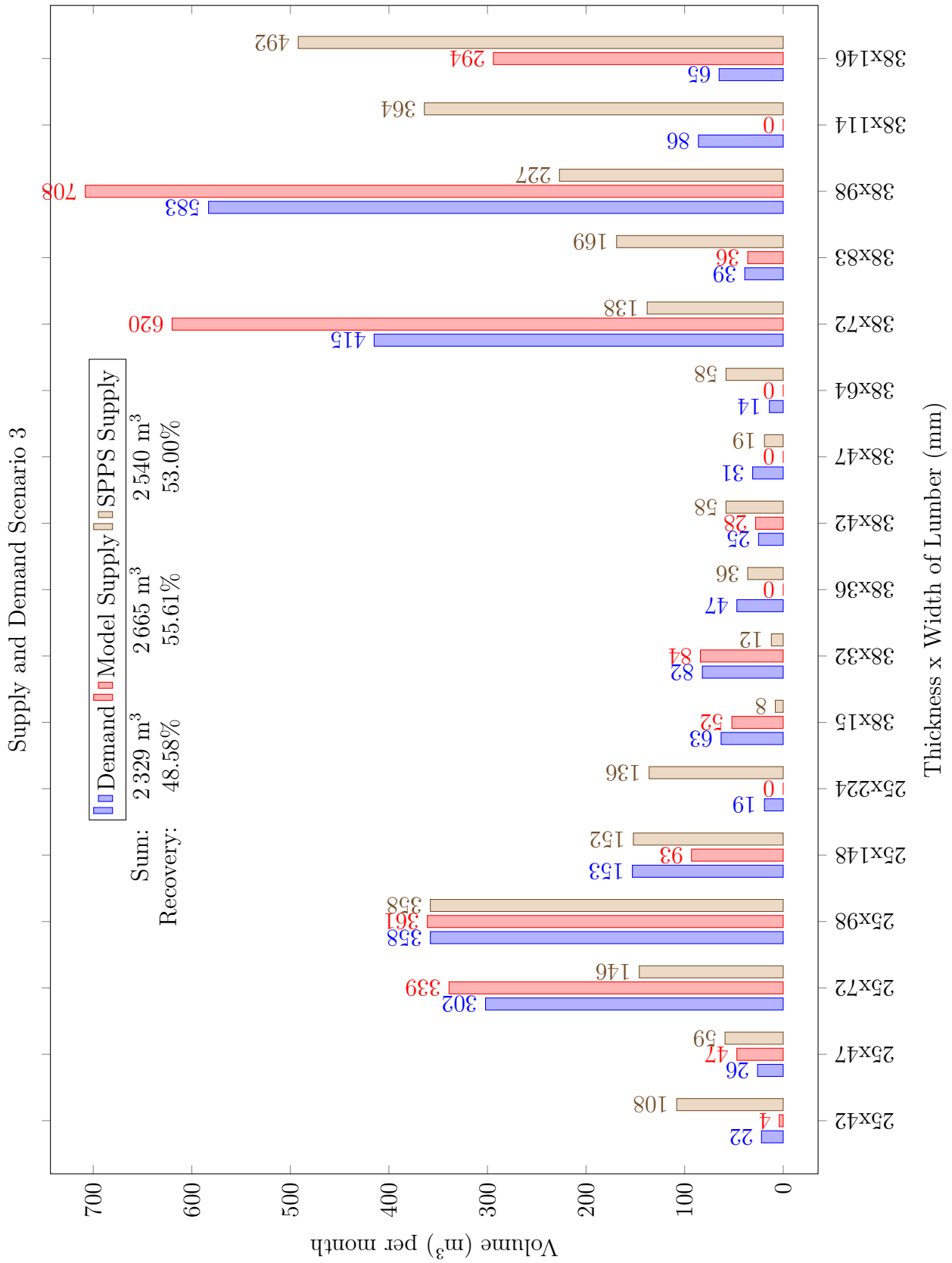


Figure 6.10: Supply versus demand results for the proposed model and the SPPS package.

**Table 6.5:** The SPSS's optimal decision variables for supply and demand scenario 1.

		Sawing pattern ( <i>c</i> ) assigned to log diameter class ( <i>d</i> )												
		1	2	3	4	5	6	7	8	9	10	11	12	
<i>d</i>	1													
	2	1			1				1		1		1	
	3			1		1				1				
	4		1						1					

**Table 6.6:** The proposed model's near optimal decision variables for supply and demand scenario 1. PBIL<sub>30-5-5</sub>

		Sawing pattern assigned to log diameter class ( $Y_{dc}$ )												
		1	2	3	4	5	6	7	8	9	10	11	12	
<i>d</i>	1			1					1	1				
	2	1												1
	3			1		1								
	4				1				1				1	

		Priority values for ripped thickness x width products ( $P_{tw}$ )											
		15	32	36	42	47	71	96	148				
<i>t</i>	25			1	64		64	64	64				
	38		100		80	1	68	64	56				

**Table 6.7:** The SPSS's optimal decision variables for supply and demand scenario 2.

		Sawing pattern (c) assigned to log diameter class (d)																				
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
d	c	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	1						1															
	2			1		1		1											1			
	3				1			1				1					1			1		
	4										1		1	1	1	1	1				1	

**Table 6.8:** The proposed model's near optimal decision variables for supply and demand scenario 2. PBIL<sub>30-5-5</sub>

		Sawing pattern assigned to log diameter class ( $Y_{dc}$ )																				
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
d	c	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	1										1	1										1
	2	1			1			1	1	1											1	
	3			1									1	1			1			1		
	4										1	1				1	1	1				

		Priority values for ripped thickness x width products ( $P_{tw}$ )														
		15	24	32	36	42	47	64	64	72	83	98	102	114	148	224
w	t	16	16	1			1		64		64	8				
25	38	1	32		104	2			112	112	112	108				

**Table 6.9:** The SPSS's optimal decision variables for supply and demand scenario 3.

		Sawing pattern (c) assigned to log diameter class (d)																				
d \ c	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
1	1	1				1			1				1	1	1	1	1					1
2				1			1	1			1											
3									1			1								1	1	
4																						

**Table 6.10:** The proposed model's near optimal decision variables for supply and demand scenario 3. PBIL<sub>50-5-5</sub>

		Sawing pattern assigned to log diameter class ( $Y_{dc}$ )																				
d \ c	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
1	1								1		1	1	1		1	1	1					1
2	1		1	1				1						1								
3							1															
4						1	1														1	1

		Priority values for ripped thickness x width products ( $P_{tw}$ )													
w \ t	15	32	36	42	47	64	72	83	98	114	148	224			
25				1	24		72		64			70	32		
38	96	88		80		16	96	84	96			92			

## 6.7 Conclusion

The proposed model outperformed the SPPS package in satisfying market demand better, namely at a lower production cost in this case. This was because the proposed model has the added decision variable of assigning priority values to the ripped products. This allowed maximisation of total value (affected by the priority values assigned) as opposed to just maximising volume, as is the case with the SPPS package.

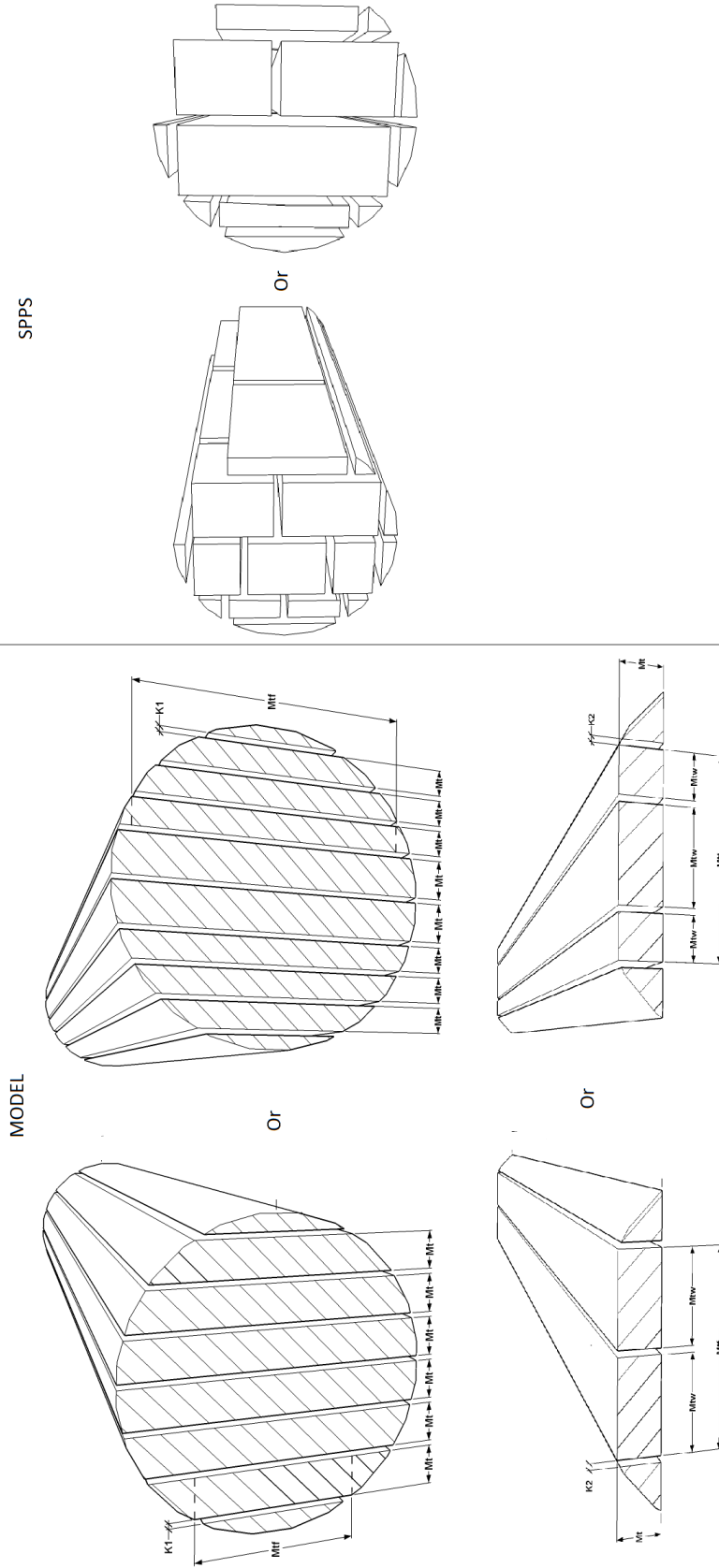


Figure 6.11: Proposed model versus SPPS's decisions.

# Chapter 7

## Conclusions and Recommendations

This chapter concludes the study and discusses its key findings. It includes how well the proposed model's optimal solution found compared to other related modelling software, namely the Sawmill Production Planning System (SPPS) package. The practical aspects and limitations of implementing such solutions is also discussed. Further possible improvements that would add more value to the model and its implementation are proposed and discussed in short detail. Finally the study concludes with summarising the work done to what was expected and its contribution.

### 7.1 Improved upon SPPS

The proposed model outperformed the SPPS package in all of the specific cost functions proposed. This means that the model satisfied demand better, namely at a lower production cost. This is attributed to the model having a different optimisation strategy, specifically for the ripping operation.

The SPPS package (and other related research and developments) typically determine optimal sawing and ripping decisions in isolation. For example the SPPS package determines optimal sawing patterns but has the subsequent ripping operation's strategy maximise volume, without any decision variable(s) to influence the ripping operation. This project's optimisation strategy used differs in that it forces the ripping machine to saw products according to part priority values assigned. This is then an extra decision variable that is optimally determined in combination with the previous operation's decision variables, namely the sawing patterns per logs. Determining these priority values (settings) as well as the combined sawing pattern decisions at the primary sawing operation, is what differentiates this project's model from most others. Having this extra decision variable (setting) helped the model satisfy



market demand better, specifically at a much lower raw material waste, over and under production cost as shown in Tables 6.2 to 6.4.

As mentioned the proposed model outperformed the SPPS package for the specific cost functions formulated. This however does not mean the model outperforms the SPPS package in general, especially with regard to the raw material cost, since both have slightly different optimising objectives. The SPPS's objective function represents the estimated profit, which it maximises instead of just minimising the three cost functions proposed in this study. Also it can take into account other constraints, such as production and kiln drying capacities. More results and comparisons would further substantiate the model's performance. However the model added great value by taking into account over production costs, while only an under production cost is used in the SPPS's objective function. This being said, there is a maximum volume demand constraint that can be adjusted in the SPPS package.

## 7.2 Implementing Solutions and a Possible DSS

The model's solutions were shown to the management of the case company and they were impressed with the outcomes. Despite this, it was felt that the model did not satisfy demand well enough, sometimes not supplying some products at all (Figures 6.8 to 6.10). Management felt that current practices, though not "optimal", were preferable. Three reasons are proposed why the model did not satisfy demand as well as would be expected by the management of the case company.

Firstly, the industry's ripping machine takes into account the subsequent chopping operation, which tries to optimally chop the lumber length-wise into demanded wood products. The proposed model does not take this into account, i.e. it keeps the length as a constant in the ripping operation's optimisation strategy (Equation 4.1.15). Adding the possible length of the product sawn into the maximising value Equation (4.1.15) would vary the mixture of products ripped. This is explained further in Section 7.3.4.

Secondly the model's ripping operation works with a flitch's maximum edged board width, which is maybe not a good assumption when mimicking the real world where a flitch has irregular width characteristics, due to taper and sweep of the log. Because of this, the industry tries to also recover side edged lumber within these irregular shaped flitches (illustrated later in Figure 7.2). Because of this extra side edged lumber option that can be recovered, different product widths can be fitted into the flitch. This would then yield a different mixture of products ripped, even with similar part priority values. This is further dis-

cussed in Section 7.3.5.

Thirdly and probably most importantly, the model is based on a long static time period of a month. In the industry, management and machine operators can change the sawing and ripping decisions on a daily (or even hourly) basis. Thus the decision maker would see that too much of one product is being produced and thus will change its priority value or even de-activate the product. It is felt that a more dynamic model would be of greater value. Such a model could represent the weekly or even daily changing processing environment as opposed to the model's current static monthly environment.

With that in mind an Excel spreadsheet was created where a user could change the two decision values, namely the sawing patterns and priority values for ripped products. This is according to a set number logs supplied and lumber demand expected, which the user can change manually. Similar to the model proposed (executed in AIMMS), SIMSAW data is used to determine the sawing operation's output and Excel's Solver is used to determine the ripping operation's output (modelled as a MIP problem). Excel's Solver is called consecutively (for each flitch ripped) and run via an external application, namely through the Visual Basic Application (VBA). Thus the project's PBIL algorithm tries to find good solutions that are used as a guideline (starting point) for a decision maker. The decision maker can then test these optimal solutions found and any other values in this Excel model. The development and application of such a Decision Support System (DSS) at the case company is still under investigation (Figure C.1 in Appendix C).

### 7.3 Further Modelling Improvements

As mentioned, further improvements to the model would be: including the supply of logs as a decision variable; expanding it to include the third operation (chopping) and its machine priority settings; more evaluation of the number and possible sawing patterns; and formulating more accurate cost equations - there is for example a greater cost involved with over production than just the warehouse cost used in this study.

Finally, the model and its results are from deterministic inputs. Techniques such as Monte Carlo (MC) simulation could be useful for the naturally varying log supply and fluctuating market demands. This would make the model and its results stochastic, adding more value to the decision maker. These improvements are all explained further in the following subsections.

### 7.3.1 Supply of Logs as another Decision Variable

In this project the number of logs supplied was a parameter for the developed optimisation model. What might be of great value is having this very important decision, namely how many logs need to be processed to meet demand as a decision variable to be optimised as well. In the SPPS package this is the case, where the user just has to define a minimum and maximum amount of logs allowed to be processed. This variable is very important since if a proposed solution does in fact translate into higher volume recovery then in fact less logs are needed to satisfy demand, and as mentioned raw material costs are by far the biggest cost factor.

### 7.3.2 Stochastic Model

At the moment the model is deterministic (using average values though variance occurs). The model could however utilise stochastic modelling for uncertainty. As can be seen in the fluctuating (box plots illustrate the variance) supply and demand values in Figures 5.5 to 5.10, the average value is not necessarily the most accurate deterministic value to use. Running the model under many different scenarios or parametric values (probably following some distribution) could be of further value to the decision maker.

### 7.3.3 Dynamic Model

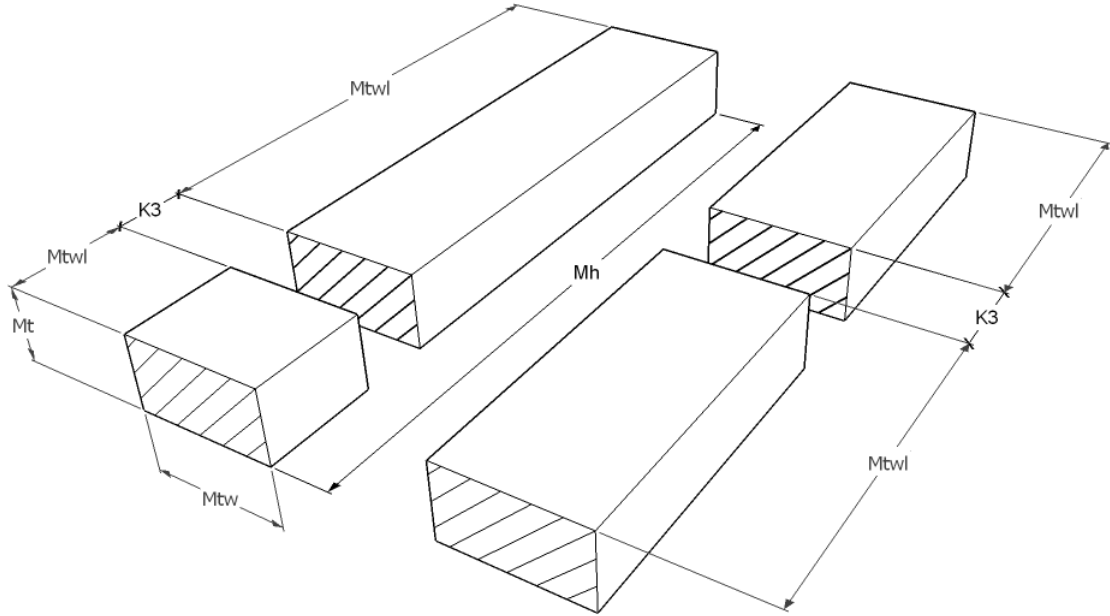
At the moment the model is static, since the variables do not represent sequences of decisions over multiple periods (Winston and Goldberg, 2004, Ch. 1). The decision variables are changed for a “snap shot” (one month), however in the ever changing production environment, these decisions are made weekly or even daily. To solve the optimal decisions for multiple periods a dynamic model could be used. Also dynamic modelling could model the inventory problem more accurately, since this project’s static model does not take into account how fast the overstocked products are demanded over time.

### 7.3.4 Operation 3: Optimally Chopping the Lumber

It would be of further value to incorporate subsequent operations in this project’s model. Specifically the operation where lumber is optimally chopped length-wise (Figure 7.1).

This operation resembles the ripping operation and could be formulated similarly, where the supply of lumber ( $S_{twh}$ ), which is the output of the ripping operation (2) becomes the input for this third operation.

Let:



**Figure 7.1:** Two possible length sawing decisions at operation 3, commonly referred to as chopping.

$$M_{twl} = \text{Length of product with set thickness } t, \quad (7.3.1)$$

width  $w$  and length  $l$

$$P_{twl} = \text{Priority value of products chopped with set thickness } t, \quad (7.3.2)$$

width  $w$  and length  $l$ .

$$Z_{twl} = \text{Number of products having set length } l \text{ chopped} \quad (7.3.3)$$

from lumber having thickness  $t$  and width  $w$ .

$$Z_{twl} \text{ is integer} \quad (7.3.4)$$

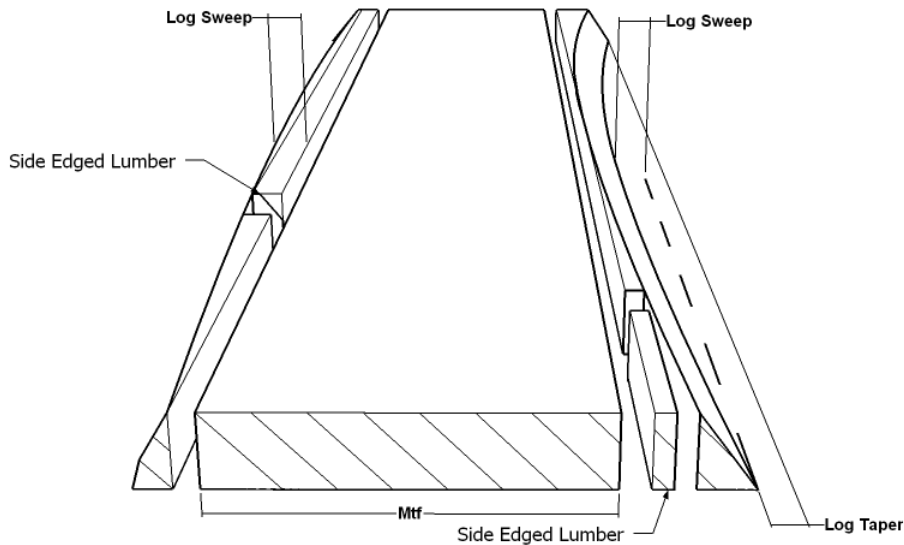
$$K_3 = \text{Kerf, thickness of saw blade, at operation 3} \quad (7.3.5)$$

Then the objective function for operation 3 is similarly formulated to operation 2 as:

$$\text{Max } B = \sum_t \sum_w \sum_l M_{twl} P_{twl} Z_{twl} (M_t M_{tw}) \quad (7.3.6)$$

### 7.3.5 Recovering Side Edged Products

In this project's model only the flitches' maximum board material ( $M_{tf}$ ) is ripped into lumber. However the ripping operation also recovers extra material from flitches, called side edged lumber as shown in Figure 7.2. This will influence the ripping decisions, namely what products will be ripped. It will also probably increase the volume of lumber recovered and can be included into future work.

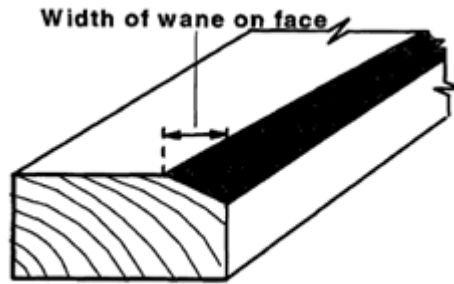


**Figure 7.2:** Side edged lumber recovered due to sweep and taper of logs.

### 7.3.6 Quality Tolerances: Bark/Wane and Knots

In this study no allowance was made for wane, which is the presence of bark or lack of wood on lumber edges (Figure 7.3). However in industry it can be tolerated on some products (Todoroki and Ronnqvist, 1997). Having this allowance bought into the model will influence the optimal decisions. Todoroki and Ronnqvist (1997) determined that when the allowance of wane was increased to 5 mm and 10 mm, flitch volume recovery increased by 2.4% and 4.6% respectively.

Also this project did not include quality aspects such as the presence of knots amongst others, which will influence the sawing decisions and results. Knots are shapes or points left on a piece of wood from where a stem or branch was growing from the tree.



**Figure 7.3:** Partial board product demonstrating wane (shaded region). Todoroki and Ronnqvist (1997) found that increasing the tolerance increased volume recovery significantly.

### 7.3.7 Objective Function(s)

As mentioned the three simple cost functions (raw material waste, over production and under production) do not represent all the factors which determine optimal sawing and ripping decisions. There are other cost functions such as, processing time costs that could be included.

Also, depending on the objectives or strategy of the wood manufacturer, some costs are of more importance than others. For example a company might have a policy of zero tolerance on under supplying (as this affects customer satisfaction) or reducing production as much as possible (which is in line with Theory of Constraints and Lean thinking). In such a case it could be of interest to add weights to each of these three cost functions. For example Equation 4.1.33 could become:

$$\text{TFC} = w_1\text{TRMWC} + w_2\text{TOPC} + w_3\text{TUPC} \quad (7.3.7)$$

where:

$w_{1,2,3}$  = weight parameters values that users can change for each of the three cost functions.

Besides the above mentioned cost “weighting” approach, multiple objective decision making might be of great value. For example two appropriate objectives could be the service rate and the associated production cost, which are usually inversely related.

## 7.4 Final Remarks

The model developed was able to link two important operations in the wood manufacturing chain that are usually optimised in isolation. Operations Research (OR) can be defined as *the art of giving bad answers to problems to which otherwise worse answers are given* (Saaty, 2004). The simulation and optimisation techniques helped the model give better answers compared to those offered by other optimisation software used in this case study. Using a different optimisation strategy, in this case the strategy on how to optimally rip (saw) flitch/board material into demanded lumber, can greatly influence the system and its expected results. Additional studies would further validate and substantiate this project's model.

## List of References

- Allpass, M. (2013). Correspondence. 5 November 2013. Email.  
Available at: [mandy@crickmay.co.za](mailto:mandy@crickmay.co.za)
- Baluja, S. (1994). Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning. *School of Computer Science, Carnegie Mellon University*.  
Available at: <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA282654>
- Baluja, S. (1996). Evolution of an artificial neural network based autonomous land vehicle controller. *Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 3, pp. 450–463.
- Bekker, J. (2012). Simulation of discrete event stochastic processes. MEng Course Notes. Quantitative Management 2. Department of Industrial Engineering. Stellenbosch University.
- Bekker, J. and Olivier, Y. (2008). Using the Population-Based Incremental Learning algorithm with computer simulation: Some applications. *South African Journal of Industrial Engineering*, vol. 19, no. May 2008, pp. 53–71.
- Brendenkamp, B.V., Upfold, S.J. and Southern African Institute of Forestry (2012). *South African Forestry Handbook*. 5th edn. Menlo Park, South Africa : Southern African Institute of Forestry.
- Buck, R. (2009). *Integrating the Least-Cost Grade-Mix Solver into ROMI*. MSc, Virginia Polytechnic Institute and State University.
- Buehlmann, U. (1998). *Understanding the relationship of lumber yield and cutting bill requirements: a statistical approach*. PhD diss., Virginia Polytechnic Institute and State University.
- Buehlmann, U. and Thomas, R. (2001 February). Lumber yield optimization software validation and performance review. *Robotics and Computer-Integrated Manufacturing*, vol. 17, no. 1-2, pp. 27–32. ISSN 07365845.
- Buehlmann, U., Thomas, R.E. and Zuo, X. (2011 August). Cost minimization through optimized raw material quality composition. *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 4, pp. 746–754. ISSN 07365845.



- Buehlmann, U., Wiedenbeck, J.K. and Kline, D.E. (2003). Effect of cutting bill requirements on lumber yield in a rip-first rough mill. *Wood and Fiber Science*, vol. 35, no. 2, pp. 187–200.
- Chang, S., Cooper, C. and Guddanti, S. (2005). Effects of the log's rotational orientation and the depth of the opening cut on the value of lumber produced in sawing hardwood logs. *Forest Products Journal*, vol. 55, no. 9838, pp. 49–56.
- Cothrell, L. and Higgins, D. (2003). Scanning Wood for Rip & Crosscut Saw. Optimizers, Inc. In: *Scan Tech 2003 International Conference*, pp. 71–80. Seattle, Washington, U.S.A.
- Crickmay and Associates (2013a). South African Lumber Index: August 2013. Tech. Rep. September, Crickmay: Supply Chain Evolution.  
Available at: [mandy@crickmay.co.za](mailto:mandy@crickmay.co.za)
- Crickmay and Associates (2013b). South African Lumber Index: September 2013. Tech. Rep. October, Crickmay: Supply Chain Evolution.  
Available at: [mandy@crickmay.co.za](mailto:mandy@crickmay.co.za)
- Crickmay, A., Hlengwa, M., Olivier, W. and Rusk, G. (2004 July). Productivity in the South African forestry industry over the past decade. *The Southern African Forestry Journal*, vol. 201, no. 1, pp. 1–2. ISSN 1029-5925.
- D'Amours, S., Rönnqvist, M. and Weintraub, A. (2008 November). Using Operational Research for Supply Chain Planning in the Forest Products Industry. *INFOR: Information Systems and Operational Research*, vol. 46, no. 4, pp. 265–281. ISSN 0315-5986.
- Dogan, C., McClain, T. and Wicklund, S. (1997). Simulation modeling and analysis of a hardwood sawmill. *Simulation Practice and Theory*, vol. 4869, no. 96.
- Du Plessis, J. (2010). *Investigation into the use of meta-heuristics in the optimisation of log positioning during sawmill processing*. MSc, Stellenbosch University.
- Erasmus, S. (1992). *A decision support system for the management of lumber mill production*. PhD, Stellenbosch University.
- Faaland, B. and Briggs, D. (1984). Log bucking and lumber manufacturing using dynamic programming. *Management Science*, vol. 30, no. 2, pp. 245–257.
- Fathi, Y. and Aksakalli, V. (2004 May). Heuristic methods for gang-rip saw arbor design. *European Journal of Operational Research*, vol. 154, no. 3, pp. 626–640. ISSN 03772217.
- Fathi, Y., Kegler, S. and Culbreth, C. (1996). A column generation procedure for gang-rip saw arbor design and scheduling. *International Journal of Production Research*, vol. 34, no. 2, pp. 313–327.
- Ferrar, R. and King, W. (1999). Rip first or cut first ? Use ' evolution ' to choose. *Wood Technology*, vol. 126, no. 7, pp. 32–33+.

- Gatchell, C. (1996). Designing a fixed-blade gang rip saw arbor with a pencil. *Forest products journal*, vol. 46, no. October 1995, pp. 37–40.
- Geerts, J. (1984). Mathematical solution for optimising the sawing pattern of a log given its dimensions and its defect core. *New Zealand Forest Service*, vol. 14, no. 1, pp. 124–134.
- HALCO Software Systems Ltd. (2006). The SAWSIM Sawmill Simulation Program. Available at: <http://www.halcosoftware.com/software/index.html>
- Hallock, H. and Lewis, D. (1971). Increasing Softwood Dimension Yield from Small Logs – Best Opening Face. Tech. Rep., Forest Products Lab Madison WIS.
- Hallock, H., Stern, A. and Lewis, D. (1976). Is There a “Best” Sawing Method. vol. No. FSRP-F.
- Hallock, H., Stern, A. and Lewis, D. (1979). *A look at centered vs. offset sawing*, vol. 321. 321st edn. Forest Products Lab Madison WIS.
- Hiller, F.S. and Lieberman, G.J. (2010). *Introduction to Operations Research*. 9th edn. McGraw-Hill.
- Kapp, S.B. (1997). *A feasibility study on the development of an integrated manufacturing planning and control system in the South African sawmill industry*. MSc, Stellenbosch University.
- Kapp, S.B., Price, C.S., Turner, P. and Vermaas, H.F. (1999). A feasibility study on the development of an integrated manufacturing planning and control system in the South African Sawmill Industry. *South African Forestry Journal*, , no. 184(1), pp. 80–87.
- Koch, S., König, S. and Wäscher, G. (2009 November). Integer linear programming for a cutting problem in the wood-processing industry: a case study. *International Transactions in Operational Research*, vol. 16, no. 6, pp. 715–726. ISSN 09696016.
- Laurens, H.C. (1972). *Normative production planning of a South African pine sawmill (a linear programming approach)*. MSc, Stellenbosch University.
- Lewis, D. (1985). *Sawmill simulation and the Best Opening Face system: A user’s guide*. USDA.
- Lindner, B., Visser, T. and Wessels, B. (2013). A model to optimise the linked sawing and ripping decisions in the South African pine wood industry. In: Willemse, E. (ed.), *42nd Annual Conference of the Operations Research Society of South Africa (ORSSA)*. Protea Hotel Technopark, Stellenbosch, South Africa.
- Lundahl, C. (2007). *Optimized processes in sawmills. Licentiate Thesis*. PhD, Lulea University of Technology.
- Maness, T., Ye, L. and Ristea, C. (2009 August). An Adaptive Optimization Method for Chop Saw Systems. *INFOR: Information Systems and Operational Research*, vol. 47, no. 3, pp. 203–213. ISSN 0315-5986.

- Maness, T.C. and Donald, W. (1994). The effect of log rotation on value recovery in chip and saw sawmills. *Wood and Fiber Science*, vol. 26, pp. 546–555.
- McAdoo, J.C. (1969). Computer simulation of small-log mill processing. *Forest Products Journal*, vol. 19, no. 4, pp. 34–35.
- Middleton, G. and Zhang, S. (2009). Characterizing the wood attributes of Canadian tree species: A thirty-year chronicle. *The Forestry Chronicle*, vol. 85.
- Murthy, P.R. (2007). *Operations Research*. New Age International, 2007. ISBN 8122420699, 9788122420692.
- Peter, R. and Bamping, J. (1962). Theoretical sawing of pine logs. *Forest Products Journal*, vol. 12, no. 11, pp. 549–557.
- Pinto, I. (2004). *Raw material characteristics of maritime pine (Pinus pinaster Ait.) and their influence on simulated sawing yield*. VTT Technical Research Centre of Finland.
- Price, C., Wessels, C.B. and Turner, P. (2002). Using modelling and integrated forestry and sawmill software systems to value the pruned log resource. In: *Connection between Forest Resources and Wood Quality: Modelling approaches and simulation software: Fourth Workshop*. Harrison Hot Springs, British Columbia, Canada.
- Reinders, M. (1989). *IDEAS, for integral logistics for centralized wood processing*. PhD, Agricultural University Wageningen, The Netherlands.
- Reinders, M. (1992). Cutting stock optimization and integral production planning for centralized wood processing. *Mathematical and Computer Modelling*, vol. 16, no. 1, pp. 37–55.
- Reinders, M. (1993). Tactical planning for a cutting stock system. *Journal of the Operational Research Society*, vol. 44, no. 7, pp. 645–657.
- Reinders, M. and Hendriks, T. (1989). Lumberproduction optimization. *European Journal of Operational Research*, vol. 42, pp. 243–253.
- Reynolds, H.W. (1970). Sawmill simulation: data instructions and computer programs. *Research Papers. Northeastern Forest Experiment Station*, vol. NE-152.
- Rönnqvist, M. (2003). Optimization in forestry. *Mathematical Programming*, vol. 284, pp. 267–284.
- Saaty, T.L. (2004). *Mathematical Methods of Operations Research*. DoverPublications. com.
- Singmin, M. (1978). *SIMSAW: A Simulation to evaluate the effect of sawing patterns on log recovery*. National Timber Research Institute, Council fo Scientific and Industrial Research (CSIR).

- Steele, P. and Wengert, E. (1987). Initial look at opportunities for optimizing lumber volume using BOF decisions. *Wood and Fiber Science*, vol. 19, no. 4, pp. 381–387.
- Thomas, R. (1996). Prioritizing parts from cutting bills when gang-ripping first. *Forest Products Journal*, vol. 46, no. 10, pp. 61–66.
- Thomas, R.E. and Buehlmann, U. (2007). Yield implications of global versus local process optimization in rip-first rough mills. *Forest products journal*.
- Todoroki, C. (1996). Developments of the sawing simulation software, AUTOSAW: linking wood properties, sawing, and lumber end-use. In: INRA (ed.), *Proceedings of the 2nd Workshop on Connection Between Silviculture and Wood Quality through Modelling Approaches and Simulation Softwares*, pp. 241–247. Nancy, South Africa.
- Todoroki, C. and Ronnqvist, E. (1997). Secondary log breakdown optimization with dynamic programming. *Journal of the Operational Research Society*, vol. 48, no. 5, pp. 471–478.
- Todoroki, C. and Rönqvist, E. (1999). Combined primary and secondary log breakdown optimisation. *Journal of the Operational Research Society*, vol. 50, no. 3, pp. 219–229.
- Todoroki, C. and Ronnqvist, M. (2001). Log Sawing Optimisation Directed by Market Demand. *New Zealand Journal of Forestry*, vol. 45, no. February, pp. 29–33.
- Todoroki, C. and Rönqvist, M. (2002). Dynamic control of timber production at a sawmill with log sawing optimization. *Scandinavian Journal of Forest Research*, vol. 17, no. 1, pp. 79–89.  
Available at: <http://www.tandfonline.com/doi/abs/10.1080/028275802317221118>
- Turner, B. (2010). Optimising the Manufacturing Process ... from Log Procurement through to Timber Sales. In: *Wood Supply Chain Optimisation*, pp. 1–13. Rotorua, New Zealand.  
Available at: [http://www.halcosoftware.com/pdfs/optimize\\_your\\_optimizers.pdf](http://www.halcosoftware.com/pdfs/optimize_your_optimizers.pdf)
- Turner, B. and Rapoport, A. (2010). Optimize Your Production Mix ... Optimize Your Optimizers. In: *Woodtech 2010 Conference*, March, pp. 1–16. Portland, Oregon.  
Available at: [http://www.halcosoftware.com/pdfs/optimize\\_your\\_optimizers.pdf](http://www.halcosoftware.com/pdfs/optimize_your_optimizers.pdf)
- Usenius, A. (1996). Optimizing the activities in the wood conversion chain from forest to the end-users. In: Nepveu (ed.), *Proceedings of Second workshop in Connection between Silviculture and wood quality through modelling approaches and simulation softwares*, pp. 214–219. INRA Nancy, Kruger National Park, South Africa.

- Usenius, A. (1999). Wood conversion chain optimisation. In: Nepveu, G. (ed.), *Connection between Forest Resources and Wood Quality: Modelling approaches and simulation software: Third Workshop*, pp. 542–548. INRA-Nancy, La Londeles-Maures, France.
- Usenius, A., Song, T. and Nepveu, G. (1996). Relating the product specification to the value yield in the wood conversion. *IUFRO WP S*, vol. 5, pp. 01–04.
- Van Zyl, F. (2011). *Determining the optimal log position during primary breakdown using internal wood scanning techniques and meta-heuristic algorithms*. MScEng Thesis, Stellenbosch University.
- Vuorilehto, J. (2001). *Size control of sawn timber by optical means in breakdown saw machines*. Helsinki University of Technology. ISBN 9512256622.
- Weining Group (). Gang rip-saws and Optimizing rip-saws for maximum yield. Available at: [http://www.weinig.com/C1256FAF0043EEBF/vwContentByKey/W26F7G5Z056ALPADE/\protect\T1\textdollarFILE/Prospekt\\_ProfiRip\\_Serie\\_E.pdf](http://www.weinig.com/C1256FAF0043EEBF/vwContentByKey/W26F7G5Z056ALPADE/\protect\T1\textdollarFILE/Prospekt_ProfiRip_Serie_E.pdf)
- Weiss, J.M. and Thomas, R.E. (2005). *ROMI-3 : Rough-Mill Simulator Version 3.0 : User's Guide*. US Department of Agriculture, Forest Service, Northeastern Research Station.
- Wessels, C. (2009 Aprila). A model to determine the theoretical maximum feed speed of a frame saw. *Southern Forests: a Journal of Forest Science*, vol. 71, no. 1, pp. 31–36. ISSN 2070-2620.
- Wessels, C., de V du Plessis, J. and Smit, N. (2011 January). Cant sawing log positioning in a sawmill: searching for an optimal solution. *Southern Forests: a Journal of Forest Science*, vol. 73, no. 1, pp. 15–22. ISSN 2070-2620.
- Wessels, C., Dell, M.P. and Price, C.S. (2001). SIMSAW 6 User Manual. Tech. Rep., CSIR, Durban, South Africa.
- Wessels, C., Price, C., Turner, P. and Dell, M. (2006). Integrating harvesting and sawmill operations using an optimized sawmill production planning system. *Proceedings International Preceion Forestry Symposium, Stellenbosch, South Africa, IUFRO, ISBN 0-7972-1121-7, pp117. Vol. 118*.
- Wessels, C.B. (2009b). Cant sawing log positioning optimization: A simulation study. *Forest Products Journal*, vol. 59, no. 4, pp. 17–22.
- Wiedenbeck, J. (2001). Deciding between crosscut vs. rip-first processing. *Wood and Wood Products, Août 2001*, , no. 2, pp. 1–6.
- Winston, W.L. and Goldberg, J.B. (2004). *Operations Research: Applications and Algorithms*. 4th edn. Thomson/Brooks/Cole Belmont.
- Ye, L. and Maness, T. (2006). A real-time goal-seeking model for optimizing chop saw systems. *Forest Products Journal*, vol. 56, no. 11-12, pp. 73–78.

Zuo, X. (2003). *Improving lumber cut-up manufacturing efficiency using optimization methods*. PhD dissertation, North Carolina State University.

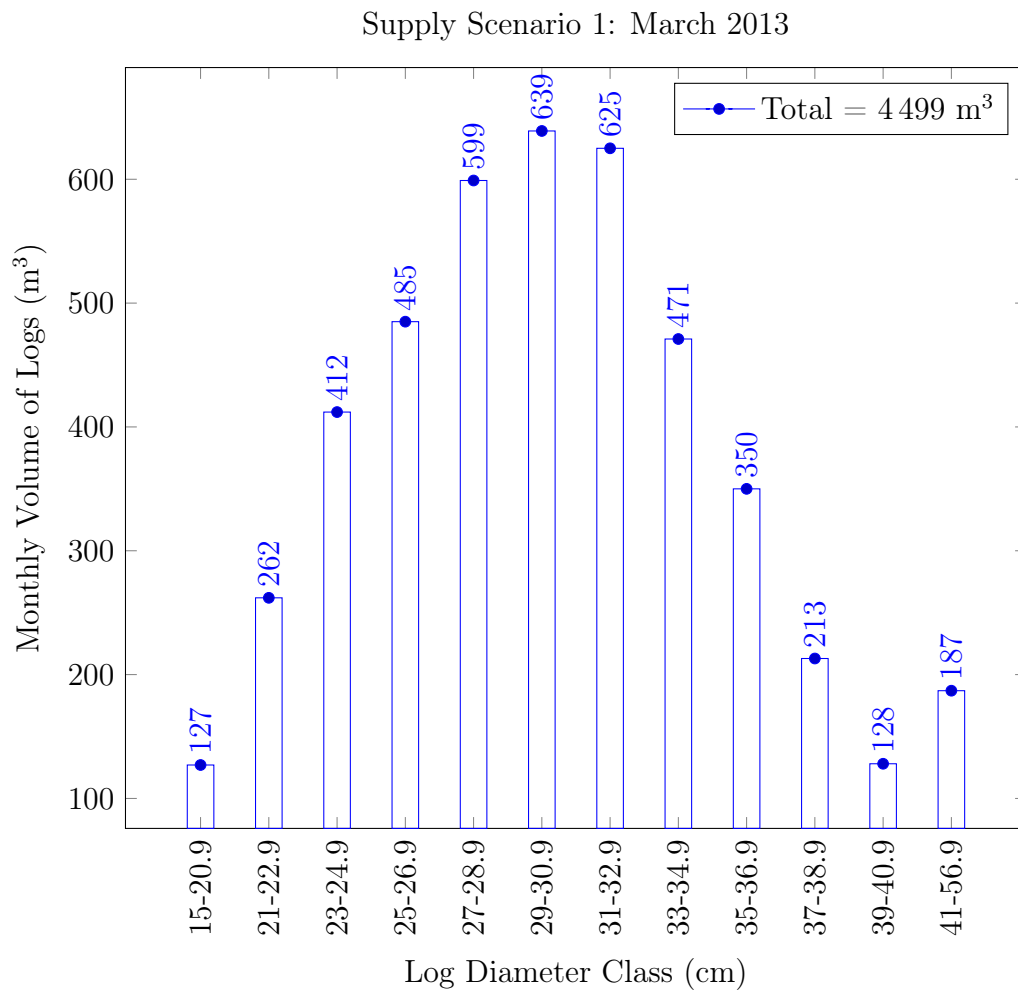
# Appendices

## Appendix A

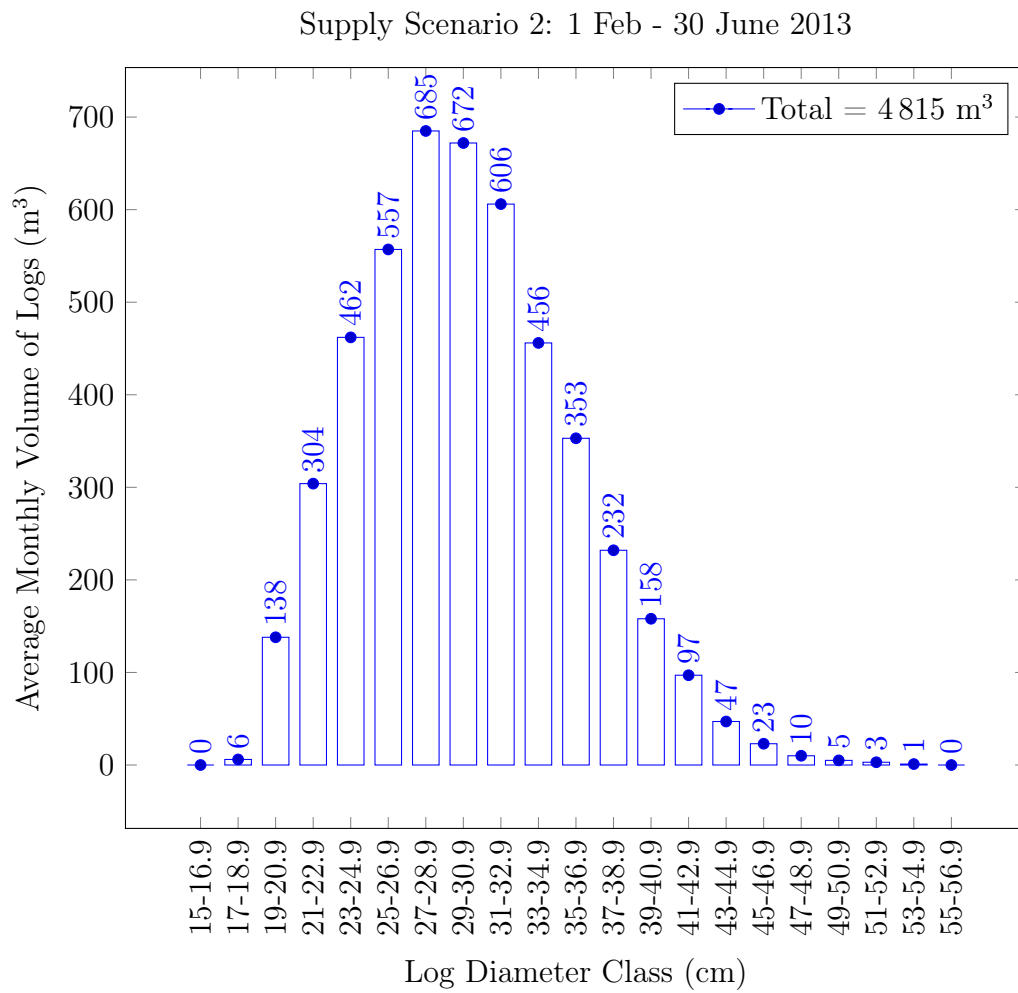
# Volume of Logs Supplied for Different Scenarios

Using Huber's log volume Equation (3.2.3) and the data of the number of small end diameter logs supplied in Section 5.4 and the fact that all logs are supplied in 3 meters lengths, the average monthly log volume per log diameter class was appropriately calculated and is illustrated in the figures below.

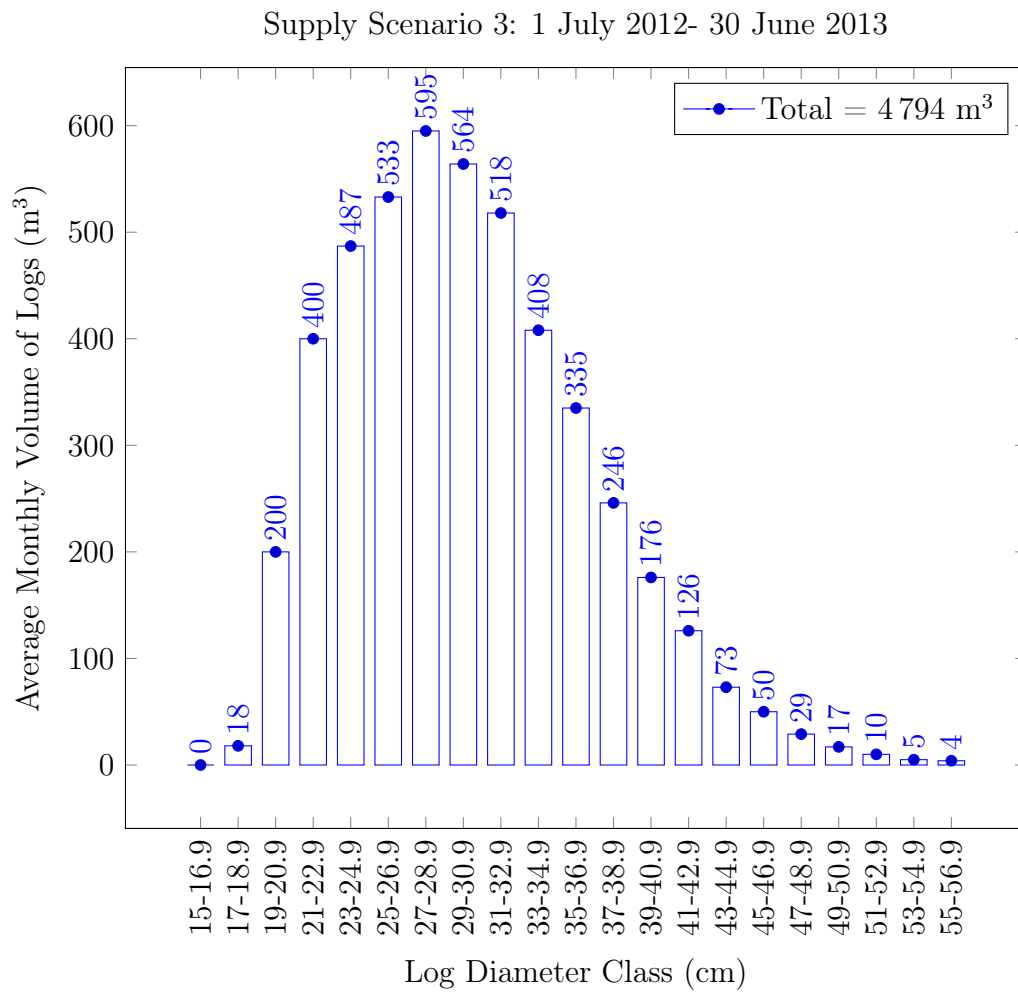




**Figure A.1:** Average monthly volume of logs supplied to the case study sawmill



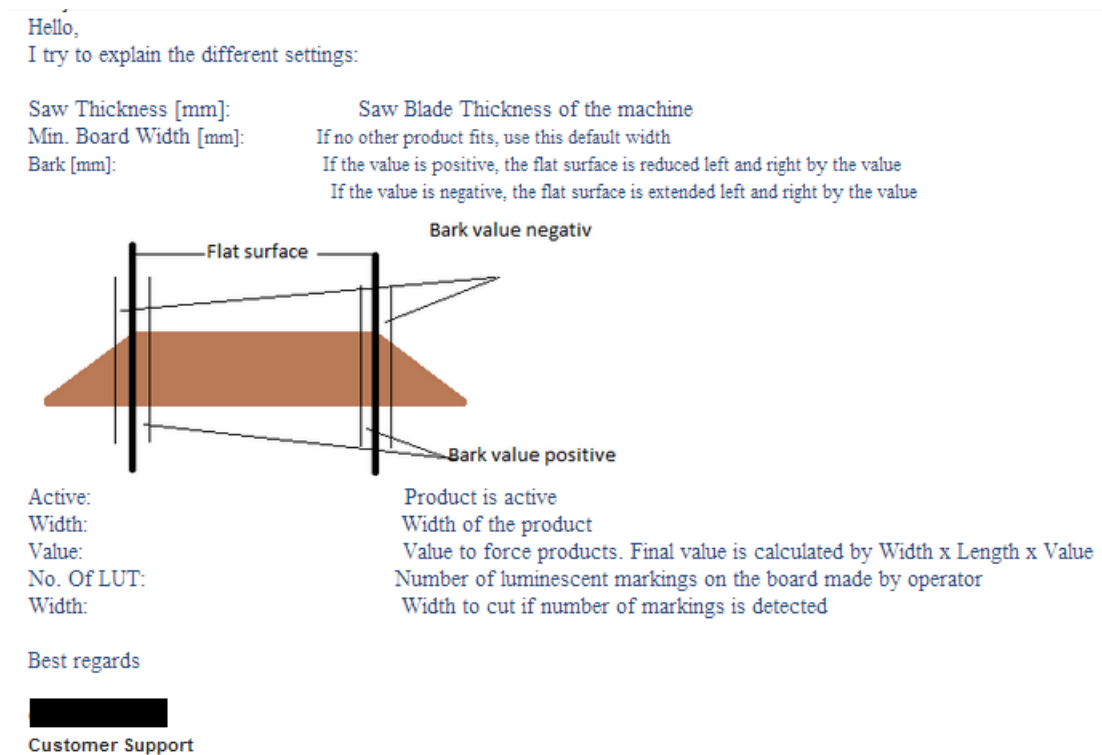
**Figure A.2:** Average monthly volume of logs supplied to the case study sawmill



**Figure A.3:** Average monthly volume of logs supplied to the case study sawmill

## Appendix B

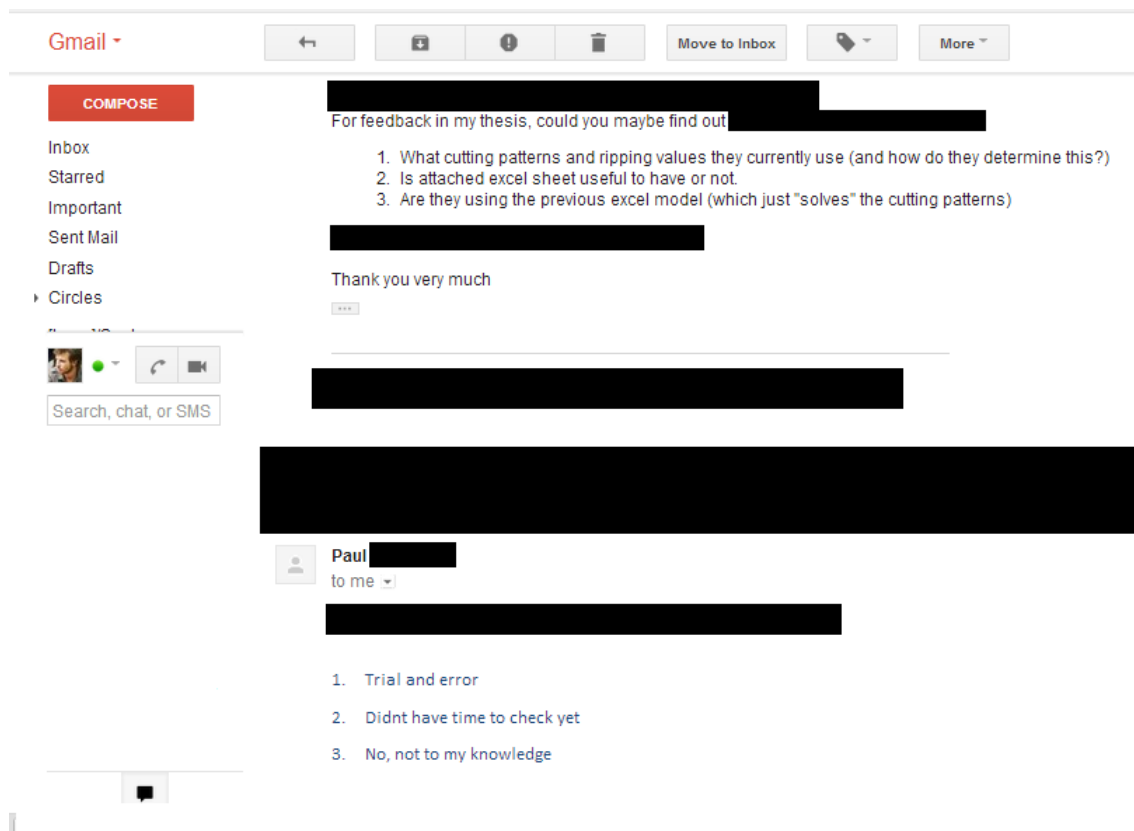
# Ripping Machine's Optimisation Strategy for Case Company



**Figure B.1:** Ripping machine's optimisation strategy for case company. Email correspondence via case company and operating machine's customer support. Note some information has been removed as too keep anonymity of case company and machine supplier.

# Appendix C

## Decision Making for Case Company

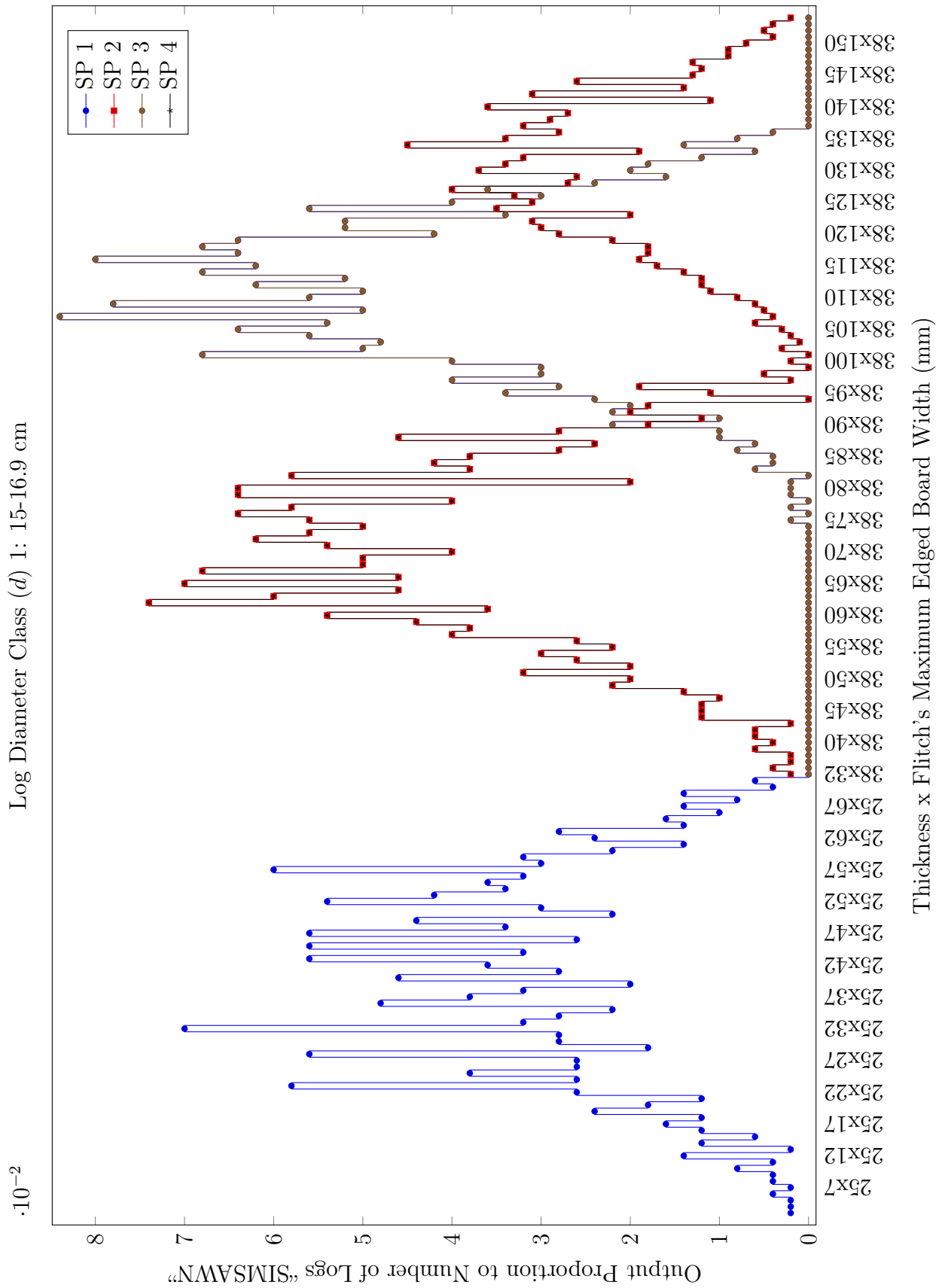


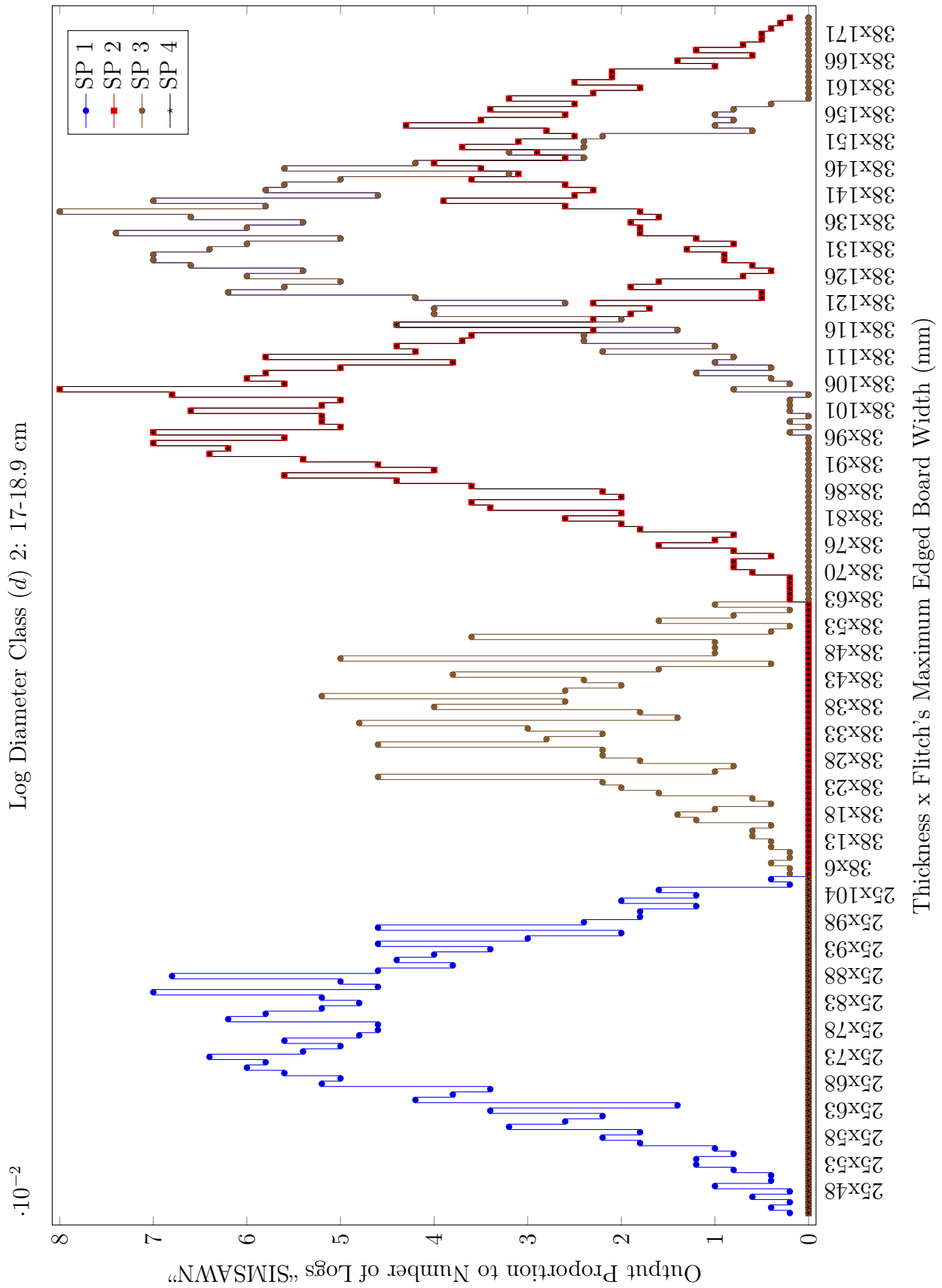
**Figure C.1:** Decision making for case company. Correspondence via management of case company. Note some information has been removed as too keep anonymity of case company.

## Appendix D

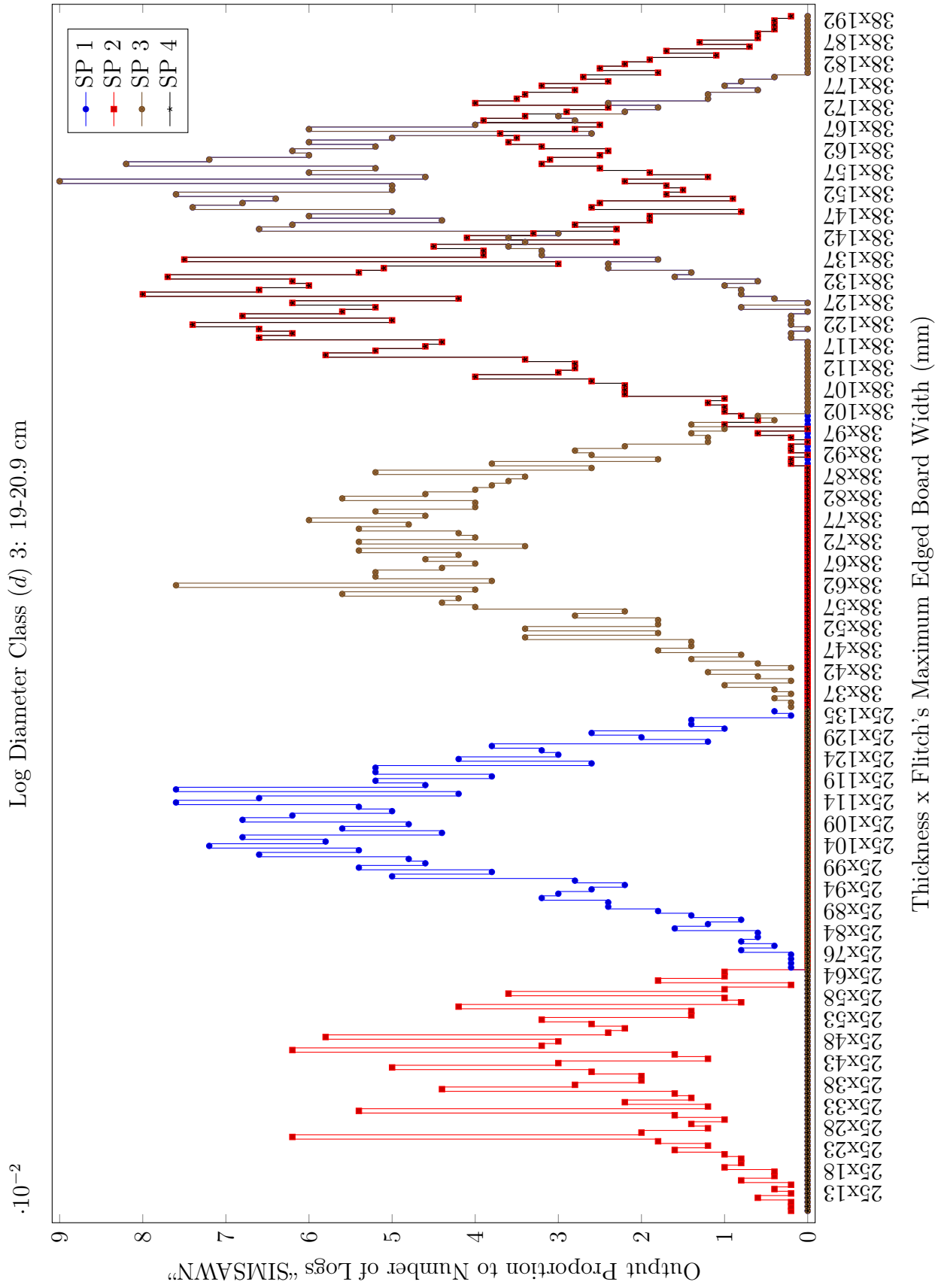
# Flitches' Maximum Board Width Simulated

For each log diameter class ( $d$ ) a 1 000 logs with varying taper (distributed normally: 95 % within 8-11 mm/m) and sweep (distributed normally: 95 % within 0-15 mm/m) were virtually sawed in SIMSAW. This was through each of the 4 possible sawing patterns (Figure 5.1) for Scenarios 2 and 3. This was done to determine the sawing operation's output, namely flitches with thickness  $t$  and width  $f$  for each possible Sawing Patterns (SPs) to choose from ( $c$ ). This represents the  $SIM_{dctf}$  variable in Equation 4.1.7.

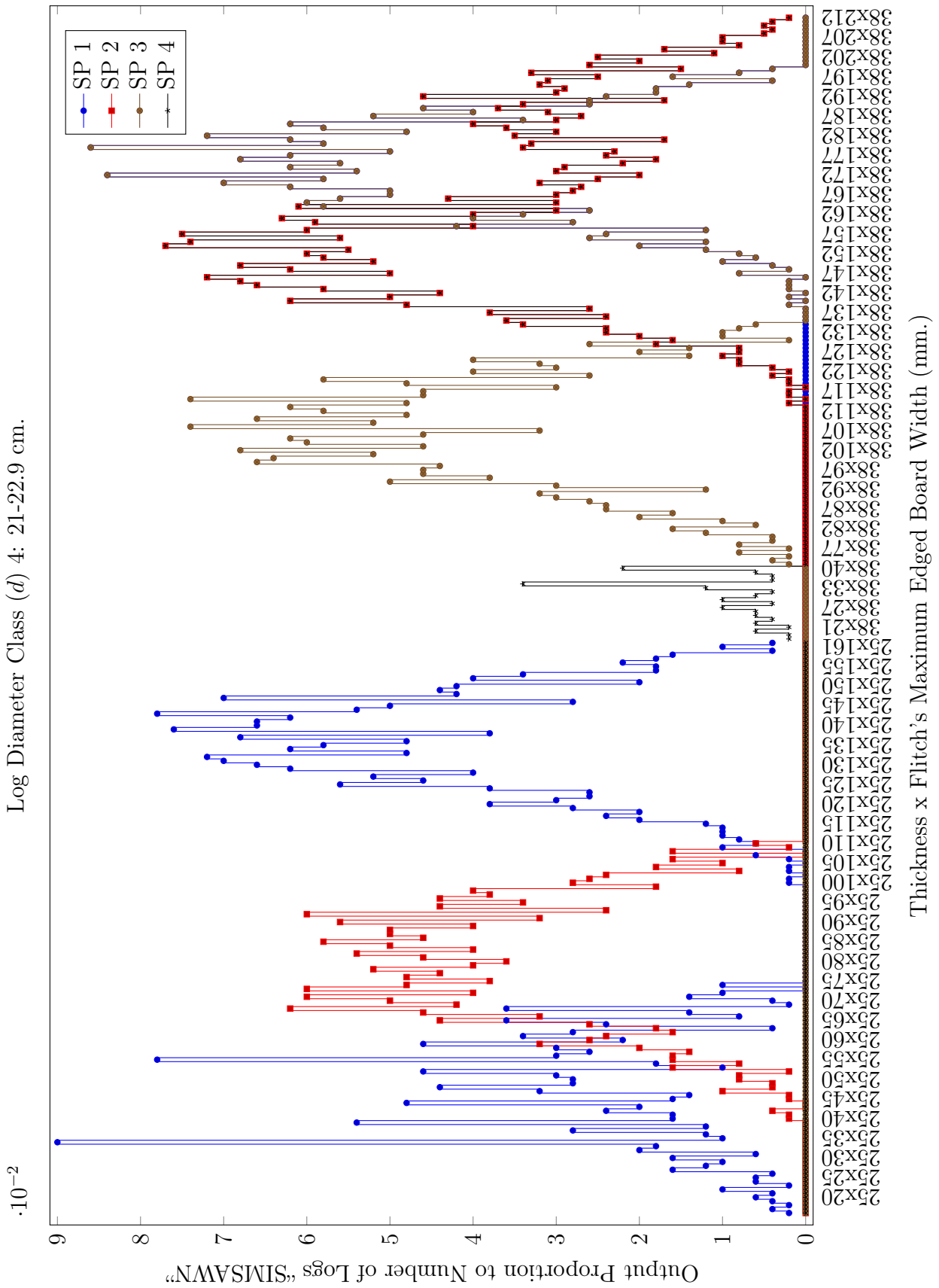


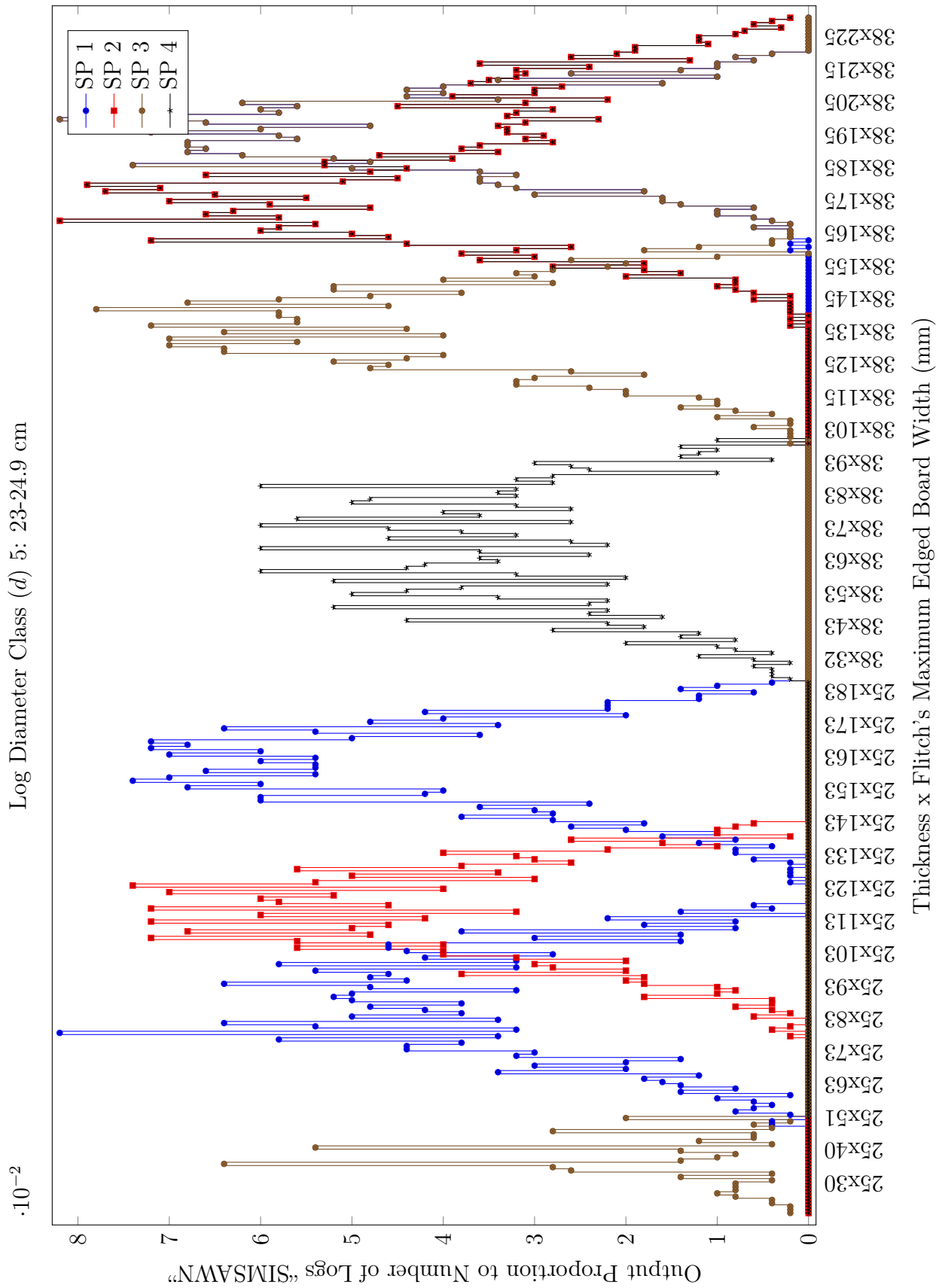


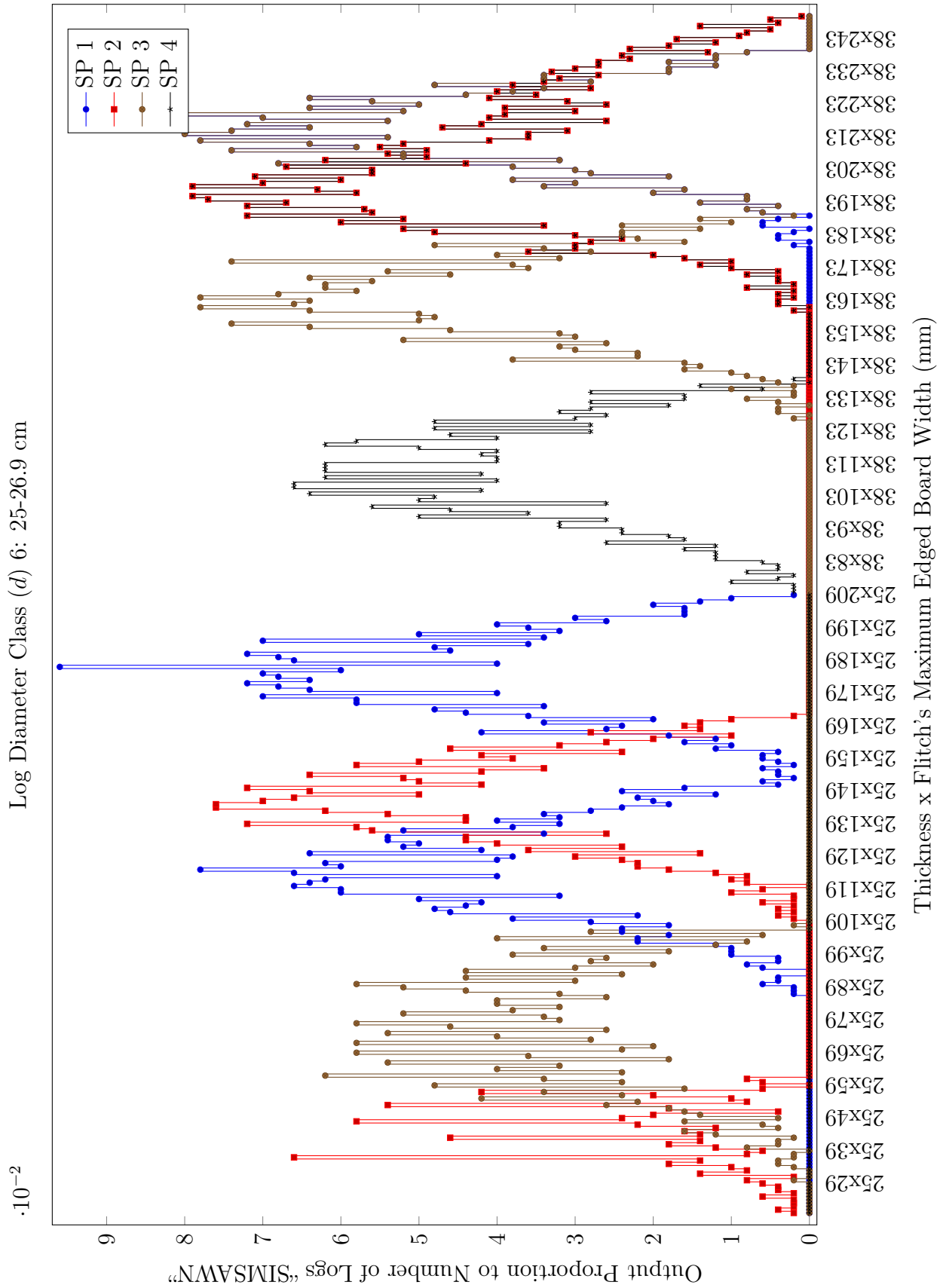


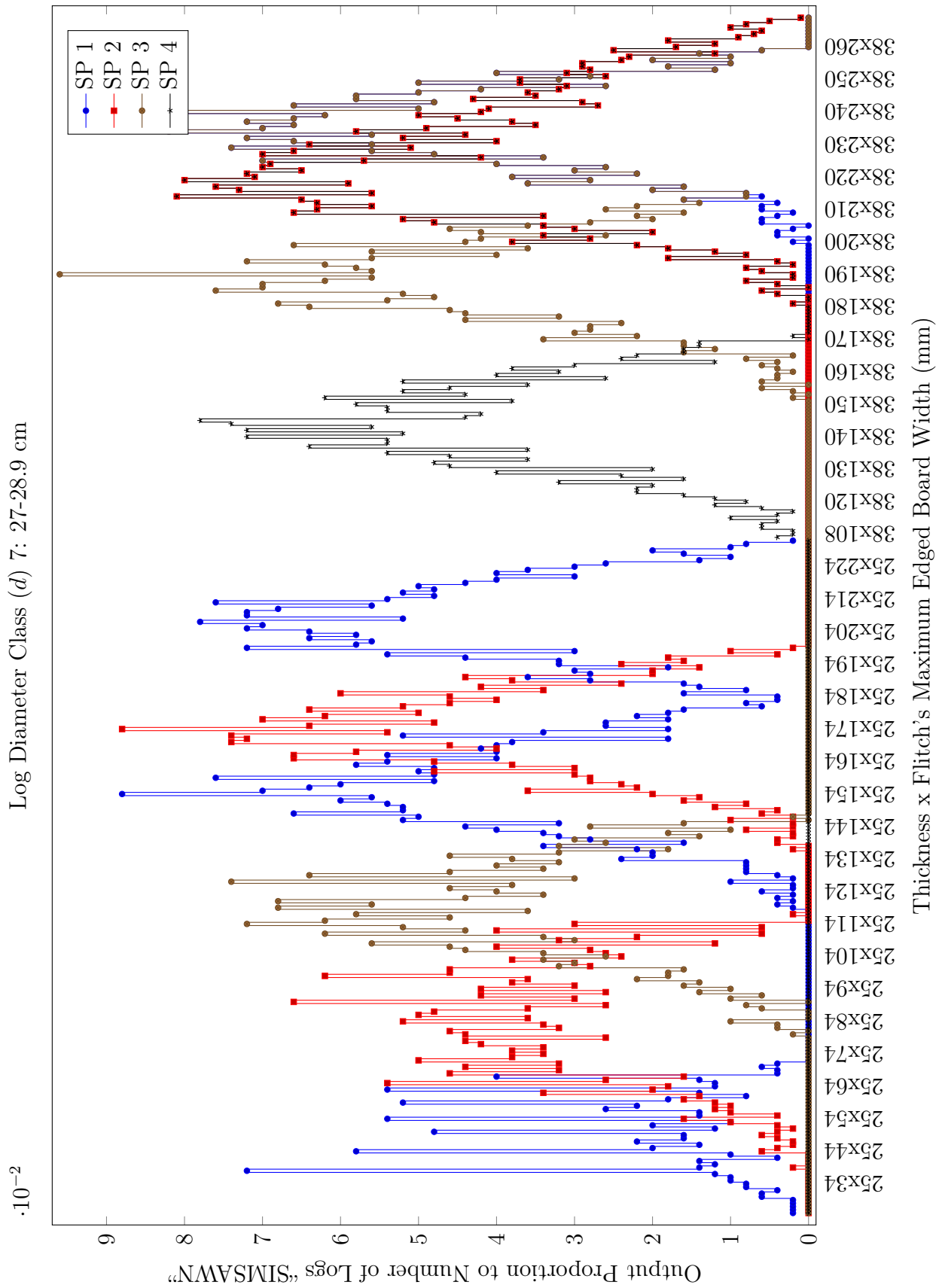


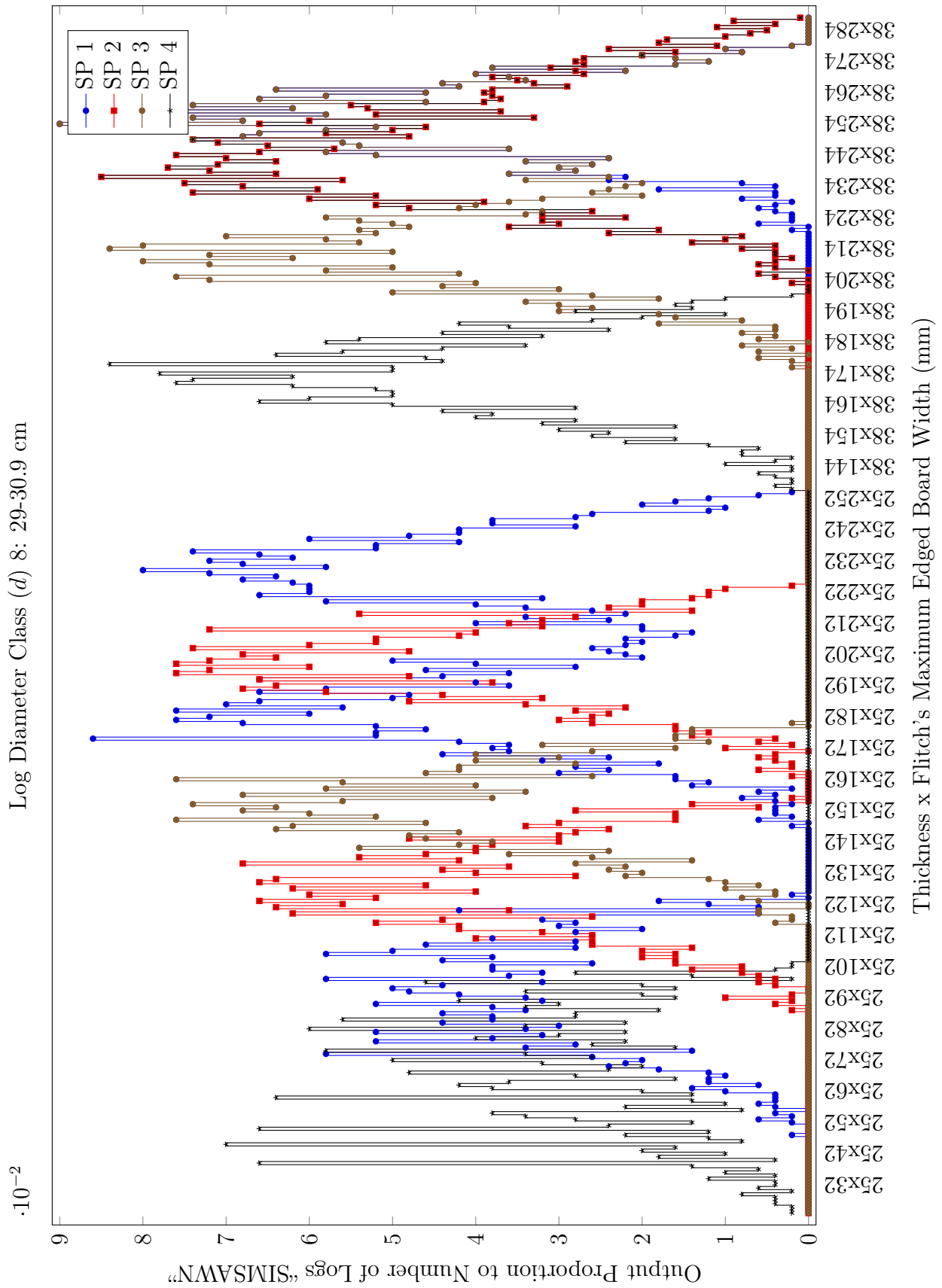
Log Diameter Class (d) 4: 21-22.9 cm.

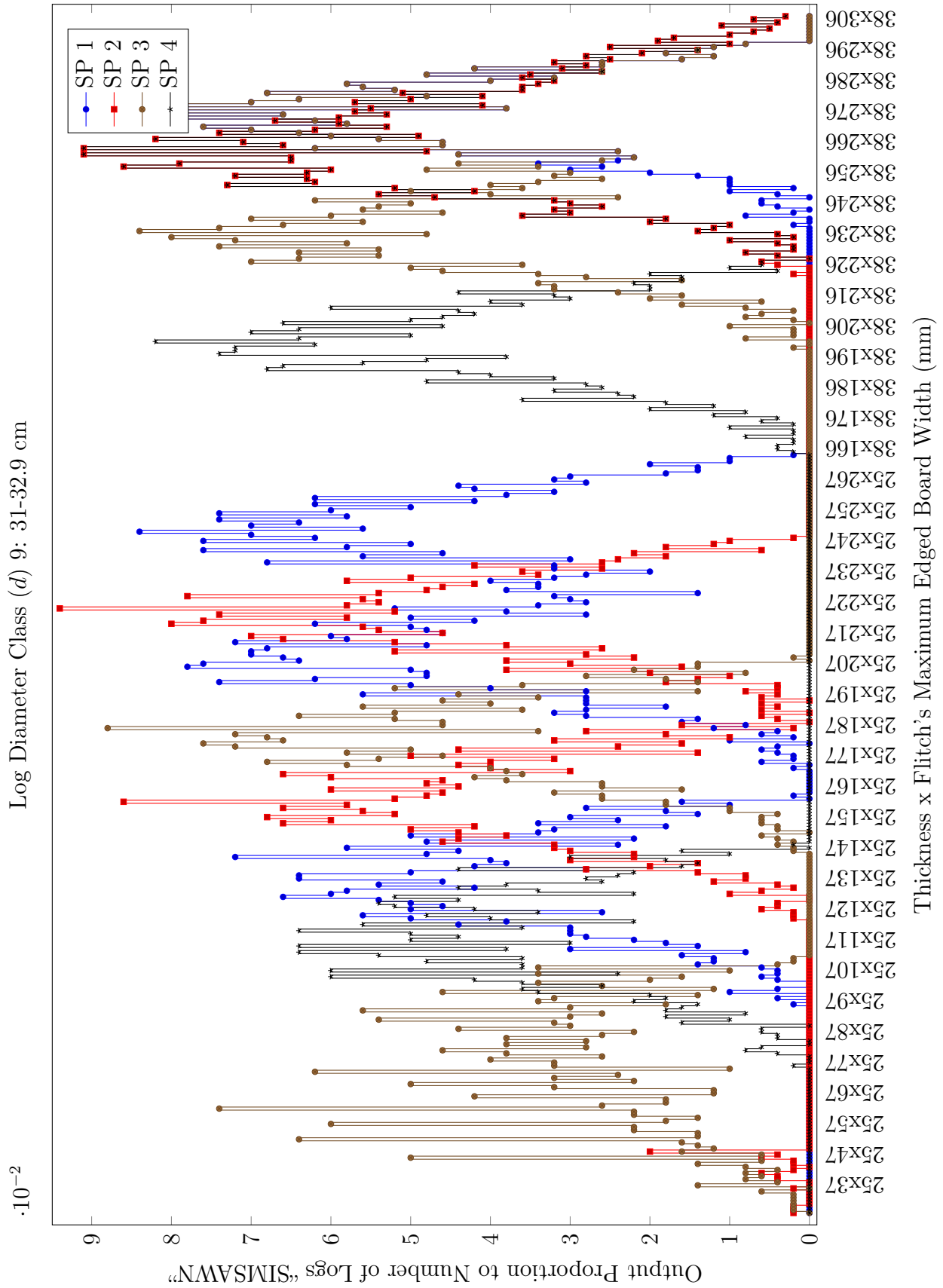


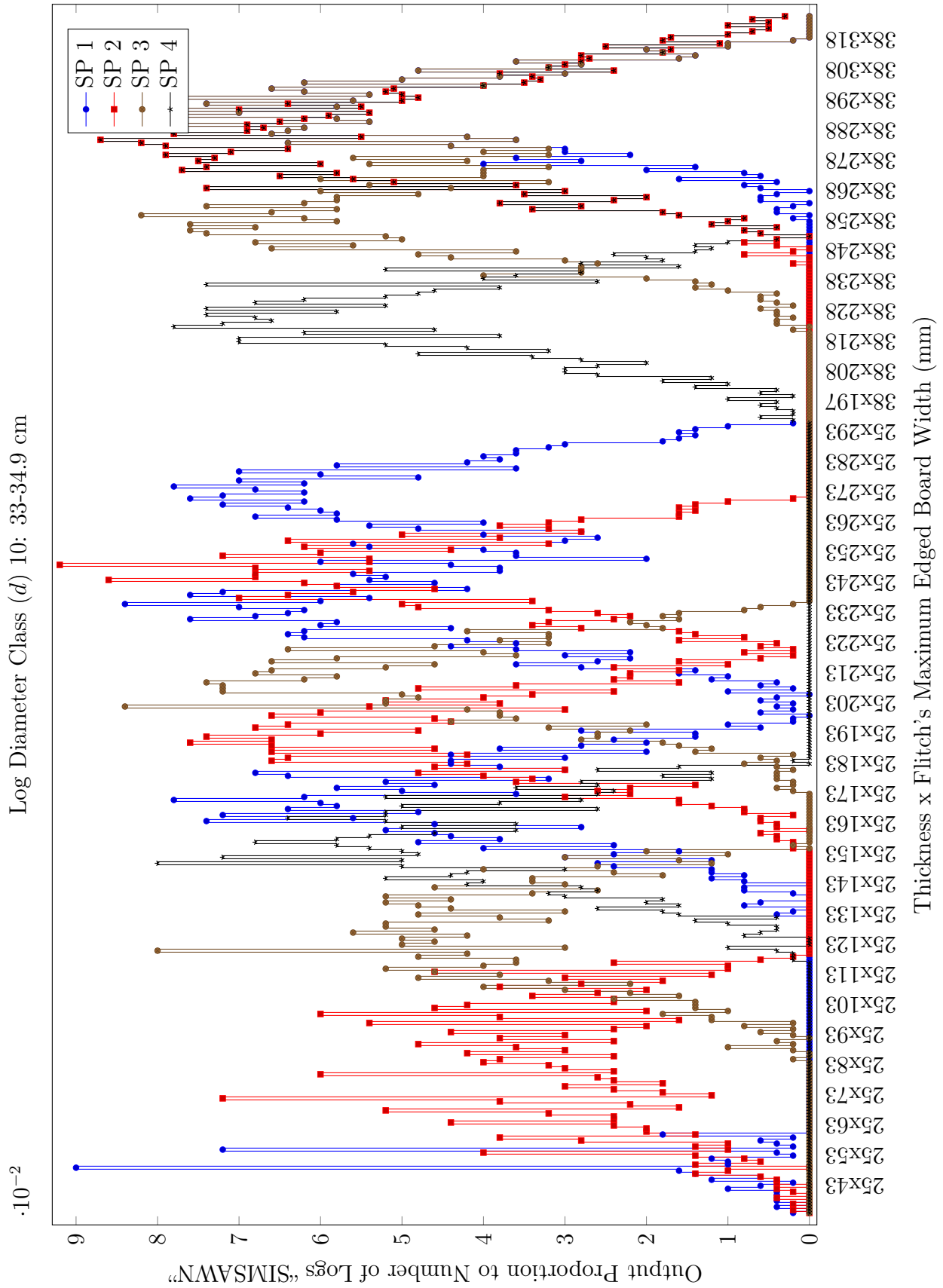






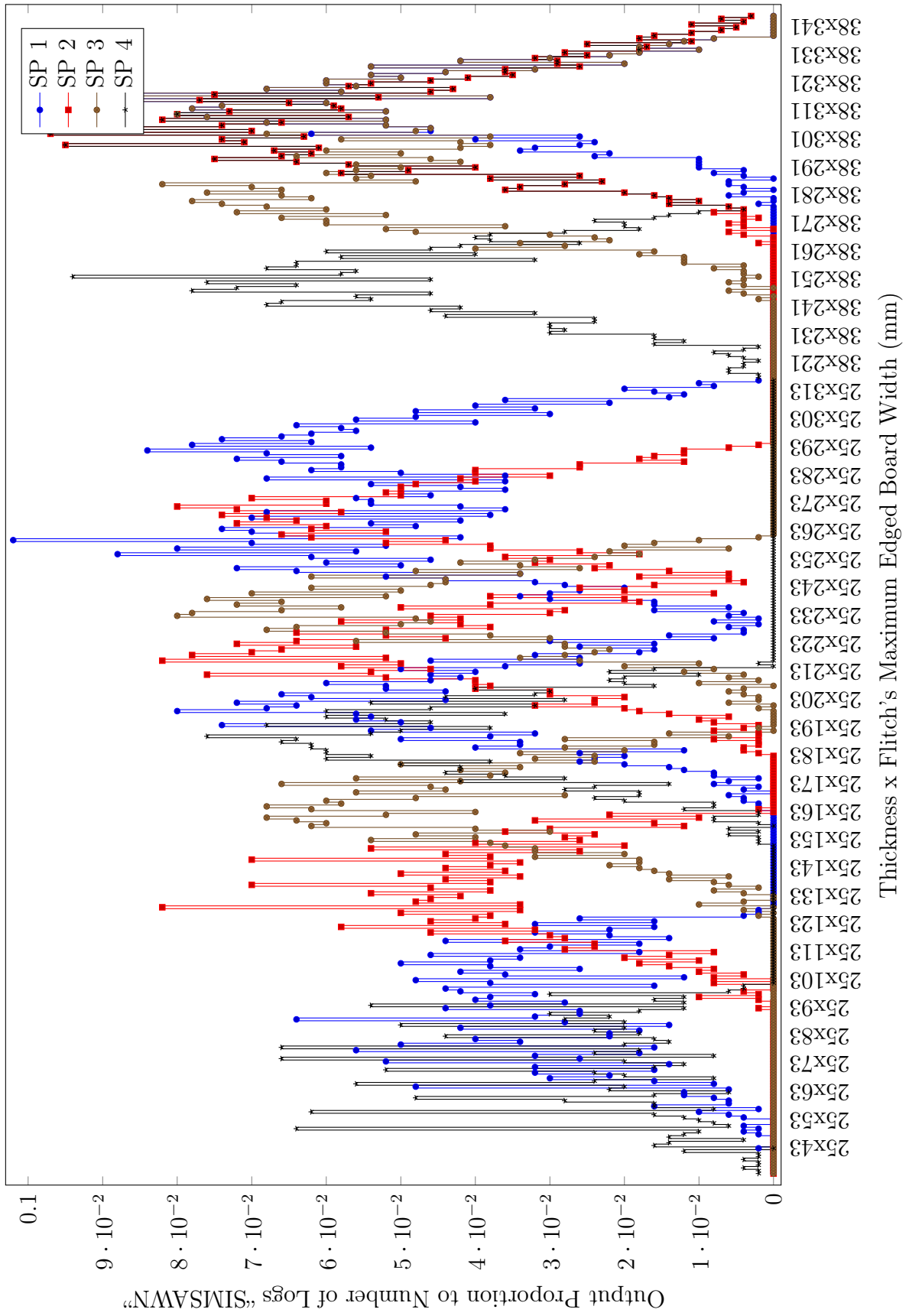




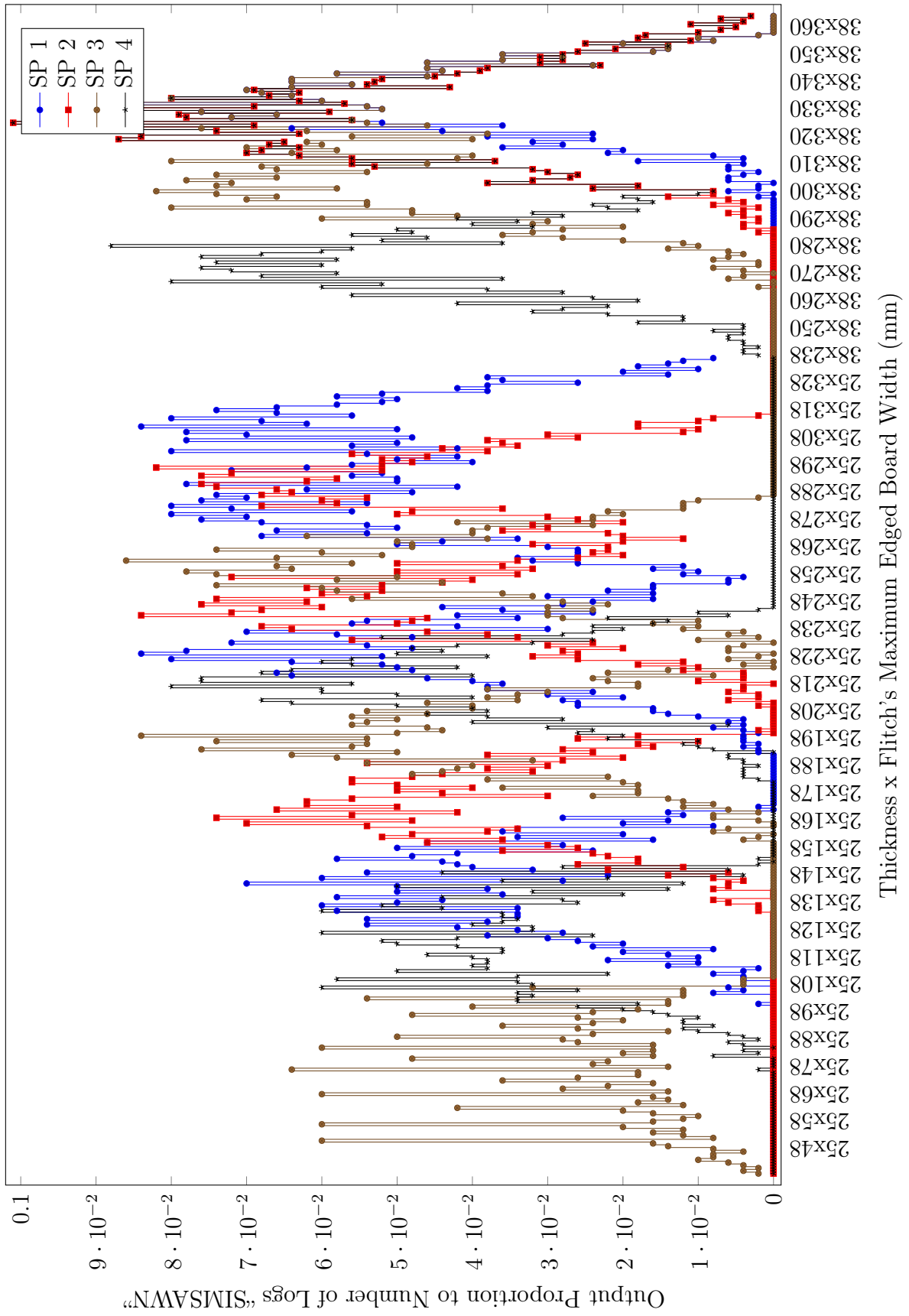




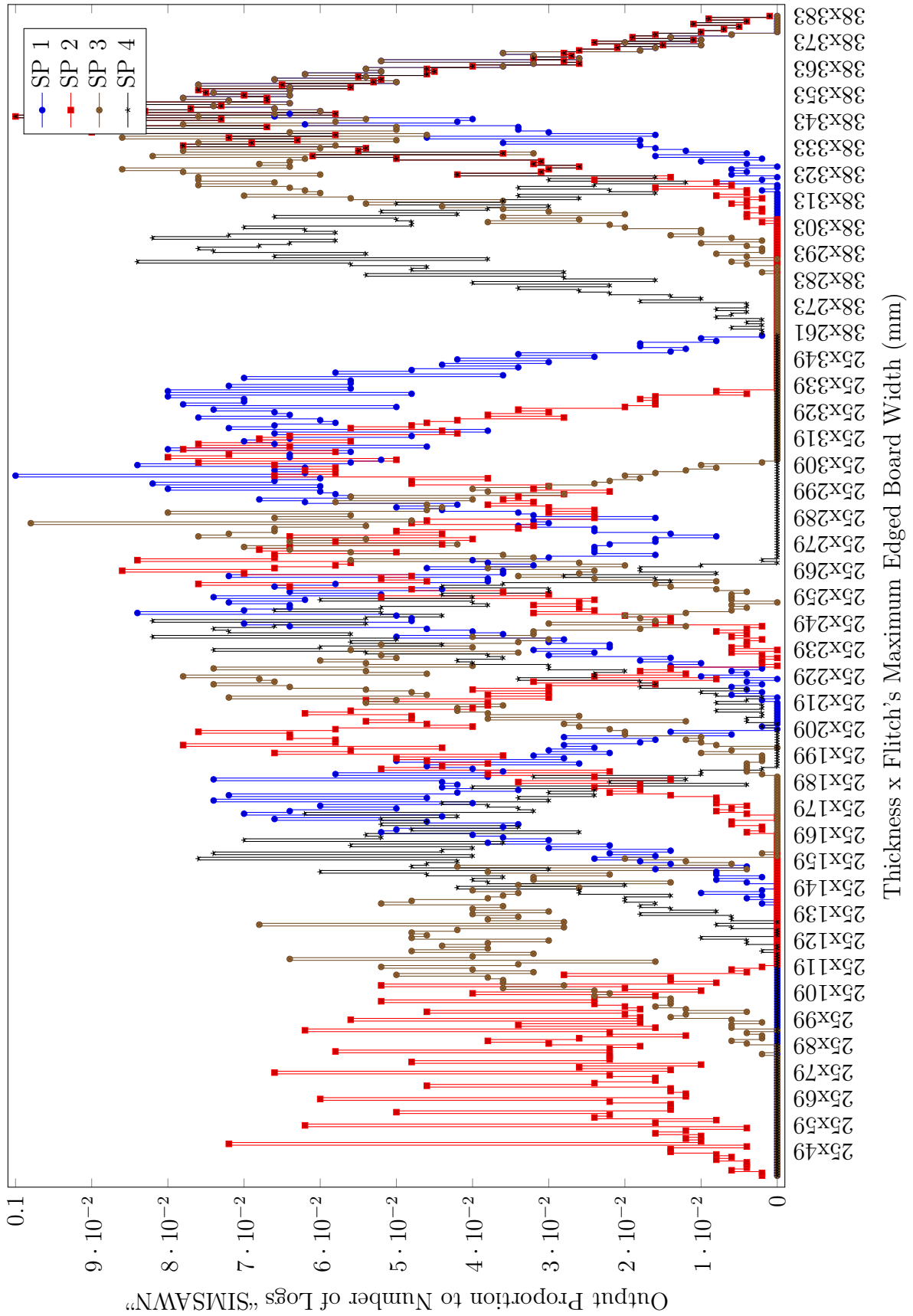
Log Diameter Class (d) 11: 35-36.9 cm



Log Diameter Class ( $d$ ) 12: 37-38.9 cm.

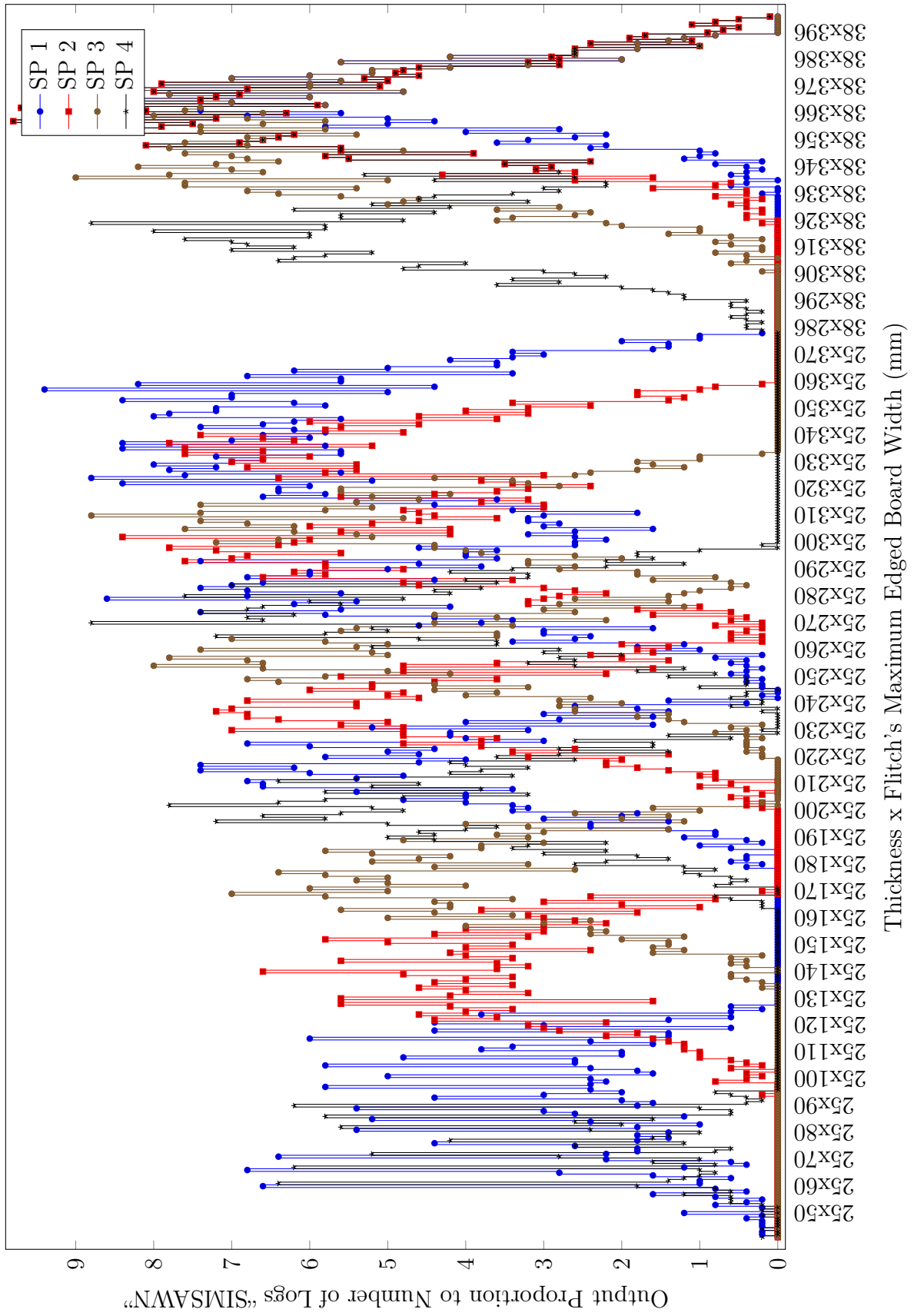


Log Diameter Class (*d*) 13: 39-40.9 cm

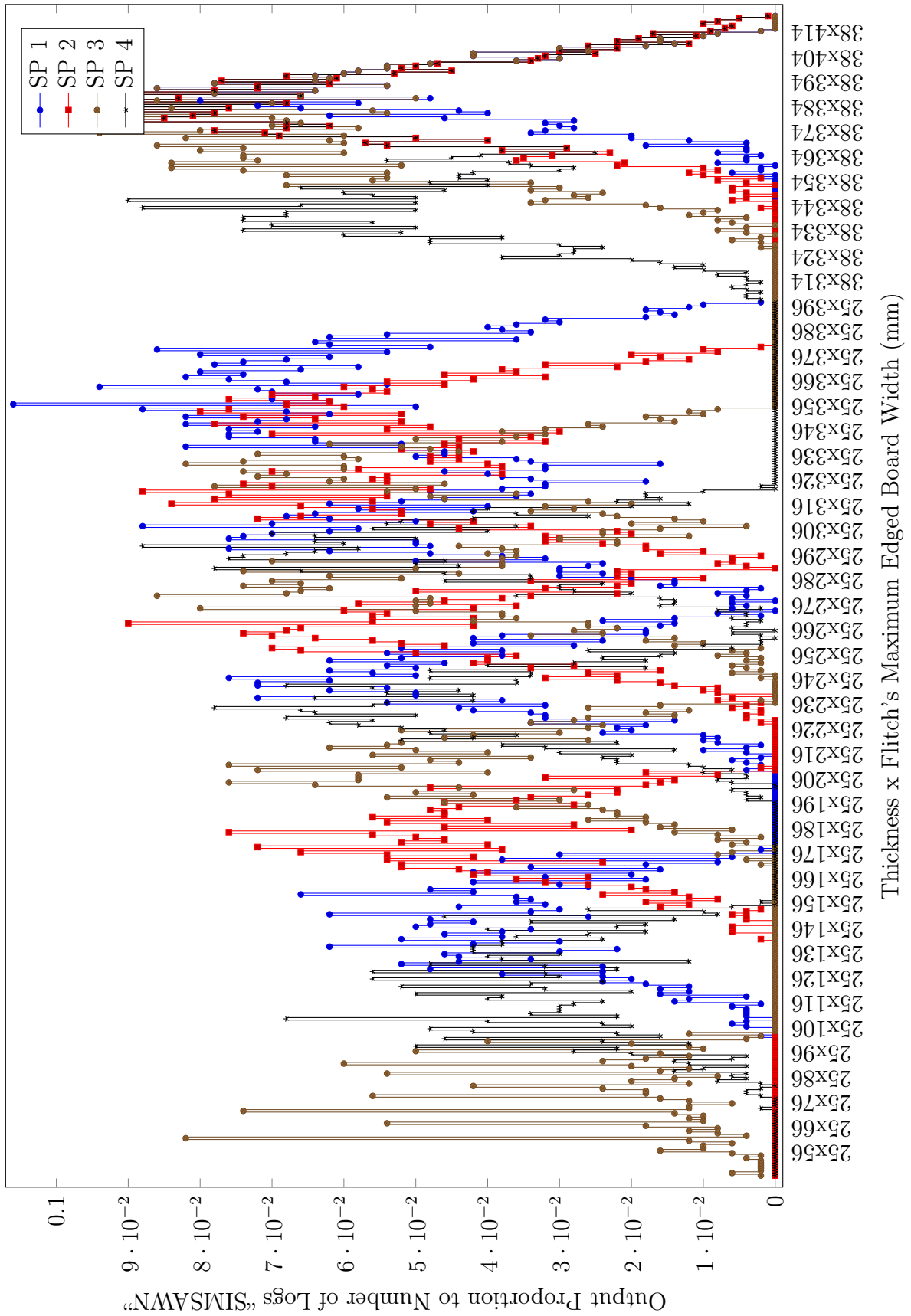


Log Diameter Class (*d*) 14: 41-42.9 cm

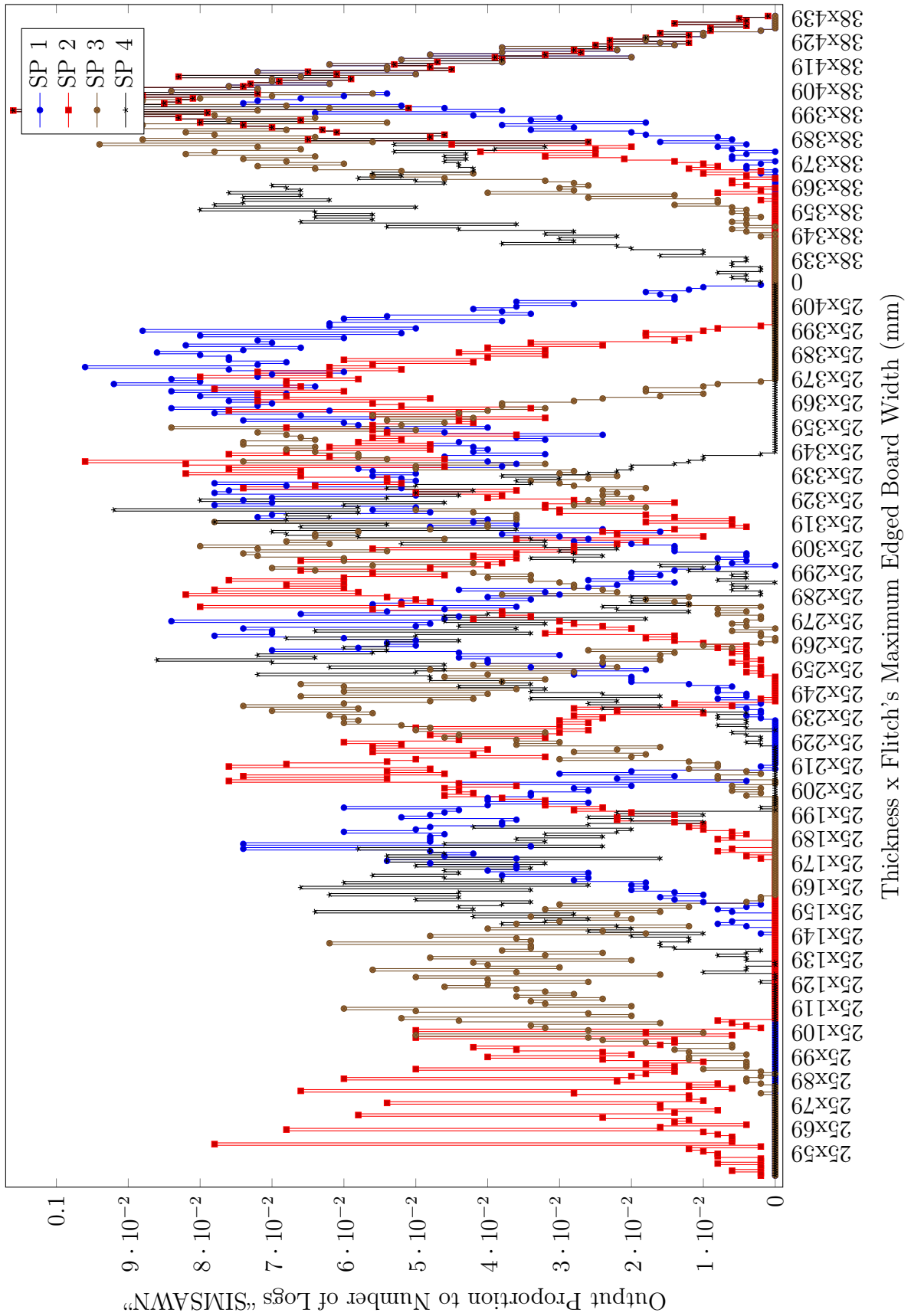
$\cdot 10^{-2}$



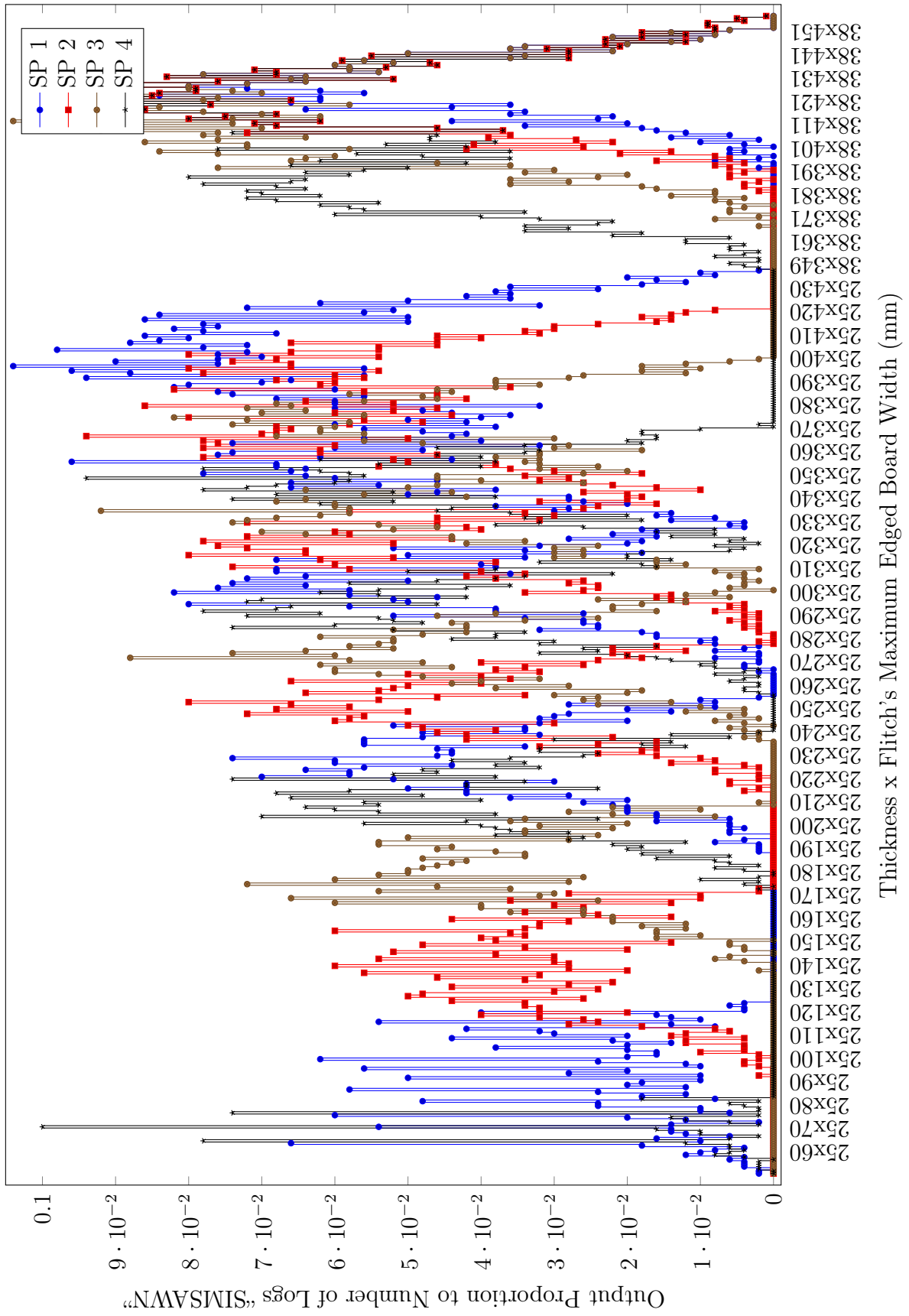
Log Diameter Class (d) 15: 43-44.9 cm



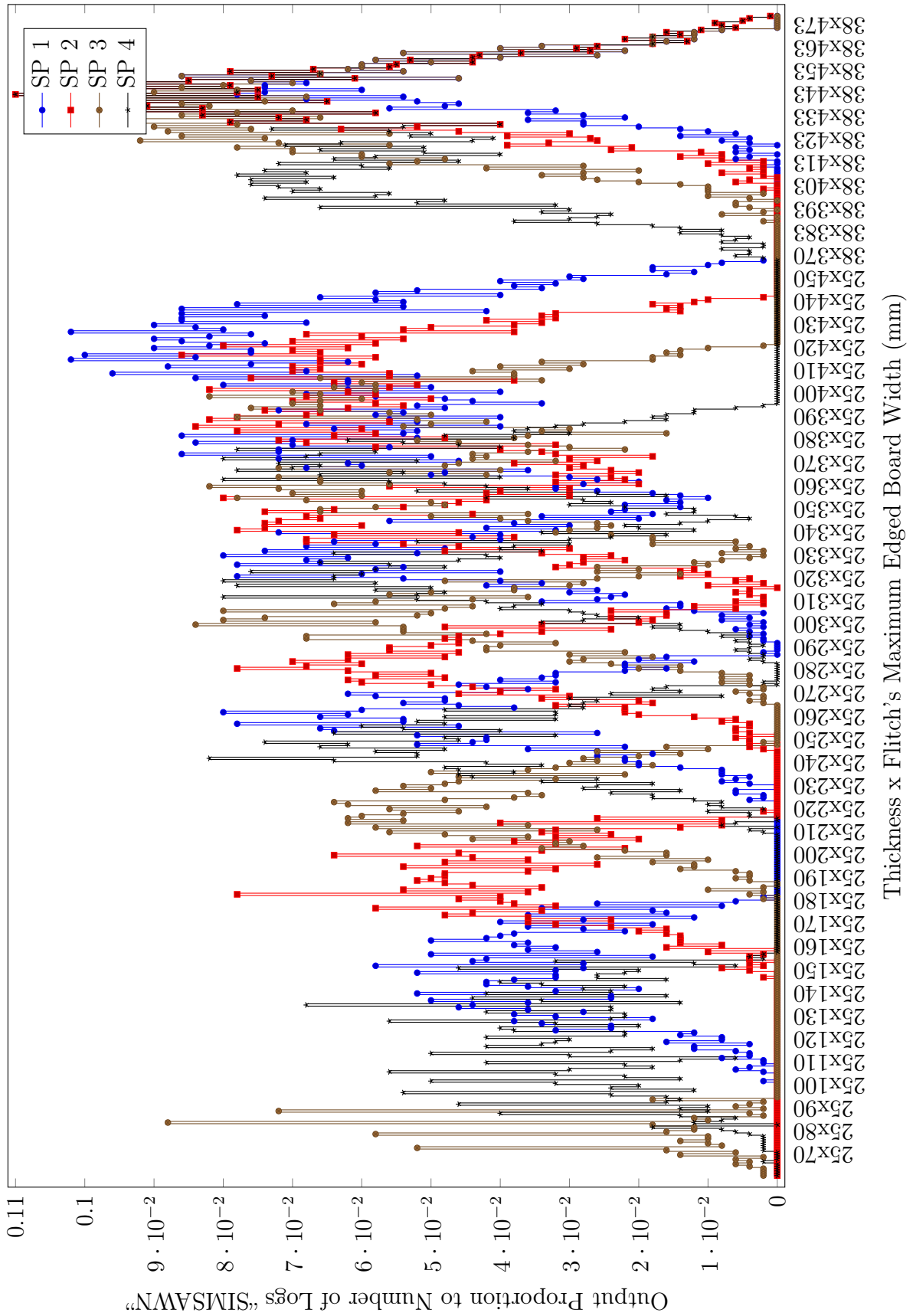
Log Diameter Class (d) 16: 45-46.9 cm



Log Diameter Class (d) 17: 47-48.9 cm

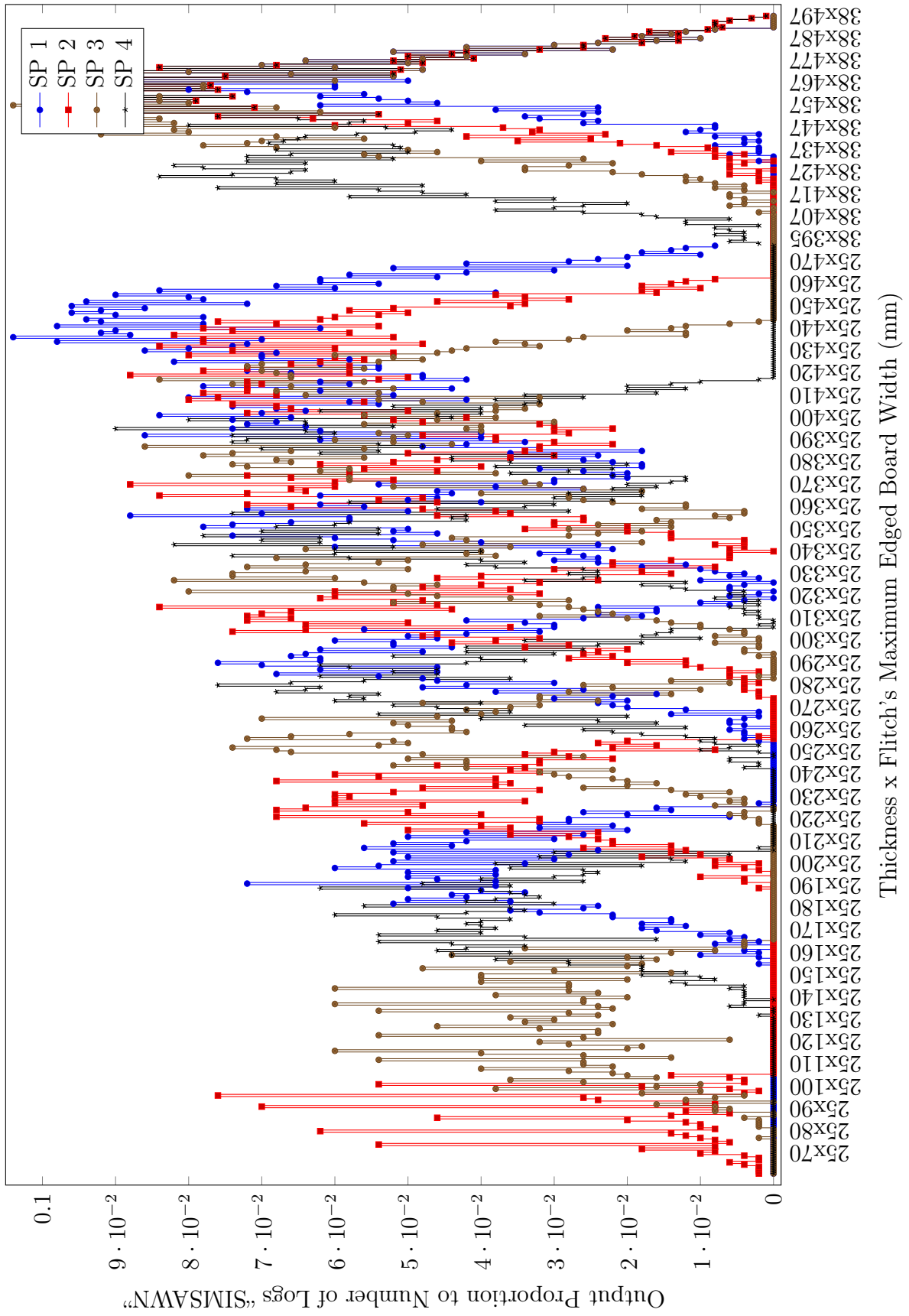


Log Diameter Class (d) 18: 49-50.9 cm





Log Diameter Class (d) 19: 51-52.9 cm







## Appendix E

# Model's Code in AIMMS 3.13 Environment

The project's model was encoded and run in the Advanced Interactive Multidimensional Modelling System (AIMMS) modelling environment and the entire code is shown below. The first part shows the declaration of the variables, parameters and constraints followed by the main execution procedure of the model.

MAIN MODEL Main\_1

DECLARATION SECTION PBIL

```
SET:
  identifier : Vector
  index      : v
  order by   : val(v)
  definition  : data{1..30} ;

SET:
  identifier : Bit
  index      : b
  order by   : val(b) ;

SET:
  identifier : Generations
  index      : g
  order by   : val(g) ;

PARAMETER:
  identifier : Component_Gene
  index domain : (t,w,b) ;

PARAMETER:
  identifier : Initial_Probability_vector
  index domain : (b)
  definition  : 0.5 ;

PARAMETER:
  identifier : Learning_Rate ;

PARAMETER:
  identifier : Mutation_Shift ;

PARAMETER:
  identifier : Mutation_Probability ;

PARAMETER:
  identifier : Priority_bin_dec_conversion
  index domain : (t,b) ;

VARIABLE:
  identifier : Cutting_Values
  index domain : (g,v,d,c)
  range      : free ;

VARIABLE:
  identifier : Cumulative_Probability_Vectors
  index domain : (g,b)
  range      : free ;

VARIABLE:
  identifier : Solution_Vectors
  index domain : (g,v,b)
  range      : free ;

VARIABLE:
  identifier : Probability_vectors
  index domain : (g,b)
  range      : free ;

VARIABLE:
```

```
    identifier : Updated_Probability_Vectors
    index domain : (g,b)
    range : free ;

VARIABLE:
    identifier : Best_Population_binarys
    index domain : (g,b)
    range : free ;

VARIABLE:
    identifier : Sum_Solution_Vectors
    index domain : (g,b)
    range : free ;

VARIABLE:
    identifier : Rand
    index domain : (g,v,d)
    range : free ;

VARIABLE:
    identifier : Count_vectors
    range : free ;

VARIABLE:
    identifier : Count_bits
    range : free ;

VARIABLE:
    identifier : Convergence
    index domain : (g)
    range : free ;

VARIABLE:
    identifier : Component_Priority_Values
    index domain : (g,v,t,w)
    range : free ;

VARIABLE:
    identifier : Component_Out_Values
    index domain : (g,v)
    range : free ;

VARIABLE:
    identifier : Component_Values
    index domain : (g,v,t,f,w)
    range : free ;

VARIABLE:
    identifier : Component_Supplies_V
    index domain : (g,v,t,w,h)
    range : free ;

VARIABLE:
    identifier : Component_Supplies_pieces
    index domain : (g,v,t,w,h)
    range : free ;

VARIABLE:
    identifier : Component_Volumes_Out
    index domain : (g,v)
    range : free ;

ENDSECTION ;
```

## DECLARATION SECTION SAWING

```
SET:
  identifier : Thickness
  index      : t
  order by   : val(t) ;

SET:
  identifier : Height
  index      : h ;

SET:
  identifier : Flitch
  index      : f
  order by   : val(f)
  definition : data {1..600} ;

SET:
  identifier : Log_class
  index      : d
  order by   : val(d)
  definition : data {1..21} ;

SET:
  identifier : Cutting_Pattern
  index      : c
  definition : data {1..4} ;

PARAMETER:
  identifier : Raw_Material_Cost ;

PARAMETER:
  identifier : M_Log_Dia
  index domain : (d) ;

PARAMETER:
  identifier : S_Logs
  index domain : (d,h) ;

PARAMETER:
  identifier : M_Log_Height
  index domain : (h) ;

PARAMETER:
  identifier : M_Flitch_board_width
  index domain : (t,f) ;

PARAMETER:
  identifier : SIM_Flitches_Proportion
  index domain : (d,C,t,f) ;

VARIABLE:
  identifier : LV
  index domain : (d,h)
  range       : free ;

VARIABLE:
  identifier : Count_log_classes
  range       : free ;

VARIABLE:
  identifier : Count_cutting_patterns
```

```
range      : free ;

VARIABLE:
  identifier : TRMV
  range     : free ;

VARIABLE:
  identifier : Best_Cutting_Values
  index domain : (g,d,c)
  range     : free ;

VARIABLE:
  identifier : Answer_Cutting_Value
  index domain : (d,c)
  range     : free ;

VARIABLE:
  identifier : Answer_Flitches_Out_pieces
  index domain : (t)
  range     : free ;

VARIABLE:
  identifier : Answer_Flitches_Out_Volumes
  index domain : (t,f,h)
  range     : free ;

VARIABLE:
  identifier : Answer_Flitches_Outs_pieces
  index domain : (t,f,h)
  range     : free ;

VARIABLE:
  identifier : Answer_Flitches_Out_Volume
  index domain : (t)
  range     : free ;

VARIABLE:
  identifier : Answer_Flitches_Out_Volume_all_t
  range     : free ;

VARIABLE:
  identifier : Answer_Framesaw_Recovery
  range     : free ;

VARIABLE:
  identifier : Flitches_Outs_pieces
  index domain : (g,v,t,f,h)
  range     : free ;

VARIABLE:
  identifier : Flitches_Out_pieces
  index domain : (g,v,t)
  range     : free ;

VARIABLE:
  identifier : Flitches_Out_Volumes
  index domain : (g,v,t,f,h)
  range     : free ;

VARIABLE:
  identifier : Flitches_Out_Volume
  index domain : (g,v,t)
  range     : free ;
```



```

VARIABLE:
  identifier : Flitches_Out_Volume_all_t
  index domain : (g,v)
  range : free ;

VARIABLE:
  identifier : Framesaw_Recovery
  index domain : (g,v)
  range : free ;

ENDSECTION ;

DECLARATION SECTION RIPPING

SET:
  identifier : Width
  index : w
  order by : val(w) ;

PARAMETER:
  identifier : S_Flitches_Supplied
  index domain : (t,f,h) ;

PARAMETER:
  identifier : M_Lumber_width
  index domain : (t,w) ;

PARAMETER:
  identifier : P_Lumber_Priority
  index domain : (t,w) ;

PARAMETER:
  identifier : K2_Kerf2 ;

VARIABLE:
  identifier : In_Volumes
  index domain : (t,f,h)
  range : free
  definition :
S_Flitches_Supplied(t,f,h)*M_Flitch_board_width(t,f)/1000*val(t)/1000*M_Log_Height(h)
;

VARIABLE:
  identifier : In_Volume
  range : free
  definition : sum[(t,f,h), In_Volumes(t,f,h)] ;

VARIABLE:
  identifier : Value
  index domain : (t,f,w)
  range : {0..3} ;

VARIABLE:
  identifier : Supply
  index domain : (t,w)
  range : free
  definition :
sum[(f,h), Value(t,f,w)*S_Flitches_Supplied(t,f,h)*(M_Lumber_width(t,w)/1000)*val(t)/1000*M_Log_Height(h)] ;

VARIABLE:
  identifier : Supply_pieces

```

```

index domain : (t,w,h)
range        : free
definition   : sum[f, Value(t,f,w)*S_Flitches_Supplied(t,f,h)] ;

VARIABLE:
identifier   : Out_Volumes
index domain : (t,f)
range        : free
definition   : sum [(w,h),
Value(t,f,w)*S_Flitches_Supplied(t,f,h)*(M_Lumber_width(t,w)/1000)*val(t)/1000*M_Log_H
eight(h)] ;

VARIABLE:
identifier   : Out_Volume
range        : free
definition   : sum [(t,f,w,h),
Value(t,f,w)*S_Flitches_Supplied(t,f,h)*(M_Lumber_width(t,w)/1000)*val(t)/1000*M_Log_H
eight(h)] ;

VARIABLE:
identifier   : Out_Value
range        : free
definition   : sum [(t,f,w),
Value(t,f,w)*P_Lumber_Priority(t,w)*M_Lumber_width(t,w)] ;

CONSTRAINT:
identifier   : Spacers
index domain : (t,f)
definition   : sum[w, Value(t,f,w)]<=3 ;

CONSTRAINT:
identifier   : Fit
index domain : (t,f)
definition   : sum[w, Value(t,f,w)*M_Lumber_width(t,w)]+(sum[w, Value(t,f,w)]-
1)*K2_Kerf2<=M_Flitch_board_width(t,f) ;

MATHEMATICAL PROGRAM:
identifier   : Most_Out
objective    : Out_Value
direction    : maximize
constraints  : AllConstraints
variables    : AllVariables
type         : Automatic ;

ENDSECTION ;

```

## DECLARATION SECTION PBIL\_Results

```
PARAMETER:
  identifier : Lumber_Cost_Over_Production
  index domain : (t,w) ;

PARAMETER:
  identifier : Lumber_Cost_Under_Production
  index domain : (t,w) ;

PARAMETER:
  identifier : D_Demand_Lumber
  index domain : (t,w) ;

VARIABLE:
  identifier : Answer
  index domain : (b)
  range : free ;

VARIABLE:
  identifier : Best_Component_Supplies
  index domain : (g,t,w)
  range : free ;

VARIABLE:
  identifier : Converged_Component_Priority
  index domain : (t,w)
  range : free ;

VARIABLE:
  identifier : Converged_Component_Supply
  index domain : (t,w)
  range : free ;

VARIABLE:
  identifier : Converged_Component_Supply_Demand
  index domain : (t,w)
  range : free ;

VARIABLE:
  identifier : Converged_Cost_Overs
  index domain : (t,w)
  range : free ;

VARIABLE:
  identifier : Converged_Cost Unders
  index domain : (t,w)
  range : free ;

VARIABLE:
  identifier : Converged_Cost_Overs Unders
  index domain : (t,w)
  range : free ;

VARIABLE:
  identifier : Converged_Cost_Over
  index domain : (t)
  range : free ;

VARIABLE:
  identifier : Converged_Cost_Under
  index domain : (t)
  range : free ;
```

```
VARIABLE:
  identifier : Converged_Cost_Over_Under
  index domain : (t)
  range : free ;

VARIABLE:
  identifier : Converged_Cost_Over_all_t
  range : free ;

VARIABLE:
  identifier : Converged_Cost_Under_all_t
  range : free ;

VARIABLE:
  identifier : Converged_Cost_Over_Under_all_t
  range : free ;

VARIABLE:
  identifier : Converged_Cost_Raw_materials
  range : free ;

VARIABLE:
  identifier : Converged_Recovery
  range : free ;

VARIABLE:
  identifier : Converged_Total_Cost
  range : free ;

VARIABLE:
  identifier : Population_Priority_W
  index domain : (g,v,t,w)
  range : free ;

VARIABLE:
  identifier : Best_Population_Priority_W
  index domain : (g,t,w)
  range : free ;

VARIABLE:
  identifier : Best_Costs_Overs
  index domain : (g,t,w)
  range : free ;

VARIABLE:
  identifier : Best_Costs_Unders
  index domain : (g,t,w)
  range : free ;

VARIABLE:
  identifier : Best_Costs_Overs_Unders
  index domain : (g,t,w)
  range : free ;

VARIABLE:
  identifier : Best_Costs_Over
  index domain : (g,t)
  range : free ;

VARIABLE:
  identifier : Best_Costs_Under
  index domain : (g,t)
```

```
range      : free ;

VARIABLE:
  identifier : Best_Costs_Over_Under
  index domain : (g,t)
  range      : free ;

VARIABLE:
  identifier : Best_Costs_Over_all_t
  index domain : (g)
  range      : free ;

VARIABLE:
  identifier : Best_Costs_Under_all_t
  index domain : (g)
  range      : free ;

VARIABLE:
  identifier : Best_Costs_Over_Under_all_t
  index domain : (g)
  range      : free ;

VARIABLE:
  identifier : Best_Costs_Raw_material
  index domain : (g)
  range      : free ;

VARIABLE:
  identifier : Fitness_Function
  index domain : (g,v)
  range      : free ;

VARIABLE:
  identifier : Best_Fitness_Function
  index domain : (g)
  range      : free ;

VARIABLE:
  identifier : Recovery
  index domain : (g,v)
  range      : free ;

VARIABLE:
  identifier : Best_Recovery
  index domain : (g)
  range      : free ;

VARIABLE:
  identifier : Costs_Overs
  index domain : (g,v,t,w)
  range      : free ;

VARIABLE:
  identifier : Costs_Unders
  index domain : (g,v,t,w)
  range      : free ;

VARIABLE:
  identifier : Cost_Over_Unders
  index domain : (g,v,t,w)
  range      : free ;

VARIABLE:
```

```
identifier : Costs_Over
index domain : (g,v,t)
range : free ;
```

```
VARIABLE:
identifier : Costs_Under
index domain : (g,v,t)
range : free ;
```

```
VARIABLE:
identifier : Costs_Over_Under
index domain : (g,v,t)
range : free ;
```

```
VARIABLE:
identifier : Cost_Over_all_t
index domain : (g,v)
range : free ;
```

```
VARIABLE:
identifier : Cost_Under_all_t
index domain : (g,v)
range : free ;
```

```
VARIABLE:
identifier : Cost_Over_Under_all_t
index domain : (g,v)
range : free ;
```

```
VARIABLE:
identifier : Costs_Raw_materials
index domain : (g,v)
range : free ;
```

```
ENDSECTION ;
```

```
PROCEDURE
identifier : MainInitialization
```

```
ENDPROCEDURE ;
```

```

PROCEDURE
  identifier : MainExecution
  body      :

empty AllVariables;

Rand(g,v,d) := Uniform(0,1);
Count_vectors:=Count(v);
Count_bits:=Count(b);
Count_log_classes:=Count(d);
Count_cutting_patterns:=Count(c);
LV(d,h) :=
S_Logs(d,h)*3.143*(M_Log_Dia(d)*100+M_Log_Height(h)/2)^2*M_Log_Height(h)/40000;
TRMV:=sum [(d,h), LV(d,h)]; !Total Raw Material Volume supplied

for g do; ! For each g (Populations/Generation) do

if val(g)=0 then
Probability_vectors(g,b) := Initial_Probability_vector(b) else !initite first
population's probability_vector (= paramter, usually 50%)
Probability_vectors(g,b) := Probability_vectors(g-1,b)*(1-
Learning_Rate)+Best_Population_binarys(g-1,b)*Learning_Rate; !normal Learning Rate
Probability_vectors(g,b) :=Probability_vectors(g,b)*(1-
Mutation_shift)+(round(Uniform(0,1)))*Mutation_shift Onlyif (Uniform(0,1)<=
Mutation_Probability); !Mutate
endif;

for d do
for b|val(d)*4-3<=val(b)<=val(d)*4 do
if val(b)=val(d)*4-3 then
Updated_Probability_Vectors(g,b) :=Probability_Vectors(g,b);
else
Updated_Probability_Vectors(g,b) :=Probability_Vectors(g,b)+Updated_Probability_Vectors
(g,b-1);
endif;
endfor;
endfor;

```

```

endfor;

for v do ! For all Solution Vectors (Population size) do

for d do

for b|val(b)<=Count_log_classes*Count(c) do

if val(b)=val(d)*4-3 then
if Rand(g,v,d)*Updated_Probability_Vectors(g,b+3)<= Updated_Probability_Vectors(g,b)
then
Solution_Vectors(g,v,b):=1;
endif;
endif;

if val(b)=val(d)*4-2 then
if Updated_Probability_Vectors(g,b-1)<
Rand(g,v,d)*Updated_Probability_Vectors(g,b+2)<= Updated_Probability_Vectors(g,b) then
Solution_Vectors(g,v,b):=1;
endif;
endif;

if val(b)=val(d)*4-1 then
if Updated_Probability_Vectors(g,b-1)<
Rand(g,v,d)*Updated_Probability_Vectors(g,b+1)<= Updated_Probability_Vectors(g,b) then
Solution_Vectors(g,v,b):=1;
endif;
endif;

if val(b)=val(d)*4 then
if Updated_Probability_Vectors(g,b-1)< Rand(g,v,d)*Updated_Probability_Vectors(g,b)
then
Solution_Vectors(g,v,b):=1;
endif;
endif;

endifor;
endifor;

endifor;

for b|val(b)> Count(d)*Count(c) do !for bits representing ripping decisions (priority
values)
for v do
if Uniform(0,1) < ( (Probability_vectors(g,b)*Count_vectors-
Sum_Solution_Vectors(g,b))/(Count_vectors-val(v)+1)) then
Solution_Vectors(g,v,b):= 1; Sum_Solution_Vectors(g,b):=Sum_Solution_Vectors(g,b)+1;
!else
endif
endifor;
endifor;

for d do
for c do
for b|val(b)=(val(d)-1)*4+val(c) do
Cutting_Values(g,v,d,c):=Solution_Vectors(g,v,b);
endifor;
endifor;
endifor;

```



```

Flitches_Out_pieces(g,v,t,f,h):=round(sum[(d,c),
Cutting_Values(g,v,d,c)*SIM_Flitches_Proportion(d,c,t,f)*S_Logs(d,h)];
Flitches_Out_pieces(g,v,t):=sum[(f,h), Flitches_Out_pieces(g,v,t,f,h)];
Flitches_Out_Volumes(g,v,t,f,h):=Flitches_Out_pieces(g,v,t,f,h)*val(t)/1000*M_Flitch_
board_width(t,f)/1000*M_Log_Height(h);
Flitches_Out_Volume(g,v,t):=sum[(f,h), Flitches_Out_Volumes(g,v,t,f,h)];
Flitches_Out_Volume_all_t(g,v):=sum[t,Flitches_Out_Volume(g,v,t)];
Framesaw_Recovery(g,v):= Flitches_Out_Volume_all_t(g,v)/TRMV;

Population_Priority_W(g,v,t,w):=
sum[b,Solution_Vectors(g,v,b)*Priority_bin_dec_conversion(t,b)*Component_Gene(t,w,b)];
!convert from binary to decimal

for v do
P_Lumber_Priority(t,w):= Population_Priority_W(g,v,t,w); !variable becomes the
parameter for MIP solver
S_Flitches_Supplied(t,f,h):=Flitches_Out_pieces(g,v,t,f,h); !variable becomes the
parameter for MIP solver

solve Most_Out;

Costs_Overs(g,v,t,w):= (Supply(t,w)-
D_Demand_Lumber(t,w))*Lumber_Cost_Over_Production(t,w) Onlyif
(Supply(t,w)>=D_Demand_Lumber(t,w));
Costs_Over(g,v,t):=sum[w, Costs_Overs(g,v,t,w)];
Cost_Over_all_t(g,v):=sum[t, Costs_Over(g,v,t)];

Costs_Unders(g,v,t,w):= (D_Demand_Lumber(t,w)-
Supply(t,w))*Lumber_Cost_Under_Production(t,w)
Onlyif(Supply(t,w)<=D_Demand_Lumber(t,w));
Costs_Under(g,v,t):=sum[w, Costs_Unders(g,v,t,w)];
Cost_Under_all_t(g,v):=sum[t, Costs_Under(g,v,t)];

Cost_Over_Unders(g,v,t,w):= Costs_Overs(g,v,t,w)+Costs_Unders(g,v,t,w);
Costs_Over_Under(g,v,t):= Costs_Over(g,v,t)+Costs_Under(g,v,t);
Cost_Over_Under_all_t(g,v):=Cost_Over_all_t(g,v)+Cost_Under_all_t(g,v);

Costs_Raw_materials(g,v):= (TRMV-Out_Volume)*Raw_Material_Cost;

Recovery(g,v):= Out_Volume/TRMV;

Fitness_Function(g,v):= Cost_Over_Under_all_t(g,v)+ Costs_Raw_materials(g,v);

if val(v)= 1 then !Find Best Vector(v) in Population(p)
Best_Fitness_Function(g):=Fitness_Function(g,v);
Best_Fitness_Function(g):= Fitness_Function(g,v);
Best_Population_binaries(g,b):= Solution_Vectors(g,v,b);
Best_Population_Priority_W(g,t,w):= Population_Priority_W(g,v,t,w);
Best_Cutting_Values(g,d,c):= Cutting_Values(g,v,d,c);

Best_Component_Supplies(g,t,w):= Supply(t,w);

Best_Costs_Overs(g,t,w):= Costs_Overs(g,v,t,w);
Best_Costs_Unders(g,t,w):= Costs_Unders(g,v,t,w);
Best_Costs_Overs_Unders(g,t,w):=Cost_Over_Unders(g,v,t,w);

Best_Costs_Over(g,t):= Costs_Over(g,v,t);

```

```

Best_Costs_Under(g,t) := Costs_Under(g,v,t);
Best_Costs_Over_Under(g,t) := Costs_Over_Under(g,v,t);

Best_Costs_Over_all_t(g) := Cost_Over_all_t(g,v);
Best_Costs_Under_all_t(g) := Cost_Under_all_t(g,v);
Best_Costs_Over_Under_all_t(g) := Cost_Over_Under_all_t(g,v);

Best_Costs_Raw_material(g) := Costs_Raw_materials(g,v);
Best_Recovery(g) := Recovery(g,v);

else
if Fitness_Function(g,v) < Best_Fitness_Function(g) then
Best_Fitness_Function(g) := Fitness_Function(g,v);
Best_Fitness_Function(g) := Fitness_Function(g,v);
Best_Population_binarys(g,b) := Solution_Vectors(g,v,b);
Best_Population_Priority_W(g,t,w) := Population_Priority_W(g,v,t,w);
Best_Cutting_Values(g,d,c) := Cutting_Values(g,v,d,c);

Best_Component_Supplies(g,t,w) := Supply(t,w);

Best_Costs_Overs(g,t,w) := Costs_Overs(g,v,t,w);
Best_Costs_Unders(g,t,w) := Costs_Unders(g,v,t,w);
Best_Costs_Overs_Unders(g,t,w) := Cost_Over_Unders(g,v,t,w);

Best_Costs_Over(g,t) := Costs_Over(g,v,t);
Best_Costs_Under(g,t) := Costs_Under(g,v,t);
Best_Costs_Over_Under(g,t) := Costs_Over_Under(g,v,t);

Best_Costs_Over_all_t(g) := Cost_Over_all_t(g,v);
Best_Costs_Under_all_t(g) := Cost_Under_all_t(g,v);
Best_Costs_Over_Under_all_t(g) := Cost_Over_Under_all_t(g,v);

Best_Costs_Raw_material(g) := Costs_Raw_materials(g,v);
Best_Recovery(g) := Recovery(g,v);

endif;
endif;

Component_Priority_Values(g,v,t,w) := P_Lumber_Priority(t,w);
Component_Values(g,v,t,f,w) := Value(t,f,w);
Component_Out_Values(g,v) := Out_Value;
Component_Supplies_pieces(g,v,t,w,h) := Supply_pieces(t,w,h);
Component_Supplies_V(g,v,t,w,h) := Component_Supplies_pieces(g,v,t,w,h) * M_Lumber_width(t,w) / 1000 * Val(t) / 1000 * M_Log_Height(h);
Component_Volumes_Out(g,v) := Out_Volume;

endifor;

Answer(b) := Round(Probability_vectors(g,b));
Converged_Component_Priority(t,w) := sum[b, Answer(b) * Priority_bin_dec_conversion(t,b) * Component_Gene(t,w,b)];

for d do
for c do
for b | val(b) = (val(d)-1) * 4 + val(c) do
Answer_Cutting_Value(d,c) := Answer(b | val(b) = (val(d)-1) * 4 + val(c));
endifor;

```

```

endfor;
endifor;

for b do
if 0.05 < Probability_vectors(g,b) < 0.95 then
Convergence(g) := Convergence(g)+1
endif;
endifor;

StatusMessage(val(g));
DialogProgress("Number of Bits Converging (%)", ! (input) string expression
[(Count_bits-Convergence(g))*100/Count_bits]); ! (optional) integer expression);

if Convergence(g) <= 0 then
break
endif;

endifor;

!Final Answer

Answer_Flitches_Out_pieces(t,f,h) := round(sum[(d,c),
Answer_Cutting_Value(d,c)*SIM_Flitches_Proportion(d,c,t,f)*S_Logs(d,h)]);
Answer_Flitches_Out_pieces(t) := sum[(f,h), Answer_Flitches_Out_pieces(t,f,h)];
Answer_Flitches_Out_Volumes(t,f,h) := Answer_Flitches_Out_pieces(t,f,h)*val(t)/1000*M_F
litch_board_width(t,f)/1000*M_Log_Height(h);
Answer_Flitches_Out_Volume(t) := sum[(f,h), Answer_Flitches_Out_Volumes(t,f,h)];
Answer_Flitches_Out_Volume_all_t := sum[t, Answer_Flitches_Out_Volume(t)];
Answer_Framesaw_Recovery := Answer_Flitches_Out_Volume_all_t/TRMV;

S_Flitches_Supplied(t,f,h) := Answer_Flitches_Out_pieces(t,f,h);
P_Lumber_Priority(t,w) := Converged_Component_Priority(t,w);

solve Most_Out;

Converged_Component_Supply(t,w) := Supply(t,w);
Converged_Component_Supply_Demand(t,w) := Converged_Component_Supply(t,w) -
D_Demand_Lumber(t,w);

Converged_Cost_Overs(t,w) := (Supply(t,w) -
D_Demand_Lumber(t,w))*Lumber_Cost_Over_Production(t,w) Onlyif
(Supply(t,w) >= D_Demand_Lumber(t,w));
Converged_Cost_Over(t) := sum[w, Converged_Cost_Overs(t,w)];
Converged_Cost_Over_all_t := sum[t, Converged_Cost_Over(t)];

```

```
Converged_Cost_Unders(t,w) := (D_Demand_Lumber(t,w) -  
Supply(t,w)) * Lumber_Cost_Under_Production(t,w) Onlyif  
(Supply(t,w) <= D_Demand_Lumber(t,w));  
Converged_Cost_Under(t) := sum[w, Converged_Cost_Unders(t,w)];  
Converged_Cost_Under_all_t := sum[t, Converged_Cost_Under(t)];  
  
Converged_Cost_Overs_Unders(t,w) := Converged_Cost_Overs(t,w) + Converged_Cost_Unders(t,w)  
;  
Converged_Cost_Over_Under(t) := Converged_Cost_Over(t) + Converged_Cost_Under(t);  
Converged_Cost_Over_Under_all_t := Converged_Cost_Over_all_t + Converged_Cost_Under_all_t;  
  
Converged_Cost_Raw_materials := (TRMV - Out_Volume) * Raw_Material_Cost;  
Converged_Recovery := Out_Volume / TRMV;  
  
Converged_Total_Cost := Converged_Cost_Raw_materials + Converged_Cost_Over_Under_all_t;
```