

Modelling the South African fresh fruit export supply chain

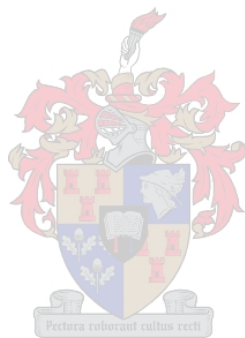
Frank Gerald Ortmann



Thesis presented in partial fulfilment of the requirements for the degree of
Master of Science in the Department of Applied Mathematics,
University of Stellenbosch, South Africa

Supervisor: Prof JH van Vuuren
Advisor: Dr FE van Dyk

December 2005

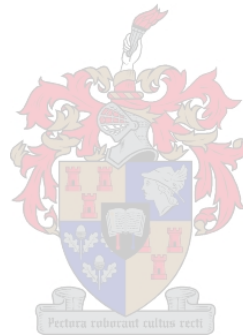


Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: _____

Date: _____





Abstract

The process of modelling the fruit export infrastructure capacity of South Africa formed part of a larger project called the “Fruit Logistics Infrastructure Capacity Optimisation Study,” which was coordinated by the Transportek division of the CSIR in Stellenbosch during the period August 2002 to March 2004. The aim of this project was to create efficiencies for, and enhance the competitiveness of, the South African fruit industry by improved usage of, and investment in, shared logistics infrastructure.

After putting the size of the fruit industry into perspective, numerous aspects of the export process are considered in this thesis so as to be able to perform a comprehensive cost analysis of the export of fruit, including the cost of handling, cooling and transportation. The capacities of packhouses, cold stores and terminals are found and presented. This information, combined with fruit export volumes of 2003, then allow an estimation of the current utilisation of the South African ports with respect to fruit export.

The infrastructure of the fruit export industry comprises regional packhouses and cold stores, cold stores at port (harbour and airport) cities and terminals at the ports through which fruit is exported. In order to determine which segments of the infrastructure is put under the most strain during peak export periods, the maximum volume of fruit that can flow through the current infrastructure is modelled mathematically and compared to 2003 export volumes. Both network flow and linear programming models are employed, and both maximum flow and minimum cost flow solutions are found for various scenarios. This is followed by a discussion of the strengths and weaknesses of the (single-commodity) network theoretic approach versus the (multicommodity) mathematical programming approach.

Finally, a number of suggestions and recommendations are made with respect to streamlining the fruit export process and the identification of possible points of expansion in the fruit export infrastructure. In short, it is found that the bottlenecks experienced by export role-players are not caused by a lack of logistics infrastructure capacity, and that the adoption of cooperation policies and scheduling procedures may lead to fewer bottlenecks. The minimum cost analyses also show that Maputo is the only port whose expansion may benefit South African exporters, save for reservations regarding problems at the Mozambique/South African border, available shipping lines and costs incurred.



Opsomming

Die modellering van die infrastruktuur in die Suid-Afrikaanse vrugte uitvoerbedryf het deel gevorm van 'n groter projek getiteld “Fruit Logistics Infrastructure Capacity Optimisation Study,” wat deur die Transportek afdeling van WNNR in Stellenbosch gedurende Augustus 2002 tot Maart 2004 gekoördineer is. Die doel van hierdie projek was om die effektiwiteit en mededingendheid van die Suid Afrikaanse vrugte uitvoerbedryf te verhoog deur beter gebruik te maak van, en te belê in, gedeelde logistieke infrastruktuur.

Die grootte van die vrugte uitvoerbedryf word in perspektief geplaas, waarna verskeie aspekte van die uitvoerproses in hierdie tesis beskou word om sodoende 'n omvattende koste-analise van die uitvoer van vars vrugte, insluitend hanteerkostes, verkoelingskoste en die vervoerkoste, te kan doen. Die kapasiteite van pakhuis, koelkamers en terminale word ondersoek en voorgelê. Hierdie inligting, gekombineer met die vrugte uitvoervolumes van 2003, word gebruik om 'n beraming van die huidige benutting van die Suid Afrikaanse hawens te maak ten opsigte van die uitvoer van vars vrugte.

Die infrastruktuur van die vrugte uitvoerbedryf bestaan uit streekspakhuis en -koelkamers in verskillende gebiede, koelkamers by see- en lughawens en terminale by die hawes waarvandaan vrugte uitgevoer word. Die maksimum volumes van vrugte wat deur die huidige infrastruktuur kan vloei, word wiskundig gemodelleer en vergelyk met die 2003 uitvoervolumes, om sodoende te bepaal watter dele van die infrastruktuur die grootste druk gedurende spitsuitvoertye ervaar. Daar word gebruik gemaak van beide netwerkvloei- en lineêre programmeringsmodelle en oplossings vir beide maksimum vloei en minimum koste-vloei situasies word vir 'n verskeidenheid van scenarios verkry. Hierdie resultate word deur 'n bespreking van sterk- en swakpunte van die (enkel-kommoditeit) netwerkvloei benadering teenoor die (multi-kommoditeit) wiskundige programmeringsbenadering.

Ten slotte word 'n aantal aanbevelings gemaak ten opsigte van hoe om die vrugte uitvoerbedryf moontlik meer doeltreffend te maak. Sekere punte in die infrastruktuur van die vrugte uitvoerbedryf word ook geïdentifiseer. In kort, daar is gevind dat die bottelnekke wat in die uitvoerbedryf deur uitvoerders ervaar word nie veroorsaak word deur 'n tekort aan logistieke infrastruktuur kapasiteit nie en dat beter samewerking, ooreenkomste en skeduleringspraktyke mag lei tot minder bottelnekke. Die minimale koste-analise wys ook duidelik dat uitbreiding van die hawe by Maputo die enigste hawe-uitbreiding is wat moontlik voordeel inhou vir Suid Afrikaanse uitvoerders, alhoewel probleme by die Mosambiek/Suid-Afrika-grens ervaar word, beskikbare skeepsroetes skaars en koste hoog is.



Terms of Reference

This thesis has its roots in a large-scale modelling project for the South African fresh fruit export industry. In 2002 the Deciduous Fruit Producers' Trust (DFPT) approached the Transportek division of the Council of Scientific and Industrial Research (CSIR) to investigate possibilities with respect to rendering the South African fruit export logistics operations more efficient and effective, by making recommendations regarding the utilisation and possible expansion of the fresh fruit export infrastructure. The DFPT members had been experiencing inefficiencies and bottlenecks in ports, and this gave rise to the Fruit Logistics and Infrastructure Optimisation Study (Fruitlog) coordinated by the CSIR, but including many institutions. This thesis forms a part of Fruitlog — it was the intention of the DFPT that students should be involved in the project, so as to give them experience in the fruit/agriculture/export industries. The project was funded by the DFPT and the Department of Trade and Industry (DTI) through its Sector Partnership Fund.

Dr FE van Dyk from the Transportek division of the CSIR was appointed project manager of Fruitlog. Dr DB Louw from Optimal Agricultural Business Systems (OABS) contracted by the DFPT as their information manager, was a key partner in the project, especially in terms of providing the necessary fruit volume production data (which were central to the models developed and implemented in this thesis). Dr NH Louw from the Department of Logistics was chosen to oversee certain aspects of the logistics deliverables of Fruitlog, and to act as thesis supervisor to a number of logistics masters and honours students, who formed a part of the larger project team.

Prof JH van Vuuren from the Department of Applied Mathematics was approached to act as supervisor to a student who would model the fruit export infrastructure mathematically in order to aid in decision making, by using graph theoretic methods and mathematical programming methods. Prof. Van Vuuren subsequently became the supervisor to the author of this thesis. At the beginning of 2003 the author joined the Fruitlog project team as a mathematical modeller.

Later during the project, Mr J Havenga and Ms I Hobbs from the Executive Development group of University of Stellenbosch Business School joined the project team to aid in the creation of the Fruitlog Scenario Model. Further studies were also to be performed by the Departments of Agricultural Economics and Industrial Engineering.

A pilot study to the Fruitlog project began during August 2002. Two industrial engineering final-year students, Ms M Mostert and Mr C Kritzinger, joined the project team in 2003. They modelled the Fresh Produce Terminals (FPT) operations in Durban and Cape Town respectively, using discrete event simulation techniques. Mr J Bekker from the Department of Industrial Engineering at the University of Stellenbosch was their project supervisor. Investigations into transaction costs for grapes and citrus fruit (conducted by Mr T Fundira — masters student in the Department of Agricultural Economics at the University of Stellenbosch), a proposal for a methodology for medium and long-term crop estimates (put forward by Dr DB Louw — OABS),

the comparison of road versus rail transport (performed by Mr M Marx, Mr D Pienaar and Mr M Brotherton — all students at the Department of Logistics at the University of Stellenbosch) and an investigation into the possible use of additional ports for fruit export (conducted by Ms LL Gerber and Dr NH Louw, masters student and lecturer in the Department of Logistics at the University of Stellenbosch respectively) formed part of the Fruitlog studies performed during 2003.

The gathering of data required for the models described in this thesis commenced in March 2003 and was due for completion by the end of October 2003. These data included packhouse and cold store capacities and fruit transport and handling costs. Contact details for packhouses and cold stores were obtained from the PPECB by the CSIR. Telephonic, facsimile and email services were provided by the CSIR and were used by Ms S Mntusanto, an intern at the CSIR at the time, and the author of this thesis to collect the capacity information of all of the packhouses and cold stores in South Africa. During the early stages of the information gathering process, a number of students from other departments assisted in the collection of data. The capacity data required for this thesis were only finalised in March 2004, when the Fruitlog Scenario Model was developed by Dr Van Dyk, Mr Havenga, Ms Hobbs, Drs DB & NH Louw and the author. The larger Fruitlog study was completed successfully in March 2004.

Many visits were made to role-players in the industry in order to gain an understanding of the various parts of the fruit industry. Visits were made by the author to Mr J Kriegler's table grape farm and the cold store *Hexkoel* (hosted by Mr R du Toit, the general manager) in De Doorns on 27 February 2003, Cape Town FPT terminal on 6 March 2003 (hosted by Ms T Nielsen, Public Relations Officer), the Cape Town container terminal on 13 March 2003 (hosted by Ms C Christian, key account manager), FPT Durban on 24 April 2003 (hosted by Mr N de Klerk, public relations officer), National Ports Authority (hosted by Mr J Willemse and Mr B Cilliers), *LauritzenCool Southern Africa* (hosted by Mr T Law, managing director) and FPT Cape Town (hosted by Mr K Bouwer, managing director) on 7 May 2003, Citrus South Africa on 8 May 2003 (hosted by Johan Keyser, amongst others), *Kromco* packhouse and cold store and *Monteith Farms* in June 2004 (hosted by Mr X Murray, quality control manager), *Logicool* cold store in Piketberg on 31 January 2005 (hosted by Mr E Jooste, owner, who later also provided cold store costs) and *Kuehne + Nagel* freight forwarders on 14 April 2005 (hosted by Mr R Opperman, perishable product manager). Transportation specialists such as Mr M Braun (4 August 2004) and Mr A du Bois (16 May 2005) — Mr Braun at his own consulting company and Mr Du Bois at FPT — were met for discussions regarding transportation costs. Mr G van Heerden, managing director of *Cape Reefers*, kindly provided shipping costs. Mr L Beetge from *Optimatix* was also contacted with regards to problems when solving the mathematical programming models in this thesis. He also introduced the author to the solving package *PCx* and the online *Network-Enabled Optimization System (NEOS) Server for Optimization*.

Work on this thesis was completed in June 2005 and work from this thesis has been presented at a colloquium of the Western Cape chapter of the Operations Research Society of South Africa (ORSSA) in 2003 and at two ORSSA national conferences, one in 2003 and the other in 2004. The work has been accepted for presentation at the 2005 International Federation of Operations Research Societies (IFORS) conference, and a paper has been submitted for publication in an Operations Research journal.

Acknowledgements

Many people played a significant role in the work leading up to and during the writing of this thesis. The author hereby wishes to express his deepest gratitude towards

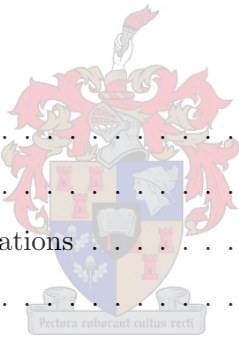
- the Department of Applied Mathematics of the University of Stellenbosch for the use of their computing facilities and office space and other forms of assistance.
- the Council for Scientific and Industrial Research, for the privilege of working on such a nationally important project and the use of their facilities and resources for the collection of vital data.
- Prof JH van Vuuren, for informing me of this project and for the supply of excellent ideas and support throughout the project.
- Dr FE van Dyk, for her tireless help with the collection of data, and for allowing me to work on this project.
- Jan Havenga and Ilse Hobbs for their help in the building of the *Fruitlog Scenario Model*.
- Stephen Berjak, for his excellent guidance and support in the compilation of report deliverables for Fruitlog.
- Leila Gerber, for all her help and supplying many useful telephone numbers or email addresses for fruit industry contacts.
- Siphokazi Mntusanto, for her patience while doing much of the telephoning and faxing to packhouses and cold stores.
- Xavier Murray for taking time on numerous occasions to show me around packhouses, cold stores and farms. Also for putting me in contact with some fruit industry role-players and answering many of my questions.
- Emma Maspero, Marize Mostert, Chris Kritzinger and Reinhardt Grobbelaar, for all their various forms of help throughout this project.
- all of the fruit industry role-players willing to part with much needed information and for their friendliness in person, on the telephone or in their emails.
- friends and family, for their loyal support and sympathy during great and tough times.



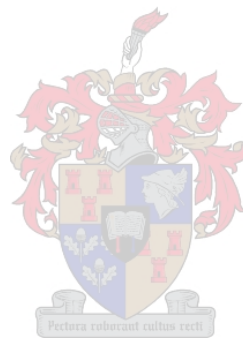
Table of Contents

List of Figures	v
List of Tables	ix
Glossary	xv
List of Acronyms	xxi
List of Reserved Symbols	xxiii
1 Introduction	1
1.1 Fruit Logistics Infrastructure Capacity Optimisation Study	2
1.2 Problem Formulation for this Thesis	4
1.3 Thesis Layout	5
2 The South African Fruit Industry	7
2.1 Fruit Produced	8
2.1.1 Fruit Seasons	10
2.1.2 Fruit Producing Regions	11
2.2 Export Infrastructure	17
2.2.1 Road Transport	18
2.2.2 Rail Transport	28
2.2.3 Sea Freight	31
2.2.4 Air Freight	32
2.3 The Export Process	32
2.3.1 Fruit on the Farm	33
2.3.2 Fruit at Communal Packing Facilities	37
2.3.3 Fruit at Regional Cold Stores	38
2.3.4 Fruit at the Port	40

2.4	Cost Data	43
2.4.1	Cold Store Handling Cost	44
2.4.2	Road and Rail Transportation Cost	45
2.4.3	Port Handling Cost	45
2.4.4	Sea and Air Freight Cost	46
2.4.5	Additional Costs	47
2.5	Chapter Summary	48
3	Infrastructure Capacities	49
3.1	Packhouse and Cold Store Capacities	49
3.2	Fruit Production and Transportation Data	51
3.2.1	Fruit Production Data by Region	51
3.2.2	Fruit Export Data	58
3.2.3	Summary of National Fruit Export Data	59
3.2.4	Origin–Destination Flow Volumes	63
3.3	Port Capacities	64
3.3.1	Container Terminals	65
3.3.2	Bulk Terminals	68
3.4	Demand on the Ports	71
3.5	Chapter Summary	73
4	Graph Theoretic Approach	77
4.1	Formulation of the Graph Model	78
4.2	Concise Survey of Literature	83
4.2.1	History of the Maximum Flow Problem	83
4.2.2	Solving a Maximum Flow Problem	85
4.2.3	Ford’s Algorithm	87
4.2.4	The Shortest Augmenting Path Algorithm	87
4.2.5	History of the Minimum Cost Flow Problem	93
4.2.6	Solving a Minimum Cost Flow Problem	97
4.2.7	The Successive Shortest Path Algorithm	98
4.3	Algorithmic Fruit Export Solutions	106
4.3.1	Maximum Flow Solutions	106
4.3.2	Minimum Cost Flow Solutions	106
4.4	Chapter Summary	110

5	Mathematical Programming Approach	113
5.1	A Concise History of the Multicommodity Flow Problem	114
5.2	Formulation of the Mathematical Program	114
5.3	Solving the Multicommodity Flow Problems	119
5.4	Maximum Flow Solutions	120
5.4.1	Scenario One	121
5.4.2	Scenario Two	128
5.4.3	Scenario Three	135
5.5	Minimum Cost Flow Model	138
5.6	Minimum Cost Flow Solution	141
5.6.1	Summer Peak Week	141
5.6.2	Winter Peak Week	143
5.7	Chapter Summary	152
6	Conclusion	155
6.1	Thesis Summary	155
6.2	Comparison of Solutions	156
6.3	Suggestions and Recommendations	160
6.4	Possible Future Work	161
		
	References	163
A	Maps of Deciduous Fruit Production regions	183
B	Packhouse and Cold Store Regions	187
C	Origin–Destination Flow Volumes	189
D	Model Abbreviations	193
E	Worked Examples of Algorithmic Implementations	195
E.1	Ford’s Algorithm	195
E.2	Shortest Augmenting Path Algorithm	198
E.3	Successive Shortest Path Algorithm	204
F	Transportation Costs	211

G Computer Codes	217
G.1 The Program <i>FruitFly</i>	217
G.2 The Program <i>Convert</i>	238
H Compact Disc Contents	255

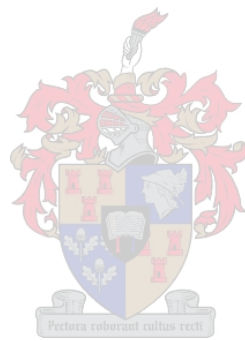


List of Figures

2.1	Fruit seasons	10
2.2	Pallets	11
2.3	Producing regions: Grapes	12
2.4	Producing regions: Stone	13
2.5	Producing regions: Pome	14
2.6	Producing regions: Hard citrus	15
2.7	Producing regions: Soft citrus	16
2.8	Producing regions: Avocado	17
2.9	Producing regions: Mango	18
2.10	Producing regions: Litchi	19
2.11	Typical Fruit Supply Chain	19
2.12	Tautliner and genset transport	20
2.13	Citrus road transport volumes	21
2.14	Deciduous fruit road transport volumes	25
2.15	Deciduous fruit road transport volumes: Western Cape	25
2.16	Subtropical fruit road transport volumes	27
2.17	South African rail network	29
2.18	Citrus rail transport volumes	30
2.19	Container and conventional vessels	31
2.20	Methods of improving fruit quality	34
2.21	Grape packing facility operations	36
2.22	Porthole and integral containers	42
2.23	Porthole refrigeration structure	43
2.24	Container terminal equipment	44
3.1	Packhouse questionnaire	50
3.2	Cold store questionnaire	50

3.3	Grapes export volumes per region	58
3.4	Hard citrus export volumes per region	59
3.5	Soft citrus export volumes per region	60
3.6	Export volumes of all fruit types	60
3.7	Estimated South African fruit export in 2003	62
3.8	Estimated volumes exported per week per port	63
3.9	The utilisation of the port infrastructure in Cape Town	72
3.10	The utilisation of the port infrastructure in Durban	73
3.11	The utilisation of the port infrastructure in Port Elizabeth	74
3.12	The utilisation of the port infrastructure in Maputo	75
4.1	Simplified representation of the fruit export infrastructure of SA	79
4.2	Indication of the locations of the 31 fruit-growing regions of SA	80
4.3	Flow-limiting technique applied to packhouses, cold stores and terminals	81
4.4	Simplification of the sub-graph representing the port of Cape Town	82
4.5	Original network and residual network comparison	89
4.6	Methods of converting multigraphs to graphs	89
4.7	An example where no feasible flow is possible	90
4.8	Transformation of arc with lower bound	98
5.1	Simplification of the network representing the KwaZulu-Natal region	116
6.1	A danger of not considering the multicommodity flow approach	157
A.1	Geographical locations of deciduous farmers	183
A.2	Geographical locations of grape farmers	184
A.3	Geographical locations of stone farmers	184
A.4	Geographical locations of pome farmers	185
A.5	Estimated volumes of deciduous fruit produced per province	185
E.1	Ford's Algorithm example: Original graph	195
E.2	Ford's Algorithm example: First iteration	196
E.3	Ford's Algorithm example: Second iteration	196
E.4	Ford's Algorithm example: Third iteration	197
E.5	Ford's Algorithm example: Fourth iteration	197
E.6	Ford's Algorithm example: Fifth iteration	197
E.7	Ford's Algorithm example: Sixth iteration	198

E.8	Ford's Algorithm example: Seventh iteration	198
E.9	Shortest Augmenting Path Algorithm: First iteration	199
E.10	Shortest Augmenting Path Algorithm: Second iteration	199
E.11	Shortest Augmenting Path Algorithm: Third iteration	200
E.12	Shortest Augmenting Path Algorithm: Fourth iteration	200
E.13	Shortest Augmenting Path Algorithm: Fifth iteration	201
E.14	Shortest Augmenting Path Algorithm: Sixth iteration	201
E.15	Shortest Augmenting Path Algorithm: Seventh iteration	202
E.16	Shortest Augmenting Path Algorithm: Eighth iteration	203
E.17	Shortest Augmenting Path Algorithm: Final flow	203
E.18	Successive Shortest Path Algorithm: Original network	204
E.19	Successive Shortest Path Algorithm: Network preparation	204
E.20	Successive Shortest Path Algorithm: First iteration	205
E.21	Successive Shortest Path Algorithm: Second iteration	206
E.22	Successive Shortest Path Algorithm: Third iteration	206
E.23	Successive Shortest Path Algorithm: Fourth iteration	207
E.24	Successive Shortest Path Algorithm: Fifth iteration	207
E.25	Successive Shortest Path Algorithm: Sixth iteration	208
E.26	Successive Shortest Path Algorithm: Seventh iteration	208
E.27	Successive Shortest Path Algorithm: Eighth iteration	209
E.28	Successive Shortest Path Algorithm: Final flow	209
F.1	Line of best fit for provided unrefrigerated fruit road transport costs	211
F.2	Line of best fit for provided containerised fruit road transport costs	212
F.3	Line of best fit for provided refrigerated fruit road transport costs	212



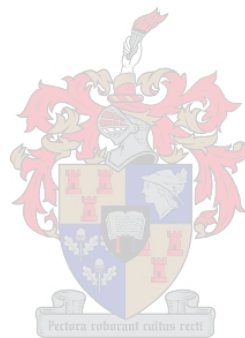
List of Tables

1.1	Core team coordinating the project	2
2.1	Citrus fruit exported from South Africa	8
2.2	Deciduous fruit exported from South Africa	8
2.3	Subtropical fruit exported from South Africa	9
2.4	Optimal fruit storage temperatures	39
2.5	Costs of cooling fruit to the correct temperature	44
2.6	The FPT fruit handling costs for 2005	46
2.7	The estimated SAFT fruit handling costs for 2005	47
2.8	Shipping costs	47
3.1	Regional packhouse capacities	52
3.2	Regional cold store capacities	54
3.3	Fruitlog Scenario Model: Grape Production and Export	55
3.4	Fruitlog Scenario Model: Hard Citrus Production and Export	56
3.5	Fruitlog Scenario Model: Soft Citrus Production and Export	56
3.6	Fruitlog Scenario Model: Pome Production and Export	56
3.7	Fruitlog Scenario Model: Stone Production and Export	57
3.8	Fruitlog Scenario Model: Subtropical Production and Export	57
3.9	Fruitlog Scenario Model: Total Production and Export	57
3.10	Fruit export during peak summer and winter peaks	61
3.11	Estimated volumes exported per week per port	64
3.12	The loading and stack capacities for fruit containers	66
3.13	Weekly loading and stack capacities for fruit export via container terminals	68
3.14	Break-bulk and container loading capacities at FPT and SAFT	69
4.1	Applications of the maximum flow problem	85
4.2	Applications of the minimum cost flow problem	96

4.3	Solution for the maximum flow problem: Largest regional export volumes	107
4.4	Solution for the maximum flow problem: Export volumes from terminals	108
4.5	Standard solution to the minimum cost flow problem: Port infrastructure	109
4.6	Solution to the minimum cost flow problem: Unrestricted port infrastructure . .	109
5.1	Some applications of the multicommodity flow problem	115
5.2	Fruit groups	116
5.3	Maximum flow results for Scenario One: General	121
5.4	Maximum flow results for Scenario One: General (Regional Restrictions)	122
5.5	Maximum flow results for Scenario One: Grapes	123
5.6	Maximum flow results for Scenario One: Stone	123
5.7	Bottlenecks of the flow of stone fruit in Scenario One: Stone	124
5.8	Maximum flow results for Scenario One: Pome	125
5.9	Bottlenecks of the flow of pome fruit in Scenario One: Pome	125
5.10	Maximum flow results for Scenario One: Citrus	126
5.11	Maximum flow results for Scenario One: Mango	127
5.12	Bottlenecks of the flow of mango fruit in Scenario One	127
5.13	Maximum flow results for Scenario One: Litchi	128
5.14	Bottlenecks of the flow of litchi fruit in Scenario One: Litchi	128
5.15	Maximum flow solution to Scenario Two	129
5.16	Maximum regional export volumes for Scenario Two	130
5.17	SA fruit export infrastructure bottlenecks: Scenario Two	132
5.18	Fruit types and volumes exported: Break-bulk	134
5.19	Fruit types and volumes exported: Containers	135
5.20	Maximum flow solution to Scenario Three	136
5.21	Production information from maximum flow solution for Scenario Three	137
5.22	Fruit types and volumes exported: Break-bulk	139
5.23	Fruit types and volumes exported: Containers	140
5.24	Minimum cost flow solution (week 7): Break-bulk	141
5.25	Minimum cost flow solution (week 7): Containers	142
5.26	Minimum cost flow solution (week 7): Only Cape Town Infrastructure	144
5.27	Shadow prices of the minimum fruit flow constraints (week 29)	145
5.28	Minimum cost flow solution (week 29): Break-bulk	146
5.29	Minimum cost flow solution (week 29): Containers	147
5.30	Corrected minimum cost flow solution (week 29): Break-bulk	148

5.31	Corrected minimum cost flow solution (week 29): Containers	149
5.32	Solution to minimum cost flow problem (no port capacities): Break-bulk	150
5.33	Solution to minimum cost flow problem (no port capacities): Containers	151
5.34	Shadow prices of the minimum fruit flow constraints (week 29)	152
6.1	Minimum cost flow solutions: GT vs MP approaches	159
C.1	Origin–Destination Volumes: Grapes	189
C.2	Origin–Destination Volumes: Total Citrus	190
C.3	Origin–Destination Volumes: Total Pome	190
C.4	Origin–Destination Volumes: Total Stone	191
C.5	Origin–Destination Volumes: Total Subtropical	191
F.1	Road transportation costs for unrefrigerated fruit	213
F.2	Road transportation costs for containerised fruit	214
F.3	Road transportation costs for fruit in refrigerated vehicles	215
F.4	Rail transportation costs for unrefrigerated hard citrus	215
G.1	Spreadsheet format for <i>Convert</i>	238

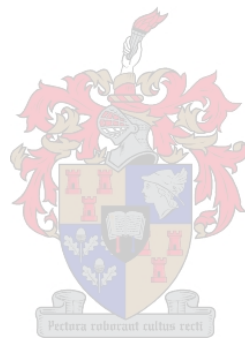




List of Algorithms

1	Ford's algorithm	88
2	Shortest Augmenting Path Algorithm	92
3	Successive Shortest Path Algorithm	105





Glossary

A-check An administration inspection of a vehicle's consignment notes performed at container *terminals*.

admissible arc An *arc* (i, j) for which the *residual capacity* is larger than zero and the *distance label* of node i is one larger than the *distance label* of node j .

admissible path A *path* along which all *arcs* are admissible.

airfreight The transport of goods by aeroplane.

algorithm An ordered sequence of procedural operations for solving a problem within a finite number of steps.

ambient temperature The temperature of the surroundings, used to indicate atmospheric temperature not inside a cooled chamber. Also refers to the temperature of uncooled fruit.

arc The elements of the *arc set* of a *graph*.

arc set A (possibly empty) finite set of ordered two-element subsets of the *vertex set* of a *graph*. An *edge* which allows flow in one direction only.

augmenting path A *path* along which flow may be increased by a certain positive value without violating any flow-restricting constraints.

Blue Store The structure in Cape Town harbour that stores and cools the contents of *porthole containers* (to be decommissioned).

bottleneck A limiting point in a supply chain that prevents more flow from taking place through the supply chain as a whole.

citrus fruit A fruit grouping that includes oranges, grapefruit, limes, lemons and mandarins.

cold store A building with refrigerated rooms for the refrigeration of products such as fruit.

consignment notes The documentation required by port authorities and *terminal* operators to handle and export fruit.

container vessel A vessel that may only transport containerised goods.

controlled atmosphere cooling Refrigeration by which only selected gasses are allowed into the cooling chamber; fruit may thus be cooled for longer periods than under cooling with typical atmospheric gas mixtures (*abbr.*: CA).

- conventional vessel** A vessel that transports fruit in bulk and sometimes also smaller volumes in containers.
- Cost, Insurance, Freight** A sales term or phrase for the cost/value of the commodity, the insurance and the sea transport cost. All these costs are carried by the seller and are recovered from the buyer. The choice of shipping lines is that of the seller (*abbr.*: CIF).
- cycle** A closed *path* with at least three *arcs* and *nodes*, in the sense that the initial *node* and terminal *node* of the *path* are the same *node*.
- deciduous fruit** A fruit grouping that includes apples, pears, grapes, apricots, plums, peaches and nectarines.
- digraph** A digraph is an ordered pair of sets $G = (V, A)$, where V is a set of vertices and A is a set of ordered pairs (called *arcs*) of *vertices* of V .
- distance label** The label on a *vertex* or *node* indicating its distance in terms of *arc* weights from an initial *node* along the *shortest path*.
- edge** The elements of the *edge set* of a *graph*.
- edge set** A (possibly empty) finite set of unordered two-element subsets of the *vertex set* of a *graph*.
- Euro-Retailer Produce Working Group Good Agricultural Practices** A set of standards and procedures set by the Euro-Retailer Produce Working Group (EUREP) for Good Agricultural Practices (GAP) that should be enforced by producers and other *role-players* in the fresh fruit export industry (*abbr.*: EUREPGAP).
- feasible flow** A flow of commodities through a *network* from a *source* to a *sink* that does not violate any conservation of flow laws or capacity constraints.
- fixed costs** The costs incurred by a transport company, regardless of whether their vehicle is moving or not. Also called *standing costs*.
- flow lower bound** The minimum volume of fruit that may flow along an *arc* of a flow *network* per time unit.
- flow upper bound** The maximum volume of fruit that may flow along an *arc* of a flow *network* per time unit.
- Free-On-Board terms** A sales term used when a price is quoted for all transport costs up to and including the loading of vessel. The buyer is responsible for any further costs (including shipping) and insurance of the cargo, and hence may decide on the choice of shipping line (*abbr.*: FOB).
- Fresh Produce Terminals** A company that operates cold storage and fruit-loading facilities in Southern African ports, mainly dealing with break-bulk fruit export — with some volumes of fruit exported in containers on the decks of the conventional vessels (*abbr.*: FPT).
- Fruit South Africa** An umbrella body overseeing the activities of most fruit growers' associations (*abbr.*: FSA).
- genset** A mode of transport in which a generator is attached to an *integral container* to supply it with electricity in order to refrigerate the contents.

graph A combinatorial object $G = (V, E)$ consisting of a non-empty, finite set V of combinatorial objects called *vertices* as well as a (possibly empty) finite set E of ordered or unordered pairs of distinct elements called *edges*. If the set E is a finite set of ordered pairs, the object is called a *digraph*. The *graphs* used in the models in this thesis are all *digraphs*.

integral container A container that has its own cooling system attached to it and requires a source of electricity to operate. It may be either a *regular atmosphere* container, or a *controlled atmosphere* container.

mass balance constraint The conservation of flow constraint which states that all flow into a *node* must equal the flow out of the *node* added to the supply (positive for *source nodes* or negative for *sink nodes*).

multicommodity flow The flow of two or more commodity types through the same *network*.

network A *graph* model that allows volumes of a commodity to flow from one *node* to another across a combination of *arcs*.

node An element of the *node set* or *vertex set* of a *network*.

node set See *vertex set*.

node potential The amount by which the *mass balance constraint* at a single *node* is violated by *pseudoflows*.

objective function The expression that is to be maximised or minimised in an optimisation problem by adjusting the values of variables in such a way that none of the constraints are violated.

P-check A physical inspection of vehicles and their consignments performed at container *terminals*.

packhouse A building in which fruit is packed into cartons, which are, in turn, packed onto *pallets*.

pallet A wood or plastic structure on which fruit cartons (or other goods) are packed or stacked, typically with dimensions $1\,200\text{mm} \times 1\,000\text{mm} \times 160\text{mm}$ and with spaces at the bottom to make transportation by forklift possible. In this thesis used in reference to a pallet with fruit cartons stacked on it.

path A *walk* in a *graph* (or *network*) with the property that no *vertex* (or *node*) is repeated.

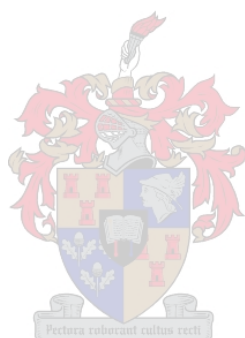
Perishable Product Export Control Board The organisation whose task it is to ensure that all fruit due for export conforms to the necessary export protocols set by the target market (*abbr.*: PPECB).

plug-in point A source of electricity for *integral containers* when they are being stored, usually associated with a space in a *terminal* (typically at a *stack*) for that container.

polynomial-time algorithm An *algorithm* with a worst-case time complexity bounded from above by a polynomial function of the number of *nodes* n and the number of *arcs* m , with the logarithm of the maximum upper bound value on all *arcs* ($\log U$) and the logarithm of the maximum cost value associated with all *arcs* ($\log C$) as possible coefficients of the terms of the function, e.g. $O(n^2)$, $O(n + m \log U)$ or $O(nm + n^2/m + m^2 \log C)$.

- porthole container** An insulated container with two ports on one end that allow cold air to enter and warm air to be sucked out (in the process of being discontinued).
- pseudopolynomial-time algorithm** An *algorithm* with a worst-case time complexity bounded from above by a polynomial function of the number of *nodes* n and the number of *arcs* m , with the maximum upper bound value on all *arcs* (U) and the maximum cost value associated with all *arcs* (C) as possible coefficients of the terms of the function, e.g. $O(n + mU)$ or $O(nm + n^2/m + m^2C)$.
- pseudoflow** A flow through a network satisfying only the capacity and non-negativity flow constraints, perhaps violating the *mass balance constraints* (conservation of flow law) at some nodes.
- punnet** A small light basket used as a measure for fruits.
- reduced cost** The cost of a unit of flow crossing an *arc*, less the *node potential* at the initial *node*, plus the *node potential* of the *terminal node*.
- reefer** A refrigerated container (*integral* or *porthole container*), vehicle or vessel.
- reefer container** Either a *porthole* or an *integral container*.
- refractometer** A device used to measure the concentration of sugars, vitamins and minerals in fruit.
- regular atmosphere cooling** Refrigeration by which air is used as the gas in the cooling chamber (*abbr.*: RA).
- residual capacity** The difference between the *flow upper bound* and the flow across an *arc*.
- residual network** A network associated with a constrained flow problem in which upper and lower bounds on *arcs* have been removed and replaced with the *residual capacities*, indicating how much capacity remains for flow along an *arc* (simply the subtraction of flow along an *arc* from the *arc*'s upper bound).
- role-player** Any person, company, grouping or association that forms part of the fresh fruit export industry.
- sensitivity analysis** The process of determining the effect on the *objective function* in an optimisation problem when affecting changes to constraints or parameter values.
- shadow price** Referring to a specific constraint, the amount by which the *objective function* value changes when one unit is added to (subtracted from) an upper bound (a lower bound).
- shortest path** A *path* in a flow *network* of shortest length between two specified *nodes* in terms of accumulated *arc* weights/costs or numbers of *arcs* in the *path*.
- sink** A *node* in a flow *network* that has a positive demand (sometimes referred to as a negative supply) for a commodity.
- SmartFresh** A gas added to a cooling chamber at a cold store allowing certain fruits to be cooled for longer periods than usually possible under *regular* or *controlled atmosphere* conditions.
- source** A *node* in a flow *network* that has a positive supply of a commodity.

- South African Fruit Terminals** One of the two major *terminal* operators specialising in break-bulk fruit, with *terminals* in the ports of Cape Town and Durban (*abbr.*: SAFT).
- South African Port Operations** The subsidiary of Transnet tasked with the operation of the ports in South Africa (*abbr.*: SAPO).
- stack** An area near or in the container *terminal*, where containers are stored until they are due to be loaded onto a vessel or vehicle. In this thesis, only the section of a *stack* is considered where *plug-in points* are available for *integral containers*.
- standing costs** See *fixed costs*.
- stevedoring** The service offered by *terminal* operators to load and secure a shipment of fruit or containers onto a vessel.
- strongly polynomial-time algorithm** An *algorithm* subject to a worst-case time complexity bounded from above by a polynomial function of the number of nodes n and the number of arcs m , e.g. $O(n^2)$, $O(n + m)$ or $O(nm + n^2/m + m^2)$.
- subtropical fruit** Fruit typically grown in subtropical regions, including mangoes, avocados, litchis, bananas and pineapples.
- super-cold store** A (virtual) collection of a number of (real) cold stores in a region (typically handling the same fruit types and grouped together for modelling purposes).
- super-packhouse** A (virtual) collection of a number of (real) packhouses in a region (typically handling the same fruit types and grouped together for modelling purposes).
- super-sink** A (virtual) collection of a number of (real) *sinks* for the same commodity into a single *sink* for that commodity (grouped together for modelling purposes).
- super-source** A (virtual) collection of a number of (real) *sources* for the same commodity into a single *source* for that commodity (grouped together for modelling purposes).
- temperature control protocols** The temperatures at which fruit must be transported or to which fruit must be cooled in order to be cleared for export, which may include changes in temperature for ripening or colouration purposes during transit or storage.
- terminal** An area of a port where the storage of cargo and loading of cargo onto vessels takes place.
- thermocouple** A special wire inserted into fruit in order to aid in the monitoring of fruit temperature.
- variable costs** The costs incurred by a transporter due to the fact that their vehicles are moving (includes fuel, oil and tyre costs).
- vertex** An element of the *node set* or *vertex set* of a *graph*.
- vertex set** A non-empty, finite set of single elements representing terminal or intersection points in a *graph*.
- walk** An alternating sequence of *vertices* (or *nodes*) and *edges* (or *arcs*) that begins and ends with a *vertex* (or *node*).



List of Acronyms

CSA Citrus South Africa

CSIR Council for Scientific and Industrial Research

CTCT Cape Town Container Terminal

CA Controlled atmosphere cooling

CGA Citrus Growers Association

CIAMD Centre for International Agricultural Marketing and Development

CIF Cost, Insurance, Freight

DFPT Deciduous Fruit Producers' Trust

EUREPGAP Euro-Retailer Produce Working Group Good Agricultural Practices

FOB Free-On-Board

FPEF Fresh Produce Exporters' Forum

FPT Fresh Produce Terminals

Fruitlog Fruit Logistics Infrastructure Capacity Optimisation Study

FSA Fruit South Africa

GDP Gross Domestic Product

GPS Global Positioning System

IBM International Business Machine

IFORS International Federation of Operations Research Societies

MPS Mathematical Programming System

NPA National Ports Authority

ORSSA Operations Research Society of South Africa

PPECB Perishable Product Export Control Board

RA Regular atmosphere cooling

SAAGA South African Avocado Growers' Association



SAAPPA South African Apples and Pears Producers' Association

SAFT South African Fruit Terminals

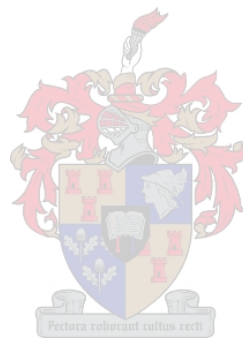
SALGA South African Litchi Growers' Association

SAMGA South African Mango Growers' Association

SAPO South African Port Operations

UK United Kingdom

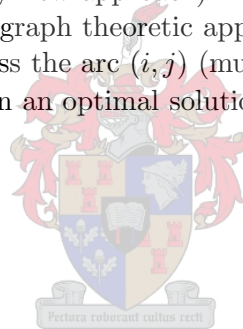
USA United States of America



List of Reserved Symbols

χ	The fraction of ambient hard citrus fruit allowed on vessels (multicommodity flow approach)
δ	The maximum augmentable flow along a path (graph theoretic approach)
$\pi(i)$	The node potential of node i , i.e. the distance from the source using reduced costs (graph theoretic minimum cost flow algorithm)
ϕ	The fraction of fruit that is lost during transportation along an arc (multicommodity flow approach)
\mathcal{A}	The set of all arcs in a network (common to all approaches)
$b(i)$	The supply (if positive) or demand (if negative) at node i (common to all approaches)
$b^k(i)$	The supply of commodity k (if positive) or demand of commodity k (if negative) at node i (multicommodity flow approach)
c_{ij}	The weight of the arc (i, j) or the cost of a unit of flow to pass along the arc
c_{ij}^d	The cost of one commodity unit flowing across an arc (i, j) (graph theoretic minimum cost flow algorithm)
c_{ij}^π	The reduced cost of the arc (i, j) , using node potentials (minimum cost flow algorithm)
$d(i)$	The distance label of node i (graph theoretic approach)
\mathcal{D}	The set of all nodes with a deficit (minimum cost flow algorithm)
$e(i)$	The excess label of node i , i.e. the amount by which the mass balance constraint has been violated (minimum cost flow algorithm)
\mathcal{E}	The set of all nodes with an excess (minimum cost flow algorithm)
G	A graph (graph theoretic approach)
\mathcal{K}	The set of all commodities (multicommodity flow approach)
l_{ij}	The lower bound on the flow across the arc (i, j) (minimum cost flow algorithm)
L_{ij}	The lower bound on the total flow of all commodities across the arc (i, j) (mathematical programming approach)
L_{ij}^k	The lower bound on the flow of commodity k across the arc (i, j) (multicommodity flow approach)
m	The number of arcs in a network (algorithm worst-case running time evaluation for graph theoretic approach)
n	The number of nodes in a network (algorithm worst-case running time evaluation for graph theoretic approach)
\mathcal{N}	The set of all nodes in a graph/network (graph theoretic approach)
$\mathcal{N}x_{ij}^k$	The net flow of commodity k out of node i (multicommodity flow approach)
$p(i)$	The predecessor of node i on a shortest path between two nodes (graph theoretic approach)
r_{ij}	The residual value of the arc (i, j) (graph theoretic approach)

\mathcal{R}	The set of arcs representing refrigerated fruit transport to the terminal loading areas (multicommodity flow approach)
s	The source node (graph theoretic approach)
\mathcal{S}	The set of all permanently labelled nodes (shortest path algorithm)
$\bar{\mathcal{S}}$	The set of all nodes in \mathcal{N} , but not in \mathcal{S} (shortest path algorithm)
t	The sink node (graph theoretic approach)
\mathcal{T}	The set of arcs that are adjacent to the nodes representing loading areas at the terminals (multicommodity flow approach)
u_{ij}	The upper bound on the flow across the arc (i, j) (graph theoretic approach)
U	The largest value in the set of upper-bounds (algorithm worst-case running time evaluation for graph theoretic methods)
U_j	The upper bound on flow through node j (multicommodity flow approach)
U_j^k	The upper bound on flow of commodity k through node j (multicommodity flow approach)
U_{ij}	The upper bound on all flow across the arc (i, j) (multicommodity flow approach)
U_{ij}^k	The upper bound on the flow of commodity k across the arc (i, j) (multicommodity flow approach)
\mathcal{U}	The set of arcs representing unrefrigerated hard citrus fruit transport to the terminal loading areas (multicommodity flow approach)
x_{ij}	The flow across the arc (i, j) (graph theoretic approach)
x_{ij}^k	The flow of commodity k across the arc (i, j) (multicommodity flow approach)
x_{ij}^*	The flow across the arc (i, j) in an optimal solution (minimum cost flow algorithm)

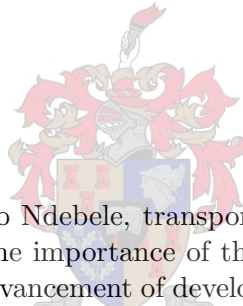


Chapter 1

Introduction

“Developing countries cannot develop unless they are prepared to invest in transport infrastructure. Transport is considered integral to all service provision as well as the supply and marketing of commodities.”

– Sibusiso Ndebele 2003 [244]



The opening quote, from Mr Sibusiso Ndebele, transport MEC of KwaZulu-Natal during the period 1994 – 2004 [288], indicates the importance of the transportation infrastructure to the South African government and the advancement of developing countries. South Africa may well be the most developed of all African countries, but it is still considered a developing country and an emerging market, when compared to North American and most European countries. It is a well established fact that a developing country’s economy may grow more readily if the infrastructure is constantly maintained and improved [128] (in contrast, improvements made to the transport infrastructure of developed countries has a smaller effect, if at all [24]). These observations do not only hold for transportation infrastructure (which is the focus of this thesis), but also for communications infrastructure, and infrastructures for health, sanitation and security services. For South African agriculture, and especially for agricultural products destined for foreign markets, the transportation infrastructure is vitally important as this increases the efficiency and competitiveness of the fruit supply chain to exporters. Competitive supply chains are increasingly more important than individual links (or companies) being competitive, in the race to capture a greater share of the world market [233]. If roads are in poor condition or vehicles have to wait a considerable time to load or unload, the cost of transport rises and this results in a decrease in income for the farmer. The aim in this thesis is to evaluate and contribute to the effectiveness of the logistics infrastructure used to export South African fresh fruit to foreign markets. It is anticipated that other sectors of the economy may also benefit from any improvement to the fresh fruit export infrastructure.

This thesis forms part of a larger project in supply chain management. The purpose of this introductory chapter is to describe this larger project in some detail (in §1.1). The formulation of the specific sub-problem to be studied in this thesis is also given (in §1.2) and a general preview of the layout of the thesis is presented in §1.3.

1.1 Fruit Logistics Infrastructure Capacity Optimisation Study

The distribution problem to be studied in this thesis forms a relatively small part of a larger project that was coordinated by Dr FE van Dyk from the Transportek division of the Council for Scientific and Industrial Research (CSIR) [290]. It is officially called the *Fruit Logistics Infrastructure Capacity Optimisation Study*, or *Fruitlog* for short. The aim of Fruitlog is to “create efficiencies for, and enhance the competitiveness of, the South African fruit industry by improved usage of, and investment in, shared logistics infrastructure.

The project was to make recommendations with respect to:

- the optimal usage of existing fruit logistics infrastructure (packing and cold storage facilities, refrigerated vehicles and road capacity, fruit terminals, etc.) on a national level,
- the use of alternative infrastructure such as rail and ports not traditionally used, and
- investment in additional infrastructure for the future benefit of the SA fruit industry.” [290]

The team that worked on the project consisted of logistics analysts from CSIR - Transportek, who were the project managers, an agricultural economist from Optimal Agricultural Business Systems (OABS) subcontracted by the Deciduous Fruit Producers’ Trust (DFPT) as their information manager, and a team of academics and graduate students from the Departments of Logistics, Applied Mathematics, Industrial Engineering and Agricultural Economics at the University of Stellenbosch.

The core team that coordinated the project is listed in Table 1.1.

Name	Position	Institution
FE van Dyk	Project manager and logistics analyst	CSIR – Transportek
EL Maspero	Logistics analyst	CSIR – Transportek
DB Louw	Information manager	OABS – DFPT
NH Louw	Senior lecturer	Department of Logistics University of Stellenbosch

Table 1.1: The names and institutions of the people who constituted the core team coordinating the project, *Fruitlog*.

The Fruit Logistics Infrastructure Capacity Optimisation Study was divided into 4 phases. Each of these phases are described separately.

Phase 1: Promoting effective and efficient fruit trade logistics operating practices

The main purpose of the first phase was to educate all role-players in the fresh fruit supply chain of the importance of effective and efficient logistics operating practices. It had previously been found that bottlenecks and costly delays had been the result of a lack of accurate and timely information exchange. Thus, the Information and Communication Procedure (ICP) was deemed necessary to encourage efficient logistics operations.

The grape industry was chosen to lead the research into the ICP as grapes require rapid export (as they are very perishable) and have the highest value of the deciduous fruit types. Through meetings with role-players in workshops, sector focus groups and on a personal basis, members of the Fruitlog group developed the ICP. This included the details of the information that would need to be exchanged between role-players and responsibilities of the parties concerned.

Phase 2a: Gathering of fruit industry data

Due to the lack of a consolidated data set of fruit industry information, identical data would often be collected for many studies performed in the industry. This information included contact details for cold stores and packhouses, production volumes for all fruit types (which was generally available from the producer associations) and capacities of packhouses and cold stores (available only from each individual facility). This led to an extensive exercise in data collection, during which questionnaires were sent to all packhouses and cold stores, which formed the basis for a database of national capacity in those facilities. Fruit transportation tariffs were also collected for many of the commonly used routes for fruit export. Some of the data was used to construct maps of deciduous fruit growing regions and the routes used to export that fruit.

A further report was compiled including an overview of citrus, deciduous and subtropical fruits. It served as a comprehensive summary of the domestic and international fruit industries. It included marketing channels, a fruit value chain and discussions regarding the production, export, consumption, production value and international competitiveness of the three kinds of fruit from a South African viewpoint.

Phase 2b: Modifying information and communication procedures

During this phase the ICP was adapted for the role-players dealing with pome and stone fruit.

Phase 3: Conducting feasibility studies and developing models

This phase of Fruitlog focussed on feasibility studies and the development of models for the fruit logistics infrastructure utilisation, as well as the simulation of some possible scenarios. The studies that occurred during this phase included:

- A transaction cost analysis of grape and citrus supply chains.
- A methodology for medium- and long-term crop forecasts in order to improve accuracy.
- Simulation models of Fresh Produce Terminals in Durban and Cape Town ports.
- A multicommodity maximum flow optimisation model of the fruit export infrastructure (a topic in this thesis).
- The use of road versus rail, alternative ports not currently used and the long-term development plans of the various South African ports were investigated.

This information proved crucial during phase 4.

Phase 4: Analysing scenarios

A crop forecast for 2010 allowed for a study into whether investment in logistics infrastructure will be required in the foreseeable future. This included an analysis of fruit export volumes on the export routes and through the ports. A spreadsheet model was developed to determine the current and potential future utilisation of the various conventional and container terminals that load fruit for export.

1.2 Problem Formulation for this Thesis

There are two aims of the work in this thesis. The first is to find the maximum possible flow of fruit through the fruit export infrastructure of South Africa. This infrastructure includes all roads, railways, packhouses, cold stores, stacks and terminals that handle fresh fruit for export. The second is to determine the minimum cost possible to transport fruit from the packhouses to the ports and load it onto a vessel. For this to be possible, there are some objectives that must be completed before the maximum flow and minimum cost flow of fruit through the South African fruit industry can be found. These objectives are:

- I Determine the cost of fruit transport, cooling at cold stores, handling at ports and shipping to foreign ports.
- II Find the capacities of all packhouses and cold stores in South Africa handling fresh fruit.
- III Determine the capacity of all terminal infrastructure, including maximum weekly loading rates.
- IV Find the fruit export volumes for 2003.
- V Determine the 2003 port infrastructure utilisation peaks.
- VI Find the maximum flow of fruit (seen as a single commodity) through the South African fruit export infrastructure using an appropriate graph theoretic algorithm.
- VII Find the minimum cost flow for the 2003 volume of fruit (seen as a single commodity) exported from South Africa using an appropriate graph theoretic algorithm.
- VIII Formulate a mathematical programming model for the maximum flow and minimum cost flow problems, distinguishing between fruit types (*i.e.* a multicommodity flow approach).
- IX Compare the results of the graph theoretic approach and those of the mathematical programming approach.
- X Make suggestions regarding:
 - (a) current port operations, and
 - (b) port expansion possibilities.

Once all ten objectives have been completed, the maximum possible flow through the fruit export infrastructure can be found, as well as the minimum cost flow for the 2003 volumes of exported fruit. Throughout the rest of the thesis, the reader will be informed of when each of the objective have been completed.

1.3 Thesis Layout

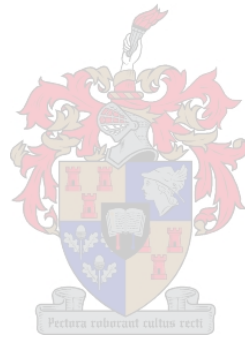
In the second chapter of the thesis, the South African fruit industry is discussed in some detail. This includes the fruits exported from South Africa, when they are in season and the areas in which they grow. The export process is also discussed in this chapter, including the methods and processes involved with the various modes of transportation of fruit from farms to foreign ports. At the end of the second chapter, the costs associated with all the different transport modes, packhouses and cold stores are estimated and summarised.

The third chapter is dedicated to determining the capacities and efficiencies of all infrastructure components that handle fruit in the production regions and the four fruit-exporting ports of Southern Africa (Cape Town, Durban, Port Elizabeth and Maputo). This chapter is based on a large data gathering task, and work undertaken by the author in collaboration with Mr J Havenga and Ms I Hobbs from the Stellenbosch Business School, Dr FE van Dyk from the CSIR, Dr DB Louw from the DFPT and Dr NH Louw from the Department of Logistics at Stellenbosch University. A model called the *Fruitlog Scenario Model* was created in *Microsoft Excel* that allowed various scenarios at the terminals to be studied and evaluated.

Graph theoretic approaches towards modelling the South African fruit export infrastructure are considered in Chapter 4. A brief history of the maximum flow and minimum cost flow problems are presented, along with a discussion of possible algorithms that are traditionally used to solve such problems. The specific algorithms that were used to solve the shortest path, maximum flow and minimum cost flow problems in this thesis are presented and discussed in detail in Chapter 4. All results gained by applying these methods to the South African fresh fruit industry supply chain are presented.

Mathematical programming approaches are discussed in Chapter 5 and some applications of multicommodity flow problems are considered. The maximum flow and minimum cost flow problems are considered again, this time focussing on multicommodity flows. All results gained by applying the multicommodity maximum flow and multicommodity minimum cost flow problems to the South African fresh fruit export infrastructure are presented.

In the final chapter (Chapter 6), the results from the two approaches of Chapters 4 and 5 are discussed, along with the strengths and weaknesses of both the graph theoretic and mathematical programming methods. Strengths and weaknesses in the South African fruit export infrastructure are also presented and recommendations made to improve the perceived problems in the infrastructure. Ideas for future work as well as a brief summary of what has been achieved conclude this thesis.



Chapter 2

The South African Fruit Industry

“Perfect as the wing of a bird may be, it will never enable the bird to fly unsupported by the air. Facts are the air of science. Without them a man of science can never rise.”

– Ivan Pavlov [220]

Agriculture in South Africa is a small part of the national economy, accounting for approximately 3.9% of the Gross Domestic Product (GDP) in 2003 [263]. There are approximately 2 600 farmers growing deciduous fruits (e.g. grapes, apples, pears, plums, nectarines and apricots), approximately 3 500 citrus farmers (growing crops such as oranges, grapefruit, mandarins, lemons and limes) and approximately 1 600 subtropical fruit growers (who grow avocados, mangoes, pineapples and macadamia nuts along with smaller quantities of bananas, litchis, papayas and ginger for export). About 350 000 people are employed on these farms (including all workers involved at packing facilities and cold stores). If the dependants of these workers are considered, approximately two million South Africans are supported by this industry [290]. That is almost 5% of South Africa’s population. In 2001 South Africa exported 532 295.5 metric tons of deciduous fruits alone, resulting in a gross income of just over R5.1 billion [49]. South Africa was ranked third in the world for total citrus export, after only Spain and the USA [48, 229]. This is a significant achievement (and is an indication of the quality of fruit produced), because South Africa was only the eleventh largest producer of citrus in 2001 [48] and eighth largest in 2002 [229]. To put the size of the fruit industry in relation to other agricultural practices, deciduous fruit accounts for 27%, citrus fruit for 17% and subtropical fruit for 1% of all agricultural export earnings in South Africa. Thus fruit accounts for 45% of all agricultural export from South Africa.

It is thus important to the country as a whole that the fruit industry is managed efficiently. The industry was regulated for many years until 1996, but deregulation occurred in 1997 when the Citrus Board and Deciduous Fruit Board disbanded [40]. This was followed by a significant increase in private export enterprises aimed at exploiting the newly deregulated market. Since the value of the South African rand generally tends to decline compared to the values of major currencies like the US dollar, UK Stirling and the Euro, export is a lucrative business. The combination of deregulation, an increase in the number of exporters and the flooding of foreign markets with class 2 fruit (which has now been banned from Europe [40]) resulted in a dramatic lowering of the prices of fruit, leading to the liquidation of many farmers, the disappearance of smaller exporters and a tarnishing of the reputation South Africa enjoyed as a producer of

quality fruit [40]. Now there are strict guidelines for exporters, so that the same situation does not arise again [191].

2.1 Fruit Produced

Many types of fruit are grown in South Africa. As mentioned in the introduction, they may be classified into three main categories: deciduous fruit (e.g. grapes, apples, apricots), citrus fruit (e.g. oranges, lemons, grapefruit) and subtropical fruit (e.g. avocados, mangoes, litchis). The citrus types grown in South Africa are shown in Table 2.1, the types of deciduous fruit are tabulated in Table 2.2 and the subtropical fruit types are shown in Table 2.3.

Citrus Type	Variety
Oranges	Valencias, Navels (Navelina, Washington [119]), Deltas, Midknights, Delta Seedless, Navelates, Benny Valencia, Proteas, Tomangoes, Juvalle, Late Navels, Salustiana, Seville, Sanguinellis Soft Citrus, Kumquats, Bella Fruita Orange, Valentines, Moro's, (Shamoutis), Maltaise
Mandarins	Clementines, Satsuma (Imamura, Kuno, Owari [119]), Nules, Novas, Marisol, Clemvilla Soft Citrus, Minneolas, Shamoutis, Exe Soft Citrus, Torro Clementines, Nartjies, Nouvelle, Valley Gold Soft Citrus, Sanguinellis Soft Citrus, Tangerines, Page Mandarin, (Tambors, Empress Mandarin, Temples, Scarlet, Fairchilds, Ellendales, Kiyomi Soft Citrus)
Grapefruit	Star Ruby, Marsh, Rosé, Ruby Red Pomelos, Pomelos, Shaddock Grapefruit, Oroblanco, Melow Gold Pomelos, [Jackson, Nartia, Rio Red, Redblusht]
Lemons and Limes [119]	Eureka, Fino, Genoa, Limoneire, Libon, Verna

Table 2.1: Citrus fruit exported from South Africa [229]. Lists are ranked by the amount that was exported during the 2002 season, in decreasing order. Cultivars listed (in parentheses) were exported during the 2001 season, but not during the 2002 season. Cultivars found in the FSA Handbook [119] but not in the Key Citrus Industry Statistics [48, 229] are also listed [in square parentheses] as they may be newer cultivars. Note that this list is not exhaustive; fruit cultivars listed as “Experimental” were omitted.

Deciduous Fruit	Cultivar
Apples	Golden Delicious, Granny Smith, Royal Gala, Topred, Braeburn, Gala, Starking, Cripps Pink, Pink Lady TM , Fuji, Sundowner, Cripps Red, Starkrimson, Golden Gala, Experimental, Delicious Apples, Jonagold, Oregon Spur Apples, York Imperial, Empire, Splendour, African Red/Carmine, Coxs Orange Pippin, Rome Beauty, Blushed Golden Apple, Canvada, Jonagored, Red Fuji, Dunns Seedling, (Fiesta Apples)
Apricots	Imperial, Supergold, Bebecco, Soldonn, Peeka, Grandir, Bulida, Palsteyn, Alpha, (Empress, Royal), Sugar Gold [119]

Table 2.2: Deciduous fruit exported from South Africa. Lists are ranked by the amount that was exported during the 2001/2002 season, in decreasing order. Cultivars listed (in parentheses) were exported during the 1999/2000 and/or 2000/2001 season, but not during the 2001/2002 season. Cultivars found in the FSA Handbook [119] but not in the Key Deciduous Fruit Statistics [49, 50] are also listed as they may be newer cultivars. Note that this list is not exhaustive; fruit cultivars listed as “Experimental” or “Other” were omitted.

Grapes	Thompson Seedless, Red Globe, Dauphine, Sugraone/Superior Seedless, Barlinka, La Rochelle, Sunred Seedless, Alphonse Lavallée, Waltham Cross, Flame Seedless, Dan Ben Hannah, Bonheur, Regal Seedless, Majestic, Victoria, Prime Seedless, Black Gem, Crimson Seedless, Bien Donne, Muscat Supreme, Queen of the Vineyard, Sonita, Datal Grapes, Black Emperor, Golden Waltham Cross, Erlihane, Barlinka Late Harvest, Rubistar, New Cross, Super Thompson, Red Globe Late Harvest, White Seedless Grapes, Sundance, Centennial Seeded, White Gem, Muscat, Almeria, Bellevue, Italia, Peridot, White Seeded Grapes, Dauphine Late Harvest, Crimson Seedless Late Harvest, Red Sultana, Mistery Grapes, Bonheur Late Harvest, Sunred Seedless Late Harvest, Lady Ann Grapes, Fantasy Seedless, (Amber Dawn, Muscat Hamburg, Muscat Seedless, White Cross, Ronelle, Olivette, Sultana, Dessert Pearl, Rosaki, White Prince, Eclipse Seedless, Canon Hall Grapes, Hanepoot, Black Prince, Henab Turki Grapes, Dawn Seedless, Pirobella, Rosete, Barbarossa, Flaming Tokay Grapes, Blush Waltham Cross, Red Ruby, Golden Pearl, Black Empress, Bailey, Dessert Thompson, Golden Hill, Gros Colmar, Gros Noir, Molinera Gorda, Salba, UNISA)
Peaches	Transvalia, San Pedro, Nova Donna, Bonnigold, Catherina, Fairtime, Culemborg, Earligrande, Oom Sarel, Classic, Suncrest, Keisie, Excellence, Rhodes, Kakamas, Springcrest, Yellow Cling, Experimental, Safari, Western Cling, Orion, Cinderella, Dessert Gold, Duncans Gold, Dessert Pearl, Snowwhite, Golden, Amber, Sonette, Babcock, Western Sun, Goodmans Choice, Albertros, Sunray, Imperani Peach, Floragold, Summer Giant, Woltemade, Don Elite, Boland, Early Alexander, Elberta, Early Dawn, De Wet, Earlibelle, (Nooiens, Pullers Cling, Snowcrest, Malherbe, Neethling, Coronet, Pucelle)
Pears	Packham's Triumph, William Bon Chretien, Forelle, Beurre Bosc, Doyenne Du Comice, Rosemarie, Beurre Hardy, Golden Russet Bosc, Bon Rouge, Vermont Beauty, Flamingo, Sempre Pears, Abate Fetel, Bartlett, Harrow Delight, General Le Clerc, Red d'Anjou, Highland, Victoria Blush, Clapps Favourite, Conference, Kieffer, Josephine de Malines, Beurre Clairgeau, Crimson, Corona, (Blushed Williams, Formosa, Beurre Diel, Glou Morceau, Beurre Superfin, Winter Nelis)
Plums	Songold, Laetitia, Sapphire, Ruby Nel, Harry Pickstone, Angeleno/Suplumsix, Pioneer, Gaviota, Souvenir, Lary Anne, Santa Rosa, Lady Red, Fortune, Casselman, Flavour King, Southern Belle, African Pride, Red Beaut, Ruby Red, Simka, Laroda, Sun Kiss, Apple Plums, Prunes, Roysum, Red Gold, Sundew, Mariposa, Beauty, October Purple, Eldorado, Eclipse, Satsuma, Red Ace, (Cellebration, Chalcot, Lady West, Golden King, Mostert, Methley, Kelsey Plums, President Plums)
Nectarines	Sunlite, Donnarine, Margaret's Pride, Flavortop, Zaigina, Fantasia, Flamekist, August Red, May Glo, Alpine, Fiesta Red, Olympia, Sungrand, Crimson Giant, August Glo, Early Sungrand, Unico, Independence, Armking, September Red, Red Jewel, Nectared-9, Flavorine, Goldmine Nectarine, Nectar, Early River, (Marina, Panamint)
Other exported deciduous fruit: Cherries, Olives [119]	

Table 2.2 (continued): *Deciduous fruit exported from South Africa. Lists are ranked by the amount that was exported during the 2001/2002 season, in decreasing order. Cultivars listed (in parentheses) were exported during the 1999/2000 and/or 2000/2001 season, but not during the 2001/2002 season. Cultivars found in the FSA Handbook [119] but not in the Key Deciduous Fruit Statistics [49, 50] are also listed as they may be newer cultivars. Note that this list is not exhaustive; fruit cultivars listed as "Experimental" or "Other" were omitted.*

Subtropical Fruit	Cultivar
Avocados	Fuerte, Hass, Pinkerton, Ryan, Edranol [266]
Mangoes	Heidi, Keitt, Kent, Sensation, Tommy Atkins, Zill [268]
Litchis	Mauritius, McClean Red
Other exported subtropical fruit: Bananas, Ginger, Macadamias, Melons, Papayas, Queen Pineapples	

Table 2.3: *Subtropical fruit exported from South Africa [119]. Note that this list is not exhaustive.*

2.1.1 Fruit Seasons

Amongst the deciduous fruits, apricots may be picked from November to February, peaches from October to February, nectarines and plums from November to mid-April, grapes from mid-November to mid-June, pears from January to mid-July according to the FSA Handbook [119] or to September according to the FPEF [116] and apples from March to mid-September. Citrus fruit is generally harvested during the cooler months, oranges being ready from early May to November, grapefruit from late March to early October, lemons from late February to mid-November and easy-peelers (soft citrus) from early April to September. Subtropical fruit is different in that it is generally harvested in summer, mangoes being ready from mid-December to mid-April, litchis from November to February, avocados are picked from March to October and pineapples are picked throughout the entire year. Harvesting seasons of the different fruit types are presented graphically in Figure 2.1.



Figure 2.1: Seasons during which various fruit types are harvested in South Africa [116]. Note that some fruit may not be exported and seasons vary slightly by production regions.

However, this is a very generalised picture of fruit harvesting seasons, as many varieties of a certain fruit type have different times of maturity. Harvesting times may even depend on where the fruit is grown. Thompson seedless grapes, for example, are ready to harvest from late November to early February in the Orange River area, while in the Berg River area, they are mature from January to March [119]. The Hex River area has an even shorter harvest period for these grapes; they may only be harvested in January and February [119]. There is a similar

situation with Red Globe grapes: in the Limpopo Province they are harvested from the end of December to the beginning of February, in the Orange River they are harvested from late November to early February, in the Hex River area they are harvested from January to March, while harvesting takes place from January to April in the Berg River area [119]. The same applies to avocados: Fuerte avocados are available from mid to late February in the Limpopo Province and until early August in KwaZulu-Natal. Hass avocados are mature from mid to late May in the Limpopo province to early October in KwaZulu-Natal. Pinkerton avocados are mature in the Limpopo Province in June and ripe in KwaZulu-Natal in July, while Edranol avocados are harvested from June to September and Ryans from August/September in the Limpopo Province to October in KwaZulu-Natal [266].

2.1.2 Fruit Producing Regions

Detailed maps representing the production regions and volumes exported of deciduous fruit may be found in Appendix A. The towns found in the regions represented by two-letter abbreviations on the maps in this section may be found in Appendix B. An attempt has been made to indicate the sizes of the regions by presenting the number of pallets of fruit (see Figure 2.2 for examples of pallets) exported during the time period studied.



(a) Typical wooden pallet [92]

(b) A pallet packed with fruit [14]

Figure 2.2: Pallets make the handling of cartons of fruit easier. A certain number of fruit cartons are typically stacked onto a wooden pallet, which may then be handled as a unit by forklifts. The number of cartons stacked on each pallet depend on the size and weight of individual cartons.

Grapes are mainly grown in the western part of South Africa, with small volumes grown in the Free State, Marble hall, Hoedspruit and Ellisras in the north and east of the country (see Figure 2.3). The largest volume of grapes was exported from the region incorporating De Doorns and Klein Karoo during the 2002/2003 season; when an estimated 98 063 pallets of grapes were exported. The Orange River region exported the second largest volume of approximately 78 663 pallets. The Ceres, Boland and GEVV regions (consisting of farms near the towns of Grabouw, Elgin, Villiersdorp and Vyeboom — hence GEVV) together exported the third largest volume, having exported 72 807 pallets of table grapes during the 2002/2003 season. The fourth largest grape exporting region is a region comprising the Free State, Marble Hall, Hoedspruit and Ellisras; exporting 13 913 pallets during that season. Namibia exported 10 376 pallets during 2002/2003, making it the fifth largest exporter of grapes. The smallest exporting region is a region consisting of Piketberg, Citrusdal and surrounding areas. Only 8 708 pallets of fruit were

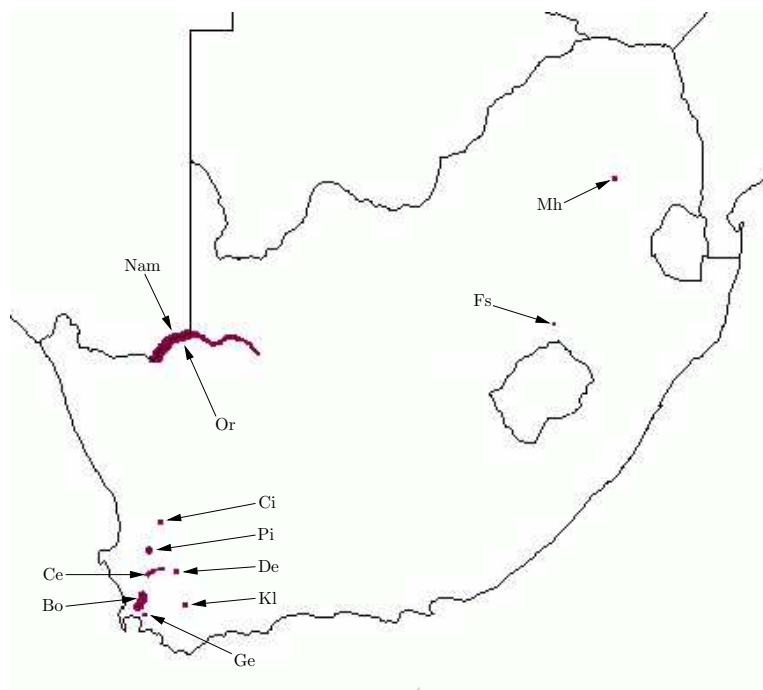


Figure 2.3: A schematic representation of the regions in which grapes are produced and from where they are exported. *Bo* \equiv Boland, *Ce* \equiv Ceres, *Ci* \equiv Citrusdal, *De* \equiv De Doorns, *Fs* \equiv Free State, *Ge* \equiv GEVV, *Kl* \equiv Klein Karoo, *Mh* \equiv Marble Hall, *Nam* \equiv Namibia, *Or* \equiv Orange River, *Pi* \equiv Piketberg.

exported from this region during the 2002/2003 season. Thus an estimated total of 282 512 pallets of table grapes were exported from South Africa during the 2002/2003 season.

Stone fruit is mainly produced in the south-western regions of South Africa (see Figure 2.4), with small volumes produced near Hartswater, Rustenburg and Johannesburg in central and north-east South Africa. The Klein Karoo region was the second largest exporter of stone fruit during the 2002/2003 harvest season, exporting an estimated 14 967 pallets of stone fruit. The largest exporting region in South Africa during that season was the Boland region, from where a total volume of 25 304 pallets of stone fruit were exported. The Langkloof region exported approximately 3 168 pallets, the Patensie region approximately 224 pallets, the Ceres region an estimated 12 326 pallets and 2 180 pallets of stone fruit were exported from the Piketberg region in 2002/2003. Approximately 1 531 pallets of stone fruit were exported from the Hartswater region and 387 pallets from the Rustenburg and Johannesburg regions combined. The GEVV region produced enough stone fruit production for 7 682 pallets to be exported from that region in 2002/2003. According to the information available [203], 297 pallets were exported from the Free State, 856 pallets from the Orange River and 279 from Mpumalanga in 2002/2003. However, according to the packhouse and cold store data collected during the packhouse and cold store capacity data gathering, there are no packhouses nor cold stores that handle stone fruit in those regions. In total approximately 69 200 pallets of stone fruit were therefore exported from South Africa during the 2002/2003 season.

The pome producing regions occur in the south-western, southern and north-eastern parts South Africa (see Figure 2.5). The largest volume of pome fruit is exported from the GEVV region, which was estimated to have exported 204 757 pallets of apples and pears during the

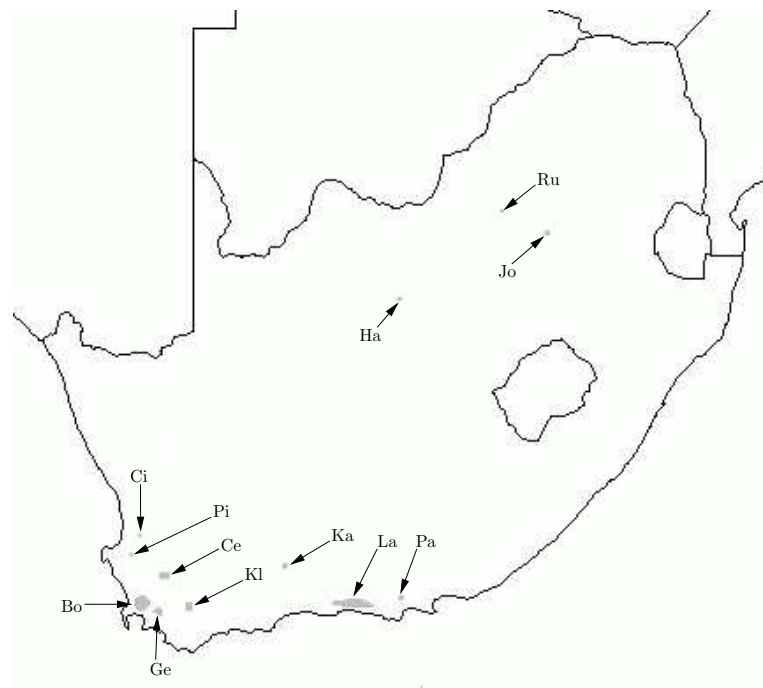


Figure 2.4: A schematic representation of the regions in which stone fruit is produced and from where it is exported. Bo ≡ Boland, Ce ≡ Ceres, Ci ≡ Citrusdal, Ge ≡ GEVV, Ha ≡ Hartswater, Jo ≡ Johannesburg, Ka ≡ Karoo, Kl ≡ Klein Karoo, La ≡ Langkloof, Pa ≡ Patensie, Pi ≡ Piketberg, Ru ≡ Rustenburg.

2002/2003 season. The second largest pome fruit exporting region is the Ceres region, exporting an estimated 140 522 pallets during the same season. The Langkloof region is the third largest exporter of pome fruit, exporting an estimated 68 366 pallets of pome fruit during the 2002/2003 season. Smaller volumes were also exported from the Klein Karoo, Boland, Piketberg/Citrusdal and Free State regions during that season, having exported an estimated 17 681, 17 390, 8 207 and 1 654 pallets of pome fruit respectively.

Hard citrus is grown in the most areas in South Africa; only the Karoo, Namibia, Ermelo, Ellisras and Johannesburg fruit regions did not export any hard citrus fruit during the 2002/2003 season (see Figure 2.6). The largest volume of citrus fruit was exported from the Tzaneen region; 179 376 pallets of hard citrus were exported in 2003. Farmers from the Malelane region exported 132 156 pallets in 2003, making them the second largest exporting region. The Senwes region (Potgietersrus, Rustenburg, Marble Hall, Middelburg) is the third largest, exporting 115 437 pallets during that season. The fourth largest volume is exported from the Sundays River Valley Region, which exported 112 977 pallets in 2003. Fifth largest was the Western Cape (Klein Karoo, GEVV, Boland, Ceres, Piketberg and Citrusdal regions combined), achieving an export volume of 96 491 pallets of hard citrus in 2003. The sixth largest hard citrus exporting region was the Levubu region in the northern parts of the Limpopo province, exporting 87 157 pallets in 2003. The Nkwalini, Pongola and KwaZulu-Natal regions combined form the seventh largest hard citrus exporting region in South Africa, exporting 63 279 pallets in 2003. The Gamtoos Valley near the town of Patensie in the Eastern Cape exported 55 530 pallets in 2003, making it the eighth largest exporter during 2003, closely followed by the Hoedspruit region, which exported 53 862 pallets of hard citrus. Although not indicated in Figure 2.6 (the towns

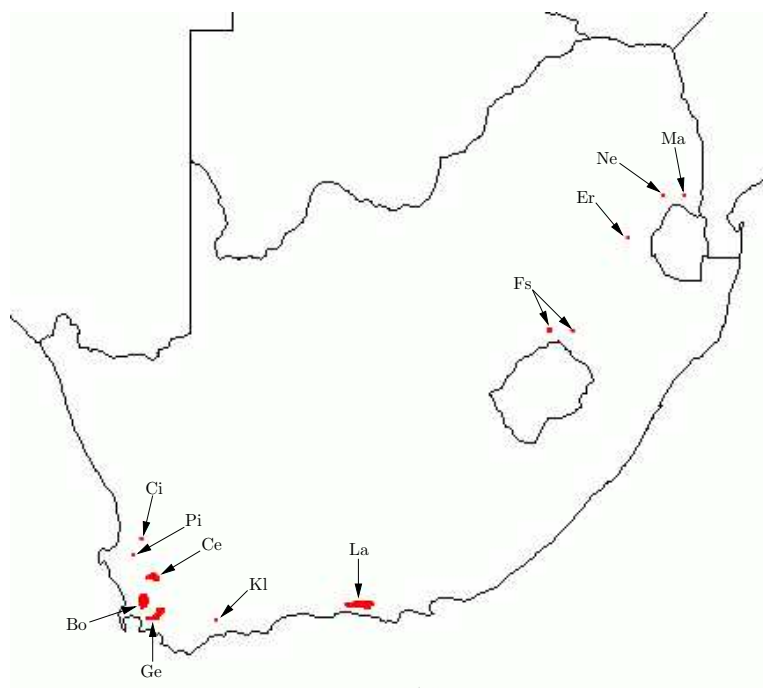


Figure 2.5: A schematic representation of the regions in which pome fruit is produced and from where it is exported. *Bo* \equiv Boland, *Ce* \equiv Ceres, *Ci* \equiv Citrusdal, *Er* \equiv Ermelo, *Fs* \equiv Free State, *Ge* \equiv GEVV, *Kl* \equiv Klein Karoo, *La* \equiv Langkloof, *Ma* \equiv Malelane, *Ne* \equiv Nelspruit, *Pi* \equiv Piketberg.

nearest the citrus plantations could not be identified), Swaziland was the tenth largest exporter of citrus from Southern Africa with an export volume of 25 547 pallets in 2003. The eleventh largest exporting region of South Africa was the Katrivier region in the Eastern Cape Midlands, which exported 11 878 pallets of fruit in 2003. Zimbabwe, also not indicated on the map in Figure 2.6, exported 4937 pallets of hard citrus, followed by the Langkloof and Hartswater regions with export volumes of 94 and 21 pallets in 2003 respectively. All the hard citrus fruit exported from these regions, together with 6 286 pallets of citrus for which the source region could not be identified, resulted in a total volume of 971 338 pallets of hard citrus exported from South Africa in 2003.

Soft citrus fruit is also grown over most of South Africa (see Figure 2.7). The largest volume comes from the Western Cape, a region comprising the smaller Citrusdal, Ceres, Boland, GEVV and Klein Karoo regions. During the 2003 season 33 587 pallets of soft citrus was exported from this region. Farmers from the Sundays River Valley (near the Eastern Cape towns of Addo and Kirkwood) exported the second largest volume of 6 765 pallets, followed closely by the region surrounding Patensie, which exported 6 677 pallets of soft citrus in 2003. Marble Hall farmers exported 3 998 pallets of soft citrus in 2003, ranking them the third largest exporters. The fourth and fifth largest exporting regions are the Eastern Cape Midlands and the Levubu region, exporting 1 870 and 1 462 pallets of soft citrus, respectively in 2003. The entire KwaZulu-Natal region (consisting of the Pongola, Nkwalini and KZN Midlands regions) exported 976 pallets of soft citrus fruit in 2003, enough to rank it the sixth largest exporter of soft citrus in South Africa. The seventh largest exporter was Swaziland, which exported 654 pallets in 2003, followed by the Tzaneen, Hoedspruit and Orange River regions with export volumes of 382, 359 and 356

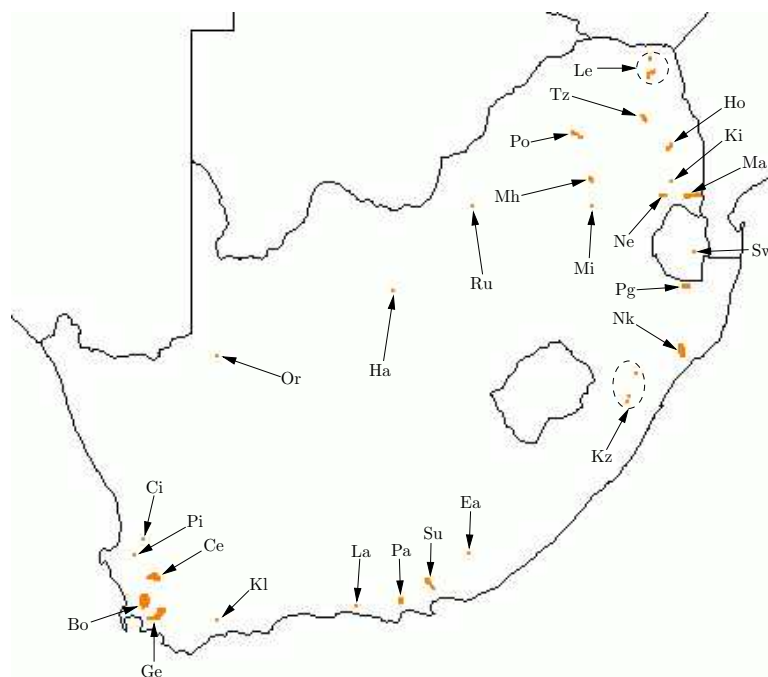


Figure 2.6: A schematic representation of the regions in which hard citrus fruit is produced and from where it is exported. Bo \equiv Boland, Ce \equiv Ceres, Ci \equiv Citrusdal, Ea \equiv Eastern Cape Midlands, Ge \equiv GEVV, Ha \equiv Hartswater, Ho \equiv Hoedspruit, Ki \equiv Kiepersol, Kl \equiv Klein Karoo, Kz \equiv KwaZulu-Natal, La \equiv Langkloof, Le \equiv Levubu, Ma \equiv Malelane, Mh \equiv Marble Hall, Mi \equiv Middelburg, Ne \equiv Nelspruit, Nk \equiv Nkwalini, Or \equiv Orange River, Pa \equiv Patensie, Pg \equiv Pongola, Pi \equiv Piketberg, Po \equiv Potgietersrus, Ru \equiv Rustenburg, Su \equiv Sundays River Valley, Sw \equiv Swaziland, Tz \equiv Tzaneen.

pallets respectively in 2003. Farmers from the Malelane region exported 109 pallets of soft citrus, making them the smallest exporter in 2003. Although Hartswater is shown in Figure 2.7, the region did not export any soft citrus in 2003. The total export volume of soft citrus in 2003, comprising the export from all the regions, together with 1876 pallets for which the origin is not known, was 59 070 pallets.

Avocados are grown only in the north-east and east of South Africa (as shown in Figure 2.8). The largest volume of avocados are exported from the Tzaneen region. In 2003, the farmers from that region exported 15 496 pallets of avocados. The area around Kiepersol produced enough avocados, for it to be ranked the second largest avocado exporting region in South Africa; exporting 9 553 pallets in 2003. The third largest region was near Nelspruit, from where 6 212 pallets were exported in 2003. The Levubu and KwaZulu-Natal (Nkwalini and KZN Midlands combined) regions were the fourth and fifth largest exporting regions, exporting 3 313 and 2 408 pallets in 2003 respectively. The PPECB additionally examined 1 605 pallets of avocados destined for export by air freight, bringing the grand total of avocados exported to 38 587 pallets in 2003.

Mangoes are grown only in north-eastern South Africa, in the Mpumalanga and Limpopo provinces (indicated in Figure 2.9). The largest volume of mangoes exported from South Africa originated from farms in the Hoedspruit region, from where an estimated 4 245 pallets were exported during the 2002/2003 season. The Tzaneen region exported the second largest volume of

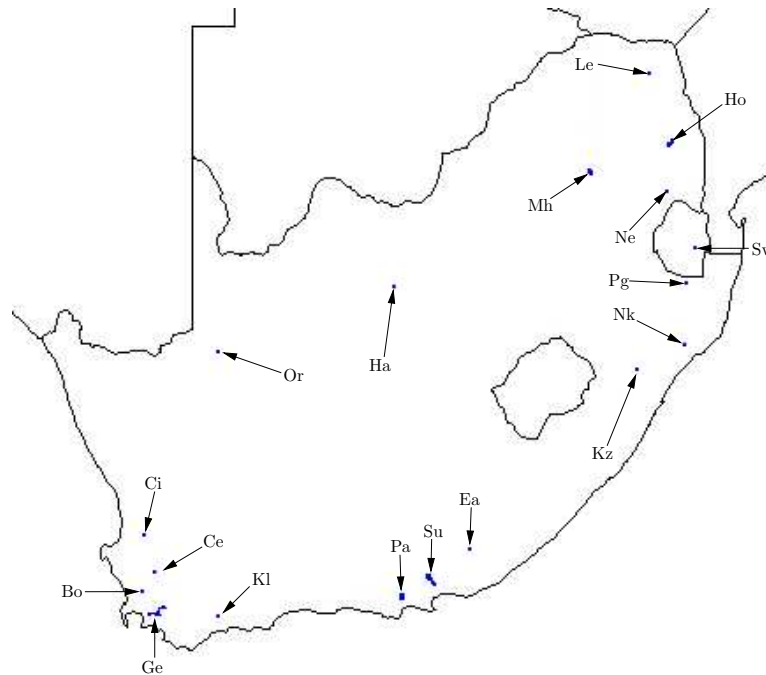


Figure 2.7: A schematic representation of the regions in which soft citrus is produced and from where they are exported. Bo \equiv Boland, Ce \equiv Ceres, Ci \equiv Citrusdal, Ea \equiv Eastern Cape Midlands, Ge \equiv GEVV, Ha \equiv Hartswater, Ho \equiv Hoedspruit, Kl \equiv Klein Karoo, Kz \equiv KwaZulu-Natal, Le \equiv Levubu, Mh \equiv Marble Hall, Ne \equiv Nelspruit, Nk \equiv Nkwalini, Or \equiv Orange River, Pa \equiv Patensie, Pg \equiv Pongola, Su \equiv Sundays River Valley, Sw \equiv Swaziland.

mangoes during the 2002/2003 season, namely 3 070 pallets. The third and fourth largest mango exporting regions were Malelane and Constantia, having exported approximately 2 057 and 878 pallets respectively during the 2002/2003 season. There is no record of mango packhouses in the list of packhouses supplied by the PPECB in the region. However, there is a large packhouse in the Tzaneen region called Constantia (which exported 682 pallets in the 2002/2003 season according to the 2002/2003 statements of fact published by the PPECB), thus it is uncertain whether the area called “Constantia” is the region near Potgietersrus, or a specific cold store. The farmers in the Levubu region exported enough mangoes to be ranked the sixth largest exporters (92 pallets), after the farmers near Nelspruit, who exported 675 pallets during the 2002/2003 season. The smallest mango exporting region is Gauteng (again not shown on the map, because there are not any packhouses in that region according to the PPECB packhouse list), which only exported an estimated 68 pallets of fruit. A possible explanation for this phenomenon may be that the PPECB inspected the mango pallets further down the supply chain and registered the origin as the area in which they were inspected. All together, the mango export volume from South Africa during the 2002/2003 season was therefore approximately 11 085 pallets.

Litchis are, like mangoes, grown in the north-eastern part of South Africa (shown in Figure 2.10). The largest exported volume originates from the Tzaneen region, from where an estimated 3 215 pallets of litchis were exported during the 2002/2003 season. Farmers in the Malelane region managed to export the second largest volume of litchis, exporting 2 784 pallets. Levubu is the third largest region, having exported an estimated 1 167 pallets during the 2002/2003

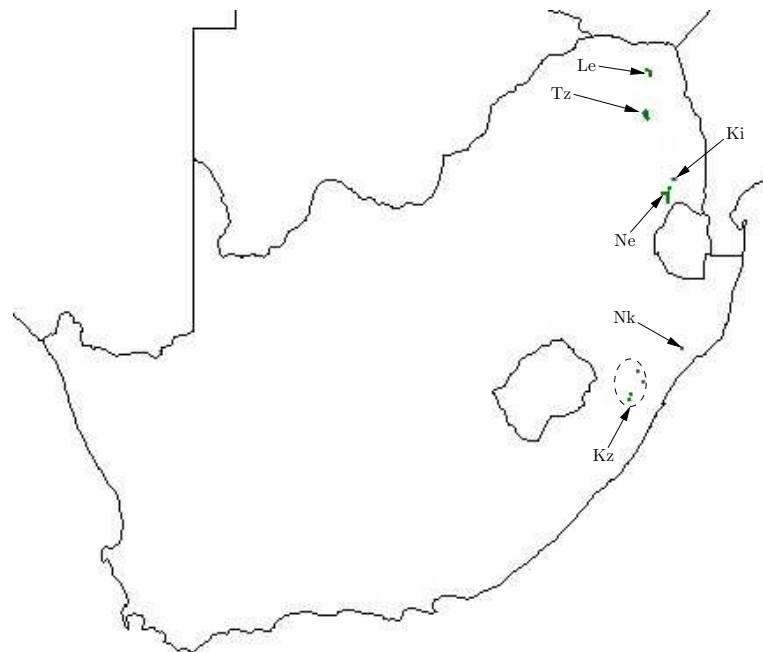


Figure 2.8: A schematic representation of the regions in which avocados are produced and from where they are exported. *Ki* \equiv Kiepersol, *Kz* \equiv KwaZulu-Natal, *Le* \equiv Levubu, *Ne* \equiv Nelspruit, *Nk* \equiv Nkwalini, *Tz* \equiv Tzaneen.

season. The region incorporating the Nelspruit and Kiepersol regions in Figure 2.10 achieved an estimated export volume of 915 pallets during the 2002/2003 season, making it the fourth largest litchi exporting region. The two smallest regions are Gauteng and Durban, exporting about 760 and 19 pallets respectively during the 2002/2003 season. No record of packhouses packing litchis in these two regions are listed in the packhouse list supplied by the PPECB. Perhaps the pallets were inspected further along the cold chain and registered as originating from the regions in which they were inspected.

2.2 Export Infrastructure

In this section, the possible modes of fruit transport from farms to foreign markets are discussed. The typical fruit supply chain is shown in Figure 2.11, in a somewhat simplified format. To take advantage of the higher prices that farmers are offered for their fruit in foreign markets than the prices offered in the local market, they attempt to supply the demand in a manner that minimises cost. The first fruit-specific infrastructure that the farmer is dependent on, is the packhouse. For grape farmers these are always located on their own farms — however, this is the exception, as farmers growing other fruit types usually transport their fruit to commercial or co-operative packhouses for packing. Here the fruit is graded and packed according to its usage: some fruit are packed for export to a foreign market, some fruit are packed for the local market and the remaining fruit is sent to a factory where it is processed into various products such as fruit juice or oil. Fruit destined for the local market and processed fruit are ignored in this study, as fruit destined for export usually takes preference over fruit destined for the local market at most packhouses and cold stores [222].

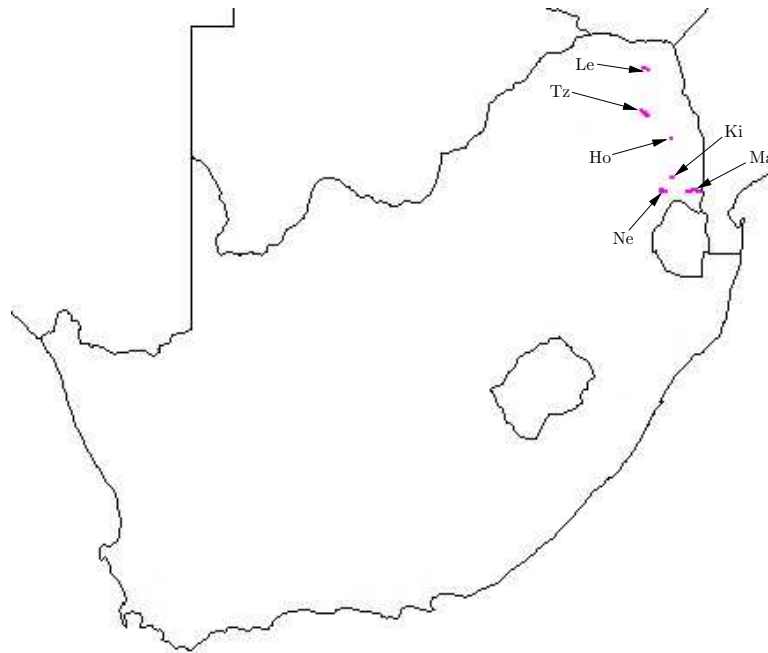


Figure 2.9: A schematic representation of the regions in which mangoes are produced and from where they are exported. *Ho* \equiv Hoedspruit, *Ki* \equiv Kiepersol, *Le* \equiv Levubu, *Ma* \equiv Malelane, *Ne* \equiv Nelspruit, *Tz* \equiv Tzaneen.

Road transport is the preferred mode used to transport fruit from cold stores to ports or airports. Transportation via the national rail network is an option that is not widely used, as the correct rolling stock are not always available for the transport of containers and because it is slower than transportation by road. Using a single train replaces 32 vehicles that would each have to transport a container of fruit, thus rail transport is more efficient than road transport if large volumes of fruit must be transported. The types of (sea) vessels that are used by exporters depend on their preferences: some exporters prefer to use refrigerated containers, as it reduces the handling of fruit — however, it is the opinion of other role-players in the industry that the circulation of cold air through an integral container is not as good as the circulation through the holds of conventional vessel [212]. Container shipping companies, makers of integral containers and producers of reefer vehicles/trailers maintain that the correct loading of reefer containers (porthole and integral), trailers and vehicles prevents inadequate airflow. If smaller volumes are to be exported, exporters tend to use containers, while for larger volumes, exporters generally prefer break-bulk shipping. Export by air is rarely used as the final supply chain link to the foreign destination, since it is much more expensive per pallet than transportation by sea.

2.2.1 Road Transport

Farmers transport their fruit to packing facilities on small vehicles or trailers pulled by tractors. From there fruit is loaded onto various types of vehicles, depending on the distance that the fruit is to be transported. Small vehicles are generally used to transport fruit from packing facilities to cold stores if the distance is short, while large vehicles are used if it is a long distance to the cold store (such as the cold stores near a port). Large vehicles are used to transport fruit

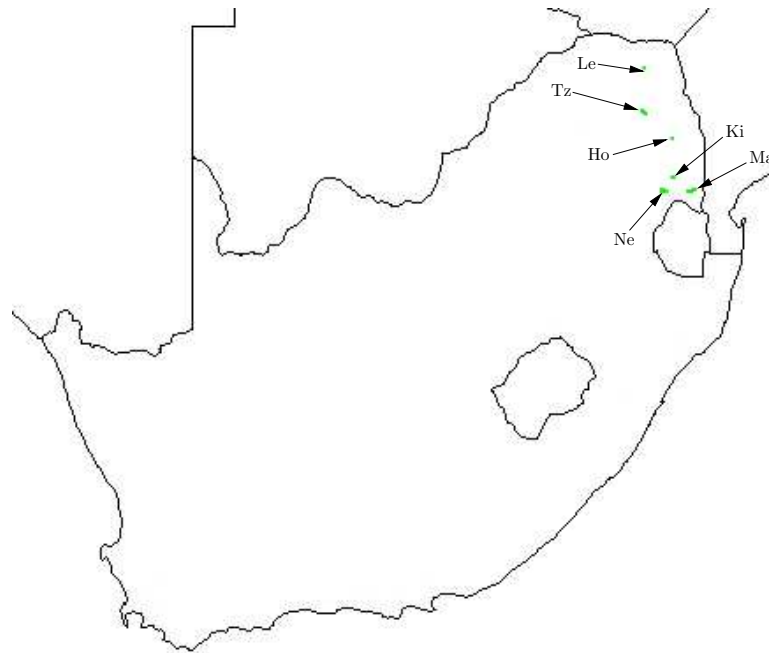


Figure 2.10: A schematic representation of the regions in which litchis are produced and from where they are exported. Ho \equiv Hoedspruit, Ki \equiv Kiepersol, Le \equiv Levubu, Ma \equiv Malelane, Ne \equiv Nelspruit, Tz \equiv Tzaneen.

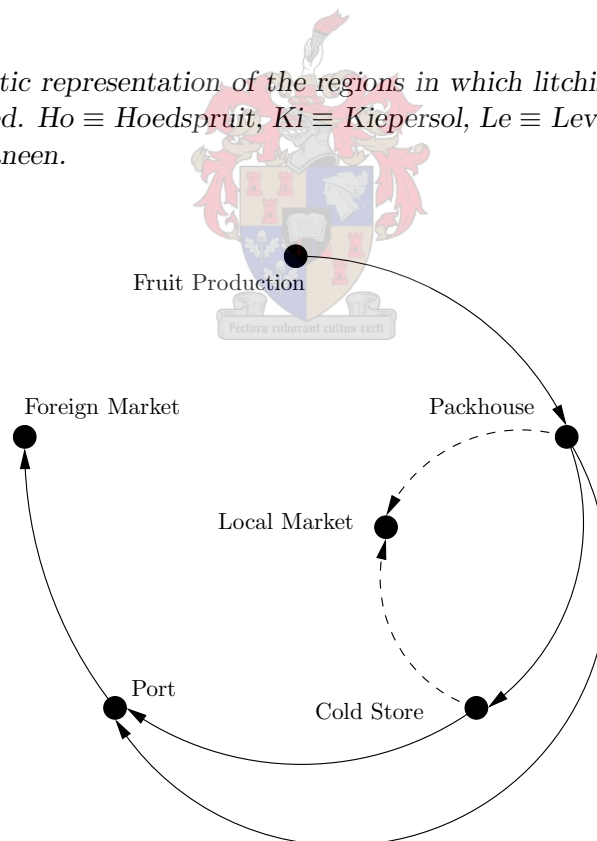


Figure 2.11: Schematic of the typical fruit supply chain.

from cold stores to ports, and usually have one of three types of trailers: either trailers suited for loading containers, trailers with insulated roofs and plastic curtains on the sides (called tautliners) or a trailer that is able to refrigerate fruit during transit (called a reefer trailer).

Fruit may also be transported from cold stores near a port (for break-bulk shipping) using flatbed trailers, as this practice results in shorter unloading and loading times at the port (cold store). Fruit can usually maintain its temperature for such short unrefrigerated trips.

Grapes from the Hex River Valley are usually sent to Cape Town harbour via tautliners (see Figure 2.12(a)) once they have been cooled to -0.5°C . A tautliner has a lightly insulated roof over the flatbed, onto which the pallets of fruit are packed, fitted with plastic curtains covering the sides [93]. For areas further away than two hours from another cold store, vehicles with refrigerator units are used. These vehicles are also called ‘reefers’ and usually hold approximately 28 pallets of fruit, but some vehicles are able to carry up to 34 pallets. Containers that have integrated cooling systems attached to them (thus called integral containers) constitute another method of transportation. This mode of transportation is called ‘genset’ (see Figure 2.12(b)) transport and refers to the scenario when a generator is attached to the integral container (so as to provide it with electricity for the cooling process, while in transit) [226]. If integral containers are used, fruit arrives at the final destination having been handled less frequently and the loading or unloading of vessels, trains and vehicles proceed much faster. The temperature within these containers is also easily kept constant. However, vehicles can only transport one 12 metre or one to three 6 metre containers per trip. Containers are only able to accommodate 20 pallets of fruit and it costs an additional amount to rent the generator in order to cool the container — thus reefer vehicles are more efficient (in terms of pallets per trip) [40] as they are able to transport 28 (or more) pallets of fruit per trip.



(a) A tautliner at FPT in Cape Town [151]. This vehicle is not refrigerated and is used only for short trips.



(b) Genset transport [251]. The generator provides electricity to the integral container to which it is attached.

Figure 2.12: Two modes of road transport used by fruit exporters.

In the remainder of this section the volumes of fruit transported along known routes are presented. These volumes are in tons¹, not pallets, as this was the unit used by Dr FE van Dyk [66], the researcher of the routes and their volumes shown in Figures 2.13 to 2.16 and Figure 2.18.

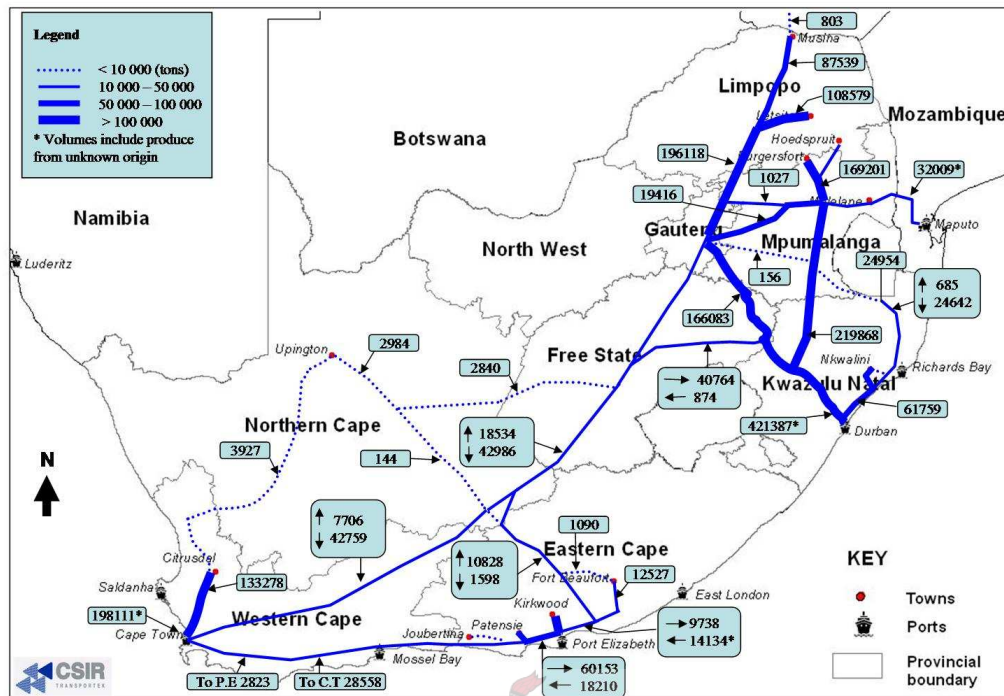


Figure 2.13: Routes and estimated volumes of citrus transported by road to the ports in 2003 [66].

Citrus Fruit Road Volumes and Routes

The routes taken by vehicles transporting citrus, and the 2003 volumes on these routes, are shown in Figure 2.13. Of the estimated 803 tons of citrus fruit exported from Zimbabwe, approximately 568 were exported via Durban harbour (distance = 1 098km from Beitbridge², toll cost = R828.50³), and the remaining 235 tons via Maputo harbour (distance = 894km from Beitbridge, toll cost = R755.00). All transportation was by road and the routes followed by the vehicles would most likely be on the N1 to Johannesburg, from where the fruit would be transported to Durban on the N3 or to Maputo on the N4. The 133 278 tons of citrus exported from Citrusdal were all transported by road, mainly to Cape Town (distance = 156km, no toll cost), with smaller volumes being diverted to Port Elizabeth (distance = 884km, toll cost = R82.00) and to Durban (distance = 1 745km, toll cost = R335.00). The fruit would have been transported on the N7 to Cape Town, and from there on the N2 to Port Elizabeth or on the N1 to Winburg, followed by the N5 to Harrismith and finally the N3 to Durban. There is a shorter route from Citrusdal to the N1 than via Cape Town: on this route vehicles would have to travel through Porterville and Ceres on the R44 and R46 (distance = 1 661km, toll cost = R335.00).

All 13 617 tons of citrus exported from the Eastern Cape Midlands region (near Fort Beaufort) were transported by road. Approximately 5 048 and 7 479 tons were transported to Cape Town (distance = 947km, toll cost = R82.00) and to Port Elizabeth (distance = 219km, no toll cost)

¹One can convert tons to pallets by dividing the mass of the exported fruit volume by the weight per pallet. However, this will lead to a small margin of error, as the pallet weights of various fruit types within a fruit group may differ, as may the pallet weights of the same fruit type destined for different markets.

²All distances were calculated from maps supplied by the Automobile Association of South Africa (AA) [22].

³All toll costs are for five or more-axle heavy vehicles and sourced from [159] (updated 1 March 2004).

respectively, on the R67 to Grahamstown, and then on the N2 to the ports. The 1 090 tons sent to Durban harbour would most likely have been transported on the R63 to the N10, which meets the N9 near Middelburg, and then joins the N1 to Winburg (distance = 1 209km, toll cost = R335.00). However, some citrus from this region may also have travelled on the R63 to King William's Town, and from there on the N2 to Durban (distance = 808km, toll cost = R50.00). Of the 43 668 tons of citrus exported from Hoedspruit, 6 602 and 1 663 tons were transported to Durban and Maputo by rail⁴ respectively (see Figure 2.18). Of the remaining volume, 4 009 tons were transported to Cape Town along the R527 and R36 to Ermelo, via Machadodorp and Lydenburg. Once at Ermelo, the N17 would have been followed until it reaches Johannesburg. From there, either the N1 would have been followed to Cape Town (distance = 2 181km, toll cost = R612.00), or to Colesburg, from where the N9 would have been followed to Middelburg (Eastern Cape), whence the N10 and eventually the N2 would have been followed to Port Elizabeth (distance = 1 513, toll cost = R577.00), which received 126 tons of citrus from Hoedspruit. However, vehicles could also take the N4 from Machadodorp, followed by the N12, N3 and finally the N1 to Cape Town (distance = 2 107km, toll cost = R689.00), shortening the trip by 74km. From Ermelo, the N11 may have been followed until it reaches the N3 near Ladysmith, which leads directly to Durban (distance = 779km, toll cost = R352.00), through which 32 347 tons of citrus were exported in 2003.

The 129 tons of citrus exported from the Langkloof region were sent along the R62 to the N2, which then leads to Port Elizabeth (distance = 185km, no toll cost). The fruit from the Sundays River Valley region comes from regions near two towns. If the fruit originated near Kirkwood, the fruit would either have been transported directly into Port Elizabeth on the R75 (distance = 96km, toll cost = R82.00), via the R75 to the N9, then along the N1, N5 and finally the N3 to Durban (distance = 1 248km, toll cost = R335.00) or along the R335, R334 and N2 to Durban (distance = 1 000km, toll cost = R50.00). If the fruit originated from Addo (all toll costs would be the same as for the routes from Kirkwood), the R355 would have been followed to the N2 (to Port Elizabeth 72km away or Cape Town 800km away), or the R336 may have been followed to Kirkwood or else the R335, R334 and N2 if the fruit was destined for Durban (distances = 1 279km, 969km respectively). Approximately 126 183 tons of citrus originated from this region in 2003; of it 13 162 tons were sent to Cape Town by road, 53 013 tons to Port Elizabeth by road, 51 972 tons to Port Elizabeth by rail (see Figure 2.18) and 8 036 tons to Durban by road.

As mentioned in §2.1.2, the Tzaneen region was the largest exporter of citrus fruit in 2003. Of the 175 510 tons of citrus exported from the region, 13 066 tons were transported to Cape Town by road (distance = 1 790km, toll cost = R598.50), 1 310 tons to Cape Town by rail, 51 tons to Port Elizabeth by road (distance = 1 496km, toll cost = R486.50), 88 994 tons to Durban by road (distance = 966km, toll cost = R656.50), 65 621 tons to Durban by rail, 6 468 tons by road to Maputo (distance = 516km, toll cost = R328.00) and 16 228 tons to Maputo by rail. The fruit from this region would have been transported on the R528 and R71 to Pietersburg and then along the N1, from where fruit could either be taken directly to Cape Town, via the N3 to Durban, or via the N9 and N10 to Port Elizabeth. To transport fruit to Maputo, vehicles could have used the R36 to Lydenburg, from where the R37 leads to Nelspruit located alongside the N4, in turn leading to Maputo. As the Malelane region lies alongside the N4, all 111 933 tons of exported hard citrus would have been transported on it. Cape Town received 5 765 tons, which would have been transported along the N4 to the N12, which leads directly to Johannesburg, from where the N1 would have been used (distance = 2 121km, toll cost = R819.00). The 22 tons destined for Port Elizabeth would have taken the same route, except that the N9 turn-off near Colesburg would have been taken to the N10 which leads into Port Elizabeth (distance =

⁴Costs of transporting fruit by rail are shown in Table F.4.

1 527km, toll cost = R707.00). However, most of the fruit was exported via Durban (distance = 793km, toll cost = R482.00), which received 68 939 tons by road and 12 046 tons by rail. Vehicles would typically have travelled on the N4 to Machadodorp, then on the R36 to Ermelo, followed by the N11 to the N3, which leads directly to Durban. Maputo received 25 161 tons of hard citrus by road (distance = 139km on the N4, toll cost = R198.00) and 13 835 tons by rail.

Farmers in the Levubu region exported a total of 94 573 tons of citrus in 2003. Of that, 8 393 tons were sent along the N1 to Cape Town (distance = 1 867km, toll cost = R680.50), while 1 030 tons went along the N1, then followed the N9, N10 and eventually the N2 to Port Elizabeth (distance = 1 573km, toll cost = R568.50). The largest volume from this region (76 521 tons) was sent to Durban (on the N1, then the N3 from Johannesburg; total distance = 1 043km, toll cost = 738.50) with a further 7 837 tons sent there by rail. Maputo received 792 tons of citrus, which would have been transported on the N1 to the turn-off onto the N11 near Potgietersrus, and then from the N11 onto the N4 near Middelburg (Mpumalanga) which leads to Maputo (distance = 839km, toll cost = R665.00). Another route that could have been taken from the Levubu region is the N1 until it intersects the R36, which leads to Lydenburg, from where the R37 could be followed to Nelspruit, which is situated on the N4, which leads to Maputo (distance = 690km, toll cost = R328.00). Of the 62 444 tons of citrus exported from the KwaZulu-Natal Province (mainly the Nkwalini region), 60 885 tons were exported from Durban. Farmers near Pongola would have sent their fruit on the N2 to Durban (distance = 380km, toll cost = R110.00), which is what the Nkwalini farmers do as well, as soon as the fruit reaches the N2 from the R66 (distance = 127km, toll cost = R74.00). The farmers from the KZN Midlands region would have transported their fruit along the N3 to Durban (distance = 87km, toll cost = R18.00). A relatively small volume of 698 tons would have been transported to Cape Town along the N3 to Harrismith, then on the N5 to Winburg, followed by the N1 to Cape Town (Pongola: distance = 1 971km, toll cost = R557.00; Nkwalini: distance = 1 760km, toll cost = R521.00; Richmond: distance = 1 089km, toll cost = R429.00; average: distance = 1 766km, toll cost = R502.00). The 176 tons exported via Port Elizabeth would either have been transported on the N2 (Pongola: distance = 1 341km, toll cost = R160.00; Nkwalini: distance = 1 130km, toll cost = R124.00; Richmond: distance = 860km, no toll cost; average: distance = 1 110km, toll cost = R95.00), or on the N3, N5, N9, N10 and finally N2 (Pongola: distance = 1 677km, toll cost = R445.00; Nkwalini: distance = 1 466km, toll cost = R409.00; Richmond: distance = 1 273km, toll cost = R317.00; average: distance = 1 472km, toll cost = R390.00). The 685 tons that were destined for Maputo would have been transported on the N2 to Ermelo, followed by the R36 to the N4 near Machado Toll Plaza and then on the N4 to Maputo (Pongola: distance = 640km, toll cost = R561.00; Nkwalini: distance = 946km, toll cost = R649.00; Richmond: distance = 1 089km, toll cost = R689.00; average: distance = 892km, toll cost = R633.00). Shorter routes from the three regions include utilising more rural (R) roads than national (N) roads. Once at Carolina, all routes follow the R38 to Kaapmuiden, then the N4 to Maputo (Pongola: distance = 591km, toll cost = R328.00; Nkwalini: distance = 732km, toll cost = R328.00; Richmond: distance = 841km, toll cost = R429.00; average: distance = 721km, toll cost = R362.00).

In 2003, the Patensie region exported 70 501 tons of citrus. The 10 348 tons that were exported through Cape Town harbour (distance = 693km, toll cost = R82.00) would have travelled along the R330 to Humansdorp, and from there on the N2. The 38 926 and 1 702 tons destined for Port Elizabeth (distance = 98km, no toll cost) and Durban respectively, would have been transported along the R331 to the N2 near Thornhill, from where the fruit exported via Durban would have been transported along the N10, N9, N1, N5 and finally along the N3 into Durban (distance = 1 369km, 1 012km if the fruit was transported along the R330 and N2 to Durban; toll cost = R335.00 and R50.00 respectively). A further 19 525 tons were transported to Port

Elizabeth by rail (see Figure 2.18). A total of 6 911 tons of citrus was exported from the Orange River region. The 3 927 tons exported via Cape Town harbour would have travelled on the R27 to Vanrhynsdorp, and on the N7 from there (distance = 812km, no toll cost). The 114 and 2 840 tons destined for Port Elizabeth and Durban respectively would have begun the journey on the N10. To reach Port Elizabeth, the N10 would have been followed until it reaches the N2 on the South African south coast (distance = 943km, no toll cost). Fruit transported to Durban would have travelled on the R64 from Groblershoop, via Kimberley to Bloemfontein, whereafter the N1, N5 and N3 would have been taken to Durban (distance = 1 202km, toll cost = R335.00).

Citrus fruit from the Senwes region comes from mainly three towns; namely Marble Hall, Groblersdal and Burgersfort. Marble Hall and Groblersdal lie on the N11 — hence fruit transported from there would have travelled on the N11 until it reaches the N4 near Middelburg (Mpumalanga). The 9 494 tons exported via Cape Town would have travelled on the N4, N12 and N1 from that point onwards (distance = 1 972km). The 4 059 tons sent to Maputo (distance = 518km) would have taken the N4 in the opposite direction to Mozambique, while 118 582 tons would have been transported on the N11 to the N3 near Ladysmith, from where it would have followed the N3 to Durban (distance = 720km). Citrus fruit from the Burgersfort region would have travelled on the R37 to Lydenburg and the R36 from Lydenburg to the N4 near Machadodorp. The total export from this region in 2003 was 132 135 tons.

Finally, the Swaziland region exported 24 954 tons of citrus in 2003. Only 127 tons were exported through Cape Town and 29 tons through Port Elizabeth. The remaining 24 798 tons were exported through Durban harbour. Since the main growing regions are near Phuzumoya, vehicles travelling to Cape Town would typically have taken the R8 to the R3 near Hlelehhele, which leads to the Swaziland/South African border, from where the N17 would have been taken to Johannesburg. From Johannesburg the N1 would have been taken either to Cape Town (distance = 2 145km, toll cost = R379.00) or Port Elizabeth (via the N9, N10 and N2; distance = 1 588km, toll cost = R267.00). Fruit destined for Durban, would have travelled along the R8 to the Swaziland–South African border, from where the N2 leads directly to Durban (distance = 464km, toll cost = R110.00) [291, 292]. If citrus fruit destined for Port Elizabeth were to travel the same route and continue along the N2, the transport distance would decrease by 163km to 1 425km for a decrease in toll fees of R107.00 to R160.00. Although no fruit was exported via Maputo harbour in 2003, there are two likely routes that vehicles could take to deliver fruit there. One possible route would be to take the R8, R16, R3 and R24 to the Mananga border post, on the R571 to Komatipoort and the N4 to Maputo (estimated distance = 396km, toll cost = R198.00). However, this route has the disadvantage of two border crossings. Another route would follow the R8, R16 and R3 to the Lomasha/Namaacha border crossing between Swaziland and Mozambique (estimated distance = 200km, no toll cost).

Deciduous Fruit Road Volumes and Routes

The routes taken by vehicles transporting deciduous fruit, and the 2003 volumes on these routes, are shown in Figure 2.14, while detailed deciduous fruit transportation volumes for the Western Cape are shown in 2.15. In 2003 an estimated total of 652 014 tons of deciduous fruit was exported from South Africa. All of the fruit was exported from the port in Cape Town, excluding 53 083 tons of apples and pears from the Langkloof region (around the Eastern Cape Town of Joubertina). This fruit would have travelled on the R62 and N2 to Port Elizabeth (distance = 183km, no toll cost). A further 2 886 tons were exported from the region via the Cape Town harbour. This fruit would have been transported on the R62 to Karedouw, then

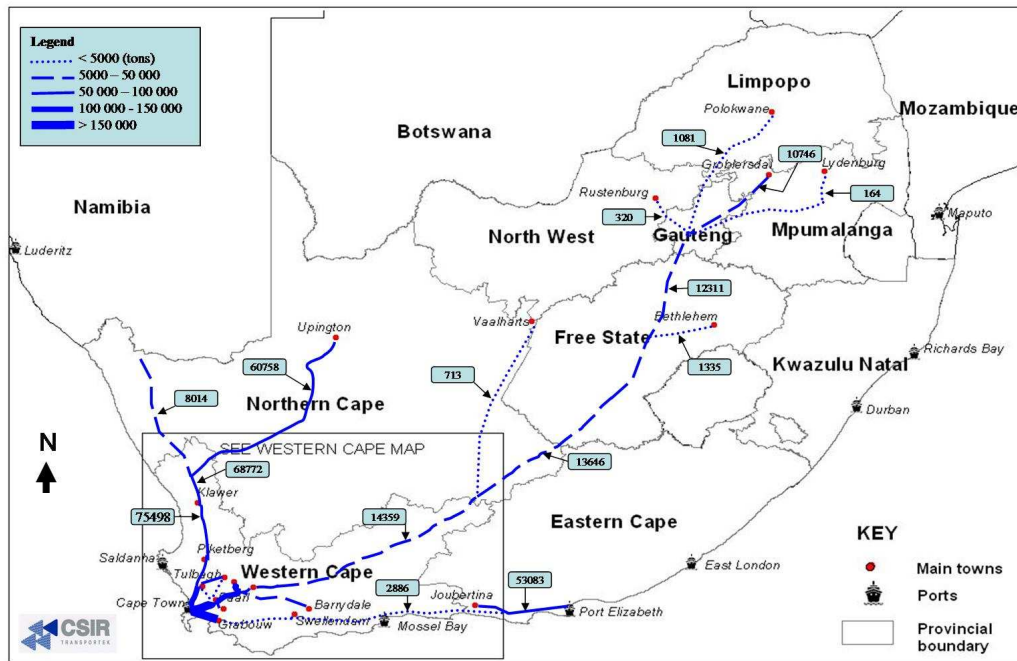


Figure 2.14: Routes and estimated volumes of deciduous fruit transported by road to the ports in 2003 [66]. See Figure 2.15 for an enlarged view of the region surrounded by the square in the map.

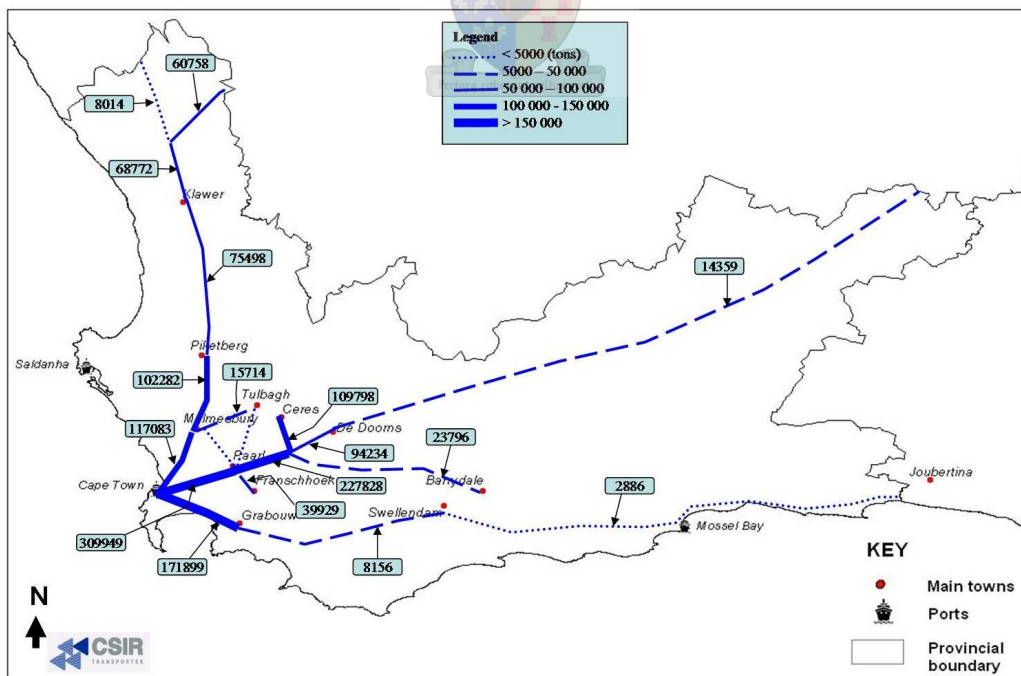


Figure 2.15: Routes and estimated volumes of deciduous fruit transported by road from farms in the Western Cape to Cape Town harbour in 2003 [66].

on the N2 to Cape Town (distance = 665km, toll cost = R82.00). There is a shorter, but not necessarily faster route along the R62 until it meets the N9, which leads to the N12 and eventually the N2 near George. If vehicles were to travel along this route, the distance would be 607km, *i.e.* 58km less than the other route and the toll at Tsitsikamma Toll Plaza between Nature's Valley and Stormsriver could be avoided (toll cost = R82.00).

The 23 796 tons of stone and pome fruit from the region around Barrydale would have been transported along the R62 to Montagu, from where the R60 would have been followed to Worcester and then the N1 to Cape Town (distance = 246km). Another route option would have been to take the R324 to the N2 near Buffeljagsrivier, which is 9km further, but avoiding the toll at the Huguenot Toll Plaza (cost = R112.00). The farmers from the Bethlehem region would have transported all their apples, nectarines, peaches and plums on the N5 to Winburg, followed by the N1 to Cape Town (distance = 1 221km, toll cost = R216.00). In 2003 they exported 1 335 tons of deciduous fruit. The farmers from the Ceres region would have transported their 109 798 tons of export deciduous fruit along the R43 to Worcester and then along the N1 to Cape Town (distance = 152km, toll cost = R112.00). As De Doorns lies adjacent to the N1, all the fruit is simply transported on it directly to Cape Town harbour (distance = 111km, toll cost = R112.00). In 2003, an estimated 79 875 tons of deciduous fruit were exported from this region.

The deciduous fruit farmers from Franschhoek would have transported their 39 929 tons of export fruit along the R43 to Paarl and the N1, which leads to Cape Town (distance = 78km, no toll cost). Farmers from Grabouw exported 163 743 tons of deciduous fruit via Cape Town harbour (distance = 72km, no toll cost). Grabouw is adjacent to the N2 — hence it is used to transport fruit to the port. The farmers from Groblersdal would have transported their 10 746 tons of deciduous fruit 1 888km along the R25 into Johannesburg, from where the N1 would have been followed to Cape Town (toll cost = R456.00). Klawer is situated next to the N7 — hence all 6 726 tons of fruit would have been transported on the N7 to Cape Town (distance = 267km, no toll cost). The 164 tons of deciduous fruit from Lydenburg would typically have been transported on the R36 to the Machado Toll Plaza, from where the N4, N12 and N1 would have been followed to Cape Town (distance = 1 989, toll cost = R585.00). If the R540 were to be used instead of the R36 to reach the N4, the Machado Toll Plaza (cost = R233.00) could have been avoided for a distance penalty of 6km.

To cross the South African border, the grapes from Namibia would have been transported through two border posts: Onseepkans near Pofadder and Vioolsdrif north of Springbok. The distance from Onseepkans to Cape Town is 759km along the N14 and N7, while the distance from Vioolsdrif to Cape Town is 660km along the N7. Neither route has any toll plazas, thus no toll cost would have been incurred. As Paarl is situated next to the N1, the 37 509 tons of deciduous fruit exported from the region would have been transported along the 53km to Cape Town on the N1 (no toll plazas on the route). Pietersburg is also situated next to the N1 — hence all of the 1 081 tons of deciduous fruit exported via Cape Town would have been transported the 1 701km to Cape Town on the N1 (toll cost = R598.50).

There are two routes along which to transport fruit from the Piketberg region. Most (25 871 tons) of the fruit would have been transported on the N7 to Cape Town (distance = 124km, no toll cost), while 913 tons would have been transported on the N7 to Malmesbury, followed by the R45 to Paarl, where the fruit would have been cooled at cold stores. Once the fruit was ready for loading onto the vessels in Cape Town harbour, it would have been transported on the N1 to the harbour (total distance = 53km, no toll cost). The farmers from the Tulbagh region did the same; transporting 15 714 tons of deciduous fruit directly to Cape Town harbour for export, while 3 770 tons were transported to Cape Town via cold stores in Paarl. The fruit

transported directly to the harbour would have been transported on the R46 to Malmesbury and then on the N7 from there to Cape Town (distance = 111km, no toll cost). The fruit sent to Paarl would have travelled along the R44 to Paarl, followed by the N1 to Cape Town (total distance = 103km, no toll cost).

The 320 tons of nectarines, peaches and plums from Rustenburg would have been transported along the N4 until it reaches the R511 near the Pelindaba Toll Plaza (cost = R8.50), which would have been avoided. The R511 leads to the N1, which then leads directly to Cape Town (distance = 1 816km, toll cost = R339.00). Swellendam is located next to the N2 — hence the 5 270 tons of deciduous fruit exported from the region would have been transported on the N2 to Cape Town (distance = 233km, no toll cost). Finally, the Vaalharts region near Hartswater in the Northern Cape exported 713 tons of deciduous fruit. This fruit would have been transported on the N18 to Warrenton, from where the N12 would have been followed until it reaches the N1 near Three Sisters, which leads directly to Cape Town (distance = 1 044km, toll cost = R112.00).

Subtropical Fruit Road Volumes and Routes

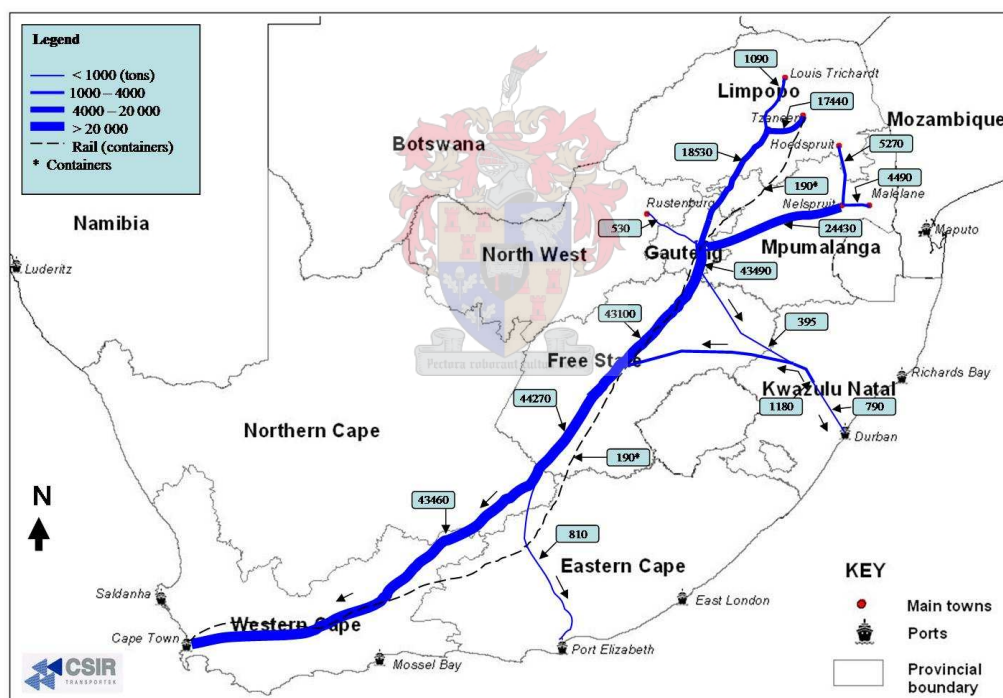


Figure 2.16: Routes and estimated volumes of subtropical fruit transported by road to the ports in 2003 [66].

The estimated subtropical fruit flow volumes of 2003 are summarised in Figure 2.16. An estimated total of 45 067 tons of subtropical fruit was exported during 2003. Farmers from the Hoedspruit region exported a total of 5 267 tons of subtropical fruit, 4 899 tons via Cape Town (distance = 2 107km, toll cost = R689.00), 263 tons via Port Elizabeth (distance = 1 513km, toll cost = R577.00) and 105 tons were exported via the Durban port (distance = 779km, toll cost = R352.00). The route followed by the vehicles would typically have been along the R36 to Machado Toll Plaza, from where the N4, N12 and N1 would be taken to Cape Town and

Port Elizabeth, or the vehicles would have continued on the R36 to Ermelo, from where the N11 and N3 would have been used to transport fruit to Durban. The same applies here as to deciduous fruit from Lydenburg; if the R540 would have been used from Lydenburg instead of the R36 for the fruit transported to Cape Town and Port Elizabeth, the toll at Machado Toll Plaza (cost = R233.00) would have been avoided for a distance penalty of 6km. The subtropical fruit from the Malelane and Nelspruit regions would have been transported on the same routes as both towns are located next to the N4. Fruit destined for Johannesburg, Cape Town and Port Elizabeth would have been transported on the N4 and N12 to Johannesburg, while fruit destined for Cape Town and Port Elizabeth would then have been transported on the N1 to Colesburg, the N9 to Middelburg, the N10 to Ncanara followed by the N2 to Port Elizabeth if it is exported via Port Elizabeth. Fruit exported via Durban harbour would have travelled on the N4 until it reaches the R36, which leads to Ermelo. From there, the N11 and N3 would have been used to transport fruit to Durban. Farmers from the Malelane region exported a total of 4 485 tons of subtropical fruit in 2003, 4 300 of which would have been exported via Cape Town along the N4 and N1 (distance = 2 121km, toll cost = R819.00), 132 tons via Port Elizabeth along the N4, N1 to Colesburg, N9 to Middelburg, N10 to the N10/N2 intersection at Ncanara (distance = 1 527km, toll cost = R707.00) and 53 tons via Durban harbour (distance = 793km, toll cost = R482.00) along the N4, N11 and N3. The Nelspruit region (which would transport export fruit along the same routes as the Malelane region) exported a total of 14 679 tons of subtropical fruit; 13 959 tons would have been sent to Cape Town (distance = 2 032km, toll cost = R689.00), 140 tons to Port Elizabeth (distance = 1 438km, toll cost = R577.00), 56 tons to Durban (distance = 704km, toll cost = R352.00) and 524 tons to Johannesburg (distance = 368km, toll cost = R350.00) for air freight to the foreign markets.

The fruit from Tzaneen would have travelled on the R528 and R71 to Pietersburg, from where it, like the fruit from Louis Trichardt, would have travelled on the N1 to Johannesburg and Cape Town, on the N1, N9, N10 and N2 to Port Elizabeth and on the N1 and N3 to Durban. The farmers in the Louis Trichardt region exported 1 093 tons of subtropical fruit, all of which was exported via Cape Town harbour (distance = 1 812km, toll cost = R680.50). Of the 17 442 tons of subtropical fruit exported from the Tzaneen region, 16 465 tons were exported via Cape Town harbour (distance = 1 790km, toll cost = R598.50), 272 tons via Port Elizabeth harbour (distance = 1 496km, toll cost = R486.50) and 181 tons via Durban harbour (distance = 966km, toll cost = R656.50). A further 190 containers of mainly avocados were transported to Cape Town by rail. The 529 tons of subtropical fruit from the Rustenburg region were all sent to Cape Town (distance = 1 816km, toll cost = R339.00). The same route would have been used as described for deciduous fruit export from Rustenburg. In 2003, KwaZulu-Natal exported 1 572 tons of avocados. There are many smaller regions in KwaZulu-Natal in which avocados are harvested for export — hence it is difficult to determine distances to the ports. An average of the distances from Richmond and Nkwalini to the ports will be used as the distance from KwaZulu-Natal to the ports. A relatively small volume (393 tons) would have been exported via Durban harbour (distance = 128km, toll cost = R83.00), while the largest export (1 179 tons) would have occurred via Cape Town harbour (distance = 1 664km, toll cost = R475.00).

2.2.2 Rail Transport

Trains used for the transport of containerised fruit typically consist of 32 wagons. This means that 32 twelve metre integral containers may be transported simultaneously. Since the 12 metre integral containers are each able to accommodate 20 pallets of fruit, one train is able to transport 640 pallets of fruit [226]. This is equivalent to 20 vehicles each carrying 32 pallets or 23 vehicles

each carrying 28 pallets. Hard citrus may be transported at ambient temperatures — hence closed wagons with slats on the sides may be used in this case.

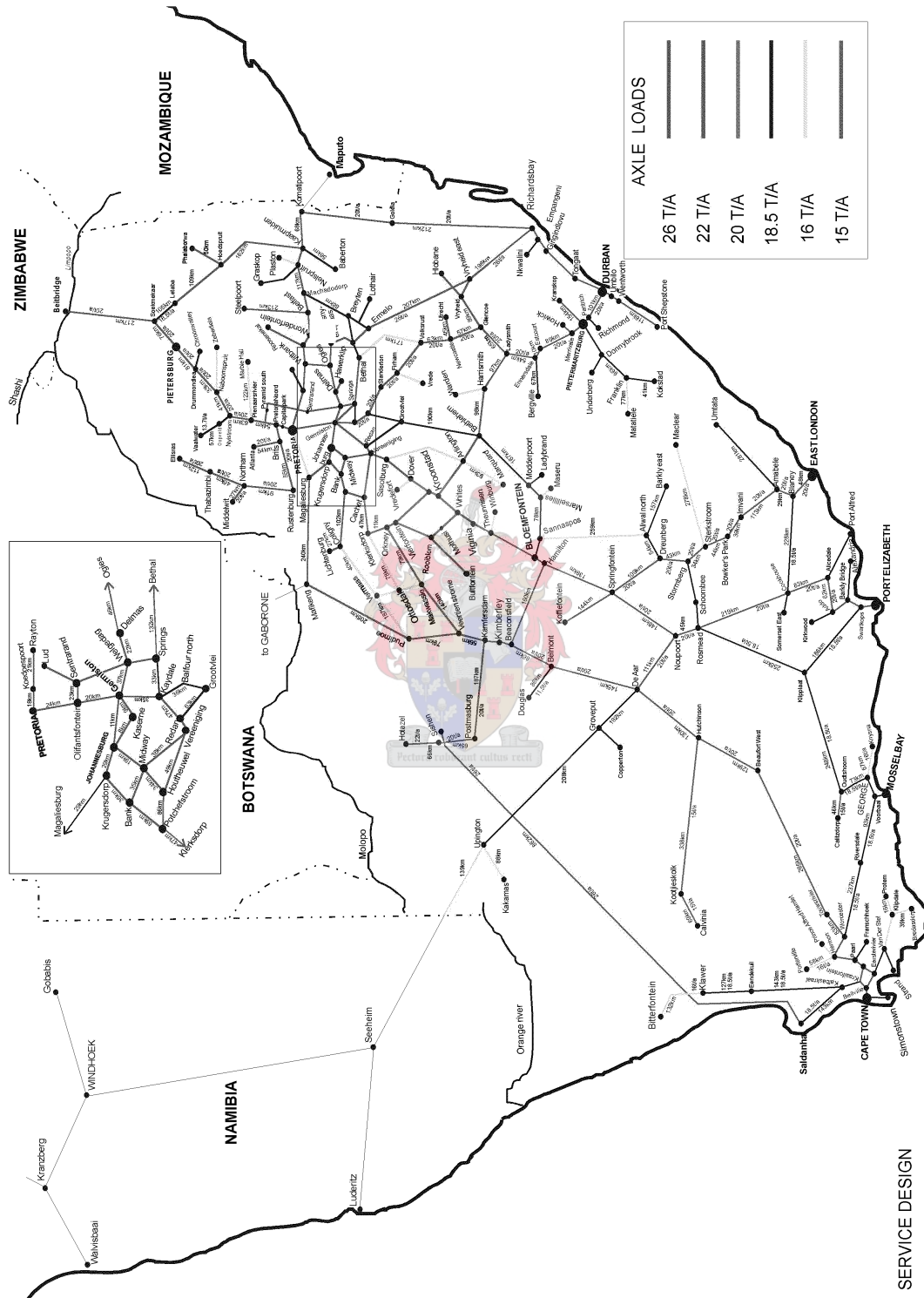


Figure 2.17: The South African national rail network. Note that the straight lines do not represent the actual routes travelled by trains.

Two areas of the country use rail transport regularly. The area around the Eastern Cape town of Patensie sends much of their hard citrus fruit to the harbour at Port Elizabeth via a dedicated railway. The rail tracks are of a smaller gauge than the typical tracks used by Spoornet (South Africa's semi-privatised mainline railway operator); thus the tracks are only used by citrus growers. However, some packing facilities in the Patensie Region send fruit to either Durban or Cape Town harbours by road. An example of this is when Port Elizabeth harbour is operating at maximum capacity during the peak in the citrus season; fruit growers then elect to use road transport in that event. The other major rail operation for hard citrus fruit originates in the Limpopo Province, then travels through Mpumalanga Province where the railway splits to Maputo in Mozambique and to Durban further south. This gives exporters the option of using the Maputo Fresh Produce Terminal (FPT) operation or the Durban terminal [36]. Reports (from a survey of citrus packing facilities conducted by Citrus South Africa) of problems with rats and dirty rail carriages indicate a need for more stringent cleaning policies by Spoornet for rail carriages that transport edibles⁵.

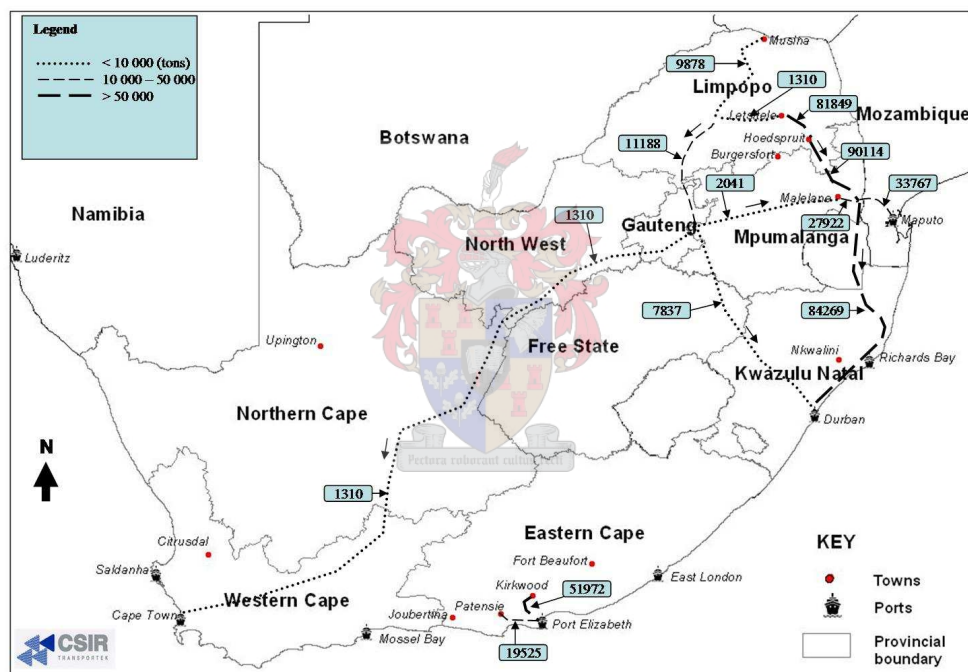


Figure 2.18: *The routes and estimated volumes of citrus transported by rail to all harbours in 2003 [66]. These volumes are discussed in the section on road transport of citrus fruit.*

There is also a train carrying containers of avocados from the Tzaneen region to Cape Town harbour. Viamax (a subsidiary of Transnet) managed a train route for mainly avocados in 2003 [100, 207]. They held contracts with exporters and Spoornet to ensure that the train would reach the ports in time for shipping [100]. This train has made it more viable to load containers inland (an advantage as the cold chain is less likely to be broken), as the common payment method for vehicle transport is to pay per ton of merchandise. Containers have a mass of approximately four tons each, which means that approximately an extra four pallets of fruit may be transported by refrigerated vehicle for an equivalent price of transporting a full

⁵This was learnt during a visit to Citrus South Africa (CSA). Citrus packhouse managers filled in questionnaires and the information regarding their packhouses, cold stores and the mode of transportation used to transport the fruit to the ports.

container. However, the cost of transport by train is per container and not by weight. The schedule is such that the train arrives in Cape Town harbour in time for weekend sailing, which is ideal for fruit sent to Rotterdam harbour and ports in France [100]. If there are not enough containers of avocados on the train, the additional space (typically 10% to 20% of the maximum capacity [193]) may be taken by other fruits (such as citrus in porthole containers — the use of porthole containers have since been discontinued) [207].

2.2.3 Sea Freight



(a) Container vessel [258]



(b) Conventional vessel [197]

Figure 2.19: Two modes of shipping fruit: container vessels and conventional vessels.

Although export vessels generally have different carrying capacities, vessels transporting only fruit from South Africa (*i.e.* non-container or conventional vessels, see Figure 2.19(b)) are typically able to carry 3 500 to 5 600 pallets under deck [225]. The older vessels were able to carry an average of 12 refrigerated containers, stored on the hatch covers [225]. This capacity has been increased in newer vessels [291]. A typical container vessel (see Figure 2.19(a)) is able to carry approximately 560 six metre containers and 120 twelve metre containers in total [226]. However, not all of these container stack positions necessarily have power points for the integral containers. Some of the six metre containers typically have to be loaded into the porthole container structure to fill available space. The holes on the containers and the refrigeration unit may be closed, so that, if there are not enough porthole containers to take up all the available space, a number of small conventional containers may be loaded in their stead. The structure required for porthole containers is very space-consuming — hence newly designed vessels generally no longer accommodate porthole containers [63]. This means that the fruit industry will only be able to use 6m and 12m integral containers for the new vessels. Since the older vessels still have this structure, porthole containers are expected to make a slow exit from the export industry.

Vessels sailing from Cape Town generally take between 12 and 14 days to reach their destination port in Europe. Using the Port Elizabeth harbour instead results in a one day extension of the trip duration, while an extension of two days [83] results from using Durban harbour. If the harbour at Lüderitz were to be used, the sailing time would decrease by just under two days. However, since this harbour is shallower than the South African ports, only smaller vessels are able to dock at Lüderitz and these smaller vessels generally sail slower, so that the distance advantage that Lüderitz offers, is negated [226].

Although most fruit exported from South Africa is shipped to Europe, the Middle East and Far East are important markets to many exporters. Due to the position of Maputo and Durban on the east coast of South Africa, they are better suited to shipping fruit to these markets because of shorter shipping distances.

2.2.4 Air Freight

The option of airlifting fruit to its final destination is usually only taken before Christmas and then only for table grapes if there is not enough time to send fruit by ship. This is done as these are typically the first grapes to enter the international market (during the first three to four weeks of the season). Very high prices are thus paid for these grapes, making air freight a viable mode of transportation. These grapes are sourced mainly from the Orange River and Limpopo Province grape regions. To transport grapes by air an aircraft must be chartered, as there are no regular flights from Upington on which exporters may book space (as they do for vessels) [226]. However, there are cargo flights from Johannesburg and Cape Town International airports on which space may be booked. The closest airports to these regions are in Upington and Johannesburg for the Orange River region and Limpopo Province respectively.

Vehicles transporting fruit from regional cold stores unload their cargo into a cold store at the airport. Here the pallets may have to be repacked to fit the shape of the aeroplane's cargo hold. Aircraft are loaded at night in Upington, as they cannot take off during the day with a full load (even though Upington airport has the longest runway in the southern hemisphere, at 4900m [214]). Aircraft are also loaded as late as possible before take-off, as they generally do not have on-board cooling systems [226].

Air freight is a rare form of transport and is very expensive when compared to export by vessel. Air freight will thus not be considered further in this thesis, nor will the option of air freight be included in the models of chapters to follow. It would be useful to include such an option if one were to know the prices paid for fruit at the foreign market. However, as these prices fluctuate daily, the option of air freight would only be useful to include in models for specific producers or exporters for a certain volume of fruit when the prices are exactly known for the time period in which the fruit is to reach the foreign market.

2.3 The Export Process

In this section, all fruit export supply chain processes are discussed in some detail. This includes the growing of fruit, picking, packing, cooling, transportation to the port, handling at the port, and shipping.

At the start of a season, farmers send estimates of their production volumes to their local packing facility and/or exporter. The exporters attempt to match the estimates of demands (provided by importers) to the production. This information leads to a provisional export plan, which is sent to the shipping line to reserve the necessary space or containers on their vessels. During the season more accurate estimates are made by both producers and importers and final bookings for space are made with this updated information. This may be done one to four weeks before sailing, according to the guidelines specified by the shipping lines [237].

2.3.1 Fruit on the Farm

In this section, the farming aspect of the export supply chain is considered. This aspect of the supply chain is included to provide the reader with an understanding of the processes involved in transporting fruit from the farm to the regional cold store. Although there might be opportunities for optimisation or streamlining of processes at each farm, no such localised optimisation was envisaged as part of the scope of this thesis (or as part of Fruitlog as a whole). Most farmers know, from many years of experience, how and when to pick, pack (in the case of grape farmers) and transport their fruit to the cooling or packing facility efficiently.

Growing

Before the picking of fruit actually begins, there is much that the farmer must first do. Crops have to be sprayed with weed killers and fungicides throughout the various growing phases to keep weeds from ‘stealing’ vital nutrients and sunlight that crops would usually absorb and to prevent fungi from destroying the fruit. It is preferable not to use pesticides in some cases as there is a perception that they are harmful to humans [191]. Some farmers invent interesting ways to prevent insects from crawling onto the fruit. These innovative ideas include, for example, putting grease on the stems of grape vines to prevent insects from crawling up the stem of the vine to the fruit — thus avoiding the use of pesticides [191]. Airborne insects are more difficult to control. It is in the farmers’ best interests to keep insects away from their fruit, as fewer markings on the flesh of the fruit increases the likelihood that the fruit will adhere to the high export standards. One method of controlling fruit flies, for example, is to release many sterilized male fruit flies onto farms. These sterile males mate with the fertile females (which then do not mate again), reducing the number of fertilized females and eventually the number of fruit flies. This is very effective if all farmers in a region release sterile males. With no fruit flies in the orchards, the fruit does not suffer from sting damage by the females attempting to lay their eggs, thereby increasing the quality of the fruit [222].

It is important to ensure that there are only a certain number of bunches/fruit on each vine or tree. The roots and leaves of a plant are limited in terms of the amount of nutrients they are able to supply the fruit — hence the less fruit there is on a tree, the more nutrients may be supplied to each individual fruit or berry [191]. This means that the individual fruit or berries may be larger and sweeter, and thus more likely to be exported (as opposed to being sold on the less profitable local market). Therefore workers have to trim away any excess fruit. Grape farmers, for example, typically trim the bunches to render them a certain shape and cut off any excess bunches that do not compare favourably with the others in terms of quality.

Before pome or stone fruit trees are planted in an orchard, large wooden poles are erected in straight lines. Strands of wire are suspended between the poles in a row. Once the saplings have been planted, their branches are attached to the wires, so that they are horizontal (see Figure 2.20(a)). This is done to slow down the growth in size of the tree, to promote fruit growth, and to give any fruit growing on the branches room to sway in windy conditions, so as to prevent any damage from bumping or rubbing against other branches [222]. Any damage to the fruit skin means the quality of the fruit is lower and the producer thus receives less money when the fruit is sold. Trees are also pruned annually to prevent damage to fruit, to remove any non-fruit bearing wood, to restrain the growth in size of the tree and to allow more light into the tree [222]. The size of the tree is kept to a minimum to allow for more trees to be planted in an orchard, to reduce the shadow cast by one tree on another and to render the harvesting process easier, as no large ladders are required — perhaps only smaller ladders to help with

the harvesting of the old trees. Trees or large shrubs are commonly grown on the borders of orchards to act as wind breaks (see Figure 2.20(b)). By slowing down the speed of the wind through the orchards, the fruit trees are less stressed, resulting in a potentially larger yield of fruit. The practice of mixing fertilizer for the orchards with the irrigation water before it is used (called fertigation), allows for the fertilizer to be spread more quickly and evenly to the trees, thus improving the required nutrient level in the ground. This is another method of promoting the growth of larger and better quality fruit⁶.



(a) The branches of young trees (plum trees in this picture) are fastened to horizontal wires so as to promote the growth of fruit and prevent damage to fruit that may grow on the branches.



(b) Other trees, such as pine trees, are grown next to orchards (apple orchards in this picture) to act as wind breaks so as to reduce the speed of wind through orchards, thus potentially improving the quality of the fruit and the size of the harvest.

Figure 2.20: Utilisation of horizontal wires and wind breaks are two methods of improving the quality of stone and pome fruit and the harvest yield of the orchards.

The weather plays a significant role in the farming of fruit. South Africa has many different weather patterns — hence a large variety of fruit types may be grown. Grapes require winter rain and no rain in summer, when the berries are on the vines. Other fruit types grow better when it rains in summer and the winters are dry. Cold temperatures are good for the colouration of some apple and pear cultivars — the better the colour of the fruit, the more likely it will be adequate for export [222]. But for all fruit types hail, snow and frost are disastrous — if any of these occur, the harvest may be lost. Fruit farms are thus not found in areas that are very cold at certain times of the year. Hail may even occur in warm weather, and hence it always poses a problem, but this does not occur often. Grapes are very sensitive to weather changes. If it should rain while the grapes are on the vines, the water that collects on the berries may cause the berries to burst their skins due to osmotic pressure. This is disastrous, as the grapes cannot be sold if they exhibit split skins. If very windy conditions prevail, dust and the leaves of the vines may also damage grapes. Any skin damage prevents grapes from being sold on the international market.

⁶The information regarding the pome and stone fruit in this section was learnt by the author from a visit to farms belonging to the Monteith Trust [222]. The farming practices of other farmers may differ.

Harvesting

If the fruit survives the onslaught of weeds, insects, fungi, animals and weather, the farmer must decide when to start the picking process. The farmer knows via visual inspections and experience during which week of the season fruit is ripe for picking. Grape farmers additionally use refractometers to ensure that their grapes contain the correct concentration of sugars. The Perishable Products Exporter's Control Board (PPECB) also uses refractometers to ensure that grapes to be exported are of sufficient quality. Once ripeness has been determined, selective picking begins. Farm workers on vineyards usually first consider berries for the correct colouring; if they are still not sure of the ripeness, they may taste a berry from the bunch and judge the ripeness according to taste. If they determine that the fruit is ripe, they will pick it and place it in a plastic crate. When the crate has one layer of fruit in it, it is loaded onto a trailer, pulled behind a small tractor that fits between the rows of vines. Grapes are picked before 9am in the Orange River area, preventing unnecessary heating of the berries, whilst picking may continue throughout the day in other areas (unless the air temperature is very hot or if the grape variety is seedless, as they are often more sensitive than seeded cultivars) [191].

Mangoes are picked using a different technique. Once the fruit is mature, but still hard (to withstand handling), a long pole with shears at the end is used to cut the mature mango from the tree. The mango then falls into a picking bag attached to the pole. Not all mangoes ripen at the same time, so picking is done every day, while there is fruit on the tree. As soon as possible after picking, the mangoes are washed in a soapy solution and then in a fungicide to prevent post-harvest decay. Mangoes are then washed again and polished before being packed [267].

Citrus fruit is picked by hand (using a citrus clipper) by a large number of workers that move through the orchards in groups. The fruit is placed into bags, which are later carefully emptied into bins or partitioned trailers. The trailers are then taken to the packing facility by a tractor, where the fruit is tipped into a chlorinated bath to remove all fungal spores or bacteria. The fruit is finally carried into the packing facility by a machine, where high pressure chlorinated water jets rinse the fruit before it enters the packing area [196].

To harvest pome fruit, workers pick the ripe fruit from the trees. For old trees they may need to use ladders to reach the higher branches, otherwise workers simply stand on the ground. The fruit is placed into bins (as carefully as possible, to prevent bruising) where sorters separate the class 3 fruit (for juicing) from the class 1 and class 2 fruit (for the international and local markets respectively). The bins are then transported to a loading area by tractor, and loaded onto vehicles for transport — either to the juicing factory or to a cold store or packhouse. Stone fruit must be placed in cold storage within eight hours of being packed, while apples and pears are allowed to remain at ambient temperatures for 24 hours before they must be in cold storage [222].

Packing

The main difference between the packing of grapes and other fruit, is the location of the packing facilities. Grape farmers have their own packing facilities on their farms, while other fruit farmers usually send their fruit to communal packing facilities, which are centrally located and serve a number of farms. At a grape farmer's packing facility, fruit crates are often loaded from the trailer onto pallets, to render the handling easier. The number of crates per pallet varies from farmer to farmer. These pallets are then put into a cold room for pre-cooling. The grapes are cooled to about 18°C as they deteriorate quickly when too warm. Furthermore, if grapes

are warm, they respire more, causing condensation in the bag (into which they are later packed) which may result in sulphur dioxide burn (if it is used as a preservative). The crates are then sent from the cold room to the packing area. Some farmers additionally prefer to cool their packing facilities (for the grapes and for the benefit of the workers) by using evaporator coolers or ceramic base paints, while others install sprinkler systems on the roof of the packing facility [191].

When the grapes enter the packing area, each bunch of grapes is tested for quality and size. Each bunch is trimmed to the desired shape and size. Any unripe, wilted or damaged berries are cut from the bunch, leaving only the best berries on the bunch. There are typically small groups of workers at packing facilities who perform this task, and each group has a supervisor that does further trimming and then sorts the grapes into grade 1 (export quality) or grade 2 (local quality). The supervisors weigh the grapes and sort them so that packets weighing as close as possible to the specified weight of, for example, 500g (called punnets) are made. The supervisors usually err on the high side of the weight specifications. The weight varies according to the packaging type. Once this is done, the punnets are packed into cartons (see Figure 2.21(a)). The sizes of cartons also vary, but they are usually 4.5kg or 9kg cartons. Depending on the packages, packers may have to add extra materials, such as sheets of paper coated with sulphur dioxide, to act as preservatives.



(a) Workers packing cartons with grapes.

(b) A worker building a pallet of grapes.

Figure 2.21: Two operations occurring in a grape packing facility [212].

Once the cartons are filled with fruit, they are packed onto pallets (see Figure 2.21(b)). Various stickers are placed on these pallets to aid the export process. One of these stickers is a bar code, which is a number that is a unique to the pallet and is used at many points throughout the export supply chain to ensure traceability and trackability. These pallets usually carry a hundred and sixty 4.5kg cartons, or eighty 9kg cartons. If the pallets are to be packed into high containers, the number of 4.5kg cartons increases to 180. This is beneficial to the farmer who only pays for transportation and terminal handling on a per pallet basis: if more cartons are able to fit onto a pallet, a slightly higher return results. Farmers then send their grapes to regional cold stores that force cool the grapes to -0.5°C . This is the temperature at which the grapes should be kept at all times during the subsequent export process. Grapes do not freeze at this temperature, due to the dissolved sugars that lower the freezing point of water. If the grapes were to freeze, they could not be sold. For this reason packers insert a thermocouple into

a berry in the centre of each pallet. The thermocouple wires are then connected to a machine that tracks the temperature throughout the export process. If it is found that the grapes were not at -0.5°C during the entire trip, the PPECB (in South Africa) or importers (in the foreign ports) may decide to turn down part of the shipment. This is not desirable for the farmer or shipping company, as they would have paid the necessary costs up to that point in the export process: if pallets are turned down, they are liable and incur a loss on those pallets [83].

2.3.2 Fruit at Communal Packing Facilities

Packing facilities that pack various types of fruit also occur. Some do this as the seasons differ between fruit types — thus when citrus is in season, citrus will be packed, while stone fruit will be packed when it is, in turn, in season. In this manner a packing facility can operate throughout an entire year without forced closure [192], while pack houses packing only one type of fruit have to close during the weeks that the fruit is not picked⁷.

At a typical communal pack house, the fruit arrives at the loading zone, where the bins of fruit are placed into rooms that perform pre-cooling. Here the fruit is cooled to the correct temperature. In the case of citrus and apples, if the fruit is to be packed (this may not be immediate — some apple varieties may be stored up to 12 months in controlled atmosphere (CA) storage [192]), the bins are placed on a machine that first “showers” the fruit, then drags the bins through a chlorinated solution. The citrus and apples float on the chlorinated water which washes the fruit and kills most bacteria. Subtropical and stone fruit, along with pears, have to be handled by hand as they either do not float on water or may be damaged by bumping into one another in the water. The fruit then goes onto a conveyor belt to an area where some class 3 fruit (used for juice) may be picked out from amongst the other (class 1 and 2) fruit. However, farmers are encouraged to remove most class 3 fruit at the farm to save costs, as they are charged per bin that is packed [222]. The more cartons of class 1 or 2 fruit that are packed from a bin, the less the cost of packing and pre-cooling per carton. Samples of the fruit are taken and either analysed for quality control, or for monitoring over a period of five weeks. This is done so that any post-harvest problems with a batch of fruit can be identified, should they occur [196]. The citrus first goes through a waxing process at this point. After the waxing, the fruit is transported to tables where workers sort the fruit further into class 1 and class 2 fruit. The class 2 fruit is then transported to tables where it is packed for local sales. Class 1 fruit is placed on a conveyor belt of cups that sorts the fruit according to size (by measuring the mass of the individual fruit in each cup) and colour (with a camera linked to a computer that analyses the image from the camera). If the fruit reaches its destination table, the cup drops and the fruit rolls to workers that pack the fruit into cartons or packets. Workers may still reject some remaining class 3 fruit at this point, which is sent for juicing. Once cartons are packed, they are stacked onto pallets, thermocouple wires are inserted and the fruit undergoes a mandatory inspection by the PPECB. Once it is checked, the fruit is either placed back into a cold room or CA storage where it is later collected for transport to a harbour (class 1), local supermarkets or fruit sellers (some class 1 and class 2) or juicing factories (class 3). No chemicals (other than chlorinated water and wax) are used throughout the entire post-harvest period due to the restrictions imposed by “Chemical-free” foreign markets, such as Japan [196].

⁷This information was learnt by the author during a visit to the new Novo pack house in Paarl, Western Cape [192]; some packhouses may differ in operation.

2.3.3 Fruit at Regional Cold Stores

There are two types of regional cold stores for fruit (in the production regions); one type stores fruit before it is packed and the other stores fruit once it is packed. The former applies mainly to pome fruits which can be stored for long periods of time, and the latter refers to all fruits, although some citrus fruit may bypass cold storage altogether and be cooled for the first time either at the port or on the vessel.

The producers would usually transport their fruit in wooden, or more recently plastic, bins filled with fruit from the orchards to the cold stores. Each bin should contain only a single variety of a fruit, as varieties are packed in groups. Forklifts remove the bins from the vehicles or trailers and place them on top of one another, usually in groups sorted by farm of origin, length of cooling period required and fruit varieties to ease the removal process, when it is time to pack the fruit. Each bin has a bar code, linked to a database, that is used throughout the export supply chain to discern which fruit variety it is and from which farm the fruit originates. This is important as the producer will later instruct the packhouse when to pack the fruit that is in storage, hence workers at the packhouses and cold stores must know where the fruit is stored. Once the storage time is over, forklifts remove the bins from the cold rooms and either load them onto a trailer or vehicle, or take them directly to the packhouse, depending on the distance between cold room and packhouse. Once there, some packhouses have holding cold rooms for bins that are due to be packed within a few days. Maintaining the cold chain is very important for the quality of fruit; it should not be broken [222].

Producers may choose between two types of cooling for their unpacked pome fruits; regular atmosphere (RA) cooling or controlled atmosphere (CA) cooling. RA cold rooms simply cool down fruit to the temperatures listed in Table 2.4 and keep it there until it can be packed. The fruit may be kept there for up to a few months, depending on the fruit type and variety. The mix of gasses (mainly nitrogen and carbon dioxide) in CA storage prevents the fruit from maturing, thus extending the time that it can be kept cool, allowing producers and exporters to choose a time of the year during which they wish to enter the market. The further out of the harvesting season the producer can sell the fruit, the higher the price that the producer can fetch from the markets, offsetting the extra storage costs for CA storage [222].

SmartFreshTM is a new gaseous product on the market which slows down the ripening of fruit even further, giving producers the opportunity to enter their fruit onto the market at an even later stage. It works on ethylene-sensitive crops, replacing the ethylene that ripens fruit with an inhibitor, reducing the ripening process dramatically and keeping the firmness and colour closer to those at the time of harvesting, resulting in better prices for the fruit [72]. Once fruit is removed from SmartFreshTM, it slowly begins to ripen again to be ready for sale [2]. SmartFreshTM can also prevent the formation of scald, a disease that occurs occasionally on fruit stored for long periods of time [2, 72]. Adding SmartFreshTM to the fruit during RA storage resulted in apples that were the same quality after 4 months as when they were stored in CA with SmartFreshTM, thus indicating that RA storage with SmartFreshTM may replace CA storage for medium term cooling [72].

Once the fruit has been packed into the appropriate cartons and is ready for transport, it is stored in RA rooms. Packhouses are not necessarily adjacent to the cold stores — hence pallets sometimes have to be loaded onto trailers pulled by tractors or vehicles to be transported to the cold stores, where forklifts would then remove the pallets and place them into their appropriate lanes in a cold room until they can be transported to a port. The lanes in the cold rooms allow groups of pallets earmarked for the same market to be placed together, thus making it easier for forklift operators to find and remove those pallets that are due to be sent to a port.

It is not desirable for the fruit to be kept in these holding cold stores for too long; it is more advantageous to send fruit to a port to be loaded onto a vessel as soon as possible [222].

Forced cooling is achieved in these lanes by blowing air through the pallets; the air temperature is typically -2°C if the fruit is to be cooled to -0.5°C (see Table 2.4 for a list of temperatures at which each fruit type should typically be kept). This cools the fruit down to its optimal transport temperature (see Table 2.4 for the transport and storing temperatures set by the PPECB) within a certain time period, depending on the temperature to which the fruit must be cooled, and the technology used to cool the fruit. Once the fruit is at the required temperature, it is immediately transported to the port, or to another nearby cold store where it remains until such time that it is ready to be shipped to a foreign port. Cold stores that perform forced cooling do not usually keep fruit once it has been cooled to the correct temperature as there are often other fruit shipments that await cooling: any cooled fruit stored too long would take up space that is required by the newly arrived fruit [93]. These cooling facilities are sometimes owned cooperatively by a number of farmers that make use of them [191].

Temperature ($^{\circ}\text{C}$)	Fruit
-0.5	Apples, apricots, cherries, grapes, kiwi fruit, nectarines, peaches, pears and plums
2	Litchis
3	Avocados
3.5	Oranges and soft citrus
7	Grapefruit, lemons, lime, other citrus
7.5	Guava
8	Mangoes and prickly pears
8.5	Passion fruit
10	Grapefruit and papaya
12	Bananas and pineapples

Table 2.4: *Optimal storage and transport temperatures for some types of fruit. These temperatures may be changed during the transport process to accelerate ripening. The PPECB have produced a list of all temperatures and temperature changes allowed for various fruit types [234]. Note that some fruit types listed here are not currently exported from South Africa in any significant volumes, if at all.*

Avocados require more stringent conditions to arrive at their final destination in a quality sufficient for the foreign markets. They must arrive at the market place within 12 days of being picked, if they are only cooled. This cannot be achieved by South African exporters under normal conditions, as it takes about two days to pick, pack and transport the fruit to a port. Cape Town harbour is used for avocado export to Europe and Durban is used for export to the Middle East, because they are the closest ports to their respective foreign ports. To overcome the problem of an effective total transport time of 16 days, avocados are shipped in controlled atmosphere (CA) containers. These containers are sometimes connected to a central unit on the conventional vessel (which differs between shipping companies, some CA containers are self contained) and are controlled by an operator, varying gas concentrations, humidity, *etc.* so as to slow down the ageing of the fruit and thus prolong the allowable transport time [198].

2.3.4 Fruit at the Port

This section is dedicated to a description of the transportation of fruit within ports from where it is exported to a foreign market. Most of the information in this section was learned by the author during visits to the Fresh Produce Terminals (FPT) [83, 225] and the container terminal of the South African Port Operations (SAPO) [63] in Cape Town and Durban harbours.

Fresh Produce Terminals

The process of shipping fruit to a foreign port begins with the owner of the vessel. The shipping company advises exporters when their vessel is expected in the port. Before the fruit is released from the forced cooling facility, the exporter books space on the vessel if there is sufficient fruit ready for export. Vehicles delivering fruit to Fresh Produce Terminals (FPT) in Cape Town drive directly to the terminal, while those travelling to Durban's FPT, first stop at a vehicle-stop belonging to FPT in order to avoid congestion in the terminal. Here drivers may report to the FPT terminal to schedule an unloading time, while taking a break [83]. This prevents bottlenecks at the terminal and driver frustration. When the vehicles arrive at the FPT, the drivers (or their assistants) hand their consignment notes to the security personnel at the gate to the terminal. If any item on the consignment notes does not agree with what is onboard the vehicle (e.g. an incorrect number of pallets or vehicle number), the vehicle is diverted and the exporter is contacted to correct the error. If the consignment notes are correct or the errors have been corrected, the vehicle is taken to the loading area at the scheduled time for unloading. At the loading area, the bar codes on the pallets are checked to ensure that the correct pallets are on the vehicle. If the bar codes contradict the information supplied by the exporter, the vehicle is off-loaded and the pallets are stored in a holding cold room, in which case the exporter is charged for the handling and storage costs, until all errors and discrepancies have been eliminated.

However, if all the information is correct, forklifts unload the pallets and place them in large cold rooms. The information linked to the bar codes on the pallets inform the FPT on which vessel the pallets are to be loaded, and the forklift operators unloading the vehicles place the pallets in a numbered row in a cold room. When the vessel is to be loaded, the forklift operator is informed which numbered rows of pallets to transport to the cranes that load the vessel. Only six pallets are allowed outside the cold room at a time, otherwise they may become too warm by the time they arrive on the vessel. Cold rooms are at -2.5°C and strategically-placed plastic sheets, fans and extractor fans keep the pallets cold to their centre. Thermocouples may be used to monitor the temperatures at all times, but the cold rooms also have sensors that may be used to verify the temperature in the pallets. Some exporters prefer to bypass the cold storage facilities so as to save on extra handling and storage costs, by sending instructions that their vehicles be unloaded directly at the cranes. They are then allowed an increase of 1°C in the temperature of the fruit before it is cooled again in the vessel. However, if the temperature rises above that single degree, the exporter forfeits the pallet and the fruit is discarded. Exporters therefore take on some risk by bypassing cold storage at the port. They still have to pay for the pallets that are not loaded onto the vessel in the case of a forfeit, as they had booked the space. The PPECB ensures that all temperature and export standards are adhered to at all times.

Forklift operators are instructed by their supervisors which rows of pallets to load into which cargo (deck) level of a vessel. The further the fruit is to travel along the voyage, the deeper into the vessel it goes, so that no unnecessary movement of pallets occurs at foreign ports. Once the previous batch of pallets has been loaded into the vessel, the forklift operator loads the cages

used by the crane to lift the pallets of fruit. These cages come in two sizes: one size handles four pallets of fruit and is used while the cargo level being packed is still fairly empty. The other cage type is only able to take two pallets and it is used when the hold is almost full and loading space is scarce. When the final few pallets are loaded, slings are wrapped around the pallets and the crane lifts the pallets one by one, by means of the slings and puts them into the hold. These slings are left on the pallets so that the vessel's cranes or the cranes at the destination ports may lift them out first. Once all the holds are closed, the final few pallets are packed into containers. These containers have plug-in points so that refrigeration occurs during the voyage. This may be done by the FPT or by the exporter. These containers are then emptied at the foreign port or by the buyer at their depot.

The FPT supplies the forklifts that are used to load fruit into vessels. The cranes lift them through the hatches into the vessel to arrange the pallets. They are then moved between levels by the cranes. The FPT may also provide cranes for the vessel if one of their cranes are broken or if the pallets are too heavy (for example, citrus pallets are heavier than pallets of deciduous fruit). While the pallets are being loaded, there is an officer who removes a copy of the bar codes on the pallets and sticks it onto a list. Once sixteen pallets have been loaded, the list is sent to a supervisor who double checks that the correct pallets are being loaded into the vessel. Incorrect pallets that have been loaded have to be removed. While the vessel is being loaded, none of the vessel's own crew may be in the hold: only FPT workers may be there. This is required so that liability may be established, should fruit be damaged. If fruit is damaged in port, the FPT is responsible for the loss. Once the vessel is loaded, the vessel's captain and the vessel's agent (e.g. Rennies, Marven Shipping, etc.) [225] sign for the consignment after an inspection. If the fruit arrives damaged at the destination port, the liner (vessel charterer, e.g. Laucool, Anlin Shipping, Star Reefers, Cape Reefers, etc.) is liable [225].

Container Terminals

Exporters have another option when they choose their mode of export. The South African Port Operations also export large quantities of fruit through their container terminals. Three types of containers are used for this purpose: a 6 metre porthole container (see Figure 2.22(a)), a six metre refrigerated container (also called a reefer, a reefer container or an integral container) and a 12 metre integral container (see Figure 2.22(b)). The six metre containers accommodate 10 pallets of fruit, while the 12 metre containers accommodate 20 pallets of fruit. Porthole containers have two holes on one side of the container, one above the other. These holes may be plugged into a refrigeration device that blows cold air into the top hole and extracts it from the bottom hole. Many vessels have the structures in their holds to accommodate these containers and Cape Town Container Terminal (CTCT) is the only port in the Southern Hemisphere that has the same structure on land and thus is the only place where these containers may be stored before they are loaded onto a vessel. The dilemma is that the refrigeration structure required to keep porthole containers cold, takes up a significant amount of space inside the vessel. They have thus become unfavoured by shipping companies and new vessels are being built without these structures.

This leaves only the integral containers for future use. They have a built-in refrigeration system which requires power. Thus the terminals have some of their container stacks dedicated exclusively to integral containers. There are power points that supply the integral containers with electricity so as to keep the fruit cold at all times and PPECB representatives ensure that cooling standards are adhered to. The temperatures of the containers can be monitored from a control room. The vessels also have the necessary plug-in points for these containers.



(a) A six metre porthole container.

(b) A 12 metre integral container.

Figure 2.22: Comparison of two types of reefer containers used to export fruit [157].

When the vehicles arrive at the container terminal with fruit loads, the driver or the driver's assistant takes the necessary documentation to the administration check (called the A-check). Here South African Port Operations (SAPO) verify that the correct vehicle, driver and container(s) have arrived. The driver hands over documentation from the exporter to a computer operator at the A-check. If the driver number, vehicle number or the container numbers do not correspond to what is expected by SAPO, the driver must obtain the correct documentation from the exporter. If all the documentation is correct, the vehicle is sent to the physical check (called the P-check). At the P-check workers have hand-held devices that are connected to the same system used by the A-check by means of a wireless connection. With these devices, they verify the vehicle registration number and container numbers. They also verify that the seals on the containers are intact. If there is a discrepancy, the vehicle is sent back to the exporter. If the information is correct, the driver is given documentation that informs him where his containers are expected. Straddle carriers (see Figure 2.24(b)) have a computer in the cabin that informs the operator where the containers are to be stacked so that it is easy to remove them and load them onto the vessel at a later stage. This is where a significant bottleneck is perceived to occur (amongst transporters), especially in the port of Cape Town, but to a lesser extent also in Durban.

However, the porthole containers are taken to the so-called blue store at Cape Town Harbour. Here the correct refrigeration infrastructure (see Figure 2.23) is available to keep the fruit in each container cold. Containers are kept here until the vessel that will be carrying the fruit is ready to be loaded. When the vessel is ready to be loaded, straddle carriers load the containers onto a bathtub trailer (see Figure 2.24(b)) — a specialized trailer with a slanted edge. Small vehicles are adequate as they do not need to travel large distances and it allows two to be able to pass one another easily in tight areas such as in the blue store. These vehicles transport containers to the large ship-to-shore container cranes (see Figure 2.24(a) for an example), which then load them onto the vessel. The company that owns the vessel calculates the placement of the containers on the vessel so as to maximise the speed of unloading and to distribute the weight of the containers evenly. This is possible, because exporters are required to book space on the vessel from before it arrives at the harbour until it is fully booked. In their booking they have to specify the destination of each container, as well as its contents and weight. Once the containers are loaded, they are secured to the vessel to prevent them from sliding or falling off in bad weather or rough seas. See Figure 2.19(a) for an example of a container vessel.



(a) Refrigeration structure for porthole containers.

(b) Porthole container plugged into refrigeration system.

Figure 2.23: Porthole refrigeration structure found in the blue store at Cape Town Harbour [212].

Once a vehicle is offloaded, it goes to the P-check (out) where SAPO once again ensures that all is in order. This is done to prevent theft of containers and to ensure that importers leave with the correct merchandise. Customs officials randomly check the contents of the containers leaving the terminal to ensure that all standards are met and that illegal substances are not imported.

2.4 Cost Data

Mathematicians often design models to describe the functioning of a real system and investigate the possible use of numerous algorithms to solve those models. However, if the data they use in the models are inaccurate, the solution is of little use and could even lead to incorrect conclusions. Much data were required to achieve the objectives of this thesis. Since the fruit industry is composed of many role-players (such as packhouse, cold store and terminal managers, transport operators, *etc.*), the gathering of data for these models has been a mammoth task, carried out by a team of people associated with Fruitlog. In certain instances simplifications had to be made, for example, to use industry average transport and handling rates. It is therefore important to perform a sensitivity analysis so as to determine the sensitivity of the results to variations in the data. In the remainder of this section, the data required to solve the models in chapters to come are presented and discussed.

The international commercial terms (also called incoterms) of the contracts of sale commonly used by South African farmers are the Cost, Insurance, Freight (CIF) terms and the Free-On Board (FOB) terms. The terms of these contracts state that the supplier is responsible for all costs and risk in harvesting, packing, cooling, transporting the fruit to the port and loading the fruit onto the vessels for the FOB terms, and shipping the fruit to a foreign port for the CIF terms. For FOB terms, the buyer takes responsibility of the fruit from when it is on the vessel in a port in South Africa [228], while if the fruit were to be sold on CIF terms, the cost of shipping is included. Once the fruit arrives at the destination port, the buyer assumes all risk and costs. This means that all transportation and handling from the foreign port to the final destination is the responsibility of the buyers in Europe, the USA, the Far East, *etc.* [94].



(a) A ship-to-shore container crane. There are various sizes of these cranes, the more popular ones being the Panamax, Post-Panamax and Super Post-Panamax sized cranes [182].



(b) A straddle carrier loading a container onto a bathtub trailer being pulled by a hauler [269].

Figure 2.24: Some container terminal equipment. The sizes of ship-to-shore container cranes are important, as Panamax cranes cannot load a vessel wider than 13 rows of containers, Post-Panamax cranes cannot load vessels wider than 18 rows of containers, and Super Post-Panamax cranes can load vessels up to 22-24 rows wide [183].

As it is very difficult to determine the cost of producing fruit, the cost of fruit production is not considered in this thesis. All fruit must be packed, and since this cost vary greatly between packhouses due to agreements farmers or exporters may have with producers, this cost will also be disregarded.

2.4.1 Cold Store Handling Cost

Typical costs of cooling fruit to the correct shipment temperature are shown in Table 2.5. These costs include the packing of pallets into containers or onto vehicles.

Fruit Group	Cost (R)
Grapes	196
Stone	200
Pome	120
Citrus	75

Table 2.5: Costs per pallet of cooling fruit to the correct temperature at a typical inland cold store [172]. Costs include loading of fruit into containers or onto vehicles.

Two managers of large cold stores handling all three types of subtropical fruits considered in this study were contacted for cooling tariffs for subtropical fruit. Mr A de Klerk [82], manager at Haffenden Groves Cold Store, stated that their subtropical fruit was cooled to a temperature adequate for road transport and that the cold store Container Fruit Terminals (CFT) cooled the fruit to export temperature. The manager of Tzaneen Fruit Packers, Mr A Leeb [200],

stated that cooling tariffs would be difficult to estimate, as farmers are charged a set rate for packing, cooling and wrapping combined. Thus, three cooling tariffs for subtropical fruit will be considered in the model. The three rates that will be considered are R120 (the inland cold store pome cooling rate), R200 (the inland cold store stone cooling rate) and R160 (an average of the two tariffs). Pome and stone fruit must be cooled to a lower temperature (-0.5°C compared to 2°C for Litchis, 3°C for Avocados and 8°C for Mangoes) thereby incurring a larger cooling cost. However, the annual export volumes of deciduous fruits are far larger than subtropical export volumes, allowing for a reduced cost per pallet per degree due to economies of scale. This may explain the discrepancy between pome and stone fruit cooling rates.

2.4.2 Road and Rail Transportation Cost

Only two types of terrestrial transport are used in South Africa. Transport by road is accomplished using vehicles (the horse and trailer variety) where the trailers are either a unrefrigerated tautliners (see Figure 2.12(a)) or trailers capable of carrying multiple six metre porthole or integral containers or single 12 metre integral containers. Transport by rail is accomplished in either railway carriages at ambient temperature or in integral containers. Most road transportation costs were obtained from A du Bois [94] at FPT in Cape Town, while some containerised transportation costs were obtained from Mr R Opperman [228] at *Kuehne + Nagel* (these costs are summarised in Appendix F). The costs in bold face in Table F.1, Table F.2 and Table F.3 were those supplied to the author; the others were calculated from the lines of best fit shown in Figure F.1 and Figure F.2. The line of best fit for refrigerated fruit is shown in Figure F.3. Due to the high margin of error for this best fit line, it was not used to estimate any further costs. Therefore, only those costs provided to the author were used in the models. Due to the high cost of transporting fruit by refrigerated vehicle, this is an unlikely choice for transport and only those routes shown in Table F.3 are occasionally used.

2.4.3 Port Handling Cost

The cost of handling fruit in break-bulk or containers in the container, FPT and SAFT terminals are discussed in this section. Although each terminal has its own complex pricing structure for all conceivable handling scenarios, only the most commonly used costs are considered here. It is assumed that fruit always arrives on time, documentation is always correct and no more than seven days of break-bulk storage or three days of container storage are required. Any further costs may be obtained from the terminal operators, or in the case of the container terminals, from their website [269]. To ship a 12m container via the container terminal costs R1058.06 and a surcharge of R427.03 is charged for 12m reefer containers (R74.25 per pallet).

FPT charges varying rates for the four terminals they operate. The costs included in the models are listed in Table 2.6. The direct reefer shipments apply to pallets that are transported directly to the wharf and loaded onto the vessel. The reefer shipments via the cold store apply to pallets that are first transported to the FPT cold store, stored in the cold store for up to seven days, and later loaded onto a vessel. Container stuffing applies to pallets that are transported to the cold store in break-bulk, stored up to seven days, and packed into a container. The container loading rate charged by FPT is the container terminal handling charge for the use of a crane to load a reefer container onto a vessel. The break-bulk stevedoring rate is the charge for the use of the workers to load the break-bulk fruit into the hatches of conventional vessels. The container stevedoring charge is for the use of the workers to load a container onto a conventional vessel. Exporters wishing to export containers on the decks of conventional vessels are charged

the Container Terminal Handling Charge, which includes all transport to and from the stack and three days of plug-in point hire at the stack. It is also assumed that a container packed at FPT for loading onto the deck of a conventional vessel does not require use of the stack, as they can pack a container so that it is ready for loading once the vessel is ready to receive it. Until then the fruit may remain in cold storage. FPT charges R40 per pallet to transport fruit to or from other terminals (in Cape Town) — therefore this is assumed to be the standard cost for all inter-terminal fruit transport or transport from cold stores near the ports to the terminals, when fruit is break-bulk or containerised. Also, it is assumed that containers packed at the FPT cold stores would not be transported to the SAFT terminal for shipping; they would only be allowed to be shipped from the FPT terminal itself.

Service	Cost (R/pallet)			
	Cape Town	Durban	Port Elizabeth	Maputo
Direct Reefer Shipments	145.00	145.00	134.00	138.00
Reefer Shipments via Cold Store	241.00	246.00	246.00	253.00
Container Stuffing via Cold Store	155.00	251.00	251.00	260.00
Break-Bulk Stevedoring	51.00	51.00	51.00	53.00
Container Stevedoring	7.30	7.30	7.30	7.52
Container Stack Lift-On/Lift-Off	12.50	12.50	n/a	12.50
Plug-in Point Hire (3 Days)	22.50	13.65	30.00	14.03
Container Terminal Handling Charge	69.50	49.50	49.50	50.50

Table 2.6: *The FPT fruit handling costs for 2005. Not all costs have been included, only those required for the models in this thesis have been included. VAT is excluded in all cases. All prices are listed per pallet. Container stuffing includes seven days of cold storage and the container terminal handling charge includes four days of plug-in point hire.*

Unfortunately only the 2003 season tariffs were available for SAFT handling tariffs, as SAFT did not have standard rates for 2005. Their tariff structure had changed to become client-specific. Thus, the tariffs were compared to the equivalent tariffs at FPT during 2003 and the price adjusted accordingly for 2005 by including a typical (over the various services) inflation of 9% per annum. This was later confirmed by Mr W Brintjies [41] as a good estimate for 2005 tariffs. These adjusted tariffs are listed in Table 2.7. It is assumed that no more than seven days of cold storage are required and 3 days of plug-in point rental tariff is included.

2.4.4 Sea and Air Freight Cost

As mentioned in previous sections, fruit may be transported to foreign markets in five modes. The most expensive mode is to fly fruit to its destination. However, this mode of transportation is not commonly used and is not considered further in this thesis. Fruit may also be shipped by integral container (see Figure 2.22(b)) on a container vessel (see Figure 2.19(a)) or conventional vessel (see Figure 2.19(b)), in a porthole container (see Figure 2.22(a)), on a container vessel or in break-bulk on a conventional reefer vessel.

The cost of shipping break-bulk fruit on conventional vessels or a container of fruit on container vessels is seasonal, as indicated by the shipping costs provided by Mr G van Heerden [294], managing director of *Cape Reefers*. The cost per pallet is shown in Table 2.8. The summer peak week in 2003 was week 7, which was the week of 10 – 16 February of that year. The winter

Service	Cost (R/pallet)	
	Cape Town	Durban
Direct Shipping	150.00	130.00
Cold Storage (break-bulk)	Sub-tropical	182.00
	Deciduous	164.00
	Soft citrus	125.00
	Hard citrus	116.00
Cold Storage (container)	Sub-tropical	195.00
	Deciduous	177.00
	Soft citrus	137.00
	Hard citrus	128.00
Bulk Stevedoring	57.00	57.00
Container Stevedoring	6.60	6.60
Transport from Rail	n/a	52.00
Plug-in Point Hire (3 Days)	n/a	26.00
Hire of Shore-cranes	37.00	37.00

Table 2.7: The SAFT fruit handling costs for 2005 estimated from 2003 data. Not all costs have been included; only those required for the models in this thesis have been included. VAT is excluded in all cases. All prices are listed per pallet. Cold store rates are assumed standard for all port cold stores. Port cold stores in Port Elizabeth are assumed to have the same cost structure as the cold stores affiliated to SAFT in Durban.

peak week of 2003 occurred during the period from 14 – 20 July (week 29); these values are shown in bold face in Table 2.8. These costs are for fruit exported to the United Kingdom, the most common destination for fruit exported from South Africa. The cost of shipping fruit in a container on the deck of a conventional vessel is approximately the same as shipping a container via the container terminal.

Month	Containerised (RA)	Containerised (CA)	Break-bulk
January	1 225	1 698	1 393
<i>February</i>	<i>1 278</i>	<i>1 750</i>	<i>1 428</i>
March	1 278	1 750	1 428
April	1 278	1 750	1 428
May	1 278	1 750	1 358
June	1 278	1 750	1 120
1 – 15 July	1 190	1 663	1 015
<i>16 July – December</i>	<i>945</i>	<i>1 418</i>	<i>1 015</i>

Table 2.8: Shipping costs for fruit exported from South Africa to the UK during 2005. All costs are listed in Rands per pallet and it was assumed that a US Dollar costs seven South African Rands, as costs were supplied in US Dollars [294].

2.4.5 Additional Costs

Further costs that are not immediately evident may be incurred by the farmer or exporter. For example, a considerable amount of documentation has to be completed for fruit to be exported and this adds additional expense to export. Many of these documents have to be completed for each shipment of fruit. In an attempt at estimating these costs per pallet of fruit, it is assumed that a typical shipment consists of 20 pallets, as suggested by Mr R Opperman [228]. The first form is the EUR 1/Form A/Certificate of Origin (depends on the destination), which

all cost R100 per document. The Phytosanitary Certificate (which costs R120 per shipment) is a document from the Department of Agriculture stating that the fruit is not infested by pests or diseases. A further cost of R6.65 per pallet is charged by the PPECB as a sea levy. The Bill of Lading Fee must also be paid for and costs R100. A further R150 is charged for a Voucher of Correction when the original documentation is incorrect. However, this voucher is not considered in this study; it is assumed that all original documentation is correct.

If the fruit is exported by container, further costs are involved. The Container Terminal Order costs R100 and the Container Release Fee is R300 for a 20m container. The Cargo Dues cost is R1 545 for a 12m container and must be paid for any exported container, regardless of the terminal from where it is exported. The container must be inspected before it is loaded onto a vessel to ensure that the seals have not broken, the temperature is set correctly, *etc.* This Reefer Inspection Fee costs a further R76 per container.

As most farmers use freight forwarders (such as *Kuehne + Nagel*) or an export company to aid them with the export process, they have to pay those companies an additional fee. This is typically R500 per shipment [228]. Thus, the total additional cost for a shipment of 20 break-bulk pallets of fruit amounts to R953, or R47.65 per pallet. For containerised fruit, the total cost is R2 974, or R148.70 per pallet.

2.5 Chapter Summary

In this chapter, the workings of and interactions between the various segments of the fruit export industry were discussed in an attempt to aid the reader in understanding, to some degree, the complex export supply chain. After a brief discussion putting the size of the South African fruit industry into context, the types and cultivars of fruits exported from South Africa were presented. The various fruit seasons were explained to vary between fruit types and cultivars, and, in some cases, the seasons vary due to the geographical locations of the regions. The various fruit-growing regions of South Africa and the estimated volumes exported by these regions were presented, followed by a presentation of the export infrastructure. This included detailed discussions on road transport (including volumes exported along certain routes, the distances from these regions to ports and the toll cost incurred), rail transport and transport by sea. Brief mention was also made of the possibility of export of fruit by air.

The transport of fruit from orchards to ports was discussed in some detail; including practices during the growing of fruit, the harvesting of fruit, the packing of fruit, the cooling, the transportation of fruit and loading at the terminals. A discussion of the procedures followed at container, FPT and SAFT terminals was followed by the cost associated with fruit export, thereby achieving Thesis Objective I, as stated in §1.2.

Chapter 3

Infrastructure Capacities

“It has been said that figures rule the world. Maybe. But I am sure that figures show us whether it is being ruled well or badly.”

– Johann Wolfgang von Goethe [307]

This chapter is dedicated to establishing the infrastructure capacities of the South African fruit export infrastructure. The data gathering methodology is described and the capacities of packhouses, cold stores and handling at terminals are summarised, as are the fruit production and export data. During the process of determining how efficiently the ports operated in the past or could be in future in relation to the flow of fruit through the infrastructure, it was decided to build a spreadsheet model of the ports that allows the user to change certain properties of the ports in order to test the effects of these changes with respect to efficiencies locally and globally within the export infrastructure. Dr FE van Dyk assembled a team consisting of Mr J Havenga and Ms I Hobbs (consultants from the Stellenbosch Business School), Dr DB Louw from the Deciduous Fruit Producers Trust (DFPT), Dr NH Louw from the Stellenbosch Department of Logistics and the author. The shell of the model was built over two brainstorming days with all present, and later refined by Dr FE van Dyk and the author. All data used for the model, and the conclusions reached are presented and discussed in this chapter.

3.1 Packhouse and Cold Store Capacities

To gather cold store and packhouse capacity data, the PPECB was approached for a list of all packhouses and cold stores. According to their records there are 850 packhouses and 315 cold stores throughout South Africa. All packhouses in which only grapes are packed (according to the PPECB list) were disregarded as each grape farmer has his own packhouse and it is assumed that he will ensure that there is sufficient capacity for the grape production on his farm — grapes were assumed to enter the infrastructure only once they are cooled (that is, at the cold store level). This significantly decreased the number of role-players that had to be contacted in order to gather capacity data.

All role-players on the lists supplied by the PPECB were sent short questionnaires by email or fax (see examples of these questionnaires in Figures 3.1 and 3.2). Reminders were sent out as appropriate and this was followed up by personally contacting the representatives of facilities

Your Name		
Position in firm		
Name of the packhouse		
Phone number of the packhouse		
Cell number of the contact person		
Fax number of the packhouse		
Email address of the packhouse		
Which fruit types do you pack (please mark with an X)?		
Grapes	Stone	Pome
Citrus	Soft citrus	Avos
Mangos	Litchis	Other (please specify)
What percentage of fruit packed at your packhouse is for export ?		
Are you located near any major road or railway , if so, which one?		
How many days a week does your facility pack fruit (e.g. $5\frac{1}{2}$, 7)?		
What is the maximum number of cartons or pallets that can be packed by your packhouse in a week (please specify cartons/pallets)?		
Is there a cold store at the packhouse?		
And, if so, how many pallets can be cooled to the required temperature in a one week cycle?		
Where is the fruit transported after it has been packed (e.g. inland cold store/ cold store at or near harbour/harbour — please specify harbour)?		
Is road or rail used to transport fruit to its next destination?		
How long does it take to reach the harbour from your facility (hours)?		
Which cold stores and/or terminals do you use at the harbours?		
Is there any other information you would like to bring to our attention?		

Figure 3.1: An example of a packhouse questionnaire.

Your Name		
Position in firm		
Name of the cold store		
Phone number of the cold store		
Cell number of the contact person		
Fax number of the cold store		
Email address of the cold store		
Which fruit types do you store (please mark with an X)?		
Grapes	Stone	Pome
Citrus	Soft citrus	Avos
Mangos	Litchis	Other (please specify)
What percentage of fruit that is cooled at your cold store is for export ?		
Are you located near any major road or railway , if so, which one?		
How many days a week do you take in and/or load out fruit (e.g. $5\frac{1}{2}$, 7)?		
How many pallets can be cooled to the required temperature in a one week cycle?		
Do you have a separate cold room where fruit can be stored once it has reached the required temperature? If yes, how many pallets can be stored?		
Where is fruit that is cooled at your facility transported next (e.g. cold store at or near harbour, harbour, etc. — please specify harbour)?		
Is road or rail used to transport fruit to its next destination?		
How long does it take to reach the harbour from your facility (hours)?		
Which cold stores and/or terminals are used at the harbours?		
Is there any other information you would like to bring to our attention?		

Figure 3.2: An example of a cold store questionnaire.

whose data had not been received. Much data was collected in this manner, but unfortunately the data set remains incomplete in some regions, due to a low response and cooperation rate from some individual facilities. In a bid to compensate for the incompleteness of the data set, additional capacities will be assigned to the infrastructure in production regions if the export volumes from these regions are found to be higher than the known capacity. The capacity data learned via the questionnaires are summarised in Tables 3.1 and 3.2.

3.2 Fruit Production and Transportation Data

In this section the fruit production, export and transportation data used in the Fruitlog Scenario Model are presented and discussed. The transportation data are those data used to generate the maps in §2.2.1 (see Figures 2.13, 2.14, 2.15, 2.16 and 2.18). All of the data were later combined to examine how efficient the ports are and in which times of the year they would be closest to capacity. The closer to full capacity the ports, the more likely that bottlenecks will occur, especially if there is a breakdown of machinery in a terminal.

3.2.1 Fruit Production Data by Region

Members of the modelling team met other fruit industry role-players to obtain estimates of fruit production and fruit export volumes in tons. Once role-players representing all fruit producers had agreed to the data that was gathered, it could be used for modelling purposes. The resulting data are shown in Tables 3.3 to 3.9.

The largest production region of grapes is clearly the Boland region (59% of total production) in the Western Cape, followed by the Northern Cape (29% of total production) which produces about half the volume of the Boland region. The other regions contributed the remaining 12% to the total production of grapes. The region with the largest expected absolute growth is the Northern Cape, with an expected increase in production of 55 000 tons to 139 000 tons by 2010 (65% of current production). However, the production from Namibia is expected to increase by 313% to 33 000 tons, making it the fastest growing region (in relative terms) exporting table grapes from Southern Africa. The production of table grapes is expected to increase more than any other fruit-type exported from South Africa. The expected national growth in table grape production is 43%, compared to a total growth of 17% for all fruit-types combined. Exports are also due to increase by an estimated 50%, compared to 35% expected growth in exports of all fruits by 2010. Table grape production accounted for 7% of all fruit production and 13% of all fruit export from South Africa during 2003.

The production data for the citrus fruit was underestimated, as the tree census data available from Citrus South Africa (CSA) was incomplete [291]. However, the export volumes are expected to be accurate. The largest volume of export fruit comes from the Limpopo region, where 35% of all export of hard citrus originates. It is also the region where the greatest expansion is expected, an extra 156 000 tons of hard citrus is expected to be exported from the Limpopo region by 2010. The Eastern Cape exports the second largest volume of hard citrus (20% of the total hard citrus export in 2003), followed by the Mpumalanga (13%), Senwes/North West (12%) and West Coast (9%) regions. The export of hard citrus is expected to increase by about 45% by 2010. Hard citrus export accounted for 56% of all fruit export from South Africa in 2003 according to the production information used in the model, but it would be more if all of the information had been known at the time the model was built. However, it does not affect the analysis of the ports as only the export information was used for that purpose.

Region	Fruit-types Packed	Capacity (pallets per week)
Levubu	Hard citrus	10 256
	All citrus	384
	Avocado	284
	Avocado, litchi	346 (A – 326)
	Mango	240
Tzaneen	Hard citrus	13 015
	Hard citrus, mango	1 540 (M – 280)
	Avocado	418
	Avocado, mango	534 (M – 510)
	Avocado, mango, litchi	840 (M – 690, L – 740)
	Avocado, litchi	220
Kiepersol	Mango	691
	Hard citrus	171
	Hard citrus, mango, litchi	550 (M – 150, L – 10)
	Avocado	436
	Avocado, mango	540
Hoedspruit	Avocado, litchi	227 (L – 75)
	Hard citrus	1 655
	All citrus	369
	All citrus, mango	650
	Hard citrus, mango	626 (C – 357)
	Hard citrus, mango, litchi	150
Nelspruit	Mango	625
	Pome, hard citrus	300 (C – 240)
	Hard citrus	1 245
	All citrus, avocado	230
	Hard citrus, avocado, mango, litchi	2 142 (A – 328, M – 219, L – 232)
	Avocado	316
	Avocado, mango, litchi	100
Malelane	Litchi	72
	Pome, hard citrus	306 (C – 214)
	Hard citrus	3 150
	Hard citrus, mango	1 950 (M – 650)
	Hard citrus, mango, litchi	650 (C – 500, L – 454)
	Mango	783
	Mango, litchi	152
Potgietersrus	Litchi	160
	Grapes, hard citrus	60
Marble Hall	Hard citrus	2 305
	Grapes, all citrus	600
	Hard citrus	5 965
Ermelo	All citrus	2 800 (E – 250)
	Pome	400
Pongola	Hard citrus	1 454
Nkwalini	Hard citrus	7 108
	All citrus	272 (C – 214)
	Avocado	65
KwaZulu-Natal Midlands	Hard citrus	160
	All citrus	535
	Hard citrus, avocado	312 (A – 120)
	Avocado	287
Eastern Cape Midlands	Hard citrus	228
	All citrus	460
Sundays River Valley	Hard citrus	2 771
	All citrus	7 580 (C – 5 312)
Karoo	Grapes, stone	109
	Stone	257

Table 3.1: The capacities of regional packhouses. Values in brackets indicate that the capacity for a fruit type is less than the capacity for another (the fruit type for which the infrastructure has lower capacity is shown in the brackets). This may be due to differences in packaging, etc. The abbreviations P, C, E and A refer to pome, hard citrus, soft citrus and avocado fruit, respectively.

Region	Fruit-types Packed	Capacity (pallets per week)
Patensie	Stone, hard citrus	100
	Hard citrus	5 066
	All citrus	1 393 (C – 983)
Langkloof	Stone, pome	6 745 (S – 1 932)
	Pome	3 677
	Pome, hard citrus	1 326 (C – 928)
Klein Karoo	Stone, pome, hard citrus	26 (S – 8, C – 18)
	Stone, hard citrus	125
	Stone, all citrus	2 420 (S – 2 345)
GEVV	Grapes, stone, pome, all citrus	5 000
	Stone, pome	6 103 (S – 3 261)
	Stone, pome, hard citrus	3 475 (P – 1 400, C – 2 100)
	Stone, pome, all citrus	780
	Pome	4 078
Boland	Stone, pome	1 100 (P – 700)
	Stone, pome, hard citrus	1 221 (S – 699, C – 623)
	Stone, pome, all citrus	2 028 (S – 1 037, C – 1 600, E – 1 872)
	Stone, hard citrus	339 (S – 250)
	Hard citrus	1 214
Ceres	Stone	215
	Stone, pome	2 085 (S – 2 065)
	Stone, pome, hard citrus	1 954 (S – 1 601, C – 1 282)
	Pome	10 133
	Pome, all citrus	100
Piketberg	Hard citrus	964
	Stone, pome, hard citrus	2 000
Citrusdal	Hard citrus	428
	Stone, pome, hard citrus	129 (S – 3, P – 12)
	Stone, all citrus	350 (S – 50, E – 200)
Orange River	Hard citrus	5 949
	Grapes, hard citrus	25
Hartswater	Grapes, all citrus	110 (E – 80)
	Stone	487
	Hard citrus	700
Johannesburg	All citrus	370 (E – 260)
	Stone	310
Rustenburg	Stone	32
	Hard citrus	26
Free State	Pome	783
Middelburg	Hard citrus	9 800

Table 3.1 (continued): *The capacities of regional packhouses. Values in brackets indicate that the capacity for a fruit type is less than the capacity for another (the fruit type for which the infrastructure has lower capacity is shown in the brackets). This may be due to differences in packaging, etc. The abbreviations P, C, E and A refer to pome, hard citrus, soft citrus and avocado fruit, respectively.*

The Boland (35%), West Coast (23%) and Eastern Cape (27%) regions collectively accounted for 85% of all soft citrus export in 2003, with the Senwes/North West region accounting for only 7%. The export of soft citrus is expected to almost double from 2003 volumes by 2010. Soft citrus export accounted for 3% of all fruit export from South Africa in 2003.

Farmers in the Boland region are expected to decrease the area of land currently under pome fruit to make space for more table grapes, citrus fruit and stone fruit, thus reducing the current production volume by approximately 10% by 2010. As this region was estimated to be responsible for 81% of all pome production and 77% of all pome export in 2003, the overall pome production and export is expected to decrease by 2010. This is despite expectations of production and export increases in the other three production regions. The Langkloof region accounted for the other 18% of production and 23% of export in 2003, with the Free State and

Region	Fruit-types Cooled	Capacity (pallets per week)
Malelane	Hard citrus, mango	350
	Hard citrus, mango, litchi	624 (C, M – 600)
	Hard citrus, litchi	100
	Mango	410
	Mango, litchi	152
Levubu	Litchi	140
	Avocado	412
	Avocado, litchi	346 (A – 326)
Tzaneen	Mango	240
	Hard citrus	80
	Hard citrus, mango	580
	Avocado	140
	Avocado, mango	564
Kiepersol	Avocado, mango, litchi	342
	Avocado, litchi	740
	Mango	440
	Avocado	526
	Avocado, mango	710
Hoedspruit	Avocado, litchi	227
	Mango, litchi	1 216
	Grapes	36
	Grapes, hard citrus	140
Nelspruit	All citrus, mango	160
	Hard citrus, mango	1 133
	Mango	670
	Hard citrus, avocado, mango, litchi	160
	Hard citrus, mango, litchi	96
Ellisras	Avocado	563
	Litchi	72
Marble Hall	Grapes	248
	Stone	522
Nkwalini	Grapes	620
	Grapes, all citrus	1 050
KwaZulu-Natal Midlands	Avocado	25
	All citrus	400
	Hard citrus, avocado	520 (C – 300)
Eastern Cape Midlands	Avocado	270
Sundays River Valley	All citrus	460
Patensie	All citrus	3 000
	Stone, hard citrus	1 224 (C – 816)
Langkloof	Hard citrus	1 114
	Stone, pome	3 340
	Stone, pome, hard citrus	2 300
	Stone, pome, all citrus	240
Klein Karoo	Pome	3 307
	Grapes, stone, all citrus	308
GEVV	Stone, pome	175
	Grapes, stone, pome	3 061 (G, S – 937)
	Grapes, stone, pome, hard citrus	500
	Grapes, stone, pome, all citrus	5 000
	Stone, pome	16 110 (S – 15 860)
	Stone, pome, hard citrus	1 188
De Doorns	Pome	2 440
	Grapes	13 704
Piketberg	Grapes, stone, pome	200
	Grapes	800
	Grapes, stone, pome, all citrus	2 650
	Grapes, pome, all citrus	1 100

Table 3.2: The capacities of the regional cold stores. Values in brackets indicate that the capacity for a fruit type is less than the capacity for another (the fruit type for which the infrastructure has lower capacity is shown in the brackets). This may be due to differences in packaging, etc. The abbreviations P, C, E and A refer to pome, hard citrus, soft citrus and avocado fruit, respectively.

Region	Fruit-types Cooled	Capacity (pallets per week)
Boland	Grapes	437
	Grapes, stone, pome, hard citrus	600
	Grapes, stone, pome, all citrus	1 080
	Grapes, stone, hard citrus	150
	Grapes, stone, all citrus, avocado	2 400
Ceres	Grapes, all citrus	1 860
	Grapes	80
	Grapes, stone, pome, all citrus	2 529
	Grapes, pome, all citrus	550
	Stone, pome	2 185
	Stone, pome, hard citrus	120
	Stone, pome, soft citrus	180 (P – 60)
	Pome	2 364
	Pome, hard citrus	1 000
Pome, all citrus	150	
Citrusdal	Grapes, all citrus	300
Orange River	Grapes	75 970
	Grapes, hard citrus	3 300
	Grapes, all citrus	756
Hartswater	Grapes, stone	180
	Grapes, stone, all citrus	68
Johannesburg	Grapes	570
	Grapes, stone	246
	Grapes, hard citrus	126
	All	800
	Grapes, stone, avocado, mango, litchi	350
	Stone	120
Rustenburg	Avocado	84
	Grapes, stone	36
Free State	Grapes	880
	Pome	300

Table 3.2 (continued): The capacities of the regional cold stores. Values in brackets indicate that the capacity for a fruit type is less than the capacity for another (the fruit type for which the infrastructure has lower capacity is shown in the brackets). This may be due to differences in packaging, etc. The abbreviations P, C, E and A refer to pome, hard citrus, soft citrus and avocado fruit, respectively.

Production Regions	Production			Export		
	2003	Growth From 2003 – 2010	2010	2003	Growth From 2003 – 2010	2010
Northern Cape	84.0	55.0	139.0	63.0	48.6	111.6
Boland (Klein Karoo)	170.0	30.0	200.0	135.0	26.5	161.5
West Coast	9.0	4.0	13.0	6.0	3.5	9.5
Mpumalanga-Lowveld	8.0	6.0	14.0	5.0	5.3	10.3
Senwes/North West	4.0	2.0	6.0	2.0	1.8	3.8
Limpopo	5.0	2.0	7.0	4.0	1.8	5.8
Namibia	8.0	25.0	33.0	7.0	24.0	31.0
Total	288.0	124.0	412.0	222.0	111.6	333.6

Table 3.3: The estimated production and export volumes of table grapes per region. Note that all values are in thousands of tons.

Mpumalanga regions producing very small volumes in comparison. Very little production or export growth is expected from these three regions. Pome production accounted for 26% of all fruit production and 23% of all fruit export from South Africa in 2003.

The production and export of pome is dominated by the Boland region, which produced and exported 90% of all stone fruit in 2003. Of the remaining production and export, 5% originated from the Langkloof region, with the Northern Cape, Free State, Mpumalanga, Senwes/North

Production Regions	Production			Export		
	2003	Growth From 2003 – 2010	2010	2003	Growth From 2003 – 2010	2010
Northern Cape	16.9	5.5	22.4	6.0	2.7	8.7
Boland (Klein Karoo)	50.8	16.4	67.2	22.4	10.2	32.6
West Coast	203.2	65.5	268.6	89.7	40.6	130.3
Langkloof	0.3	0.1	0.3	0.1	0.1	0.2
Eastern Cape	728.0	234.6	962.6	198.0	89.6	287.6
Pongola/Nkwalini	153.0	49.3	202.3	63.9	28.9	92.8
Swaziland	67.7	21.8	89.5	23.2	10.5	33.7
Mpumalanga-Lowveld	220.1	70.9	291.0	126.1	57.1	183.2
Senwes/North West	101.6	32.7	134.3	119.9	54.3	174.2
Limpopo	145.6	46.9	192.5	344.6	156.0	500.7
Zimbabwe	6.8	2.2	9.0			
Total	1 693.9	545.8	2 239.8	994.0	450.0	1 444.0

Table 3.4: The estimated production and export volumes of hard citrus per region. Note that all values are in thousands of tons.

Production Regions	Production			Export		
	2003	Growth From 2003 – 2010	2010	2003	Growth From 2003 – 2010	2010
Northern Cape	0.5	0.2	0.8	0.3	0.3	0.5
Boland (Klein Karoo)	46.7	17.9	64.7	18.3	17.5	35.9
West Coast	31.2	12.0	43.1	12.2	11.7	23.9
Eastern Cape	67.5	25.9	93.4	14.1	13.4	27.5
Pongola/Nkwalini	1.7	0.7	2.4	0.9	0.9	1.8
Swaziland	3.5	1.3	4.8	0.6	0.6	1.2
Mpumalanga-Lowveld	5.2	2.0	7.2	0.1	0.1	0.2
Senwes/North West	5.2	2.0	7.2	3.8	3.7	7.5
Limpopo	12.1	4.6	16.8	2.0	1.9	3.9
Total	173.6	66.6	240.2	52.3	50.0	102.3

Table 3.5: The estimated production and export volumes of soft citrus per region. Note that all values are in thousands of tons.

Production Regions	Production			Export		
	2003	Growth From 2003 – 2010	2010	2003	Growth From 2003 – 2010	2010
Boland (Klein Karoo)	900.0	-90.0	810.0	307.8	-9.2	298.6
Langkloof	202.0	2.0	204.0	90.0	0.5	90.5
Free State	3.0	1.0	4.0	1.1	1.0	2.1
Mpumalanga-Lowveld	3.0	1.0	4.0	1.1	1.0	2.1
Total	1 108.0	-86.0	1 022.0	400.0	-6.8	393.2

Table 3.6: The estimated production and export volumes of pome fruit per region. Note that all values are in thousands of tons.

West and Limpopo regions accounting for the other 5%. Stone fruit accounted for 7% of all fruit production in South Africa in 2003, and 3% of all fruit export by mass.

Most of the subtropical fruit grown in South Africa originates in the Mpumalanga region, where 33% of all subtropical fruit is grown and 42% of exported subtropical fruit originates. The Eastern Cape is the second largest producer of subtropical fruit, producing 25% of subtropical fruit in 2003, but exporting none of it. The third largest subtropical fruit producing region is Limpopo, which is also the largest exporter of subtropical fruit; accounting for 53% of all exported subtropical fruit in 2003. The Pongola/Nkwalini and Midlands regions in KwaZulu-Natal each produced approximately 70 000 tons of mainly avocados in 2003, together producing

Production Regions	Production			Export		
	2003	Growth From 2003 – 2010	2010	2003	Growth From 2003 – 2010	2010
Northern Cape	3.0	0.6	3.6	0.5	0.1	0.6
Boland (Klein Karoo)	270.9	54.2	325.1	47.7	10.4	58.1
Langkloof	15.1	3.0	18.1	2.7	0.6	3.2
Free State	3.0	0.6	3.6	0.5	0.1	0.6
Mpumalanga-Lowveld	3.0	0.6	3.6	0.5	0.1	0.6
Senwes/North West	3.0	0.6	3.6	0.5	0.1	0.6
Limpopo	3.0	0.6	3.6	0.5	0.1	0.6
Total	301.0	60.2	361.2	53.0	11.5	64.5

Table 3.7: The estimated production and export volumes of stone fruit per region. Note that all values are in thousands of tons.

Production Regions	Production			Export		
	2003	Growth From 2003 – 2010	2010	2003	Growth From 2003 – 2010	2010
Eastern Cape	177.0	8.9	185.9			
KZN Midlands	70.0	3.5	73.5	2	0.1	2.1
Pongola/Nkwalini	70.0	3.5	73.5			
Swaziland	7.0	0.4	7.4			
Mpumalanga-Lowveld	231.0	11.6	242.6	19.0	1.0	20.0
Senwes/North West	5.5	0.3	5.6	0.5	0.0	0.5
Limpopo	147.0	7.4	154.4	24.0	1.2	25.2
Total	707.5	35.4	742.9	45.5	2.3	47.8

Table 3.8: The estimated production and export volumes of subtropical fruits (mainly avocados, litchis and mangoes) per region. Note that all values are in thousands of tons.

about 20% of all subtropical fruit, but only exporting 4% of exported subtropical fruit from the Midlands region. Subtropical fruits accounted for 17% of all fruit production and 3% of fruit export in 2003.

Production Regions	Production			Export		
	2003	Growth From 2003 – 2010	2010	2003	Growth From 2003 – 2010	2010
Northern Cape	104.5	61.3	165.8	69.8	51.7	121.5
Boland (Klein Karoo)	1 438.4	28.5	1 466.9	531.3	55.4	586.7
West Coast	243.2	81.4	324.7	107.9	55.8	163.8
Langkloof	217.3	5.1	222.4	92.8	1.1	93.9
Eastern Cape	972.5	269.3	1 241.8	212.0	103.1	315.1
Free State	6.0	1.6	7.6	1.6	1.1	2.7
KZN Midlands	70.0	3.5	73.5	2.0	0.1	2.1
Pongola/Nkwalini	224.7	53.5	278.2	64.8	29.8	94.6
Swaziland	78.2	23.5	101.5	23.8	11.1	34.9
Mpumalanga-Lowveld	470.3	92.1	562.4	151.8	64.6	216.4
Senwes/North West	119.3	37.6	156.9	126.8	59.9	186.6
Limpopo	312.7	61.5	374.2	375.2	161.0	536.2
Zimbabwe	6.8	2.2	9.0			
Namibia	8.0	25.0	33.0	7.0	24.0	31.0
Total	4 272.0	746.1	5 018.1	1 766.8	618.6	2 385.4

Table 3.9: The estimated production and export volumes of all exported fruits per region. Note that all values are in thousands of tons.

From Table 3.9 it may be seen that the largest fruit producing and exporting region in 2003 (in terms of mass of fruit) was the Boland region, producing 34% of all fruit and exporting 30% of all exported fruit. The second largest fruit production region was the Eastern Cape,

responsible for 23% of fruit production and 12% of fruit export. Mpumalanga is the third largest, producing 11% of fruit and exporting 9% of all the fruit exported in 2003. The Limpopo region is the second highest exporter of fruit, as 21% of export fruit originated from this area in 2003. The discrepancy between the production in and export from the Limpopo region is due to the incomplete production data obtained from CSA.

3.2.2 Fruit Export Data

The export data of table grapes, hard citrus and soft citrus are such that the export from each region is known for every week. The stone, pome and subtropical fruit information is not as detailed, thus only weekly export is known, so all regions are assumed to produce fruit at the same percentage of the total each week (hence they are not individually plotted and can be found in Figure 3.6). So, the peak export week for all regions is the same; while for table grapes, hard citrus and soft citrus the peak weeks may differ from region to region.

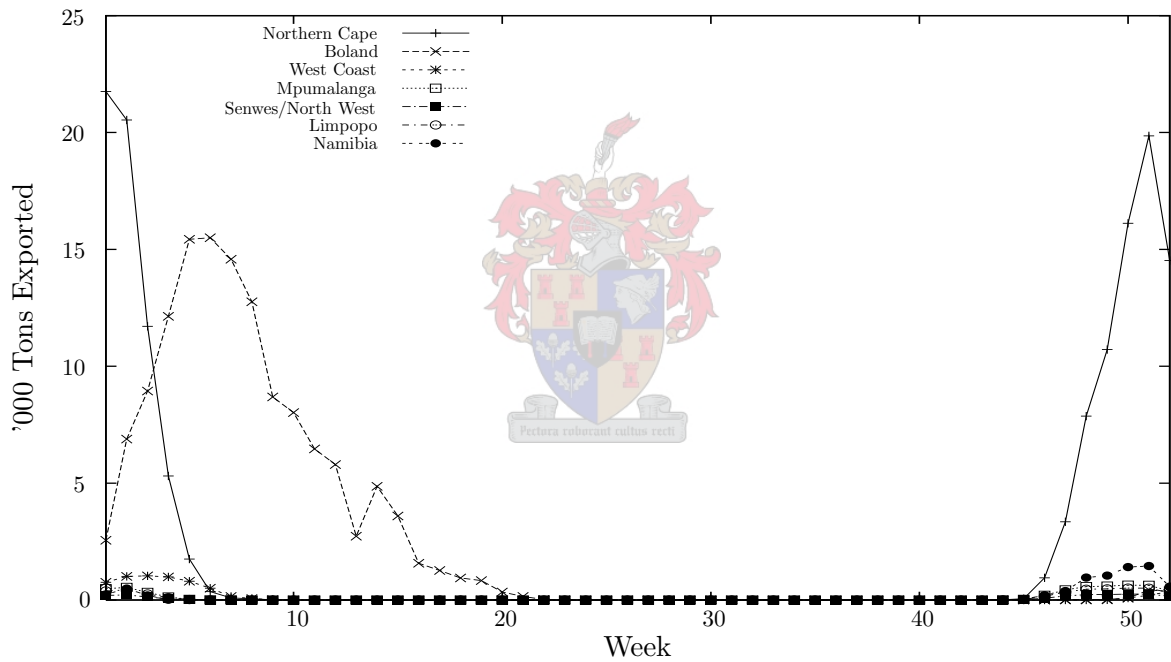


Figure 3.3: Volumes of grapes exported per region per week. Note that the regions may have different peak weeks. The export volume from the Mpumalanga, Senwes/North West and Limpopo regions was only available as the export from a collective region incorporating these three regions, thus they have the same profile.

Figure 3.3 shows the export of table grapes from each region for each week. From the graph it is clear that the Northern Cape and Boland/Klein Karoo regions produce the largest volumes of export table grapes. The harvesting season for the Northern Cape is at the end of one calendar year and early in the next. It is thus viable for the producers from this region to send grapes to foreign markets by air freight, as very high prices are available in Europe before the new year begins [291, 191]. The Boland region harvests the largest volumes of export grapes during weeks 5 to 7, which means that producers may expect worse prices for the grapes, as more countries have entered the market by that stage [191]. The Namibia region exports its largest volumes of grapes at the same time as the Northern Cape, but this is expected as the regions are

on opposite sides of the Orange River on the border between South Africa and Namibia. The volumes from the Namibia region are very small compared to the volumes from the Northern Cape and Boland regions, but the growth in this region is expected to increase the most in relation to its current size (see Table 3.3).

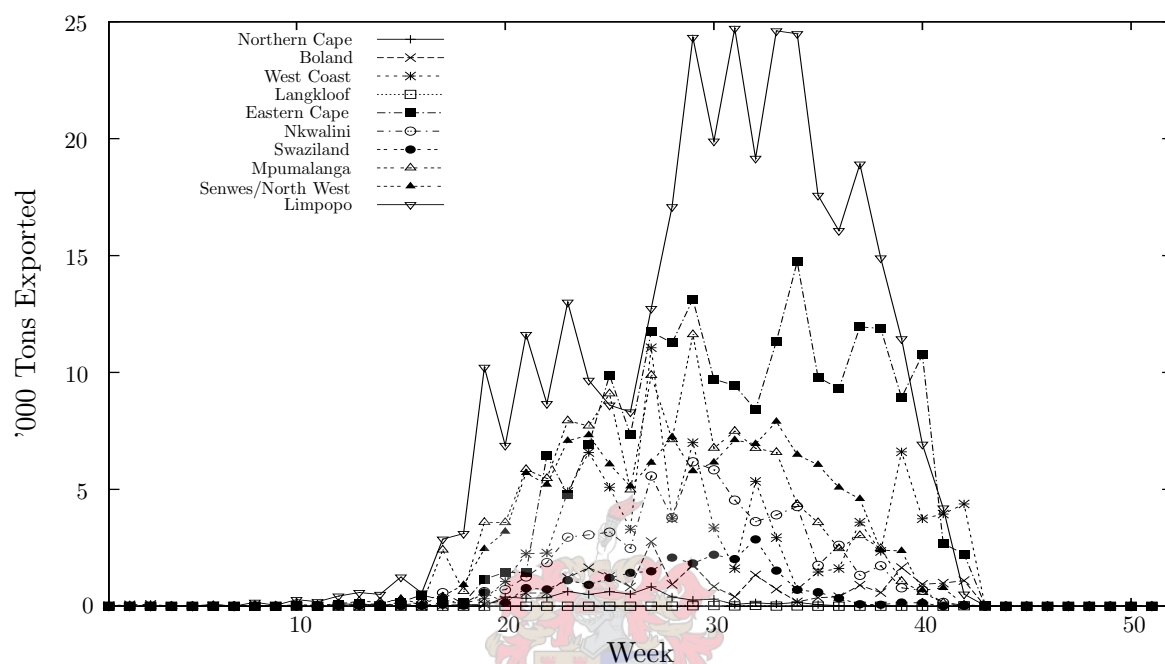


Figure 3.4: Volumes of hard citrus exported per region per week. Note that the regions may have different peak weeks. The export volume from the Boland/Klein Karoo and West Coast regions was only available as the export from a collective region incorporating these two regions, thus they have the same profile.

The hard citrus export per week from the various regions shown in Figure 3.4 does not have single peaks that are high, such as with the export of grapes. Instead, the export of citrus is spread over a larger number of weeks. It is also clear from the graph that export volumes of hard citrus are less sensitive with respect to the regions in which it is grown. All regions have similar peak weeks, where the largest volumes of hard citrus is exported during weeks 27 – 39.

The lack of a clear differentiation between harvesting peaks is also apparent in the harvesting volumes of soft citrus. Figure 3.5 shows that most of the soft citrus is exported during weeks 15 – 29; hence soft citrus has an earlier season than hard citrus. However, the peak volumes of soft citrus are only about a tenth of the volumes of hard citrus or table grapes, thus exerting far less pressure on the export infrastructure.

The weekly export data for all fruits is summarised in Figure 3.6.

3.2.3 Summary of National Fruit Export Data

The expected fruit production volumes were calculated for each fruit type per week for each of the refined production regions. These new regions are no longer those listed in Tables 3.3 to 3.9, but have been reduced in number such that transportation costs could be used more effectively. Groupings of towns with packhouses and cold stores that are located in close proximity were

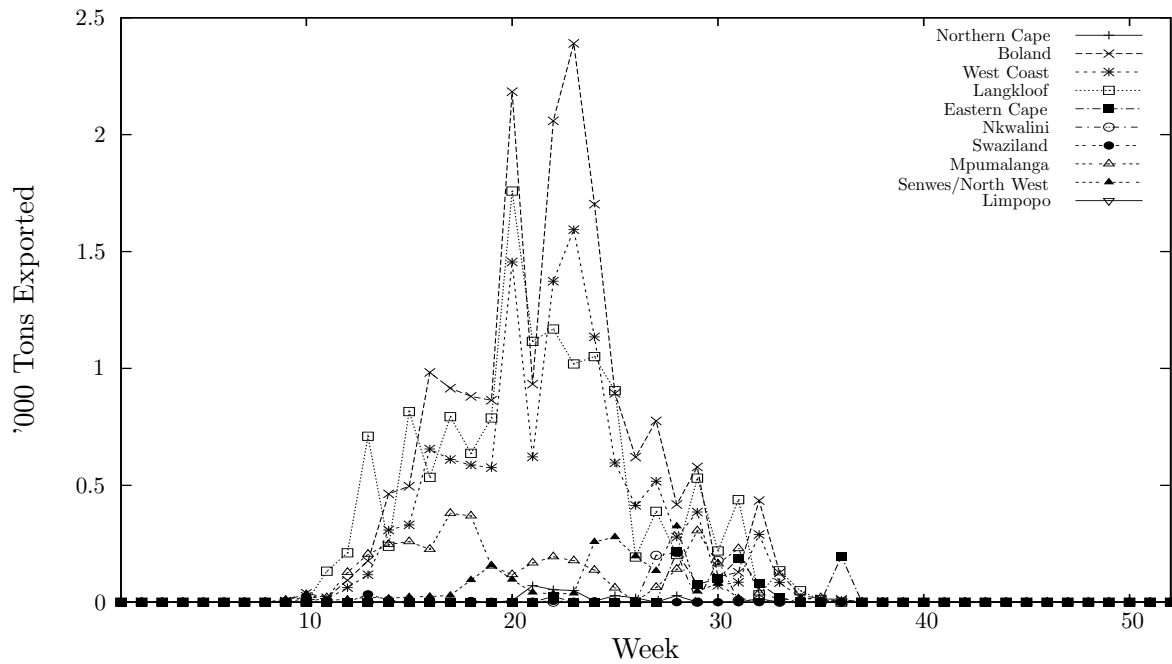


Figure 3.5: Volumes of soft citrus exported per region per week. Note that the regions may have different peak weeks. The export volume from the Boland/Klein Karoo and West Coast regions was only available as the export from a collective region incorporating these two regions, thus they have the same profile.

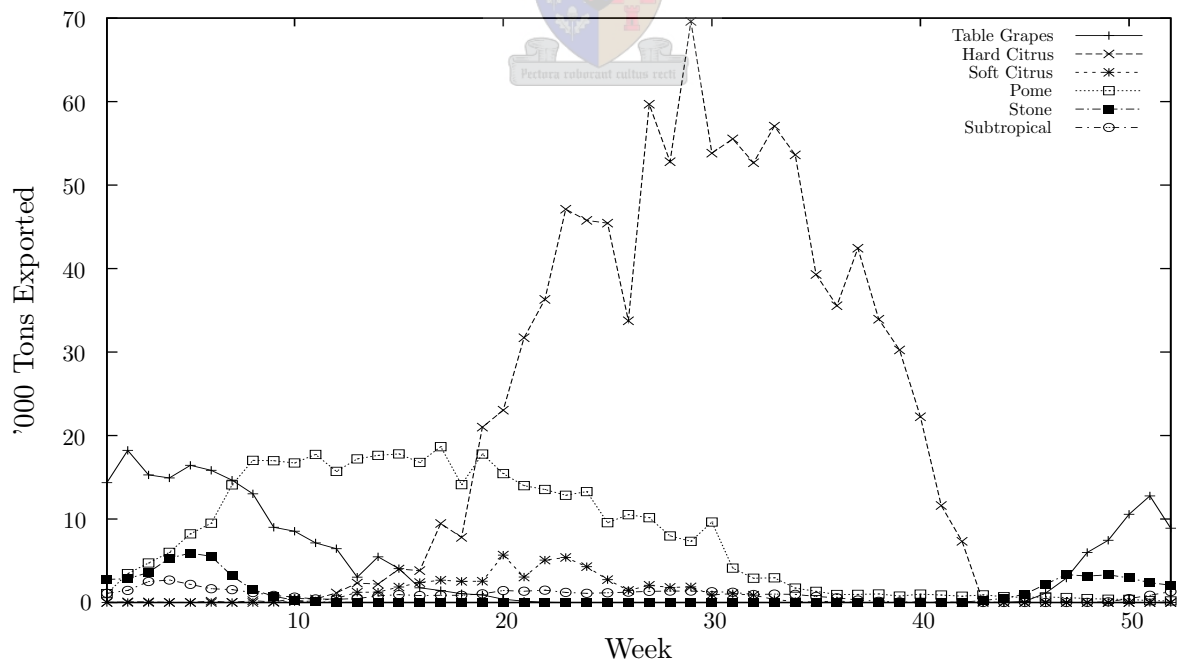


Figure 3.6: Volumes of all six fruit types considered in the Fruitlog Scenario Model during each week of 2003.

assumed to constitute a region (see Appendix B for a list of packhouse/cold store regions). Sometimes the names given to production regions differ between the role-players in each fruit type, for example, the Letsitele citrus region is the same geographical region called “Tzaneen” by the subtropical fruit role-players. The names given to the production regions by the role-players involved with those fruit types have been retained in this thesis.

The fruit production figures available for subtropical fruits (avocados, mangoes and litchis) derive from various sources. The South African Mango Growers’ Association (SAMGA) [69] provided information on mango and litchi exports that had been provided to them by the PPECB, which included the volume of export fruit that was inspected and rejected in each production region. This was used in conjunction with weekly shipping data, to estimate the export of litchis and mangoes per region per week. The South African Avocado Growers’ Association (SAAGA) [265] were also able to provide detailed statistics that listed the export of each cultivar of avocado per region per week.

For the deciduous fruits, tree census data [50] was used to estimate the percentage production per region of total production. These estimates were performed with the help of Dr DB Louw [203]. The total fruit production per week was used to estimate the weekly production percentages, so that when the two percentages were multiplied together, a per region per week production estimate could be made if the total production for a season was known. In cases where two years of weekly export data were available, the estimated production per week was averaged.

For citrus fruit, very detailed weekly shipping data information were available from Citrus South Africa (CSA) [297]. This included the volume of fruit that was shipped, the production region from where the fruit originated, and the target market. This allowed for the compilation of a very accurate data set.

There are two weeks of fruit export in 2003 that are of particular interest. The largest volume of fruit is exported during week 29, the peak week of the winter season (see Figure 3.7). During this week only hard citrus, soft citrus, pome fruit and avocados are exported, while grapes, stone fruit, pome fruit, citrus, mangoes and litchis are all exported during week seven, the peak week of the summer season. See Table 3.10 for export volumes of fruit during these two peak weeks of 2003.

Fruit-type	Week 7 (pallets)	Week 29 (pallets)
Grapes	18 653	0
Stone	4 398	0
Pome	15 718	8 581
Hard citrus	23	70 043
Soft citrus	0	2 109
Avocados	0	1 989
Mangoes	1 120	0
Litchis	327	0
Total	40 239	82 722

Table 3.10: Fruit export during week seven of 2003 (summer peak, 10 – 16 February in 2003) and week 29 of 2003 (winter peak, 14 – 20 July in 2003). All volumes are listed in pallets.

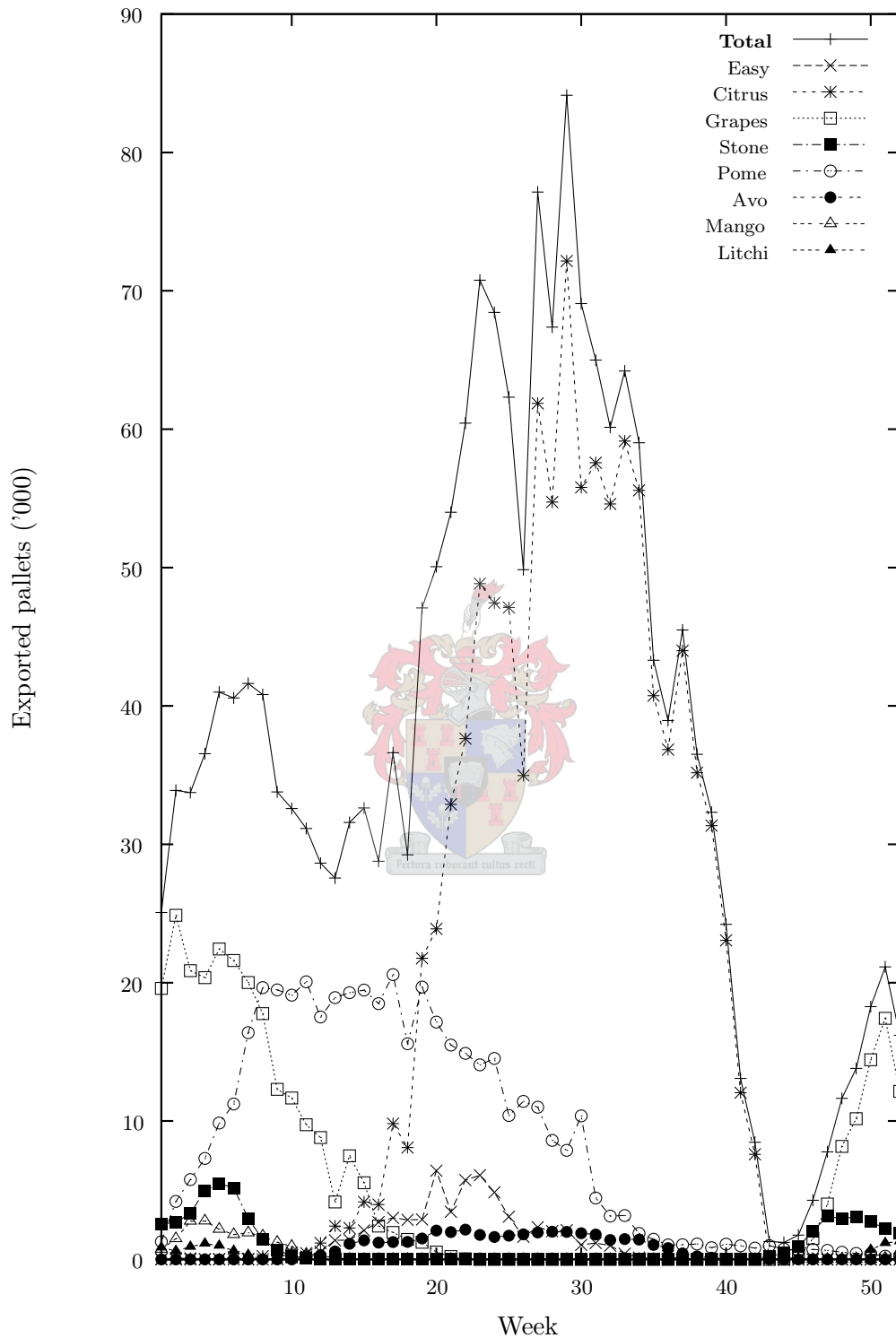


Figure 3.7: The estimated South African fruit export in 2003. Notice the winter peak volume of fruit being exported during week 29 (14 – 20 July in 2003) and the summer peak during week seven (10 – 16 February in 2003). This figure may differ from Figure 3.6 due to the difference in volume measurements; in this figure the measured volume is pallets, while tons were used as a unit of measure for Figure 3.6.

3.2.4 Origin–Destination Flow Volumes

To determine the volumes of fruit that flowed to each port during 2003, Dr FE van Dyk calculated the percentages of fruit from each production region to each port. These origin–destination flow volumes that were calculated after consultation with fruit industry role-players are listed in Appendix C. A brief description of the results is presented here, but more detail may be found in the appendix, and in §2.2.1.

All table grapes, all stone fruit and most pome fruit was exported through the port of Cape Town during 2003. However, pome fruit from the Langkloof region was sent to Port Elizabeth harbour. All deciduous fruit was transported by road. Citrus fruit was sent to all four harbours; with more than half going to Durban in 2003, and similar amounts to Cape Town and Port Elizabeth. Maputo received less than half of what was sent to either Cape Town or Port Elizabeth. Approximately one fifth of all exported citrus fruit was transported by rail to the harbours. Most (94.5% of the total) subtropical fruit passed through Cape Town on its way to foreign markets in 2003, while small amounts were sent to Durban, Port Elizabeth and Johannesburg (1.6%, 1.7% and 2.2% respectively). Just over seven percent of all subtropical fruit exported was transported to Cape Town by train. This would mainly have been avocados on the “avo train.” The volume of subtropical fruit exported through Johannesburg International Airport was also mainly avocados.

Combining the transportation data with the export volumes found in Tables 3.3 – 3.8, the volume (in tons) of fruit that was exported through each port in 2003 could be estimated (see Table 3.11 and Figure 3.8).

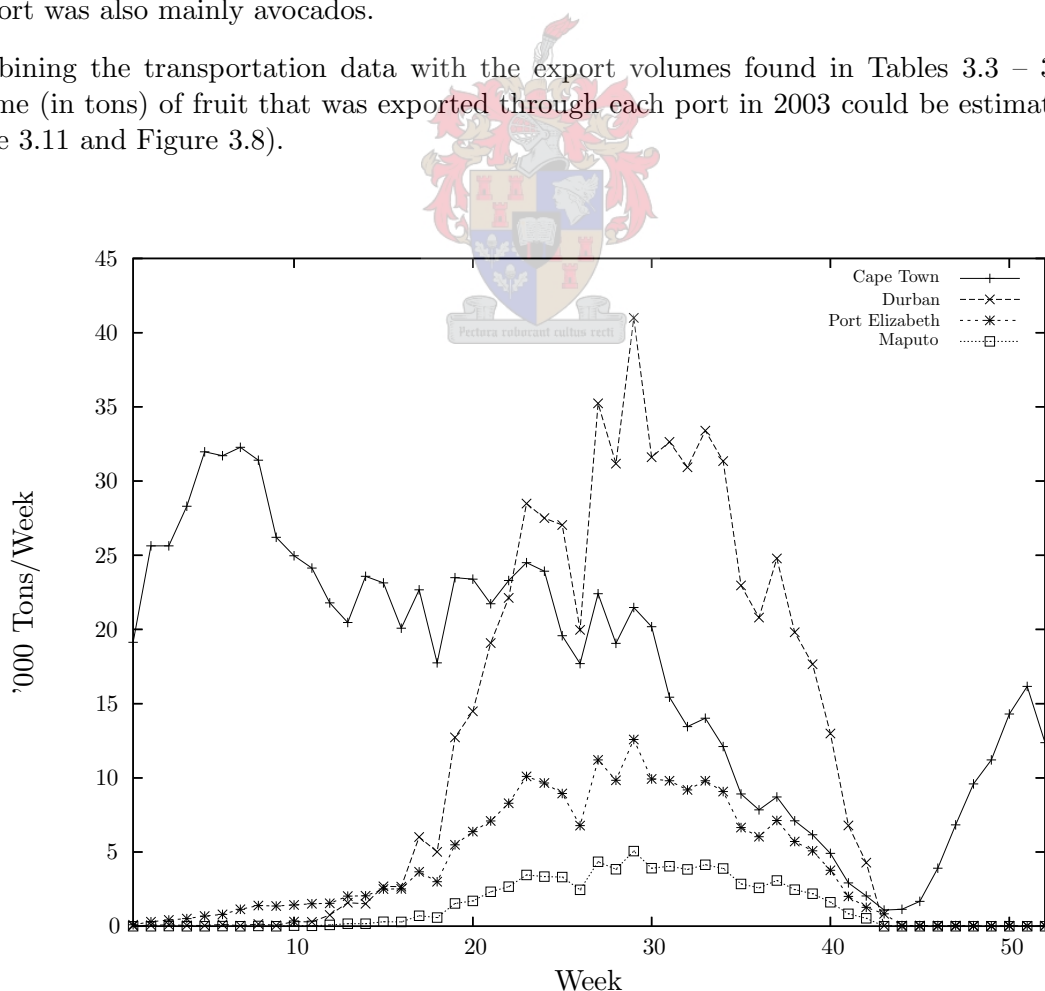


Figure 3.8: *Estimated volumes of fruit (in tons) exported per week per port.*

Week	Cape Town	Durban	Port Elizabeth	Maputo
1	19 136	34	105	2
2	25 630	71	308	5
3	25 629	96	427	6
4	28 301	57	518	1
5	31 968	54	687	2
6	31 704	121	800	11
7	32 275	42	1 139	2
8	31 400	151	1 397	16
9	26 211	58	1 366	5
10	24 962	316	1 434	36
11	24 133	292	1 518	32
12	21 792	759	1 544	86
13	20 468	1 582	2 030	176
14	23 581	1 515	2 052	167
15	23 139	2 684	2 510	302
16	20 084	2 665	2 520	292
17	22 669	6 014	3 672	704
18	17 753	5 016	3 007	582
19	23 492	12 730	5 493	1 540
20	23 388	14 482	6 379	1 708
21	21 734	19 085	7 095	2 322
22	23 302	22 123	8 290	2 668
23	24 503	28 482	10 100	3 454
24	23 924	27 505	9 655	3 350
25	19 577	27 038	8 948	3 317
26	17 701	19 976	6 783	2 459
27	22 409	35 236	11 206	4 347
28	19 066	31 173	9 833	3 845
29	21 478	40 993	12 584	5 066
30	20 186	31 608	9 926	3 914
31	15 431	32 634	9 806	4 039
32	13 456	30 926	9 192	3 830
33	14 015	33 396	9 805	4 146
34	12 110	31 343	9 081	3 894
35	8 916	22 968	6 652	2 854
36	7 843	20 800	6 033	2 583
37	8 717	24 782	7 126	3 082
38	7 105	19 811	5 715	2 464
39	6 162	17 661	5 083	2 196
40	4 911	12 990	3 773	1 616
41	2 926	6 791	2 009	844
42	2 046	4 282	1 277	533
43	1 092	44	842	5
44	1 143	0	54	0
45	1 679	0	48	0
46	3 914	0	52	0
47	6 833	0	48	0
48	9 589	0	38	0
49	11 205	1	33	0
50	14 301	9	37	0
51	16 163	17	31	0
52	12 377	25	36	0
Total	893 527	590 438	209 339	72 501

Table 3.11: *Estimated volumes of fruit exported per week per port. All values listed in tons.*

3.3 Port Capacities

In this section, the capacities of the export infrastructure in the four fruit export harbours are considered. These include the capacities of all four FPT terminals, both SAFT terminals and the container terminals in all of the ports. The capacities found here upper bounds, but there are many factors that may reduce the capacities in practice. Some examples of such interference

include weather (high wind speeds and/or rain may slow down loading, or bring it to a complete halt), labour disruption (striking or unproductive workers can prevent 100% efficiency), large volumes of traffic (this could prevent vehicles from arriving at the terminals when scheduled, disrupting the loading process which requires the pallets/containers to be placed in a certain order or pattern), incorrect documentation (fruit pallets may then not arrive at the berth on time), fruit that is not on temperature, cold stores not being able to load fruit fast enough in the correct sequence (the large cold rooms in some cold stores experience difficulties in finding the exact pallets that should be transported to the harbour), and equipment breakdown (resulting in a direct loss of capacity). Inevitably, at least one of these may occur during the same week, which would reduce the capacities significantly.

One worksheet of the model (which may be found on the compact disc accompanying this thesis, as explained in Appendix H) contained all the assumptions made about the ports to determine how efficient they were. By varying these assumptions, the effects on the performance and capacities of the ports could be studied. Changing the assumptions allows the user to mimic decreasing efficiency due to some of the reasons mentioned above, or to mimic increases in efficiency. The necessary data for the ports of Cape Town, Durban, Port Elizabeth and Maputo were gathered and inserted into the model. The worksheet infrastructure for the port of Lüderitz was also inserted, but as there is no infrastructure for fruit export yet, it will be excluded in this discussion.

3.3.1 Container Terminals

The first item of information required to determine the capacity of the container terminals is the number of plug-in points in the stack. This, multiplied by the number of days in a week, divided by the average dwell-time (in days) of integral containers in the stack, results in an estimation of the number of integral containers that may pass through the stack. Subtracting the number of containers that are imported and the number of export integral containers not containing fruit, leaves the number of plug-in points available for fruit export in integral containers. This value is the first of four capacities found in the port, the lowest of which is taken as the capacity of the port (for the *Fruitlog Scenario Model*, the stack was considered separately).

The next three possible limitations to the port require a computation of the number of vessels that can be accommodated by the various container terminals per week. Once this parameter has been calculated, one can calculate the spare loading capacity for export containers. This is the first of the three limitations and the calculations for the three container terminals in South Africa are shown below. Multiplying the number of vessels that can be accommodated per week by either half of the call size¹ of the vessels or the number of plug-in points per vessel result in the third and fourth parameters. As the number of plug-in points per vessel are always larger than half of the call size according to information gathered by the team that created the *Fruitlog Scenario Model* [291], it can be ignored in further calculations (it was included for the case where there are only few plug-in points on a vessel, but the call size is large due to unrefrigerated containers). Since the call size is included in the calculation for the number of vessels per week, it is always larger than the spare capacity, as the spare capacity has the number of imported and non-fruit containers exported, subtracted from the total number of containers that could be handled — which is essentially the call size multiplied by the number of vessels that can be handled per week.

¹The call size of a vessel is the number of container moves performed on the vessel. If, for example, 225 containers are offloaded and 221 containers loaded onto the vessel, the call size is 446.

The calculations for the capacities of the container terminals in Cape Town, Durban and Port Elizabeth are shown separately below. Maputo was not included as shipping lines berth at the port infrequently compared to ports in South Africa. A summary of the results of the calculations are shown in Table 3.12. Although the plug-in points are always the lower value for the 2003 data and thus the bottleneck in the *Fruitlog Scenario Model*, both have been included in the calculations, because they are required for modelling purposes later in this thesis. For 2004 data estimates, the loading capacity becomes the limiting factor as there is an expected increase in container traffic and the number of plug-in points is expected to increase by approximately 50%, from 995 to 1 445 plug-in points.

Port	Plug-in points for fruit export	Container loading capacity
Cape Town	1 599	2 919
Durban	748	6 114
Port Elizabeth	645	2 836

Table 3.12: *The loading and stack capacities for integral containers containing fruit. All volumes in containers. Source: Fruitlog Scenario Model [292].*

Cape Town Container Terminal

The stacks in the Cape Town container terminal (with 995 plug-in points) close twice a week, thus the reefer stack may be used twice a week. On average, 319 non-fruit containers pass through the stack during a week, leaving $995 \times 2 - 319 = 1\,599$ plug points for export fruit integral containers. According to information from South African Port Operations in Cape Town [291], 3.7% of all integral containers passing through the stack are 6m containers and the remaining 96.3% are 12m integral containers. This means that $(1\,599 \times 3.7\% \times 10) + (1\,599 \times 96.3\% \times 20) \approx 590 + 30\,800 = 31\,390$ pallets of fruit could potentially pass through the stack per week.

There are six gantry cranes (that each achieve 17 moves per crane hour) available during the peak weeks (usually one is down for maintenance, *i.e.* five would operate at any one time) and three berths may accommodate container vessels, thus $17 \times 6/3 = 34$ containers may either be loaded or offloaded per hour per berth. However, each eight-hour shift usually only contains 7.5 effective working hours². So, for vessels with an average call size of 446 containers, it takes $446/(34 \times 7.5/8) = 14$ hours to complete the offloading and loading of a single vessel. Assuming it takes five hours to deberth the vessel and berth the next, the average turn-around time for a vessel is estimated to be 19 hours. The container terminal can therefore handle $24 \times 7/19 = 8.84$ vessels per berth per week, or 26.5 per week in total. This means that $26.5 \times 446 = 11\,831$ container moves may be performed per week. Since approximately the same number of containers are offloaded as are loaded in Cape Town, and an average of 2 999 non-fruit containers are loaded onto vessels per week, $11\,831/2 - 2\,999 = 2\,916$ container export moves per week remain that may possibly be taken by fruit integral containers. Of all fruit containers that are loaded onto vessels, 42.8% are 6m reefer containers (either porthole or integral) and the remaining 57.2% are 12m integrals. The additional containers that are not loaded via the

²The operators of the gantry cranes are not able to work a full eight hour shift as they have to bring the crane to a complete stop before the replacement operator may climb onboard. The replacement operator must then wait for the previous operator to climb off the crane before recommencing with the loading of a vessel. Tea and lunch breaks also make up some of the down-time of the crane.

stack are either porthole containers, or direct shipments³. Thus $(2916 \times 42.8\% \times 10) + (2916 \times 57.2\% \times 20) \approx 12480 + 33360 = 45840$ pallets of fruit may potentially be loaded onto the vessels per week.

Durban Container Terminal

The stacks in the Durban container terminal close twice a week, thus the reefer stack may be used twice a week. On average, 452 non-fruit containers pass through the stack in a week, leaving $600 \times 2 - 452 = 748$ plug points for export fruit integral containers. There was no detailed data available regarding the split between small and large reefer containers of fruit for the port of Durban, so 30% were assumed to be 6m containers and the other 70% to be 12m integral containers⁴. Under this assumption $(748 \times 30\% \times 10) + (748 \times 70\% \times 20) \approx 2240 + 10480 = 12720$ pallets of fruit may be exported through Durban via the stack per week.

There are 13 gantry cranes (that each achieve 18 moves per crane hour) available during the peak weeks, and seven berths may accommodate container vessels; thus $18 \times 13/7 = 33.4$ containers can either be loaded or offloaded per hour per berth. However, each eight-hour shift usually only contains 6.67 effective working hours. So, for vessels with an average call size of 863 containers, it takes $863/(33.4 \times 6.67/8) = 31.0$ hours to complete the offloading and loading of a single vessel. Assuming it takes four hours to de berth a vessel and berth the next, the average turn-around time for a vessel is estimated to be 35.0 hours. The container terminal can therefore handle $24 \times 7/35.0 = 4.80$ vessels per berth per week, or 33.6 per week in total. This means that $33.6 \times 863 = 29014$ container moves can be performed per week. Since approximately the same number of containers are offloaded as are loaded in Durban, and an average of 8393 non-fruit containers are loaded onto vessels per week, $29014/2 - 8393 = 6114$ container export moves per week remain that could possibly be taken by fruit integral containers. Assuming the 30/70 split of 6m and 12m containers also applies to the loading rate, $(6114 \times 30\% \times 10) + (6114 \times 70\% \times 20) \approx 18340 + 85600 = 103940$ pallets of fruit may be loaded onto container vessels per week.

Port Elizabeth Container Terminal

The average dwell-time of integral containers in Port Elizabeth is only 2.8 days, compared to the 3.5 days of the two larger terminals in Cape Town and Durban. On average, 90 non-fruit containers pass through the stack in a week, leaving $294 \times 7/2.8 - 90 = 645$ plug points for export of fruit integral containers. As all fruit containers would have to pass through the stack on the way to the gantry cranes, it is assumed that the percentage split is the same as for the loading rate. From container export information from SAPO in Port Elizabeth [291], it is estimated that 21.8% of reefer containers are of the 6m variety, while the remaining 78.2% are the longer version. Hence, $(645 \times 21.8\% \times 10) + (645 \times 78.2\% \times 20) \approx 1400 + 10080 = 11480$ pallets of fruit could be exported via the stack per week.

³Direct shipments are loaded directly from the vehicle that transported the container to the terminal onto a bathtub vehicle, which is then taken to the gantry crane for loading onto the vessel. This may be done effectively if the containers are sourced from a location very close to the terminal, so that the timing of the arrival will have a low chance of being disrupted.

⁴These values were chosen as they were between the split for Cape Town and Port Elizabeth, for which more detailed data were available. By the time of submission of this thesis, no complaints had been received about the accuracy of these assumptions, which had been released via reporting deliverables more than a year prior to submission.

There are four gantry cranes (that each achieve 18 moves per crane hour) available during peak weeks, and seven berths may accommodate container vessels; thus $18 \times 4/2 = 36$ containers can either be loaded or offloaded per hour per berth. However, each eight-hour shift usually only contains 7.5 effective working hours. So, for vessels with an average call size of 371 containers, it takes $371/(36 \times 7.5/8) = 11.0$ hours to complete the offloading and loading of a single vessel. Assuming it takes four hours to de berth a vessel and berth the next, the average turn-around time for a vessel is estimated to be 15.0 hours. The container terminal could therefore handle $24 \times 7/15.0 = 11.2$ vessels per berth per week, or 22.4 per week in total. This means that $22.4 \times 371 = 8\,315$ container moves can be performed per week. Since approximately the same number of containers are offloaded than are loaded in Port Elizabeth, and an average of 1 321 non-fruit containers are loaded onto vessels per week, $8\,315/2 - 1\,321 = 2\,836$ container export moves per week remain that could possibly be taken by fruit integral containers. Thus $(2\,836 \times 21.8\% \times 10) + (2\,836 \times 78.2\% \times 20) \approx 6\,180 + 44\,360 = 50\,540$ pallets of fruit could potentially be loaded onto container vessels by the Port Elizabeth Container Terminal per week.

Maputo Container Terminal

The Maputo container terminal was disregarded in the model. Although there are 68 plug points, with a further 48 available if required, there are few vessels that berth at the port. Only the shipping line *Unifeeder* (servicing the Southern African coast) have a weekly service at Maputo, while *Mediterranean Shipping Company (MSC)* and *Global Container Line (GCL)* (both servicing only East Africa and South Asia) visit Maputo once every ten days. Furthermore, *MACS* (servicing Southern Africa and West Europe) and *Messina* (servicing East Africa and Mediterranean) only have a monthly service to Maputo [209]. Exporting through Maputo container terminal would therefore result in long delays before fruit containers reach their final destinations. This is something that exporters attempt to avoid in order to increase the shelf-life of fruit in foreign markets. It is may therefore be safely assumed that all fruit passing through Maputo is exported via the Fresh Produce Terminal berths.

The results of the estimations presented in this section are summarised in Tables 3.12 – 3.13.

Port	Stack capacity (pallets/week)	Container loading capacity (pallets/week)
Cape Town	31 390	45 840
Durban	12 720	103 940
Port Elizabeth	11 480	50 540

Table 3.13: *The weekly loading and stack capacities for fruit export in integral containers via the container terminals in Cape Town, Durban and Port Elizabeth.*

3.3.2 Bulk Terminals

The capacities of the four Fresh Produce Terminals (FPT) and two South African Fruit Terminals (SAFT) are estimated in this subsection. The volume of fruit that can be transported in conventional vessels in break-bulk or on conventional vessels in integral containers is then summed together for both terminals (if the port has both) to obtain an estimated total volume of fruit that can be exported from the break-bulk fruit terminals. The results are summarised in Table 3.14.

Port	FPT Bulk	FPT Container	SAFT Bulk	SAFT Container
Cape Town	31 500	11 200	10 500	3 733
Durban	45 320	7 769	13 140	2 253
Port Elizabeth	22 615	3 877	n/a	n/a
Maputo	22 615	3 877	n/a	n/a

Table 3.14: Break-bulk and container loading rates for all four FPT terminals in Southern Africa and the two SAFT terminals at the ports of Cape Town and Durban. All volumes are in pallets per week.

FPT Cape Town

The Fresh Produce Terminal in Cape Town has three berths. Each berth is guaranteed to load 2 000 pallets a day (if required) and each day consists of two eight-hour shifts. Assuming that the average vessel size is 4 500 pallets under deck, it takes $2\frac{1}{4}$ days to load as a result, or two days and four hours. The approximately 80 containers then have to be loaded onto the deck, but only after an average of 100 empty containers are offloaded. Assuming that between eight and ten containers may be loaded on deck per crane hour and two cranes are used, it takes approximately nine hours to offload and load all containers on deck.

The total time it takes to fill an entire vessel is thus two days and 13 hours. Allowing for berthing and deberthing time, taking into account that only 16 hours are worked per day and that berthing/deberthing may take place after the second shift, gives a total of three days to berth, load and de berth a vessel. Thus, the maximum number of pallets of fruit that can be loaded at FPT Cape Town is $7 \text{ days/week} \div 3 \text{ days/vessel/berth} \times 3 \text{ berths} = 7 \text{ vessels/week}$.

Now that the number of vessels that may be serviced in a week are known, it is possible to estimate how many pallets may be loaded onto the vessels in containers and in break-bulk. The maximum break-bulk loading rate can be calculated as follows: $7 \text{ vessels/week} \times 4 500 \text{ pallets/vessel} = 31 500 \text{ pallets/week}$. Similarly, the maximum container loading rate is $7 \text{ vessels/week} \times 20 \text{ pallets/container} \times 80 \text{ containers/vessel} = 11 200 \text{ pallets/week}$.

SAFT Cape Town

There is only a single berth at the South African Fruit Terminals in Cape Town. They guarantee a loading rate of 2 200 pallets per day, so it takes approximately two days and an hour to load a vessel with a capacity of 4 500 pallets of fruit under deck. Assuming it takes five hours for two cranes to offload all 100 empty containers and four hours to load the containers on deck, and five hours to de berth the vessel and berth the next one, it takes two days and 15 hours to turn a single vessel around. As it is unlikely that loading of the next vessel will only begin in the last hour of the 16 hour working day, one may assume it takes three days to turn a vessel around. Hence, SAFT is able to load $7 \text{ days/week} \div 3 \text{ days/vessel} = 2.33 \text{ vessels/week}$. Thus, at SAFT a maximum of $2.33 \text{ vessels/week} \times 4 500 \text{ pallets/vessel} = 10 500 \text{ pallets/week}$ could be loaded in break-bulk and a further $2.33 \text{ vessels/week} \times 20 \text{ pallets/container} \times 80 \text{ containers/vessel} = 3 733 \text{ pallets/week}$ in containers on the decks of conventional vessels per week.

FPT Durban

Fresh Produce Terminals in Durban guarantee that they can load 30 pallets per gang. The average vessel size limits loading to 3 500 pallets per vessel. At 30 pallets per gang (or hatch) it takes $3\,500/(30 \times 4) = 29.2$ hours to load the pallets under deck. The cranes are able to load six containers per crane hour. Typically 30 empty integral containers must be offloaded and 30 full integral containers loaded onto the deck per vessel. Using two cranes, it takes $30/(6 \times 2) = 2.5$ hours to either offload or load the containers. Allowing for eight hours to de berth a vessel and berth the next vessel, this estimate therefore gives an average turn-around time of 42.2 hours per vessel.

The terminal operates 24 hours a day, six days a week and an additional 12 hours on the seventh day, giving a total of 156 working hours per week. FPT Durban may thus handle $156/42.2 = 3.70$ vessels per berth per week. They have 3 permanent berths available, but a fourth may be used if the port has no other use for it at that time, resulting in an average of 3.5 berths per week. The maximum number of vessels that FPT Durban may load in a week is therefore $3.5 \times 3.70 = 12.9$ vessels per week. As the average vessel in Durban can accommodate 3 500 pallets in the hatches, the maximum break-bulk loading rate is $12.9 \text{ vessels/week} \times 3\,500 \text{ pallets per vessel} = 45\,320 \text{ pallets/week}$. Hence, the average vessel in Durban has space for 30 containers on deck, and so the maximum container loading rate is $12.9 \text{ vessels/week} \times 30 \text{ containers/vessel} \times 20 \text{ pallets/container} = 7\,769 \text{ pallets/week}$. Consequently a total of $45\,320 + 7\,769 = 53\,089$ pallets can be loaded in Durban per week.

SAFT Durban

South African Fruit Terminals guarantee that their Durban facilities can load 2 400 pallets of fruit during every 24-hour day, if there is fruit and/or a vessel to be loaded at their single berth. It thus takes $3\,500/2\,400 =$ one day and 11 hours (35 hours) to load the pallets under deck. On average 30 empty containers have to be offloaded and 30 full containers loaded onto the vessel at a rate of eight containers per crane hour, using two cranes per vessel, resulting in a combined loading time of $60/(8 \times 2) = 3.75$ hours. Assuming it takes six hours to de berth a vessel and berth the next, the total time taken to turn a vessel around is 44.8 hours. Thus, the number of pallets of fruit that can be loaded is $24 \times 7 \text{ hours/week} \div 44.8 \text{ hours/vessel} = 3.75 \text{ vessels/week}$. As the average vessel berthing at SAFT in Durban is the same size as at FPT, approximately $3.75 \times (3\,500 + (30 \times 20)) = 15\,392$ pallets can be loaded per week, of which 13 140 are in break-bulk and the remaining 2 253 in integral containers.

FPT Port Elizabeth

The Fresh Produce Terminals operations in Port Elizabeth guarantee a loading rate of 2 000 pallets per day on each of their two berths. It takes $3\,500 \div 2\,000 = 1.75$ days = one day and 18 hours to load a vessel. A further five hours are added to offload 30 empty containers and load the approximately 30 full containers on board. Berthing and de berthing take approximately five hours, which means it typically takes two days and four hours (52 hours) to turn a vessel around. The maximum loading rate of FPT in Port Elizabeth is thus $24 \times 7 \text{ hours/week} \div 52 \text{ hours/vessel/berth} \times 2 \text{ berths} = 6.46 \text{ vessels/week}$. Thus the maximum break-bulk loading rate is $6.46 \text{ vessels/week} \times 3\,500 \text{ pallets/vessel} = 22\,615 \text{ pallets/week}$, while the maximum container loading rate is $6.46 \text{ vessels/week} \times 20 \text{ pallets/container} \times 30 \text{ containers/vessel} = 3\,877 \text{ pallets/week}$.

FPT Maputo

The Fresh Produce Terminals operations in Maputo guarantee loading 2 000 pallets per day on each of their two berths⁵. The remaining calculations are as for Port Elizabeth, which means that an estimated 22 615 pallets of fruit can be exported in break-bulk and a further 3 877 in containers per week.

3.4 Demand on the Ports

The information in §3.3, together with the weekly fruit export volumes of stone, pome and subtropical fruit (for all regions) allows the *Fruitlog Scenario Model* user to determine how much of the capacity of the infrastructure in the ports is utilised by the export of fruit. As the capacity of the ports is typically measured in pallets or containers, the volumes of fruit are converted from tons to pallets. Once the volume has been converted to pallets, it is divided between fruit exported via the container terminal and conventional terminals. In Cape Town, 52.6% of the fruit is exported via the container terminal. Estimating that 27.2% of the pallets are in 6m porthole or integral containers and the rest in 12m integral containers, means that the maximum utilisation of the port during week seven is 85.5% of the capacity available to it. During the same week, only 34.0% of the capacity of the conventional terminals is required. The container terminal in Durban receives less fruit than the conventional terminals, with only about a quarter of all fruit exported through Durban leaving South Africa by container vessel. This means that the spare capacity is utilised to 83.0% of the full potential of the container terminal (assuming a 20%/80% split between fruit exported in 6m/12m containers), while the conventional terminal is utilised to only 49.4% of its capacity in week 29. In Port Elizabeth (12.2% of pallets in 6m containers), 40.0% of the capacity at the container terminal is utilised during week 29, the peak week. Only 30.8% of the conventional terminal's capacity is utilised during that week, even though 63.9% of the fruit passing through Port Elizabeth is exported via the FPT terminal. It has been assumed that no fruit passes through the container terminal in Maputo (see §3.3.1 for a motivation of this assumption) — hence all fruit passes through the FPT terminal in the port. Week 29 is the peak week in Maputo and during this week 33.2% of the capacity of FPT was required in 2003 to export all the fruit.

Since the expected export in 2010 is known, the same analysis may be performed for the ports during that year, making some additional assumptions. The first assumption is that fruit is still exported via the ports in the same percentages as in 2003 — hence if all pome fruit from the Free State region was exported through Cape Town in 2003, the same is true for 2010. The second assumption is that the exports will exhibit the same weekly profile in 2010 as in 2003. For example, if 10.4% of the hard citrus fruit was exported from the Northern Cape region during week 25 of 2003, the same percentage of the 2010 volume will be exported during week 25 of 2010.

Port expansion plans are typically known long in advance — hence new infrastructure may be incorporated to the assumptions made to construct the *Fruitlog Scenario Model*, in order to determine what would happen in 2010. The number of plug-in points in Cape Town increased from 995 in 2003 to 1 995 by the end of 2004. Other increases include the number of non-fruit

⁵Due to the lack of cold storage in the Maputo area, the port can only load 3 000 pallets per day according to Mr P Franco, General Manager of FPT Maputo [291]. The maximum will be used for modelling purposes as it takes the cooling facilities into account, as well as the warm citrus restrictions. If the objective were to build the cold store restriction into the loading rate, one could have assumed that there were only 1.5 berths to obtain a maximum throughput of 19 869 pallets per week [291].

integral containers from 391 to 550, the percentage of fruit exported through the container terminal from 52.6% to 60%, the containers shifted per crane hour from 17 to 20, the number of cranes from five to seven, the number of berths from three to four and the percentage of pallets in 6m containers from 27.2% to 30%. The capacities of the conventional terminals in Cape Town are assumed to remain constant for the foreseeable future. This results in a maximum capacity increase from 1599 to 3440 pallets per week for fruit export containers. The peak week for fruit export in Cape Town becomes week two when 58.0% of the container terminal capacity and 47.9% of the conventional terminals capacity is utilised. This scenario may be seen in Figure 3.9.

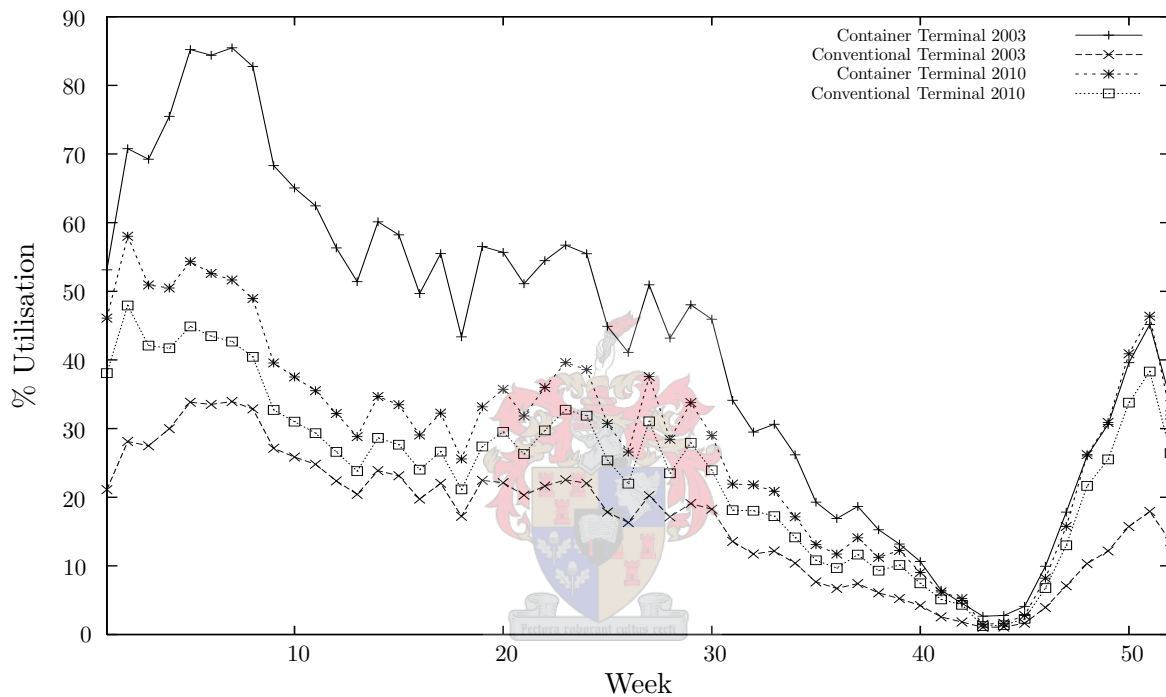


Figure 3.9: The utilisation of the container and conventional terminals (FPT and SAFT) in the port of Cape Town in 2003 and 2010, found using the Fruitlog Scenario Model.

In Durban, an additional 400 plug-in points will be added to the stack by the end of 2005 so as to increase the capacity to 1000 plug-in points. Other changes include an increase from 452 to 636 of plug-in points used by other integral containers (import or non-fruit export) by 2010 (assuming a compound increase of 5% per year), the number of cranes is expected to increase to 18 cranes from 13 by the end of 2005, the number of berths is expected to increase from 7 to 10 by the end of 2007 and the number of non-fruit containers exported is expected to increase from 8393 to 11810 containers per week by 2010 (assuming a compound increase of 5% per year). This means that the capacity of the container terminal for fruit export should increase to 1364 containers per week. Assuming a quarter of the fruit is exported via the container terminal and the capacity of the conventional terminals remain the same, 66.7% of the maximum capacity of the container terminal and 66.4% of the conventional terminals would be utilised to export all the expected fruit (see Figure 3.10) in 2010.

Port Elizabeth sees no increase in plug-in points and the number of import or non-fruit export containers using the plug-in points are expected to increase from 90 to 127 per week by 2010. Assuming only three of the four cranes are available at any one time, the capacity for fruit export

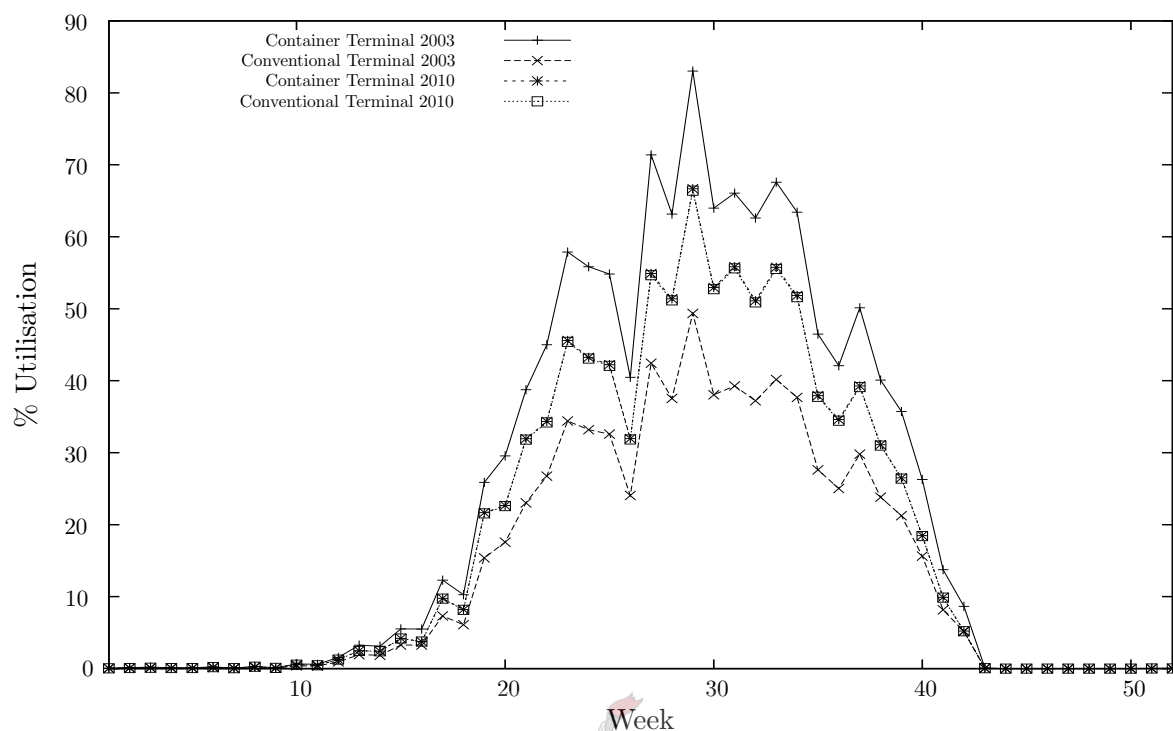


Figure 3.10: The utilisation of the container and conventional terminals (FPT and SAFT) in the port of Durban in 2003 and 2010, as predicted by the Fruitlog Scenario Model.

is therefore expected to decrease from 645 to 608 containers per week by 2010. Assuming the capacity of the FPT terminal remains the same, 59.5% of the capacity of the container terminal and 43.9% of the capacity of FPT for fruit export will be required to export fruit in 2010. See Figure 3.11 for the utilisation of the port on a weekly basis in 2003 and the expected utilisation in 2010.

Assuming the container terminal in Maputo remains unused for fruit and the FPT terminal manages to increase its use of both berths, thus increasing the possible number of pallets to be exported from 19 869 to 26 492, 29.2% of the capacity will be required in week 29 of 2010, on expectation. See Figure 3.12 for a graphical representation of the utilisation of the FPT terminal in 2003 and the expected utilisation in 2010.

3.5 Chapter Summary

The purpose in this chapter was to present the work done to gather fruit export infrastructure data. The chapter also contains a description of the calculations developed to calculate the capacities of ports and the estimated utilisation during 2003.

First, the packhouse and cold store capacities for all fruit regions in South Africa were summarised in two tables (thereby achieving Thesis Objective II, as stated in §1.2) and the form of the questionnaires sent to role-players was also shown in Figures 3.1 and 3.2. Secondly, the fruit production and export volumes for 2003 were determined (thereby achieving Thesis Objective IV, as stated in §1.2), along with an estimation of the expected fruit production and export volumes of 2010. This is followed by a discussion of fruit export volumes and weeks

were identified when peak volumes are exported from South Africa. A brief discussion on the destinations of most of the fruit export volumes follows, and these origin–destination volumes were used to estimate volumes of fruit that were exported via each of the ports during 2003 (as shown in Table 3.11).

The assumptions made for determining the harbour capacities were explained and justified for all of the nine fruit terminals, to assist the reader in understanding how these capacities were found (thereby achieving Thesis Objective III, as stated in §1.2). As the fruit flow volumes from the various production regions could be estimated for the peak weeks, approximate maximum utilisations of the terminals were found (thereby achieving Thesis Objective V, as stated in §1.2). It was found that Cape Town reached up to 85.5% of its capacity available for containerised fruit export during week seven of 2003 and Durban up to 83% of its containerised fruit capacity during week 29 of 2003. In 2010 the utilisation of the container terminal in Cape Town is expected to decrease, while the utilisation of the conventional terminals is expected to marginally increase. The same happens in Durban, where the utilisation of the container and conventional terminals becomes almost identical in 2010. In Port Elizabeth the utilisation of container and the FPT terminals is expected to increase, as the export fruit volumes are expected to increase and there are no planned capacity upgrades due to occur before 2010. In Maputo, the utilisation of the port is expected to decrease by 2010.

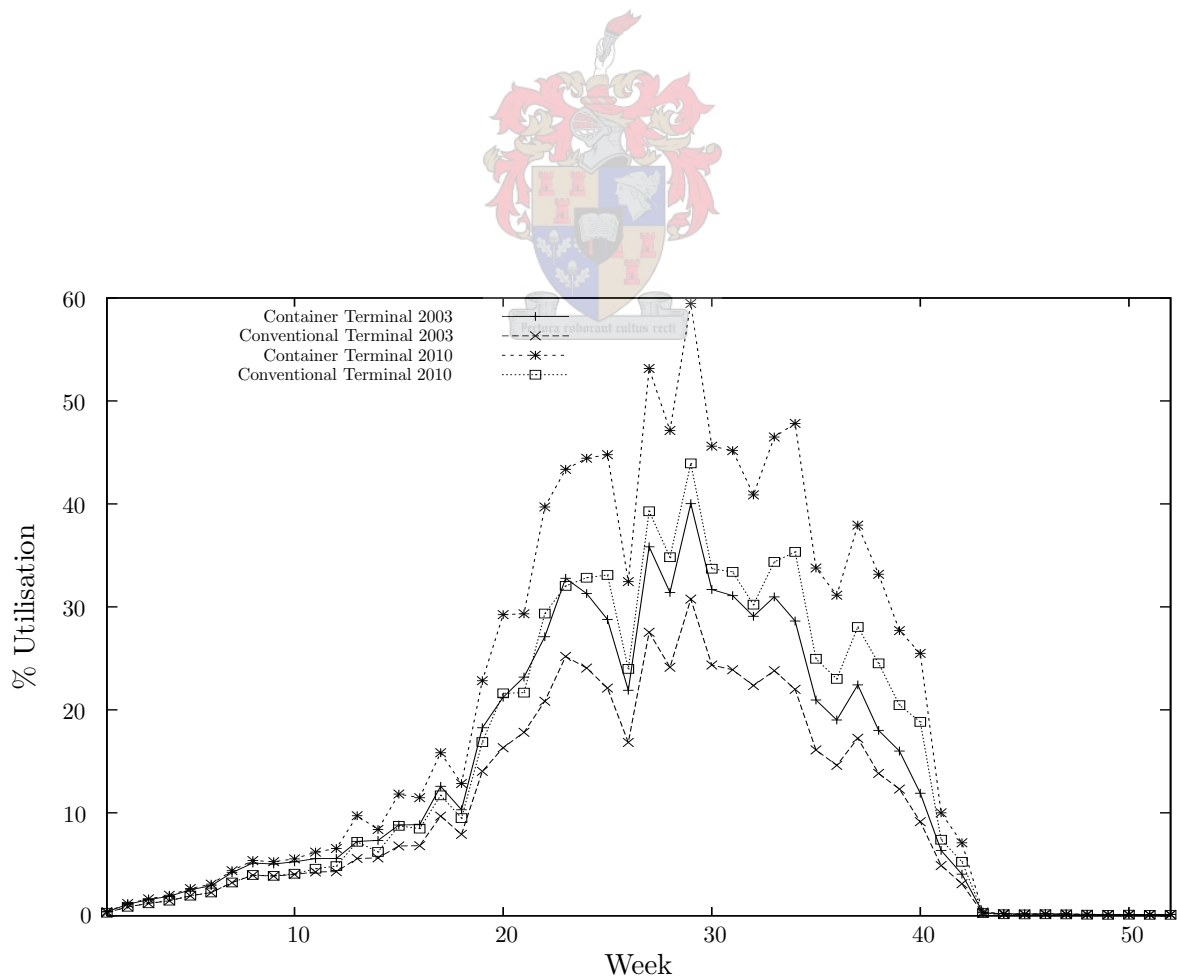


Figure 3.11: The utilisation of the container and conventional terminals (FPT) in the port of Port Elizabeth in 2003 and 2010, as predicted by the Fruitlog Scenario Model.

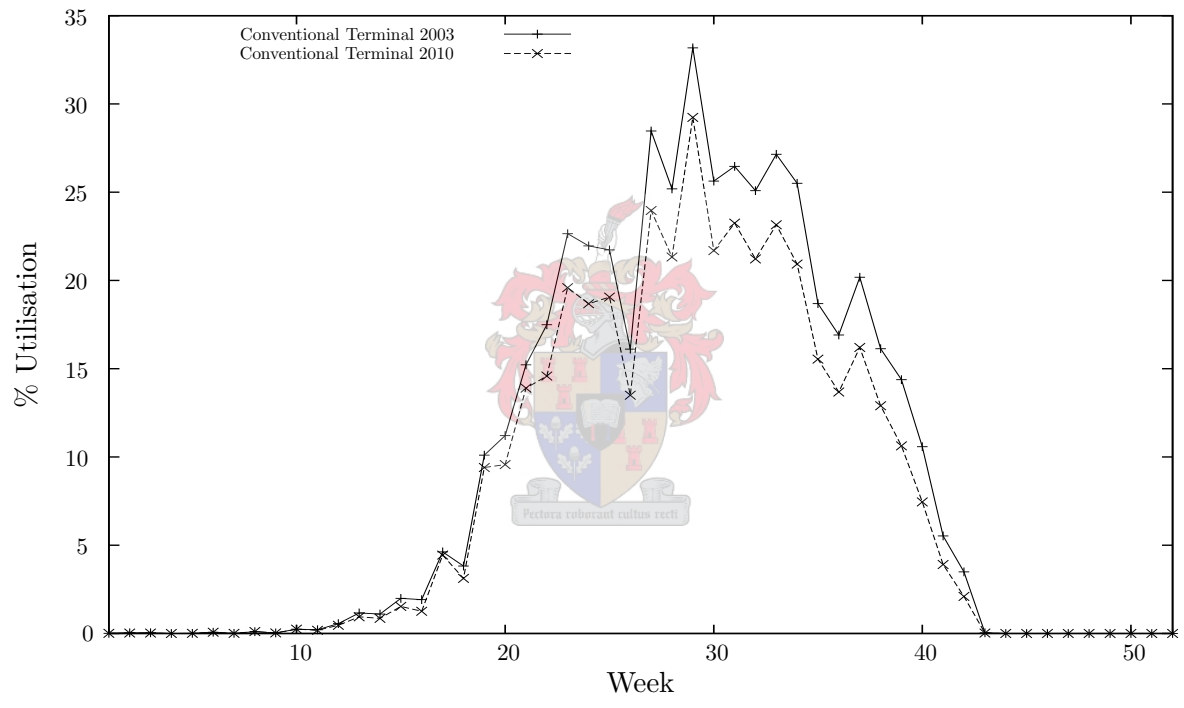
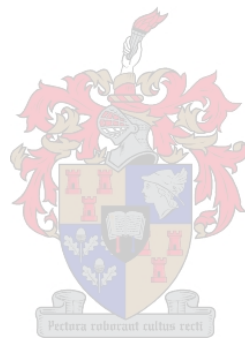


Figure 3.12: The utilisation of the FPT terminal in Maputo harbour in 2003 and 2010, as predicted by the Fruitlog Scenario Model.

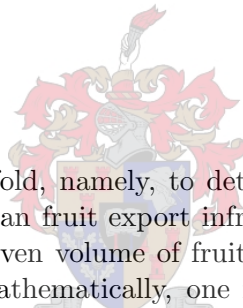


Chapter 4

Graph Theoretic Approach

“Structures are the weapons of the mathematician.”

– Bourbaki [307]



The objective in this chapter is twofold, namely, to determine the maximum volume of fruit able to pass through the South African fruit export infrastructure and to determine the least expensive means of transporting a given volume of fruit through the same infrastructure. To be able to resolve these problems mathematically, one may represent the infrastructure as a directed network of nodes and arcs. Each node may represent a part of the infrastructure or a possible decision point. Examples of infrastructure being represented as nodes include packhouses, cold stores and terminals. Decision points may include decisions between transport by road or rail, by conventional vessel or container vessel, in break-bulk or containerised, via sea freight or air freight, and refrigerated or unrefrigerated, amongst others. Arcs may be used to link the nodes and represent possible modes of transport enabling flows of a commodity (fruit in this case). These arcs may have various attributes associated with them, such as cost per unit of flow, minimum flow requirements, maximum flow restrictions and restrictions on commodity types allowed on the arcs.

In the remainder of this chapter, the formulation of a graph model, such as described above, is to be presented. This is followed by a brief literature survey of the maximum flow and minimum cost flow problems. A brief history of the maximum flow problem is presented for the algorithms used to solve the problem, and a number of applications of the above problems are also outlined. *Ford's algorithm* is presented in detail as the shortest path algorithm required by the maximum flow and minimum cost flow algorithms. This is followed by a detailed description of the *Shortest augmenting path algorithm*, which is used to solve the maximum flow models. Hereafter, a brief history of algorithms and applications of the minimum cost flow problem is given, followed by a detailed discussion on the *Successive shortest path algorithm*, which is used to solve the minimum cost flow models. The results obtained by solving the maximum flow and minimum cost flow models of the South African fresh fruit export infrastructure are discussed at the end of the chapter.

4.1 Formulation of the Graph Model

Due to the size and the complexity of the fruit export industry in South Africa, designing a model that describes the system in great detail is both impossible and unnecessary. Excessive detail also causes compromises in the individuality of various businesses as well as their (confidential) data sets within the industry. When designing a model of a system for an individual company, greater detail means greater accuracy as the company controls each aspect of the model. It may have been possible to delve into greater detail were the South African fruit export industry still regulated. However, when designing a model for a deregulated industry such as the one considered in this thesis, the model must allow for a certain degree of freedom for each of the companies involved, as they are all independent.

A significantly simplified extract of a graph model of the South African fruit export infrastructure is shown in Figure 4.1 — there are, in fact, many more regions than shown in the figure. Due to the distances between the main fruit production regions and the ports, as well as the limited information available, it was decided to group the 850 packhouses and 286 cold stores countrywide (excluding those at the ports) together into 31 packhouse and cold store regions. These production regions are generally the same regions as those specified by producer associations, such as CSA, DFPT, SAAGA, SAMGA and the South African Litchi Growers' Association (SALGA). This choice was made to minimise confusion when interested parties interpret model solutions, and in a bid to keep the fruit production information as accurate as possible. Certain regions only contain a number of farms, packhouses and cold stores in or near a single town (such as the De Doorns region, which contains only the infrastructure in and around the small Western Cape town of De Doorns), while other regions include a large number of towns (such as the Boland region, which includes all infrastructure in and near Somerset West, Stellenbosch, Simondium, Paarl, Wellington, Windmeul, Paardeberg, Groendal and Groot Drakenstein).

Four important assumptions were made during construction of the model. The first assumption is that export fruit takes precedence over fruit sold on the local market. Under this assumption, only export volumes are considered and the remaining fruit (sold locally, juiced or processed in other ways) may be disregarded. Past export volumes may be found with some degree of accuracy, as all fruit are required to pass cold store and port inspections, records of which are kept by the Perishable Products Export Control Board (PPECB) — the national inspection body — and distributed to the various producer organisations. In contrast, volumes of fruit sold or processed locally would be near impossible to determine, as all producers would have their individual methods of profiting from sub-export quality fruit. However, this assumption does not affect the realism of the model significantly, since export fruit would generally take preference over local fruit in situations of congestion [222].

Ignoring the possibility of exporting fruit by air freight is the second important assumption. This assumption forces all fruit to be exported via the ports. The effect of this assumption on the realism of the model is negligible, as air freight is used on very rare occasions. One such occasion is when producers in the Orange River region harvest grapes early in the season due to the particular climatic patterns of the region, before any other producers in the world, and the prices for grapes in Europe are high enough to warrant the significantly higher transportation cost [291].

Although table grapes are generally transported from farms to the farmers' packhouses to be packed, it is thirdly assumed in the model that grapes are, in fact, transported directly from the farms to cold stores. This was done to decrease the number of packhouses for which data had

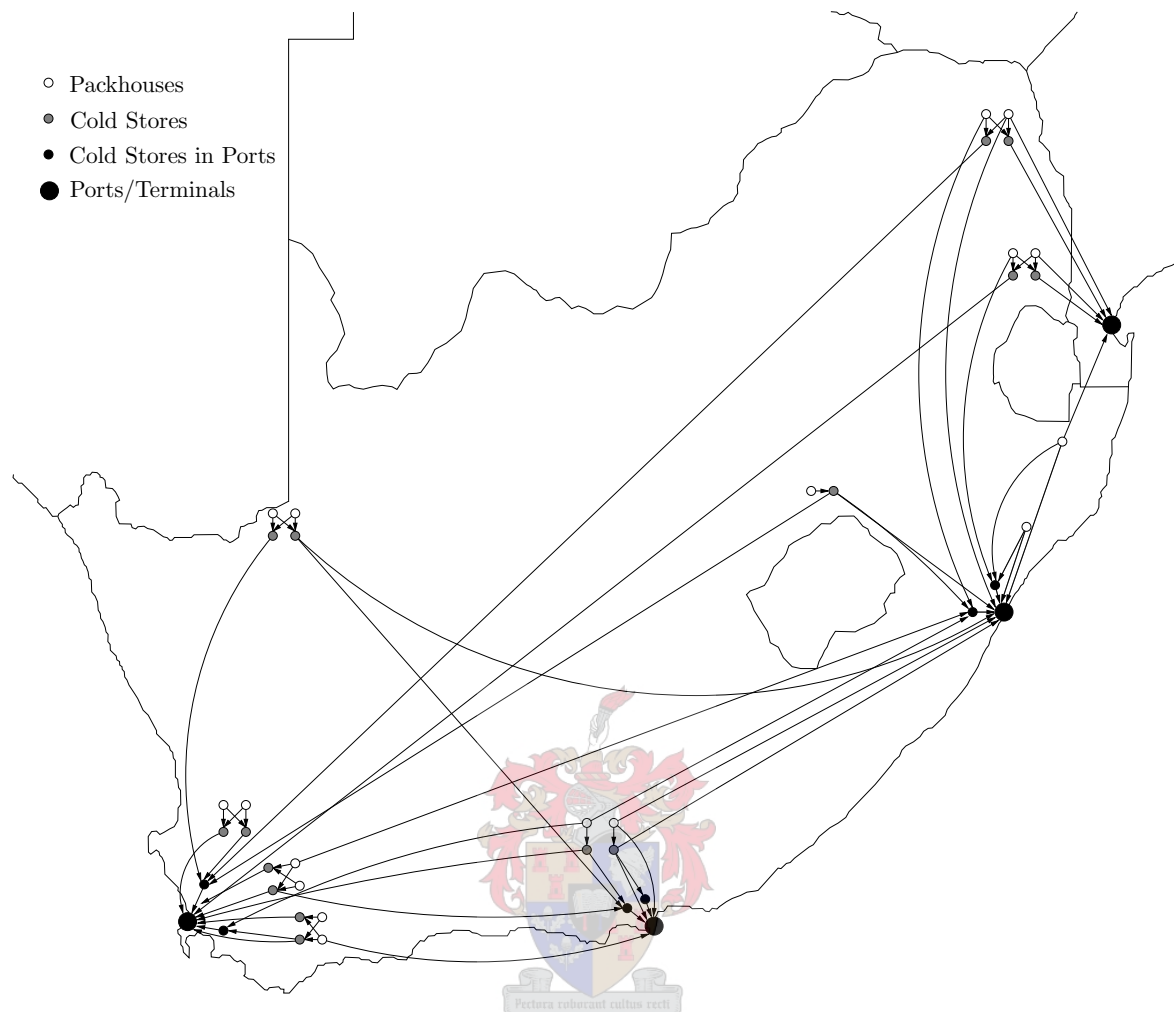


Figure 4.1: A simplified graphical representation of a network representing the fruit export infrastructure of South Africa.

to be gathered, as each table grape farmer generally has a packhouse on his/her own farm (as opposed to the other fruit groupings, for which packhouses are usually operated by independent companies or co-operatives) [191]. This assumption does not have a large impact on the realism of the model, as each farmer will generally ensure that his/her packhouse has enough capacity to pack all fruit harvested from his/her farm. Otherwise, if a farm's packhouse is too small, a farmer may come to an agreement with a nearby farmer (with spare capacity in his/her packhouse) to pack excess fruit.

Another assumption is that one may group together all the packhouses in each region into single super-packhouse nodes where the same groupings of fruits are packed or cooled. This significantly reduces the number of network nodes required to represent the infrastructure, and the assumption would only have a noticeable effect on the realism of the model if one were to know to which cold stores each individual packhouse transported its fruit. However, the gathering of such data would be extremely time-consuming, assuming that the managers of all packhouses would be willing to part with such information. Even though this information is not known, it is still possible to model each packhouse individually, but as all packhouses in a region are adjacent in the infrastructure network to those cold stores in the region that cool

the fruit groups packed by the packhouse, it does not make a difference whether one groups such packhouses and cold stores together, if they pack or cool the same fruit groups. Thus the grouping of infrastructure does not influence the accuracy of the model if it is unknown to which cold stores the individual packhouses send their fruit.

In this chapter it will be assumed that fruit may be considered as a single commodity, and not made of various groups of fruit types with various requirements. Thus, all fruit may pass over the same arcs and the cost will be taken as the highest of those of each of the fruit types, except for the cost of fruit cooling in the winter months, as the citrus volumes dominate the flow through the infrastructure. Therefore the cost of cooling citrus fruit to temperature will be assumed as the standard cost of cooling fruit for those cold stores that do cool citrus.

For the model to correspond to the fruit production data that were available, the source nodes representing regional production are each assigned a supply similar to the production volumes of the regions served by those packing facilities. This forces the exact amount of fruit produced by a region to flow through the packing facilities in that region.

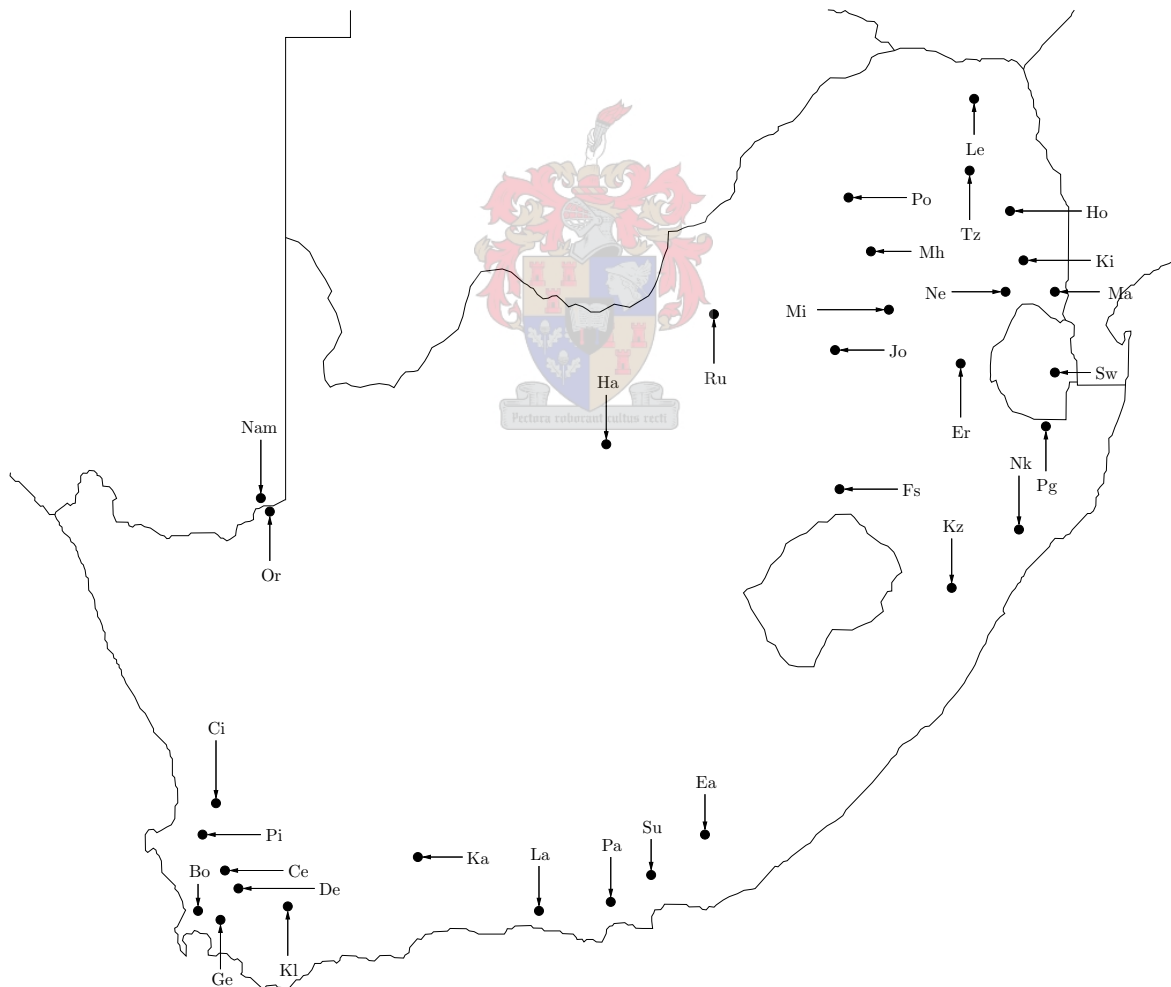


Figure 4.2: The locations of the 31 South African fruit production regions. The full names of the regions abbreviated by two-letter labels in the figure are given in Appendix B. Packhouse and cold store data for Namibia (Nam) and Swaziland (Sw) could not be found and were thus not included in the model, even though fruit is grown in those regions.

The graph model has a form similar to the network shown in Figure 4.1, but includes all 31 fruit production regions (see Figure 4.2). Fruit flow originates at nodes representing the production regions for each fruit group. Arcs join the production regions to nodes representing the intake points of packhouses. These arcs have very high fruit flow capacities associated with them, representing the fact that there is no significant restriction on the volume of fruit that may pass along the routes (typically roads) from production regions to packhouses. The nodes representing packhouses were determined from a list of packhouse and cold store data provided by the PPECB — all packhouses within a region were therefore grouped according to the fruits which had been packed by them in the past. This was also done for cold stores — hence in a single infrastructure region one may find, for example, packhouses that either pack only fruit A, or fruit A and fruit B, or fruit A, fruit B and fruit C, *etc.* There are nodes for each of these packhouse types representing a number of fruit-specific virtual super-packhouse nodes, where the capacities of the packhouses packing the same fruit are added together. The arcs from the packhouse intake point to the output point are used to restrict the flow of fruit through the packhouse (see Figure 4.3(a)), as the algorithms used to solve the model do not allow for capacities on nodes. Thus, the arcs terminating in super-packhouse discharge nodes were assigned maximum flow values equivalent to the capacity of the super-packhouses incident to them in the model.

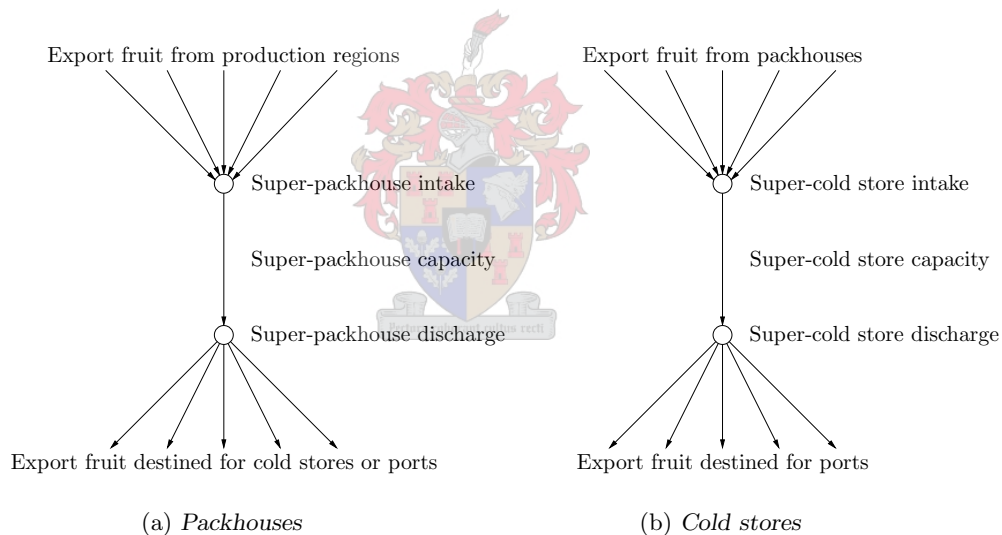


Figure 4.3: A section of a graph illustrating the technique used to limit fruit flow to the capacities of super-packhouses and super-cold stores in the fruit export infrastructure by means of arc flow restrictions instead of node flow restrictions. The same technique was also used for port infrastructure such as cold stores and terminals.

The arcs originating from super-packhouse discharge nodes are adjacent to nodes representing intake points at cold stores that cool at least one of those fruits packed by the corresponding super-packhouses. As hard citrus fruit may be transported to the ports at ambient temperatures (the only fruit type for which refrigeration is not compulsory for travel times longer than two hours), certain arcs originating from nodes representing discharge points of packhouses packing hard citrus (and perhaps other fruits) lead to nodes representing intake points of cold stores in or near ports, or to terminals where break-bulk fruit is loaded onto vessels, via nodes and arcs representing unrefrigerated transport (by road and/or rail). The capacities of cold stores are modelled as for packhouses, shown in Figure 4.3(b).

Finally, arcs link the cold stores to either the Fresh Produce Terminals (FPT), the South African Fruit Terminals (SAFT), the container terminals (Cape Town – CTCT, Durban – DCT and Port Elizabeth – PECT) or to cold stores near the ports, via arcs representing refrigerated fruit transport in break-bulk and/or containers. There are also arcs modelling fruit packed into containers at the cold stores and transported by vehicles with a generator attached (to provide electricity for the cooling of the container), and arcs modelling fruit transported in break-bulk by refrigerated vehicles. Fruit is refrigerated in a cold store until the vessel, on which space is booked for fruit transport, is ready for loading. The fruit is then packed into a container at the cold store, or the fruit may simply be transported directly to the wharf at a terminal, from where it is loaded directly onto a vessel. Fruit in containers must usually be delivered to a stack (or on rare occasions directly to a vessel) at a container terminal, where it is stored until the container must be loaded onto the container vessel on which space has been booked for transport of that specific container. The ports are the most complex part of the graph model — the section of the graph representing the port of Cape Town is shown in Figure 4.4, as an example.

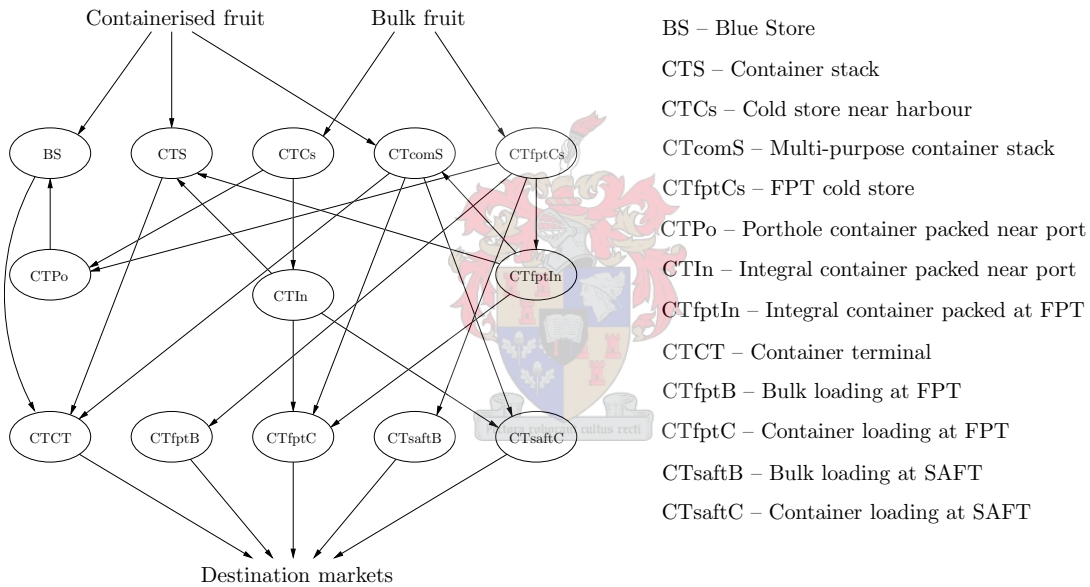


Figure 4.4: Simplification of a part of the graph model representing the South African fruit export infrastructure. This subgraph represents the possible flow of fruit through the infrastructure in the port of Cape Town. Note that the intake and discharge nodes for all the elements of this network are condensed into single nodes.

Model Statistics

Two peak week scenarios were considered when designing the graph model: the week in summer which experienced the largest export (week seven in 2003) and the week in winter 2003 during which the greatest volume of fruit was exported (week 29). The model created for the summer scenario consists of 670 nodes and 2 890 arcs. This includes arcs and nodes for the infrastructure to and in the ports of Durban, Port Elizabeth and Maputo, for which the maximum flow is set to zero to remove the arcs from consideration for the maximum flow if only Cape Town is considered as a port for fruit export. The winter scenario, on the other hand, consists of 607 nodes and 2 883 arcs.

4.2 Concise Survey of Literature

In this section a brief survey of literature regarding the algorithms used to solve maximum flow and minimum cost flow problems in networks is given. Brief histories of the maximum flow and minimum cost flow problems are also presented.

4.2.1 History of the Maximum Flow Problem

In this section the history of the maximum flow problem is traced in three parts. The very beginning of the problem is described first, followed by a description of the history of the development of algorithms for solving the maximum flow problem in the second part. Finally, some traditional applications of the problems are cited in the third part.

The Beginning

According to Schrijver [249], the roots of the maximum flow problem begin with a study of the old Soviet railway system by Americans in the mid 1950s. Harris and Ross [158]¹ proposed the maximum flow problem in 1955 in their classified report *Fundamentals of a Method for Evaluating Rail Net Capacities* (dated 24 October 1955 and unclassified on 21 May 1999) for the US Air Force, with regards to finding the maximum flow (actually the best way of disrupting the supply chain, now known as the min-cut problem) between cities in the Soviet railway system [114, 249]. Initially, a flooding technique was used which pushed as much flow as possible through a network, followed by a reverse push from bottlenecks which forced the flow to obey a conservation of flow law (namely that the net flow from a node must be equal to the supply of that node). However, the solution was not guaranteed to be optimal. After discussions with Harris and Ross in 1954, Ford and Fulkerson [110] suggested using the simplex method that Dantzig [77] had used to solve a transportation problem in 1951. This was not well received by Harris and Ross due to the time consuming nature of the algorithm and the fact that such accuracy was unnecessary for the inaccurate data that were available. In December 1955, Ford and Fulkerson [111] developed the *Flow-augmenting path algorithm*, which did guarantee optimality [6, 249]. Elias, Feinstein and Shannon [99] developed an augmenting path algorithm independently in 1956 which was also capable of solving the problem to optimality.

The Algorithmic History

In 1970, Dinic [87] introduced the concept of layered networks, where shortest paths were found in terms of numbers of arcs and not in terms of edge weights. This eventually led (via many other iterations [9]) to the *Shortest augmenting path algorithm* by Ahuja and Orlin [9] in 1991 in which distance labels (discussed in some detail later in this chapter) were used instead of layered networks. Gabow [125] introduced the *Bit-scaling algorithm* in 1985 and in 1991 Ahuja and Orlin [9] developed the similar, but improved *Capacity scaling algorithm*. This algorithm (and the *Shortest augmenting path algorithm*) was later modified by Ahuja and Orlin [10] in 1995 to be used on a maximum flow problem, where the cost of the maximum flow may not exceed a given budget constraint.

Edmonds and Karp [96] also modified the original *Flow-augmenting path algorithm* in 1972. They presented two versions of their generalisation: one in which the flow was augmented along

¹This reference, and some others, could not be obtained for verification purposes.

shortest augmenting paths and another in which the flow was augmented along paths with the maximum residual capacity. In 1993 Goldfarb and Hao [147] presented two modifications to the *Maximum capacity augmentation algorithm* developed by Edmonds and Karp. It depended heavily on a property used by Tardos [277] in 1985 to develop the first strongly polynomial time algorithm for the minimum cost circulation problem (the problem of finding the minimum cost of a flow in a graph with at least one cycle as well as lower and upper constraints on the arcs and zero net flow out of the nodes). This property was that the non-negativity or capacity bound for at least one additional arc in the network can be relaxed permanently at certain points in the algorithms. In the same paper, Goldfarb and Hao developed a primal simplex algorithm version of their algorithm. They proved that it, in fact, had a worst-case running time not as good as their own *Smallest label network simplex algorithm* (developed in 1990 and implemented by Goldberg, Grigoriadis and Tarjan [140] in 1991) [147]. In 2003 Fujishige [124] presented a maximum flow algorithm based on the algorithms developed by Edmonds and Karp. This algorithm used maximum adjacency ordering and was claimed to run in polynomial time, but was later proven not to be strongly polynomial for directed graphs by Shioura [256].

The preflow-push algorithm was first introduced by Karzanov in 1974 [178]. Malhotra, Kumar and Maheshwari [206] developed a similar algorithm in 1978, and Cherkassky [58] and Galil [126] made further improvements to Karzanov's algorithm in 1977 and 1980, respectively [4]. In 1982 Shiloach and Vishkin [255] presented another preflow-push algorithm, which became the precursor to the *FIFO preflow-push algorithm* developed by Goldberg [138] in 1985 when he used distance labels instead of layered networks. A year later, Goldberg and Tarjan [142] developed the *Highest-label* and *Lowest-label preflow-push algorithms*, and refined the *FIFO preflow-push algorithm* further by using a dynamic tree data structure developed by Sleator and Tarjan [259] in 1983 [4, 6].

In 1989 and 1990 a flurry of activity surrounded the maximum flow problem. In 1989 Cheriyan and Hagerup [56] presented a randomised algorithm to solve the maximum flow problem, for which Alon [13] developed a non-randomised version in 1990. Ahuja and Orlin [8] proposed the first *Excess scaling algorithm* in 1989 and this was followed up by the *Stack-scaling algorithm* and the *Wave-scaling algorithm* in the same year, which they developed with Tarjan [12]. In 1990, Cheriyan, Hagerup and Melhorn [57] presented the *Incremental generic algorithm*, the *Incremental excess scaling algorithm* for integer networks and the *Incremental strongly polynomial algorithm* in an attempt to reduce the worst-case bound on the running time.

A very detailed report on the computational efficiency of a collection of ten of the above algorithms was presented by Ahuja, Kodialam, Mishra and Orlin [4] in 1997. In it, the authors study *Dinic's algorithm* (1970), Karzanov's *Preflow-push algorithm* (1974), Ahuja and Orlin's *Shortest augmenting path algorithm* (1991), Gabow's, and Ahuja and Orlin's *Capacity scaling algorithm* (1985 and 1991 respectively), Goldberg and Tarjan's *Highest-label preflow-push algorithm* (1986), the *FIFO preflow-push algorithm* (1986) and the *Lowest-label algorithm* (1986), Ahuja and Orlin's *Excess-scaling algorithm* (1989), Ahuja, Orlin and Tarjan's *Stack-scaling algorithm* (1989) and the *Wave-scaling algorithm* (1989). They performed tests on five classes of networks, and concluded that the *Highest-label preflow-push algorithm* was the most efficient in practice.

More general maximum flow algorithms have also appeared since the 1960s. These algorithms allow some flow to "disappear" along an arc, so that only a fraction of the flow leaving node a arrives at node b when the flow travels along the arc (a, b) . Wayne [300] studied these algorithms in detail in his PhD dissertation completed in 1999, but since they do not apply for this thesis, these algorithms will not be considered further.

Certain other non-standard maximum flow algorithms have also been developed. In 1981 Chanas and Kołodziejczyk [53] considered maximum flows in networks with arc capacities that were not strict. In 1994 Buckley and Hayashi [42] applied genetic algorithms to the fuzzy maximum flow problem. More work on fuzzy maximum flow problems was done by Chanas, Delgado, Verdegay and Vila [52] in 1995 and in 2001 by Diamond [91]. More recent work on maximum flow algorithms has included work on the maximum flow problem for networks whose properties change over time [46, 223] (first studied by Ford and Fulkerson in 1958 [113]). However, none of these algorithms are applicable in the context of this thesis and will not be discussed further.

Traditional Applications of the Maximum Flow Problem

The book by Ahuja, *et al.* [6] and the chapter Ahuja, *et al.* wrote in *Handbooks in Operations Research and Management Science, Volume 7: Network Models* [7] give many examples of applications of the maximum flow problem. Table 4.1 summarises a collection of the applications listed in these two texts.

Authors	Year	Application
Dantzig & Fulkerson	1954	Tanker scheduling [79]
Hall	1956	Determining best political representation [156]
Ford & Fulkerson	1958	Maximum dynamic flows [113]
Berge & Ghouila-Houri	1962	Feasible flow problem [30]
Bacharach	1966	Matrix rounding [23]
Schwartz	1966	Baseball elimination problem [250]
Johnson	1968	Open pit mining [171]
Rhys	1970	Selecting freight handling terminals [240]
Van Slyke & Frank	1972	Network reliability testing [295]
Stone	1977	Distributed computing on a two-processor model [275]
Picard & Ratliff	1978	Multi-facility rectilinear distance location problem [236]
Khan	1979	Optimal deployment of police patrols [173]
Chamlet, Francis & Saunders	1982	Modelling building evacuation [51]
Mamer & Smith	1982	Optimal repair kit (flyaway kit problem) [208]
Martel	1982	Pre-emptive scheduling on machines with different speeds [211]
Ahuja	1986	Minimax transportation problem [3]
Federgruen & Groeneveldt	1986	Scheduling jobs on parallel machines [107]
Lin	1986	Solving a system of equations [202]
Khan & Lewis	1987	Nurse staff scheduling [174]
Orlin	1987	Optimal destruction of military targets [230]
Kelly, Golden & Assad	1990	Statistical security of data [185]
Wayne	2001	Baseball elimination problem [301]
Arata, Iwata, Makino & Fujishige	2002	Locating sources to meet flow demands in undirected networks [15]
Hochbaum & Queyranne	2003	Minimizing a convex cost closure set [161]
Rieger	2003	Roughening transition in a model for elastic manifolds [241]
Shai	2005	Plastic analysis of trusses [254]

Table 4.1: Applications of the maximum flow problem. Many applications were found in either [6] or [7], or both.

4.2.2 Solving a Maximum Flow Problem

To find the maximum possible flow through a network, the network must first be modified. All nodes with positive supply values (sources of flow) must be joined by means of arcs from a

single node, called the *source*, and all flow destinations with negative supply values should be joined by means of arcs to a single node, called the *sink*. Thus, all flow may be considered to originate from the single virtual source and drain at the single virtual sink. There are two main categories of algorithms that exist to solve the maximum flow problem [6], namely

- those that maintain a feasible solution by adhering to a conservation of flow law (except at the source and sink), by incrementally adjusting the flow along paths that start at the source and end at the sink, until optimality is achieved (known as *augmenting path algorithms*).
- those that push more flow through the network than is allowed, so that some flow is either sent back to the source or forward to the sink, until optimality is achieved (known as *preflow-push algorithms*).

The standard algorithms mentioned in §4.2.1 each fall into one of the categories mentioned above and may be classified further into one of three sub-categories, depending on the mechanism of the algorithm. The first is the sub-category of algorithms based on the *Labelling algorithm* (which has a worst-case running time $O(nmU)$, where n is the number of nodes in the network, m is the number of arcs and U is the maximum value of the flow upper bounds on the arcs). The fastest of these algorithms is the *Shortest augmenting path algorithm* which has a running time $O(n^2m)$. The second sub-category of algorithms comprise those algorithms based on the mechanism of the *Generic preflow-push algorithm* (worst-case running time is $O(n^2m)$). This sub-category includes the fastest algorithm tested by Ahuja, *et al.* in 1997 [4], namely the *Highest-label preflow-push algorithm*, with a worst-case running time $O(n^2\sqrt{m})$. In 1989, Ahuja and Orlin developed the *Excess scaling algorithm* (worst-case running time $O(nm + n^2 \log U)$), which is an evolution of the preflow-push algorithms and which forms the basis for the third sub-category of algorithms. Ahuja, *et al.* developed two more algorithms based on the *Excess scaling algorithm* in 1989, the fastest being the *Stack scaling excess flow algorithm* with a worst-case running time $O(nm + n^2\sqrt{\log U})$ [4].

The algorithm chosen to solve the maximum flow problem in this thesis is the *Shortest augmenting path algorithm*, due to its efficiency and the relative ease of implementation [6]. The algorithm was implemented in *FORTRAN 95* by the author to arrive at the numerical results presented at the end of this chapter. Although the *Highest-label preflow-push algorithm* is faster [4], the class of preflow-push algorithms are more difficult to implement and heuristics are required to make them efficient [6].

The *Shortest augmenting path algorithm* requires a shortest path algorithm in its implementation, as does the minimum cost flow algorithm discussed later in this chapter. In order to be as robust as possible, the shortest path algorithm must accept negative arc lengths and networks with cycles, be able to detect negative cycles and find a shortest path from a single chosen node to all other nodes in the network. The first three restrictions imply that all label-setting algorithms, such as *Dijkstra's algorithm* [6, 105] (and its variants — *Dial's implementation*, the *d-Heap implementation*, the *Fibonacci heap implementation* and the *Radix heap implementation* [6]) are unsuitable for this purpose. The *all-pairs label-correcting algorithm* [6], the *Floyd-Warshall Algorithm* [6, 105], the *Dantzig shortest-path algorithm* [105] and the *k-shortest path algorithm* [105] are also unsuitable without modifications, as they find the shortest distances from *each* node in a network to *all other* nodes in the network. The predecessor to *Dijkstra's algorithm* (called *Ford's algorithm* [105]), was found to be the most appropriate algorithm for the application at hand.

4.2.3 Ford's Algorithm

The algorithm described in this section was presented by Ford [109] in 1956 and was the first in the class of label-setting algorithms (so called, because labels of nodes are made permanent at each iteration). To find a shortest path from a node s to node t in a network, a series of label changes are made on nodes which constitute paths from s to t . The algorithm begins by adding the node s to the set \mathcal{S} (which is empty at initialisation) of permanently labelled nodes and by then finding the distances from it to all nodes adjacent from it. The node with the smallest distance label is permanently labelled, and the shortest distances from all permanently labelled nodes to the nodes adjacent from them are found. This process continues until either node t has been labelled, or all nodes have been labelled permanently (if t is not labelled permanently, then there is no path from s to t).

To describe this process in more detail, the steps of solving a shortest distance problem are presented (adapted from [105]):

Step 1 Begin with all arcs and nodes unlabelled and the sets of permanently labelled nodes \mathcal{S} and the set of temporarily labelled nodes $\bar{\mathcal{S}}$ empty. Each vertex j must be assigned two values: $d(j)$ and $p(j)$, where $d(j)$ is the shortest distance from s to j , and $p(j)$ is the predecessor of j along which a shortest path may be traced back to s along permanently labelled nodes (nodes in \mathcal{S}). Initially, let $d(s) = 0$ and $d(j) = \infty$ for all $j \in \mathcal{N}$, where \mathcal{N} is the set of all nodes in graph G . Let i be the last vertex that was labelled, set $i = s$ and set $\mathcal{S} = \mathcal{S} \cup s$.

Step 2 Adjust the labels of all nodes adjacent from i so that $d(j) = \min\{d(j), d(i) + c_{ij}\}$, where c_{ij} is the distance from node i to node j . Adjust $p(j)$ accordingly and set $\bar{\mathcal{S}} = \bar{\mathcal{S}} \cup j$ if j is not in the set of temporarily labelled nodes already. Stop if $d(j) = \infty$ for all unlabelled nodes as this would mean that there is no path from s to t . Otherwise, if the node with the smallest $d(j)$ value is unlabelled, add it to the set of permanently labelled nodes \mathcal{S} , remove it from the set of temporarily labelled nodes $\bar{\mathcal{S}}$ and set $i = j$. If there are no temporarily labelled nodes and no distance labels of labelled nodes j may be lowered by checking whether $d(j) > d(i) + c_{ij}$, terminate the algorithm.

Step 3 Terminate the algorithm when there are no more temporarily labelled nodes. The shortest distance from node s to all other nodes may be determined by following the trail from any node back to s , using the predecessor information contained in the $d(j)$ values. If all nodes (to which there exists a path from s) have not been labelled permanently, repeat step 2.

The above steps represent the essence of *Ford's algorithm* (which is given in pseudocode form as Algorithm 1). The reader is referred to Appendix E.1 for an example illustrating the working of *Ford's algorithm*.

4.2.4 The Shortest Augmenting Path Algorithm

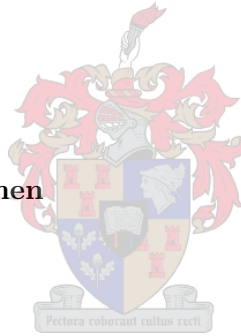
Before the *Shortest augmenting path algorithm* may be implemented, two notions have to be introduced. The first is the notion of a *residual network*, which appears in algorithmic solutions to many network problems, and refers to an alternate network representation. Instead of an algorithm keeping track of the flow and maximum flow constraint of an arc (i, j) in a network, the residual capacity r_{ij} is used in this algorithm and in the *Shortest augmenting path algorithm*.

Algorithm 1 Ford's algorithm**Input:** A graph with edge weights, the node from where the shortest path should be found**Output:** The shortest path from the specified node to all other nodes

```

1:  $\mathcal{S} = \emptyset$ 
2:  $\bar{\mathcal{S}} = \emptyset$ 
3:  $d = \infty$ 
4:  $d(s) = 0$ 
5:  $p(s) = 0$ 
6: LabelCount = 0
7: while  $|\mathcal{S}| \leq n$  do
8:   let  $i \in \bar{\mathcal{S}}$  such that  $d(i) = \min\{d(j) : j \in \bar{\mathcal{S}}\}$ 
9:    $\mathcal{S} = \mathcal{S} \cup \{i\}$ 
10:   $\bar{\mathcal{S}} = \bar{\mathcal{S}} - \{i\}$ 
11:  LabelCount( $i$ ) = LabelCount( $i$ ) + 1
12:  if LabelCount( $i$ ) =  $n$  then
13:    stop [negative cycle has been found]
14:  end if
15:  if  $(i, j) \in \mathcal{A}(G)$  then
16:    if  $d(j) > d(i) + c_{ij}$  and  $(i \in \mathcal{S} \text{ or } i \in \bar{\mathcal{S}})$  then
17:       $d(j) = d(i) + c_{ij}$ 
18:       $p(j) = i$ 
19:      if  $j \in \mathcal{S}$  then
20:         $\mathcal{S} = \mathcal{S} - \{j\}$ 
21:         $\bar{\mathcal{S}} = \bar{\mathcal{S}} \cup \{j\}$ 
22:      else if  $j \notin \bar{\mathcal{S}}$  and  $j \notin \mathcal{S}$  then
23:         $\bar{\mathcal{S}} = \bar{\mathcal{S}} \cup \{j\}$ 
24:      end if
25:    end if
26:  end if
27:  if  $\bar{\mathcal{S}} = \emptyset$  then
28:    stop [no temporary nodes to set permanent]
29:  end if
30: end while

```



The residual capacity represents the remaining flow that may still pass through a network over and above the flow that has already been allocated to an arc, and is defined to be $r_{ij} = u_{ij} - x_{ij}$ (see the conversion from the normal network to the residual network in Figure 4.5), where u_{ij} is the capacity of the arc (i, j) and x_{ij} is the flow across that arc. It is important to note that one may also send x_{ij} units of flow in the direction (j, i) , which would have the effect of cancelling the flow across (i, j) . If costs are involved (as they are for the minimum cost problem), increasing the flow across (i, j) by a unit increases the cost by c_{ij} units, while increasing the flow along the direction (j, i) decreases the cost by c_{ij} .

Some networks contain parallel arcs (where two or more (i, j) arcs exist with different costs and capacities associated with them), or cases where both arcs (i, j) and (j, i) are present. This may be countered by adding nodes into one (or more) of the existing arcs. Algorithms may then be applied to the revised network to find solutions to maximum flow or minimum cost flow problems (amongst others), as illustrated in Figure 4.6.

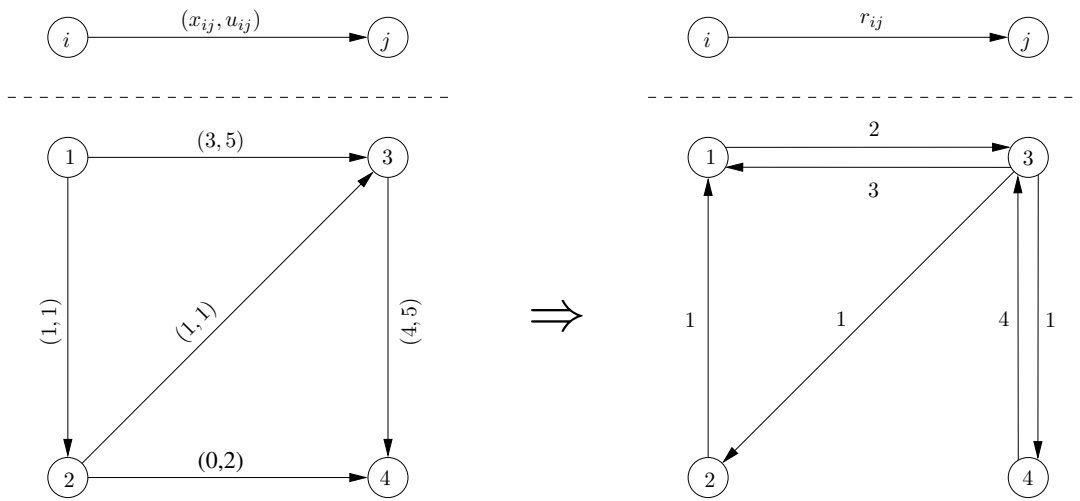


Figure 4.5: An illustration of the conversion from an original network (left) to a residual network (right). To find a residual network from another bounded network, simply subtract the flow through all arcs from the flow upper bound on the arcs to find the arc's residual capacities. The residual capacity of the backward arcs (such as arc (j, i)) is equal to the flow across the arc (i, j) . As the flow across (i, j) increases, the residual capacities of (i, j) and (j, i) decrease and increase respectively. Note that arcs with a zero residual value have been omitted from the residual network.

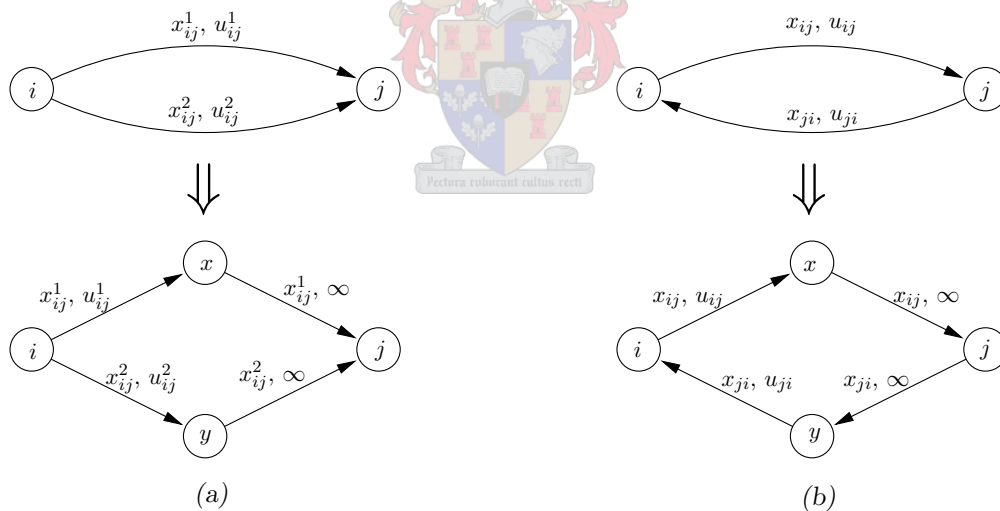


Figure 4.6: (a) Multigraphs may be converted to graphs with no parallel arcs by adding a node into each arc and transferring the property of the original arc to at least one of the two new arcs in its place. (b) Both the arcs (i, j) and (j, i) can be present in a graph if an additional node is added into each arc and the properties of the original arc transferred to at least one of the two new arcs.

It is also possible to construct a residual network for a network containing arcs with lower bounds. Then the flow through an arbitrary arc from node i to node j must satisfy the property $l_{ij} \leq x_{ij} \leq u_{ij}$. In maximum flow problems with no lower bounds, a feasible flow always exists, as no flow through the network is still a feasible flow, but if there are lower bounds present, no feasible flow may exist. Such a case is illustrated in Figure 4.7. The residual flow has a new form

when lower bounds appear in the network; it becomes $r_{ij} = (u_{ij} - x_{ij}) + (x_{ji} - l_{ji})$ where the first term indicates the maximum amount by which the flow may increase across the arc (i, j) and the second term is the maximum flow possible across the arc due to the decrease in flow across the arc (j, i) . To find the real flow through the arcs, the identity $x_{ij} = l_{ij} + \max\{u_{ij} - r_{ij} - l_{ij}, 0\}$ may be used. This is only true if all values are non-negative [6]. A network could potentially contain an infeasibility, as shown in Figure 4.7 — therefore algorithms capable of finding an initial feasible flow have to be used to determine whether a feasible flow exists. Other algorithms may then attempt to improve on such an initial feasible flow (typically in iterated fashion). However, lower bounds were not required for the maximum flow problem, thus the *Shortest augmenting path algorithm* may be used to find the maximum flow through the South African fruit export infrastructure.

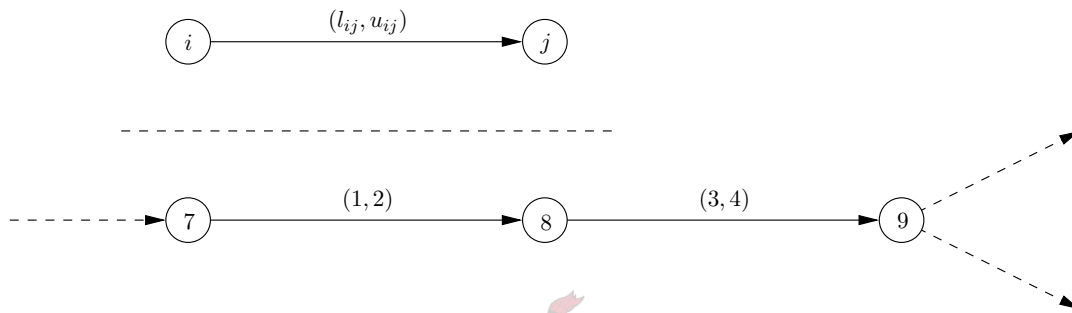


Figure 4.7: A section of a network showing that it is possible that no feasible flow exists.

The second important prerequisite for the *Shortest augmenting path algorithm* (and the preflow-push algorithms) is that of the *distance label*. This differs from the distance label used by *Ford's algorithm*, because it does not measure the distance from the source node using the weights of the arcs. Instead, the distance label $d(i)$ is valid with respect to the flow in the network if it satisfies two conditions, $d(t) = 0$, and $d(i) \leq d(j) + 1$, for all arcs (i, j) in the residual network. There are three important properties of the distance label. The first property is that if all the distance labels are valid, $d(i)$ represents a lower bound on the length of a shortest path from node i to node t in the residual network [9]. This may be shown to hold if one takes the validity conditions into account when considering a path $i = i_1 - i_2 - \dots - i_k - i_{k+1} = t$, which has length k . Using the two validity conditions repeatedly, the system of inequalities

$$\begin{aligned} d(i_k) &\leq d(i_{k+1}) + 1 = d(t) + 1 = 1 \\ d(i_{k-1}) &\leq d(i_k) + 1 \leq 2 \\ d(i_{k-2}) &\leq d(i_{k-1}) + 1 \leq 3 \\ &\vdots \\ d(i_2) &\leq d(i_3) + 1 \leq k - 1 \\ d(i) = d(i_1) &\leq d(i_2) + 1 \leq k \end{aligned}$$

is found. Since the distance label from node t to node i is at most k along that path, the distance labels cannot be larger than the length of a shortest path (if one exists), as all distance labels are assumed to be valid.

The second property states that if $d(s) \geq n$ (the number of nodes in the network), then the residual network does not contain a valid path from the source to the sink node [9]. This follows from the first property of distance labels and the property of graphs that the length of a longest path cannot exceed $n - 1$ without passing through the same node more than once.

Two further notations occur in distance labels. A distance label $d(i)$ is called *exact*, if it equals the shortest distance from node i to node t . Although $d(i) = 0$ is a valid distance label for all nodes, it is not exact, as the shortest distance from node i to node t cannot be zero, unless $i = t$. Arcs of the form (i, j) in residual networks are called *admissible* if $d(i) = d(j) + 1$ and $r_{ij} > 0$, and *inadmissible* if $d(i) \neq d(j) + 1$ or $r_{ij} = 0$. An admissible path is a directed path of admissible arcs from node s to node t .

This is the basis of the third property of distance nodes, which states that an admissible path from node s to node t is a shortest augmenting path from the source s to the sink t [9]. For all arcs to be admissible, two conditions have to be met, namely $r_{ij} > 0$ and $d(i) = d(j) + 1$ for all arcs (i, j) on the path. The first condition implies that the path is an augmenting path — hence more flow may be assigned to that path. The second condition implies that if the path is of length k , then the distance label of the source is $d(s) = k$. Due to the first property of distance labels, $d(s)$ is a lower bound on the length of the paths from s to t ; therefore the path of length k must be a shortest augmenting path.

The *Shortest augmenting path algorithm* uses these properties to find the maximum possible flow through a network, by augmenting the flow through the residual network along admissible paths iteratively. All nodes are assigned distance labels from t , using a shortest path or backward breadth-first search algorithm. Then paths are created from s to t along admissible arcs (where $d(i) = d(j) + 1$ for all arcs (i, j)) and the maximum permissible volume of flow is augmented for each path found, until no further paths exist. The path building begins at node s and follows an admissible arc ($r_{sj} > 0$) to an admissible node ($d(j) = d(s) - 1$). This process continues iteratively until either node t is encountered or a node with no admissible arcs becomes the end of the path. In the first case augmentation takes place, while the distance label of the arc is increased on the node with no admissible arcs, and the predecessor of that node becomes the end of the path.

To describe this process in more detail, the various steps of solving the maximum flow problem are presented. Pseudocode for the *Shortest augmenting path algorithm* is given in Algorithm 2.

Step 1 Begin with all nodes unlabelled. Each vertex must be assigned two values: $d(i)$ and $p(i)$, where $d(i)$ is the shortest distance from the sink to node i (assuming that the directions of the arcs are reversed for this step) and $p(i)$ is the predecessor of node i along the shortest path when the shortest augmenting path is searched for in later steps. Initially, $d(t) = 0$ and the shortest distances from the sink to all other nodes are found. Let i be the last node in the path from source to sink, set $i = s$.

Step 2 Find an admissible arc (i, j) where $r_{ij} > 0$ and $d(i) = d(j) + 1$, set $i = j$ and set the predecessor of j to be i . If there is no admissible arc adjacent from j , set $d(j) = d(j) + 1$ and $i = p(j)$ (not if $j = s$), where $p(j)$ is the predecessor of j on the path from s to j , and repeat this step if $d(s) < n$.

Step 3 If $d(s) \geq n$ terminate the algorithm, the maximum flow has been found. Otherwise, if $j \neq t$ then repeat step 2. If $j = t$, then find $\delta = \min\{r_{ij} : (i, j) \in P\}$ where P is the path found from the source to the sink nodes. Increase the flow through P by δ .

To prove that the *Shortest augmenting path algorithm* finds the maximum flow in a graph, the following lemma is required.

Algorithm 2 Shortest Augmenting Path Algorithm**Input:** A network with supply information for each node and upper bounds on the edges**Output:** The maximum flow possible through the network

```

1: let  $(j, i) \in \text{ReverseMatrix} \forall (i, j) \in A(G)$ 
2: determine the exact distance labels using Ford's algorithm and ReverseMatrix
3:  $r_{ij} = u_{ij} \forall (i, j) \in A(G)$ 
4:  $i = s$ 
5: while  $d(s) \leq n$  do
6:   if  $r_{ij} > 0$  and  $d(i) = d(j) + 1$  then
7:      $p(j) = i$ 
8:      $i = j$ 
9:     if  $j = s$  then
10:      use  $p(i)$  to determine the path  $P$  from  $s$  to  $t$ 
11:      let  $\delta = \min\{r_{ij} : (i, j) \in P\}$ 
12:      augment  $\delta$  units of flow along path  $P$ 
13:     end if
14:   else
15:      $d(i) = \min\{d(j) + 1 : (i, j) \in A(G) \text{ and } r_{ij} > 0\}$ 
16:     if  $i \neq s$  then
17:        $i = p(i)$ 
18:     end if
19:   end if
20: end while

```

Lemma 4.1 *Valid distance labels are maintained at each step of the Shortest augmenting path algorithm, and each relabel operation strictly increases the distance label of a node.*

Proof. The proof is by induction on the number of flow augmentation and node relabelling operations. The algorithm, by design, assigns valid distance labels to each of the nodes in the network. Now, assume that the distance labels remain valid after an arbitrary number of iterations of the algorithm, i.e. $d(k) \leq d(i) + 1$ for all arcs (k, i) in the network. It must now be shown that the labels remain valid after the next flow augmenting operation and node relabelling operation.

If an admissible arc (i, j) is not on the augmenting path from source to sink, no flow will be augmented along the arc — thus the augmentation cannot influence the validity of the nodes adjacent to the arc. This may only be performed by the relabelling operation. If the arc is on the augmenting path and the augmented flow is less than the residual value of the arc, the arc will remain admissible. If the augmenting removes the arc from the residual graph (the maximum flow is achieved along that arc), the distance labels for this arc remain valid. An augmentation will create or increase the residual capacity of the arc (j, i) , ensuring that the condition that $d(j) \leq d(i) + 1$ for the distance labels remains valid. This inequality does hold, as the admissibility conditions of an arc require that $d(i) = d(j) + 1$, which, when rearranged, gives $d(j) = d(i) - 1 < d(i) + 1$. Thus, the distance labels remain valid.

As the relabelling operation changes the distance label $d(i)$ when there are no admissible arcs adjacent from it, it must be shown that the new distance labels (say $d'(i)$) satisfy the validity conditions. A relabelling operation is performed only when there exists no arc (i, j) adjacent from node i that satisfy the validity conditions $d(i) = d(j) + 1$ and $r_{ij} > 0$. This fact, along with the validity condition $d(i) \leq d(j) + 1$, implies that $d(i) < d(j) + 1$ for all arcs (i, j) with

$r_{ij} > 0$. This means that $d(i)$ is strictly less than the minimum distance label $d(i) + 1 = d'(i)$, which will be the new distance label after relabelling is complete. Thus, the validity conditions remain intact for all arcs adjacent from node i with $r_{ij} > 0$, and the distance labels may only increase during relabelling operations. The induction hypothesis states that $d(k) \leq d(i) + 1$ for all arcs (k, i) with $r_{ki} > 0$, thus the relabelling operation leaves the validity of arc (k, i) intact, as $d(i) < d'(i)$, so $d(k) < d'(i) + 1$ [6, 143]. ■

It may now be shown that the *Shortest augmenting path* algorithm does, in fact, find the maximum flow of a network.

Theorem 4.2 *The Shortest augmenting path algorithm finds the maximum possible flow through a network.*

Proof. The algorithm terminates once the distance label of the source node $d(s) \geq n$, where n is the number of nodes in the graph. This implies that there is no longer a path of admissible arcs in the graph, so no further augmentation is possible. Thus, the maximum flow has been found [9]. ■

For an example of the working of the *Shortest augmenting path algorithm*, the reader is referred to Appendix E.2.

4.2.5 History of the Minimum Cost Flow Problem

In this section a brief history of the minimum cost flow problem is presented in three parts. The very beginning of the problem will be discussed in the first subsection, followed by the history of the algorithms for solving the maximum flow problem. Finally, traditional applications of the problems will be listed.

The Beginning

The minimum cost flow problem appears in the literature earlier than does the maximum flow problem. In his 2002 paper *On the history of the transportation and maximum flow problems*, Schrijver [249] reports that Russian mathematician Tolstoï [281] published an article in 1930 called *Methods of finding the minimal total kilometrage in cargo-transportation planning in space* in a book on transportation planning, which the National Commissariat of Transportation of the Soviet Union had issued. In the article, Tolstoï studied a problem of transporting goods (including salt and cement) from factories (or other sources) to predetermined destinations along the railway network in the Soviet Union. He began by considering only two sources and noted that one could order the destinations according to the difference between the distances to each of the sources. One source would then satisfy the destinations with cargo until the source is depleted, which is when the remaining source would satisfy the demand of the other destinations. Tolstoï noted further that the list was independent of supply and demand and could thus be used until a source or destination was removed from the list, as long as supply and demand information was available. He next studied the transportation case where all supply and demand points lie on a circular railway line. An optimal solution was found by considering the difference between two sums of cost.

The Algorithmic History

Tolstoï combined the ideas from the two examples mentioned above to derive a heuristic capable of solving a problem concerning the transport of concrete along the railway network in the Soviet union at the time. The problem was large for the time, with 10 sources, 68 destinations and 155 links between sources and destinations. His heuristic takes into account the geography of the Soviet Union by listing, for each source (starting at the most remote), the destinations for which that source is closest, second closest, *etc.* He assigned flow from each source to that list until the supply of the source was depleted. He did deviate from this rule when he foresaw a negative cycle in the residual graph, indicating that he may have assumed the negative cycle optimality conditions (see Theorem 4.4 later in this chapter), without explicitly stating them. It has been shown, with the help of modern solvers, that Tolstoï's solution was, in fact, optimal [249].

In their book on flows in networks, Ford and Fulkerson [114] mention a survey conducted by Smith [262] in 1956, in which he found that approximately half of all linear programming applications up to that point had been transportation problems, or minimum cost flow problems. They report that Tolstoï [282] published a further article in 1939 called *Methods of Removing Irrational Shipment in Planning*, which was followed by papers by Hitchcock [160] in 1941, Kantorovich in 1942 [175], Kantorovich and Gavurin [176] in 1949 and Koopmans and Reiter [187] in 1951. In a paper titled *Application of the Simplex Method to a Transportation*, Dantzig [77] showed that his simplex algorithm for general linear programs may be specialized to solve the minimum cost flow problem more efficiently. Kuhn [195] presented an algorithm for the assignment problem (assigning people to jobs) in 1955, which Ford and Fulkerson generalized to solve the minimum cost flow problem. They also developed the primal-dual algorithms to solve the capacitated transportation problem, which they later generalized for the minimum cost flow problem [6].

The *Successive shortest path* algorithm was developed independently by Jewell [169] in 1958, Iri [166] in 1960 and Busaker and Gowen [43] in 1961, and was later modified by Edmonds and Karp [96] to remove negative arc lengths by using node potentials (in 1972). At the same time, Minty [217] (1960) and Fulkerson [121] (1961) developed the *Out-of-kilter algorithm*, which was rendered more efficient by Aashtiani and Magnanti [1] in 1976.

Klein [186] developed the *Cycle-cancelling algorithm* in 1967, which was later refined to three special implementations which ran in polynomial time instead of the pseudo-polynomial time of the original algorithm. Goldberg and Tarjan [144] developed the first implementation in 1988, where flow is augmented along minimum mean cost cycles. The second refinement is due to Barahona and Tardos [25] in 1989 (based on work by Weintraub [303] in 1974), where flow is augmented along negative cycles that yield the greatest possible improvement (the authors admit that theirs is not the best algorithm, instead citing the work by Orlin [232] in 1988 as the best to date), while the third implementation was by Wallacher and Zimmermann [298] in 1991, in which they augmented flow along minimum ratio cycles. Sokkalingam, Ahuja and Orlin [264] developed an algorithm based on the *Cycle-cancelling algorithm* in 2000 which has a better running time than procedures before it.

In 1988, Bertsekas and Tseng [33] presented the *Relaxation algorithm*. In the article, the authors tentatively postulated that the *Relaxation algorithm* is faster than the *Primal simplex algorithm* implementation by Grigoriadis and Hsu [150] in 1979 and the *Out-of-kilter algorithm* implementation by Aashtiani and Magnanti [1], but admitted that further testing would be necessary to show this conclusively. The *Relaxation algorithm* has further advantages in that it is able to solve a wide range of problems and is good for large-scale parallel computing. The

superiority of the *Relaxation algorithm* over a special implementation of the *Network simplex algorithm* (which had originally been developed by Dantzig [77] in 1951 when he specialized his *Linear programming simplex algorithm* for the uncapacitated transportation problem) was validated by Grigoriadis [149] in 1986 for some classes of graphs, but for other classes the *Network simplex algorithm* outperformed the *Relaxation algorithm*. However, it must be pointed out that the study by Grigoriadis used an older version of the *Relaxation algorithm* than the study by Bertsekas and Tseng, and the latter study did not consider Grigoriadis' implementation of the *Network simplex algorithm*. The *Network simplex algorithm* had been shown to be superior to the *Primal-dual* and *Out-of-kilter algorithms* in previous studies by Glover, Karney and Klingman [135] in 1974 and by Bradley, Brown and Graves [37] in 1977. Goldfarb and Hau [147], and Tarjan [279] developed polynomial-time versions of the *Network simplex algorithm* in 1988 and 1991 respectively [6]. In 2002 Ahuja, *et al.* [11] presented a better pivot rule for the *Network simplex algorithm*, which reduced the number of pivots performed by the algorithm for certain problem structures.

Most of the polynomial-time algorithms make use of the scaling techniques introduced by Edmonds and Karp in 1972 [6]. In that article, Edmonds and Karp challenged all other mathematicians to develop a strongly polynomial-time algorithm, as theirs was only weakly polynomial. Tardos [277] was the first to solve the problem (publishing in 1985), but was closely followed by Orlin [231] (1984), Fujishige [123] (1986), Galil and Tardos [127] (1986), Goldberg and Tarjan [143, 144] (in 1987 and 1988), Orlin [232] again in 1988 and Ervolina and McCormick [102] in 1990 [6]. At the time Ahuja, *et al.* wrote their book in 1993, the minimum cost flow algorithms with the best time bounds were due to Goldberg and Tarjan in 1987, Ahuja, *et al.* in 1992 and Orlin in 1988 [6, 232]. The papers by Orlin [232] and Ahuja, *et al.* [5] lists all pseudo-polynomial and polynomial algorithms for the minimum cost flow problem by developers, publication date and running time. In 1997 Goldberg [139] presented further improvements to the *Scaling push-relabel method* based on work he did with Kharitonov [141] in 1993 and the *Successive approximation push-relabel method* he developed in his PhD dissertation [138] in 1985 and with Tarjan [144, 145] in 1988 and 1990. He used cost-scaling ideas from Röck's work [243] in 1980 and Bland and Jensen's work [34] in 1992, from the *Push-relabel maximum flow method* by himself and Tarjan [143, 144] in 1987 and 1988 and the *Relaxation algorithm* by Bertsekas and Tseng [33] in 1988. He concluded from his computational studies that the new method was generally faster in practice than some established network simplex codes that existed, such as SPUR (the *Scaling push-relabel method* he developed with Kharitonov [141]), RNET (a network simplex algorithm by Grigoriadis [149]) and RELAX (from the *Relaxation algorithm* by Bertsekas and Tseng [33]).

Various modifications to the minimum cost flow problem have been made for various purposes. In 1995, Chen [60] proposed a new minimum cost flow problem, where the cost structure on the arcs consisted of variable costs and fixed costs and the goal was to minimise the average cost. He called this the minimal average cost flow problem. Chopra, Gilboa and Sastry [62] considered a minimum cost flow problem in 1998 where some of the cost could include the purchase of additional capacity. In 2001, Cai, Sha and Wong [45] presented their work on time-varying minimum cost flow problems, where transit time, transit cost and arc capacity change with time. Demgensky, Noltemeier and Wirth [85] formally introduced the flow cost lowering problem in 2002 (based on work presented by Krumke, Noltemeier, Ravi, Schwarz and Wirth at the International Conference of Operations Research in Zürich in 1998), where the flow cost can be decreased by spending money on upgrading the arcs. With a given flow value and an upper bound on the budget for moving the flow, an approximation for the minimum cost can be found.

Traditional Applications of the Minimum Cost Flow Problem

Ahuja, *et al.* have performed excellent surveys of literature prior to 1993 for applications of both the maximum and minimum cost flow problems [6, 7]. Thus their list of applications of the minimum cost flow problem is summarised in Table 4.2. More detailed information may be found in these books or in the references made therein.

Authors	Year	Application
Jacobs	1954	Choosing when to buy new or launder napkins (caterer problem) [167]
Jewell	1957	Warehousing and distribution of a seasonal product [168]
Prager	1957	Deciding when to buy, sell or hold goods (entrepreneur's problem) [239]
Berrisford	1960	Economic distribution of coal supplies in the gas industry [32]
Fulkerson	1961	Time-cost trade-off on project management [120]
Kelly	1961	Time-cost trade-off on project management [184]
Dantzig	1962	Allocating transmitters to receivers to minimise set-up time [78]
Veinott & Wagner	1962	Modelling linear programs with consecutive 1's in columns [296]
Lawler	1964	Scheduling with deferral costs [199]
Bacharach	1966	Optimal rounding of a matrix [23]
Esau & Williams	1966	Terminal assignment problem for teleprocessing networks [98]
Zangwill	1969	Dynamic lot sizing [308]
Cabot, Francis & Stary	1970	Rectilinear distance facility location problem [44]
Horn	1971	Optimal container inventory and routing [164]
Belford & Ratliff	1972	Racial balancing of schools [28]
White	1972	Distribution of empty rail containers [305]
Daniel	1973	Phasing out capital equipment [76]
Edmonds & Johnson	1973	Directed Chinese postman problem [95]
Picard & Ratliff	1973	Optimal defence of a network [235]
Dearing & Francis	1974	Multifacility minimax location problem with rectilinear distances [84]
Gavish & Schweitzer	1974	Vehicle scheduling problem [131]
Segal	1974	Telephone operator scheduling [252]
Srinivasan	1974	Cash management problems [273]
Dirickx & Jennergren	1975	Optimal parking in an urban area [88]
Dorsey, Hodgson & Ratliff	1975	Multiproduct multifacility production-inventory planning [90]
Francis & White	1976	Cost effective warehouse layout [115]
Arisawa & Elmaghraby	1977	'Hub' and 'wheel' scheduling problems [18]
Glover & Klingman	1977	Distribution problems [136]
Evans	1977	Multistage production-inventory planning [103]
Fulkerson & Harding	1977	Lengthening an opponent's supply network (network interdiction) [122]
Elmaghraby	1978	Just-in-time scheduling [101]
Love & Vemuganti	1978	Tyre mould allocation [204]
Garey & Johnson	1979	Finding the minimum spanning tree on a capacitated network [130]
Lowe, Francis & Reinhardt	1979	Warehouse leasing problem [205]
Mulvey	1979	Finding a good faculty-course assignment [221]
Srinivasan	1979	Multi-attribute marketing models [274]
Bartholdi, Orlin & Ratliff	1980	Modelling linear programs with circular 1's in columns [27]
Farley	1980	Optimal levelling of mountainous terrain [106]
Barr & Turner	1981	Microdata file merging [26]
Maxwell & Wilson	1981	Material handling systems [213]

Table 4.2: Applications of the minimum cost flow problem. Most referenced applications were found in either [6] or [7], or both. Note that this list is not exhaustive.

Authors	Year	Application
Chamlet, Francis & Saunders	1982	Modelling building evacuation [51]
Cox & Ernest	1982	Optimal rounding of a matrix [68]
Slump & Gerbrands	1982	Left ventricle reconstruction from x-ray projections [260]
Cheshire, McKinnon & Williams	1984	Allocation of private contractors to public works [61]
Evans	1984	Optimal storage policy for libraries [104]
Evans	1984	Storage strategy for zoned warehousing [104]
Kourtz	1984	Control of forest fires [190]
Sapountzis	1984	Allocating blood to hospitals from a central blood bank [247]
Gupta	1985	Optimal loading of a hopping aeroplane [154]
Dafermos & Nagurney	1984	Market equilibrium problems [75]
Tso	1986	Automatic chromosome classifications [285]
Zawack & Thompson	1987	City traffic congestion [309]
Denardo, Rothblum & Swersey	1988	Optimal deployment of firefighting companies [86]
Servi	1989	Satellite scheduling [253]
Wagner	1990	Determining k disjoint cuts in a network [299]
Tso, Kleinschmidt, Mitterreiter & Graham	1991	Automatic karyotyping of chromosomes [286]
Levner & Nemirovsky	1991	Just-in-time scheduling [201]
Ahuja, Magnanti & Orlin	1993	Scheduling the harvesting of forests [6] Vehicle fleet planning [6] Assigning students to projects [6] Airline passenger routing [6]
Cova & Johnson	2003	Lane-based evacuation routing [67]
Rieger	2003	Studying the ground state properties of physical systems [241]
Garcia, Lozano & Canca	2004	Coordinated scheduling of production and delivery from multiple plants [129]
Smith	2004	Assignment of telecommunication traffic [261]
Egidi & Maponi	2005	Planar grid generation [97]
Mo, Qi & Wei	2005	Optimization of a distillation process [218]

Table 4.2 (continued): *Applications of the minimum cost flow problem. Most referenced applications were found in either [6] or [7], or both. Note that this list is not exhaustive.*

4.2.6 Solving a Minimum Cost Flow Problem

Many algorithms exist that may be used to solve the minimum cost flow problem as described in §4.2.5; some specialised algorithms may be hundreds of times faster than the all-purpose algorithms typically used by linear programming solvers [105]. Ahuja, *et al.* [6] mention many examples of minimum cost flow algorithms — however, the *Successive shortest path algorithm* discussed in detail by both Ahuja, *et al.* and by Evans and Minieka [105] was chosen to solve this specific problem due to its flexibility [6] and simplicity. It was included in the program written by the author in *FORTTRAN 95* to compute the minimum cost flow. Although it is possible to restrict the flow appropriately when investigating the flow of each fruit type individually, all fruit types were grouped together and treated as a single commodity to find an upper bound to the minimum cost flow solution. This assumption is relaxed in the next chapter, in which each fruit type is considered separately from the others in some parts of the model (such as in fruit-specific packhouse capacities), and together in other sections (such as in the overall capacity of a packhouse).

4.2.7 The Successive Shortest Path Algorithm

Lower bounds on flow over certain arcs potentially play a vital role in the minimum cost model. There are many cases in the fruit industry where role-players are contractually required to send a predetermined volume of fruit to a certain packhouse, cold store, terminal or customer. Farmers and exporters often have agreements with links in the supply chain such as packhouses, cold stores, shipping lines (which means that they are bound to certain terminals, as some shipping lines transport containers and others transport break-bulk fruit), importers and fruit sellers (such as chains of grocery stores) to supply them with a certain volume of fruit. This may even mean that farmers and exporters make a loss on selling fruit in a certain market, just to keep their share of the market in the hope that it will be profitable in future years [192].

To modify an arc (i, j) with both an upper bound u_{ij} and a lower bound ℓ_{ij} on the permissible flow associated with it to an arc with only an upper bound, the lower bound may be subtracted from the upper bound and the supply of node i , and the lower bound may be added to the supply of node j . This process is shown graphically in Figure 4.8.

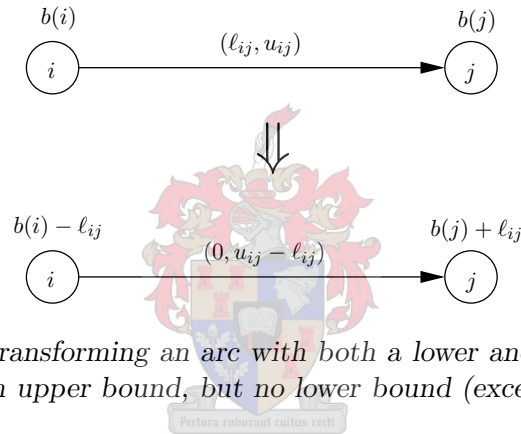


Figure 4.8: A method of transforming an arc with both a lower and upper bound on flow via that arc, to an arc with an upper bound, but no lower bound (except for the trivial bound of non-negativity).

The mechanism of this procedure may be motivated by considering the linear programming formulation of the minimum cost flow problem, in which the objective is to

$$\text{minimise } z(x) = \sum_{(i,j) \in A} c_{ij} x_{ij},$$

where c_{ij} is the cost weight of arc (i, j) and x_{ij} is the flow along the arc. If $b(i)$ represents the supply (or demand) at node i , the objective function is to be minimised subject to

$$\sum_{j:(j,i) \in A} x_{ji} - \sum_{j:(i,j) \in A} x_{ij} = b(i), \quad i \in \mathcal{N}, \quad (4.1)$$

$$\ell_{ij} \leq x_{ij} \leq u_{ij}, \quad (i, j) \in A.$$

The first constraint ensures that the flow out of a node is equal to the flow out of the node in addition to the supply at that node. If the flow x_{ij} is substituted by $x'_{ij} - \ell_{ij}$, the linear programming formulation becomes

$$\text{minimise } z(x') = \sum_{(i,j) \in A} c_{ij} x'_{ij} - \sum_{(i,j) \in A} c_{ij} l_{ij},$$

subject to

$$\begin{aligned} \sum_{j:(j,i) \in A} x'_{ji} - \sum_{j:(i,j) \in A} x'_{ij} &= b(i) + \sum_{j:(j,i) \in A} l_{ji} - \sum_{j:(i,j) \in A} l_{ij}, \quad i \in \mathcal{N}, \\ 0 \leq x'_{ij} &\leq u_{ij} - l_{ij}, \quad (i,j) \in A. \end{aligned}$$

It is clear that the new format allows one to remove the lower bound from the flow restrictions. The only significant change occurs on the right hand side of the mass balance constraint, where the lower bounds on flows along the incoming arcs are added and the lower bounds of the outgoing arcs are subtracted. By making the node i in an arc (i, j) a sink with a mass imbalance of $-\ell_{ij}$ units, at least ℓ_{ij} units are necessarily sent to node i . By making node j a source with a mass imbalance of ℓ_{ij} units, at least ℓ_{ij} units of flow will be assigned to arc (i, j) when the network is transformed back to its original form [105]. The constant term in the objective function may be ignored, as it does not affect the flow in the optimal solution (it is not dependant on the variable for flow, x_{ij}).

The *Successive shortest path algorithm* also makes use of distance labels. However, these are not the same distance labels as those used by the *Shortest augmenting path algorithm*, as they do not represent the shortest distance (in terms of the number of arcs) from the sink node: instead they represent the shortest distance from the source node to each node (in terms of the arc costs). The validity conditions of the maximum flow problem change accordingly to

$$d(s) = 0 \quad \text{and} \quad d(j) - d(i) \leq c_{ij}.$$

So, if $s - i_1 - i_2 - \dots - i_k - t$ is any path in the network from node s to node t , then

$$\begin{aligned} d(i_1) - d(s) &\leq c_{si_1} \\ d(i_2) - d(i_1) &\leq c_{i_1i_2} \\ d(i_3) - d(i_2) &\leq c_{i_2i_3} \\ &\vdots \\ d(i_k) - d(i_{k-1}) &\leq c_{i_{k-1}i_k} \\ d(i_t) - d(i_k) &\leq c_{i_ki_t}. \end{aligned}$$

Adding all the above inequalities together and recalling the first validity condition ($d(s) = 0$), the inequality

$$d(t) \leq c_{si_1} + c_{i_1i_2} + c_{i_2i_3} + \dots + c_{i_{k-1}i_k} + c_{i_ki_t}$$

is obtained. This shows that the distance label on node t is a lower bound on the path “length” from the source to that node. This can, of course, also be shown for all the other nodes that are labelled by distance labels in the network.

As with the *Shortest augmenting path algorithm*, the residual capacities and distance labels are also important concepts for the *Successive shortest path algorithm*, but additional concepts must be introduced. A pseudoflow x_{ij} in a network is a flow which satisfies the capacity and non-negativity constraints ($0 \leq x_{ij} \leq u_{ij}$), but does not satisfy the mass balance constraints (see equation (4.1)). So, for a pseudoflow in a network, the imbalance of a node represents the amount

$$e(i) = b(i) + \sum_{j:(j,i) \in A} x_{ji} - \sum_{j:(i,j) \in A} x_{ij}, \quad i \in \mathcal{N},$$

by which the mass balance constraint has been violated. The node potentials (denoted by $\pi(i)$) are derived from the dual of the linear programming minimum cost flow problem. In the primal problem the objective is to

$$\text{minimise } z(x) = \sum_{(i,j) \in A} c_{ij} x_{ij},$$

subject to

$$\begin{aligned} \sum_{j:(j,i) \in A} x_{ji} - \sum_{j:(i,j) \in A} x_{ij} &= b(i), \quad i \in \mathcal{N} \\ 0 \leq x_{ij} &\leq u_{ij} \quad (i,j), \end{aligned}$$

where $b(i)$ is the supply or demand (negative supply) at node i . In the dual problem the objective is therefore to

$$\text{maximise } w(\pi, \alpha) = \sum_{i \in \mathcal{N}} b(i) \pi(i) - \sum_{(i,j) \in A} u_{ij} \alpha_{ij},$$

subject to

$$\begin{aligned} \pi(i) - \pi(j) - \alpha_{ij} &\leq c_{ij}, \quad (i,j) \in A, \\ \alpha &\geq 0, \quad (i,j) \in A \quad \text{and} \quad u_{ij} \text{ unrestricted}, \quad j \in \mathcal{N}. \end{aligned}$$

A special case of the dual minimum cost flow problem is the shortest path problem. In this case, the supply at the node from where the shortest path is to be found may be set equal to $b(s) = 1$ and the supply at the node to which the path must be found may be set to $b(t) = -1$, where $b(i) = 0$ for all other nodes. As the arc capacities become obsolete for the shortest path case, the variables α_{ij} may be removed from the problem, which means that if $d(i) = -\pi(i)$, this special case of the minimum cost flow problem becomes the shortest path problem [6].

The reduced cost may be found via the node potentials. The reduced cost is derived from the validity conditions set on the distance nodes. It was shown earlier (in §4.2.4) that $d(j) - d(i) \leq c_{ij}$ for all arcs (i,j) , and that equality holds if that arc were on the shortest path from node s to node j . If one were to take all terms to one side of the inequality, the result would be the cost of sending flow along that arc instead of along the shortest path, or

$$c_{ij}^d = c_{ij} - d(j) + d(i) \geq 0, \quad (i,j) \in A.$$

Instead of using the distances, the *Successive shortest path algorithm* makes use of node potentials (recall that $\pi(i) = -d(i)$). Hence the reduced cost is defined as

$$c_{ij}^\pi = c_{ij} - \pi(i) + \pi(j) \geq 0, \quad (i,j) \in A$$

and applies to both the original and residual networks. Two interesting results concerning the reduced costs may be derived from the definition. The first is that for a directed path P from a node a to another node b in the network, the sum of the reduced costs c_{ij}^π of the arcs on that path is

$$\sum_{(i,j) \in P} c_{ij}^\pi = \sum_{(i,j) \in P} c_{ij} - \pi(a) + \pi(b).$$

The second result is that for any directed cycle C , the sum of the reduced costs of the arcs on that cycle is the same as the sum of the costs of the same arcs, i.e.

$$\sum_{(i,j) \in C} c_{ij}^\pi = \sum_{(i,j) \in C} c_{ij}.$$

The mechanism behind the *Successive shortest path algorithm* rests on the following claim. An algorithm has found the minimum cost flow x^* through the network, if and only if the following conditions hold:

- 1 If $c_{ij}^\pi > 0$, then $x_{ij}^* = 0$.
- 2 If $0 < x_{ij}^* < u_{ij}$, then $c_{ij}^\pi = 0$.
- 3 If $c_{ij}^\pi < 0$, then $x_{ij}^* = u_{ij}$.

In order to prove the above claim, three theorems are required.

Theorem 4.3 (Augmenting Cycle Theorem). *Let x and x° be any two feasible solutions of a network flow problem. Then x equals x° plus the flow on at most m directed cycles in $G(x^\circ)$, where m is the number of edges in network G . Furthermore, the cost of x equals the cost of x° plus the cost of flow on these augmenting cycles.*

This theorem, for which a proof may be found in [6] on pp. 83, is a useful tool for proving the *Negative Cycle Optimality Conditions*, stated next.

Theorem 4.4 (Negative Cycle Optimality Conditions). *A feasible solution x^* is an optimal solution of the minimum cost flow problem if and only if it satisfies the negative cycle optimality condition that the residual network $G(x^*)$ contains no negative cost (directed) cycle.*

Proof. Consider a network $G(x)$ with a negative flow cycle. The feasible flow x cannot be optimal, as augmenting flow along the cycle would decrease the objective function. So if the optimal flow is x^* , then $G(x^*)$ cannot contain any negative cycles.

Now, suppose that x^* is not optimal, yet feasible and that $G(x^*)$ contains no negative cycles. Define x^\dagger as the minimum cost flow where $x^\dagger \neq x^*$. Theorem 4.3 states that the difference between the optimal and feasible flows $x^\dagger - x^*$ in G may be decomposed into at most $|\mathcal{A}(G)|$ augmenting cycles in $G(x^*)$, where $\mathcal{A}(G)$ denotes the set of arcs in the network G . Furthermore, the cost of the flows on these cycles is $cx^\dagger - cx^*$. Due to the hypothesis, the cycles all have non-negative total costs; hence $cx^\dagger - cx^* \geq 0$, or $cx^\dagger \geq cx^*$. But since the cost of the optimal flow must be less than that of a non-optimal feasible flow, $cx^\dagger \leq cx^*$. For both results to hold, it follows that $cx^\dagger = cx^*$, and so x^* must be optimal, proving that if a network has no negative cycles, a feasible flow is an optimal one [6]. ■

Theorem 4.4 may now be used to establish a result on the reduced cost optimality conditions of a minimum cost flow solution.

Theorem 4.5 (Reduced Cost Optimality Conditions). *A feasible solution x^* is an optimal solution of the minimum cost flow problem if and only if some set of node potentials π satisfy the reduced cost optimality condition that $c_{ij}^\pi \geq 0$ for every arc (i, j) in $G(x^*)$.*

Proof. Suppose the solution to a minimum cost flow problem x^* satisfies the conditions that $c_{ij}^\pi \geq 0$ for all arcs in the residual network $G(x^*)$. This means, that for every directed cycle C in $G(x^*)$, $\sum_{(i,j) \in C} c_{ij}^\pi \geq 0$. Since the sum of the reduced costs of the arcs in a cycle is equal to the sum of the costs of the same arcs in the cycle (mentioned earlier in this section), $\sum_{(i,j) \in C} c_{ij}^\pi = \sum_{(i,j) \in C} c_{ij} \geq 0$. This means that $G(x^*)$ contains no negative cycles.

Now assume that $G(x^*)$ contains no negative cycles. Let $d(i)$ be the “distance” (measured in terms of arc costs c_{ij}) from the source to any node i in the network $G(x^*)$. It must be recalled, from the discussion on the distance labels earlier in this section, that if there are no negative cycles in a network, these distance labels satisfy the conditions that $d(j) \leq d(i) + c_{ij}$ for any (i, j) arc in $G(x^*)$. The condition may be rewritten as $c_{ij} - (-d(i)) + (-d(j)) \geq 0$, which becomes $c_{ij}^\pi \geq 0$ if we set $\pi(i) = -d(i)$. Thus the reduced cost optimality condition is satisfied by the solution to the minimum cost flow problem [6]. ■

It may now be shown that if the three optimality conditions are satisfied, the flow found in the network is optimal.

Theorem 4.6 (Complementary Slackness Optimality Conditions). *A feasible solution x^* is an optimal solution of the minimum cost flow problem if and only if, for some set of node potentials π , the reduced costs and flow values satisfy the following complementary slackness optimality conditions for every arc $(i, j) \in A(G)$:*

- 1 If $c_{ij}^\pi > 0$, then $x_{ij}^* = 0$.
- 2 If $0 < x_{ij}^* < u_{ij}$, then $c_{ij}^\pi = 0$.
- 3 If $c_{ij}^\pi < 0$, then $x_{ij}^* = u_{ij}$.

Proof. It may now be shown that the three cases listed above are equivalent to the reduced cost optimality conditions found in Theorem 4.5. First, it must be proved that if the potentials of the nodes π and the flow x through the network satisfy the reduced cost optimality conditions, then they must satisfy the complementary slackness optimality conditions. Each condition listed above is considered separately.

Condition 1 If the reduced cost $c_{ij}^\pi > 0$, the arc (j, i) cannot be found in the residual network as $c_{ji}^\pi = -c_{ij}^\pi < 0$ for the arc, which contradicts Theorem 4.5, and so $x_{ij}^* = 0$.

Condition 2 If $0 < x_{ij}^* < u_{ij}$, then both the arcs (i, j) and (j, i) exist, which means that $c_{ij}^\pi \geq 0$ and $c_{ji}^\pi \geq 0$. Since $c_{ij}^\pi = -c_{ji}^\pi$, it follows that $c_{ij}^\pi = c_{ji}^\pi = 0$.

Condition 3 If $c_{ij}^\pi < 0$, the arc (i, j) cannot appear in the residual network as its reduced cost is less than zero, contradicting Theorem 4.5. Thus, the flow in the optimal network across the arc (i, j) is $x_{ij}^* = u_{ij}$.

It has thus been shown that if the node potentials and the flow satisfy the reduced cost optimality conditions, they also satisfy the complementary slackness optimality conditions [6].

Now assume that the three complementary slackness optimality conditions hold for a flow x in G . If $0 \leq x < u$ the first two conditions imply that $c_{ij}^\pi \geq 0$. The residual capacity of an arc (i, j) where $x_{ij} = u_{ij}$ is zero; thus it is not included in the residual network. However, the arc (j, i) does exist as $r_{ji} = u_{ij}$. Since it has been assumed that $c_{ij}^\pi < 0$ for x_{ij} , it holds that $c_{ji}^\pi > 0$ for the arc (j, i) . Thus, the condition $c_{ij}^\pi \geq 0$ holds for all arcs in the residual network. By Theorem 4.5 the flow in the network must therefore be optimal. ■

The optimality conditions are now known with regards to the pseudoflow (or real flow) and the reduced cost. Two further lemmas point out further results that are necessary to prove that the *Successive shortest path algorithm* (presented later in this section) finds the optimum flow for the minimum cost flow problem.

Lemma 4.7 *Suppose that a pseudoflow (or a real flow) x satisfies the reduced cost optimality conditions with respect to some node potentials π . Let d be a vector that represents the shortest distances from source node s to any node in the network, where c_{ij}^π is taken as the length of any arc (i, j) . Then the following two properties hold:*

- 1 *The pseudoflow x also satisfies the reduced cost optimality conditions with respect to the node potentials $\pi' = \pi - d$.*
- 2 *The reduced costs satisfy $c_{ij}^{\pi'} = 0$ for all arcs (i, j) in a shortest path from sink node s to every other node in the network.*

Proof. The reduced costs satisfy $c_{ij}^\pi \geq 0$ for all arcs in the residual network, due to the assumption that the flow x satisfies the reduced cost optimality conditions. Since the distances to the nodes (represented by the vector d) are the shortest distances from the source node, they satisfy the shortest path optimality conditions, where $d(j) \leq d(i) + c_{ij}^\pi$ for all arcs (i, j) in the network.

Using the definition of reduced cost, the shortest path optimality conditions become

$$d(j) \leq d(i) + c_{ij} - \pi(i) + \pi(j).$$

Replacing the above inequality by the two simultaneous inequalities

$$\begin{aligned} c_{ij} - (\pi(i) - d(i)) + (\pi(j) - d(j)) &\geq 0 \\ c_{ij} - \pi'(i) + \pi'(j) &\geq 0, \end{aligned}$$

makes it clear that $c_{ij}^{\pi'}$ is non-negative, proving the first part of the lemma.

To prove that the coefficients $c_{ij}^{\pi'}$ of the arcs on a shortest path from the source to any other node are zero, consider any arc (i, j) on this path. For this arbitrary arc, $d(j) = d(i) + c_{ij}^\pi$. Substituting the definition of the reduced cost into the equation and rearranging, gives

$$\begin{aligned} c_{ij} - (\pi(i) - d(i)) + (\pi(j) - d(j)) &\geq 0 \\ c_{ij}^{\pi'} &\geq 0, \end{aligned}$$

thus proving the second part of the lemma. ■

The next lemma shows that augmenting the flow along the shortest paths does not violate the reduced cost optimality conditions.

Lemma 4.8 *Suppose that a real flow or pseudoflow x satisfies the reduced cost optimality conditions and a new flow x' is found by augmenting flow along the shortest path from a source node to a sink node, then the new flow x' also satisfies the reduced cost optimality conditions.*

Proof. Assume the definitions for π and π' are the same as in Lemma 4.7. That lemma also states that $c_{ij}^{\pi'} = 0$ for all arcs (i, j) on a shortest path P to a sink node. If a flow were augmented along the path, the arc (j, i) would either be added to the residual network for all arcs on P , or the residual capacity would simply increase. Since $c_{ij}^{\pi'} = 0$ for any arc (i, j) on P , it holds that $c_{ji}^{\pi'} = 0$, which means the reverse arcs would also satisfy the reduced cost optimality conditions, thus proving the lemma [6]. ■

The results of Theorems 4.3 – 4.6 and Lemmas 4.7 and 4.8 will be used to prove that the *Successive shortest path algorithm* does, in fact, find the minimum cost flow in a network, after the algorithm has been presented.

Now that a number of optimality conditions and related results have been proved, the *Successive shortest path algorithm* may be presented and shown to produce an optimal solution to the minimum cost flow problem. It is an algorithm that attempts to maintain the optimality of the solution at every iteration and produces a flow that satisfies the capacity constraints of the arcs, at the expense of violating the mass balance constraints. The algorithm selects a node s with a positive supply ($b(s) > 0$) and finds a shortest path P from that node to a node t with a negative supply (or a demand, so $b(t) < 0$). Then the largest possible flow x_{st} is augmented along that path where $x_{st} = \min\{b(s), -b(t), r_{ij} : (i, j) \in P\}$. This continues until all the mass balance constraints are satisfied by the solution, so that $b(i) = 0$ for all $i \in \mathcal{N}$.

The following is a step-by-step guide to the *Successive shortest path algorithm*, adapted from [105]. The pseudo-code may be found in Algorithm 3. An example of the working of the algorithm may be found in Appendix E.3.

Step 1 Let $x = 0$ for all arcs, and let $\pi = 0$ for all nodes. Remove all lower bounds of arcs (i, j) by setting $b(i) = b(i) - l_{ij}$, $b(j) = b(j) + l_{ij}$ and $u_{ij} = u_{ij} - l_{ij}$.

Step 2 Calculate the excess labels e for all nodes and determine which nodes belong to the excess set $\mathcal{E} = \{i : e(i) > 0\}$ and which belong to the deficit set $\mathcal{D} = \{i : e(i) < 0\}$. If

$$\sum_{i \in \mathcal{N}} e(i) \neq 0,$$

terminate the algorithm as the supply and demand are not balanced; hence the mass balance constraints will always be violated. Otherwise, create the residual network and find the reduced costs $c_{ij}^\pi = c_{ij} - \pi(i) + \pi(j)$ for all arcs (i, j) .

Step 3 Select a node s from the set \mathcal{E} and a node t from the set \mathcal{D} , and find the shortest path P from node s to node t using the reduced costs. Let $d(i)$ be the distance from node s to all nodes i .

Step 4 Set $\pi(i) = \pi(i) - d(i)$ for all $i \in \mathcal{N}$, where \mathcal{N} is the set of all nodes. Find the maximum flow δ that may be augmented along the path, where $\delta = \min_{(i,j) \in P} \{e(s), -e(t), r_{ij}\}$. Augment δ units of flow along P to obtain $e(s) = e(s) - \delta$, $e(t) = e(t) + \delta$, $r_{ij} = r_{ij} - \delta$ and $r_{ji} = r_{ji} + \delta$. Return to step 2.

Theorem 4.9 *The Successive shortest path algorithm (Algorithm 3) finds an optimal solution to the minimum cost flow problem.*

Proof. When the algorithm is initialised, the flow on all arcs is set equal to zero. Assuming all arc costs c_{ij} are non-negative, this is a feasible pseudoflow as $c_{ij}^\pi = c_{ij} \geq 0$, because the node potentials π are still zero, thereby satisfying the reduced cost optimality conditions. The algorithm augments flow along the shortest paths — hence, according to Lemma 4.8, the new flow also satisfies the reduced cost optimality conditions. Each iteration of the algorithm augments only along shortest augmenting paths, systematically reducing the excess and deficit at nodes in the network, thus satisfying the reduced cost optimality conditions at each iteration. This continues until the nodes are all balanced (or there are no more augmentable paths in the residual network), which means the final flow satisfies the optimality conditions and the minimum cost flow has been found. ■

Algorithm 3 Successive Shortest Path Algorithm

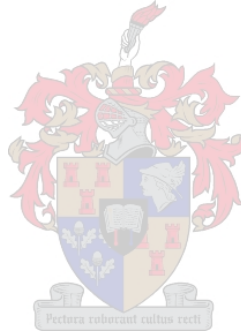
Input: A network with supply information for each node, upper and lower bounds on the arcs and arc weights

Output: The minimum cost flow through the network for the given supply and demand

```

1:  $x = 0$ 
2:  $\pi = 0$ 
3: if  $\sum_{i \in \mathcal{N}} b(i) \neq 0$  then
4:   stop [mass balance constraint is violated]
5: end if
6:  $e(i) = b(i) \forall i \in \mathcal{N}$ 
7: for  $i = 1, |\mathcal{N}|$  do
8:   for  $j = 1, |\mathcal{N}|$  do
9:     if  $(i, j) \in \mathcal{N}$  and  $l_{ij} > 0$  then
10:       $e(i) = e(i) - l_{ij}$ 
11:       $e(j) = e(j) + l_{ij}$ 
12:       $u_{ij} = u_{ij} - l_{ij}$ 
13:    end if
14:  end for
15: end for
16:  $r_{ij} = u_{ij} \forall (i, j) \in A$ 
17:  $r_{ji} = 0 \forall (i, j) \in A$ 
18:  $\mathcal{E} = \{i : e(i) > 0\}$ 
19:  $\mathcal{D} = \{i : e(i) < 0\}$ 
20: while  $\mathcal{E} \neq \emptyset$  do
21:   let  $s = i$ , where  $i \in \mathcal{E}$ 
22:   let  $t = j$ , where  $j \in \mathcal{D}$ 
23:    $c_{ij}^\pi = c_{ij} - \pi(i) + \pi(j) \forall (i, j) \in A$ 
24:    $\pi = \pi - d$ 
25:   find shortest path  $d(i)$  from  $s$  to all nodes  $i$  with respect to the reduced cost  $c_{ij}^\pi$ 
26:   let  $P$  be the shortest path from  $s$  to  $t$ 
27:    $\delta = \min [e(s), -e(t), \min\{r_{ij} : (i, j) \in P\}]$ 
28:   augment  $\delta$  units of flow along path  $P$ 
29:    $r_{ij} = r_{ij} - \delta \forall (i, j) \in P$ 
30:    $r_{ji} = r_{ji} + \delta \forall (i, j) \in P$ 
31:    $e(s) = e(s) - \delta$ 
32:   if  $e(s) = 0$  then
33:      $\mathcal{E} = \mathcal{E} \setminus \{s\}$ 
34:      $e(t) = e(t) + \delta$ 
35:   end if
36:   if  $e(t) = 0$  then
37:      $\mathcal{D} = \mathcal{D} \setminus \{t\}$ 
38:   end if
39: end while

```



4.3 Algorithmic Fruit Export Solutions

This section is dedicated to a presentation and discussion of graph theoretic solutions of the maximum flow and minimum cost flow problems for the South African fresh fruit export infrastructure. The results of the maximum flow problem for peak weeks seven and 29 of 2003 are presented first. Thereafter, the results of the minimum cost flow problem for the same peak weeks are presented and discussed.

4.3.1 Maximum Flow Solutions

The maximum flow of fresh fruit through the national export infrastructure was found for both peaks in the fruit export year, namely week seven in 2003 (summer peak week, 10 – 16 February) and week 29 (the winter peak week, 14 – 20 July). The flow for the summer model was only allowed to pass through the infrastructure in the port of Cape Town. The resulting maximum flow for the summer scenario is 102 773 pallets for the week, while the maximum flow for the winter scenario, where all ports were available for export, is 151 346 pallets for the week. The maximum flow exported from each region is shown in Tables 4.3 and 4.4.

It is clear that the flow restriction for the summer scenario lies with the infrastructure in the port of Cape Town. As the maximum flow algorithm cannot force more pallets of fruit through the Cape Town port infrastructure, it seems as if no fruit is produced in some regions. This indicates that the combined capacities of the packhouses are larger than the loading capacities of the fruit export terminals in Cape Town harbour. In the winter scenario, the situation is reversed. As all ports may now be considered, it turns out that the combined capacity of the four ports is larger than the combined capacity of the packhouses. Thus, either all packhouses or cold stores or a combination of both are used to their maximum potential to export all 151 346 pallets of fruit. Note that only the infrastructure at the ports that allow direct shipping (as opposed to shipping via cold stores or stacks) are considered in this solution of the maximum flow problem for the winter scenario. This is due to the method used by the *Shortest augmenting path algorithm* to find the maximum flow. During the initialisation of the network, the shortest distance (in terms of the number of arcs) is found from the sink to the source. The shortest path from sink to source would be via terminals that allow direct shipping, as the next arcs would indicate transport from packhouses or cold stores. For fruit to be exported via the container terminals, the containers would have to be shipped via the stack, which adds an extra node and arc to the shortest path from sink to source. This means that the stack or harbour cold stores would only be used once the loading rates of the terminals allowing direct shipping are already used to maximum capacity. The fruit types shown in Table 4.3 are misleading, as all fruit was treated as a single commodity. Thus, once the fruit flow has passed the fruit production region part of the network, it is no longer possible to distinguish the different fruit types from one another. Hence, no fruit types are mentioned in Table 4.4. This situation is unsatisfactory and clearly calls for a multicommodity flow approach, as taken in the following chapter. The results obtained nevertheless yield upper bounds on the maximum flow scenario in the fruit industry.

4.3.2 Minimum Cost Flow Solutions

Reputable cost data with respect to the cooling cost of cooling subtropical fruit could not be found; thus the minimum cost flow was found for three cooling costs; namely for R120, R160 and R200 per pallet, as described in §2.4.1. For the summer scenario (peak week seven of 2003), the flow results for the R160 and R200 cooling costs are identical, while the flow pattern for the

subtropical cooling cost of R120 is different. For the winter scenario, the flow pattern for the R120 and R160 are identical, while the flow pattern for the R200 cooling cost differs from that of the other two. The other costs included in the model were those costs for the cooling of fruit

Region	Summer Peak		Winter Peak	
	Fruit	Volume	Fruit	Volume
Levubu	Hard citrus	10 640	Hard citrus Avocado	10 640 610
Tzaneen	Hard citrus	14 555	Hard citrus Avocado	14 555 1 786
Hoedspruit	Hard citrus Mango	3 181 269	Hard citrus	3 181
Kiepersol	Hard citrus	721	Hard citrus Avocado	721 1 203
Nelspruit	Pome Hard citrus	300 3 617	Pome Hard citrus Avocado	300 3 617 416
Malelane	Pome Hard citrus Mango	306 5 600 150	Pome Hard citrus	306 5 600
Pongola	Hard citrus	1 454	Hard citrus	1 454
Nkwalini	Hard citrus	6 207	Hard citrus Soft citrus Avocado	7 322 58 25
KwaZulu-Natal			Hard citrus Avocado	1 007 287
Eastern Cape Midlands			Hard citrus	688
Sundays River Valley			Hard citrus Soft citrus	8 083 2 268
Patensie			Hard citrus Soft citrus	6 149 410
Langkloof			Pome	10 513
Klein Karoo	Grapes	308	Pome Hard citrus	26 2 545
GEVV	Grapes	6 437	Pome Hard citrus	17 361 700
Boland	Grapes	6 527	Pome Hard citrus Soft citrus	1 298 3 776 428
Ceres	Grapes	3 159	Pome Hard citrus	9 750 2 346
De Doorns	Grapes	13 904		
Piketberg	Grapes	4 550	Hard citrus	2 428
Citrusdal	Grapes	300	Hard citrus	6 428
Orange River	Grapes	20 588	Hard citrus	135
Hartswater			Hard citrus	1 070
Potgietersrus			Hard citrus	2 365
Marble Hall			Hard citrus	9 365
Rustenburg			Hard citrus	26
Middelburg			Hard citrus	9 800
Free State			Pome	300
Total		102 773		151 346

Table 4.3: Largest possible fresh fruit export volumes (in pallets) from each production region for the summer and winter peak weeks (weeks 7 and 29 of 2003).

Summer Peak		Winter Peak	
Infrastructure	Volume	Infrastructure	Volume
BS	5 000	MpftB	22 615
CTS	39 390	MfptC	3 877
CTcomS	1 450	DfptB	45 320
CTCT	45 840	DfptC	7 769
CTfptC	11 200	DsaftB	13 140
CTsaftC	3 733	DsaftC	2 253
CTfptB	31 500	PEfptB	22 615
CTsaftB	10 500	PEfptC	3 877
		CTfptB	14 947
		CTfptC	11 200
		CTsaftC	3 733
Total	102 773	Total	151 346

Table 4.4: *Largest possible fruit export volumes (in pallets) from the terminals for the summer and winter peak weeks (weeks 7 and 29 of 2003). Explanations to the abbreviations listed in this table may be found in Appendix D.*

(as summarised in Table 2.5), transportation (as shown in Tables F.1 – F.4), port handling (as discussed in §2.4.3) and shipping (see Table 2.8). All additional costs, as discussed in §2.4.5, are also considered.

The reason for this discrepancy in flows, is that exporting containerised fruit is cheaper than break-bulk export. Transporting cooled fruit in containers to a container terminal, for example, leads to a cheaper export cost than transporting break-bulk fruit to the ports. However, since specific fruit types are not taken into account in the graph theoretic models of this chapter, volumes of subtropical fruits from packhouses that handle both subtropical and citrus fruits may flow along arcs representing ambient transport usually reserved for hard citrus fruit only. Thus, there is a threshold at which cooling subtropical fruit to the correct temperature before transport by container becomes too expensive, and it becomes cheaper to transport the fruit uncooled to the ports. In the summer scenario, this threshold value lies between R120 and R160, while in the winter scenario, the threshold is between R160 and R200. The volumes of fruit exported through each of the terminals is shown in Table 4.5.

The total minimum cost of exporting the 40 239 pallets exported during week seven of 2003 was approximately R71 509 814, R71 521 073 and R71 531 113 for the cases where subtropical cooling cost are taken as R120, R160 and R200 per pallet, respectively. For the same three cooling rates, the estimated minimum costs of exporting all 82 722 pallets of fruit exported during week 29 of 2003 were approximately R120 244 670, R120 317 150 and R120 384 973, respectively. If one were to allow fruit to be exported via Cape Town harbour only in the summer peak, the total cost would be R73 237 346 for a subtropical cooling cost of R160, which represents a difference of R1 716 272 when compared to the solution when allowing fruit to be exported through all ports. These costs do not include packing and PPECB inspections.

To determine which port infrastructure worked out cheapest to export fruit for each of the regions, the port capacities were set much larger than the volumes exported during either week seven or 29 of 2003. The results for a subtropical fruit cooling cost of R160 are shown and discussed from this point onwards. Also, the cost of sending fruit through only Cape Town harbour was considered to determine the cost of closing the other ports to fruit export. The results of the models also show a pattern emerging with regards to the ports to which export from regions should occur to minimise cost. To verify the pattern, the models for each peak

Infrastructure	Summer Scenario		Winter Scenario	
	R120 cooling	R160 & R200 cooling	R120 & R160 cooling	R200 cooling
MfptCs	712	818		
MfptB	23	23	22 615	22 615
MfptC	1 567	1 567	3 877	3 877
DfptB			15 382	15 619
DsaftB			13 140	13 140
DsaftC	130	130	1 198	961
PEfptB	1 326	1 326	14 151	14 151
PEfptC	883	883	108	108
CTS	21 853	21 853		
CTfptCs	8 000	8 000	8 000	8 000
CTCT	21 853	21 853		
CTfptB	2 724	2 724	518	518
CTfptC	8 000	8 000	8 000	8 000
CTsaftC	3 733	3 733	3 733	3 733

Table 4.5: Volumes of fruit (in pallets) passing through the infrastructure at the ports in the standard solution of the minimum cost flow problem. Explanations to abbreviations listed in this table may be found in Appendix D.

week with no restrictions on the port capacities was used to find the destination harbour from each region. This is due to the absence of constraining capacities that would force fruit flow to a more expensive port (in terms of transportation, handling and cooling costs from a region). The Levubu, Tzaneen, Hoedspruit, Kiepersol, Nelspruit, Malelane, Marble Hall and Middelburg regions should export via Maputo. Durban should only export fruit from the Pongola, Nkwalini, KwaZulu-Natal Midlands, Hartswater and Free State regions, while Port Elizabeth need only serve the Eastern Cape Midlands, Sundays River Valley, Patensie and Langkloof regions in a bid to save costs. Cape Town harbour should export fruit from many regions in the minimum cost flow solution, including Klein Karoo, GEVV, Boland, Ceres, De Doorns, Piketberg and Orange River regions. The volumes that could potentially flow through the ports for the lowest possible cost are shown in Table 4.6. The total export costs for the summer and winter scenarios where the port capacities are very large, are approximately R70 547 328 and R118 066 732, respectively.

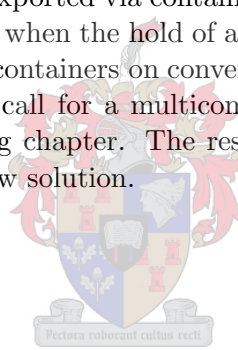
Infrastructure	Summer Scenario	Winter Scenario	Capacity
MfptCs	818		3 100
CTfptCs	13 164	6 886	8 000
MfptB	23	41 349	22 615
MfptC	1 567	2 335	3 877
DsaftB		8 834	13 140
DsaftC	130	1 198	2 253
PEfptB	1 326	14 151	22 615
PEfptC	883	108	3 877
CTfptC	13 164	6 886	11 200
CTsaftC	23 146	7 861	3 733

Table 4.6: Volumes of fruit (in pallets) passing through the infrastructure at the ports in the solution of the minimum cost flow problem where port capacities have been set to numbers large enough not to restrict any fruit flow. Infrastructure components with zero flow for this scenario are omitted. Explanations to abbreviations listed in this table may be found in Appendix D.

From Table 4.6 it may seem that it would be cheapest to export by far the largest volume of fruit via Maputo in bulk in winter, but in summer Cape Town would have to deal with the largest volumes. This result may be attributed to the fact that large volumes of hard citrus fruit are produced in the north-eastern provinces of South Africa in winter, for which Maputo is the closest port. The south-western part of South Africa produces the largest volumes of deciduous fruit in the country in summer, and for those producers Cape Town is the nearest port. In the winter peak week, the export volumes shown in Table 4.6 would require an 83% increase in break-bulk loading capacity at the FPT operation in Maputo. Also, large volumes of fruit are exported via containers on conventional vessels, while little is exported in break-bulk in the summer solution of the minimum cost flow model. The bias to containerised shipping is due to the lower shipping cost of containerised fruit. The break-bulk terminals are chosen by the model for container export because, although the shipping cost and additional costs are the same, the container terminal handling charges at the break-bulk terminals are lower than at the container terminal. This can be seen when comparing the costs in Table 2.6 to the container terminal handling costs given in §2.4.3.

The single-commodity approach to solving the minimum cost problem is unsatisfactory for many reasons. One of these reasons is the requirement of using a single cost for fruit cooling, when the fruit types often have different costs associated with their cooling. Another unsatisfactory result was that most of the fruit should be exported via containers on the deck of conventional vessels. This is, of course, only typically done when the hold of a vessel is already full. Thus, one would have to restrict the export of fruit in containers on conventional vessels according to the volume of fruit in the holds. These reasons call for a multicommodity (mathematical programming) approach, as is taken in the following chapter. The results obtained here nevertheless give a lower bound to the minimum cost flow solution.

4.4 Chapter Summary



The purpose of this chapter was to present graph theoretic methods for solving the maximum flow and minimum cost flow problems. These were required to find the maximum volume of fruit that could possibly pass through the South African fruit export infrastructure in a week, and to investigate at what cost the fruit could be exported from South Africa. The chapter began with the formulation of the graph model. In this section, important assumptions made during the modelling process were presented and discussed, and simplified graphical representations of the model were shown in an attempt at rendering the structure of the model clearer.

In the survey of literature which followed some statistics of the sizes of the graph theoretic models, the origin of the maximum flow problem was shown to be in the US military sector in the 1950s. A brief history on the evolution of algorithms used to solve maximum flow problems was followed by some examples of applications of the maximum flow problem to many different topics. Thereafter, *Ford's algorithm* to find a shortest path was discussed, as the *Shortest augmenting path algorithm*, the maximum flow algorithm used to find the maximum flow of fruit through the South African fruit export infrastructure, requires the shortest path in a network to be found, so that flow volumes may be augmented. The *Shortest augmenting path algorithm* was also presented, discussed in detail and it was shown to find the maximum flow in a network.

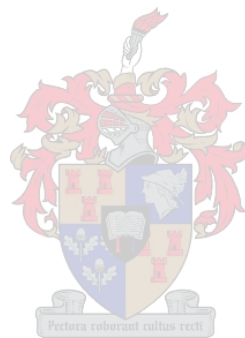
The same structure was followed for the minimum cost flow problem. After a brief discussion on the origins of the problem, the algorithmic history of the algorithms used to find the minimum cost flow in a network, and some applications of the problem, were summarised. This was

followed by a presentation of the *Successive shortest path algorithm*, which was proved to find the minimum cost flow in a network. A method of removing lower limits on arc flows from a residual network made it possible to add lower limit constraints to the models when required.

Finally the solutions to the maximum flow and minimum cost flow problems were found in the context of the South African fresh fruit export industry (thereby realising Thesis Objectives VI and VII, as stated in §1.2). The maximum flow of fruit through the export infrastructure would appear to be 102 773 pallets during the summer peak (only infrastructure in Cape Town was allowed to export fruit during the summer peak so as to best recreate the demand on the port Cape Town during this time). The maximum flow possible through the entire country's fruit export infrastructure (now including Maputo, Durban and Port Elizabeth ports) during the winter peak was found to be 151 346 pallets. Both of these capacities are far larger than the export of 2003 and even exceed the expected fruit volumes of 50 385 pallets during the summer peak in 2010, and 108 052 pallets of fruit during the winter peak in 2010.

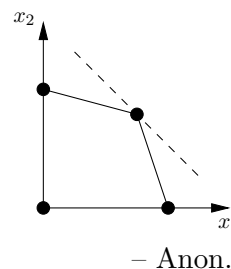
A number of scenarios were considered for the minimum cost flow problem. The cost of cooling subtropical fruit to the correct temperature could not be determined — hence three values were chosen and the corresponding minimum cost flow results compared. It was found that, for the average cooling cost, R71.5 million plus packing costs and a PPECB inspection cost would need to be spent to export the volume of fruit exported during week seven of 2003 (the summer peak). However, if fruit were only allowed to be exported from Cape Town, an additional R1.7 million would have to be spent. To export the fruit volumes exported during week 29 of 2003 (the winter peak), the cost (if a subtropical cooling cost of R160 per pallet is assumed) would be R120.3 million plus a packing cost and a PPECB inspection cost. It was found that the exporters in the Mpumalanga and Limpopo provinces could potentially save significant amounts of money in the winter months if the fruit export capacity in Maputo were to be expanded and upgraded. When comparing the standard solutions to the solutions to the models where the port capacities were set very high, these exporters could save up to R2.2 million in that week alone.

It became clear during the construction of the models and the analysis of the maximum flow and minimum cost flow solutions, that the single-commodity graph theoretic approach taken in this chapter is inadequate for plausible results and that a multicommodity (typically mathematical programming) approach is called for. However, the results in this chapter are not meaningless, as the results for the maximum flow solutions represent upper bounds to the real maximum flow, and the minimum cost flow solutions represent lower bounds to the real minimum cost flow.



Chapter 5

Mathematical Programming Approach



This chapter is dedicated to a multicommodity mathematical programming approach to the modelling of the South African fresh fruit export infrastructure. In the previous chapter it was found that the single-commodity graph theoretic approach was not sufficient for accurate maximum flow and minimum cost flow estimates. There were cases where fruit types were allowed to flow along arcs not intended for them, and a super-cold store arc was assigned a single fruit cooling cost, even though the cooling cost differs between fruit types. It was also difficult to discern which fruit types flowed in which arcs, as all were treated as a single commodity. This made it impossible to assign subtropical fruits to CA containers for transport.

In the remainder of the chapter, a brief survey of literature on the multicommodity flow problem is performed, and some applications of multicommodity flows are presented in Table 5.1. After a formulation of the multicommodity maximum flow model, some model statistics are presented and the tools used to solve the linear programming models are introduced and discussed. Thereafter the results for the maximum flow problem are presented for three scenarios. The first scenario is in the summer peak week (week seven of 2003), where only the port of Cape Town was allowed to be used for export. Many sub-scenarios are also considered, where the flow of each fruit type was allowed a higher weighting than the other fruit types. The second scenario is a maximum flow problem for the entire fruit export infrastructure during the winter peak week (week 29 of 2003). The aim of the final scenario is to find the maximum flow possible through the South African fresh fruit export infrastructure during the winter peak, but with the citrus export from Durban allowed an additional weighting in the objective function so as to mimic the pressure under which Durban is placed during the winter peak export week. This is followed by a formulation of the minimum cost flow model, which is followed by a minimum cost flow investigation into the volumes of fruit exported during the summer and winter peaks of 2003.

5.1 A Concise History of the Multicommodity Flow Problem

There is a large body of literature on the multicommodity flow problem. A very brief algorithmic history of the multicommodity flow problem is presented in this section, and some applications are listed in Table 5.1.

Interest in solving multicommodity flow problems faster than via the simplex algorithm began as early as 1958, when Ford and Fulkerson [113] proposed the column generating approach. This paper led to work by Dantzig and Wolfe [80, 81] in which they developed what is now known as the *Dantzig-Wolfe decomposition* procedure. In 1966 Tomlin [283] did further work on the column generating approach for his study on traffic assignment. In his survey of multicommodity flow problems, Assad [19] states that work by Jewell in 1958¹ and 1966 [169, 170] led to the first algorithm specifically for the multicommodity maximum flow problem (which Jewell used to solve sub-problems in a minimum cost problem). Cremeans, Smith and Tyndall [71] first introduced the *price-directive decomposition algorithm* in 1970, which Swoveland [276] in 1971, Chen and Dewald [59] in 1974 and Assad [21] in 1980 developed further. The *resource-directive decomposition algorithm* was put forth by Sakarovitch [246] in 1969, Geoffrion and Graves [132] in 1974, and further work was done on it in 1977 by Kennington and Shalaby [180] and in 1980 by Kennington and Helgason [181]. Other algorithms arose in the 1980's to solve the multicommodity flow problems. For example, in 1986 Tardos [278] developed an algorithm which solves this problem in strongly polynomial time.

Further surveys on the uses and algorithmic aspects of multicommodity flow problems include the book by Ahuja, Magnanti and Orlin [6], and papers by Kennington [179], Assad [19] and Chardaire and Lissner [54].

5.2 Formulation of the Mathematical Program

One weakness of the graph theoretic approach towards solving flow problems (as introduced in Chapter 4) is that it is cumbersome to incorporate multicommodity flow constraints in that approach. In multicommodity flow problems, one has more than one commodity (in this case fruit group) that must be transported between points along the same infrastructure components (roads, railways, packhouses, cold stores and terminals), where rules or constraints regarding each commodity type may be different. As all fruit types have distinct temperature control protocols set by the PPECB, they have to be considered as separate commodities (see Table 5.2 for a list of fruit groups). During the data gathering process it became clear that the packhouses and cold stores have capacities that vary between fruit groups, making it even more important to adopt a multicommodity flow approach — this complication may be resolved by allowing a volume of each fruit group to flow through each packhouse node that is no more than the fruit-specific capacity of the super-packhouse.

An example of where the single commodity flow approach is inadequate occurs in the KwaZulu-Natal region (see Figure 5.1). It is possible to group the packhouses in that region into three groups: those that pack hard citrus only, those that pack avocados only and those that pack both. As the peak volumes for both fruits occur in winter, it is necessary to consider both commodities when studying the winter peak weeks. Some avocado volumes may be allocated along the arc leading from the packhouses directly to the ports. However, this arc should only be reserved for hard citrus as avocados must be transported and stored at 3°C according to PPECB

¹Assad incorrectly referenced Jewell's PhD thesis [169] as completed in 1968. It was completed in 1958.

Authors	Date	Application
Jewell	1957	Warehousing of seasonal products [168]
Clarke & Surkis	1968	Racial balancing of schools [65]
Bellmore, Bennington & Lubore	1971	Multivehicle tanker scheduling problem [29]
Kaplan	1973	Optimal deployment of resources [177]
Geoffrion & Graves	1974	Multicommodity distribution system design [132]
Golden	1975	Routing of multiple commodities [146]
Evans	1977	Multiproduct multistage production-inventory planning [103]
Assad	1980	Rail freight planning [20]
Bodin, Golden, Schuster & Rowing	1980	Rail freight planning [35]
Cranic , Ferland & Rousseau	1984	Routing of multiple commodities [70]
Korte	1988	VLSI chip design [188]
Moiseenko	1990	Interdependent flows in regional transportation networks [219]
Berge & Hopperstad	1993	Aircraft capacity assignment [31]
Gu, Johnson, Nemhauser & Wang	1994	Airline fleet assignment [153]
Tsujino, Fukushima & Ibaraki	1994	Prediction of traffic congestion in highway networks [287]
Greenwald & Dean	1996	Routing packages among vehicles in transportation networks [148]
Haghani & Oh	1996	Disaster relief [155]
Oh & Haghani	1997	Disaster relief [227]
Powell & Carvalho	1997	Dynamic fleet management [238]
Rushmeier & Kontogiorgis	1997	Airline fleet assignment [245]
Iakovou, Douligeris, Li, Ip & Yudhbir	1999	Routing of hazardous materials in marine waters [163]
Cappanera & Gallo	2004	Crew rostering [47]

Table 5.1: Some applications of the multicommodity flow problem. All unreferenced applications were found in either [6] or [7], or both. Chardaire and Lisser list many more examples of multicommodity flow applications in their paper [54]. However, only those in the transportation and logistics sector were reproduced here.

guidelines [234]. Avocados are also generally transported in controlled atmosphere containers, which are not used for citrus transport. Thus, a multicommodity flow approach must be taken to be able to differentiate between requirements for and restrictions on the different fruit groups.

Minimum Flow Constraints

If a role-player in the fresh fruit export industry is contractually obligated to send some minimum volume of fruit to a particular packhouse, cold store or terminal or to use a certain mode of transportation, the arc leading to and/or from that node may be assigned a lower limit equal to the volume of fruit that should be sent there by contract. This leads to a constraint of the form

$$x_{ij}^k \geq L_{ij}^k \geq 0, \quad (i, j) \in \mathcal{A}, k \in \mathcal{K},$$

indicating that the flow volume x_{ij}^k of commodity $k \in \mathcal{K}$ along the arc (i, j) in the set of all existing arcs \mathcal{A} , must be equal to or larger than the lowest allowed flow along that arc, L_{ij}^k . Of course, if the flow of a specific commodity or all commodities from all arcs must be more than or equal to a certain minimum value for a specific node, one may sum over all i (such that $(i, j) \in \mathcal{A}$) and sum over all $k \in \mathcal{K}$. This will force fruit to that node, if it is possible to do so. The lower bound L_{ij}^k may just be zero in many instances.

Fruit Group	Included fruit types
Grapes	Table grapes
Stone	Apricots, nectarines, peaches, plums
Pome	Apples, pears
Hard citrus	Oranges, grapefruit, lemons, limes
Soft citrus	Mandarins
Avocados	Avocados
Mangoes	Mangoes
Litchis	Litchis

Table 5.2: Groups into which the fruit types have been sorted for a multicommodity flow approach towards modelling fruit transportation, as the previous three groups (deciduous, citrus and subtropical fruit) are not sufficient for temperature protocol differentiation. These groups were determined through experience gained during the data gathering process and by studying the cold chain protocols set by the PPECB. Note that fruits belonging to the soft citrus group are often called easy-peelers.

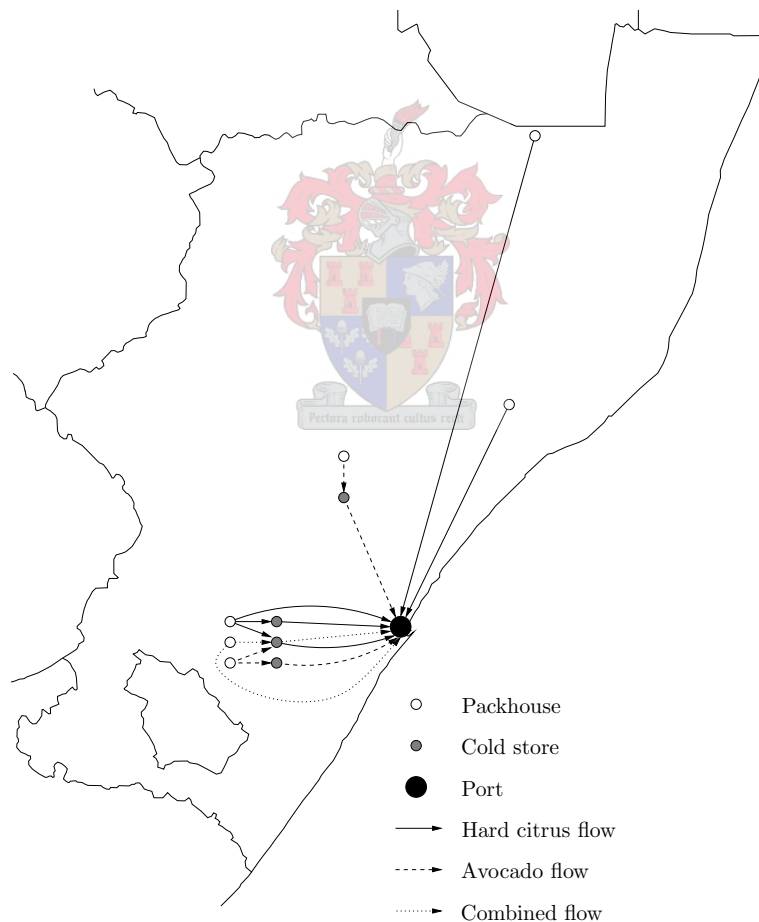


Figure 5.1: Schematic of a simplification of the KwaZulu-Natal portion of the network representing the South African fruit export infrastructure. Note that nodes representing transport options (such as unrefrigerated transport, refrigerated transport or transport in containers) have been omitted.

Maximum Flow Constraints

The flow of fruit may be constrained by placing maximum flow constraints on arcs or nodes. These maximum flow constraints may represent the capacities of packhouses, cold stores or terminals. For example, this value may be the maximum rate at which a packhouse may pack, the maximum rate at which cranes at the terminals are able to load fruit onto a vessel, *etc.* Some maximum flow constraints may be considered infinite for all practical reasons. This may apply to the constraints that determine maximum road usage (results discussed by Van Dyk and Maspero [293] indicate that this is the current situation in South Africa). Since the limitation on fruit flow imposed by the infrastructure is often dependent of fruit type, maximum flow restrictions are assumed to take the form

$$\sum_k x_{ij}^k \leq U_{ij}, \quad (i, j) \in \mathcal{A}, \quad (5.1)$$

if the restriction is on an arc. However, for the maximum flow and minimum cost flow models, restrictions were placed on the volumes of fruit that would pass through nodes as well. This kind of restriction typically takes the form

$$\sum_k x_{ij}^k \leq U_j, \quad (i, j) \in \mathcal{A},$$

where x_{ij}^k again represents the volume of flow of commodity $k \in \mathcal{K}$ along the arc $(i, j) \in \mathcal{A}$. This constraint now states that all flow into a node must be less than or equal to the maximum flow that the node is able to accommodate (capacity of the infrastructural element). The same does not apply for the lower limit constraint in this case study, as contracts may exist that contractually bind one role-player to another — thus the lower bound would still remain on an arc between two infrastructural elements.

For example, the capacity of a packhouse or cold store may be different for the various fruit groups, such as in the Boland region where the group of packhouses handling stone, pome, hard citrus and soft citrus have different capacities for the four fruit groups: 1 037 pallets of stone fruit, 2 028 pallets of pome fruit, 1 600 pallets of hard citrus and 1 872 pallets of soft citrus may be packed per week. Thus, there would have to be a maximum flow constraint for each of the fruit types, except the one that has the highest capacity. The maximum flow capacity for the node (including all fruit types) would thus be equal to the capacity of the fruit with the highest individual capacity (pome in this example). There are many other examples of this phenomenon throughout the South African fresh fruit export supply chain.

Ambient Citrus Restriction

Only a certain percentage of the hard citrus on a vessel may be at ambient temperature. This is determined by the PPECB, taking into account the shape of the hold and the cooling system used by the vessel. On average, 31% of the fruit may be ambient hard citrus [242]. Thus a constraint such as

$$\sum_{(i,j) \in \mathcal{U}} x_{ij}^C \leq \chi \left(\sum_{(i,j) \in \mathcal{R}} \sum_{k \in \mathcal{K}} x_{ij}^k + \sum_{(i,j) \in \mathcal{U}} x_{ij}^C \right), \quad (i, j) \in \mathcal{A}, \quad 0 \leq \chi \leq 1$$

has to be imposed, where \mathcal{U} is the set of all arcs representing routes into a terminal j on which ambient hard citrus is transported. Here \mathcal{R} is the set of all arcs representing routes by which refrigerated fruit may be transported to a terminal j and χ is the fraction of hard citrus that may be loaded onto a vessel at ambient temperature (taken as 0.31 throughout the models to follow). This constraint may be simplified to

$$\sum_{(i,j) \in \mathcal{R}} \sum_{k \in \mathcal{K}} x_{ij}^k \geq \left(\frac{1}{\chi} - 1 \right) \sum_{(i,j) \in \mathcal{U}} x_{ij}^C, \quad (i,j) \in \mathcal{A}, \quad 0 \leq \chi \leq 1.$$

Conservation of Flow Constraints

All fruit flowing through the export infrastructure from production regions should be accounted for (no fruit may go missing). Therefore a conservation of flow law is included in the model. All flow into node i ($\sum_h x_{hi}^k$) subtracted from the flow out of the node ($\sum_j x_{ij}^k$) must be equal to the supply of that node, for each commodity k . Thus all nodes representing production regions will have a positive supply/net flow (since there is no flow into such nodes) and all nodes representing foreign destination markets will have negative supply values (as there is no flow out of such nodes). If one ignores all fruit rejected by the PPECB, packed for the local market or processed in South Africa (*i.e.* one considers only export fruit), then all intermediate nodes should have their supply set equal to zero ($b_i = 0$) as no fruit is to remain behind, but should follow through to a foreign market.

However, some net flows at packhouses or cold stores may be assigned negative values to simulate fruit being rejected or packed/cooled for the local market. This may be achieved by changing the conservation law to

$$\sum_j x_{ij}^k - \phi \sum_h x_{hi}^k = b_i, \quad (h,i), (i,j) \in \mathcal{A}, \quad k \in \mathcal{K},$$

where ϕ is the expected or average proportion of fruit packed, cooled or loaded for export, expressed as a percentage of the fruit that arrives at the node. Here ϕ may be assigned the value 1 if only export values are used throughout the data gathering process. For example, if it is known what percentage of fruit is packed for export by a packhouse, one may simply multiply the maximum packing volume by this percentage to obtain an upper limit on the volume packed for export. As fruit not destined for the export market may be returned to a cold store until all export fruit has been packed, a value of $\phi = 1$ was assumed during this study.

Variable Constraints

A final constraint type should be included to ensure that all flow is non-negative, *i.e.* all x_{ij}^k values should be positive or zero. This constraint, of course, becomes obsolete if non-trivial lower-bound constraints are included for all $(i,j) \in \mathcal{A}$ and all $k \in \mathcal{K}$.

The final model

The objective in the maximum flow problem is therefore to

$$\left. \begin{aligned} \text{maximise } z &= \sum_k \sum_{(i,j) \in \mathcal{T}} x_{ij}^k \\ \text{subject to } x_{ij}^k &\geq L_{ij}^k, & (i,j) \in \mathcal{A}, k \in \mathcal{K} \\ \sum_k x_{ij}^k &\leq U_j^k, & (i,j) \in \mathcal{A}, k \in \mathcal{K} \\ \sum_{(i,j) \in \mathcal{R}} \sum_{k \in \mathcal{K}} x_{ij}^k &\geq \left(\frac{1}{\chi} - 1\right) \sum_{(i,j) \in \mathcal{U}} x_{ij}^C \\ \sum_j x_{ij}^k - \sum_h x_{hi}^k &= b_i^k, & (h,i), (i,j) \in \mathcal{A}, k \in \mathcal{K} \\ x_{ij}^k &\geq 0, & (i,j) \in \mathcal{A}, k \in \mathcal{K}, \end{aligned} \right\} \quad (5.2)$$

where \mathcal{T} is the set of arcs that are adjacent to nodes representing loading areas at the container, FPT or SAFT terminals, where \mathcal{A} is the set of all arcs and where \mathcal{K} is the set of all commodity types.

The variables representing the flow (x_{ij}^k) are decision variables which are to be found when solving the model — the solution will thus indicate the volume of fruit that should flow along the link between nodes i and j in an optimal solution. The values represented by L_{ij}^k , U_{ij}^k and b_i^k are parameters that are to be found or estimated before the model may be solved. The lower bounds on the fruit flow from each of the regions during the winter peak may be found in Table 5.21 (the summer peak week regional export volumes for 2003 can be found in the seventh file of the summer scenario spreadsheet input files, e.g. `SumMinUnresLP7.csv`). The upper bounds on the nodes representing packhouse nodes may be found in Table 3.1, whilst Table 3.2 lists upper bounds on the flow through cold stores. Furthermore, Table 3.13 lists upper bounds on nodes representing stacks and container terminal loading rates, while Table 3.14 summarises the maximum volume of fruit that may be loaded at the break-bulk terminals. These parameters may also all be found in the spreadsheet files on the CD attached to the thesis.

Model Statistics

Two versions of the model (5.2) had to be designed to solve the multicommodity maximum flow problems. The model designed to solve the maximum flow problem for the summer peak week (week seven in 2003) and all its variations, has 906 constraints and 2542 variables. The model designed to solve the maximum flow problem for the winter peak week (week 29 in 2003) has 850 constraints and consists of 4069 variables.

5.3 Solving the Multicommodity Flow Problems

The solutions presented in this thesis were solved on the *XPRESS-MP* mixed-integer linear programming solver developed by Dash Associates [17]. A program was written by the author in *FORTTRAN 95* to convert models in a spreadsheet format (it was found to be the most suitable format in which to create the model) to *Mathematical Programming System (MPS)* format, which is one of the input formats required by the *XPRESS-MP* mixed-integer linear programming solver and *PCx* [73], another program used to verify the results. The *MPS* files were submitted to the online *Network-Enabled Optimization System (NEOS) Server for Optimization* [16], a free resource to solve optimisation problems. The *NEOS Server* is an Internet-based optimisation service open to the public to use for

- Mixed Integer Nonlinearly Constrained Optimisation,
- Mixed Integer Linear Programming,
- Nonlinearly Constrained Optimisation,
- Semidefinite and Second Order Cone Programming,
- Semi-Infinite Optimisation,
- Linear Programming,
- Bound Constrained Optimisation,
- Unconstrained Optimisation,
- Linear Network Optimisation,
- Complementarity Problems,
- Nondifferentiable Optimisation,
- Stochastic Linear Programming,
- Global Optimisation, and
- Combinatorial Optimisation [16].



Users can submit models (in various formats) to the server via email, the NEOS Submission Tool (a program installed on a user's computer) or via their Internet browsers. The server identifies which of the solvers in its solver library is best suited to the model, further refines the information in the model, executes the solver and returns the results to the user [74]. The *NEOS Server* is explained in greater detail in [74, 89, 152].

5.4 Maximum Flow Solutions

In this section, solutions to the maximum multicommodity flow model (5.2) are presented and discussed for three scenarios chosen by FE van Dyk, Fruitlog project manager at the CSIR — Transportek. To determine whether a binding constraint (such as the capacity of a cold store or stack), in fact, represents an infrastructure bottleneck, the shadow price of the constraint had to be an integer (depending on the weighting given to the flow). This means that the objective function of the model (the maximum flow in this case) would have to increase by an integer number of units for every unit by which the constraint were to be relaxed in order for the constraint to be associated with a bottleneck. Thus, if the flow throughout the network were to be given a weighting of 1 and there were a segment of infrastructure preventing more flow, the shadow price on the constraint representing that infrastructure would be 1, as the overall flow could be increased by 1 unit if the constraint would be relaxed by 1 unit. The shadow price may be found on the sensitivity report made available by the NEOS solver.

5.4.1 Scenario One

The first scenario concerns the summer peak week seven (10 – 16 February), during which a total of 40 239 pallets of fruit were exported in 2003. As most of the volume is made up of deciduous and subtropical fruits, one may assume that most of the fruit is exported via the port of Cape Town. Thus, for this scenario, only Cape Town was allowed as an exit point for the fruit. Initially, a general maximum flow solution is presented in this section, in which all fruits are given the same weighting in the objective function (in the sense that all fruit flows accounted equally for the volumes in the maximum flow solution), and the 2003 export volume during week seven (summer peak) of each fruit type is set as a minimum for each of the fruits. A sensitivity analysis then follows where each fruit type is given a greater weighting on the flow in the objective function (the flow of a fruit group is multiplied by 2 in the objective function, instead of by 1 as the others are), to ascertain what effect such an increase has on the maximum flow distribution. This additional study was performed so that it could be determined what maximum volume of a fruit type could be exported in week seven of 2003.

Standard Solution

According to the model built for this scenario, the maximum possible flow through the infrastructure during summer peak week seven is 100 523 pallets of fruit. The associated export volume for each fruit type is shown in Table 5.3, while the maximum volume of exported fruit in the maximum flow solution where week seven of 2003 volumes were set as a minimum flow, are shown in Table 5.4.

Fruit Type	Possible Export Volume (Pallets)					2003 Export	
	FPT (B)	SAFT (B)	FPT (C)	SAFT (C)	Cont. Term.		Total
Grapes	14 286	2 504	11 200	2 016	11 536	41 542	18 653
Stone	523	301	0	0	3 455	4 279	4 398
Pome	60	2 587	0	0	13 746	16 393	15 718
Citrus	16 631	5 108	0	0	9 803	31 542	23
Mango	n/a	n/a	0	1 717	2 721	4 438	1 120
Litchi	n/a	0	0	0	2 329	2 329	327
Total	31 500	10 500	11 200	3 733	43 590	100 523	40 239

Table 5.3: *The maximum possible flow of fruit through the South African fruit export infrastructure during the summer peak week seven, assuming Cape Town harbour is the only point of exit for fruit. Legend: B = Bulk, C = Containerised, n/a = not allowed. Notice that the solution to the maximum flow problem allows less stone fruit to be exported than was exported during week seven of 2003.*

The volume of fruit exported from each production region, as found in the above solution to the maximum flow problem is not interesting, because all bottlenecks occur at the port of Cape Town — in this scenario no bottlenecks occur at a regional packhouse or cold store level. This means that all the fruit that is exported may be allocated to originate from a select few production regions by the algorithm that solves the model. However, one could add constraints to the model to force each region to produce at least the amount of fruit produced in 2003. But this does not increase the maximum flow through the infrastructure as all fruit flows to a single port, and to the maximum flow model the origin of fruit is arbitrary as there is no cost involved in transporting fruit. Nevertheless, constraints were added to force the overall export volumes of the individual fruit groups to be at least those exported in 2003 in a bid to render the solution more realistic.

Fruit Type	Possible Export Volume (Pallets)					Total	2003 Export
	FPT (B)	SAFT (B)	FPT (C)	SAFT (C)	Cont. Term.		
Grapes	2 099	0	0	162	38 050	40 311	18 653
Stone	0	8 015	0	0	0	8 015	4 398
Pome	8 186	0	7 143	631	431	16 391	15 718
Citrus	19 757	2 485	4 057	1 128	5 000	32 427	23
Mango	n/a	n/a	0	1 812	109	1 921	1 120
Litchi	1 458	0	0	0	0	1 458	327
Total	31 500	10 500	11 200	3 733	43 590	100 523	40 239

Table 5.4: *The maximum possible flow of fruit through the South African fruit export infrastructure during the summer peak week seven, assuming Cape Town harbour is the only point of exit for fruit. Legend: B = Bulk, C = Containerised, n/a = not allowed. Fruit flow volumes have been forced to be at least those exported during week seven of 2003.*

The maximum break-bulk loading rates at both the FPT and SAFT terminals are reached for the maximum flow to pass through Cape Town harbour, as are the maximum container loading rates at those terminals. Both the Cape Town stack and the combi-stack at the multi-purpose terminal are filled at the optimal solution, and the Blue Store should turn over all 500 containers a week for the maximum of 100 523 pallets to flow through Cape Town. With the eventual demise of the Blue Store, Cape Town harbour officials must ensure that an equivalent of 125 plug-in points² for 12m containers are added for the terminal not to reduce its capacity of moving 43 590 pallets of fruit a week.

The constraint on the volume of ambient citrus allowed onto the vessels is not a restricting factor in this scenario. Nor is the loading rate of containers from stacks onto vessels, because the stacks are the limiting factors. The gantry cranes could load an additional $45\,840 - 43\,590 = 2\,250$ pallets a week. Assuming the 6m/12m container ratio remains the same for any additional containers of fruit, set the number of 6m containers to be x and the number of 12m containers to be y . Then $10x + 20y = 2\,250$ and $0.572(x + y) = y$, which means that an additional 82 twelve-metre integral containers could be loaded onto container vessels (assuming there are enough container vessels available) during summer peak week seven, along with an additional 61 six-metre containers.

Solution with an additional Weighting on Grapes

The maximum fruit flow does not change when an additional weighting is put on the flow of grapes in the objective function; the flow during summer peak week seven remains 100 523 pallets of fruit. However, a change in the distribution of flow between the fruit types is observed. The additional weighting was put onto the flow of grapes to determine the maximum flow of grapes that could potentially be achieved, assuming that the flow of the other fruits export volumes remain as large as they were in week seven of 2003. The results are listed in Table 5.5.

²As the porthole containers that are stored in the Blue Store are only 6m long and hold 10 pallets of fruit, for every 2 porthole containers that would flow through the Blue Store a single 12m container would have to pass through the stack. This means that 250 additional 12m integral containers would need to be plugged into the stack at some point. The average dwell-time of a container in the stack in Cape Town is approximately 3.5 days, so $250 \times 3.5/7 = 125$ additional plug-in points would have to be added to compensate for the loss of the Blue Store. This would have been achieved with the increase in plug-in points of 450 in February 2004, while a further 550 were, in fact, built in November 2004.

Fruit Type	Possible Export Volume (Pallets)						2003 Export
	FPT (B)	SAFT (B)	FPT (C)	SAFT (C)	Cont. Term.	Total	
Grapes	24 938	5 120	5 176	113	43 590	78 937	18 653
Stone	3 688	395	315	0	0	4 398	4 398
Pome	2 851	4 985	4 462	3 420	0	15 718	15 718
Citrus	23	0	0	0	0	23	23
Mango	n/a	n/a	920	200	0	1 120	1 120
Litchi	n/a	n/a	327	0	0	326	326
Total	31 500	10 500	11 200	3 733	43 590	100 523	40 239

Table 5.5: *The maximum possible flow of fruit through the South African fruit export infrastructure in summer, assuming Cape Town harbour is the only point of exit for fruit and giving the flow of grapes an additional weighting in the objective function (i.e. giving grapes priority in the export process).*

It is clear, from Table 5.5, that the flow volumes of all other fruit types remain as they were during week seven of 2003, except that the flow of Grapes could potentially increase by 60 248 pallets to 78 937 pallets per week. The bottlenecks in this scenario are the break-bulk and container loading rates of both FPT and SAFT terminals in Cape Town; both the container terminal stack and the combi-stack would be filled to capacity and the Blue Store would be filled. The loading rate at the container terminal is not reached, nor is the restriction on the volume of ambient fruit that is allowed in the holds of conventional vessels (only citrus fruit may be loaded at ambient temperatures and only 23 pallets of citrus are exported). No regional packhouses or cold stores reach their capacities; the facilities at the harbour are again the limitations.

Solution with an additional Weighting on Stone Fruit

To determine the maximum flow of stone fruit through the infrastructure during the summer peak week seven, the flow of stone was given a higher weighting in the objective function than the flow of the other fruit groups. The maximum combined flow of all fruit types remained at 100 523 pallets of fruit, with a maximum of 25 007 pallets of stone fruit. The resulting potential export volumes are listed in Table 5.6.

Fruit Type	Possible Export Volume (Pallets)						2003 Export
	FPT (B)	SAFT (B)	FPT (C)	SAFT (C)	Cont. Term.	Total	
Grapes	6 794	7 166	0	0	15 200	29 160	18 653
Stone	6 153	966	0	0	17 888	25 007	4 398
Pome	6 861	2 363	6 520	0	647	16 391	15 718
Citrus	11 692	5	4 680	1 812	9 528	27 717	23
Mango	n/a	n/a	0	1 921	0	1 921	1 120
Litchi	0	0	0	0	327	327	327
Total	31 500	10 500	11 200	3 733	43 590	100 523	40 239

Table 5.6: *The maximum possible flow of fruit through the South African fruit export infrastructure in summer, assuming Cape Town harbour is the only point of exit for fruit and giving the flow of stone fruit an additional weighting in the objective function.*

In this scenario, the harbour facilities are not a restriction on the flow of fruit as they were for grapes. It is clear from Table 5.6 that the flow of grapes and citrus are higher than the export during week seven of 2003, even though the weighting of the flow of stone fruit was higher than that of all the other fruits. Thus, although the break-bulk and container loading rates at the FPT and SAFT terminals are pushed to the maximum (both stacks and the Blue Store in Cape Town are bottlenecks when achieving maximum flow) they restrict the flow of all fruits other than stone fruit. The flow of stone fruit is restricted by packhouses and regional cold stores — not by the harbour facilities (see Table 5.7 for a list of associated bottlenecks).

Region	Bottleneck	Other fruit handled by bottleneck	Restriction (pallets)
Patensie	Packhouse	Citrus	100
Langkloof	Packhouse	Pome	1 932
Karoo	Packhouse	Grapes	109
	Packhouse		257
Klein Karoo	Cold store	Grapes, hard citrus, soft citrus	308
	Cold store	Pome	175
GEVV	Packhouse	Grapes, pome, hard citrus, soft citrus	5 000
	Packhouse	Pome	3 261
	Packhouse	Pome, hard citrus	3 475
	Packhouse	Pome, hard citrus, soft citrus	780
Boland	Packhouse	Pome	1 100
	Packhouse	Pome, hard citrus	699
	Packhouse	Pome, hard citrus, soft citrus	1 037
	Packhouse	Hard citrus	250
Ceres	Packhouse	Pome	215
	Packhouse		2 065
	Packhouse		Pome, hard citrus
Piketberg	Packhouse	Pome, hard citrus	2 000
Citrusdal	Packhouse	Pome, hard citrus	3
	Packhouse	Hard citrus, soft citrus	50
Hartswater	Cold store	Grapes	180
	Cold store	Grapes, hard citrus, soft citrus	68
Johannesburg	Packhouse		310
Rustenburg	Packhouse	Grapes	32
Total			25 007

Table 5.7: The bottlenecks of stone fruit flow for Scenario One. Bottlenecks of the fruit flow of the other fruit types not listed here occurred at the fruit export terminals of the port of Cape Town.

Solution with an additional Weighting on Pome Fruit

In this case, the flow of pome fruit was assigned an additional weighting in the objective function of the maximum flow problem. The overall maximum flow remained 100 523 pallets of fruit, but the flow of pome increased by 23 806 pallets of fruit (see Table 5.8 for a summary of the fruit flow for this scenario), when the flow of pome through the infrastructure was given a higher weighting than the flows of the other fruits. However, the flow of grapes, citrus, mangos and litchis also increased, indicating that the harbour facilities were not the bottleneck on the flow of pome fruit.

Fruit Type	Possible Export Volume (Pallets)					2003 Export	
	FPT (B)	SAFT (B)	FPT (C)	SAFT (C)	Cont. Term.		Total
Grapes	5 073	0	4 991	752	18 452	29 268	18 653
Stone	1 742	0	0	76	2 580	4 398	4 398
Pome	5 282	7 241	4 147	615	2 242	39 524	15 718
Citrus	19 404	3 259	428	1 138	0	24 228	23
Mango	n/a	n/a	1 522	24	0	1 546	1 120
Litchi	n/a	n/a	85	1 128	346	1 559	327
Total	31 500	10 500	11 200	3 733	43 590	100 523	40 239

Table 5.8: The maximum possible flow of fruit through the South African fruit export infrastructure during summer peak week seven, assuming Cape Town harbour is the only point of exit for fruit and giving the flow of pome fruit an additional weighting in the objective function.

The fact that the flow of pome fruit was limited by regional packhouses and cold stores, and not harbour facilities, allowed for the assignment of additional flow to other fruits. The flow of Grapes increased by 10 615 pallets, citrus increased by 24 205 pallets, an additional 426 pallets of mangos could be exported and litchi export increased by 1 232 pallets per week. Only stone fruit did not show an increase in flow. The associated packhouse and cold store bottlenecks are listed in Table 5.9.

Region	Bottleneck	Other fruit handled by bottleneck	Restriction (pallets)
Langkloof	Cold store	Stone	3 340
	Cold store	Stone, hard citrus	2 300
	Cold store	Stone, hard citrus, soft citrus	240
	Cold store		3 307
Klein Karoo	Packhouse	Stone, hard citrus	26
GEVV	Packhouse	Grapes, stone, hard citrus, soft citrus	5 000
	Packhouse	Stone	6 103
	Packhouse	Stone, hard citrus	1 400
	Packhouse	Stone, hard citrus, soft citrus	780
	Packhouse		4 078
Boland	Cold store	Grapes, stone, hard citrus	600
	Cold store	Grapes, stone, hard citrus, soft citrus	1 080
Ceres	Cold store	Grapes, stone, hard citrus, soft citrus	2 529
	Cold store	Grapes, hard citrus, soft citrus	550
	Cold store	Stone	2 185
	Cold store	Stone, hard citrus	120
	Cold store	Stone, soft citrus	60
	Cold store		2 364
	Cold store	Hard citrus	1 000
	Cold store	Hard citrus, soft citrus	150
Piketberg	Packhouse	Stone, hard citrus	2 000
Citrusdal	Packhouse	Stone, hard citrus	12
Free State	Cold Store		300
Total			39 524

Table 5.9: The bottlenecks of pome fruit flow for Scenario One. Bottlenecks of the fruit flow of the other fruit types occurred at fruit export terminals in the port of Cape Town.

Solution with an additional Weighting on Citrus Fruit

With additional weighting on citrus the maximum flow did not change from 100 523 pallets for summer peak week seven. In this scenario only citrus fruit registers an increase in flow, while the others remain at their 2003 week seven levels, indicating that the citrus packhouse and cold store capacities are sufficient to stretch the capacity of Cape Town harbour to the limit. See Table 5.10 for a list of flow volumes for this scenario.

Fruit Type	Possible Export Volume (Pallets)					Total	2003 Export
	FPT (B)	SAFT (B)	FPT (C)	SAFT (C)	Cont. Term.		
Grapes	11 736	0	2012	36	4 869	18 653	18 653
Stone	3 628	0	0	391	379	4 398	4 398
Pome	0	0	5 781	1 122	8 815	15 718	15 718
Citrus	16 136	10 500	1 960	2 185	29 527	60 307	23
Mango	n/a	n/a	1 120	0	0	1 120	1 120
Litchi	n/a	n/a	327	0	0	327	327
Total	31 500	10 500	11 200	3 733	43 590	100 523	40 239

Table 5.10: *The maximum possible flow of fruit through the South African fruit export infrastructure during summer peak week seven, assuming Cape Town harbour is the only point of exit for fruit and giving the flow of citrus fruit an additional weighting in the objective function.*

When exporting such a large volume of fruit through Cape Town harbour, the break-bulk and container loading rates of the FPT and SAFT terminals would be reached and the stacks (main and combi-stack) would be filled to capacity, along with the Blue Store. The loading rate of the container terminal would not be stretched to capacity, nor would the limit on ambient citrus be reached (however, this would be reached if the minimisation of cost were also incorporated, as it is cheaper to export fruit without first cooling it to the correct temperature).

Solution with an additional Weighting on Mangos

Although the possible total maximum export through Cape Town remains 100 523 pallets per week when the flow of mangoes is given an additional weighting, only 6 485 of these pallets are mangos (only 6.45% of the total flow; an increase of 4 564 pallets). In this case there is a larger flow of each fruit type; the flow of grapes increases by 13 456 pallets, the flow of stone fruit could be increased by 5 150 pallets, 6 949 additional pallets of pome fruits could be exported, citrus export could increase by 30 084 pallets and 44 additional pallets of litchis could flow through the export infrastructure. These results are tabulated in Table 5.11.

The bottlenecks associated with maximum mango flow volumes are shown in Table 5.12. The other fruits cause the bottlenecks in the harbour. Due to the increase in fruit flow, the maximum break-bulk and container loading rates of the FPT and SAFT terminals in Cape Town are reached and both stacks and the Blue Store are filled to capacity. The container loading rate need not reach its maximum potential in the container terminal, assuming that other container traffic (not integral containers containing fruit) does not increase dramatically.

Fruit Type	Possible Export Volume (Pallets)						2003 Export
	FPT (B)	SAFT (B)	FPT (C)	SAFT (C)	Cont. Term.	Total	
Grapes	18 023	1 252	4 031	2 812	5 992	32 110	18 653
Stone	0	1 100	0	0	7 011	8 111	4 398
Pome	0	1 959	1 220	12	20 149	23 340	15 718
Citrus	13 477	6 189	5 949	125	4 367	30 107	23
Mango	n/a	n/a	0	414	6 071	6 485	1 120
Litchi	0	0	0	370	0	370	327
Total	31 500	10 500	11 200	3 733	43 590	100 523	40 239

Table 5.11: The maximum possible flow of fruit through the South African fruit export infrastructure in summer peak week seven, assuming Cape Town harbour is the only point of exit for fruit and giving the flow of mangos an additional weighting in the objective function.

Region	Bottleneck	Other fruit handled by bottleneck	Restriction (pallets)
Levubu	Cold store		200
Tzaneen	Cold store	Hard citrus	580
	Cold store	Avocado	564
	Cold store	Avocado, litchi	342
	Cold store		440
Hoedspruit	Packhouse	Hard citrus, soft citrus	650
	Packhouse	Hard citrus	626
	Packhouse		625
Kiepersol	Packhouse	Avocado	540
	Packhouse	Hard citrus, litchi	150
Nelspruit	Cold store	Hard citrus, avocado, litchi	160
	Cold store	Hard citrus, litchi	96
Malelane	Cold store	Hard citrus	350
	Cold store	Hard citrus, litchi	600
	Cold store		410
	Cold store	Litchi	152
Total			6 485

Table 5.12: The bottlenecks of the mango flow for Scenario One. Bottlenecks of the fruit flow of the other fruit types occur at fruit export terminals in the port of Cape Town.

Solution with an additional Weighting on Litchis

The solution in the case where litchis are given preference is similar to the solution found when considering mango flow. The flow of litchis could only increase by 2 158 pallets per week as the regional packhouses and cold stores limit the flow of litchis to the ports. Due to this fact, the flows of the other fruits are allowed to increase until limited by the fruit export infrastructure in Cape Town harbour. See Table 5.13 for a list of fruit flows through the export infrastructure in this case.

The same bottlenecks occur in the harbour for this scenario as they do for all others. The break-bulk and container loading rates of the SAFT and FPT terminals in Cape Town need to operate at maximum capacity to achieve these volumes of flow. Both stacks would need to be filled, as would the Blue Store. However, the container loading capacity would not pose a problem to the flow of fruit if the volumes of other container flow remain constant.

Fruit Type	Possible Export Volume (Pallets)					2003 Export	
	FPT (B)	SAFT (B)	FPT (C)	SAFT (C)	Cont. Term.		Total
Grapes	3 345	7 241	2 612	880	34 539	48 617	18 653
Stone	925	0	1 413	1 949	112	4 398	4 398
Pome	12 769	0	2 840	110	0	15 718	15 718
Citrus	14 462	3 259	1 583	795	7 797	27 895	23
Mango	n/a	n/a	870	0	540	1 410	1 120
Litchi	n/a	n/a	1 883	0	602	2 485	327
Total	31 500	10 500	11 200	3 733	43 590	100 523	40 239

Table 5.13: *The maximum possible flow of fruit through the South African fruit export infrastructure during summer peak week seven, assuming Cape Town harbour is the only point of exit for fruit and giving the flow of litchis an additional weighting in the objective function.*

Region	Bottleneck	Other fruit handled by bottleneck	Restriction (pallets)
Levubu	Packhouse	Avocado	346
Tzaneen	Packhouse	Avocado	220
	Packhouse	Avocado, mango	740
Kiepersol	Packhouse	Hard citrus, mango	10
	Packhouse	Avocado	75
Nelspruit	Cold store	Hard citrus, avocado, mango	160
	Cold store	Hard citrus, mango	96
	Cold store		72
Malelane	Packhouse	Mango	152
	Packhouse	Hard citrus, mango	454
	Packhouse		160
Total			2 485

Table 5.14: *The bottlenecks of the litchi flow for Scenario One. Bottlenecks of the fruit flow of the other fruit types occur at the fruit export terminals.*

5.4.2 Scenario Two

The second scenario concerns the winter peak during week 29 (14 – 20 July 2003). During this week, only pome, hard citrus, soft citrus and avocados were exported. For this scenario, the maximum flow through the national fruit export infrastructure is found, when considering only these four fruit groups.

The solution found for the maximum flow problem indicates a total maximum possible flow of 150 004 pallets of fruit through the national infrastructure during that week. The resulting flow of each fruit type is listed in Table 5.15. If one compares the figures in this table to the export volumes of the four fruit types in 2003, it is clear that the maximum flow solution allows significantly more of each fruit type through the export infrastructure than was exported during 2003.

To achieve the maximum flow listed in Table 5.15, the regions would have to produce the volumes of fruit shown in Table 5.16. The estimated fruit volumes exported from each region, during week 29 of 2003, were set as the minimum export volume from each region for each fruit type. However, there were few regions where the 2003 export volume was larger than the capacity of the packhouse and/or cold store infrastructure in that region.

Fruit Type	Possible Export Volume (Pallets)					2003 Export
	Cape Town	Durban	Port Elizabeth	Maputo	Total	
Pome	16 577	13 649	3 214	240	33 680	8 581
Hard Citrus	39 525	15 612	24 598	22 615	102 351	70 043
Soft Citrus	4 985	1 360	3 201	0	9 546	2 109
Avocado	786	25	3 082	534	4 427	1 989
Total	61 874	30 646	34 095	23 389	150 004	82 722

Table 5.15: *The maximum flow of each individual fruit type in Scenario Two, compared to the actual flow of fruit during the 2003 season.*

There are four regions where there appears to be a smaller volume that can be exported than was actually exported during 2003. The first such region is the Malelane citrus growing region, which incorporates the Kiepersol, Nelspruit and Malelane packhouse/cold store regions. According to the information available, a maximum of 10 538 pallets of citrus could be packed there for export in a week (only 10 292 pallets passed through the packhouses in this region in the solution to this maximum flow problem as the minimum citrus production constraints had to be relaxed; the packhouse capacities were insufficient to also cater for the avocado production). However, according to the export information available, 12 170 pallets of citrus fruit were exported from this region during week 29 of 2003. Another problem is found in the Eastern Cape Midlands, where a maximum of 645 pallets of pome fruit should be packed and cooled for export according to the maximum flow solution, but farmers managed to export an estimated 1 459 pallets of fruit from the region during week 29 of 2003. A possible reason for these discrepancies may be due to the long time that pome fruits may be kept in storage in controlled atmosphere cold rooms before they need to be exported. Another reason may be inaccurate or insufficient data regarding packhouses in the region. There is a similar problem in the results for the Klein Karoo region, where the maximum packing rate is less than what was estimated to actually have been exported from the region during week 29 of 2003. The final problem occurs in the Orange River region around Upington. A maximum of 135 pallets of citrus fruit can be packed in this region according to the maximum flow solution, but the region managed to export 272 pallets during week 29 of 2003. Besides the fact that some of these fruits may have been kept in storage before being exported, another explanation can be given for the discrepancies between what should be a maximum export volume and the higher true export values. During the information gathering process not all packhouses or cold stores could be contacted successfully and thus their information was not included in the constraints of the model.

Of the packhouses in these regions, most would have to operate at full capacity to achieve the maximum fruit flows shown in Table 5.15. In the Levubu region the packhouses for hard citrus, soft citrus and avocado are stretched to the limit to achieve a total export of 11 250 pallets of fruit per week. In Tzaneen only the packhouses packing hard citrus are bottlenecks, while there is space for an additional 226 pallets of avocados to be packed per week. The total export from Tzaneen may be as high as 16 341 pallets of fruit per week. In Hoedspruit (total export can be 3 181 pallets of hard citrus and soft citrus), Kiepersol (total export can be 1 924 pallets of hard citrus and avocados), Nelspruit (total export potential of 4 273 pallets of hard citrus and avocado), Malelane (can export up to 5 814 pallets of hard citrus), Potgietersrus (2 365 pallets of hard citrus at maximum), Marble Hall (at most 9 615 pallets of hard citrus and soft citrus), Pongola (1 454 pallets of hard citrus at maximum), Nkwalini (7 594 pallets of hard citrus, soft citrus and avocado at maximum), KwaZulu-Natal (1 294 pallets of hard citrus, soft citrus and avocado at maximum), Eastern Cape Midlands (688 pallets of hard citrus and soft citrus at maximum), Sundays River Valley (10 351 pallets of hard citrus and soft citrus at maximum)

Region	Fruit Type	Maximum Production (pallets)	2003 Export Production (pallets)
Levubu	Hard citrus	10 637	6 985
	Soft citrus	27	27
	Avocado	610	72
Tzaneen	Hard citrus	14 555	13 240
	Avocado	1 786	760
Hoedspruit	Hard citrus	3 024	3 087
	Soft citrus	157	24
Kiepersol	Hard citrus	721	12 462 (Ma)
	Avocado	1 203	487
Nelspruit	Pome	0	Unknown
	Hard citrus	3 757	12 462 (Ma)
	Soft citrus	0	0
	Avocado	520	520
Malelane	Pome	0	Unknown
	Hard citrus	5 814	12 462 (Ma)
Potgietersrus	Hard citrus	2 365	5 698 (Se)
Marble Hall	Hard citrus	8 515	5 698 (Se)
	Soft citrus	850	322
Ermelo	Pome	0	Unknown
Pongola/Nkwalini/KZN	Hard citrus	9 758	8 330
	Soft citrus	84	84
	Avocado	312	150
Eastern Cape Midlands	Hard citrus	645	1 494
	Soft citrus	43	43
Sundays River Valley	Hard citrus	7 351	6 564
	Soft citrus	3 000	406
Patensie	Hard citrus	6 149	4 185
	Soft citrus	410	133
Langkloof	Pome	9 187	1 422
	Hard citrus	928	12
Klein Karoo	Pome	26	167
	Hard citrus	2 237	7 707 (WC)
	Soft citrus	308	1 070 (WC)
GEVV	Pome	13 761	4 479
	Hard citrus	700	7 707 (WC)
	Soft citrus	3 600	1 070 (WC)
Boland	Pome	1 298	182
	Hard citrus	3 412	7 707 (WC)
	Soft citrus	792	1 070 (WC)
Ceres	Pome	8 958	2 127
	Hard citrus	2 246	7 707 (WC)
	Soft citrus	100	1 070 (WC)
Piketberg	Pome	150	163
	Hard citrus	2 278	7 707 (WC)
Citrusdal	Pome	0	Unknown
	Hard citrus	6 228	7 707 (WC)
	Soft citrus	200	1 070 (WC)
Orange River	Hard citrus	135	279
	Soft citrus	0	0
Hartswater	Hard citrus	1 070	0
	Soft citrus	0	0
Rustenburg	Hard citrus	26	5 698 (Se)
Free State	Pome	300	41
Middelburg	Hard citrus	9 800	5 698 (Se)

Table 5.16: Largest possible export volumes of each fruit type (pome fruit, hard citrus, soft citrus and avocados) in Scenario 2 for each region. The figures in bold face in the right-most column correspond to actual production that exceeded the production required for maximum flow to occur. The characters in parentheses indicate regions for which the production data are common to many smaller regions (Ma – Malelane, Se – Senwes, WC – Western Cape).

and Patensie (6 559 pallets of hard citrus and soft citrus at maximum) the packhouses are bottlenecks. In the Langkloof region (10 115 pallets of pome and hard citrus at maximum) only the hard citrus packhouses are bottlenecks, while in the GEVV region all packhouses are bottlenecks and restrict the maximum export from this region to 18 061 pallets of pome, hard citrus and soft citrus. In the Boland region the packhouses are bottlenecks, but an export volume of 5 502 pallets of pome, hard citrus and soft citrus may be achieved. In Ceres (11 304 pallets of pome, hard citrus and soft citrus at maximum) only the restriction on hard citrus and soft citrus packing are bottlenecks, while in Piketberg (2 428 pallets of pome and hard citrus at maximum), Citrusdal (6 428 pallets of pome, hard citrus and soft citrus at maximum), the Orange River (135 pallets of hard citrus and soft citrus at maximum), Hartswater (1 070 pallets of hard citrus and soft citrus at maximum), Rustenburg (26 pallets of hard citrus at maximum) and Middelburg (9 800 pallets of hard citrus per week at maximum) the packhouses are bottlenecks. In the Free State the packing facilities are not restrictive, allowing 300 pallets of pome to be exported.

In Levubu the cold stores have a larger handling capacity of avocados per week than the packhouses — hence they are not bottlenecks. In Tzaneen the cold stores handling avocados restrict the flow of avocados by 226 pallets per week and are thus the bottlenecks, while the citrus cold stores in the region present no problem. In the Hoedspruit, Kiepersol, Nelspruit, Malelane, Marble Hall, KwaZulu-Natal, Eastern Cape, Sundays River Valley and Patensie regions the cold stores do not restrict the volume of fruit that could potentially be exported; the packhouses are the bottlenecks there. The avocado cold stores are bottlenecks in the Nkwalini region. In the Langkloof region the cold stores are bottlenecks, allowing a total of 9 078 pallets of fruit to pass through per week, while the packhouses allow up to 11 748 pallets of fruit to be packed per week (a difference of 2 561 pallets per week). In the Klein Karoo and GEVV regions the cold stores are not restrictive to the flow of fruit. In the Boland region, the flow of pome fruit to Cape Town harbour is hindered by the capacity of the cold stores. Up to 4 349 pallets of pome fruit could potentially be packed per week by the packhouses in the region, but infrastructure to cool only 1 680 pallets per week exists (a difference of 2 669 pallets per week). In Ceres, the lack of cold stores restricts the packhouse from allowing up to 6 158 pallets per week of additional pome fruit to be exported from the region, while the De Doorns, Piketberg, Citrusdal, Orange River, Hartswater and Johannesburg regions have sufficient cold storage. The Free State lacks cold storage for its pome as only 300 pallets per week can be cooled to the required temperature in that region, while 783 pallets could be packed per week (difference of 483 pallets per week). This model was created with the necessary nodes and arcs required to model the option of sending fruit to the destination markets by airfreight in case they were to be added at a later stage. However, the capacities of the airports were set to zero as airfreight is an option rarely used, especially during week 29 of the season (it is more common in the summer months). Although Matola cold store exists in Maputo and is used by citrus exporters, it has also been omitted from the model by means of the incorporation of a zero capacity, as its node capacity could not be obtained. A sensitivity analysis of the node reveals whether its inclusion is a necessity or not: it turns out that increasing the capacity of the cold store does not increase the maximum fruit flow out of South Africa. However, its importance may increase when the minimum cost flow analysis is considered. The FPT cold store in the Maputo harbour is used to 1 067 pallets per week short of its maximum capacity in order to achieve the maximum flow of pallets, and the break-bulk loading rate at the FPT terminal is used to full capacity. In Durban only the stack and the container loading rate at the SAFT terminal are required to operate to their limits. The fruit transported to Port Elizabeth fills up the non-FPT cold stores, and the FPT break-bulk loading rate and the stack are required to operate at 100% of their peak capacity. In Cape Town, all cold stores, the Blue Store, the break-bulk and container loading

Region	Bottleneck	Fruit types handled	Capacity (pallets)
Levubu	Packhouse	Hard citrus	10 256
	Packhouse	Hard citrus, soft citrus	384
	Packhouse	Avocado	284
	Packhouse	Avocado, litchi	326
Tzaneen	Packhouse	Hard citrus	13 015
	Packhouse	Hard citrus, mango	1 540
	Cold store	Avocado	140
	Cold store	Avocado, mango	564
	Cold store	Avocado, mango, litchi	342
Hoedspruit	Cold store	Avocado, litchi	740
	Packhouse	Hard citrus	1 655
	Packhouse	Hard citrus, soft citrus	369
	Packhouse	Hard citrus, soft citrus, mango	650
	Packhouse	Hard citrus, mango	357
Kiepersol	Packhouse	Hard citrus, mango, litchi	150
	Packhouse	Hard citrus	171
	Packhouse	Hard citrus, mango, litchi	550
	Packhouse	Avocado	436
	Packhouse	Avocado, mango	540
Nelspruit	Packhouse	Avocado, litchi	227
	Packhouse	Pome, hard citrus	240 of 300
	Packhouse	Hard citrus	1 245
	Packhouse	Hard citrus, soft citrus, avocado	230
	Packhouse	Hard citrus, avocado, mango, litchi	2 142
Malelane	Packhouse	Avocado	316
	Packhouse	Avocado, mango, litchi	100
	Packhouse	Pome, hard citrus	214 of 306
	Packhouse	Hard citrus	3 150
Potgietersrus	Packhouse	Hard citrus, mango	1 950
	Packhouse	Hard citrus, mango, litchi	500
	Packhouse	Grapes, hard citrus	60
	Packhouse	Hard citrus	2 305
Marble Hall	Packhouse	Grapes, hard citrus, soft citrus	600
	Packhouse	Hard citrus	5 965
	Packhouse	Hard citrus, soft citrus	2 800
Pongola	Packhouse	Hard citrus	1 454
Nkwalini	Packhouse	Hard citrus	7 108
	Packhouse	Hard citrus, soft citrus	272
	Cold store	Avocado	25
KZN	Packhouse	Hard citrus	160
	Packhouse	Hard citrus, soft citrus	535
	Packhouse	Hard citrus, avocado	312
	Packhouse	Avocado	287
Eastern Cape Midlands	Packhouse	Hard citrus	228
	Packhouse	Hard citrus, soft citrus	460
Sundays River Valley	Packhouse	Hard citrus	2 771
	Packhouse	Hard citrus, soft citrus	7 580
Patensie	Packhouse	Stone, citrus	100
	Packhouse	Hard citrus	5 066
	Packhouse	Hard citrus, soft citrus	1 393
Langkloof	Packhouse	Pome, hard citrus	928
	Cold store	Stone, pome	3 340
	Cold store	Stone, pome, hard citrus	2 300
	Cold store	Stone, pome, hard citrus, soft citrus	240
	Cold store	Pome	3 307
Piketberg	Packhouse	Stone, pome, hard citrus	2 000
	Packhouse	Hard citrus	428

Table 5.17: Bottlenecks in the South African fruit export infrastructure for Scenario 2. Fruit types in bold face indicate that the capacity is of that specific fruit; the packhouse/cold store group may be able to pack/cool more of the other fruits, but the bottleneck is limited to the fruit in bold face.

Region	Bottleneck	Fruit types handled	Capacity (pallets)
Klein Karoo	Packhouse	Stone, pome, hard citrus	26
	Packhouse	Stone, hard citrus	125
	Packhouse	Stone, hard citrus, soft citrus	2 420
GEVV	Packhouse	Grapes, stone, pome, hard citrus, soft citrus	5 000
	Packhouse	Stone, pome	6 103
	Packhouse	Stone, pome, hard citrus	2 100
	Packhouse	Stone, pome, hard citrus, soft citrus	780
	Packhouse	Pome	4 078
Ceres	Packhouse	Stone, pome, hard citrus	1 282
	Packhouse	Pome, hard citrus, soft citrus	100
	Packhouse	Hard citrus	964
	Cold store	Grapes, stone, pome, hard citrus, soft citrus	2 529
	Cold store	Grapes, Pome, hard citrus, soft citrus	550
	Cold store	Stone, pome	2 185
	Cold store	Stone, pome, hard citrus	120
	Cold store	Stone, pome , soft citrus	60
	Cold store	Pome	2 364
	Cold store	Pome, hard citrus	1 000
	Cold store	Pome, hard citrus, soft citrus	150
Boland	Packhouse	Stone, pome	700
	Packhouse	Stone, pome, hard citrus	1 221
	Packhouse	Stone, pome, hard citrus, soft citrus	2 028
	Packhouse	Stone, hard citrus	339
	Packhouse	Hard citrus	1 214
	Cold store	Grapes, stone, pome, hard citrus	600
	Cold store	Grapes, stone, pome, hard citrus, soft citrus	1 080
Citrusdal	Packhouse	Stone, pome, hard citrus	129
	Packhouse	Stone, hard citrus, soft citrus	350
	Packhouse	Hard citrus	5 949
Orange River	Packhouse	Grapes, hard citrus	25
	Packhouse	Grapes, hard citrus, soft citrus	110
Hartswater	Packhouse	Hard citrus	700
	Packhouse	Hard citrus, soft citrus	370
Rustenburg	Packhouse	Hard citrus	26
Free State	Cold store	Pome	300
Middelburg	Packhouse	Hard citrus	9 800

Table 5.17 (continued): *Bottlenecks in the South African fruit export infrastructure for Scenario 2. Fruit types in bold face indicate that the capacity is of that specific fruit; the packhouse/cold store group may be able to pack/cool more of the other fruits, but the bottleneck is limited to the fruit in bold face.*

rates of the FPT terminal and the container loading rate of the SAFT terminal are required to operate at peak efficiency.

A total of 47 571 pallets of fruit should be exported per week from South Africa via containers to achieve maximum throughput, and the remaining 102 433 pallets per week should be exported in break-bulk. It is important to realise that the values presented above for the maximum flow results are not the only solution, as there exist many other solutions that will yield the same maximum flow; only one solution is shown and discussed in this section. See Tables 5.18 and 5.19 for a break-down of the sources of fruit to each cold store and terminal in the harbours that correspond to the maximum flow solution.

Although the binding constraints seen in the *Total* rows of Tables 5.18 and 5.19 seem to be bottlenecks at a first glance, they are, in fact, not. This is often the case in optimisation problems where there are multiple options (such as harbour cold stores or terminals in this case) and the coefficients of the variables in the objective function are all the same (unity, in this maximum flow problem) leading to non-uniqueness of optimal solutions. Thus, if one

Destination	Source	Transport Type	Fruit Type	Volume (pallets)
Maputo FPT Cold Store	Hoedspruit	Un	Hard citrus	2 033
	Total			2 033 of 3 100
Maputo FPT Bulk Loading	Levubu	Un	Hard citrus	10 232
	Malelane	Un	Hard citrus	5 314
	Nkwalini	Un	Hard citrus	18
	Sundays River Valley	Un	Hard citrus	1 548
	Langkloof	Un	Hard citrus	928
	Piketberg	Un	Hard citrus	1 850
	Citrusdal	Un	Hard citrus	664
	Hoedspruit	Re	Soft citrus	24
	MfptCs		Hard citrus	2 033
	Total			22 611 of 23 184
Durban Cold Stores	Klein Karoo	Re	Pome	26
	Ceres	Re	Pome	5 429
	Ceres	Re	Soft citrus	100
	GEVV	In	Pome	964
	Boland	In	Pome	698
	Boland	In	Soft citrus	792
	Ceres	In	Pome	3 529
	Piketberg	In	Pome	163
	Total			11 698 of 20 104
Durban FPT Cold Store	Levubu	Un	Hard citrus	381
	Patensie	Un	Hard citrus	6 149
	Ceres	Un	Hard citrus	2 246
	Citrusdal	Re	Hard citrus	300
	Total			9 076 of 17 520
Durban FPT Bulk Loading	Tzaneen	Un	Hard citrus	4 084
	Potgietersrus	Un	Hard citrus	60
	Sundays River Valley	Un	Hard citrus	1 744
	DfptCs		Hard citrus	9 076
	Total			14 964 of 45 320
Durban SAFT Bulk Loading	Tzaneen	Un	Hard citrus	176
	Tzaneen	Re	Hard citrus	392
	Total			568 of 13 140
Port Elizabeth Cold Store	Sundays River Valley	Un	Hard citrus	4 059
	Levubu	Re	Avocado	610
	Kiepersol	Re	Avocado	1 011
	Nelspruit	Re	Hard citrus	230
	Boland	Re	Hard citrus	150
	Total			6 060 of 6 060
Port Elizabeth FPT Cold Store	Patensie	Un	Soft citrus	410
	Orange River	Un	Hard citrus	135
	Total			545 of 6 760
Port Elizabeth FPT Bulk Loading	Hartswater	Un	Hard citrus	876
	Tzaneen	Un	Hard citrus	1 540
	Marble Hall	Un	Hard citrus	8 303
	Pongola	Un	Hard citrus	1 454
	KwaZulu-Natal	Un	Hard citrus	1 006
	Eastern Cape Midlands	Un	Hard citrus	645
	Klein Karoo	Un	Hard citrus	2 237
	GEVV	Un	Hard citrus	700
	Boland	Un	Hard citrus	3 262
	GEVV	Re	Soft citrus	2 592
		Total		
Cape Town Cold Stores	Nelspruit	Un	Hard citrus	3 523
	Malelane	Un	Hard citrus	500
	GEVV	Un	Soft citrus	1 008
	Rustenburg	Un	Hard citrus	26
	Citrusdal	Un	Hard citrus	5 264
	KwaZulu-Natal	Re	Soft citrus	84
	Sundays River Valley	Re	Soft citrus	3 000
	Total			13 405 of 13 405
Cape Town FPT Cold Store	Kiepersol	Un	Hard citrus	721
	Nkwalini	Un	Hard citrus	7 279
	Total			8 000 of 8 000
Cape Town FPT Bulk Loading	Tzaneen	Un	Hard citrus	8 283
	Marble Hall	Un	Hard citrus	12
	Middelburg	Un	Hard citrus	4 800
	Marble Hall	Re	Hard citrus	200
	Marble Hall	Re	Soft citrus	850
	Eastern Cape Midlands	Re	Soft citrus	43
	CTCs		Hard citrus	9 313
CTfptCs		Hard citrus	8 000	
	Total			31 500 of 31 500
Cape Town SAFT Bulk Loading	Hoedspruit	Un	Hard citrus	851
	Potgietersrus	Un	Hard citrus	2 305
	Langkloof	Re	Pome	2 300
	Piketberg	Re	Hard citrus	415
	Hartswater	Re	Hard citrus	68
	Johannesburg	Re	Hard citrus	126
	CTCs		Soft citrus	4 092
	Total			10 158 of 10 500

Table 5.18: Solution of the maximum flow problem at the point where the transportation reaches the ports and port cities, where break-bulk fruit is handled for Scenario 2. The transport type refers to whether transportation is unrefrigerated (Un), refrigerated (Re) or in integral containers (In).

Destination	Source	Fruit Type	Volume (pallets)
Maputo FPT Container Loading	Kiepersol	Avocado	192
	Tzaneen	Avocado	342
	Patensie	Pome	240
	Total		774 of 3 877
Durban Stack	Hoedspruit	Soft citrus	99
	Klein Karoo	Soft citrus	308
	Boland	Avocado	25
	Boland	Pome	600
	DIn	Pome	10 796
	DIn	Soft citrus	892
Total		12 720 of 12 720	
Durban FPT Container Loading	Tzaneen	Hard citrus	80
	Hoedspruit	Soft citrus	61
	Total		141 of 7 769
Durban SAFT Container Loading	GEVV	Pome	2 253
	Total		2 253 of 2 253
Port Elizabeth Stack	Tzaneen	Avocado	1 174
	KwaZulu-Natal	Avocado	287
	Langkloof	Pome	2 914
	Free State	Pome	300
	Piketberg	Soft citrus	200
	PEIn	Hard citrus	4 439
	PEIn	Avocado	1 621
	PEfptIn	Hard citrus	135
	PEfptIn	Soft citrus	410
	Total		11 480 of 11 480
Port Elizabeth FPT Container Loading	Total		0 of 3 877
Cape Town Stack	Total		0 of 31 390
Cape Town Combi Stack	Tzaneen	Avocado	270
	Total		270 of 7 200
Cape Town FPT Container Loading	Hoedspruit	Hard citrus	136
	Nelspruit	Avocado	520
	GEVV	Pome	10 544
	Total		11 200 of 11 200
Cape Town SAFT Container Loading	Langkloof	Pome	3 733
	Total		3 733 of 3 733
Blue Store	Middelburg	Hard citrus	5 000
	Total		5 000 of 5 000
CTCT Container Loading	BS	Hard citrus	5 000
	CTcomS	Avocado	270
	Total		5 270 of 45 840
DCT Container Loading	DS	Pome	11 396
	DS	Soft citrus	1 299
	DS	Avocado	25
	Total		12 720 of 103 940
PECT Container Loading	PES	Pome	3 214
	PES	Hard citrus	4 574
	PES	Soft citrus	610
	PES	Avocado	3 082
	Total		11 480 of 50 540

Table 5.19: Solution of the maximum flow problem at the point where the transportation reaches the ports and port cities, where containerised fruit is handled. All fruit arriving at ports directly from the production regions would do so in containers, otherwise it would first pass through harbour cold stores.

increases the capacity of an element in the infrastructure which is not a bottleneck, but has a binding constraint (such as the container stack in Port Elizabeth), flow from another element of the infrastructure could be diverted in that direction. This means that the flow through the first part of the infrastructure will increase, but the overall flow of the system remains the same as the flow through the other part will simply decrease.

5.4.3 Scenario Three

The third scenario concerns the same week as the second scenario (see §5.4.2), except that, since most citrus fruit is exported through Durban harbour, the volume of fruit that passes through Durban is given a larger weighting in the objective function than the fruit that is sent through the other harbours. This is done to simulate the large volumes of fruit that are exported through Durban and the subsequent pressure applied on the local fruit export infrastructure.

The solution found for the maximum flow problem, where an additional weighting was placed on fruit flow to Durban, indicates that the total maximum possible flow remains the same when the citrus flow out of Durban is given a weighting of 2 compared to the original weighting of 1 that is assigned to all fruit flow from the other ports in the objective function. As in the previous scenario, the maximum flow solution indicates that significantly more of each fruit type could be exported through the export infrastructure than was required in 2003.

Fruit Type	Possible Export Volume (Pallets)					2003 Export
	Cape Town	Durban	Port Elizabeth	Maputo	Total	
Pome	4 425	0	7 187	18 870	30 482	8 581
H. Citrus	18 019	81 202	6 760	964	106 945	70 043
S. Citrus	4 719	0	410	3 021	8 150	2 109
Avocado	2 781	0	906	740	4 427	1 989
Total	29 944	81 202	15 263	23 595	150 004	82 722

Table 5.20: *The resultant maximum flow of each individual fruit type in Scenario 3, compared to the actual flow of fruit during the 2003 season.*

To achieve the maximum flow listed in Table 5.20, the regions would have to produce the volumes of fruit shown in Table 5.21. As the estimated 2003 export from each region was set as the minimum export from each region for each fruit type, there are not many regions where there is a larger 2003 export volume than that found in the solution to Scenario Three. There are, however, four such regions where there appears to be a smaller volume that can be exported than was actually exported in 2003. The possible reasons for these discrepancies were discussed in §5.4.2 and they also apply to Scenario Three.

Due to the larger weight placed on citrus flow out of the port of Durban, the export volumes from some regions change with respect to those in Scenario Two. The first region in which the flow changes is the Hoedspruit region. Instead of 3 024 pallets of hard citrus and 157 pallets of soft citrus being exported per week from the region, the hard citrus pallet flow from the region increases by 133 pallets per week and the soft citrus export reduces by the same amount. The soft citrus export from the Marble Hall region reduces from 850 to 319 pallets per week, and the hard citrus export increases by the same amount. For Scenario Three, the hard citrus export from Sundays River Valley increases by 732 pallets per week, which means that the soft citrus export from the region falls by the same volume. In the GEVV region the pome export volume reduces by 3 580 pallets per week, while the hard citrus export volume increases by 2 634 pallets per week and the soft citrus export volume increases by 946 pallets per week. Soft citrus experiences a decrease in export from Boland, so that an increase of 364 pallets in hard citrus exports may occur per week, along with an increase of 382 pallets per week of pome pallets from the region. In Citrusdal the infrastructure used to pack the 200 pallets per week of soft citrus in Scenario 2 are used to pack 200 more pallets of hard citrus in Scenario 3. In all regions not mentioned, the fruit export volumes remain as in Scenario 2.

The same rules regarding airfreight and the Matola cold store were applied to the model for Scenario Three as for Scenario Two. With the additional weighting given to the export of fruit through Durban harbour, it is possible to simulate the peak fruit traffic that Durban harbour experiences during the citrus peak in the winter months. With such a model one should be able to determine the capacity of Durban, assuming that there are no unexpected delays at any of the key infrastructure components.

Region	Fruit Type	Maximum Production (pallets)	2003 Export Production (pallets)
Levubu	Hard citrus	10 637	6 985
	Soft citrus	3	27
	Avocado	610	72
Tzaneen	Hard citrus	14 555	13 240
	Avocado	1 786	760
Hoedspruit	Hard citrus	3 157	3 087
	Soft citrus	24	24
Kiepersol	Hard citrus	721	12 462 (Ma)
	Avocado	1 203	487
Nelspruit	Pome	0	Unknown
	Hard citrus	3 757	12 462 (Ma)
	Soft citrus	0	0
	Avocado	516	520
Malelane	Pome	0	Unknown
	Hard citrus	5 814	12 462 (Ma)
Potgietersrus	Hard citrus	2 365	5 698 (Se)
Marble Hall	Hard citrus	9 046	5 698 (Se)
	Soft citrus	319	322
Ermelo	Pome	0	Unknown
Pongola/Nkwalini/KZN	Hard citrus	9 758	8 330
	Soft citrus	83	84
	Avocado	312	150
Eastern Cape Midlands	Hard citrus	645	1 494
	Soft citrus	43	43
Sundays River Valley	Hard citrus	8 083	6 564
	Soft citrus	2 268	406
Patensie	Hard citrus	6 149	4 185
	Soft citrus	410	133
Langkloof	Pome	9 187	1 422
	Hard citrus	928	12
Klein Karoo	Pome	26	167
	Hard citrus	2 237	7 707 (WC)
	Soft citrus	308	1 070 (WC)
GEVV	Pome	10 181	4 479
	Hard citrus	3 334	7 707 (WC)
	Soft citrus	4 546	1 070 (WC)
Boland	Pome	1 680	182
	Hard citrus	3 776	7 707 (WC)
	Soft citrus	46	1 070 (WC)
Ceres	Pome	8 958	2 127
	Hard citrus	2 246	7 707 (WC)
	Soft citrus	100	1 070 (WC)
Piketberg	Pome	150	163
	Hard citrus	2 278	7 707 (WC)
Citrusdal	Pome	0	Unknown
	Hard citrus	6 428	7 707 (WC)
	Soft citrus	0	1 070 (WC)
Orange River	Hard citrus	135	279
	Soft citrus	0	0
Hartswater	Hard citrus	1 070	0
	Soft citrus	0	0
Rustenburg	Hard citrus	26	5 698 (Se)
Free State	Pome	300	41
Middelburg	Hard citrus	9 800	5 698 (Se)

Table 5.21: The production of each fruit type for each region for the maximum flow to pass through the fruit export infrastructure of South Africa for Scenario 3. The figures in bold in the right-most column correspond to actual production that exceeded the production required for maximum flow to occur. The characters in parentheses indicate regions for which the production data is common to many smaller regions (Ma - Malelane, Se - Senwes, WC - Western Cape).

All fruit export infrastructure in Durban is used to capacity in the maximum flow solution of this scenario, except the container loading rate of the container terminal. However, only five of these infrastructure components are, in fact, bottlenecks. Increases to the break-bulk and container loading rates of the FPT and SAFT terminals would increase the objective function value, as would an increase to the size of the container stack at the container terminal. The loading rate at the FPT terminal is the only infrastructure in Port Elizabeth pushed to its maximum capacity. In Cape Town the cold stores, the blue store and both loading rates at the SAFT terminal have no slack. All other fruit export infrastructure at the four ports do not need to operate at maximum capacity (see Tables 5.22 and 5.23 for more detail).

In this scenario, some of the harbour infrastructure components are bottlenecks, unlike in Scenario Two (where some components had reached a maximum, but increasing that maximum did not necessarily increase the maximum flow). In this scenario the flow through Durban harbour has been assigned an additional weighting, thus if one diverted fruit from Durban harbour to another, the overall maximum flow would remain the same as before, but the objective function value would decrease. In Tables 5.22 and 5.23 the infrastructure components that are bottlenecks have the totals flowing through them in bold face. Although it would seem that the cold stores in and around Durban are bottlenecks (they are used to full capacity), they are in fact not, as refrigerated vehicles could deliver refrigerated fruit directly from the regional cold stores (in the growing regions) to the berths at FPT or SAFT for direct loading onto vessels, thereby bypassing the cold stores in Durban. However, the disadvantage of this option is that the travelling time of vehicles or trains are variable due to long travelling distances from some regions to the ports. As the port facilities do not always have buffer facilities for fruit arriving from these areas, and the fruit must be loaded onto the vessels in a certain order, delays are likely to occur, resulting in a decrease in productivity in the ports.

5.5 Minimum Cost Flow Model

The minimum cost flow model is very similar to the maximum flow model (5.2). However, instead of the flow out of the ports being maximised, flow on all arcs (i, j) multiplied by the cost of the flow across those arcs c_{ij} is to be minimised. The objective in the minimum cost flow problem is therefore to

$$\left. \begin{aligned} \text{minimise } z &= \sum_k \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij}^k \\ \text{subject to } x_{ij}^k &\geq L_{ij}^k, & (i, j) \in \mathcal{A}, k \in \mathcal{K} \\ \sum_k x_{ij}^k &\leq U_j^k, & (i, j) \in \mathcal{A}, k \in \mathcal{K} \\ \sum_{(i,j) \in \mathcal{R}} \sum_{k \in \mathcal{K}} x_{ij}^k &\geq \left(\frac{1}{\lambda} - 1\right) \sum_{(i,j) \in \mathcal{U}} x_{ij}^C \\ \sum_j x_{ij}^k - \sum_h x_{hi}^k &= b_i^k, & (h, i), (i, j) \in \mathcal{A}, k \in \mathcal{K} \\ x_{ij}^k &\geq 0, & (i, j) \in \mathcal{A}, k \in \mathcal{K}, \end{aligned} \right\} \quad (5.3)$$

where \mathcal{A} is the set of all arcs and where \mathcal{K} is the set of all commodity types.

The variables representing the flow (x_{ij}^k) are decision variables which are to be found when solving the model — the solution will thus indicate the volume of fruit that should flow along the link between nodes i and j in a minimum cost flow solution. The values represented by c_{ij} , L_{ij}^k , U_j^k and b_i^k are parameters that are to be found or estimated before the model may be solved. The lower bounds on the fruit flow from each of the regions during the winter peak may be found in Table 5.21. The upper bounds on the nodes representing packhouse nodes may

Destination	Source	Transport Type	Fruit Type	Volume (pallets)
Maputo FPT Cold Store	Klein Karoo	Un	Hard citrus	708
	KwaZulu-Natal	Re	Soft citrus	84
	Klein Karoo	Re	Soft citrus	308
	Total			1 100 of 3 100
Maputo FPT Bulk Loading	Nelspruit	Re	Hard citrus	253
	Marble Hall	Re	Soft citrus	322
	Eastern Cape Midlands	Re	Soft citrus	43
	Sundays River Valley	Re	Soft citrus	2 268
	Langkloof	Re	Pome	3 580
	GEVV	Re	Pome	9 981
	Ceres	Re	Pome	5 609
	MfptCs		Hard citrus	468
	MfptCs		Soft citrus	391
Total			22 615 of 22 615	
Durban Cold Stores	Levubu	Un	Hard citrus	357
	Malelane	Un	Hard citrus	876
	Patensie	Un	Hard citrus	983
	Tzaneen	Un	Hard citrus	14 473
	Hoedspruit	Un	Hard citrus	1 805
	Nelspruit	Un	Hard citrus	1 485
	Klein Karoo	Un	Hard citrus	125
Total			20 104 of 20 104	
Durban FPT Cold Store	Kiepersol	Un	Hard citrus	190
	Piketberg	Un	Hard citrus	1 850
	Hartswater	Un	Hard citrus	506
	Rustenburg	Un	Hard citrus	26
	Levubu	Un	Hard citrus	4 746
	Nkwalini	Un	Hard citrus	7 322
	GEVV	Un	Hard citrus	2 880
Total			17 520 of 17 520	
Durban FPT Bulk Loading	Nelspruit	Un	Hard citrus	2 015
	Malelane	Un	Hard citrus	1 574
	Eastern Cape Midlands	Un	Hard citrus	232
	Langkloof	Un	Hard citrus	928
	Levubu	Un	Hard citrus	5 510
	Kiepersol	Un	Hard citrus	171
	Potgietersrus	Un	Hard citrus	2 365
	Marble Hall	Un	Hard citrus	6 493
	Sundays River Valley	Un	Hard citrus	5 315
	Hartswater	Un	Hard citrus	370
	Middelburg	Un	Hard citrus	9 800
	Patensie	Re	Hard citrus	816
	Johannesburg	Re	Hard citrus	126
	DCs		Hard citrus	5 081
	DfptCs		Hard citrus	4 524
Total			45 320 of 45 320	
Durban SAFT Bulk Loading	Pongola	Un	Hard citrus	1 454
	Klein Karoo	Un	Hard citrus	1 404
	Orange River	Un	Hard citrus	110
	GEVV	Un	Hard citrus	454
	Citrusdal	Un	Hard citrus	300
	DCs		Hard citrus	9 418
Total			13 140 of 13 140	
Port Elizabeth Cold Store	Patensie	Un	Soft citrus	410
	Total			410 of 6 060
Port Elizabeth FPT Cold Store	Citrusdal	Un	Hard citrus	4 971
	Boland	Un	Hard citrus	1 789
	Total			6 760 of 6 760
Port Elizabeth FPT Bulk Loading	Total			0 of 22 615
Cape Town Cold Stores	Marble Hall	Un	Hard citrus	1 819
	Ceres	Un	Pome	120
	Ceres	Un	Hard citrus	2 246
	Ceres	Un	Soft citrus	100
	Total			4 285 of 13 405
Cape Town FPT Cold Store	Boland	Un	Hard citrus	623
	Citrusdal	Un	Hard citrus	1 157
	Piketberg	Re	Pome	163
	Free State	Re	Pome	300
Total			2 243 of 8 000	
Cape Town FPT Bulk Loading	Kiepersol	Un	Hard citrus	360
	Tzaneen	Un	Hard citrus	82
	Patensie	Un	Hard citrus	4 350
	Malelane	Re	Hard citrus	350
	CTfptCs		Pome	450
Total			5 592 of 31 500	
Cape Town SAFT Bulk Loading	KwaZulu-Natal	Un	Hard citrus	981
	Boland	Un	Pome	892
	CTCs		Hard citrus	4 065
	CTCs		Soft citrus	100
	Total			6 038 of 10 500

Table 5.22: Solution of the maximum flow problem at the point where the transportation reaches the ports and port cities, where break-bulk fruit is handled for Scenario Three. The transport type refers to whether the vehicle is unrefrigerated (Un) or refrigerated (Re).

Destination	Source	Fruit Type	Volume (pallets)
Maputo FPT Container Loading	Tzaneen	Avocado	740
	MfptIn	Hard citrus	240
	Total		980 of 3 877
Durban Stack	Hoedspruit	Hard citrus	1 352
	Malelane	Hard citrus	600
	Orange River	Hard citrus	25
	DfptIn	Hard citrus	10 743
	Total		12 720 of 12 720
Durban FPT Container Loading	Sundays River Valley	Hard citrus	732
	Hartswater	Hard citrus	68
	Boland	Hard citrus	1 364
	DIn	Hard citrus	5 605
	Total		7 769 of 7 769
Durban SAFT Container Loading	DfptIn	Hard citrus	2 253
	Total		2 253 of 2 253
Port Elizabeth Stack	Langkloof	Pome	2 300
	GEVV	Pome	500
	Boland	Pome	510
	Tzaneen	Avocado	906
	PEIn	Soft citrus	410
	PEfptIn	Hard citrus	6 760
	Total		11 386 of 11 480
Port Elizabeth FPT Container Loading	Boland	Pome	570
	Langkloof	Pome	3 307
	Total		3 877 of 3 877
Cape Town Stack	Kiepersol	Avocado	1 203
	KwaZulu-Natal	Avocado	287
	Total		1 490 of 31 390
Cape Town Combi Stack	Hoedspruit	Soft citrus	51
	Nkwalini	Avocado	25
	Klein Karoo	Pome	26
	GEVV	Soft citrus	3 654
	Boland	Avocado	140
	Boland	Pome	600
	Ceres	Pome	2 679
	Total		7 175 of 7 200
Cape Town FPT Container Loading	Levubu	Avocado	610
	Nelspruit	Avocado	520
	Marble Hall	Hard citrus	731
	Eastern Cape Midlands	Hard citrus	413
	Boland	Soft citrus	46
	Piketberg	Hard citrus	415
	Total		2 735 of 11 200
Cape Town SAFT Container Loading	CTIn	Pome	120
	CTfptIn	Hard citrus	1 780
	Total		1 900 of 3 733
Blue Store	Malelane	Hard citrus	2 414
	Sundays River Valley	Hard citrus	2 036
	Boland	Pome	550
	Total		5 000 of 5 000
CTCT Container Loading	BS	Pome	550
	BS	Hard citrus	4 450
	CTS	Avocado	1 490
	CTcomS	Pome	3 305
	CTcomS	Soft citrus	3 681
	CTcomS	Avocado	165
	Total		13 641 of 45 840
DCT Container Loading	DS	Hard citrus	12 720
	Total		12 720 of 103 940
PECT Container Loading	PES	Pome	3 310
	PES	Hard citrus	6 760
	PES	Soft citrus	410
	PES	Avocado	906
	Total		11 386 of 50 540

Table 5.23: Solution of the maximum flow problem at the point where the transportation reaches the ports and port cities, where containerised fruit is handled for Scenario Three. All fruit arriving at ports directly from the production regions would do so in containers, otherwise it would first pass through harbour cold stores.

be found in Table 3.1, whilst Table 3.2 lists the upper bounds on the flow through cold stores. Furthermore, Table 3.13 lists the upper bounds on nodes representing stacks and container terminal loading rates and Table 3.14 summarises the maximum volume of fruit that may be loaded at the break-bulk terminals. Tables 2.5, 2.6 and 2.7 show the costs of cooling fruit, and handling fruit at the FPT and SAFT terminals respectively. The container terminal costs, as well as other costs, were discussed in §2.4.3 and §2.4.5. Shipping costs may be found in Table 2.8 and transportation costs are summarised in Appendix F. These parameters may also all be found in the spreadsheet files on the CD attached to the thesis.

5.6 Minimum Cost Flow Solution

In this section, the results obtained by solving the minimum cost flow problem are summarised. Also, the effect of small changes to the model parameters are investigated. First the export during week seven of 2003 is considered (the summer peak week), followed by the export during week 29 of 2003 (the winter peak week). The models used for solve the minimum cost flow problem for the summer and winter scenarios had 1 111 constraints and 3 575 variables, and 852 constraints and 4 069 variables, respectively.

5.6.1 Summer Peak Week

The aim in this section is to find the minimum cost flow of transporting all pallets of export fruit from their respective production regions to the ports during the summer peak (week seven of 2003). For this model, 137 pallets of pome fruit that were estimated to have been exported from the De Doorns region, were allocated to the Ceres region (region with similar distances to the ports), as no pome packhouse was identified during the packhouse capacity data gathering process.

Destination	Source	Transport Type	Fruit Type	Volume (pallets)
Maputo FPT Cold Store	Marble Hall	UnTk	Hard citrus	7
	Middelburg	UnTk	Hard citrus	9
	Total			16 of 3 100
Maputo FPT Bulk Loading				0
	Middelburg	UnTk	Hard citrus	7
	MfptCs		Hard citrus	16
	Total			23 of 23 184
Durban Cold Stores				0 of 20 104
				0
Durban FPT Cold Store				0 of 17 520
				0
Durban FPT Bulk Loading				0 of 45 320
				0
Durban SAFT Bulk Loading				0 of 13 140
				0
Port Elizabeth Cold Store				0 of 6 060
				0
Port Elizabeth FPT Cold Store				0 of 6 760
				0
Port Elizabeth FPT Bulk Loading				0 of 22 615
				0
Cape Town Cold Stores				0 of 13 405
				0
Cape Town FPT Cold Store				0 of 8 000
				0
Cape Town FPT Bulk Loading				0 of 31 500
				0
Cape Town SAFT Bulk Loading				0 of 10 500
				0

Table 5.24: Solution of the minimum cost flow problem at the point where the transportation reaches the ports and port cities, where break-bulk fruit is handled for the summer peak week (week seven of 2003). The mode of transportation refers to whether transport is via an unrefrigerated vehicle (UnTk), an unrefrigerated train (UnTn), a refrigerated vehicle (ReTk) or in integral containers (In). Very low volumes of fruit make use of the port infrastructure handling break-bulk fruit.

Solutions were obtained for three possible subtropical fruit cooling tariffs, namely $R120$, $R160$ and $R200$ per pallet. It was found that the results were identical in all three cases, except for the shadow prices on some constraints involving subtropical fruits. However, the shadow prices differed by the same margin as the tariffs. Thus, from this point on, only the $R160$ tariff per pallet is considered and the shadow prices on constraints that are dependent on variables representing subtropical fruit flow can either be added to $R40$ to find the shadow price for a

Destination	Source	Fruit Type	Volume (pallets)
Maputo FPT Container Loading	Malelane	Litchi	4
	Total		4 of 3 877
	Shadow Price		0
Durban Stack	Levubu	Mango	9
	Levubu	Litchi	47
	Tzaneen	Mango	401
	Tzaneen	Litchi	130
	Hoedspruit	Mango	432
	Nelspruit	Mango	69
	Nelspruit	Litchi	37
	Malelane	Mango	209
	Malelane	Litchi	109
	Hartswater	Stone	45
	Johannesburg	Stone	79
	Rustenburg	Stone	25
	Free State	Pome	33
Total		1 625 of 12 720	
Shadow Price		0	
Durban FPT Container Loading	Total		0 of 7 769
	Shadow Price		0
Durban SAFT Container Loading	Total		0 of 2 253
	Shadow Price		0
Port Elizabeth Stack	Langkloof	Stone	198
	Langkloof	Pome	2 011
	Total		2 209 of 11 480
Port Elizabeth FPT Container Loading	Total		0 of 3 877
	Shadow Price		0
Cape Town Stack	Klein Karoo	Pome	588
	GEVV	Grapes	6 437
	GEVV	Pome	5 910
	Boland	Grapes	8 115
	Boland	Stone	1 960
	Ceres	Pome	5 986
	Piketberg	Pome	166
	Total		29 162 of 31 390
	Shadow Price		0
	Cape Town Combi Stack	Klein Karoo	Stone
GEVV		Stone	663
Boland		Pome	930
Ceres		Grapes	2 543
Ceres		Stone	629
De Doorns		Grapes	1 272
Piketberg		Grapes	218
Piketberg		Stone	94
Piketberg		Pome	94
Orange River		Grapes	52
Total		7 200 of 7 200	
Shadow Price		0	
Cape Town FPT Container Loading	Total		0 of 11 200
	Shadow Price		0
Cape Town SAFT Container Loading	Total		0 of 3 733
	Shadow Price		0
CTCT Container Loading	C'TcomS	Grapes	14 552
	C'TcomS	Stone	1 960
	C'TcomS	Pome	12 650
	Total		36 362 of 45 840
Shadow Price		0	
DCT Container Loading	DS	Stone	149
	DS	Pome	33
	DS	Mango	1 120
	DS	Litchi	323
	Total		1 625 of 103 940
Shadow Price		0	
PECT Container Loading	PES	Stone	198
	PES	Pome	2 011
	Total		2 209 of 50 540
Shadow Price		0	

Table 5.25: Solution of the minimum cost flow problem at the point where the transportation reaches the ports and port cities, where containerised fruit is handled for the summer peak week (week seven of 2003). All fruit arriving at ports directly from the production regions would do so in containers, otherwise it would first pass through harbour cold stores (seen in Table 5.24).

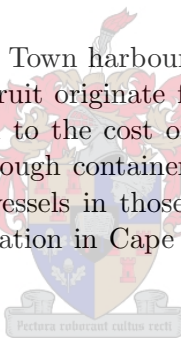
cooling tariff of $R200$ per pallet, or $R40$ can be subtracted to find the equivalent shadow price for a cooling price of $R120$ per pallet. Changing the infrastructure to allow any volume of fruit through would not affect the solution in this case, as none of the infrastructural components are used to their maximum capacities. The only problematic component is encountered at the Cape Town Stack, because if the volumes of export fruit flowing through the stack and the

combi-stack are added together, the total is greater than the capacity of the stack. Thus, if only the stack were available to the container terminal, there would be a shortage of plug-in points. However, this shortfall is for the infrastructure in 2003; by 2005 an additional thousand plug-in points will have been installed, which would cover the demand if the combi-stack were unavailable.

It may be seen in Tables 5.24 and 5.25, that Maputo, Durban and Port Elizabeth harbours are used for very small volumes of fruit in the minimum cost flow solution. This makes it unlikely that the fruit terminals can operate at a profit during this period (the SAFT in Durban and the FPT in Durban and Port Elizabeth are not even required). However, the container terminals can continue operating as there are many other non-fruit commodities that make use of these terminals. In Table 5.26 a summary is shown of the minimum cost flow that fruit could achieve if only Cape Town harbour were available to fruit exporters.

It is clear from the summary shown in Table 5.26 that the cheapest method of export is by container via the container terminal. Only 1 680 pallets of fruit are exported per week in break-bulk in the minimum cost solution, while a total of 38 543 pallets are exported by container (of which approximately 98.5% is via the container terminal). A comparison of the shadow prices associated with the constraints forcing the fruit export volumes to be the same as during week seven of 2003 is shown in Table 5.27. These shadow prices indicate approximately how much it costs to export a single pallet of fruit.

By allowing fruit to pass through Cape Town harbour, only R1.86 million of additional transportation costs are incurred (as most fruit originate from regions to which Cape Town is the nearest port). This must be compared to the cost of opening other terminals during such a quiet period. Fruit could still pass through container terminals, but the viability of such an option depends on the availability of vessels in those ports. It would be more likely to find vessels travelling to the required destination in Cape Town, where most of the fruit would be sent.



5.6.2 Winter Peak Week

The purpose of this subsection is to present the results of the minimum cost flow evaluation for the volumes exported during the winter peak week of 2003. The packhouse capacities of the Kiepersol (153 pallets), Nelspruit (792 pallets), Malelane (1 226 pallets), Eastern Cape Midlands (849 pallets), Klein Karoo (141 pallets) and the Orange River (144 pallets) regions had to be increased to allow for the volume of fruit that was estimated to have been exported during week 29 of 2003. Also, the hard and soft citrus volumes from unknown regions were distributed amongst the regions in proportion to their export volumes. Soft citrus from the Tzaneen region was allocated to the Levubu region as there are no packhouses or cold stores in this region that handle soft citrus to the best knowledge of the author, and the fraction of avocados said to be from “Airfreight” in the data provided by SAAGA was distributed amongst Tzaneen and Nelspruit (the two largest avocado exporting regions). Hard citrus fruit export from Zimbabwe was added to the Levubu (closest) region and hard citrus export from Swaziland was added to the Nkwalini (closest) region as no packhouse or cold store data was available for those regions. Three different model variations were solved for this week and the results are summarised in Tables 5.28 – 5.33.

After the model had been solved for all three costs for cooling subtropical fruit ($R120$ /pallet, $R160$ /per pallet and $R200$ /pallet), the results indicated identical fruit flow patterns in the network for all three costs. This was predictable as subtropical fruit must be cooled before

Destination	Source	Transport Type	Fruit Type	Volume (pallets)
Cape Town Cold Stores	Total	0 of 13 405		
	Shadow Price	0		
Cape Town FPT Cold Store	Langkloof	ReTk	Pome	597
	Total	597 of 8 000		
Cape Town FPT Bulk Loading	Shadow Price	0		
	Middelburg	UnTk	Hard citrus	16
	Marble Hall	ReTk	Hard citrus	7
	Langkloof	ReTk	Stone	198
	Langkloof	ReTk	Pome	1 414
	Hartswater	ReTk	Stone	45
Cape Town SAFT Bulk Loading	Total	1 680 of 31 500		
	Shadow Price	0		
Cape Town Stack	Total	0 of 10 500		
	Shadow Price	0		
	Tzaneen	InTk	Mango	401
	Malelane	InTk	Litchi	113
	GEVV	InTk	Grapes	6 437
	GEVV	InTk	Pome	5 910
	Boland	InTk	Grapes	8 115
	Boland	InTk	Stone	1 960
	Ceres	InTk	Grapes	2 543
	Ceres	InTk	Pome	3 901
	De Doorns	InTk	Grapes	1 272
	Piketberg	InTk	Stone	94
	Total	30 746 of 31 390		
	Shadow Price	0		
Cape Town Combi Stack	Levubu	InTk	Mango	9
	Levubu	InTk	Litchi	47
	Tzaneen	InTk	Litchi	130
	Hoedspruit	InTk	Mango	432
	Nelspruit	InTk	Mango	69
	Nelspruit	InTk	Litchi	37
	Malelane	InTk	Mango	209
	Klein Karoo	InTk	Stone	705
	Klein Karoo	InTk	Pome	588
	GEVV	InTk	Stone	663
	Boland	InTk	Pome	930
	Ceres	InTk	Stone	629
	Ceres	InTk	Pome	2 085
	Piketberg	InTk	Grapes	218
	Piketberg	InTk	Pome	260
	Orange River	InTk	Grapes	52
	Johannesburg	InTk	Stone	79
	Rustenburg	InTk	Stone	25
	Free State	InTk	Pome	33
Total	7 200 of 7 200			
Shadow Price	0			
Cape Town FPT Container Loading	CTiptIn		Pome	597
	Total	597 of 11 200		
Cape Town SAFT Container Loading	Shadow Price	0		
	Total	0 of 3 733		
CTCT Container Loading	Shadow Price	0		
	CTS		Grapes	18 367
	CTS		Stone	2 054
	CTS		Pome	9 811
	CTS		Mango	401
	CTS		Litchi	113
	CTcomS		Grapes	270
	CTcomS		Stone	2 101
	CTcomS		Pome	3 896
	CTcomS		Mango	719
	CTcomS		Litchi	214
	Total	37 946 of 45 840		
Shadow Price	0			

Table 5.26: Solution of the minimum cost flow problem if only the infrastructure in Cape Town is available for fruit exporters during the summer peak week (week seven of 2003). The transport type refers to whether the mode of transportation is via an unrefrigerated vehicle (UnTk), an unrefrigerated train (UnTn), a refrigerated vehicle (ReTk) or in integral containers (In).

transport, and since the same cooling rates were assumed for all cold stores in all regions, the costs could not be avoided. Thus the factors that influenced the fruit flow were after the cold store level in the export supply chain (except for citrus fruit, as this may be transported unrefrigerated and with no prior cooling), which included a transportation cost and handling tariffs at the terminals or cold stores near ports. Thus, from this point on, only the costs related to the R160 cooling cost for subtropical fruit are discussed. To calculate the cost for the lower or higher cooling costs, R40 may be subtracted from or added if the cost involves subtropical fruit cooling. For example, it costs approximately R2 543 to export a pallet of avocados from the Levubu region in winter (to which packing and PPECB inspections must be added) if the

Region	Fruit Type	Exported Volumes	All Ports	Cape Town Port Only
Levubu	Mango	9	2 915	3 467
	Litchi	47	2 915	3 467
Tzaneen	Mango	401	2 833	3 415
	Litchi	130	2 833	3 415
Hoedspruit	Mango	432	2 738	3 632
Ki, Ne and Ma	Mango	209	2 716	3 637
	Litchi	113	2 716	3 637
Nelspruit	Mango	69	2 681	3 577
	Litchi	37	2 681	3 577
Po, Mh, Ru and Mi	Hard citrus	23	1 939	2 310
Langkloof	Stone	198	1 907	2 200
	Pome	2 011	1 827	2 120
Klein Karoo	Stone	705	1 898	1 898
	Pome	588	1 818	1 818
Kl, Ge, Ce and De	Grapes		1 882	1 882
GEVV	Stone	663	1 383	1 796
	Pome	5 910		1 716
Boland	Grapes	182	1 796	1 796
	Stone	1 960	1 800	1 800
	Pome	930	1 720	1 720
Pi and Ci	Grapes	218	1 872	1 872
Ceres	Stone	629	1 886	1 886
	Pome	5 986	1 806	1 806
Piketberg	Stone	94	1 876	1 876
	Pome	260	1 796	1 796
Orange River	Grapes	52	2 347	2 347
Hartswater	Stone	45	2 298	2 450
Johannesburg	Stone	79	2 185	2 725
Rustenburg	Stone	25	2 264	3 001
Free State	Pome	33	1 964	2 522
Objective function	All	82 722	73 579 886	75 442 432

Table 5.27: Shadow prices of the minimum fruit flow constraints. All values were rounded to nearest Rand. A cooling cost of R160 per pallet of subtropical fruit was assumed. To calculate the estimated cost of export, R40 may be subtracted from the estimated export cost of subtropical fruit to find the cost if cooling cost was R120 per pallet, or alternatively R40 may be added to find the estimated export cost if the cooling cost was R200 per pallet. The costs of PPECB carton inspections, packing and VAT are not included.

cooling cost for avocados is R120 per pallet, R2 583 for R160 per pallet cooling costs or R2 623 if the cooling cost is R200 per pallet.

In the results shown in Tables 5.28 and 5.29 indicate that exporting fruit via containers from the terminals specialising in break-bulk export is the cheapest method of fruit export. The shadow prices on the constraints restricting the flow of containerised fruit through these terminals show this fact. For every increase in container export capacity from the FPT terminal in Maputo, R207 could thus be saved. The Durban and Cape Town SAFT container handling capacities would decrease fruit export costs by up to R67 per pallet. However, it became clear that in some cases (such as the SAFT and FPT terminals in Cape Town) more pallets were exported by container than by break-bulk. This is not a likely scenario as there must be enough break-bulk fruit to fill the holds of the conventional vessels before any containers of fruit may be packed on the decks. Thus, an additional constraint was added to the model to ensure that the ratio of break-bulk to containerised fruit remained as was assumed for the calculations of the terminal capacities (see §3.3). This constraint had the form

$$\sum_{(i,j) \in \mathcal{C}_f} x_{ij} - \frac{\text{Terminal Container Capacity}}{\text{Terminal Break-bulk Capacity}} \sum_{(i,j) \in \mathcal{B}_f} x_{ij} \leq 0,$$

where \mathcal{C}_f is the set of arcs representing the flow of containerised fruit out of terminal f and \mathcal{B}_f is the set of arcs representing possible break-bulk fruit flow out of terminal f . This restriction was also added to the summer scenario results shown in §5.6.1. The results are shown in Tables 5.30 and 5.31 respectively.

Destination	Source	Transport Type	Fruit Type	Volume (pallets)
Maputo FPT Cold Store	Malelane	UnTk	Hard citrus	3 100
	Total			3 100 of 3 100
	Shadow Price			-430.64
Maputo FPT Bulk Loading	Malelane	UnTk	Hard citrus	1 395
	MfptCs		Hard citrus	3 100
	Total			4 495 of 23 184
	Shadow Price			0
Durban Cold Stores	Nelspruit	UnTk	Hard citrus	3 910
	Nkwalini	UnTk	Hard citrus	7 322
	Nkwalini	UnTk	Soft citrus	58
	Middelburg	UnTk	Hard citrus	5 428
	Total			16 718 of 20 104
	Shadow Price			0
Durban FPT Cold Store	Tzaneen	UnTn	Hard citrus	13 240
	Hoedspruit	UnTn	Hard citrus	2 306
	Nelspruit	UnTn	Hard citrus	379
	Malelane	UnTn	Hard citrus	1 595
	Total			17 520 of 17 520
	Shadow Price			-28.14
Durban FPT Bulk Loading	Levubu	UnTk	Hard citrus	3 241
	Kiepersol	UnTk	Hard citrus	877
	Middelburg	UnTk	Hard citrus	270
	DfptCs		Hard citrus	9 751
	Total			14 139 of 45 320
	Shadow Price			0
Durban SAFT Bulk Loading	Levubu	UnTk	Hard citrus	3 744
	Pongola	UnTk	Hard citrus	27
	KwaZulu-Natal	UnTk	Hard citrus	307
	DCs		Hard citrus	9 062
	Total			13 140 of 13 140
	Shadow Price			-2.20
Port Elizabeth Cold Store	Total			0 of 6 060
	Shadow Price			0
Port Elizabeth FPT Cold Store	Sundays River Valley	UnTk	Hard citrus	3 970
	Patensie	UnTk	Hard citrus	1 312
	Patensie	UnTk	Soft citrus	133
	Total			5 415 of 6 760
	Shadow Price			0
Port Elizabeth FPT Bulk Loading	Eastern Cape Midlands	UnTk	Hard citrus	1 494
	Patensie	UnTk	Hard citrus	943
	PEfptCs		Hard citrus	5 282
	PEfptCs		Soft citrus	133
	Total			7 852 of 22 615
	Shadow Price			0
Cape Town Cold Stores	Total			0 of 13 405
	Shadow Price			0
Cape Town FPT Cold Store	GEVV	UnTk	Hard citrus	3 931
	Boland	UnTk	Hard citrus	3 776
	Orange River	UnTk	Hard citrus	279
	Total			7 986 of 8 000
	Shadow Price			0
Cape Town FPT Bulk Loading	Total			0 of 31 500
	Shadow Price			0
Cape Town SAFT Bulk Loading	Total			0 of 10 500
	Shadow Price			0

Table 5.28: Solution of the minimum cost flow problem at the point where the transportation reaches the ports and port cities, where break-bulk fruit is handled for the winter peak week (week 29 of 2003). The transport type refers to whether the mode of transportation is via an unrefrigerated vehicle (UnTk), an unrefrigerated train (UnTn), a refrigerated vehicle (ReTk) or in integral containers (In).

The corrected results indicate that the cold store at the FPT terminal in Maputo becomes a considerable problem. As only 31% of the fruit loaded onto a vessel may be ambient, the cold store is required to cool large amounts of citrus fruit. It has the largest shadow price of all port infrastructure constraints and if the cold store could be expanded, could save some producers up to R518 per pallet of costs. A similar problem occurs at the FPT cold store in Durban (with a shadow price of $-R37.66$), where there is not enough cooling space to allow cheaper export. The SAFT terminal is used to capacity (as it seems to be cheapest), so remaining fruit is diverted to FPT. Due to the restriction that only 31% of the citrus fruit is allowed to be at ambient temperature when loaded, as much fruit is cooled as possible. However, there is more loading capacity at FPT than can be filled from the FPT cold store (which is required to allow uncooled fruit to be loaded into a conventional vessel) — hence it is the cold store with the

Destination	Source	Fruit Type	Volume (pallets)
Maputo FPT Container Loading	Levubu	Avocado	72
	Tzaneen	Avocado	760
	Hoedspruit	Hard citrus	781
	Hoedspruit	Soft citrus	51
	Kiepersol	Avocado	487
	Nelspruit	Hard citrus	256
	Nelspruit	Avocado	520
Malelane	Hard citrus	950	
	Total		3 877 of 3 877
	Shadow Price		-207.23
Durban Stack	Marble Hall	Soft citrus	322
	KwaZulu-Natal	Avocado	150
	Free State	Pome	41
	DIn	Hard citrus	6 103
	Total		6 616 of 12 720
	Shadow Price		0
Durban FPT Container Loading	DfptIn	Hard citrus	7 769
	Total		7 769 of 7 769
	Shadow Price		-42.81
Durban SAFT Container Loading	KwaZulu-Natal	Hard citrus	674
	KwaZulu-Natal	Soft citrus	26
	DIn	Hard citrus	1 495
	DIn	Soft citrus	58
	Total		2 253 of 2 253
	Shadow Price		-67.65
Port Elizabeth Stack	Sundays River Valley	Soft citrus	406
	Patensie	Hard citrus	690
	Langkloof	Pome	1 422
	Total		2 530 of 11 480
	Shadow Price		0
Port Elizabeth FPT Container Loading	Eastern Cape Midlands	Soft citrus	43
	Sundays River Valley	Hard citrus	2 594
	Patensie	Hard citrus	1 240
	Total		3 877 of 3 877
	Shadow Price		-17.45
Cape Town Stack	Total		0 of 31 390
	Shadow Price		0
Cape Town Combi Stack	GEVV	Pome	1 258
	GEVV	Soft citrus	1 070
	Ceres	Pome	2 127
	Total		4 455 of 7 200
	Shadow Price		0
Cape Town FPT Container Loading	CTfptIn	Hard citrus	7 986
	Total		7 986 of 11 200
	Shadow Price		0
Cape Town SAFT Container Loading	Klein Karoo	Pome	167
	GEVV	Pome	3 221
	Boland	Boland	182
	Piketberg	Pome	163
	Total		3 733 of 3 733
	Shadow Price		-67.65
CTCT Container Loading	CTcomS	Pome	3 385
	CTcomS	Soft citrus	1 070
	Total		4 455 of 45 840
	Shadow Price		0
DCT Container Loading	DS	Pome	41
	DS	Hard citrus	6 103
	DS	Soft citrus	322
	DS	Avocado	150
	Total		6 616 of 103 940
	Shadow Price		0
PECT Container Loading	PES	Pome	1 422
	PES	Hard citrus	690
	PES	Soft citrus	406
	Total		2 530 of 50 540
	Shadow Price		0

Table 5.29: Solution of the minimum cost flow problem at the point where the transportation reaches the ports and port cities, where containerised fruit is handled for the winter peak week (week 29 of 2003). All fruit arriving directly from production regions would do so in containers, otherwise it would first pass through harbour cold stores (seen in Table 5.28).

non-zero shadow price. An additional factor for this non-zero shadow price is that it seems to be cheaper for the fruit to be loaded onto the deck of the vessel in a container than into the hold of the vessel in break-bulk format (as seen in the first solution to the winter minimum cost flow problem).

Destination	Source	Transport Type	Fruit Type	Volume (pallets)
Maputo FPT Cold Store	Malelane	UnTk	Hard citrus	3 100
	Total			3 100 of 3 100
Maputo FPT Bulk Loading	Shadow Price			-518.13
	Malelane MfptCs	UnTk	Hard citrus	1 395
Durban Cold Stores	Total			4 495 of 23 184
	Shadow Price			0
Durban FPT Cold Store	Tzaneen	UnTk	Hard citrus	1 352
	Kiepersol	UnTk	Hard citrus	877
	Nelspruit	UnTk	Hard citrus	4 545
	Pongola	UnTk	Hard citrus	27
	Nkwalini	UnTk	Hard citrus	3 888
	Nkwalini	UnTk	Soft citrus	84
	KwaZulu-Natal	UnTk	Hard citrus	307
	Middelburg	UnTk	Hard citrus	5 698
Durban FPT Bulk Loading	Total			16 778 of 20 104
	Shadow Price			0
Durban SAFT Bulk Loading	Tzaneen	UnTn	Hard citrus	11 888
	Hoedspruit	UnTn	Hard citrus	3 087
Port Elizabeth Cold Store	Malelane	UnTn	Hard citrus	2 545
	Total			17 520 of 17 520
Port Elizabeth FPT Cold Store	Shadow Price			-37.66
	Levubu	UnTk	Hard citrus	6 315
Port Elizabeth FPT Bulk Loading	DfptCs		Hard citrus	14 033
	Total			20 347 of 45 320
Durban SAFT Bulk Loading	Shadow Price			0
	Levubu	UnTk	Hard citrus	670
Cape Town Cold Stores	Nkwalini	UnTk	Hard citrus	3 408
	DCs		Hard citrus	9 062
Cape Town FPT Cold Store	Total			13 140 of 13 140
	Shadow Price			-13.79
Cape Town FPT Bulk Loading	Total			0 of 6 060
	Shadow Price			0
Cape Town SAFT Bulk Loading	Eastern Cape Midlands	UnTk	Hard citrus	1 494
	Sundays River Valley	UnTk	Hard citrus	1 533
Cape Town SAFT Bulk Loading	Patensie	UnTk	Hard citrus	2 255
	Patensie	UnTk	Soft citrus	133
Cape Town SAFT Bulk Loading	Total			5 415 of 6 760
	Shadow Price			0
Cape Town SAFT Bulk Loading	Sundays River Valley	UnTk	Hard citrus	2 437
	Patensie	UnTk	Hard citrus	943
Cape Town SAFT Bulk Loading	PEfptCs		Hard citrus	5 282
	PEfptCs		Soft citrus	133
Cape Town SAFT Bulk Loading	Total			7 852 of 22 615
	Shadow Price			0
Cape Town SAFT Bulk Loading	Total			0 of 13 405
	Shadow Price			0
Cape Town SAFT Bulk Loading	Orange River	UnTk	Hard citrus	215
	Total			215 of 8 000
Cape Town SAFT Bulk Loading	Shadow Price			0
	Orange River	UnTk	Hard citrus	64
Cape Town SAFT Bulk Loading	CTfptCs		Hard citrus	142
	Total			206 of 31 500
Cape Town SAFT Bulk Loading	Shadow Price			0
	Total			0 of 10 500
Cape Town SAFT Bulk Loading	Shadow Price			0

Table 5.30: The corrected solution of the minimum cost flow problem at the point where the transportation reaches the ports and port cities, where break-bulk fruit is handled for the winter peak week (week 29 of 2003). The transport type refers to whether the mode of transportation is via an unrefrigerated vehicle (UnTk), an unrefrigerated train (UnTn), a refrigerated vehicle (ReTk) or in integral containers (In).

The fact, that the break-bulk loading capacity of the SAFT terminal does not allow more fruit to be loaded, means that not more fruit can be loaded in containers (due to the restriction that the ratio of containerised volumes to break-bulk volumes must be below a certain threshold). Containerised fruit export from the terminal proved to be the cheapest export method from Durban (due to the low shadow price shown in Table 5.29), which is why the break-bulk loading volume of the terminal has a non-zero shadow price. Increasing the volume of break-bulk fruit that may be loaded onto vessels by SAFT in Durban would mean that more fruit may be packed onto the decks of vessels in containers, thus saving on costs.

Destination	Source	Fruit Type	Volume (pallets)
Maputo FPT Container Loading	Kiepersol	Avocado	487
	Nelspruit	Hard citrus	283
	Total		770 of 3 877
	Shadow Price		0
Durban Stack	Levubu	Avocado	72
	Tzaneen	Avocado	760
	Nelspruit	Avocado	237
	Marble Hall	Soft citrus	322
	Free State	Pome	41
	DIn	Hard citrus	6 365
	Total		7 797 of 12 720
	Shadow Price		0
Durban FPT Container Loading	DfptIn	Hard citrus	3 488
	Total		3 488 of 7 769
	Shadow Price		0
Durban SAFT Container Loading	Hoedspruit	Soft citrus	51
	KwaZulu-Natal	Hard citrus	700
	KwaZulu-Natal	Avocado	150
	DIn	Hard citrus	1 267
	DIn	Soft citrus	84
	Total		2 252 of 2 253
	Shadow Price		0
Port Elizabeth Stack	Sundays River Valley	Hard citrus	2 594
	Sundays River Valley	Soft citrus	406
	Patensie	Hard citrus	639
	Langkloof	Pome	1 422
	Total		5 061 of 11 480
	Shadow Price		0
Port Elizabeth FPT Container Loading	Eastern Cape Midlands	Soft citrus	43
	Patensie	Hard citrus	1 291
	Langkloof	Hard citrus	12
	Total		1 346 of 3 877
	Shadow Price		0
Cape Town Stack	GEVV	Pome	2 007
	GEVV	Citrus	6 688
	Total		8 695 of 31 390
	Shadow Price		0
Cape Town Combi Stack	Klein Karoo	Pome	167
	GEVV	Pome	2 472
	Boland	Pome	182
	Boland	Hard citrus	1 019
	Boland	Soft citrus	1 070
	Ceres	Pome	2 127
	Piketberg	Pome	163
	Total		7 200 of 7 200
	Shadow Price		0
Cape Town FPT Container Loading	CTfptIn	Hard citrus	73
	Total		73 of 11 200
	Shadow Price		0
Cape Town SAFT Container Loading	Total		3 733 of 3 733
	Shadow Price		0
CTCT Container Loading	CTS	Pome	2 007
	CTS	Hard citrus	6 688
	CTcomS	Pome	5 111
	CTcomS	Hard citrus	1 019
	CTcomS	Soft citrus	1 070
	Total		15 895 of 45 840
	Shadow Price		0
DCT Container Loading	DS	Pome	41
	DS	Hard citrus	6 365
	DS	Soft citrus	322
	DS	Avocado	1 069
	Total		7 797 of 103 940
	Shadow Price		0
PECT Container Loading	PES	Pome	1 422
	PES	Hard citrus	3 233
	PES	Soft citrus	406
	Total		5 061 of 50 540
	Shadow Price		0

Table 5.31: The corrected solution of the minimum cost flow problem at the point where the transportation reaches the ports and port cities, where containerised fruit is handled for the winter peak week (week 29 of 2003). All fruit arriving directly from production regions would do so in containers, otherwise it would first pass through harbour cold stores (see Table 5.30).

To determine to which harbour producers would send their fruit in order to maximise cost savings, the constraints on the harbour infrastructure were relaxed by setting the maximum fruit export volumes to a very large number. However, the restrictions on the containerised volume to break-bulk volume ratio and the unrefrigerated fruit to refrigerated fruit ratio remained the same in the model. The results are shown in Tables 5.32 and 5.33.

Destination	Source	Transport Type	Fruit Type	Volume (pallets)
Maputo FPT Cold Store	Levubu	UnTk	Hard citrus	6 985
	Tzaneen	UnTk	Hard citrus	13 240
	Nelspruit	UnTk	Hard citrus	4 289
	Middelburg	UnTk	Hard citrus	3 256
	Total			27 770 of 3 100
Maputo FPT Bulk Loading	Hoedspruit	UnTk	Hard citrus	3 087
	Kiepersol	UnTk	Hard citrus	877
	Malelane	UnTk	Hard citrus	6 090
	Middelburg	UnTk	Hard citrus	2 442
	MfptCs	UnTk	Hard citrus	27 770
Total			40 266 of 23 184	
Durban Cold Stores	Nkwalini	UnTk	Hard citrus	5 350
	Nkwalini	UnTk	Soft citrus	84
	Total			5 434 of 20 104
Durban FPT Cold Store	Total			0 of 17 520
Durban FPT Bulk Loading	Total			0 of 45 320
Durban SAFT Bulk Loading	Pongola	UnTk	Hard citrus	27
	Nkwalini	UnTk	Hard citrus	1 946
	KwaZulu-Natal	UnTk	Hard citrus	307
	DCs		Hard citrus	4 982
	DCs		Soft citrus	84
Total			7 346 of 13 140	
Port Elizabeth Cold Store	Total			0 of 6 060
Port Elizabeth FPT Cold Store	Sundays River Valley	UnTk	Hard citrus	3 970
	Patensie	UnTk	Hard citrus	1 312
	Patensie	UnTk	Soft citrus	133
	Total			5 415 of 6 760
Port Elizabeth FPT Bulk Loading	Eastern Cape Midlands	UnTk	Hard citrus	1 494
	Patensie	UnTk	Hard citrus	943
	PEfptCs		Hard citrus	5 282
	PEfptCs		Soft citrus	133
Total			7 852 of 22 615	
Cape Town Cold Stores	Total			0 of 13 405
Cape Town FPT Cold Store	Orange River	UnTk	Hard citrus	215
	Total			215 of 8 000
Cape Town FPT Bulk Loading	Orange River	UnTk	Hard citrus	64
	CTfptCs		Hard citrus	142
	Total			206 of 31 500
Cape Town SAFT Bulk Loading	Total			0 of 10 500

Table 5.32: Solution of the minimum cost flow problem (where port capacities have been removed) at the point where the transportation reaches the ports and port cities, where break-bulk fruit is handled for the winter peak week (week 29 of 2003). The transport type refers to whether the mode of transportation is via an unrefrigerated vehicle (UnTk), an unrefrigerated train (UnTn), a refrigerated vehicle (ReTk) or in integral containers (In).

The results obtained by allowing any volume of fruit through the port infrastructures indicate that much more fruit should pass through Maputo's FPT terminal if the producers in the Levubu, Tzaneen, Hoedspruit, Kiepersol, Nelspruit, Malelane, Marble Hall and Middelburg regions were attempting to maximise cost savings. If these regions were to export more fruit through Maputo, Durban would lose much of the volume that currently flows through it. In the results summarised in Tables 5.30 and 5.31, 5 265 pallets of fruit were exported via Maputo and 47 024 pallets were exported via Durban. However, if the restrictions on infrastructure are removed from the models (as summarised in Tables 5.32 and 5.33), it is found that 43 684 pallets should have been exported from Maputo and only 8 605 pallets from Durban if the hypothetical minimum cost were to be obtained. The volumes exported via Port Elizabeth and Cape Town remain the same as these ports are used by the Eastern Cape Midlands, Sundays River Valley, Patensie and Langkloof regions, and the Klein Karoo, GEVV, Boland, Ceres, Piketberg and Orange River regions, respectively. If Maputo had greater capacity, only the Pongola, Nkwalini, KwaZulu-Natal Midlands and Free State regions would use Durban harbour for fruit export. However, this scenario assumes that exporters would only be interested in reducing the cost of export as much as possible. When exporting subtropical fruits, exporters often additionally consider the time it takes for fruit to reach the destination. Thus, they may rather transport the subtropical fruit to Cape Town by road to save on the sailing time from Maputo or Durban to Cape Town [291].

Destination	Source	Fruit Type	Volume (pallets)
Maputo FPT Container Loading	Levubu	Avocado	72
	Tzaneen	Avocado	760
	Hoedspruit	Soft citrus	51
	Kiepersol	Avocado	487
	Nelspruit	Hard citrus	256
	Nelspruit	Avocado	520
	Malelane	Hard citrus	950
	Marble Hall	Soft citrus	322
	Total		3 418 of 3 877
Durban Stack	Total		0 of 12 720
Durban FPT Container Loading	Total		0 of 7 769
Durban SAFT Container Loading	KwaZulu-Natal	Hard citrus	700
	KwaZulu-Natal	Avocado	150
	Free State	Pome	41
	DIn	Hard citrus	368
	Total		1 259 of 2 253
Port Elizabeth Stack	Sundays River Valley	Hard citrus	2 594
	Patensie	Hard citrus	1 930
	Langkloof	Pome	537
	Total		5 061 of 11 480
Port Elizabeth FPT Container Loading	Eastern Cape Midlands	Soft citrus	43
	Sundays River Valley	Soft citrus	406
	Patensie	Hard citrus	885
	Langkloof	Hard citrus	12
	Total		1 346 of 3 877
Cape Town Stack	Klein Karoo	Pome	167
	GEVV	Pome	4 479
	GEVV	Hard citrus	6 688
	Boland	Pome	182
	Boland	Hard citrus	1 019
	Ceres	Pome	2 127
	Piketberg	Pome	163
	Total		14 825 of 31 390
	Total		0 of 7 200
Cape Town FPT Container Loading	CTfptIn	Hard citrus	73
	Total		73 of 11 200
Cape Town SAFT Container Loading	Total		0 of 3 733
Cape Town Combi Stack	CTS	Pome	7 118
CTCT Container Loading	CTS	Hard citrus	7 707
	CTS	Soft citrus	1 070
	Total		15 895 of 45 840
DCT Container Loading	Total		0 of 103 940
PECT Container Loading	PES	Pome	537
	PES	Hard citrus	4 524
	Total		5 061 of 50 540

Table 5.33: Solution of the minimum cost flow problem (where port capacities have been removed) at the point where the transportation reaches the ports and port cities, where containerised fruit is handled for the winter peak week (week 29 of 2003). All fruit arriving directly from production regions would do so in containers, otherwise it would first pass through harbour cold stores (seen in Table 5.32).

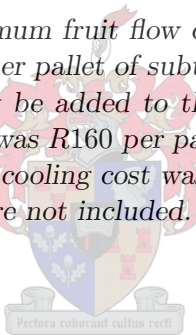
To compare the effects of the changes to the model constraints, the shadow prices of the restrictions ensuring that fruit export was at least the same as during week 29 of 2003 were used to estimate the cost (to the fruit industry as a whole) of exporting a unit of fruit from each region. This is because the shadow price of the constraint indicates by how much the objective function (the sum of the all flow costs) would increase if the constraint were relaxed by a single unit. These comparisons are shown in Table 5.34.

If the capacity of the FPT cold store in Maputo could be increased ten-fold, and the break-bulk loading rate would double; the fresh fruit exporters in the Limpopo, Mpumalanga and Gauteng provinces could save approximately R7.36 million in that week alone. However, this would affect the FPT and SAFT operations in Durban harbour adversely, as they are currently used by the exporters in those regions to export their fruit (mainly hard citrus, because the bulk of the avocados may be exported via Cape Town), as it is currently the closest port to those regions with sufficient capacity for the volume exported in week 29 of 2003.

Region	Fruit Types	Exported Volumes	No Ratio Constraint	With Ratio Constraint	No Port Capacities
Levubu	Hard citrus	6 985	1 875	1 875	1 667
	Soft citrus	27	1 930	2 025	1 723
	Avocado	72	2 537	2 583	2 290
Tzaneen	Hard citrus	13 240	1 794	1 795	1 598
	Avocado	760	2 421	2 501	2 174
Hoedspruit	Hard citrus	3 087	1 753	1 754	1 533
	Soft citrus	24	1 753	1 848	1 546
Ki, Ne and Ma	Hard citrus	12 462	1 765	1 765	1 500
Kiepersol	Avocado	487	2 254	2 393	2 007
Nelspruit	Avocado	520	2 230	2 369	1 990
Po, Mh, Ru and Mi	Hard citrus	5 698	1 684	1 684	1 562
	Soft citrus	322	1 818	1 818	1 657
Pg, Nk and Kz	Hard citrus	8 330	1 615	1 615	1 603
	Soft citrus	84	1 615	1 615	1 603
	Avocado	150	1 942	1 942	1 890
Eastern Cape Midlands	Hard citrus	1 494	1 472	1 469	1 469
	Soft citrus	43	1 473	1 473	1 473
Sundays River Valley	Hard citrus	6 564	1 402	1 399	1 399
	Soft citrus	406	1 402	1 399	1 399
Patensie	Hard citrus	4 185	1 402	1 399	1 399
	Soft citrus	133	1 402	1 399	1 399
Langkloof	Pome	1 422	1 494	1 494	1 494
	Hard citrus	12	1 449	1 449	1 449
Kl, Ge, Bo, Ce and Pi	Hard citrus	7 707	1 320	1 342	1 342
	Soft citrus	1 070	1 338	1 342	1 342
GEVV	Pome	4 479	1 383	1 383	1 383
Boland	Pome	182	1 387	1 387	1 387
Ceres	Pome	2 127	1 446	1 446	1 446
Piketberg	Pome	163	1 463	1 463	1 463
Orange River	Hard citrus	279	1 638	1 689	1 689
Objective function	All	82 722	130 024 915	131 617 740	124 256 230

Table 5.34: Shadow prices of the minimum fruit flow constraints. All values were rounded to nearest Rand. A cooling cost of R120 per pallet of subtropical fruit was assumed. To calculate the estimated cost of export, R40 may be added to the estimated export cost of subtropical fruit to find the cost if the cooling cost was R160 per pallet, or alternatively R80 may be added to find the estimated export cost if the cooling cost was R200 per pallet. The costs of PPECB carton inspections, packing and VAT are not included.

5.7 Chapter Summary

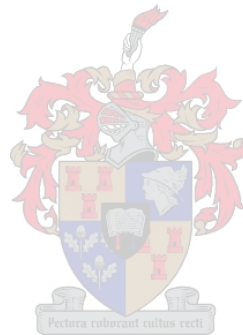


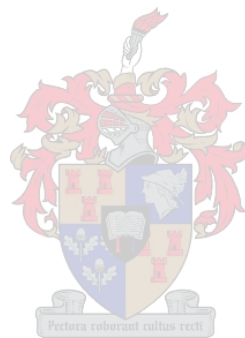
In this chapter, maximum flow and minimum cost flow models were derived for the South African fresh fruit export infrastructure. After a brief literature survey, the formulation of the integer program was presented by focussing on the multicommodity aspect of the problem. The minimum and maximum flow limits were formulated mathematically, as was the restriction on the volume of ambient temperature citrus allowed on a vessel, and the conservation of flow constraints. After a brief discussion on the sizes of the resulting models, a method for solving the multicommodity flow problem was discussed. Thereafter, the solution to the maximum flow problem was presented. This was followed by further studies into the effects of increasing the weight placed on the flow of single fruit types in the objective function.

The maximum flow model is also used to determine the maximum possible fruit volumes that may pass through the South African fruit export infrastructure. Firstly, the country was regarded as a whole, and then the flow of citrus through Durban was allowed an additional weight, so as to promote export via Durban in the solution to the maximum flow problem. This was done to mimic the pressure on the infrastructure in Durban harbour in the winter, when hard citrus export volumes peak. Two variations of the minimum cost flow problem for the summer peak week were considered. One allowed fruit to be exported from any of the four fruit-exporting harbours, while the other restricted the export of fruit to be via Cape Town only. The second variation was included to mimic the pressure applied on the port of Cape Town during the summer months. The winter peak week minimum cost flow problem had three variations. The

first allowed any volume of fruit (below the maximum capacity) to be exported by container on a conventional vessel. The second stipulated that there be a ratio of fruit that may go on deck in containers, relative to the volume of fruit inside each vessel. The final variation set all capacities equal to a very large number, so that the cheapest harbour for export would be chosen by the modelling package for the producers.

For the summer months, it was found that the cost of only exporting fruit via Cape Town was approximately *R*1.78 million, than if fruit could be exported via any port. It was found for the winter months that, due to the restrictions on the cooling capacity in Maputo, most fruit from the north-eastern region of the country should be exported via Durban in the winter months. However, large cost savings could be incurred for the peak week (approximately *R*7.36 million) if the capacity of the FPT terminal in Maputo were expanded considerably.





Chapter 6

Conclusion

“No human investigation can be called real science if it cannot be demonstrated mathematically.”

– Leonardo da Vinci [307]

This purpose of this chapter is to summarise the work contained in this thesis, and to discuss the results obtained in Chapter 4 and 5. Some recommendations are also made regarding the use and expansion of the fruit industry and possible future work is outlined.

6.1 Thesis Summary

The thesis opened with an introduction to the *Fruit Logistics Infrastructure Capacity Optimisation Study*. The main role-players in the study were introduced, followed by a discussion of the structure of the study, consisting of a summary of the purposes of the four phases of the project. The objectives of the thesis were presented in the first chapter, along with a layout of the thesis structure.

The second chapter was dedicated to a description of the fruit industry in some detail, so as to aid the reader in gaining an understanding of the inner workings of the fruit industry. The size of the fruit industry was put into context, and the exported fruit types were discussed. The regions from which the fruit types originate were shown in a series of figures. This was followed by discussions on the fruit export infrastructure; including road and rail transport, shipping and the costs involved with transportation and handling at terminals (thereby achieving Thesis Objective I, as stated in §1.2).

The third chapter provided the reader with an insight into how the capacities of the infrastructure components were found, as well as the fruit export volumes. A description of the data gathering process for finding the packhouse and cold store capacities was given at the beginning of the chapter, followed by a number of tables showing the capacities of these infrastructure components (thereby achieving Thesis Objective II, as stated in §1.2). Fruit production and export volumes from the various regions, as well as transportation volumes, were estimated and presented (thereby achieving Thesis Objective IV, as stated in §1.2). The *Fruitlog Scenario Model* was presented to aid the reader in understanding the approach taken towards the port capacity calculations. Once the terminal capacities had been found (thereby achieving Thesis Objective III, as stated in §1.2), the estimated utilisation of the ports in 2003 was presented to

show during which weeks of the year the exported fruit volumes place the most pressure on the infrastructure (thereby achieving Thesis Objective V, as stated in §1.2).

Graph theoretic methods for finding the maximum flow and minimum cost flow of fruit through the infrastructure were discussed at some length in Chapter 4. The chapter opened with a formulation of the graph model and the assumptions made about the infrastructure so that it could be modelled as a graph. The introduction to maximum flow problems included references to the very beginning of maximum flow problems according to the literature. A discussion on the evolution of algorithms for maximum flow problems was followed by a list of applications of the maximum flow problem. *Ford's algorithm* and the *Shortest Augmenting Path algorithm* were presented and the *Shortest Augmenting Path algorithm* was shown indeed to find a maximum flow. This was followed by a brief history of the minimum cost flow problem, its associated algorithmic development and some applications of the problem. The *Successive shortest path algorithm* was then presented and shown to find a (single-commodity) minimum cost flow in a network. Finally, the results of the maximum flow and minimum cost flow problems were summarised (thereby achieving Thesis Objectives VI and VII, as stated in §1.2).

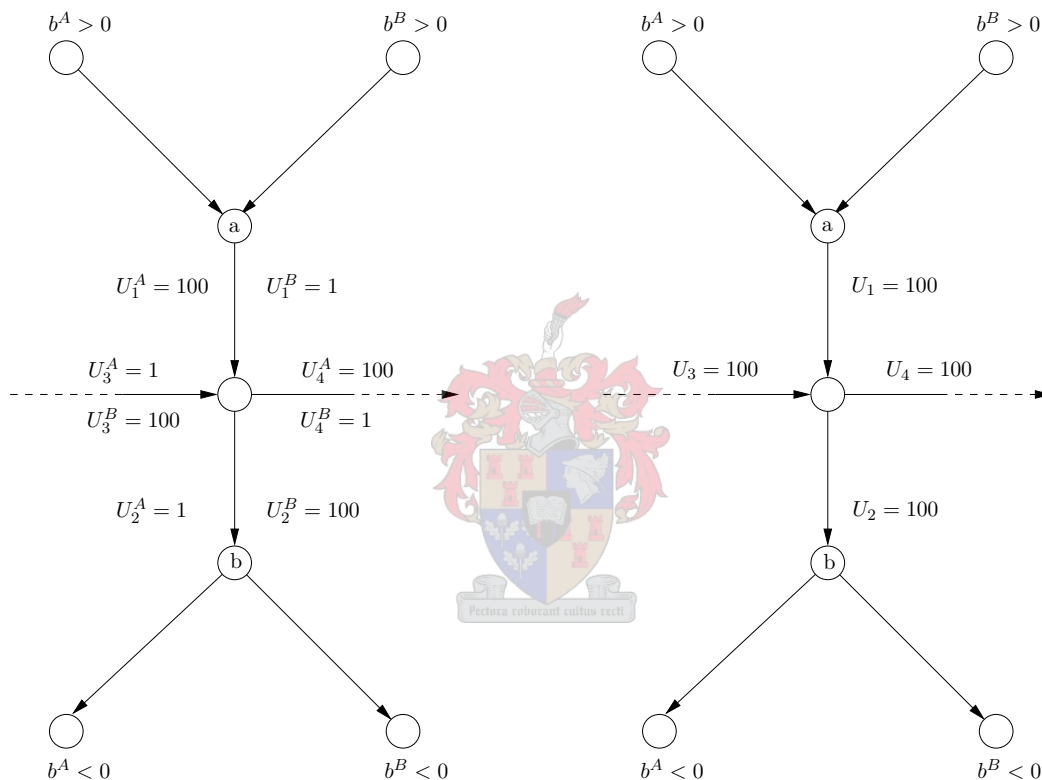
The fifth chapter is dedicated to a multicommodity approach towards finding maximum flows and minimum cost flows. After a brief literature survey, a mathematical programming model was formulated for the maximum flow problem and the minimum cost flow problem (thereby achieving Thesis Objective VIII, as stated in §1.2). The maximum flow and minimum cost flow solutions were summarised for various scenarios. These scenarios included the two peak weeks (summer peak during week 7 and winter peak during week 29 of 2003), as well as additions and omissions to the standard models. Some of the omissions included the removal of the port infrastructure capacities in the minimum cost flow models, while additions included solving the maximum flow problems when selected flow coefficients in the objective function were allowed to be higher than for other flows.

6.2 Comparison of Solutions

If the solutions of the graph theoretic (Chapter 4) and mathematical programming models (Chapter 5) are compared, the significance and necessity of the multicommodity flow approach becomes evident. The maximum flow solution for the summer scenario yielded 102 773 pallets per week for the graph theoretic model, compared to 100 523 pallets per week found for the mathematical programming model. For the winter scenario, the graph theoretic model yielded a maximum flow of 151 346 pallets per week, while the mathematical programming result was a maximum flow of 150 004 pallets of fruit in a week. The discrepancies may be attributed to the fact that only citrus fruit may be transported at ambient temperatures (not having been cooled to the correct transport temperature in a cold store). With the multicommodity flow approach, only a flow of hard citrus pallets may pass through the unrefrigerated transportation arcs from packhouses. However, without the multicommodity flow approach, any fruit that shares a packhouse with hard citrus fruit may make use of the unrefrigerated transportation arcs.

The (single-commodity) graph theoretic approach provides a good upper bound on the maximum flow of fruit in this case. However, there are two weaknesses of this approach. First, although each fruit type was considered separately in the maximum flow models and great care was taken to restrict the flow of each fruit type into the packhouses according to the capacities of each fruit in the the packhouses (there were cases, for example, where the maximum capacity for hard citrus was not the same as the maximum capacity for pome fruit), once the fruit

flow entered a node representing a packhouse packing a variety of fruits, it became impossible to know which fruit was shipped directly from the building: all the fruit types merged into one commodity at that point. The second drawback of the single commodity graph theoretic approach taken in this thesis, is the danger illustrated in Figure 6.1. When converting the multicommodity upper bounds to single commodity upper bounds, the upper bound for the sum of the fruit types becomes 100 (if the constraint $x_i^A + x_i^B \leq 100$ holds for all arcs i , where x_i^k is the flow of commodity k along arc i) in Figure 6.1. This would allow, say, 100 units of commodity A to flow along the path $a - b$, which is not possible in the multicommodity approach. If there were a large difference between the arc weights for commodities A and B , the results would be skewed significantly — thus highlighting the necessity of a multicommodity flow approach.



(a) Graphical representation of a multicommodity flow problem.

(b) Graphical representation of a single-commodity conversion from the multicommodity flow problem in (a).

Figure 6.1: A danger of not considering the multicommodity flow approach. Assume the maximum flow along any arc is 100 units. Although the maximum flow of each commodity along the path $a - b$ is 1, a single commodity approach would yield a maximum flow of 100, of an unknown commodity, along the path $a - b$. Here b^k is the supply of commodity k at a node and U_x^k is the upper bound on the flow of commodity k along arc x .

If the maximum flow solution for Scenario 3 in Chapter 5 is compared to the actual flow in 2003 (as shown in Table 5.20), it becomes clear that Durban port alone can handle the entire volume of hard citrus exported in week 29 of 2003, and additionally even the entire soft citrus volume exported in that peak week. It is most likely that the pome fruit volume and most of the volume of avocados would have been exported via Cape Town harbour, thus leaving

Durban with a theoretical spare capacity of 6941 pallets in that peak week (week 29 of 2003). This does not even take into consideration that the hard citrus harvested in the Western Cape would, most likely, have been exported via the port of Cape Town and the hard citrus harvested in the Eastern Cape would probably have been exported via Port Elizabeth. Thus, it would seem, that the port of Durban should be able to handle all export during the peak week of the season. However, FE van Dyk [291] reports that the feeling amongst many role-players was that Durban could not handle the export volumes required of it. In this context it is important to recall that the maximum capacities were calculated for ideal weather conditions and that no unforeseen circumstances such as, for example, vessel delays were taken into account. Although adverse weather conditions play a role in Cape Town, this does not play a significant role in the case of Durban, as Durban has very little winter rainfall (when the peak volumes of citrus are exported) and the wind is rarely strong.

Some terminal operators reported that they felt the calculated capacities were accurate and within their reach [291]. The most likely reason for the terminals not achieving the calculated capacities lies with exporters and transporters. It often happens that port operators have to wait for vehicles to arrive, as they do not arrive within a time window the role-players decide on. This leads to inefficiencies and an overall inability of the operators to achieve the stated loading volumes. Information from the FPT terminal in Durban [108] indicates that 53% of all delays at Durban's FPT during 2003 were due to the terminal waiting for vehicles with fruit due to be loaded. Thus, it would seem that producers and exporters can improve their own perception of port bottlenecks by ensuring that fruit arrives at ports timeously.

The results for the summer scenario indicate that the port of Cape Town would have had no problem handling the volumes exported during the summer peak week in 2003. The maximum flow solution of 100 523 pallets per week is comfortably more than the 40 239 pallets that were exported during the summer peak week of 2003. This indicates that the bottlenecks experienced by exporters and transporters are not caused by a lack of logistics infrastructure capacity. For example, during an excursion to the container terminal during what should have been a very busy time (close to stack closing), there were periods where many vehicles were queued for the two inspection points (A-check and P-check), and during other times there would be no vehicles and the workers at the container terminal would be idle. However, there were troubling reports from vehicle drivers about complacency and inefficiencies inside the terminal during certain periods. Unfortunately, these complaints could not be verified. However, Cape Town is known to experience very windy days, which leads to reduced loading capacities¹ in summer, while the winter rainfall decreases efficiency in the FPT and SAFT, which have to load exposed pallets. In contrast, containers are waterproof and can thus be loaded during periods of rain.

Comparing the minimum cost flow solutions of the (single commodity) graph theoretic and (multicommodity) mathematical programming models for the summer peak scenarios, reveals that the graph theoretic method gives good approximations, but is sometimes inaccurate. For the subtropical fruit cooling cost of R120, for example, the graph theoretic approach rendered

¹The figures arrived at for the loading rates at the terminals assume no interference that would hinder the efficiency of the terminals. Some examples of such interference include weather as mentioned in the main text (high wind speeds and/or rain may slow down loading, or bring it to a complete halt), labour disruption (striking or unproductive workers can prevent 100% efficiency), large volumes of traffic (this could prevent vehicles from arriving at the terminals when scheduled, disrupting the loading process which requires the pallets/containers to be placed in a certain pattern), incorrect documentation (fruit pallets may then not arrive at the berth on time), fruit that is not on temperature, cold stores not being able to load fruit fast enough in the correct sequence (the large cold rooms in some cold stores lead to difficulties in finding the exact pallets that should be transported to the harbour), and equipment breakdown (resulting in a direct loss of capacity). Inevitably at least one of these may occur during the same week, which would reduce the capacities significantly.

Scenario	Graph Theoretic	Mathematical Programming	Difference
Standard summer (R120)	71 509 814	73 522 006	2 012 191
Standard summer (R160)	71 521 073	73 579 886	2 058 812
Standard summer (R200)	71 531 113	73 637 766	2 106 652
Summer: CT only (R160)	73 237 235	75 442 432	2 205 197
Summer: Unrestricted (R160)	70 547 328	73 579 886	3 032 557
Standard winter (R120)	120 244 670	131 617 741	11 373 071
Standard winter (R160)	120 317 150	131 697 301	11 380 151
Standard winter (R200)	120 384 973	131 776 861	11 391 888
Winter: Unrestricted (R160)	118 066 732	124 256 230	6 189 498

Table 6.1: A comparison of the total minimum cost flow solutions indicating inadequacy of adopting a single-commodity flow approach instead of a multicommodity flow approach. Values in brackets indicate the cost assumed for subtropical cooling. All values are given in South African Rands.

a result of R71.5 million per week while the mathematical programming approach resulted in a minimum cost flow of R73.5 million per week — a discrepancy of R2 million. However, for the winter scenario, the graph theoretic total minimum cost was R120.2 million per week, while the corresponding mathematical programming result was R131.6 million per week — a significant difference of R11.4 million. The differences in total export cost for equivalent models are summarised for the different scenarios in Table 6.1.

A significant problem resulting from the single commodity approach is that one cannot consider the different costs for the fruit cooling or shipping. For example, some cold stores cool both citrus and stone fruit, or citrus and subtropical fruit. Thus, a single cooling cost is used in the model to represent the cooling of all fruit types, although the difference between the fruit-specific cooling costs may be as high as R125 per pallet. In the single commodity models representing the summer peak week, the larger of the values was always used, while the cooling value for citrus was used in the single commodity models, as the volumes for citrus were much larger than for any other fruit. Another problem is that, as commodity types cannot be identified, volumes of mangoes may be assigned to arcs representing transport on unrefrigerated vehicles or trains from a packhouse that also packs hard citrus fruit. Once fruit volumes have passed a packhouse node at which two or more fruit types are packed, it is impossible to discern how much of the volume is one commodity, and how much volume is another commodity. At the ports it is important to distinguish between fruit-types that are containerised: one can neither assign subtropical fruit to CA containers nor assign the other fruit types to RA containers, which are cheaper to ship. These are some of the reasons for the discrepancies between the single-commodity and multicommodity flow model results.

One problem experienced during the minimum cost flow modelling was that the ratio of containerised fruit to break-bulk fruit loaded at the FPT and SAFT terminals could not be set to be a specific value for the graph theoretic models. It was easier to achieve this in the linear programming models by the addition of an additional constraint (as described in §5.6.2), ensuring that this ratio would not be violated. If this ratio were to be violated, it would mean that containers of fruit would be loaded onto empty conventional vessels. In those models in which the ratio was not included, large volumes of fruit were exported in containers on conventional vessels due to the cheaper terminal handling charges and shipping costs — another reason for the cheaper graph theoretic solutions.

It is thus clearly both important and beneficial to adopt a multicommodity approach for the purpose of examining the fruit export infrastructure of South Africa. However, there was one significant advantage to the graph theoretic approach. While solving the maximum flow models, it became clear that the estimated fruit exported volumes from some regions were larger than the capacity of the infrastructure in that region, indicating incomplete packhouse and/or cold store capacity data. When programming the *Successive shortest path algorithm* for finding the minimum cost flow problem, it was possible to allow some supply at a node to remain behind if there was insufficient infrastructure to allow fruit flow to reach the sink nodes. The remaining fruit would thus indicate by how much the packhouse or cold store capacities would have to be increased to allow all fruit through. This was not possible with the mathematical programming approach (without incurring the cost of a large number of sensitivity analysis trials, or adding a large number of additional variables and/or constraints), as an infeasibility would result when the exported volume was greater than the infrastructure capacity. Thus, after the minimum cost flow problem had been solved using the graph theoretic method, the mathematical programming models could be adjusted accordingly to avoid infeasibility.

Thesis Objective IX, as stated in §1.2, has been achieved in this section.

6.3 Suggestions and Recommendations

It is clear from the solutions to the maximum flow problems that there is sufficient fruit export infrastructure for the volumes that were exported during 2003. Even the peak week in 2003 (week 29), which saw a larger citrus crop than in 2004 and which is larger than the crop expected in 2005 [94], did not produce a volume of citrus (hard citrus and soft citrus combined) that is too large for the capacity of the combined infrastructure in Durban. Thus, it would appear that many bottlenecks experienced by role-players in the export industry are either only perceived bottlenecks, or are due to inefficiencies (the terminals waiting for fruit). An example of perceived bottlenecks were experienced during a visit to the Cape Town container terminal during a period close to the close of the stack (usually the busiest period [291]). Vehicles would often arrive in batches, which resulted in long queues (hence a bottleneck according to those there at the time), but once these vehicles had passed the various check points, there were periods during which the container terminal workers would be idle.

Implementing a strict programme with penalties for latecomers to schedule the arrival of vehicles transporting fruit may resolve many of these perceived bottlenecks. Such a schedule would also reduce the time terminal operators have to wait for fruit to arrive for loading, which is a significant problem at some terminals [108]. This is especially problematic when fruit is to be loaded directly onto the vessel, as fruit must be loaded onto a vessel in a certain order and pattern so as to minimise and increase the speed of the handling at the destination port. Thesis Objective Xa, as stated in §1.2, has thus been achieved.

Upgrading the FPT terminal in Maputo to have a larger handling capacity would have a significant impact on the cost effectiveness of fruit export from the Limpopo and Mpumalanga provinces. However, it is acknowledged that the transport cost for fruit to Maputo may not be as estimated, and that there may not be enough vessels passing through Maputo to handle increased volumes of fruit. One negative impact of implementing this suggestion would be that FPT in Durban may suffer from a financial point of view. Only fruit from Swaziland, Pongola, Nkwalini, KwaZulu-Natal Midlands and Free state regions would use this port, reducing the export volumes to such an extent that the operation may become unprofitable.

If the export growth were to realise as estimated (in Tables 3.3 – 3.9), Durban would be under considerable pressure in 2010. If one takes the growth in export into account, an estimated 101 753 pallets per week of hard citrus would have to be exported in 2010. However, it is made clear in Table 3.4 that the largest exporting regions have Maputo as their nearest port, while the other large regions are the Eastern Cape and on the West Coast, closest to the ports of Port Elizabeth and Cape Town respectively. It is thus unlikely that the fruit would be exported via Durban. This evidence does give more support to the conclusion that the infrastructure in Maputo should receive an upgrade to enlarge capacity, if indeed, infrastructure expansions were to be considered. However, there are some reservations regarding exporting via Maputo: while performing the packhouse and cold store capacity survey, some role-players complained about long queues at the Mozambique/South African border, for example. It is also not known whether there is space available in the port for further expansion, nor is it known at what cost the infrastructure could be upgraded.

Thesis Objective Xb, as stated in §1.2, has been achieved in this section.

6.4 Possible Future Work

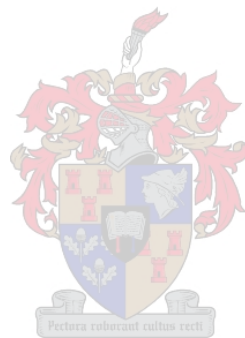
The fruit export industry is a very dynamic industry with on-going expansions to infrastructure, particularly in the ports. Thus it would be beneficial to update the fruit flow volumes every year, or as the port infrastructure changes. With better fruit traceability, fruit export volume estimates may also improve, allowing for updated and more realistic minimum cost flow models. This, of course, would require an update to the handling and transport cost estimations, as well as estimations of the capacities of packhouses and cold stores.

It is also unfortunate that no more rail costs were available. If one compares the rail cost of transporting unrefrigerated hard citrus to the equivalent road cost (see Table F.1 and Table F.4), the rail cost is always cheaper than the equivalent road cost. Thus, it may be assumed that more knowledge about rail costs could lead to significant changes to the preferred modes of transportation in the minimum cost flow solutions.

Another possibility for future work would be to combine and expand the summer and winter scenario minimum cost flow models to include all 52 weeks of the year, so that annual costs may be found as a function of increasing capacities at the ports.

Also, more research could be done into the inclusion of the possibility of air freight into the models if more was known about prices paid for fruit types at critical times of the year in foreign markets. More research into the availability of vessels and the cost of chartering additional vessels could increase the accuracy of the terminal capacity calculations and allow for a better cost analysis of sending small volumes of fruit to a port.

There are also many other opportunities for further applications of Operations Research techniques in the fruit industry. Minimum cost flow analysis could, for example, be performed for individual exporters, allowing them to make better decisions regarding the choices with respect to transportation modes and terminal usage. Furthermore, applying discrete event simulation techniques on infrastructure components could allow managers to make decisions that lead to greater efficiency and thus larger capacity. This may be necessary to aid in the redesigning of roads and other infrastructure in ports so as to increase vehicle throughput.



References

- [1] AASHTIANI HA & MAGNANTI TL, 1976, *Implementing primal-dual network flow algorithms*, (Unpublished) Technical Report OR 055-76, Operations Research Center, MIT, Cambridge (MA).
- [2] AGROFRESH INC., 2001, *SmartFresh, The Smart Choice — Main Page*, [Online], [Cited 13 May 2004], Available from <http://www.rohmhaas.com/agrofresh>.
- [3] AHUJA RK, 1986, *Algorithms for the minimax transportation problem*, Naval Research Logistics Quarterly, **33**, pp. 725–740.
- [4] AHUJA RK, KODIALAM M, MISHRA AK & ORLIN JB, 1997, *Computational investigations of maximum flow algorithms*, European Journal of Operations Research, **97**, pp. 509–542.
- [5] AHUJA RK, MAGNANTI TL & ORLIN JB, 1991, *Some recent advances in network flows*, SIAM Review, **33(2)**, pp. 175–219.
- [6] AHUJA RK, MAGNANTI TL & ORLIN JB, 1993, *Network Flows — Theory, Algorithms and Applications*, Prentice-Hall, Englewood Cliffs (NJ).
- [7] AHUJA RK, MAGNANTI TL, ORLIN JB & REDDY MR, 1995, *Applications of Network Optimization*, pp. 1–84, in BALL MO, MAGNANTI TL, MONMA CL & NEMHAUSER GL (EDS.), *Handbooks in Operations Research and Management Science, Volume 7: Network Models*, North-Holland, Amsterdam.
- [8] AHUJA RK & ORLIN JB, 1989, *A fast and simple algorithm for the maximum flow problem*, Operations Research, **37**, pp. 748–759.
- [9] AHUJA RK & ORLIN JB, 1991, *Distance-Directed Augmenting Path Algorithms for Maximum and Parametric Maximum Flow Problems*, Naval Research Logistics, **38**, pp. 413–430.
- [10] AHUJA RK & ORLIN JB, 1995, *A Capacity Scaling Algorithm for the Constrained Maximum Flow Problem*, Networks, **25**, pp. 89–98.
- [11] AHUJA RK, ORLIN JB, SHARMA P & SOKKALINGAM PT, 2002, *A network simplex algorithm with $O(n)$ consecutive degenerate pivots*, Operations Research Letters, **30**, pp. 141–148.
- [12] AHUJA RK, ORLIN JB & TARJAN RE, 1989, *Improved time bounds for the maximum flow problem*, SIAM Journal of Computing, **18**, pp. 939–954.

- [13] ALON N, 1990, *Generating pseudo-random permutations and maximum flow algorithms*, Information Processing Letters, **35**, pp. 201–204.
- [14] ANDERSON FOREST PRODUCTS INC, 2000, *AFP, Pallets*, [Online], [Cited 19 May 2003], Available from <http://www.afp-reelsandpallets.com>.
- [15] ARATA K, IWATA S, MAKINOK & FUJISHIGE S, 2002, *Locating Sources to Meet Flow Demands in Undirected Networks*, Journal of Algorithms, **42**, pp. 54–68.
- [16] ARGONNE NATIONAL LABORATORY, 2005, *NEOS Server for Optimization*, [Online], [Cited 7 April 2005], Available from <http://www-neos.mcs.anl.gov>.
- [17] ARGONNE NATIONAL LABORATORY, 2005, *NEOS Server: XpressMP*, [Online], [Cited 16 June 2005], Available from <http://www-neos.mcs.anl.gov/neos/neos-5/Web/solvers/milp:XpressMP/MPS.html>.
- [18] ARISAWA S & ELMAGHRABY SE, 1977, *The ‘hub’ and ‘wheel’ scheduling problems*, Transportation Science, **11**, pp. 124–146.
- [19] ASSAD AA, 1978, *Multicommodity Network Flows — A Survey*, Networks, **8**, pp. 37–91.
- [20] ASSAD AA, 1980, *Models for rail transportation*, Transportation Research **14A**, pp. 205–220.
- [21] ASSAD AA, 1980, *Solving linear multicommodity flow problems*, in Proceedings of the IEEE International Conference on Circuits and Computers, pp. 157–161.
- [22] AUTOMOBILE ASSOCIATION OF SOUTH AFRICA, 1999, *South Africa*, Map of South Africa last updated 21 May 1999, Available by request from the Automobile Association of South Africa.
- [23] BACHARACH M, 1966, *Matrix rounding problems*, Management Science, **9**, pp. 732–742.
- [24] BANISTER B & BERECHMAN Y, 2001, *Transport investment and the promotion of economic growth*, Journal of Transport Geography, **9**, pp. 209–218.
- [25] BARAHONA F & TARDOS É, 1989, *Note on Weintraub’s minimum cost circulation algorithm*, SIAM Journal on Computing, **18**, pp. 579–583.
- [26] BARR RS & TURNER JS, 1981, *Microdata file merging through large scale network technology*, Mathematical Programming Study, **15**, pp. 1–22.
- [27] BARTHOLDI JJ, ORLIN JB & RATLIFF HD, 1980, *Cyclic scheduling via integer programs with integer ones*, Operations Research, **28**, pp. 1 074–1 085.
- [28] BELFORD PC & RATLIFF HD, 1972, *A network-flow model for racially balancing schools*, Operations Research, **20**, pp. 619–628.
- [29] BELLMORE M, BENNINGTON G & LUBORE S, 1971, *A multivehicle tanker scheduling problem*, Transportation Science, **5**, pp. 36–47.
- [30] BERGE C & GHOUILA-HOURI A, 1962, *Programming, Games and Transportation Networks*, Wiley, New York.
- [31] BERGE M & HOPPERSTAD C, 1993, *Demand-driven dispatch – a method for dynamic aircraft capacity assignment, models and algorithms*, Operations Research, **41(1)**, pp. 153–168.

- [32] BERRISFORD HG, 1960, *The economic distribution of coal supplies in the gas industry: An application of the linear programming transport theory*, *Operations Research Quarterly*, **11**, pp. 139–150.
- [33] BERTSEKAS DP & TSENG P, 1988, *Relaxation methods for minimum cost ordinary and generalized network flow problems*, *Operations Research*, **36(1)**, pp. 93–114.
- [34] BLAND RG & JENSEN DL, 1992, *On the computational behaviour of a polynomial-time network flow algorithm*, *Mathematical Programming*, **54**, pp. 1–39.
- [35] BODIN LD, GOLDEN BL, SCHUSTER AD & ROWING W, 1980, *A model for the blockings of trains*, *Transportation Research*, **14B**, pp. 115–120.
- [36] BOOYENS G, 2003, *Personal Communication*, CEO of Citrus South Africa.
- [37] BRADLEY G, BROWN G & GRAVES G, 1977, *DESIGN AND IMPLEMENTATION OF LARGE SCALE PRIMAL TRANSSHIPMENT ALGORITHMS*, *Management Science*, **21**, pp. 1–38.
- [38] BRAUN M, 2004, *Truck Operating Costs*, [Online], [Cited 8 February 2005], Available from <http://www.fleetwatch.co.za/magazines/NovDec04/69-TruckOpCosts.htm>.
- [39] BRAUN M, 2005, *Personal Communication*, Transport management, distribution and logistics consultant, Contactable at: maxbraun@iafrica.com.
- [40] BROENS DF, VAN DYK FE & TAVASSZY LA, 2000, *The cold fruit supply chain between South Africa and The Netherlands*, (Unpublished) Report, TNO Inro Department of Logistics and Transport, Delft.
- [41] BRUINTJIES W, 2005, *Personal communication*, Business Assistant for Dispatch at SAFT, Cape Town.
- [42] BUCKLEY JJ & HAYASHI Y, 1994, *Fuzzy genetic algorithm and applications*, *Fuzzy Sets and Systems* **61(2)**, pp. 129–136, [Online], [Cited 11 March 2005], Abstract available from: <http://www.sciencedirect.com>
- [43] BUSAKER RG & GOWEN PJ, 1961, *A procedure for determining minimal-cost network flow patterns*, (Unpublished) ORO Technical Report 15, Operations Research Office, Johns Hopkins Office, Baltimore (MD).
- [44] CABOT AV, FRANCIS RL & STARY MA, 1970, *A network flow solution to rectilinear distance facility location problem*, *AIIE Transactions*, **2**, pp. 132–141.
- [45] CAI X, SHA D & WONG CK, 2001, *Time-Varying minimum cost flow problems*, *European Journal of Operations Research*, **131**, pp. 352–374.
- [46] CAI X, SHA D & WONG CK, 2001, *Time-Varying Universal Maximum Flow Problems*, *Mathematical and Computer Modelling*, **33**, pp. 407–430.
- [47] CAPPANERA P & G GALLO, 2004, *A Multicommodity Flow Approach to the Crew Rostering Problem*, *Operations Research*, **52(4)**, pp. 583–596.
- [48] CENTRE FOR INTERNATIONAL AGRICULTURAL MARKETING AND DEVELOPMENT, 2002, *Key Citrus Industry Statistics 2001*.
- [49] CENTRE FOR INTERNATIONAL AGRICULTURAL MARKETING AND DEVELOPMENT, 2002, *Key Deciduous Fruit Statistics 2001*.

- [50] CENTRE FOR INTERNATIONAL AGRICULTURAL MARKETING AND DEVELOPMENT, 2003, *Key Deciduous Fruit Statistics 2002*.
- [51] CHAMLET LG, FRANCIS RL & SAUNDERS PB, 1982, *Network models for building evacuation*, Management Science, **28**, pp. 86–105.
- [52] CHANAS S, DELGADO M, VERDEGAY JL & VILA MA, 1995, *Fuzzy optimal flow on imprecise structures*, European Journal of Operational Research, **83(3)**, pp. 568–580.
- [53] CHANAS S & KOŁODZIEJCZYK W, 1981, *Maximum flow in a network with fuzzy arc capacities*, Fuzzy Sets and Systems, **8(2)**, pp. 165–173, [Online], [Cited 24 March 2005], Abstract available: <http://www.sciencedirect.com>.
- [54] CHARDAIRE P & LISSER A, 2002, *Minimum cost multicommodity flow*, pp. 199–220, in PARDALOS PM & RESENDE M (EDS.), *Handbook of Applied Optimization*, University of Florida, Gainesville and Resende, M.G.C., AT & T Laboratories, New Jersey.
- [55] CHASSIS KING, 2003, *Containers For Sale*, [Online], [Cited 7 February 2005], Available from <http://www.chassisking.com/containersforsale.php>.
- [56] CHERIYAN J & HAGERUP T, 1995, *A randomized maximum-flow algorithm*, SIAM Journal on Computing, **24(2)**, pp. 203–226, [Online], [Cited 26 March 2005], Abstract available: <http://epubs.siam.org/sam-bin/dbq/article/22152>. Originally appeared in *Proceedings of the 30th IEEE Conference on the Foundations of Computer Science*, pp. 118–123.
- [57] CHERIYAN J, HAGERUP T & MEHLHORN K, 1990, *Can a maximum flow be computed in $O(nm)$ time?*, Proceedings of the 17th International Colloquium on Automata, Languages and Programming, pp. 235–248.
- [58] CHERKASSKY RV, 1977, *Algorithm of construction of maximum flow in networks with complexity $O(V^2E^{1/2})$ operations*, Mathematical Methods of Solution of Economical Problems, **7**, pp. 112–115.
- [59] CHEN H & DEWALD CG, 1974, *A generalized chain labeling algorithm for solving multi-commodity flow problems*, Computers and Operations Research, **1**, pp. 437–465.
- [60] CHEN YL, 1995, *The minimal average cost flow problem*, European journal of Operations Research, **81**, pp. 561–570.
- [61] CHESHIRE M, MCKINNON KIM & WILLIAMS HP, 1984, *The efficient allocation of private contractors to public works*, Journal of the Operational Research Quarterly, **35**, pp. 705–709.
- [62] CHOPRA S, GILBOA I & SASTRY ST, 1998, *Source sink flows with capacity installation in batches*, Discrete Applied Mathematics, **85**, pp. 165–192.
- [63] CHRISTIAN C, 2003, *Personal Communication*, Key Account Manager at Cape Town Container Terminal
- [64] CITRUS GROWERS ASSOCIATION, 2002, *FPEF – List of Members Exporters Forum*, [Online], [.pdf file last updated 15 October 2002], [Cited 24 March 2003], Available from <http://www.cga.co.za>

- [65] CLARKE S & SURKIS J, 1968, *An operations research approach to racial desegregation of school systems*, Socio-Economic Planning Sciences, **1**, pp. 259–272.
- [66] COUNCIL FOR SCIENTIFIC AND INDUSTRIAL RESEARCH – TRANSPORTEK, 2004, *Fruit Logistics Scenario Analyses*, (Unpublished) Contract Report CR-2004/19.
- [67] COVA TJ & JOHNSON JP, 2003, *A network flow model for lane-based evacuation routing*, Transportation Research Part A, **37**, pp. 579–604.
- [68] COX LH & ERNST LR, 1982, *Controlled rounding*, INFOR, **20**, pp. 423–432.
- [69] CRAFFORD S, 2003, *Personal Communication via FE van Dyk*, General Manager — South African Mango Growers' Association, Contactable at: stephan@mango.co.za.
- [70] CRAINIC T, FERLAND JA & ROUSSEAU JM, 1984, *A tactical planning model for rail freight transportation*, Transportation Science, **18**, pp. 165–184.
- [71] CREMEANS JE, SMITH RA & TYNDALL GR, 1970, *Optimal multicommodity network flows with resource allocation*, Naval Research Logistics Quarterly, **17**, pp. 269–280.
- [72] CROUCH IJ, DE KOCK VA & TAYLOR MA, 2002, *SmartFreshTM: The Latest Innovation in Fruit Storage. Some preliminary results on South African pome and stone fruit.*, [Online], [Cited 13 May 2004], Available from <http://www.dfptresearch.co.za>.
- [73] CZYZYK J, MEHROTRA S, WAGNER M & WRIGHT SJ, 1997, *PCx User Guide (Version 1.1)*, Technical Report OTC 96/01, Optimization Technology Center, [Online], [Ghostscript file last updated 3 November 1997, [Cited 17 March 2005], Available from <http://www.mcs.anl.gov/otc/Tools/PCx/>.
- [74] CZYZYK J, MESNIER MP & MORÉ JJ, 1998, *The NEOS Server*, IEEE Computational Science and Engineering, **5**, pp. 68–75.
- [75] DAFERMOS S & NAGURNEY A, 1984, *A network formulation of market equilibrium problems and variational inequalities*, Operations Research Letters, **5**, pp. 247–250.
- [76] DANIEL RC, 1973, *Phasing out capital equipment*, Operations Research Quarterly, **24**, pp. 113–116.
- [77] DANTZIG GB, 1951, *Application of the simplex method to a transportation problem*, pp. 359–373, in KOOPMANS TC (ED.), *Activity Analysis and Production and Allocation*, Wiley, New York.
- [78] DANTZIG GB, 1962, *Linear Programming and Extensions*, Princeton University Press, Princeton (NJ).
- [79] DANTZIG GB & FULKERSON DR, 1954, *Minimizing the number of tankers to meet a fixed schedule*, Naval Research Logistics Quarterly, **1**, pp. 217–222.
- [80] DANTZIG GB & WOLFE P, 1960, *Decomposition principle for linear programs*, Operations Research, **8**, pp. 101–111.
- [81] DANTZIG GB & WOLFE P, 1961, *The decomposition method for linear programming*, Econometrica, **29**, pp. 767–778.
- [82] DE KLERK A, 2005, *Personal Communication*, Cold Store Manager at Haffenden Groves cold store, Contactable at: +27 (0)13 773 3851.

- [83] DE KLERK N, 2003, *Personal Communication*, Public Relations Officer at Fresh Produce Terminals (Durban), Contactable at: Neville_Klerk@fpt.co.za.
- [84] DEARING PM & FRANCIS RL, 1974, *A network flow solution to a multifacility minimax location problem involving rectilinear distances*, *Transportation Science*, **8**, pp. 126–141.
- [85] DEMGENSKY I, NOLTEMEIER H & WIRTH H-C, 2002, *On the flow cost lowering problem*, *European Journal of Operational Research*, **137**, pp. 265–271
- [86] DENARDO EV, ROTHBLUM UG & SWERSEY AJ, 1988, *A transportation problem in which costs depend on the order of arrival*, *Management Science*, **34**, pp. 774–783.
- [87] DINIC EA, 1970, *Algorithm for solution of a problem of maximum flow in networks with power estimation*, *Soviet Mathematics Doklady*, **11**, pp. 1 277–1 280.
- [88] DIRICKX YMI & JENNERGREN LP, 1975, *An analysis of the parking situation in the downtown area of West Berlin*, *Transportation Research*, **9**, pp. 1–11.
- [89] DOLAN E, 2002, *The NEOS Server 4.0 Administrative Guide*, Technical Memorandum ANL/MCS-TM-250, Mathematics and Computer Science Division, Argonne National Laboratory, [Online], [Cited 7 April 2005], Available from <http://www-neos.mcs.anl.gov/neos/faq.html>.
- [90] DORSEY RC, HODGSON TJ & RATLIFF HD, 1975, *A network approach to a multifacility, multi-product production scheduling problem without backordering*, *Management Science*, **21**, pp. 813–822.
- [91] DIAMOND P, 2001, *A fuzzy max-flow min-cut theorem*, *Fuzzy Sets and Systems*, **119(1)**, pp. 139–148.
- [92] DULCE MMO SA, 2003, *Dulce MMO, finest quality fruit from Costa Rica*, [Online], [Cited 19 May 2003], Available from <http://www.dulce.nl>.
- [93] DU TOIT R, 2003, *Personal Communication*, Manager at Hexkoel cold store.
- [94] DU BOIS A, 2005, *Personal Communication*, Transport Manager at FPT, Cape Town, Contactable at: Albertyn_DuBois@fpt.co.za.
- [95] EDMONDS J & JOHNSON EL, 1973, *Matching, Euler tours and the Chinese postman*, *Mathematical Programming*, **5**, pp. 88–124.
- [96] EDMONDS J & KARP RM, 1972, *Theoretical improvements in algorithmic efficiency for network flow problems*, *Journal of ACM*, **19**, pp. 248–264.
- [97] EGIDI N & MAPONI P, 2005, *A class of network optimization methods for planar grid generation*, *Applied Numerical Mathematics*, **52**, pp. 363–379.
- [98] ESAU LR & WILLIAMS KC, 1966, *On teleprocessing system design II*, *IBM Systems Journal*, **5**, pp. 142–147.
- [99] ELIAS P, FEINSTEIN A & SHANNON CE, 1956, *Note on the maximum flow through a network*, *IRE Transactions on Information Theory*, **IT-2**, pp. 117–119.
- [100] ELLIOT M, 2003, *Personal communication via FE van Dyk*, Manager at Westfalia Marketing.

- [101] ELMAGHRABY SE, 1978, *Activity Networks: Project Planning and Control by Network Models*, Wiley-Interscience, New York.
- [102] ERVOLINA TR & MCCORMICK ST, 1990, *Two strongly polynomial cut cancelling algorithms for minimum cost network flow*, (Unpublished) Technical Report, Faculty of Commerce and Business Administration, University of British Columbia, Vancouver.
- [103] EVANS JR, 1977, *Some network flow models and heuristics for multiproduct production and inventory planning*, AIIE Transactions, **9**, pp. 75–81.
- [104] EVANS JR, 1984, *The factored transportation problem*, Management Science, **30**, pp. 1 021–1 024.
- [105] EVANS JR & MINIEKA E, 1992, *Optimization Algorithms for Networks and Graphs (Second Edition)*, Marcel Dekker, New York.
- [106] FARLEY AR, 1980, *Levelling terrain trees: A transshipment problem*, Information Processing Letters, **10**, pp. 189–192.
- [107] FEDERGRUEN A & GROENEVELT H, 1986, *Preemptive scheduling of uniform machines by ordinary flow techniques*, Management Science, **32**, pp. 341–349.
- [108] FITZMAURICE E, 2005, *Personal communication via FE van Dyk*, Manager Technical Services & IT at FPT Durban.
- [109] FORD LR, 1956, *Network Flow Theory*, (Unpublished) Technical Report P-923, The RAND Corporation, Santa Monica (CA).
- [110] FORD LR & FULKERSON DR, 1956, *Maximum flow through a network*, Canadian Journal of Mathematics, **8**, pp. 399–404.
- [111] FORD LR & FULKERSON DR, 1957, *A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem*, Canadian Journal of Mathematics, **9**, pp. 210–218.
- [112] FORD LR & FULKERSON DR, 1958, *A suggested computation for maximal multicommodity network flows*, Management Science, **5**, pp. 97–101.
- [113] FORD LR & FULKERSON DR, 1958, *Constructing maximum dynamic flows from static flows*, Operations Research, **6**, pp. 419–433.
- [114] FORD LR & FULKERSON DR, 1962, *Flows in Networks*, Princeton University Press, Princeton (NJ).
- [115] FRANCIS RL & WHITE JA, 1976, *Facility Layout and Location*, Prentice Hall, Englewood Cliffs (NJ).
- [116] FRESH PRODUCE EXPORTERS' FORUM, 2003, *Fresh Produce Exporters' Forum*, [Online], [Cited 28 March 2003], Available from <http://www.fpef.co.za>.
- [117] FRESH PRODUCE TERMINALS, 2003, *Deciduous and Exotic Tariffs for FPT Terminals – 2003*, Fruit Exporters, Cape Town, Available by request.
- [118] FRESH PRODUCE TERMINALS, 2003, *Tariffs for Citrus Season 2003*, Available by request.
- [119] FRUIT SOUTH AFRICA, 2003, *Fresh Fruit Exports from South Africa – Handbook 2003*.

- [120] FULKERSON DR, 1961, *A network flow computation for project cost curve*, Management Science, **7**, pp. 167–178.
- [121] FULKERSON DR, 1961, *An out-of-kilter method for minimal cost flow problems*, SIAM Journal of Applied Mathematics, **9**, pp. 18–27.
- [122] FULKERSON DR & HARDING GC, 1977, *Maximizing the minimum source-sink path subject to a budget constraint*, Mathematical Programming, **13**, pp. 116–118.
- [123] FUJISHIGE S, 1986, *A capacity-rounding algorithm for the minimum cost circulation problem: A dual framework of Tardos' algorithm*, Mathematical Programming, **35**, pp. 298–308.
- [124] FUJISHIGE S, 2003, *A maximum flow algorithm using MA ordering*, Operations Research Letters, **31**, pp. 176–178.
- [125] GABOUW HN, 1985, *Scaling Algorithms for Network Problems*, Journal of Computer and System Sciences, **31**, pp. 148–168.
- [126] GALIL Z, 1980, *An $O(n^{5/3}m^{2/3})$ algorithm for the maximum flow problem*, Acta Infomatica, **14**, pp. 221–242.
- [127] GALIL Z & TARDOS É, 1986, *An $O(n^2(m + n \log n) \log n)$ min-cost flow algorithm*, in Proceedings of the 27th Annual Symposium on the Foundations of Computer Science, pp. 136–146. Full paper in Journal of ACM, **35**, pp. 374–386 (1987).
- [128] GANNON C, GWILLIAM K, LIU Z & MALMBERG-CALVO C, 2002, *Poverty Reduction Strategy Sourcebook: Chapter 22 – Transport*, Edited by J. Klugman, [Online], [Cited 14 May 2003], Available from <http://www.worldbank.org/poverty/strategies/chapters/transport/transprt.htm>.
- [129] GARCIA JM, LOZANO S & CANCA D, 2004, *Coordinated scheduling of production and delivery from multiple plants*, Robotics and Computer-Integrated Manufacturing, **20**, pp. 191–198.
- [130] GAREY MS & JOHNSON DS, 1979, *Computers and Intractability: A guide to the Theory of NP-Completeness*, W.H. Freeman, New York.
- [131] GAVISH B & SCHWEITZER KN, 1979, *$O(n^2)$ algorithms for sensitivity analysis of minimal spanning trees and related subgraphs*, (Unpublished) Working paper 8003, Graduate School of Management, University of Rochester, Rochester (NY).
- [132] GEOFFRION AM & GRAVES GW, 1974, *Multicommodity distribution system design by Benders decomposition*, Management Science, **20**, pp. 822–844.
- [133] GE SEACO SERVICES, 2003, *Genset Clip-on*, [Online], [Cited 9 February 2005], Available from http://www.geseaco.com/GESeaCo/container_fleet/reefers_gscospecs.htm.
- [134] GE SEACO SERVICES, 2005, *Refrigerated containers – specifications*, [Online], [Cited 9 February 2005], Available from http://www.geseaco.com/GESeaCo/container_fleet/reefers_specs.htm.
- [135] GLOVER F, KARNEY D & KLINGMAN D, 1974, *Implementation and computational comparisons of primal, dual and primal-dual computer codes for minimum cost network flow problem*, Networks, **4**, pp. 191–212.

- [136] GLOVER F & KLINGMAN D, 1977, *Network applications in industry and government*, AIIE Transactions, **9**, pp. 363–376.
- [137] GERBER L, 2003, *Personal Communication*, Masters Logistics Student, University of Stellenbosch, Contactable at: leila@sun.ac.za.
- [138] GOLDBERG AV, 1985, *A new max-flow algorithm*, (Unpublished) Technical report MIT/LCS/TM-291, Laboratory for Computer Science, MIT, Cambridge (MA).
- [139] GOLDBERG AV, 1997, *An Efficient Implementation of a Scaling Minimum-Cost Flow Algorithm*, Journal of Algorithms, **22**, pp. 1–29.
- [140] GOLDBERG AV, GRIGORIADIS MD & TARJAN RE, 1991, *Use of dynamic trees in a network simplex algorithm for the maximum flow problem*, Mathematical Programming, **50**, pp. 277–290.
- [141] GOLDBERG AV & KHARITONOV, 1993, *On implementing scaling push-relabel algorithms for the minimum-cost flow problem*, pp. 157–198, in JOHNSON DS & MCGEOCH CC (EDS.), *Network Flows and Matching: First DIMACS Implementation Challenge*, AMS, Providence.
- [142] GOLDBERG AV & TARJAN RE, 1986, *A new approach to the maximum flow problem*, Proceedings of the 18th ACM Symposium on the Theory of Computing, pp. 136–146. Full paper in Journal of ACM, **36**, pp. 873–886 (1988).
- [143] GOLDBERG AV & TARJAN RE, 1987, *Solving minimum-cost circulations by successive approximation*, Proceedings of the 19th ACM Symposium on the Theory of Computing, pp. 7–18. Full paper in Mathematics of Operations Research, **15**, pp. 430–466 (1990).
- [144] GOLDBERG AV & TARJAN RE, 1988, *Finding minimum-cost circulations by cancelling negative cycles*, Proceedings of the 20th ACM Symposium on the Theory of Computing, pp. 388–397. Full paper in Journal of ACM, **36**, pp. 873–886 (1989).
- [145] GOLDBERG AV & TARJAN RE, 1990, *Finding minimum-cost circulations by successive approximation*, Mathematics of Operations Research, **15**, pp. 430–466.
- [146] GOLDEN BL, 1975, *A minimum cost multicommodity network flow problem concerning imports and exports*, Networks, **5**, pp. 331–356.
- [147] GOLDFARB D & HAO J, 1993, *On the maximum capacity augmentation algorithm for the maximum flow problem*, Discrete Applied Mathematics, **47**, pp. 9–16.
- [148] GREENWALD L & DEAN T, 1996, *Package routing in transportation networks with fixed vehicle schedules*, Networks, **27(1)**, pp. 81–93.
- [149] GRIGORIADIS MD, 1986, *An efficient implementation of the network simplex method*, Mathematical Programming Study, **26**, pp. 83–111.
- [150] GRIGORIADIS MD & HSU Y, 1979, *The Rutgers minimum cost network flow subroutines*, SIGMAP Bulletin of the ACM, **26**, pp. 17–18.
- [151] GROBBELAAR R, 2003, *Personal Communication*, Logistics Masters Student, University of Stellenbosch.

- [152] GROPP W & MORÉ JJ, 1997, *Optimization Environments and the NEOS Server*, pp. 167–182 in BUHMANN MD & ISERLES A (EDS.), *Approximation Theory and Optimization*, Cambridge University Press, Cambridge.
- [153] GU Z, JOHNSON E, NEMHAUSER G & WANG Y, 1994, *Some properties of the fleet assignment problem*, *Operations Research Letters*, **15(2)**, pp. 59–71.
- [154] GUPTA SK, 1985, *Linear Programming and Network Models*, Affiliated East-West Press, New Delhi.
- [155] HAGHANI A & OH S-C, 1996, *Formulation and solution of a multi-commodity multi-modal network flow model for disaster relief operations*, *Transportation Research A*, **30(3)**, pp. 231–250.
- [156] HALL M, 1956, *An algorithm for distinct representatives*, *American Mathematical Monthly*, **63**, pp. 716–717.
- [157] HAMBURG SÜD, 2005, *Containers*, [Online], [Cited 8 May 2003], Available from http://www.hamburg-sued.com/Com_cont.htm.
- [158] HARRIS TE & ROSS FS, 1955, *Fundamentals of a Method for Evaluating Rail Net Capacities*, (Unpublished) Research Memorandum RM-1573, The RAND Corporation, Santa Monica (CA).
- [159] HELLBERG TRANSPORT MANAGEMENT, 2004, *Toll Tariffs: with effect 01 March 2004 – Updated 01 March 2004*, [Online], [Cited 29 June 2004], Available from <http://www.htm.co.za>.
- [160] HITCHCOCK FL, 1941, *The Distribution of a Product from Several Sources to Numerous Localities*, *Journal of Mathematical Physics*, **20**, pp. 224–230.
- [161] HOCHBAUM DS & QUEYRANNE M, 2003, *Minimizing a convex cost closure set*, *SIAM Journal of Discrete Mathematics*, **16(2)**, pp. 192–207.
- [162] HOLCROFT C, 2003, *Personal Communication*, Serco Industries, Contactable at: clintonh@serco.co.za.
- [163] IAKOVOU E, DOULIGERIS C, LI H, IP C & YUDHBIR L, 1999, *A maritime global route planning model for hazardous materials transportation*, *Transportation science*, **33(1)**, pp. 34–48.
- [164] HORN WA, 1971, *Determining optimal container inventory and routing*, *Transportation Science*, **5**, pp. 225–231.
- [165] INTERNATIONAL FREIGHT LOGISTICS N.Z. LTD, 2004, *Container Specifications*, [Online], [Cited 7 February 2005], Available from <http://www.ifl.co.nz/container-specs.html>.
- [166] IRI M, 1960, *A new method of solving transportation-network problems*, *Journal of Operations Research Society of Japan*, **3**, pp. 27–87.
- [167] JACOBS WW, 1954, *The caterer problem*, *Naval Research Logistics Quarterly*, **1**, pp. 154–165.
- [168] JEWELL WS, 1957, *Warehousing and distribution of a seasonal product*, *Naval Research Logistics Quarterly*, **4**, pp. 29–34.

- [169] JEWELL WS, 1958, *Optimal flow through networks*, (Unpublished) PhD, Interim Technical Report 8, Operations Research Center, MIT, Cambridge (MA).
- [170] JEWELL WS, 1966, *A Primal-Dual Multicommodity Flow Algorithm*, (Unpublished) Technical Report ORC 66-24, Operations Research Center, University of California, Berkeley.
- [171] JOHNSON TB, 1968, *Optimum pit mine production scheduling*, (Unpublished) Technical report, University of California, Berkeley.
- [172] JOOSTE E, 2005, *Personal communication*, General Manager at Logicool cold store, Picketberg, Contactable at: ericjooste@telkomsa.net.
- [173] KAHN MR, 1979, *A capacitated network formulation for manpower scheduling*, *Industrial Management*, **21**, pp. 24–28.
- [174] KAHN MR & LEWIS DA, 1987, *A network model for nursing staff scheduling*, *Zeitschrift für Operations Research*, **31**, pp. B161–B171.
- [175] KANTOROVICH L, 1942, *On the Translocation of Masses*, *Compt. Rend. (Doklady) Acad. Sci.*, **37**, pp. 199–201.
- [176] KANTOROVICH L & GAVURIN MK, 1949, *The Application of Mathematical Methods in Problems of Freight Flow Analysis*, pp. 110–138, in *Collection of Problems Concerned with Increasing the Effectiveness of Transports*, Publication of the Akademii Nauk SSSR, Moscow-Leningrad.
- [177] KAPLAN S, 1973, *Readiness and the optimal redeployment of resources*, *Naval Research Logistics Quarterly*, **20**, pp. 625–638.
- [178] KARZANOV AV, 1974, *Determining the maximum flow in a network by the method of preflows*, *Soviet Mathematics Doklady*, **15**, pp. 434–437.
- [179] KENNINGTON JL, 1978, *A Survey of Linear Cost Multicommodity Network Flows*, *Operations Research*, **26(2)**, pp. 209–236.
- [180] KENNINGTON JL & SHALABY M, 1977, *An effective subgradient procedure for minimal cost multicommodity flow problems*, *Management Science*, **23**, pp. 994–1004.
- [181] KENNINGTON JL & HELGASON, 1980, *Algorithms for Network Programming*, Wiley-Interscience, New York.
- [182] KCI KONECRANES PLC, 2003, *::: KCI KONECRANES Photo Gallery :::*, [Online], [Cited 26 June 2003], Available from <http://www.atomi.com/kcipictures/>.
- [183] KCI KONECRANES PLC, 2003, *Ship-to-Shore Cranes*, [Online], [Cited 26 June 2003], Available from <http://www.konecranes.com>.
- [184] KELLY JR, 1961, *Critical path planning and scheduling: Mathematical basis*, *Operations Research*, **9**, pp. 296–320.
- [185] KELLY JP, GOLDEN BL & ASSAD AA, 1992, *Cell suppression: Disclosure protection for sensitive tabular data*, *Networks*, **22**, pp. 397–412.
- [186] KLEIN M, 1967, *A primal method for minimal cost flows with application to the assignment and transportation problems*, *Management Science*, **14**, pp. 205–220.

- [187] KOOPMANS TC & REITER S, 1951, *A Model of Transportation, Activity Analysis of Production and Allocation*, Cowles Commission Monograph, **13**, pp. 222–259.
- [188] KORTE B, 1988, *Applications of combinatorial optimization*, (Unpublished) Technical Report 88541-OR, Institute für Okonometrie und Operations Research, Bonn.
- [189] KOTZE T, 2003, *Personal Communication*, Business Development Manager for SAPO, Cape Town.
- [190] KOURTZ P, 1984, *A network approach to least cost daily transfers of forest fire control resources*, INFOR, **22**, pp. 283–290.
- [191] KRIEGLER J, 2003, *Personal Communication*, President of the Hex Table Grapes Association.
- [192] KRIEGLER L, 2003, *Personal Communication*, Head: Topfruit for Colors Fruit (SA) (Pty) Ltd.
- [193] KRUGER W, 2003, *Personal Communication via FE van Dyk*, General Manager of Afrupro Exporters.
- [194] KUEHNE + NAGEL, 2005, *KN Container Guide*, [Online], [Cited 7 February 2005], Available from <http://www.kn-portal.com/ocean/information/?meta=7232&type=News>.
- [195] KUHN HW, 1955, THE HUNGARIAN METHOD FOR THE ASSIGNMENT PROBLEM, *Naval Research Logistics Quarterly*, **2**, pp. 83–97.
- [196] LATHAM AND CHIAZZARI CC, 2003, *Picking – Sorting & Preparation – Export*, [Online], [Cited 19 May 2003], Available from <http://www.lathamcitrus.co.za/pages/process.html>.
- [197] LAURITZEN A/S, 2003, *LauritzenCool AB*, [Online], [Cited 25 June 2003], Available from <http://www.lauritzencool.com>.
- [198] LAW T, 2003, *Personal Communication*, General Manager at LauritzenCool Southern Africa, Contactable at: trevor.law@fc.laucool.com.
- [199] LAWLER EL, 1964, *On scheduling problems with deferral costs*, *Management Science*, **11**, pp. 280–287.
- [200] LEEB A, 2005, *Personal Communication*, Cold Store Manager at Tzaneen Fruit Packers, Contactable at: +27 (0)15 307 1031.
- [201] LEVNER EV & NEMIROVSKY AS, 1991, *A network flow algorithm for just-in-time project scheduling*, (Unpublished) Memorandum COSOR 91-21, Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven.
- [202] LIN TF, 1986, *A system of linear equations related to the transportation problem with application to probability theory*, *Discrete Applied Mathematics*, **14**, pp. 47–56.
- [203] LOUW DB, 2003, *Personal Communication*, owner of Optimal Agricultural Business Systems (OABS) based at the DFPT, Contactable at: daan@dfpt.co.za.
- [204] LOVE RR & VEMUGANTI RR, 1978, *The single-plant mold allocation problem with capacity and changeover restriction*, *Operations Research*, **26**, pp. 159–165.

- [205] LOWE TJ, FRANCIS RL & REINHARDT EW, 1979, *A greedy network flow algorithm for a warehouse leasing problem*, AIIE Transactions, **11**, pp. 170–182.
- [206] MALHOTRA VM, KUMAR MP & MAHESHWARI SN, 1978, *An $O(n^3)$ algorithm for finding maximum flows in networks*, Information Processing Letters, **7**, pp. 277–278.
- [207] MALISHA S, 2003, *Personal Communication via FE van Dyk*, representative from Viamax.
- [208] MAMER JW & SMITH SA, 1982, *Optimizing field repair kits based on job completion rates*, Management Science, **28**, pp. 1328–1334.
- [209] MAPUTO PORT DEVELOPMENT COMPANY, 2004, *Port Maputo – Port Information*, [Online], [Cited 26 March 2004], Available from <http://www.portmaputo.com>.
- [210] Marais C, 2005, *Personal Communication*, Sales representative for SA Truck Bodies – Henred Freuehauf, Contactable at: +27 (0)21 959 1132.
- [211] MARTEL C, 1982, *Preemptive scheduling with release times, deadlines, and due times*, Journal of ACM, **29**, pp. 812–829.
- [212] MASPERO E, 2003, *Personal Communication*, Logistics Analyst at the CSIR – Transportek, Contactable at: EMaspero@csir.co.za.
- [213] MAXWELL WL & WILSON RC, 1981, *Dynamic network flow modelling of fixed path material handling systems*, AIIE Transactions, **13**, pp. 12–21.
- [214] MCKENZIE R, WHITLOW C, CRAIG A & STOFFBERG F, 2003, *Upington*, [Online], [Cited 18 March 2003], Available from <http://www.dwaf.gov.za/orange/upington.htm>.
- [215] MEHROTRA S, 1992, *On the implementation of a primal-dual interior point method*, SIAM Journal on Optimization, **2**, pp. 575–601, [Online], [Cited 18 March 2005], Abstract available from http://locus.siam.org/SIOPT/volume-02/art_0802028.html.
- [216] MEIFERT C, 2003, |eurep|, [Online], [Cited 18 March 2003], Available from <http://www.eurep.org/>.
- [217] MINTY GJ, 1960, *Monotone Networks*, Proceedings of the Royal Society of London, **275A**, pp. 194–212.
- [218] MO J, QI L & WEI Z, 2005, *A manufacturing supply chain optimization model for distilling process*, to appear in Applied Mathematics and Computation.
- [219] MOISEENKO G, 1990, *Vehicle flow optimization in regional transportation networks*, Automation and remote control, **51(9:2)**, pp. 1277–1285.
- [220] MONCUR M, 2005, *The Quotations Page — Your Source for Famous Quotes*, [Online], [Cited 30 May 2005], Available from <http://www.quotationspage.com/>.
- [221] MULVEY JM, 1979, *Strategies in modelling: A personal scheduling example*, Interfaces, **9**, pp. 66–76.
- [222] MURRAY X, 2004, *Personal Communication*, Quality Control Manager for Monteith Trust Farms, Contactable at: xavier@monteith.co.za.
- [223] NAGY N & AKL SG, 2003, *The maximum flow problem: a real-time approach*, Parallel Computing, **29**, pp. 767–794.

- [224] NDELA S, 2003, *Personal Communication via N de Klerk*, Operations Manager at Fresh Produce Terminals (Durban), Contactable at: Neville_Klerk@fpt.co.za.
- [225] NIELSEN T, 2003, *Personal Communication*, Public Relations Officer at Fresh Produce Terminals (Cape Town), Contactable at: Tessa_Nielsen@fpt.co.za.
- [226] NUUS EG, 2002, *Network model for the table grape industry in South Africa*, (Unpublished paper), Stellenbosch.
- [227] OH S & HAGHANI A, 1997, *Testing and evaluation of a multi-commodity multi-modal network flow model for disaster relief management*, *Journal of Advanced Transportation*, **31(3)**, pp. 249–282.
- [228] OPPERMAN R, 2005, *Personal Communication*, Perishable Product Manager for Kuehne + Nagel (Pty) Ltd, Contactable at: rudi.opperman@kuehne-nagel.com.
- [229] OPTIMAL AGRICULTURAL BUSINESS SYSTEMS, 2002, *Key Citrus Industry Statistics 2002*.
- [230] ORLIN D, 1987, *Optimal weapons allocation against layered defences*, *Naval Research Logistics Quarterly*, **34**, pp. 605–617.
- [231] ORLIN JB, 1984, *Genuinely polynomial simplex and non-simplex algorithms for the minimum cost flow problem*, (Unpublished) Technical Report 1615-84, Sloan School of Management, MIT, Cambridge (MA).
- [232] ORLIN JB, 1993, *A faster strongly polynomial minimum cost flow algorithm*, *Operations Research*, **41(2)**, pp. 338–350. Previously appeared in *Proceedings of the 20th ACM Symposium on the Theory of Computing*, pp. 377–387 (1988).
- [233] ORTMANN GF, 2001, *Industrialisation of agriculture and the role of supply chains in promoting competitiveness*, *Agrekon* **40(4)**, pp. 459–489.
- [234] PERISHABLE PRODUCTS EXPORT CONTROL BOARD, 2003, *Carrying Temperature Regimes of Fresh Produce for Sea Export: Official PPECB Instructions — Rev22*, [Online], [Cited 10 April 2003], Available from <http://www.ppecb.co.za/Community/Procedures.asp>.
- [235] PICARD JC & RATLIFF HD, 1973, *Minimal cost cut equivalent networks*, *Management Science*, **19**, pp. 1 087–1 092.
- [236] PICARD JC & RATLIFF HD, 1978, *A cut approach to the rectilinear distance facility location problem*, *Operations Research*, **26**, pp. 422–433.
- [237] POLDERDIJK A, VOGELS J, DEMKES R & VAN DYK FE, 2002, *Chain analysis and user requirements: Results of Workpackage 2 of the Fruitful project*, TNO Inro Department of Logistics and Transport, Delft.
- [238] POWELL W & CARVALHO T, 1997, *Dynamic control of multicommodity fleet management problems*, *European Journal of Operations Research*, **98(3)**, pp. 522–541.
- [239] PRAGER W, 1957, *On warehousing problems*, *Operations Research*, **5**, pp. 504–512.
- [240] RHYS JMW, 1970, *A selection problem of shared fixed costs and network flows*, *Management Science*, **17**, pp. 200–207.

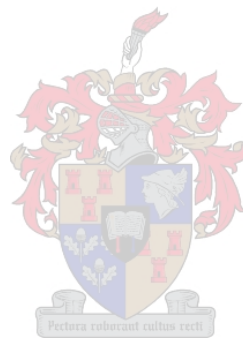
- [241] RIEGER H, 2003, *Polynomial combinatorial methods for analysing the ground states of disordered systems*, Journal of Physics A: Mathematical and General, **36**, pp. 1–15.
- [242] ROBINSON R, 2003, *Personal Communication*, Cold Chain & Special Shipment Specialist at the PPECB, Contactable at: RobbieR@ppecb.com.
- [243] RÖCK H, 1980, *Scaling techniques for minimal cost network flows*, pp. 181–191, in PAPE U (ED.), *Discrete Structures and Algorithms*, Carl Hansen, Munich.
- [244] RS & P DIGITAL PUBLISHING, 2003, *:: CARToday.com/News ::*, [Online], [Cited 14 April 2003], Available from <http://www.cartoday.com/content/news/singlepage.asp?num=2&in=4315>.
- [245] RUSHMEIER R & KONTOGIORGIS S, 1997, *Advances in the optimization of airline fleet assignment*, Transportation science, **31(2)**, pp. 159–169.
- [246] SAKAROVITCH M, 1973, *Two commodity network flows and linear programming*, Mathematical Programming, **4**, pp. 1–20.
- [247] SAPOUNTZIS C, 1984, *Allocating blood to hospitals from a central blood bank*, European Journal of Operations Research, **16**, pp. 157–162.
- [248] SCHENKER INC., 2005, *Ocean Container Dimensions*, [Online], [Cited 17 February 2005], Available from http://www.schenkerusa.com/Container/index.asp?file=Container\htm\Ocean_cont_specs.htm&br=0.
- [249] SCHRIJVER A, 2002, *On the history of the transportation and maximum flow problems*, Mathematical Programming, **91**, pp. 437–445.
- [250] SCHWARTZ BL, 1966, *Possible winners in partially completed tournaments*, SIAM Review, **8**, pp. 302–308.
- [251] SECURICOR OMEGA CONTAINER LOGISTICS, 2000, *Top marks for tanks*, [Online], [Cited 8 May 2003], Available from http://www.securicor-omega.com/containerlogistics/media_desk/june14_00.html.
- [252] SEGAL M, 1974, *The operator-scheduling problem: A network flow approach*, Operations Research, **22**, pp. 808–823.
- [253] SERVI LD, 1989, *A network flow approach to a satellite scheduling problem*, Research Report, GTE Laboratories, Waltham (MA).
- [254] SHAI O, 2005, *Multidimensional max-flow method and its application for plastic analysis*, to appear in Advances in Engineering Software.
- [255] SHILOACH Y & VISHKIN U, 1982, *An $O(n^2 \log n)$ parallel max-flow algorithm*, Journal of Algorithms, **3**, pp. 128–146.
- [256] SHIOURA A, 2004, *The MA-ordering max-flow algorithm is not strongly polynomial for directed networks*, Operations Research Letters, **32**, pp. 31–35.
- [257] SHIPPING CONTAINER HOUSING GUIDE, 2005, *Shipping container standard dimensions*, [Online], [Cited 17 February 2005], Available from <http://www.shipping-container-housing.com/shipping-container-standard-dimensions.html>.

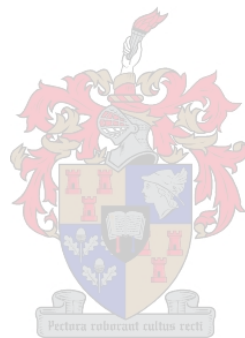
- [258] SHUM D & LEUNG T, 2003, *Ship Photos – Container Ships From major Companies*, [Online], [Cited 7 May 2003], Available from <http://members.tripod.com/shumsw/>.
- [259] SLEATOR DD & TARJAN RE, 1983, *A data structure for dynamic trees*, Journal of Computer and Systems Sciences, **24**, pp. 362–391.
- [260] SLUMP CH & GERBRANDS JJ, 1982, *A network flow approach to reconstruction of the left ventricle from two projections*, Computer Graphics and Image Processing, **18**, pp. 18–36.
- [261] SMITH JC, 2004, *Algorithms for distributing telecommunication traffic on a multiple-ring SONET-based network*, European Journal of Operations Research, **154**, pp. 659–672.
- [262] SMITH (JR) LW, 1956, *Current State of Industrial Use of Linear Programming*, Management Science, **2**, pp. 156–158.
- [263] STATISTICS SOUTH AFRICA, 2004, *Gross Domestic Product: Fourth quarter 2003*, [Online], [Cited 21 June 2004], Available from <http://www.statssa.gov.za>.
- [264] SOKKALINGAM PT, AHUJA RK & ORLIN JB, 2000, *New Polynomial-Time Cycle-Cancelling Algorithms for Minimum Cost Flows*, Networks, **36(1)**, pp. 53–63.
- [265] SOUTH AFRICAN AVOCADO GROWERS' ASSOCIATION, 2003, *Personal Communication via FE van Dyk*, Contactable at: saaga@saaga.co.za.
- [266] SOUTH AFRICAN AVOCADO GROWERS ASSOCIATION, 2003, *Welcome to the SAAGA website*, [Online], [Cited 27 March 2003], Available from <http://www.avocado.co.za/>.
- [267] SOUTH AFRICAN MANGO GROWERS' ASSOCIATION, 2003, *Growing, picking and packing Mangoes*, [Online], [Cited 27 March 2003], Available from <http://www.mango.co.za/moreabout/handling.html>.
- [268] SOUTH AFRICAN MANGO GROWERS' ASSOCIATION, 2003, *More About Mangoes – Common South African Cultivars*, [Online], [Cited 27 March 2003], Available from <http://www.mango.co.za/moreabout/cultivars.html>.
- [269] SOUTH AFRICAN PORT OPERATIONS, 2005, [*South African Port Operations*] — *The Total Maritime Solution*, [Online], [Cited 29 April 2005], Available from <http://www.saponet.co.za/>.
- [270] SOUTHERN AFRICAN FRUIT TERMINALS (PTY) LTD, 2003, *SAFT Durban 2003 Season Tariffs*, Available by request.
- [271] SOUTHERN AFRICAN FRUIT TERMINALS (PTY) LTD, 2003, *SAFT Group of Companies*, Brochure available by request.
- [272] SOUTHERN AFRICAN FRUIT TERMINALS (PTY) LTD, 2003, *SAFT Western Cape 2003 Season Tariffs*, Available by request.
- [273] SRINIVASAN V, 1974, *A transshipment model for cash management decisions*, Management Science, **20**, pp. 1 350–1 363.
- [274] SRINIVASAN V, 1979, *Network models for estimating brand-specific effects in multiattribute marketing models*, Management Science, **25**, pp. 11–21.

- [275] STONE HS, 1977, *Multiprocessor scheduling with the aid of network flow algorithms*, IEEE Transactions on Software Engineering, **3**, pp. 85–93.
- [276] SWOVELAND C, 1971, *Decomposition algorithms for the multicommodity distribution problem*, Working paper 184, Western Management Science Institute, University of California, Los Angeles.
- [277] TARDOS É, 1985, *A strongly polynomial minimum cost circulation algorithm*, Combinatorica, **5**, pp. 247–255.
- [278] TARDOS É, 1986, *A strongly polynomial algorithm to solve combinatorial linear programs*, Operations Research, **34**, pp. 250–256.
- [279] TARJAN RE, 1991, *Efficiency of the primal network simplex algorithm for the minimum-cost circulation problem*, Mathematics of Operations Research, **16**, pp. 272–291.
- [280] TELKOM DIRECTORY SERVICE, 2003, *Yellow Pages Internet*, [Online], [Cited 24 March 2003], Available from <http://www.yellowpages.co.za/>.
- [281] TOLSTOÏ AN, 1930, *Metody nakhozhdeniya naimenshego summovogo kilometrazha pri planirovaniiperevozok v prostranstve [English: Methods of finding the minimal total kilometrage in cargo-transportation planning in space]*, in *Planirovanie Perevozok, Sbornik pervyĭ [English: Transportation Planning, Volume I]*, Transpechat NKPS [TransPress of the National Commissariat of Transportation], Moscow, pp. 23–55.
- [282] TOLSTOÏ AN, 1939, *Methods of Removing Irrational Shipments in Planning*, Sotsialisticheskii Transport, **9**, pp. 28–51.
- [283] TOMLIN JA, 1966, *A linear programming model for the assignment of traffic*, Proceedings of the 3rd Conference of the Australian Road Research Board, **3**, pp. 263–271.
- [284] TRANS INFO MANAGEMENT CO., LTD, 2005, *Bangkok Shipowners and Agents Association*, [Online], [Cited 17 February 2005], Available from http://www.thaibsaa.com/knowledge_base.asp.
- [285] TSO M, 1986, *Network flow models in image processing*, Journal of the Operational Research Society, **37**, pp. 31–34.
- [286] TSO M, KLEINSCHMIDT P, MITTERREITER I & GRAHAM J, 1991, *An efficient transportation algorithm for automatic chromosome karyotyping*, Pattern Recognition Letters **12**, pp. 117–126, [Online], [Cited 27 March 2005], Abstract available from <http://www.sciencedirect.com>.
- [287] TSUJINO M, FUKUSHIMA M & IBARAKI T, 1994, *A model of time-varying flows on a congested multidestination network*, Asia-pacific Journal of Operational Research, **11(2)**, pp. 141–153.
- [288] UNWEMBI COMMUNICATIONS, 2004, *Profile: S’bu Ndebele*, [Online], [Cited 18 March 2005], Available from http://www.anc.org.za/people/s_ndebele.html.
- [289] VAN DER MERWE C, 2003, *Personal Communication*, Cape Town Shipping Terminal Manager for SAFT, Contactable at: cornel@saft.co.za.
- [290] VAN DYK FE, 2002, *Proposal For: Fruit Logistics Infrastructure Capacity Optimisation Study*.

- [291] VAN DYK FE, 2003 – 2005, *Personal Communication*, Project Manager at CSIR – Transportek, Contactable at: fevandyk@csir.co.za.
- [292] VAN DYK FE, HAVENGA J, ORTMANN FG, HOBBS I, LOUW DB & LOUW NH, 2004, *Fruit Logistics Scenario Model*, Port capacity and fruit export model, [Online], [Cited 18 March 2005], Available from <http://www.dfpt.co.za/fruitlog/index.htm>.
- [293] VAN DYK FE & MASPERO E, 2004, *An analysis of the South African fruit logistics infrastructure*, ORION, **20**, pp. 55–72.
- [294] VAN HEERDEN G, 2005, *Personal Communication*, General Manager of the shipping company Cape Reefers, Contactable at: gerhard_van_heerden@capereefers.co.za.
- [295] VAN SLYKE R & FRANK H, 1972, *Network reliability analysis: Part 1*, Networks, **1**, pp. 279–290.
- [296] VEINOTT AF & WAGNER HM, 1962, *Optimal capacity scheduling: Parts I and II*, Operations Research, **10**, pp. 518–547.
- [297] VERMAAK I, 2003, *Personal Communication via FE van Dyk*, Information Manager at Citrus South Africa, Contactable at: data@citrussa.co.za.
- [298] WALLACHER C & ZIMMERMANN UT, 1991, *A combinatorial interior point method for network flow problems*, presented at the 14th International Symposium on Mathematical Programming, Amsterdam.
- [299] WAGNER DK, 1990, *Disjoint (s, t) -cuts in a network*, Networks, **20**, pp. 361–371.
- [300] WAYNE KD, 1999, *Generalized Maximum Flow Algorithms*, PhD Dissertation, School of Operations Research and Industrial Engineering, Cornell University.
- [301] WAYNE KD, 2001, *A new property and a faster algorithm for baseball elimination*, SIAM J. Discrete Mathematics, **14**(2), pp. 223–229.
- [302] WEI L, YANGZHOU & JIANGSU, 2005, *Yangzhou Tonlee Reefer Container Co., Ltd.*, [Online], [Cited 7 February 2005], Available from <http://www.js.cei.gov.cn/jsmeio98/1000008/eyztllcq.htm>.
- [303] WEINTRAUB A, 1974, *A primal algorithm to solve network flow problems with convex costs*, Management Science, **15**, pp. 87–97.
- [304] WESTCOTT D, 2003, *Personal Communication via FE van Dyk*, Manager at Tabanelli Export.
- [305] WHITE WW, 1972, *Dynamic transshipment networks: An algorithm and its applications to the distribution of empty containers*, Networks, **2**, pp. 211–230.
- [306] WIKIPEDIA, 2003, *Pallet Sizes*, [Online], [Cited 7 February 2005], Available from <http://www.americanmusicscene.com/site/27114-pallet-size.html>.
- [307] WOODARD M, 2003, *Mathematical Quotation Server*, [Online], [Cited 16 March 2003], Available from <http://math.furman.edu/~mwoodard/mquot.html>.
- [308] ZANGWILL WI, 1969, *A backloging model and a multi-echelon model of dynamic economic size production system: A network approach*, Management Science, **15**, pp. 506–527.

- [309] ZAWACK DJ & THOMPSON GL, 1987, *A dynamic space-time network flow model for city traffic congestion*, *Transportation Science*, **21**, pp. 153–162.





Appendix A

Maps of Deciduous Fruit Production regions

This appendix contains maps drawn with the aid of GPS data, made available by the CSIR. Section 2.1.2 contains less accurate maps drawn without the use of GPS, as only the deciduous producers had GPS data available.

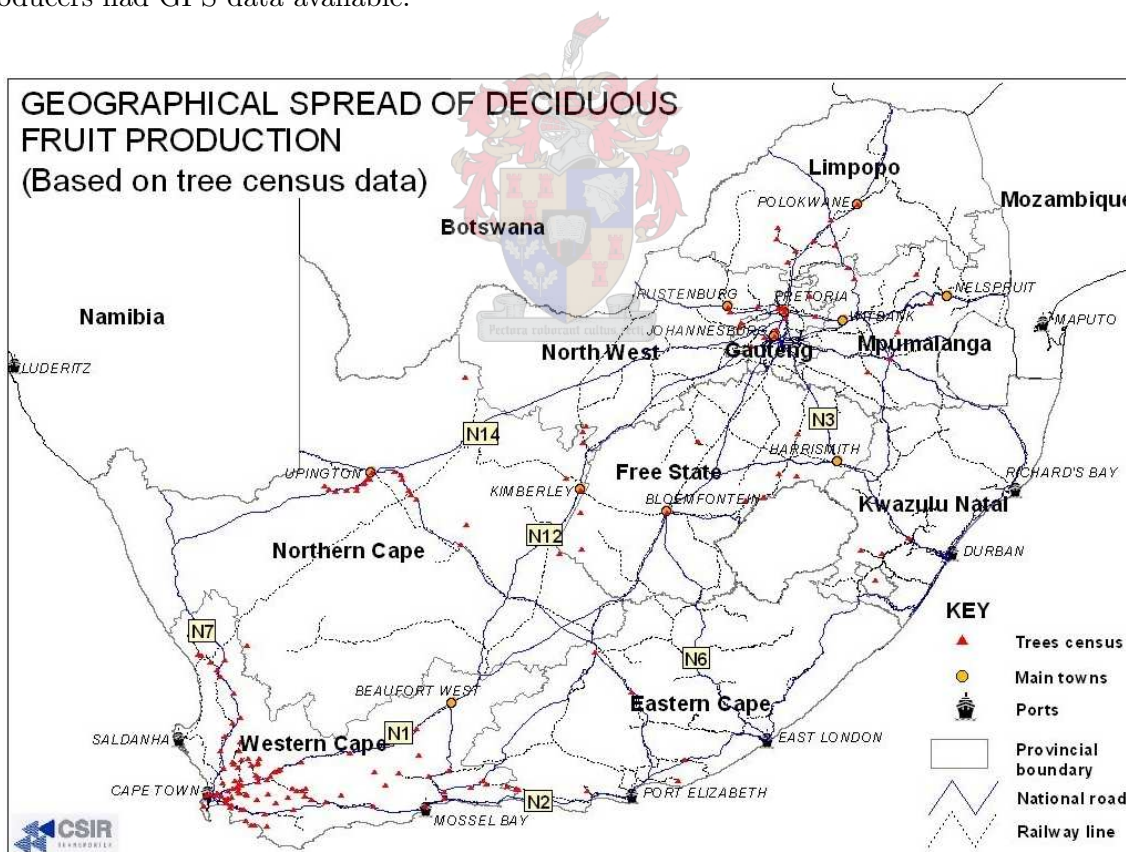


Figure A.1: A map with points representing locations where deciduous fruit is grown.

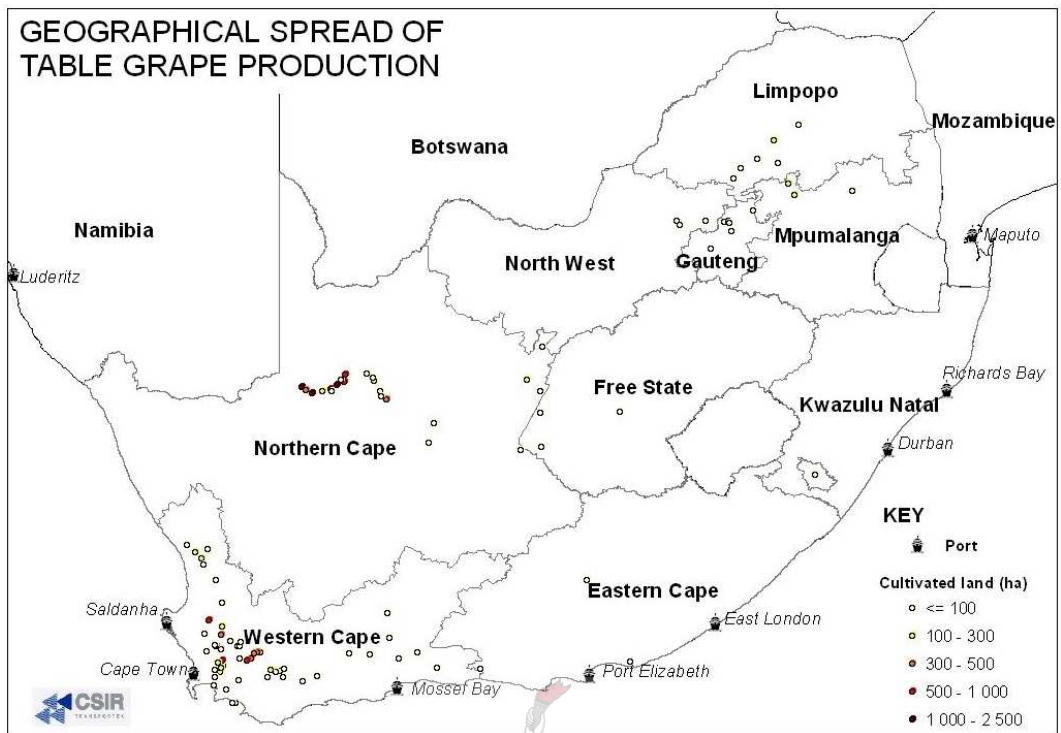


Figure A.2: A map with points representing locations where grapes are grown.

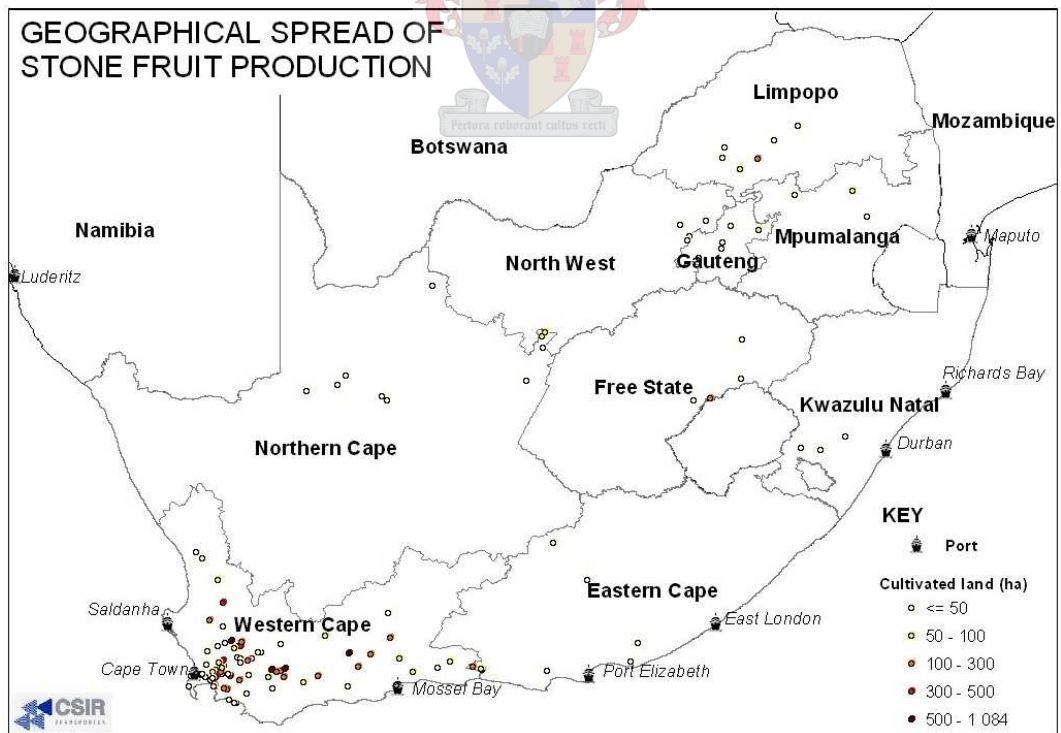


Figure A.3: A map with points representing locations where stone fruit is grown.

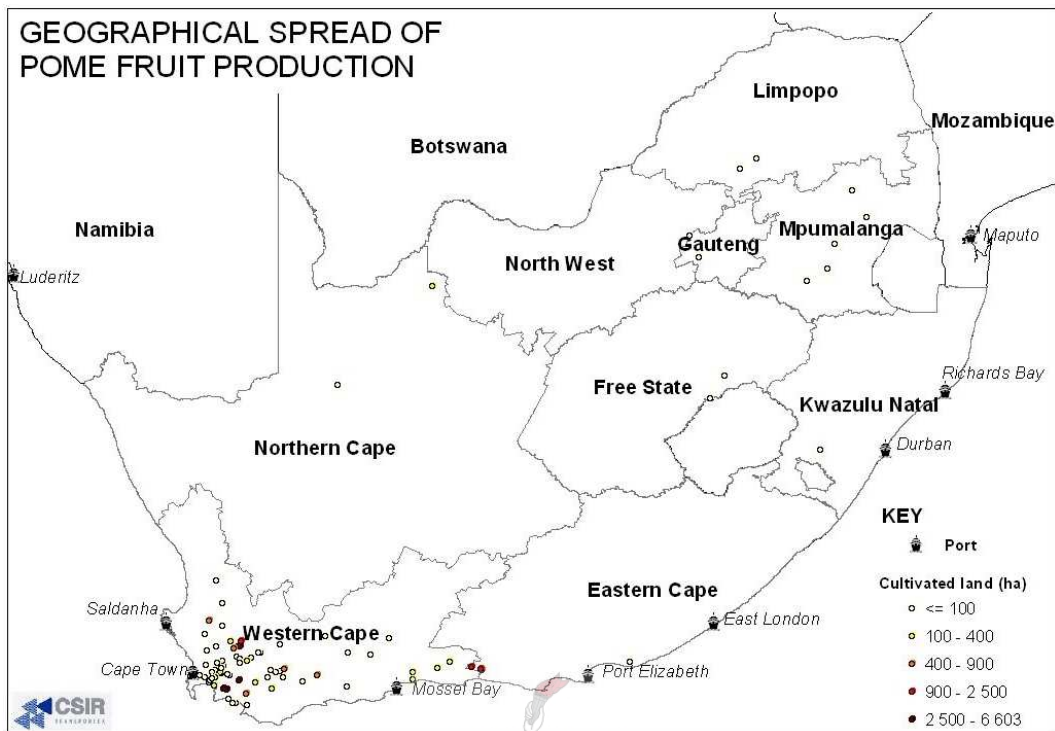


Figure A.4: A map with points representing locations where pome fruit is grown.

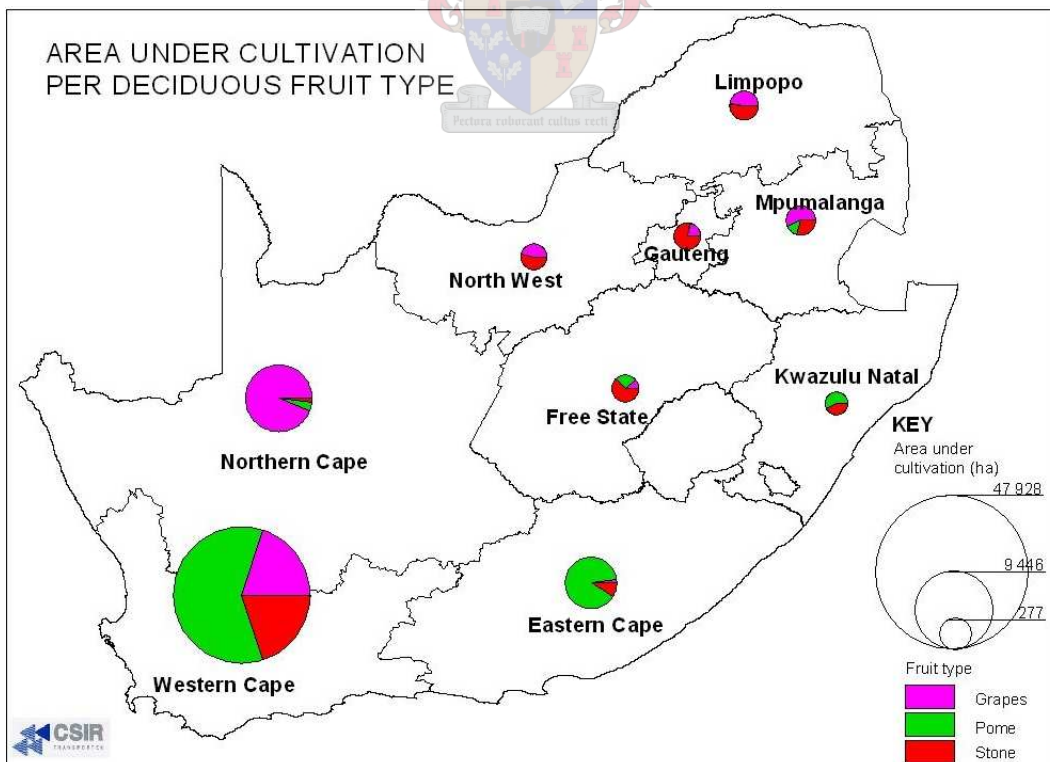
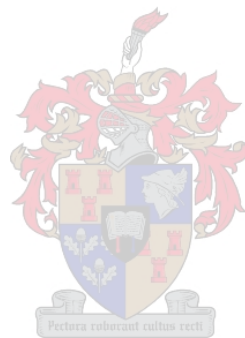


Figure A.5: A map with pie charts indicating approximate areas under cultivation of deciduous fruit in each province during 2003.

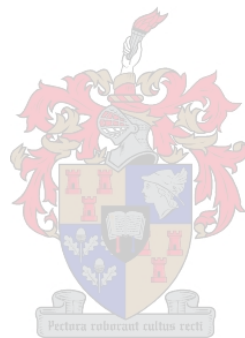


Appendix B

Packhouse and Cold Store Regions

This appendix contains a list of virtual packhouse and cold store virtual groupings.

Name	Abbr.	Grouping
Levubu	Le	Levubu, Messina, Tshipise, Louis Trichardt, Weipe
Tzaneen	Tz	Tzaneen, Mooketsi, Duiwelskloof, Politsi, Letaba, Letsitele
Hoedspruit	Ho	Hoedspruit, Burgersfort
Kiepersol	Ki	Kiepersol, Hazyview
Nelspruit	Ne	Nelspruit, White River, Schagen, Schoemanskloof, Barberton, Brondal
Malelane	Ma	Malelane, Hectorspruit, Komatipoort
Potgietersrus	Po	Potgietersrus, Constantia, Naboomspruit, Vaalwater
Ellisras	El	Ellisras, Stokpoort
Marble Hall	Mh	Marble Hall, Groblersdal
Ermelo	Er	Ermelo
Pongola	Pg	Pongola, Paulpietersburg
Nkwalini	Nk	Nkwalini, Melmoth, Eshowe, Stanger
KwaZulu–Natal	Kz	Weenen, Greytown, Thornville, Richmond, Wartburg
Eastern Cape Midlands	Ea	Fort Beaufort, Grahamstown
Sundays River Valley	Su	Kirkwood, Addo
Patensie	Pa	Patensie, Hankey
Langkloof	La	Avontuur, Misgund, Louterwater, Krakeelrivier, Joubertina, Knoll, George
Karoo	Ka	Prince Albert
Klein Karoo	Kl	Robertson, Ashton, Montagu, Tradouw, Buffeljagsrivier
GEVV	Ge	Grabouw, Elgin, Villiersdorp, Vyeboom, Botrivier
Boland	Bo	Somerset West, Stellenbosch, Simondium, Paarl, Wellington, Windmeul, Paardeberg, Groendal, Groot Drakenstein
Ceres	Ce	Ceres, Wolsley, Riebeek–Kasteel, Gouda, Tulbagh, Saron
De Doorns	De	De Doorns
Piketberg	Pi	Piketberg, Porterville
Citrusdal	Ci	Citrusdal, Clanwilliam, Trawal, Vredendal
Orange River	Or	Upington, Pella, Onseepkans, Blouputs, Augrabies, Kakamas, Friersdale, Kanoneiland, Skuitdrift, Groblershoop
Hartswater	Ha	Hartswater
Johannesburg	Jo	Isando, Kempton Park, Brits, Westonaria, Bronkhorstspuit
Rustenburg	Ru	Rustenburg
Free State	Fs	Bethlehem, Harrismith
Middelburg	Mi	Middelburg



Appendix C

Origin–Destination Flow Volumes

To determine the expected flow of fruit to each of the ports from the various production regions, the percentage of fruit sent to each port must be known. The origin–destination fruit flow calculated by FE van Dyk [292], after consultation with fruit industry role-players, is presented below.

Region	Total	To Cape Town		
		Road	Rail	% Total
Northern Cape	60 758	60 758		100
Boland	131 978	131 978		100
West Coast	6 726	6 726		100
Senwes/North West	1	1		100
Limpopo	1	1		100
Namibia	8 014	8 014		100

Table C.1: *The volume (in tons) of table grapes sent from the various production regions to each of the ports. Note that these volumes may differ from those in Table 3.3 due to conversion errors from pallets to tons.*

The only port used by table grape producers and exporters is Cape Town. This is due to the proximity of Cape Town to the grape production regions. Cape Town is also the closest port to the largest markets in Europe, thus reducing the shipping time and increasing the likely shelf-life of the fruit. A harbour such as the one at Lüderitz may become an important port for table grape export in future due to its proximity to large growing regions in Namibia and South Africa along the Orange River. All transportation of table grapes is achieved by road.

It is clear from Table C.2 that citrus is the fruit sent to most ports, and is the fruit with the largest percentage of the total transported by rail (19.3% of all citrus fruit is transported to the ports by rail). Durban receives most of the citrus — with an estimated total of 56.8% of all citrus fruit; followed by Cape Town with 19.2%, Port Elizabeth with 17.0% and Maputo with 7.0% of all citrus. These figures can be misleading, as Durban may receive most of the citrus traffic during the peak weeks around weeks 27 to 34 when the Valencias are harvested, while Cape Town typically receives much more soft citrus (easy-peeler) traffic than the other ports during weeks 19 to 26. Unfortunately, no data were known by which to differentiate between hard and soft citrus fruits when calculating the origin–destination volumes; hence the total was used to calculate the destination ports for both citrus types.

TOTAL CITRUS							
Region	Total	To Cape Town			To Durban		
		Road	Rail	% Total	Road	Rail	% Total
Northern Cape	6 911	3 927		56.8	2 840		41.1
Boland	26 656	24 550		92.1	1 541		5.8
West Coast	106 622	98 199		92.1	6 165		5.8
Langkloof	129						
Eastern Cape	210 301	28 558		13.6	10 828		5.1
Nkwalini	62 444	698		1.1	60 885		97.5
Swaziland	24 954	127		0.5	24 798		99.4
Mpumalanga-Lowveld	125 768	5 765		4.6	68 939	12 046	64.4
Senwes/North West	132 135	9 494		7.2	118 582		89.7
Limpopo	333 683	25 468	1 310	8.0	197 862	80 060	83.3
Zimbabwe	803				568		70.7

Region	Total	To Port Elizabeth			To Maputo		
		Road	Rail	% Total	Road	Rail	% Total
Northern Cape	6 911	144		2.1			
Boland	26 656	565		2.1			
West Coast	106 622	2 258		2.1			
Langkloof	129	129		100.0			
Eastern Cape	210 301	99 418	71 497	81.3			
Nkwalini	62 444	176		0.3	685		1.1
Swaziland	24 954	29		0.1			
Mpumalanga-Lowveld	125 768	22			25 161	13 835	31.0
Senwes/North West	132 135				4 059		3.1
Limpopo	333 683	1 207		0.4	7 844	19 932	8.3
Zimbabwe	803				235		29.3

Table C.2: The volume (in tons) of all citrus fruit sent from the various production regions to each of the ports. Note that these volumes may differ from the sums from those in Tables 3.4 and 3.5 due to conversion errors from pallets to tons.

TOTAL POME							
Region	Total	To Cape Town			To Port Elizabeth		
		Road	Rail	% Total	Road	Rail	% Total
Boland	319 784	319 784		100.0			
Langkloof	53 083				53 083		100.0
Free State	1 161	1 161		100.0			

Table C.3: The volume (in tons) of pome fruit sent from the various production regions to each of the ports. Note that these volumes may differ from the those in Table 3.6 due to conversion errors from pallets to tons.

All of the pome fruit exported from the Boland/Klein Karoo and Free State regions is sent to the port of Cape Town, while the producers and exporters operating in the Langkloof region prefer to send their fruit to Port Elizabeth harbour, as it is the closest port to them. The Boland/Klein Karoo region produces 85.5% of all export pome fruit, while the Langkloof and Free State regions export 14.2% and 0.3% of the total pome fruit respectively. All transportation of pome fruit is by road.

All stone fruit is exported via Cape Town harbour. As with pome fruit, the Boland/Klein Karoo region exports the largest volume, followed by the Langkloof, Limpopo, Northern Cape, Senwes/North West, Free State and Mpumalanga-Lowveld regions. All transportation of stone fruit is by road.

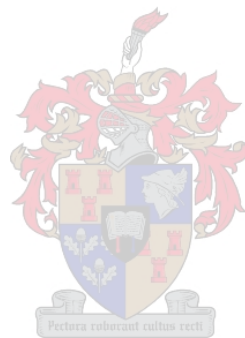
TOTAL STONE				
Region	Total	To Cape Town		
		Road	Rail	% Total
Northern Cape	713	713		100.0
Boland	54 426	54 426		100.0
Langkloof	2 886	2 886		100.0
Free State	174	174		100.0
Mpumalanga-Lowveld	164	164		100.0
Senwes/North West	320	320		100.0
Limpopo	1 081	1 081		100.0

Table C.4: *The volume (in tons) of stone fruit sent from the various production regions to each of the ports. Note that these volumes may differ from the those in Table 3.7 due to conversion errors from pallets to tons.*

TOTAL SUBTROPICAL							
Region	Total	To Cape Town			To Durban		
		Road	Rail	% Total	Road	Rail	% Total
KZN Midlands	1 572	1 179		75.0	393		25.0
Mpumalanga-Lowveld	19 164	18 238		95.2	130		0.7
Senwes/North West	529	529		100.0			
Limpopo	27 268	22 473	3 466	95.1	270		1.0
Region	Total	To Port Elizabeth			To Johannesburg		
		Road	Rail	% Total	Road	Rail	% Total
KZN Midlands	1 572						
Mpumalanga-Lowveld	19 164	272		1.4	524		2.7
Senwes/North West	529	529					
Limpopo	27 268	535		2.0	524		1.9

Table C.5: *The volume (in tons) of subtropical fruit sent from the various production regions to each of the ports. Note that these volumes may differ from the those in Table 3.8 due to conversion errors from pallets to tons.*

Four ports are used for subtropical fruit export, with most fruit exported through Cape Town harbour. Three quarters of the subtropical fruit from KZN Midlands is sent to Cape Town, with the remaining fruit exported via Durban. The producers and exporters from the Mpumalanga-Lowveld region send an estimated 95.2% of their subtropical fruit to Cape Town, 0.7% to Durban and 2.7% to Johannesburg International Airport. All of the fruit from the Senwes/North West region is sent to Cape Town, while the Limpopo exporters and producers send 95.1% to Cape Town, 1.0% to Durban and 1.9% to Johannesburg International Airport. The Limpopo region is the only region in which a train is used to transport integral containers of subtropical fruit to Cape Town. This accounts for 12.7% of all subtropical fruit from the region. All other subtropical fruit transport from that region, as well as from the other regions, is by road.



Appendix D

Model Abbreviations

This appendix contains abbreviations used to label infrastructure components in the graph theoretic and mathematical programming models of Chapters 4 and 5. Tzaneen is used as an example of a region, but may be substituted by any of the regional groupings listed in Appendix B.

G	Grapes
S	Stone
P	Pome
C	Citrus
E	Easy-peeler
A	Avocado
M	Mango
L	Litchi
LC	Letsitele Citrus production region
TM	Tzaneen Mango production region
TA	Tzaneen Avocado production region
TL	Tzaneen Litchi production region
TzCPh	Tzaneen Citrus Packhouse
TzCMPH	Tzaneen Citrus and Mango Packhouse
TzMPH	Tzaneen Mango Packhouse
TzAMPh	Tzaneen Avocado and Mango Packhouse
TzAMLPh	Tzaneen Avocado, Mango and Litchi Packhouse
TzAPh	Tzaneen Avocado Packhouse
TzALPh	Tzaneen Avocado and Litchi Packhouse
TzCCs	Tzaneen Citrus Cold Store
TzCMCs	Tzaneen Citrus and Mango Cold Store
TzMCs	Tzaneen Mango Cold Store
TzAMCs	Tzaneen Avocado and Mango Cold Store
TzACs	Tzaneen Avocado Cold Store
TzAMLCs	Tzaneen Avocado, Mango and Litchi Cold Store
TzALCs	Tzaneen Avocado and Litchi Cold Store
TzUnTk	Tzaneen Unrefrigerated (bulk) Truck transportation
TzUnTn	Tzaneen Unrefrigerated (bulk) Train transportation
TzReTk	Tzaneen Refrigerated (bulk) Truck transportation
TzInTk	Tzaneen Integral Container on a Truck transportation
TzInTn	Tzaneen Integral Container on a Train transportation
TzPo	Tzaneen Packed Porthole Container transportation



MCs	Maputo C old S tore
MIn	Maputo Cold Store Packed I ntegral Container
MfptCs	Maputo P roduce T erminal C old S tore (FPT)
MfptIn	Maputo F resh P roduce T erminal Packed I ntegral
MfptB	Maputo P roduce T erminal – B ulk Loading
MfptUn	Maputo P roduce T erminal – U nrefrigerated Bulk Loading
MfptC	Maputo P roduce T erminal – C ontainer Loading
BS	B lue S tore
CTS	Cape T own Container Terminal Reefer S tack
CTCs	Cape T own Harbour C old S tore
CTfptCs	Cape T own F resh P roduce Terminal C old S tore
CTIn	Cape T own Cold Store Packed I ntegral Container
CTPo	Cape T own Cold Store Packed P orthole Container
CTfptIn	Cape T own F resh P roduce T erminal Packed I ntegral Container
CTcomS	Cape T own C ombi S tack
CTCT	Cape T own Container T erminal
CTsaftB	Cape T own's S outh A frican F ruit T erminal – B ulk Loading
CTsaftUn	Cape T own's S outh A frican F ruit T erminal – U nrefrigerated Bulk Loading
CTsaftC	Cape T own's S outh A frican F ruit T erminal – C ontainer Loading
CTfptB	Cape T own's F resh P roduce Terminal – B ulk Loading
CTfptUn	Cape T own's F resh P roduce T erminal – U nrefrigerated Bulk Loading
CTfptC	Cape T own's F resh P roduce T erminal – C ontainer Loading
DS	D urban Container Terminal Reefer S tack
DCs	D urban Harbour C old S tore
DfptCs	D urban F resh P roduce Terminal C old S tore
DIn	D urban Cold Store Packed I ntegral Container
DfptIn	D urban F resh P roduce Terminal Packed I ntegral Container
DCT	D urban Container T erminal
DsaftB	D urban's S outh A frican F ruit T erminal – B ulk Loading
DsaftUn	D urban's S outh A frican F ruit T erminal – U nrefrigerated Bulk Loading
DsaftC	D urban's S outh A frican F ruit T erminal – C ontainer Loading
DfptB	D urban's F resh P roduce Terminal – B ulk Loading
DfptUn	D urban's F resh P roduce Terminal – U nrefrigerated Bulk Loading
DfptC	D urban's F resh P roduce Terminal – C ontainer Loading
PES	P ort E lizabeth Container Terminal Reefer S tack
PECs	P ort E lizabeth Harbour C old S tore
PEfptCs	P ort E lizabeth F resh P roduce Terminal C old S tore
PEIn	P ort E lizabeth Cold Store Packed I ntegral Container
PEfptIn	P ort E lizabeth F resh P roduce Terminal Packed I ntegral Container
PECT	P ort E lizabeth Container T erminal
PEfptB	P ort E lizabeth's F resh P roduce Terminal – B ulk Loading
PEfptUn	P ort E lizabeth's F resh P roduce Terminal – U nrefrigerated Bulk Loading
PEfptC	P ort E lizabeth's F resh P roduce Terminal – C ontainer Loading
JIA	J ohannesburg International Airport
JCs	J ohannesburg International Airport C old S tore
UIA	U pington International Airport
UCs	U pington International Airport C old S tore
CTIA	C ape T own International Airport

Appendix E

Worked Examples of Algorithmic Implementations

In this appendix worked examples of *Ford's algorithm*, the *Shortest augmenting path algorithm* and the *Successive shortest path algorithm* are presented for further clarification.

E.1 Ford's Algorithm

Consider the network in Figure E.1 consisting of nodes and arcs with both positive and negative weights, and suppose a shortest path from node s to all other nodes is sought. The distance label on node s is initially zero and the distance labels on all other nodes are initially infinite. Only node s has been labelled permanently, so $L(s) = 1$ and L , the number of times a node has been labelled permanently, is zero for all other nodes. The distance labels of the nodes adjacent from s are adjusted so that the distance label of node 1 is temporarily labelled $d(1) = 0$, $p(1) = s$ and the distance label of node 2 becomes $d(2) = 0$, $p(2) = s$. The current situation is depicted in Figure E.1.

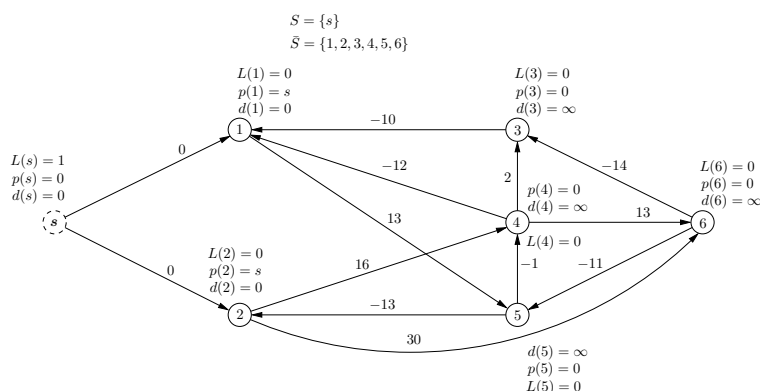


Figure E.1: *Ford's Algorithm example: Original graph.*

As both vertices 1 and 2 have the same distance label, one may arbitrarily choose node 1 to be labelled permanently ($L(1) = 1$). Now that node 1 is labelled permanently, the nodes adjacent from it may be labelled temporarily; thus node 5 receives the temporary label $d(5) = 13$, $p(5) = 1$. These changes to the labels are shown in Figure E.2.

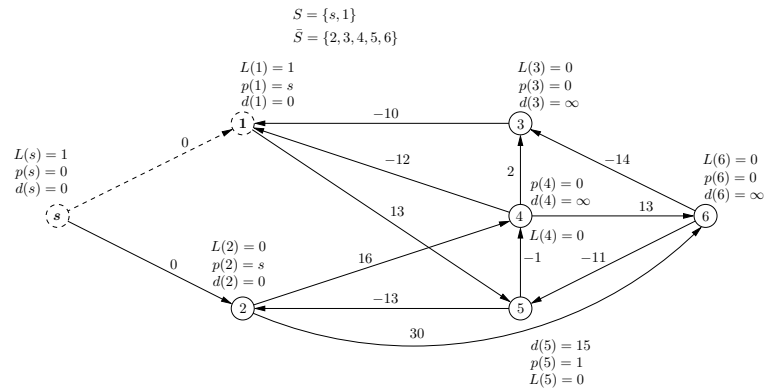


Figure E.2: Ford's Algorithm example: First iteration.

Node 2 now has the lowest distance label of the temporarily labelled nodes; thus it is permanently labelled ($L(2) = 1$). Node 2 is adjacent to nodes 4 and 6, thus they receive the temporary labels $d(4) = 16$, $p(4) = 2$ and $d(6) = 30$, $p(6) = 2$ respectively. These changes to the nodes are shown in Figure E.3.

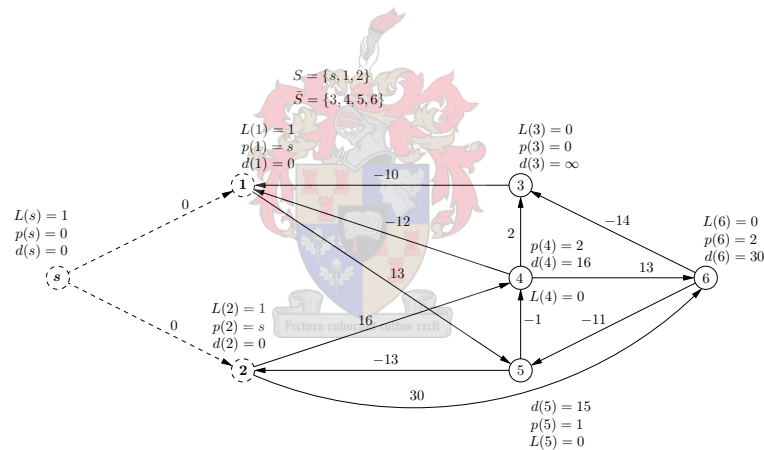


Figure E.3: Ford's Algorithm example: Second iteration.

Now node 5 is the lowest temporarily labelled node with a distance of 13 from s ; thus it is permanently labelled ($L(5) = 1$). Node 5 is adjacent to nodes 2 and node 4. As the distance label to node 5 is $d(5) = 13$ and the weight of the arc from node 5 to node 2 is -13 , the resulting distance from node s to node 2 via node 5 is 0. This does not reduce the distance label of node 2; thus the labels of node 2 do not change. The current distance label of node 4 is $d(4) = 16$, but since the weight of the arc $(5, 4)$ is $c_{5,4} = -1$, the labels of node 4 change to $d(4) = 12$, $p(4) = 5$ ($p(4)$ was previously 2). The current situation is depicted in Figure E.4.

Thus, as node 4 is the temporary node with the smallest distance label, it is permanently labelled ($L(4) = 1$). Node 4 ($d(4) = 12$) is adjacent to node 1, node 3 and node 6. The current distance labels of these nodes are $d(1) = 0$, $d(3) = \infty$ and $d(6) = 30$ respectively. As $c_{4,1} = -12$ the distance label of node 1 does not improve; thus its labels remain the same. The weight of the arc $(4, 3)$ is $c_{4,3} = 2$, thus the distance label of node 3 improves temporarily to $d(3) = 14$, $p(3) = 4$. The weight of the arc from node 4 to node 6 is $c_{4,6} = 13$, thus the distance label of node 6 improves to $d(6) = 25$, $p(6) = 4$ ($p(6)$ was previously 2), as shown in Figure E.5.

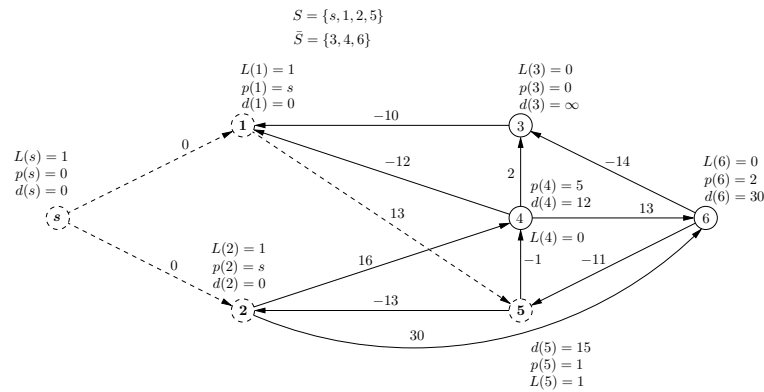


Figure E.4: Ford's Algorithm example: Third iteration.

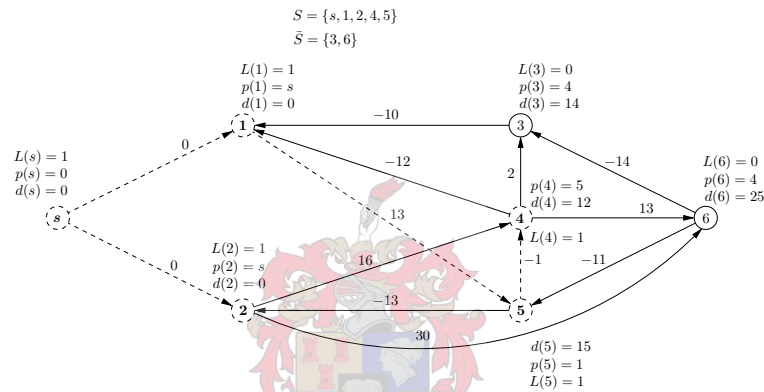


Figure E.5: Ford's Algorithm example: Fourth iteration.

As node 3 has the lowest distance label of the temporarily labelled nodes, it is permanently labelled ($L(3) = 1$). Node 3 ($d(3) = 14$) is only adjacent to node 1. Since the weight of the arc $(3, 1)$ is $c_{3,1} = -10$, the distance label of node 1 cannot improve, so no distance labels change (see Figure E.6).

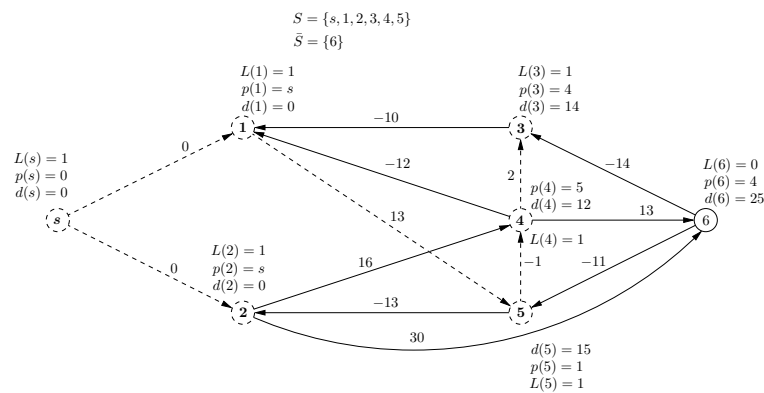


Figure E.6: Ford's Algorithm example: Fifth iteration.

The last temporarily labelled node is node 6, so it is permanently labelled $d(6) = 25, p(6) = 4$ ($L(6) = 1$). Node 6 ($d(6) = 25$) is adjacent to node 3 and to node 5. As the weight of the arc

$(6, 5)$ is $c_{6,5} = -11$ and $d(5) = 13$, the distance label of node 5 remains unchanged. However, as the weight of the arc $(6, 3)$ is $c_{6,3} = -14$ and the distance label of node 3 is $d(3) = 13$, node 3 is removed from the set of permanently labelled nodes and is temporarily labelled with $d(3) = 11$ and $p(3) = 6$.

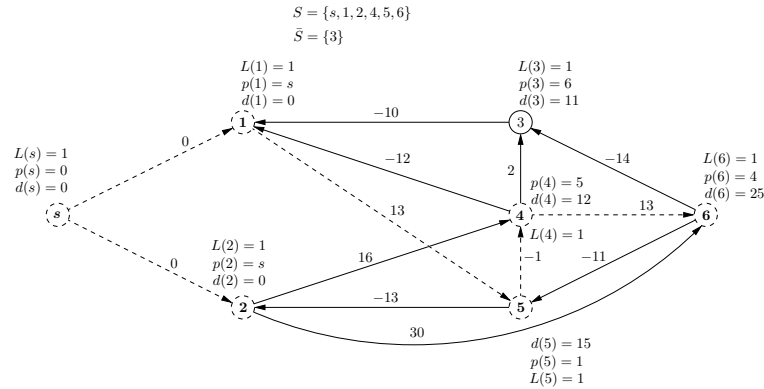


Figure E.7: Ford's Algorithm example: Sixth iteration.

As node 3 is the only temporarily labelled node, it is permanently labelled ($L(3) = 2$). It is only adjacent to node 1 and since the weight of the arc $(3, 1)$ is $c_{3,1} = -10$ and $d(1) = 0$, the distance label of node 1 does not change. Since all nodes are now permanently labelled and there are no shorter paths from s to any of the nodes, the algorithm terminates. No vertex has been labelled more than $|N|$ times, thus there are no negative cycles. The paths from node s to nodes 1 – 6 therefore have lengths 0, 0, 11, 12, 15 and 25 respectively, as depicted in Figure E.8.

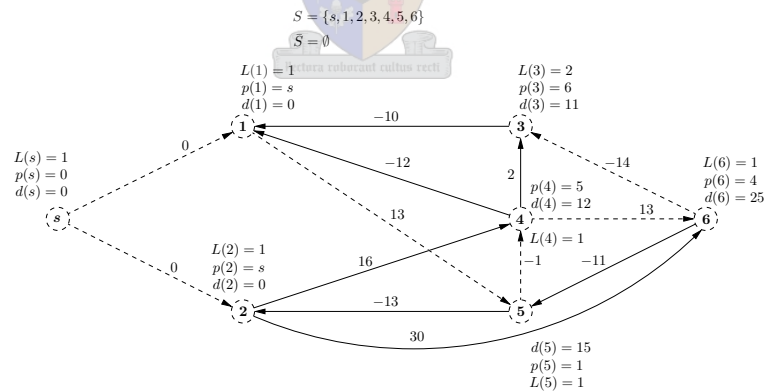


Figure E.8: Ford's Algorithm example: Seventh iteration.

E.2 Shortest Augmenting Path Algorithm

Consider the network of arcs and nodes with upper limits on the possible flow across the arcs shown in Figure E.9 and suppose the maximum flow that can pass through the network from node s to node t is sought. The shortest distances from node t are found to all other nodes by reversing the arc directions and labelling the nodes with these distance values $d(i)$, where $d(t) = 0$. In this case the distance labels are: $d(s) = 2, d(1) = 2, d(2) = 1, d(3) = 1, d(4) = 1$ and $d(5) = 1$.

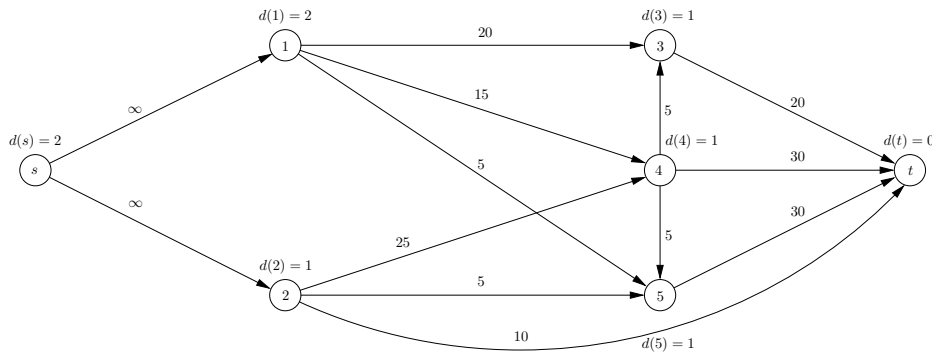


Figure E.9: Shortest Augmenting Path Algorithm: First iteration.

To find an admissible arc, the algorithm first tests whether the arc $(s, 1)$ is valid. The residual value is infinite, so the condition that $r_{s,1} > 0$ is satisfied. However, $d(s) = 2$ and $d(1) = 2$, so the second necessary condition that $d(s) = d(1) + 1$ is not met; hence the arc $(s, 1)$ is inadmissible. Thus the algorithm considers the second option, namely the arc $(s, 2)$. This arc meets both conditions, so a path from s to node 2 is created. The predecessor label of node 2 is set to $p(2) = s$. As both nodes 4 and 5 have a distance label equal to the distance label of node 2, they cannot be added to the path. The only valid arc by which to extend the path is the arc $(2, t)$. The predecessor label of node t becomes $p(t) = 2$. Since node t has been reached, the smallest residual capacity is sought along the path that has been created. Since the possible choices are ∞ or 10, only 10 units of flow will be sent from source to sink along the path $s - 2 - t$. This means that $r_{2,t} = 0$ and $r_{t,2} = 10$. Since there is no upper capacity for the arc $(s, 2)$, the residual capacity for the arc does not change, but the residual capacity for the reverse arc $(2, s)$ increases to $r_{2,s} = 10$. These changes may be found in Figure E.10.

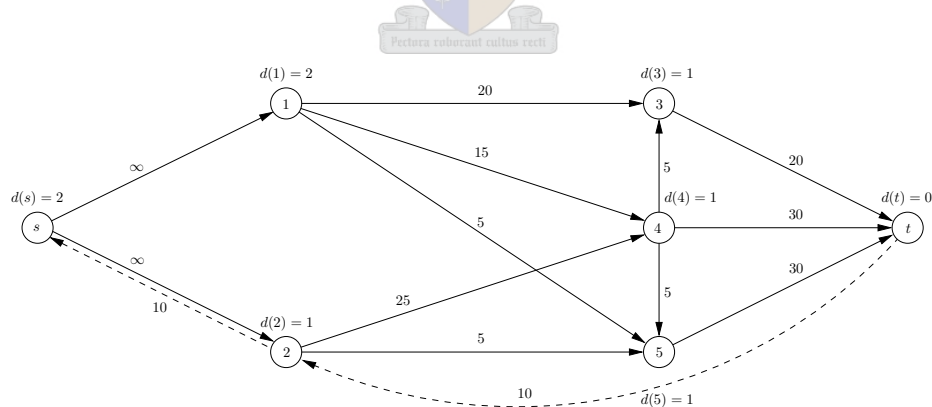


Figure E.10: Shortest Augmenting Path Algorithm: Second iteration.

As $d(s) < |N| = 7$, the algorithm continues. The arc $(s, 2)$ is still the only admissible arc adjacent from node s ; thus it is included in the path and $p(2) = s$. However, there is no admissible path from node 2 to any other nodes — hence the distance label of node 2 changes to $d(2) = d(2) + 1 = 2$ and $p(2) = s$ is considered again, thereby removing the arc $(s, 2)$ from the path. Now node s has no admissible arcs adjacent to it, so its distance label is changed to $d(s) = 3$. The arc $(s, 1)$ becomes admissible as $r_{s,1} = \infty > 0$ and $d(s) = d(1) + 1$; hence the arc $(s, 1)$ becomes the first part of the path P and $p(1) = s$. Node 1 is adjacent to three nodes which are admissible. The node with the lowest label is arbitrarily chosen and $p(3) = 1$, thus

making it the third node in path P . Node 3 is only adjacent to node t and since the arc $(3, t)$ is admissible, t becomes the final node of P . The predecessor labels are now used to find the path back to s and the lowest residual capacity is recorded. For this path, a maximum of 20 units can be sent along P to t ; hence that amount of flow is augmented along the path. Thus $r_{s,1}$ remains unchanged, $r_{1,3} = r_{3,t} = 0$ and $r_{t,3} = r_{3,1} = r_{1,s} = 20$. These changes may be found in Figure E.11.

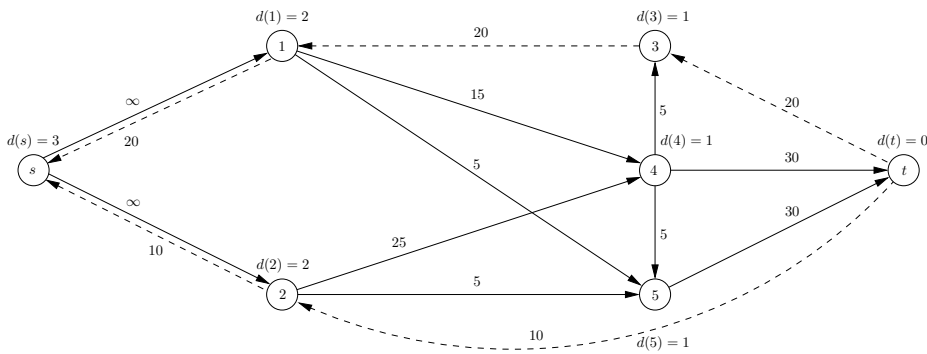


Figure E.11: *Shortest Augmenting Path Algorithm: Third iteration.*

The algorithm continues, because $d(s) < |N|$. The arc $(s, 1)$ is still admissible (so $p(1) = s$), but the arc $(1, 3)$ is no longer admissible, as $r_{1,3} = 0$ — thus the algorithm considers the arc $(1, 4)$, which is admissible. The path is lengthened to node 4 and $p(4) = 1$. Node 4 is adjacent to three nodes, but node 3 and node 5 are inadmissible because $d(3) = d(4) = d(5) = 1$, which leaves node t , which is admissible — hence $p(t) = 4$. The path is followed back to s , using the predecessor labels, and the smallest residual capacity is found. The smallest residual capacity is 15 and is found on arc $(1, 4)$; hence 15 units of flow are augmented along the path $s - 1 - 4 - t$. The residual capacities of the forward arcs now become $r_{1,4} = 0$, $r_{4,t} = 15$, while the residual capacity of arc $(s, 1)$ remains infinite. The residual values of the backward arcs change to $r_{t,4} = r_{4,1} = 15$ and $r_{1,s} = 35$ (see Figure E.12).

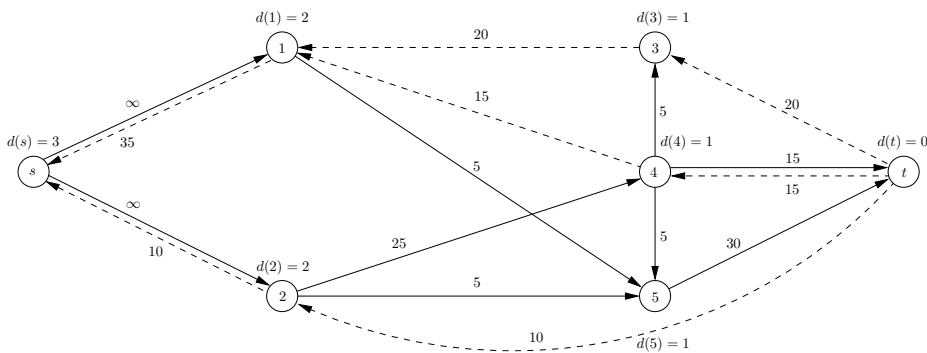


Figure E.12: *Shortest Augmenting Path Algorithm: Fourth iteration.*

Due to the fact that $d(s) < |N|$, the algorithm continues. The arc from s to node 1 remains admissible. Therefore that arc is made part of the path by setting $p(1) = s$. Only the arc from node 1 to node 5 is admissible, as $r_{1,5} = 5 > 0$ and $d(5) = d(1) + 1$. Thus the arc $(1, 5)$ is added to the path by setting $p(5) = 1$. Node t is the only node adjacent from node 5 and the arc $(5, t)$ is admissible; thus the path is extended to t and $p(t) = 5$. Now, the maximum amount

by which the flow can be augmented must be found by tracing the path back to s , using the predecessor labels and recording the residual capacity of the arc with the lowest value in the path $s - 1 - 5 - t$. The lowest residual capacity occurs on the arc $(1, 5)$, which has a residual capacity of 5; therefore the flow along the path is augmented by 5 units. This means that the residual capacities of the forward arcs become $r_{1,5} = 0$ and $r_{5,t} = 25$, while $r_{s,1} = \infty$ remains unchanged. The residual capacities of the backward arcs become $r_{t,5} = r_{5,1} = 5$ and $r_{1,s} = 40$. These changes may be found in Figure E.13.

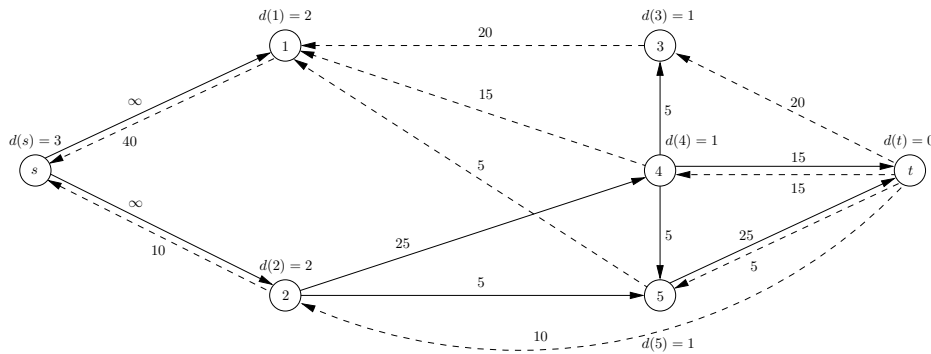


Figure E.13: Shortest Augmenting Path Algorithm: Fifth iteration.

The distance label of node s is still smaller than the order of the network (number of nodes), thus the algorithm continues. The arc $(s, 1)$ is still admissible; thus it is added to the path P and $d(1) = s$. However, node 1 does not have any admissible arcs adjacent from it — hence the distance label is increased by one to be $d(1) = 3 = d(s)$. The arc is also removed from P and the arc $(s, 2)$ is tested for admissibility. As node 2 is admissible, the arc $(s, 2)$ is added to the path P and $p(2) = s$. Node 2 has only one admissible arc adjacent from it, so the arc $(2, 4)$ is added to P and $p(4) = 2$. There is a single admissible arc adjacent from node 4 (the other two arcs point to nodes 3 and 5 which have the same distance labels as node 4) and this is the arc $(4, t)$, so it is added to the path P . Since a path has been found to t , it must be traced back to node s to determine the largest amount of flow that can be augmented on the path $s - 2 - 4 - t$. This happens to be 15 units ($r_{4,t}$ has the lowest residual capacity); so 15 units of flow are augmented along the path P . Thus the forward residual capacities become $r_{2,4} = 10$ and $r_{4,t} = 0$, while $r_{s,2} = \infty$ remains unchanged. The residual values of the backward arcs change to $r_{t,4} = 30$, $r_{4,2} = 15$ and $r_{2,s} = 25$. These changes may be found in Figure E.14.

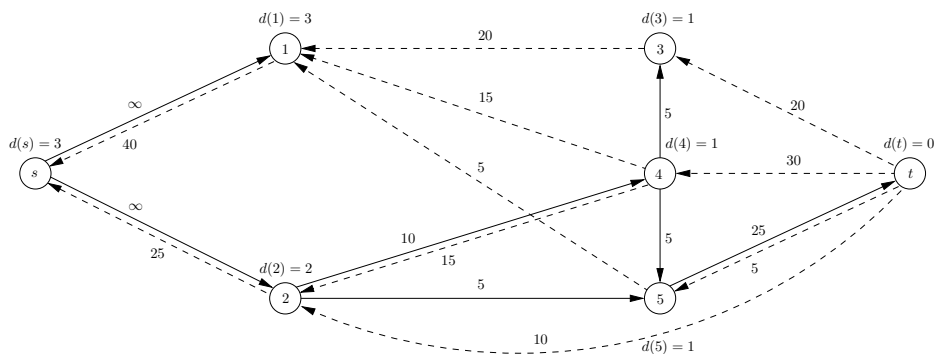


Figure E.14: Shortest Augmenting Path Algorithm: Sixth iteration.

The condition for termination has not been met, so the algorithm continues. The arc $(s, 2)$ remains the only admissible arc adjacent from the source — so it is added to the path P and $p(2) = s$. The arcs $(2, 4)$ and $(2, 5)$ are admissible, and since node 4 has the lower node label, the arc $(2, 4)$ is added to P . However, there are no admissible arcs adjacent to node 4; hence the distance label of node 4 is increased by one and the path retreats to the predecessor of node 4, namely to node $p(4) = 2$. Now the arc $(2, 4)$ is inadmissible, so the arc $(2, 5)$ is added to P and the predecessor of node 5 is set to $p(5) = 2$. There is a single admissible arc adjacent from node 5. Therefore the arc $(5, t)$ is added to P and $p(t) = 5$. As P now leads from source to sink, the maximum augmenting volume must be found. The arc $(2, 5)$ has the lowest residual capacity in P ; hence the flow is augmented by $r_{2,5} = 5$ units along P . Thus the residual capacities of the forward arcs become $r_{2,5} = 0$ and $r_{5,t} = 20$, while $r_{s,2}$ remains infinite. The residual capacities of the backward arcs, in turn, become $r_{t,5} = 10$, $r_{5,2} = 5$ and $r_{2,s} = 30$. These changes may be found in Figure E.15.

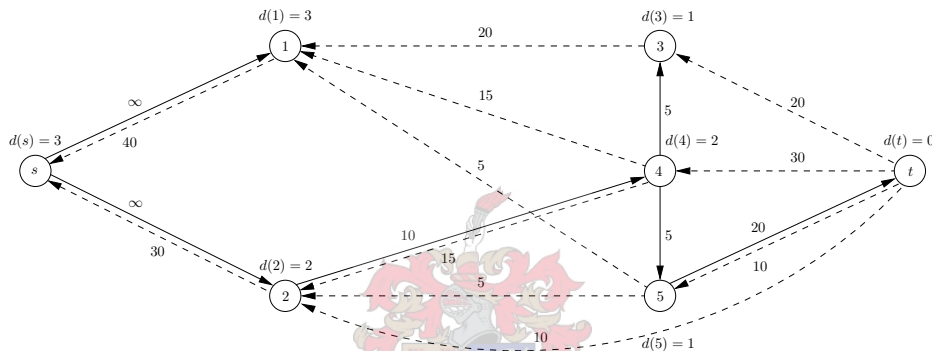


Figure E.15: Shortest Augmenting Path Algorithm: Seventh iteration.

The distance label on s is still smaller than the number of nodes in the network — therefore the algorithm continues. The arc $(s, 2)$ is the only admissible arc adjacent from the source, so it is added to the path P and $p(2) = s$. There are no more admissible arcs remaining that are adjacent from node 2. Hence the distance label on node 2 is increased to $d(2) = 3$ and the arc is removed from P . As the source now has no admissible arcs adjacent from it, the distance label is increased to $d(s) = 4$. The arc $(s, 1)$ is now admissible again, so it is added to the path P . However, there are no arcs adjacent from node 1 with $r > 0$, so the distance label of node 1 is increased to $d(1) = 4$ and the arc is removed from P . The arc $(s, 2)$ is admissible — therefore it is added to the path and $p(2) = s$; so is the arc $(2, 4)$ and the predecessor of node 4 is set to be node 2. There are now two admissible arcs adjacent from node 4 and since node 3 has the lower label, it is chosen first. The arc $(4, 3)$ is added to P and $p(3) = 4$, but since node 3 has no adjacent nodes from it satisfying the admissibility conditions, its distance label is increased to $d(3) = 2$. This leaves node 5, which replaces node 3 in P , so $p(5) = 4$. It is also adjacent to node t by means of an admissible arc. Therefore node t is added to the path and $p(t) = 5$. To find the maximum amount by which the flow can be augmented, the path must be traced back to s , noting the residual capacities of all of the forward arcs, the minimum of which is 5 units — found on the arc $(4, 5)$. Thus 5 units of flow are augmented along the path $s - 2 - 4 - 5 - t$ so that the residual capacities of the forward arcs are $r_{s,2} = \infty$, $r_{2,4} = 5$, $r_{4,5} = 0$ and $r_{5,t} = 15$. The residual capacities of the backward arcs become $r_{t,5} = 15$, $r_{5,4} = 5$, $r_{4,2} = 20$ and $r_{2,s} = 35$. These changes may be found in Figure E.16.

The termination condition has not yet been met — so the algorithm continues. The arc $(s, 2)$ is admissible, so the path begins at s and moves to node 2, which means that $p(2) = s$. The

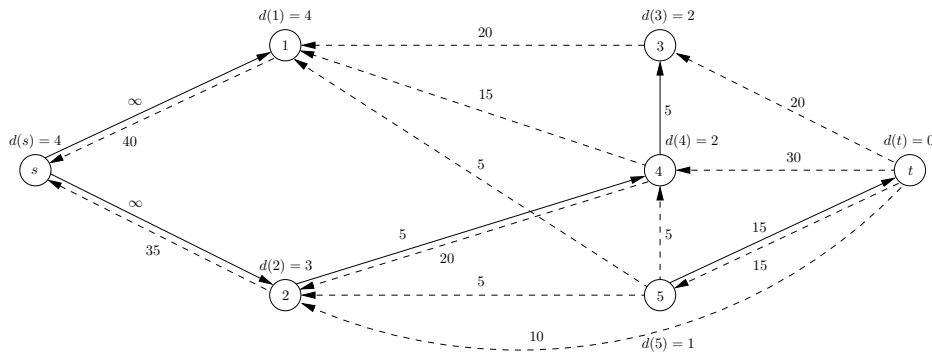


Figure E.16: Shortest Augmenting Path Algorithm: Eighth iteration.

arc $(2, 4)$ is the only arc adjacent from node 2 that is admissible, so the path continues to node 4 and $p(4) = 2$. There are no admissible arcs adjacent to node 4 (as $r_{4,x} = 0$ or $d(4) = d(x)$ for all arcs adjacent from node 4). Therefore the distance label on node 4 is increased to $d(4) = 3$ and the path retreats to the predecessor of node 4 in the path, namely node $p(4) = 2$. Node 2 now has no admissible paths adjacent from it; hence its distance label is also increased, this time to $d(2) = 4$. Similarly, node s now has no admissible arcs adjacent from it, so the distance label of the source becomes $d(s) = 5$. Node s has two admissible arcs adjacent from it; thus a path is created from node s to node 1. There is no opportunity for the path to extend beyond node 1. Therefore the distance label of node 1 is increased to $d(1) = 5$. The only remaining admissible arc adjacent from node s is $(s, 2)$; thus a path is created from node s to node 2 and $p(2) = s$. Node 2 has only one admissible arc incident to it and this arc allows flow from node 2 to node 4, where $p(4) = 2$. The path can be extended to node 3 along the admissible arc $(4, 3)$, so $p(3) = 4$. The path cannot extend further than node 3, so its distance node is increased to $d(3) = 3$ and node 4 becomes the end of the path. This node now has no admissible arcs adjacent from it, so node 2 becomes the end of the path and the distance label of node 4 becomes $d(4) = 4$. This process continues until the source is reached, which leaves it with a distance label of $d(s) = 6$. The arc $(s, 1)$ becomes valid again. Hence it is placed in a path, which cannot continue, so $d(1) = 6$. The path $s - 2 - 4 - 3$ will again be discovered, but as it cannot continue, the distance labels increase and the path is traced back until $d(s) = 7$, which satisfies the termination condition — hence the algorithm terminates. This leaves the residual capacities from which the maximum flow that can flow through the network can be found. The network is shown in Figure E.17 together with the true flow values (not the residual capacities).

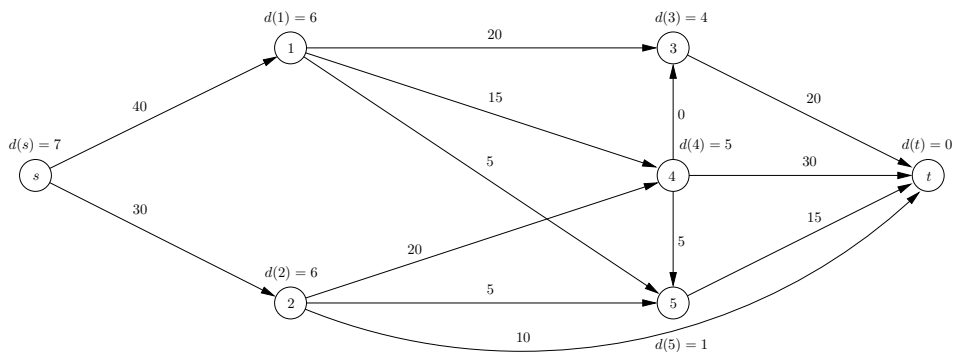


Figure E.17: Shortest Augmenting Path Algorithm: Final flow.

E.3 Successive Shortest Path Algorithm

Consider a network of arcs and nodes with upper and lower limits on the possible flow across the arcs, and suppose the minimum cost flow that can pass through the network is sought so that the demand and supply are exhausted. A network with a supply of 30 units at nodes 1 and 2, which balances with the demand of 60 units at node 6 is shown in Figure E.18. The arcs are labelled in the form $(c_{ij}, \ell_{ij}, u_{ij})$, where c_{ij} denotes the cost of a unit of flow moving from node i to node j and ℓ_{ij} and u_{ij} are the lower and upper bounds on the flow across the arc (i, j) respectively. For example, the arc $(4, 6)$ has a cost of 13 per unit flow, a lower bound of 10 units and an upper bound of 20 units that may flow along it.

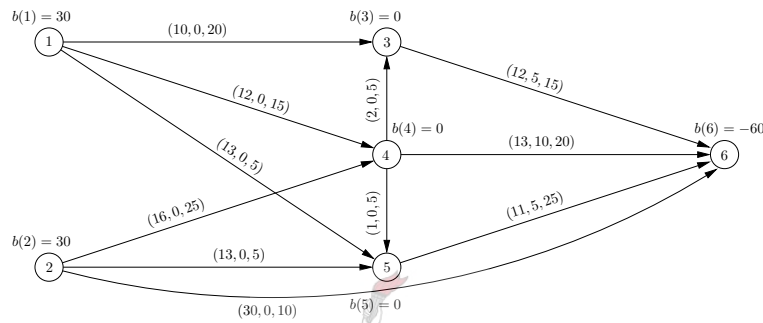


Figure E.18: *Successive Shortest Path Algorithm: Original network.*

The *Successive shortest path algorithm* begins by setting the flow x and potentials π equal to zero. The imbalance labels e are set equal to the supply b and the lower bounds are removed. Since there are only three arcs with non-zero lower bounds, only these have to be considered. To remove the lower bound of 5 units from the arc $(3, 6)$, 5 units are subtracted from the excess of node 3, so that $e(3) = -5$; 5 units are added to the excess of node 6, so that $e(6) = -60 + 5 = -55$, and 5 units are subtracted from the upper bound on the arc, so that $u_{3,6} = 15 - 5 = 10$. The same is done for arcs $(4, 6)$ and $(5, 6)$; so that $e(4) = -10$, $e(5) = -5$, $e(6) = -40$, $u_{4,6} = 10$ and $u_{5,6} = 20$. The residual capacities can now be set such that $r_{ij} = u_{ij}$, $r_{ji} = 0$ for all arcs in the network. The result is as shown in Figure E.19.

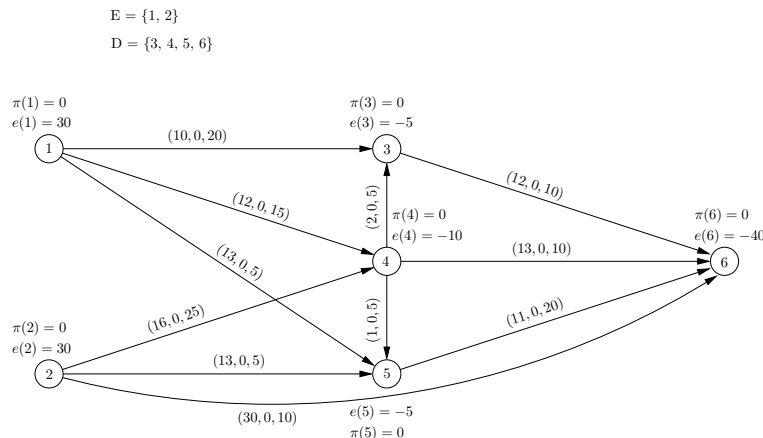


Figure E.19: *Successive Shortest Path Algorithm: Network preparation.*

Node 1 is the first node with a positive imbalance. Therefore $s = 1$ and since node 3 is the node with a negative imbalance with the lowest node label, $t = 3$. The reduced costs may now be calculated for the entire network. Since the node potentials are still zero, the reduced costs are simply the same as the corresponding arc costs. With these reduced costs, the new node potentials may be calculated (see the π node labels in Figure E.20), and the shortest distance may be found from a source node to all nodes (see the value of $d(i)$ for the distance from s to i in Figure E.20). It turns out that the shortest path from node 1 to node 3 is along the arc (1, 3). Thus $\min\{e(1), -e(3), r_{1,3}\} = \min\{30, 5, 20\} = 5$, so that 5 units of flow are augmented along the path. This means that the residual capacities $r_{1,3} = 15$ and $r_{3,1} = 5$, and the imbalances change to $e(1) = 30 - 5 = 25$ and $e(3) = (-5) + 5 = 0$. Note that arc labels in Figure E.20 have been changed to (c_{ij}^π, r_{ij}) , where c_{ij}^π denotes the reduced cost of the arc (i, j) and r_{ij} denotes the residual capacity of the arc.

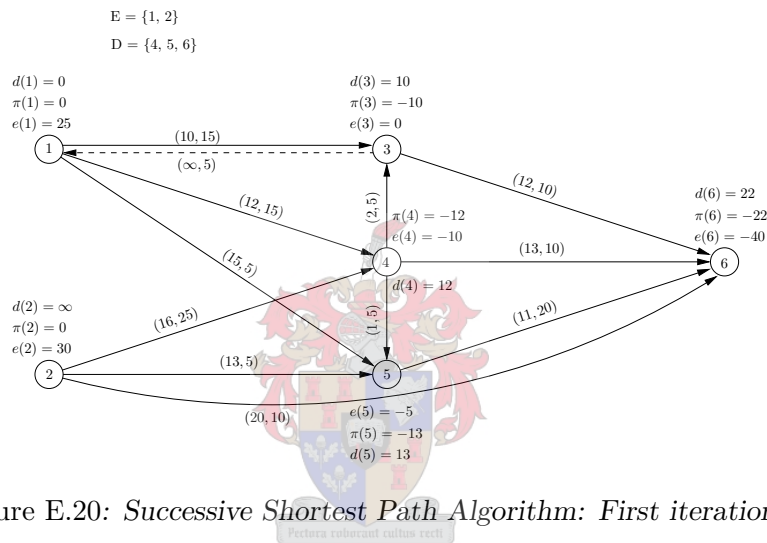


Figure E.20: Successive Shortest Path Algorithm: First iteration.

The demand at node 3 has now been met. Therefore the algorithm looks for the next node with a negative imbalance. Node 4 has $e(4) = -10$, so it is made the sink and $t = 4$. The source remains at node 1 as its imbalance is still larger than zero, so that $s = 1$. The new reduced cost of the arcs may now be calculated (see the arc labels in Figure E.21 for the calculated values), for example $c_{1,4}^\pi = c_{1,4} - \pi(1) + \pi(4) = 12 - 0 + (-12) = 0$. Now the shortest distances to each of the nodes may be found. Using the distance labels the new potentials may also be calculated for the next iteration. There is no path from node 1 to node 2 — hence the potential must be set equal to 0, otherwise $\pi(2) = \pi(2) - d(2) = -\infty$. It turns out that the shortest path from node 1 to node 4 is along the arc (1, 4) which has a residual capacity $r_{1,4} = 15$. Now $\min\{e(1), -e(4), r_{1,4}\} = \min\{25, 10, 15\} = 10$ units of flow must be augmented along the arc (1, 4), so that the residual capacities become $r_{1,4} = 5$ and $r_{4,1} = 10$, while all other residual capacities remain the same. The imbalance of node 4 is now zero — therefore it may be removed from the demand set D .

The imbalance of node 1 is still larger than zero, so that $s = 1$. Since the imbalance of node 4 is now zero, the next node to be investigated for a negative imbalance is node 5. Since $e(5) = -5$, node 5 is labelled as the sink by setting $t = 5$. The new reduced costs may be found — for example, $c_{1,5}^\pi = 13 - 0 + (-13) = 0$ — and using these to find the shortest path distances results in the $d(i)$ labels in Figure E.22. The new distance labels may be used to update the potentials, taking care to set $\pi(2) = 0$, since there is no 1 – 2 path in the network. Since the shortest path is along arcs (1, 4) and (4, 5), $\min\{15, 5, 5, 5\} = 5$ units of flow may be augmented along the

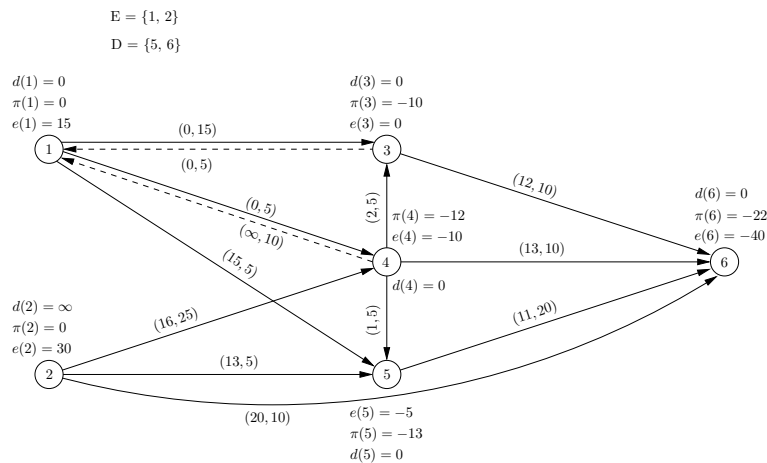


Figure E.21: Successive Shortest Path Algorithm: Second iteration.

path. This leaves $e(i) = 10$, $e(5) = 0$, $r_{1,4} = 0$, $r_{4,1} = 15$, $r_{4,5} = 0$ and $r_{5,4} = 5$, which means that node 5 can be removed from set D .

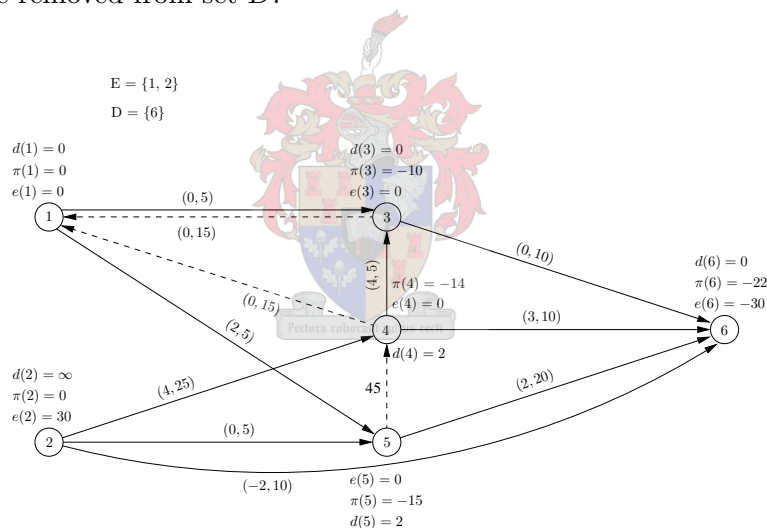


Figure E.22: Successive Shortest Path Algorithm: Third iteration.

The imbalance of node 5 is now cancelled, but as node 1 still has a positive imbalance, $s = 1$. Node 6 is the only remaining node with a deficit. Therefore a shortest path must be found from node 1 to node 6. Now the new reduced costs may be calculated, e.g. $c_{5,4}^\pi = -1 - (-13) + (12) = 0$. Using these reduced costs, the shortest distance may be found from node 1 to all other nodes in the network, and then the potentials of each node is known. The shortest distance to node 6 is along the path 1 – 3 – 6, where the arc (1, 3) has a residual capacity of 15 and the arc (3, 6) has a residual capacity of 10. As the excess at node 1 is 10 and the deficit at node 6 is 40, the minimum must be chosen as the augmentation volume. The minimum is 10; thus 10 units of flow are augmented along the path and the deficit at node 6 becomes 30. The excess at node 1 becomes 0 and the residual capacity of the arc (1, 3) is reduced to 5, whilst the residual capacity of the arc (3, 1) becomes 15, and that of the arc (3, 6) reduces to zero. The residual capacity of the arc (6, 3) increases to 10. Now node 1 may be removed from the set E (see Figure E.23).

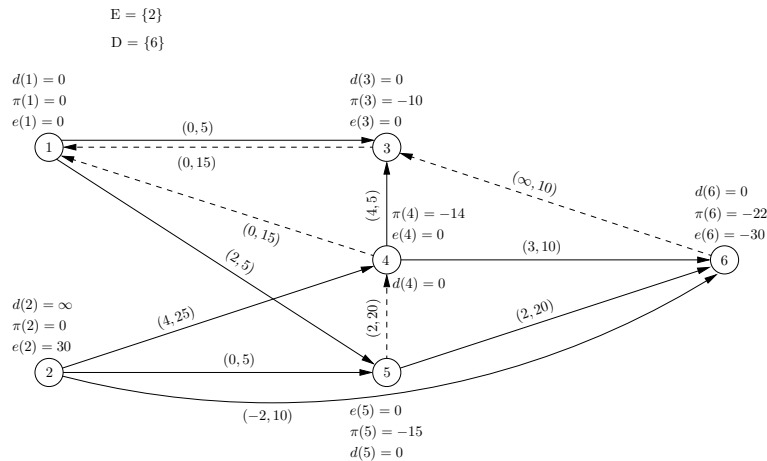


Figure E.23: Successive Shortest Path Algorithm: Fourth iteration.

The excess at node 1 is zero, thus the next element in the set E is considered; node 2 in this case. Node 6 remains the sink as its demand has not been satisfied. Therefore the shortest path from node 2 to node 6 must now be found. The new reduced costs may be found from the new source — for example, $c_{2,5}^\pi = 13 - 0 + (-15) = -2$. Using these new reduced costs, the shortest distances to all nodes may be found from node 2 (see label d in Figure E.24). The new potentials may then be calculated using the new shortest distances, so that, for example, $\pi(5) = \pi(5) - d(5) = -15 - (-2) = -13$. The shortest distance from source to sink is thus -2 along the arc $(2, 6)$ with a residual capacity of $r_{2,6} = 10$ units. Now $\min\{e(2), r_{2,6}, -e(6)\} = \min\{30, 10, -(-30)\} = 10$ units of flow must be augmented along the arc $(2, 6)$, so that $e(2) = 20$, $r_{2,6} = 0$, $r_{6,2} = 10$ and $e(6) = -20$, while all other parameters remain constant. The imbalance of neither the source nor the sink has been satisfied; therefore the excess and demand sets remain as in the previous iteration.

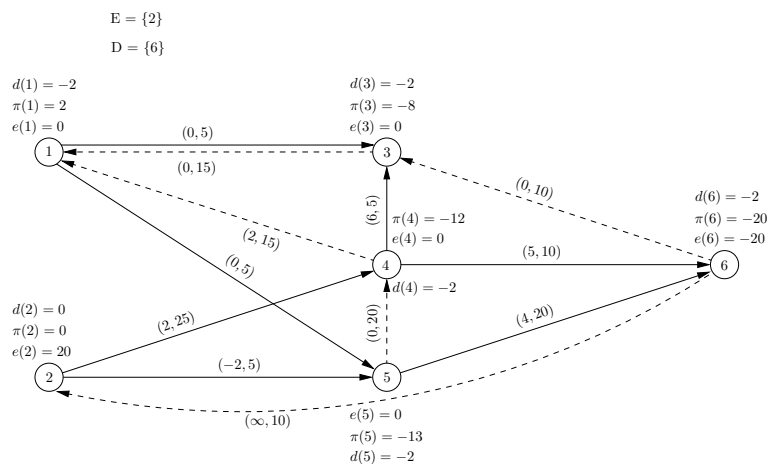


Figure E.24: Successive Shortest Path Algorithm: Fifth iteration.

The imbalances of node 2 and node 6 are still larger and less than zero respectively, so that $s = 2$ and $t = 6$, as before. The new reduced costs may be found via the potentials from the previous iteration — for example, $c_{2,4}^\pi = c_{2,4} - \pi(2) + \pi(4) = 16 - 0 + (-12) = 4$. These may be used to find the shortest path distances, resulting in the $d(i)$ labels shown in Figure E.25. The new distance

labels may then be used to update the potentials, e.g. $\pi(6) = \pi(6) - d(6) = -20 - 4 = -24$. Since the shortest path is along arcs (2, 5) and (5, 6), the maximum number of units of flow that may be augmented along the path is $\min\{e(2), r_{2,5}, r_{5,6}, -e(6)\} = \min\{20, 5, 20, 20\} = 5$. This leaves the imbalances and residual capacities as follows: $e(2) = 15$, $e(6) = -15$, $r_{2,5} = 0$, $r_{5,2} = 5$, $r_{5,6} = 15$ and $r_{6,5} = 5$, which means that the excess and deficit sets remain unchanged.

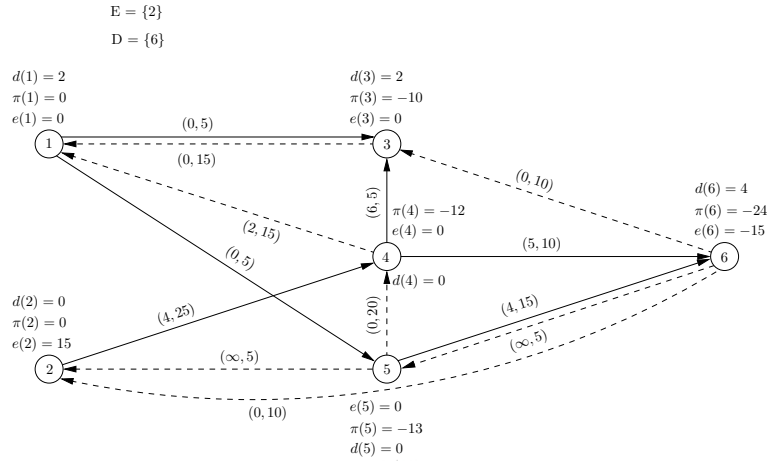


Figure E.25: Successive Shortest Path Algorithm: Sixth iteration.

There is no change in the excess and deficit sets at this stage of the algorithm execution, so that node 2 remains the source and node 6 remains the sink. The new reduced costs may be found with the new potentials, e.g. $c_{4,3}^{\pi} = c_{4,3} - \pi(4) + \pi(3) = 2 - (-12) + (-10) = 4$. The reduced costs contribute to finding the distance labels on each node, which, in turn, are used to find the new potentials, such as $\pi(4) = \pi(4) - d(4) = -12 - 4 = -16$. The shortest distance from source to sink is along the arcs (2, 4) and (4, 6). Therefore the maximum flow that may be augmented along the path is $\min\{e(2), r_{2,4}, r_{4,6}, -e(6)\} = \min\{15, 25, 10, 15\} = 10$ units. This means that the imbalances reduce to $e(2) = 5$ and $e(6) = -5$, while the residual capacities become $r_{2,4} = 15$, $r_{4,2} = 10$, $r_{4,6} = 0$ and $r_{6,4} = 10$. The excess and deficit sets do not change, as shown in Figure E.26.

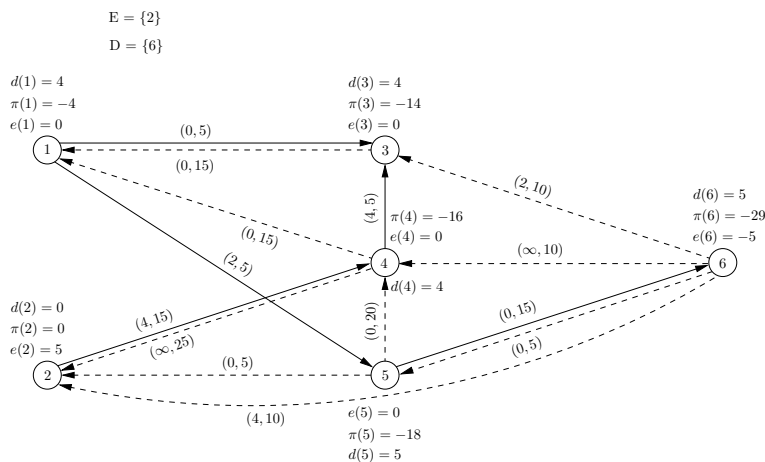


Figure E.26: Successive Shortest Path Algorithm: Seventh iteration.

The source and sink nodes remain the same for this iteration of the Successive Shortest Path algorithm. The new reduced costs are calculated from the original cost matrix and the potentials found in the previous iteration, e.g. $c_{4,1}^\pi = c_{4,1} - \pi(4) + \pi(1) = -12 - (-16) + (-4) = 0$. These reduced costs are then used to find the shortest distances from node 2 to all other nodes (the d labels in Figure E.27), which, in turn, are used to find the new potentials. The shortest distance from source to sink is along the path $2 - 4 - 1 - 5 - 6$ and the maximum number of units of flow that may be augmented along the path is $\min\{e(2), r_{2,4}, r_{4,1}, r_{1,5}, r_{5,6}, -e(6)\} = \min\{5, 15, 15, 5, 15, 5\} = 5$ units. This cancels the imbalances at nodes 2 and 6 and the residual capacities become $r_{2,4} = 10, r_{4,2} = 15, r_{4,1} = 10, r_{1,4} = 5, r_{1,5} = 0, r_{5,1} = 5, r_{5,6} = 10$ and $r_{6,5} = 10$. As all imbalances have been cancelled, the excess and deficit sets are now empty.

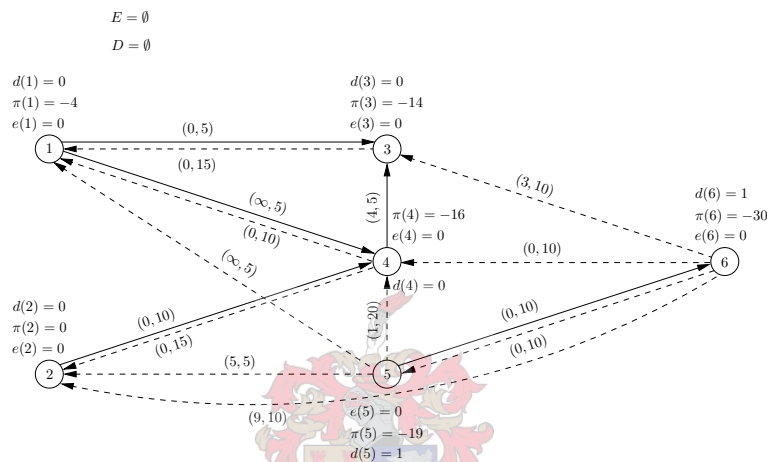


Figure E.27: Successive Shortest Path Algorithm: Eighth iteration.

Now that there are no nodes with an imbalance, the algorithm terminates. The resulting minimum cost flow is shown in Figure E.28. The minimum network admits cost flow of 1 460 cost units.

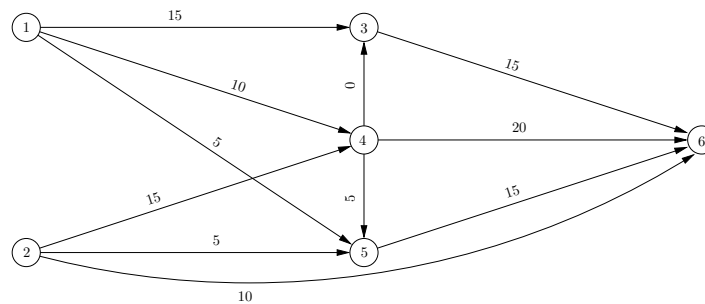


Figure E.28: Successive Shortest Path Algorithm: Final flow.



Appendix F

Transportation Costs

This appendix contains road transportation costs given to the author by A du Bois [94] (transport manager at FPT in Cape Town) and by R Opperman [228] (perishable product manager at Kuehne + Nagel in Airport Industria near Cape Town). The costs in bold face in Tables F.1 and F.3 were those given to the author, the others were calculated from the lines of best fit shown in Figures F.1 and F.2. The line of best fit for the cost of the transportation of fruit in refrigerated vehicles is shown in Figure F.3. Due to the high margin of error for this best fit line, it was not used to estimate any further costs. Therefore, only those costs provided to the author were used in the models. Due to the high costs of transporting fruit by refrigerated vehicles, this is an unlikely choice for transportation and only those routes shown in Table F.3 are occasionally used.

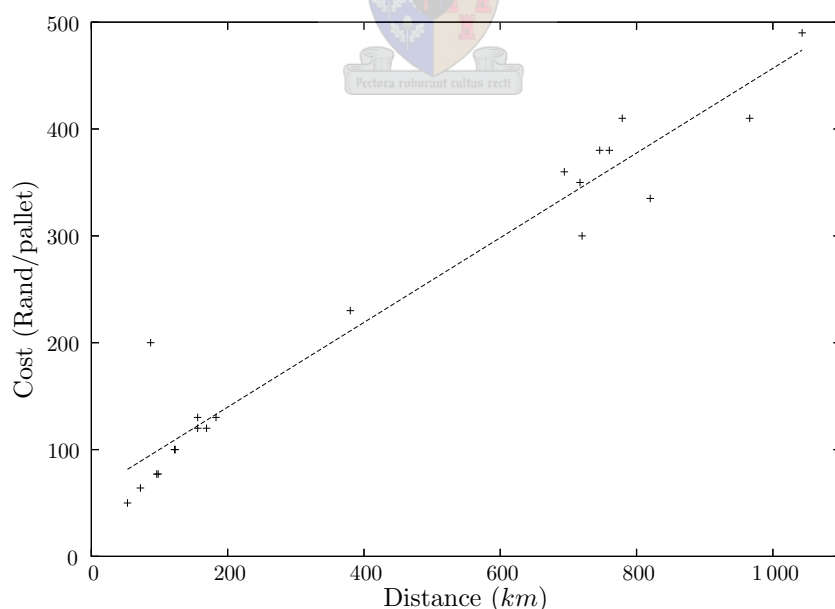


Figure F.1: Line of best fit for provided unrefrigerated road transport costs. The slope of the line of best fit is 0.396 Rand/km/pallet and the y -axis intersection is 60.5 Rand/pallet ($R^2 = 0.9428$).

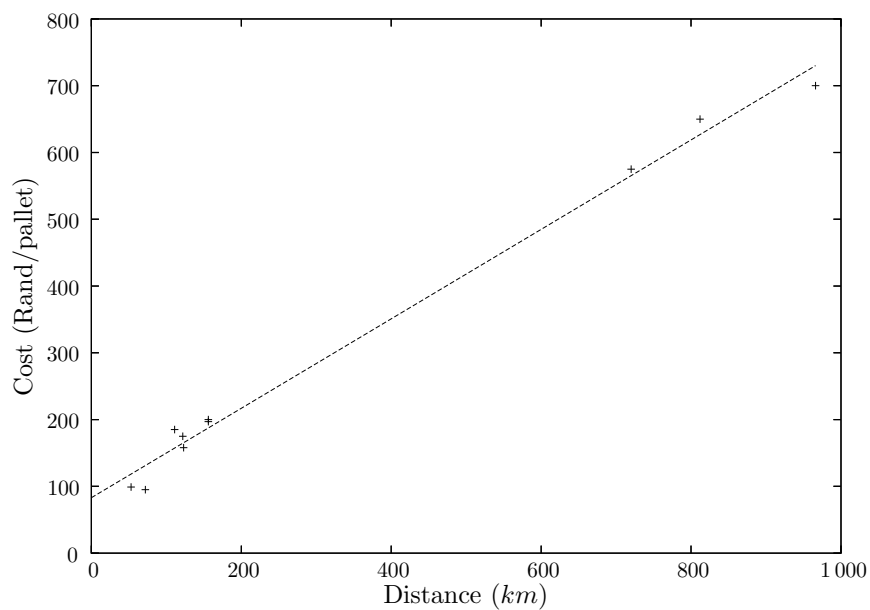


Figure F.2: Line of best fit for provided road transport costs for containerised fruit. The slope of the line of best fit is 0.670 Rand/km/pallet and the y-axis intersection is 82.9 Rand/pallet ($R^2 = 0.9914$).

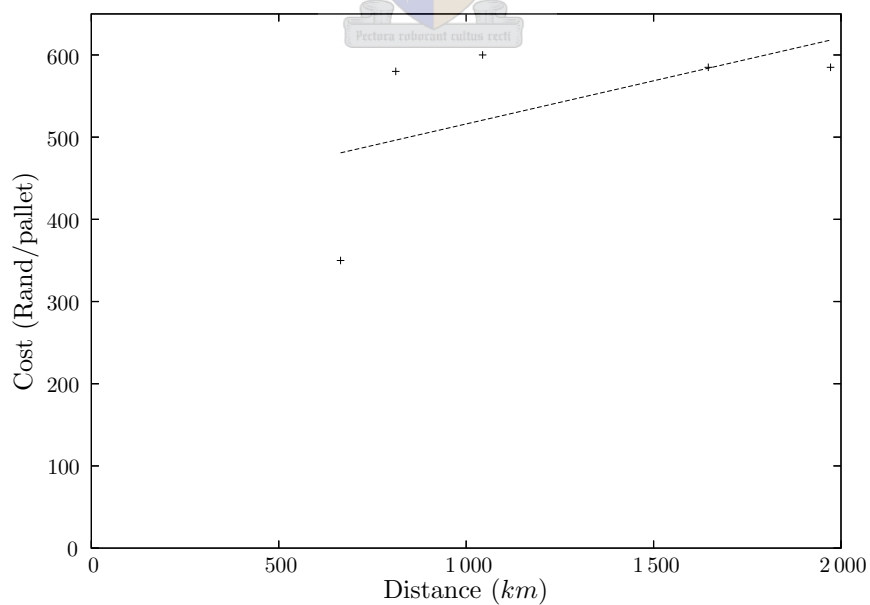


Figure F.3: Line of best fit for provided refrigerated fruit road transport costs. The slope of the line of best fit is 0.105 Rand/km/pallet and the y-axis intersection is 411 Rand/pallet ($R^2 = 0.3049$).

	Cape Town (R/pallet)	Durban (R/pallet)	Port Elizabeth (R/pallet)	Maputo (R/pallet)
Levubu	800	490	683	334
Tzaneen	769	410	653	265
Hoedspruit	897	410	660	200
Kiepersol	891	380	656	167
Nelspruit	865	360	630	140
Malelane	900	380	665	115
Potgietersrus	712	335	596	305
Ellisras	752	418	618	350
Marble Hall	841	300	606	266
Ermelo	811	256	579	234
Pongola	841	230	591	294
Nkwalini	757	120	641	350
KZN Midlands	689	200	401	393
EC Midlands	435	380	147	758
Sundays River	386	456	77	834
Patensie	335	481	77	753
Langkloof	324	511	130	779
Karoo	130	811	234	579
Klein Karoo	130	683	317	822
GEVV	64	686	321	825
Boland	50	680	345	819
Ceres	100	664	357	803
De Doorns	80	673	349	808
Piketberg	100	703	369	842
Citrusdal	120	718	411	857
Orange River	382	536	434	573
Hartswater	474	364	414	458
Johannesburg	617	297	481	282
Rustenburg	780	350	559	324
Free State	544	214	428	339
Middelburg	792	299	603	229

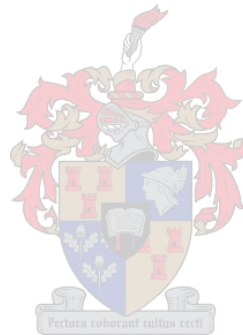
Table F.1: *The estimated costs of transporting fruit by tautliner during 2005. Bold figures indicate values that were given to the author, while those in normal font were calculated by multiplying the distance between two points with the slope of the line of best fit (see Figure F.1) and adding the y -intersection value.*

	Cape Town (R/pallet)	Durban (R/pallet)	Port Elizabeth (R/pallet)	Maputo (R/pallet)
Levubu	1 334	782	1 137	545
Tzaneen	1 282	700	1 085	429
Hoedspruit	1 499	605	1 097	319
Kiepersol	1 489	592	1 091	262
Nelspruit	1 444	548	1 046	218
Malelane	1 504	583	1 106	176
Potgietersrus	1 186	632	989	497
Ellisras	1 253	689	1 027	573
Marble Hall	1 404	575	1 006	430
Ermelo	1 353	414	961	376
Pongola	1 403	338	981	479
Nkwalini	1 262	196	1 065	573
Richmond	1 146	141	659	646
EC Midlands	717	624	230	1 263
Sundays River	634	753	147	1 391
Patensie	547	794	149	1 254
Langkloof	528	845	206	1 298
Karoo	299	1 353	376	961
Klein Karoo	197	1 135	518	1 371
GEVV	95	1 142	524	1 377
Boland	99	1 131	565	1 367
Ceres	158	1 104	584	1 339
De Doorns	185	1 120	571	1 348
Piketberg	175	1 170	606	1 405
Citrusdal	200	1 196	675	1 431
Orange River	650	888	715	950
Hartswater	782	597	681	756
Johannesburg	1 024	484	794	458
Rustenburg	1 300	563	926	529
Free State	901	343	704	555
Middelburg	1 320	486	1 000	368

Table F.2: *The estimated costs of transporting fruit stored in 12m integral containers during 2005. Bold figures indicate values that were given to the author, while those in normal font were calculated by multiplying the distance between two points with the slope of the line of best fit (see Figure F.2) and adding the y-intersection value.*

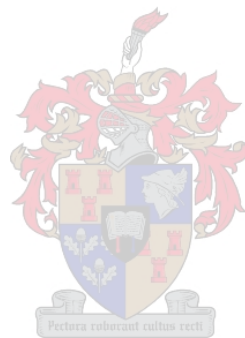
Origin	Cost (R/pallet)
Potgietersrus	585
Ellisras	640
Marble Hall	585
Langkloof	350
Orange River	580
Hartswater	600

Table F.3: *Road transportation costs to Cape Town during 2005 for fruit in refrigerated vehicles. Only the costs provided to the author are listed, as the line of best fit was found to yield an unsatisfactory approximation.*



Origin	Destination	Cost (R/pallet)
Tzaneen	Durban	371
Hoedspruit	Durban	330
Nelspruit	Durban	322
Malelane	Durban	322
Sundays River Valley	Port Elizabeth	77

Table F.4: *Unrefrigerated rail transportation costs for 2005 for hard citrus fruit.*



Appendix G

Computer Codes

This appendix contains the *Fortran 90* code for the two programs written by the author during this study. The first, called *FruitFly*, is the program used to read a network structure from a spreadsheet and find either the maximum flow or the minimum cost flow, as specified by the user. The second program, simply called *Convert*, converts a linear programming model created in a spreadsheet to the *Mathematical Programming Standard (MPS)* format, developed by *IBM* and used by a large number of solvers as an input format. The program then converts the output from either *PCx* or the online *Network-Enabled Optimization System (NEOS) Server*, to a more easily readable spreadsheet format.

A file format commonly used for both programs is the *CSV* format. This is a file format used by spreadsheet programs to save data in a text format. The rows of cells in a spreadsheet are saved as rows in the *CSV* file and the columns are separated by commas. If a cell in a spreadsheet is text, the text string is enclosed by inverted commas, thus making it possible to have text strings with commas that will not be considered as delimiters between columns. This file format is useful as programs written in *Fortran 90* require text files as input and recognise the commas as delimiters. However, the drawback of the *CSV* format as a format in which to save spreadsheets is that formulas or cell formats cannot be stored. Instead, only the result of the formula is saved and no formatting is saved. This property makes the *CSV* format ideal for *FruitFly* and *Convert*, as they have both been written to read values and not formulas.

G.1 The Program *FruitFly*

The program *FruitFly* requires the user to create a *CSV* file in which a network model is described. This input file has one line which may contain any information useful to the user. It may typically be used to store a label for the information stored in each column. There are six columns in the input file. The first column represents a node and the second another node, when there is an arc from the first node to the second. The third column is the supply that the first node has and the fourth column is the weight of the arc between the node in the first column to the node in the second column. The fifth and sixth columns are for lower and upper bounds on the flow across the arc between the nodes in the first and second columns. It is important that all nodes appear in the first column. Hence, if a node has no arcs incident from it, it must be entered in the first column of the input file, with 'END' appearing as the node in the second column. This informs the program that there are no arcs adjacent from the node. The supply (or demand in this case) of this node should be a negative value. If it were zero, it

may as well be removed from the model, and if it is positive it is a source node, but as there are no arcs adjacent from the node, the supply cannot be removed. Another important requirement is that all arcs adjacent from an arc must be listed consecutively. If a node name appears more than once in the first column and other node names appear between the appearances, the program will report an error. The *Fortran 90* code is shown below, and may also be found on the compact disc attached to this thesis.

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Module Tools !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

module Tools
!! Module tools contains all useful subroutines for the main program
  implicit none

  ! Make the integers double precision
  integer, parameter :: ip = selected_int_kind(8)

  contains

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Ford's Algorithm Real !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

  subroutine FordR(Start,Order,Large,Weight,Exist,Distance,Predecessor)
!! This subroutine uses Ford's Algorithm to find the shortest distances between the source and all other nodes in the
  graph
!! It accepts an integer (Start) which indicates which node is that start node for the shortest distance algorithm
!! An integer (Order) indicating the order of the graph (the number of nodes in the graph, excluding the source.
  There is no sink)
!! An integer (Large) indicating which large number was used to indicate missing arcs in Cost
!! A real array (Weight) indicating the weight on each arc, Large is used to indicate missing arcs
!! An integer array (Exist) indicating the upper limit of an arc i.e. 0 if it does not exist > 0 if it does exist
!! It returns a real array (Distance) indicating the shortest distance to each node from the source
!! An integer array (Predecessor) indicating which node is the predecessor for the shortest distance from the source
  implicit none
  integer(ip), intent(in) :: Order, Start
  integer(ip), intent(in) :: Large
  real, dimension(0:), intent(out) :: Distance
  real, dimension(0:,0:), intent(in) :: Weight
  integer(ip), dimension(0:), intent(out) :: Predecessor
  integer(ip), dimension(0:,0:), intent(in) :: Exist
  real :: Minimum ! Used to find the minimum from a group of options
  integer(ip) :: h, i, j, err
  integer(ip), allocatable, dimension(:) :: NodeLabelCount ! Count how many times a node has been labelled,
  if it equals Order, a negative cycle exists
  integer, allocatable, dimension(:) :: NodeLabel ! Check which labels have been permanently labelled

  ! Make the size of the array containing node labels equal to the number of nodes in the network
  allocate(NodeLabel(0:Order))
  allocate(NodeLabelCount(0:Order))

  Distance = real(Large) ! Set the distance labels equal to a large number
  Distance(Start) = 0. ! Set the distance label at the source equal to zero
  Predecessor = -1 ! Set the predecessor labels equal to an impossible number to prevent accidental predecessor
  allocation
  Minimum = real(Large) ! So that finding a minimum is a minimum of the network, not accidentally set by the
  compiler
  NodeLabel = 0 ! All nodes have yet been labelled
  NodeLabelCount = 0 ! All nodes have yet been labelled

  j = Start ! Set the starting point equal to the node which the main program sets
  err = 0 ! Allows the conditional do loop to be entered into

  ! Continue the do loop until all nodes have been permanently labelled
  do while (err < 1)

    h = j

```

```

Minimum = real(Large) ! Set the min. equal to a high number so that the min. distance can be found
if (NodeLabel(h) == 2) then
  write(*,*)'Node ', trim(adjustl(Name(h))), ' is already permanently labelled'
  read(*,*)
end if
NodeLabel(h) = 2 ! Permanently label the node

! Check if any negative cycles exist before they make the program run continuously, could also mean that not
! all nodes have been connected properly and thus will not get labelled permanently
NodeLabelCount(h) = NodeLabelCount(h) + 1
if (NodeLabelCount(h) == Order) then
  write(*,*)'There is a negative cycle.'
  return ! Return to the main program as there is a negative cycle
end if

! Find the shortest path to each of the nodes
do i=1,Order
  if (Exist(h,i) > 0_ip) then
    Minimum = Weight(h,i) + Distance(h) ! Temporarily label the forward star nodes

    if (Minimum < Distance(i)) then ! If the new path distance is less than the previous iteration's
      Distance(i) = Minimum ! Set the new distance label
      Predecessor(i) = h ! Set the predecessor so that the shortest route can be traced back to the source

      ! Ensure that there are no cases where two nodes are the predecessors of each other
      if (Predecessor(h) /= i) then
        Predecessor(i) = h ! Set the predecessor so that the shortest route can be traced back to the source
        Distance(i) = Weight(h,i) + Distance(h) ! Set the new distance label
      else
        write(*,*)
        write(*,*)'Predecessor error'
        write(*,*)trim(adjustl(Name(h))), ' ', trim(adjustl(Name(i)))
        write(*,*)'Distance(h) = ', Distance(h), ' Weight = ', Weight(h,i), ' Distance(i) = ',
          Distance(i)
        write(*,*)
      end if

      ! If the shortest distance to a node can be lowered, unlabel the node and let it be chosen from the
      ! others if it is the lowest (likely)
      NodeLabel(i) = 1
      err = 0
    end if
  end if
end do

! Terminate only once all nodes have been permanently labelled
if (err == 1) then
  do i=1,Order

    if (NodeLabel(i) == 1) then
      err = 0
    end if
  end do
end if

! Find the temporarily labelled node with the shortest distance to the source
if (err == 0) then
  Minimum = real(Large)
  do i=0,Order
    if (NodeLabel(i) == 1 .and. Distance(i) < Minimum) then
      Minimum = Distance(i) ! Find the minimum temporarily labelled node for permanent labelling
      j = i
    end if
  end do
end if

end do

! Remove unnecessary information from the memory
deallocate(NodeLabel)

```



```

    deallocate(NodeLabelCount)

    return
end subroutine FordR

!!*****!!
!! Ford's Algorithm Integer !!
!!*****!!

subroutine FordI(Start,Order,Large,Weight,Exist,Distance,Predecessor)
!! This subroutine uses Ford's Algorithm to find the shortest distances between the source and all other nodes in the
graph
!! It accepts is an integer (Start) which indicates which node is that start node for the shortest distance algorithm
!! An integer (Order) indicating the order of the graph (the number of nodes in the graph, excluding the source.
There is no sink)
!! An integer (Large) indicating which large number was used to indicate missing arcs in Cost
!! An integer array (Weight) indicating the weight on each arc, Large is used to indicate missing arcs
!! An integer array (Exist) indicating the upper limit of an arc i.e. 0 if it does not exist > 0 if it does exist
!! It returns an integer array (Distance) indicating the shortest distance to each node from the source
!! An integer array (Predecessor) indicating which node is the predecessor for the shortest distance from the source
implicit none
integer(ip), intent(in) :: Order, Start
integer(ip), intent(in) :: Large
integer(ip), dimension(0:), intent(out) :: Distance
integer(ip), dimension(0:), intent(out) :: Predecessor
integer(ip), dimension(0:,0:), intent(in) :: Exist
integer(ip), dimension(0:,0:), intent(in) :: Weight
integer(ip) :: Minimum ! Used to find the minimum from a group of options
integer(ip) :: h, i, j, err
integer(ip), allocatable, dimension(:) :: NodeLabelCount ! Count how many times a node has been labelled,
if it equals Order, a negative cycle exists
integer, allocatable, dimension(:) :: NodeLabel ! Check which labels have been permanently labelled

! Make the size of the array containing node labels equal to the number of nodes in the network
allocate(NodeLabel(0:Order))
allocate(NodeLabelCount(0:Order))

Distance = Large ! Set the distance labels equal to a large number
Distance(Start) = 0 ! Set the distance label at the source equal to zero
Predecessor = -1 ! Set the predecessor labels equal to an impossible number to prevent accidental predecessor
allocation
Minimum = Large ! So that finding a minimum is a minimum of the network, not accidentally set by the
compiler
NodeLabel = 0 ! All nodes have yet been labelled
NodeLabelCount = 0 ! All nodes have yet been labelled

j = Start ! Set the starting point equal to the node which the main program sets
err = 0 ! Allows the conditional do loop to be entered into

! Continue the do loop until all nodes have been permanently labelled
do while (err < 1)

    h = j
    Minimum = Large ! Set the min. equal to a high number so that the min. distance can be found
    if (NodeLabel(h) == 2) then
        write(*,*)'Node ', trim(adjustl(Name(h))), ' is already permanently labelled'
        read(*,*)
    end if
    NodeLabel(h) = 2 ! Permanently label the node

! Check if any negative cycles exist before they make the program run continuously, could also mean that not
all nodes have been connected properly and thus will not get labelled permanently
NodeLabelCount(h) = NodeLabelCount(h) + 1
if (NodeLabelCount(h) == Order) then
    write(*,*)'There is a negative cycle.'
    return ! Return to the main program as there is a negative cycle
end if

! Find the shortest path to each of the nodes
do i=0,Order

```

```

if (Exist(h,i) > 0_ip) then
  Minimum = Weight(h,i) + Distance(h) ! Temporarily label the forward star nodes

  if (Minimum < Distance(i)) then ! If the new path distance is less than the previous iteration's
    Distance(i) = Minimum ! Set the new distance label
    Predecessor(i) = h ! Set the predecessor so that the shortest route can be traced back to the source

    ! Ensure that there are no cases where two nodes are the predecessors of each other
    if (Predecessor(h) /= i) then
      Predecessor(i) = h ! Set the predecessor so that the shortest route can be traced back to the source
      Distance(i) = Weight(h,i) + Distance(h) ! Set the new distance label
    else
      write(*,*)
      write(*,*)'Predecessor error'
      write(*,*)trim(adjustl(Name(h))), ' ', trim(adjustl(Name(i)))
      write(*,*)'Distance(h) = ', Distance(h), ' Weight = ', Weight(h,i), ' Distance(i) = ',
        Distance(i)
      write(*,*)
    end if
  end if

  ! If the shortest distance to a node can be lowered, unlabel the node and let it be chosen from the
  others if it is the lowest (likely)
  NodeLabel(i) = 1
  err = 0
end if
end if
end do

! Terminate only once all nodes have been permanently labelled
if (err == 1) then
  do i=0,Order

    if (NodeLabel(i) == 1) then
      err = 0
    end if
  end do
end if

! Find the temporarily labelled node with the shortest distance to the source
if (err == 0) then
  Minimum = real(Large)
  do i=0,Order
    if (NodeLabel(i) == 1 .and. Distance(i) < Minimum) then
      Minimum = Distance(i) ! Find the minimum temporarily labelled node for permanent labelling
      j = i
    end if
  end do
end if

end do

return
end subroutine FordI

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Successive Shortest Path Algorithm !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

subroutine SuccessiveShortestPath(Order, Large, Exist, Cost, Lower, Upper, Supply, Flow, Name)
! This subroutine uses the Successive Shortest Path Algorithm to find the minimum cost of transferring all from
the source to the sinks.
! It accepts is an integer (Order) indicating the order of the graph (the number of nodes in the graph, excluding the
source. There is no sink.
! An integer (Large) indicating which large number was used in the main program
! An integer (Exist) indicating which arcs exist, if Exist = 1, then the arc exists, if 0, then not
! A real array (Cost) indicating the cost of moving 1 unit along each arc, Large is used to indicate missing arcs
! An integer array (Lower) indicating the lower limit of an arc i.e. 0 or Large if it does not exist > 0 if it does exist
! An integer array (Upper) indicating the upper limit of an arc i.e. 0 if it does not exist > 0 if it does exist
! An integer array (Supply) indicating how much each node supplies (if positive) or requires (if negative), transfer
arc if 0
! A character array (Name) containing the names of each of the nodes

```

```

!! The subroutine returns the integer array (Flow) indicating how many units flow through each node
implicit none
integer(ip), intent(in) :: Order ! The order of the graph (number of nodes)
integer(ip), intent(in) :: Large
real, dimension(0:,0:), intent(in) :: Cost ! Cost of transporting a pallet to the next point in the graph
integer(ip), dimension(0:,0:), intent(in) :: Exist ! Existence of arcs
integer(ip), dimension(0:), intent(in) :: Supply ! The supply at a node
integer(ip), dimension(:,), intent(in) :: Lower ! Minimum flow allowed along the arc
integer(ip), dimension(0:,0:), intent(in) :: Upper ! Maximum flow allowed along the arc
integer(ip), dimension(0:,0:), intent(out) :: Flow ! The flow through the network
character(*), dimension(0:), intent(in) :: Name ! The names of the nodes
integer(ip) :: g, h, i, j, k, a
integer(ip) :: Minimum
real, allocatable, dimension(:) :: d ! Shortest distance from source to node j
real, allocatable, dimension(:) :: Potential ! The shortest path to a node
real, allocatable, dimension(:,) :: Cost2 ! A copy of the cost, except that reverse (negative) costs are
    included
real, allocatable, dimension(:,) :: ReducedCost ! The reduced cost of the flow through an arc
integer(ip), allocatable, dimension(:) :: pred ! Predecessor node for shortest path
integer(ip), allocatable, dimension(:) :: Imbalance ! The supply/need at each node
integer(ip), allocatable, dimension(:,) :: ResidualCap ! The upper limit in the residual network

! Allocate memory to the various arrays required in this subroutine
allocate(d(0:Order))
allocate(pred(0:Order))
allocate(Potential(Order))
allocate(Imbalance(0:Order))
allocate(Cost2(0:Order,0:Order))
allocate(ResidualCap(0:Order,0:Order))
allocate(ReducedCost(0:Order,0:Order))

! Set up the cost structure used to calculate the residual costs later
Cost2 = real(Large) ! Set the cost matrix equal to a large number, so that non-existent arcs will have a very
    large cost
do i=0,Order
    do j=0,Order
        if (Cost(i,j) < Large - 1000) then ! Subtract 1000 from the large number to allow for possible error
            Cost2(i,j) = Cost(i,j)
            Cost2(j,i) = -1.*Cost(i,j)
        end if
    end do
end do

! Create the residual networks
Potential = 0.
Imbalance = Supply ! The imbalance must initially equal the supply
ResidualCap = Upper ! Initially the residual upper limit is equal to the real upper limit
ReducedCost = Large ! Set all elements in the ReducedCost array equal to a large number

! Find the shortest distances to all nodes to be used for the potentials
call FordR(0_ip,Order,Large,Cost,ResidualCap,d,pred)

! Remove any possible lower bounds
Minimum = SUM(Lower)
if (Minimum /= 0) then ! If any lower bounds exist
    do i=1,Order
        do j=1,Order
            if (Exist(i,j) == 1_ip) then ! If the arc exists...
                if (Lower(i,j) > 0_ip) then ! ...and has a lower bound
                    Imbalance(i) = Imbalance(i) - Lower(i,j) ! Adjust the node imbalances...
                    Imbalance(j) = Imbalance(j) + Lower(i,j)
                    ResidualCap(i,j) = ResidualCap(i,j) - Lower(i,j) ! ...and arc residual capacities
                end if
            end if
        end do
    end do
end if

! Successive shortest path algorithm
h = 1_ip

```

```

do g=1,Order ! Start at each node
  do while (Imbalance(g) > 0_ip) ! If the imbalance is positive

    ! If the imbalance of the node is non-negative, move to the next
    if (Imbalance(h) >= 0_ip) h = h + 1_ip

    ! If there are no more nodes with negative imbalances, not all the supply can be removed
    if (h > Order) then
      write(*,'(1x,A)')'The program was not able to remove all of the product from the source.'

      ! Find the remaining supply
      Minimum = 0_ip
      do a=1,Order
        Minimum = Minimum + Imbalance(a)
      end do
      write(*,*)'Remaining fruit = ', Minimum, ' pallets'
      Imbalance = 0_ip
      return
    end if

    ! Augment flow while there is a source and sink
    do while (Imbalance(g) > 0 .and. Imbalance(h) < 0_ip)

      ! Update the reduced costs with the new potentials
      do i=1,Order
        do j=1,Order

          ! If the link exists, calculate it's reduced cost
          if (ResidualCap(i,j) > 0_ip) then
            ReducedCost(i,j) = Cost2(i,j) - Potential(i) + Potential(j)

            ! Make the reduced cost high on non-existent arcs to prevent accidental inclusion
            else if (ResidualCap(i,j) == 0_ip) then
              ReducedCost(i,j) = real(Large)

            ! There should not be a negative reduced cost, but this warns the user if there is.
            else
              write(*,'(A,A,A,A,A)')'There is a negative reduced cost at node (' , trim(adjustl(Name(i))),
                ',', trim(adjustl(Name(j))), ', )'
            end if
          end do ! i
        end do ! j

        ! Update the potentials at all nodes
        call FordR(g,Order,Large,ReducedCost,ResidualCap,d,pred)

        ! Update the potentials
        do i=1,Order
          if (d(i) < Large - 1000) then ! Subtraction of 1000 allows for any change that may occur
            Potential(i) = Potential(i) - d(i)
          else
            Potential(i) = 0_ip
          end if
        end do ! i

        ! Find the shortest distance to the node with a negative imbalance
        call FordR(g,Order,Large,ReducedCost,ResidualCap,d,pred)

        ! Find the largest amount of flow that can be augmented
        Minimum = -1_ip*Imbalance(h) ! Find the minimum augmentable flow at the sink
        j = h
        k = h

        ! Find the minimum along the source-sink path
        do while (j /= g)
          j = pred(k)
          if (ResidualCap(j,k) < Minimum) then
            Minimum = ResidualCap(j,k)
          end if
        end do

        ! If there should be a 0 possibility, there must be a coding mistake

```

```

        if (Minimum == 0_ip) then
            write(*,*)'Problem: Minimum = 0'
            write(*,')(1x,A,11,A,11,A,12)''ResidualCap(', j, ', ', k, ') = ', ResidualCap(j,k)
            write(*,*)'Please contact the program developer.'
            write(*,*)'Program will now be stopped...'
            read(*,*)
            stop
        end if

        end if ! ResidualCap(j,k) < Minimum
        k = j
    end do ! while (j /= g)

    ! Check if the imbalance at the source is smallest augmentable flow
    if (Imbalance(g) < Minimum) Minimum = Imbalance(g)

    ! Augment the flow
    Imbalance(h) = Imbalance(h) + Minimum
    j = g + 1_ip
    k = h ! Start at the sink
    do while (j /= g)
        j = pred(k) ! Move along the path of augmentation
        Flow(j,k) = Flow(j,k) + Minimum
        ResidualCap(j,k) = ResidualCap(j,k) - Minimum
        ResidualCap(k,j) = ResidualCap(k,j) + Minimum
        if (j == g) then ! Augment the flow at the source
            Imbalance(j) = Imbalance(j) - Minimum
        end if
        k = j
    end do ! while (j /= g)

    end do ! while (Imbalance(g) > 0 .and. Imbalance(h) < 0_ip)
    end do ! while (Imbalance(g) > 0_ip)
end do ! g=1,Order

! Find the maximum flow from the final residual capacities
do i=1,Order
    do j=1,Order
        if (Cost(i,j) < Large) then
            Flow(i,j) = Upper(i,j) - ResidualCap(i,j)
        else
            Flow(i,j) = 0_ip
        end if
    end do
end do

! Clear memory of information no longer required
deallocate(d)
deallocate(pred)
deallocate(Cost2)
deallocate(Potential)
deallocate(Imbalance)
deallocate(ResidualCap)
deallocate(ReducedCost)

return
end subroutine SuccessiveShortestPath

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Shortest Augmenting Path Algorithm !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

subroutine ShortAugPath(Order, Large, Exist, Upper, Supply, Flow, Name)
!! Subroutine ShortAugPath finds the maximum flow allowable though a network using the Shortest Augmenting Path
Algorithm
!! It accepts an integer (Order) indicating the order of the network i.e. the number of nodes
!! An integer (Large) indicating which large number was used in place of infinity in the main program for consistency
!! An integer array (Exist) indicating which arcs exist
!! An integer array (Upper) indicating the upper limits of the flow on each arc of the network
!! An integer array (Supply) indicating which nodes are sources (Supply > 0) and which nodes are sinks (Supply < 0)

```

```

!! A character array (Name) containing the names of each of the nodes
!! The subroutine returns an integer array (Flow) indicating how much flows through each arc of the network
implicit none
integer(ip), intent(in) :: Order
integer(ip), intent(in) :: Large
integer(ip), dimension(0:), intent(in) :: Supply
integer(ip), dimension(0:,0:), intent(in) :: Exist
integer(ip), dimension(0:,0:), intent(in) :: Upper
integer(ip), dimension(0:,0:), intent(out) :: Flow
character(*), dimension(0:), intent(in) :: Name
integer(ip) :: i, j, k, err
integer(ip) :: Minimum
integer(ip), allocatable, dimension(:) :: Distance ! Distance label from sink to other nodes along
    admissible paths
integer(ip), allocatable, dimension(:) :: Predecessor ! The predecessor of a node along the shortest path
integer(ip), allocatable, dimension(:,:) :: RevConMatrix ! A matrix indicating reverse connectivity
integer(ip), allocatable, dimension(:,:) :: ResidualCap ! The residual capacity of arcs
character(50), allocatable, dimension(:) :: Name1 ! The names of nodes

! Allocate memory space to the variables
allocate(Name1(0:Order+1))
allocate(Distance(0:Order+1))
allocate(Predecessor(0:Order+1))
allocate(ResidualCap(0:Order+1,0:Order+1))
allocate(RevConMatrix(0:Order+1,0:Order+1))

! Set up the new name array
do i=0,Order
    Name1(i) = trim(adjustl(Name(i)))
end do
Name1(Order+1) = 'Sink'

! Set up the arrays representing residual capacity and reverse connectivity
ResidualCap = 0_ip
RevConMatrix = 0_ip
do i=1,Order
    do j=1,Order
        ResidualCap(i,j) = Upper(i,j)
        if (Exist(i,j) == 1) then
            RevConMatrix(j,i) = 1_ip
        end if
    end do
end do

do i=1,order
    ! Set up the super source and super sink
    if (Supply(i) > 0_ip) then
        ResidualCap(0,i) = Large
        RevConMatrix(i,0) = 1_ip
    else if (Supply(i) < 0) then
        ResidualCap(i,Order+1) = Large
        RevConMatrix(Order+1,i) = 1_ip
    end if
end do

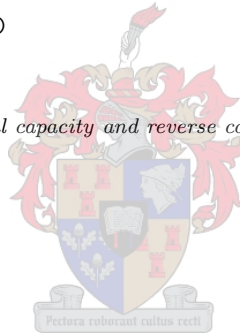
! Find the shortest distance from super sink to all other nodes (units = no. of arcs)
call FordI(Order+1,Order+1,Large,RevConMatrix,RevConMatrix,Distance,Predecessor)

Predecessor = -1_ip
do while (Distance(0) < Order+5_ip) ! Continue until there are no more paths that can be followed
    k = 0_ip
    err = 0_ip
    i = 0_ip

! Find the path along which flow can be augmented
do while (err == 0_ip)
    k = k + 1_ip

! Find the first admissible arc i.e.  $d(i) = d(j) + 1$ 
    j = k

```




```

if (ResidualCap(i,j) > 0_ip .and. Distance(i) == Distance(j) + 1_ip) then
  k = 0
  Predecessor(j) = i ! Set the predecessor of the next node equal to the current node to aid in finding
    the path back to the source
  i = j

  ! If a path has been found to the sink node, find the maximum that can be augmented
  if (j == Order+1_ip) then
    Minimum = Large

    ! Move along the path until the source is reached
    do while (i /= 0_ip)
      i = Predecessor(j)

      ! Find the maximum that can be augmented
      if (ResidualCap(i,j) < Minimum) Minimum = ResidualCap(i,j)
      j = i ! Move to the next node
    end do

    j = Order + 1_ip ! Start the augmentation from the sink
    i = 1_ip
    ! Once the minimum has been found, augment this amount along the valid path until the source is
      reached
    do while (i /= 0_ip)
      i = Predecessor(j)

      ! The change in flow is shown by a change in the residual value
      ResidualCap(i,j) = ResidualCap(i,j) - Minimum
      ResidualCap(j,i) = ResidualCap(j,i) + Minimum
      if (i /= Order+1_ip .and. j /= Order+1_ip) Flow(i,j) = Flow(i,j) + Minimum
      j = i ! Move to the next node
    end do

    ! Once the path to the end has been found, start from scratch with the changed residual values
    err = 1_ip
  end if ! j == Order+1_ip

end if ! ResidualCap(i,j) > 0_ip .and. Distance(i) == Distance(j) + 1_ip

! If all possible nodes are not admissible ( $d(i) \neq d(j) + 1$ ), then increase the distance to the current node
  and start from scratch
if (k == Order + 1_ip .and. err == 0_ip) then
  Distance(i) = Distance(i) + 1_ip
  i = Predecessor(i) ! Move back to the previous node and look for another node that may be considered
  err = 1_ip
  k = 0_ip
end if
end do
end do

! Release memory that is no longer required
deallocate(Distance)
deallocate(ResidualCap)
deallocate(Predecessor)
deallocate(RevConMatrix)

return
end subroutine ShortAugPath

end module Tools

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Main Program !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

program fruit
  use Tools ! Tell the program to use the subroutines in module Tools
  implicit none

```

```

real :: seconds ! Number of seconds that the program ran
real :: initial1, initial2, final1, final2 ! Numbers used to calculate how long the program runs
real :: ArbSupply, ArbCost, ArbLower ! Values read in by the program
integer :: err ! Integer used to check validity of files
integer :: MaxMin, FullCond
integer(ip) :: h, i, j, k
integer(ip) :: Order ! The order of the graph (number of nodes)
integer :: count_rate ! Required to find the time it takes for something to complete
integer :: hours, minutes ! The number of hours and minutes that the program ran
integer(ip) :: Minimum
character(3) :: Check
character(50) :: DataFile ! Name of the input file
character(50) :: ArbUpper ! The upper limit on an arc
character(50) :: NewName, OldName
integer(ip), parameter :: Large = 1000000000
character(1), parameter :: Dot = '.' ! The character .
character(4), parameter :: DotMin = '.min' ! The extension .min for the minimum cost analysis
character(4), parameter :: DotMax = '.max' ! The extension .max for the maximum flow analysis
real, allocatable, dimension(:, :) :: Cost ! Cost of transporting a pallet to the next point in the graph
integer(ip), allocatable, dimension(:) :: Supply ! The supply at a node
integer(ip), allocatable, dimension(:, :) :: Flow ! The flow through the network
integer(ip), allocatable, dimension(:, :) :: Exist ! A matrix of existing arcs
integer(ip), allocatable, dimension(:, :) :: Lower ! Minimum flow allowed along the arc
integer(ip), allocatable, dimension(:, :) :: Upper ! Maximum flow allowed along the arc
character(50), allocatable, dimension(:) :: Name ! A record of the names of the nodes

write(*, '(A)') 'Copyright 2005 University of Stellenbosch'
write(*, '(A)') 'This program may be distributed freely amongst all members'
write(*, '(A)') 'of the South African fruit industry for research purposes.'
write(*, *)
write(*, '(A)') 'Press Ctrl + C at any point during the program to quit.'

Check = 'y'
do while (Check == 'y') ! Allows the user to run consecutive calculations without restarting the program
  err = 1

  ! Find which calculations to perform
  do while (err /= 0)
    write(*, *)
    write(*, '(A)') 'Do you wish to perform (1) a minimum cost analysis, or'
    write(*, '(23x,A)', advance='no') '(2) a maximum flow analysis? '
    read(*, *, iostat=err) MaxMin
    if (err /= 0 .or. (MaxMin /= 1 .and. MaxMin /= 2)) write(*, *) 'Your input was incorrect, please enter'
    '1" or "2".'
  end do

  ! Enquire the detail of the screen report
  err = 1
  do while (err /= 0)
    write(*, *)
    write(*, '(A)', advance='no') 'Do you wish to (1) have a full, (2) condensed or (3) no screen report? '
    read(*, *, iostat=err) FullCond
    if (err /= 0 .or. (FullCond /= 1 .and. FullCond /= 2 .and. FullCond /= 3)) then
      write(*, *) 'Your input was incorrect, please enter "1", "2" or "3".'
      err = 1
    end if
  end do

  ! Ask for the name of the file containing the network information
  err = 1
  do while (err /= 0)
    write(*, *)
    write(*, '(A)', advance='no') 'Please enter the name of the file containing the costs: '
    read(*, *) DataFile
    write(*, *)

    ! Opens the file and makes sure the file exists, stops the program if not
    open(unit=12, file=trim(adjustl(DataFile)), status='old', action='read', iostat=err)
    if (err == 134) then
      write(*, '(1x,A,A,A)') 'Problem opening ', trim(adjustl(DataFile)), ', it may be open in another program.'
    end if
  end do
end do

```

```

write(*,*)
write(*,'(A)',advance='no')'Do you wish to try again (Y/N)? '
read(*,*)Check
if (trim(adjustl(Check)) == 'N' .or. trim(adjustl(Check)) == 'n' .or. trim(adjustl(Check)) ==
'no' .or. trim(adjustl(Check)) == 'NO' .or. trim(adjustl(Check)) == 'No') then
stop
end if
else if (err == 128) then
write(*,'(Ix,A,A,A)')'Cannot find ', trim(adjustl(DataFile)), ', file does not exist.'
write(*,*)
write(*,'(A)',advance='no')'Do you wish to try again (Y/N)? '
read(*,*)Check
if (trim(adjustl(Check)) == 'N' .or. trim(adjustl(Check)) == 'n' .or. trim(adjustl(Check)) ==
'no' .or. trim(adjustl(Check)) == 'NO' .or. trim(adjustl(Check)) == 'No') then
stop
end if
else if (err /= 0) then
write(*,'(Ix,A,A,A)')'Problem opening ', trim(adjustl(DataFile)), ', please make certain it exists.'
write(*,*)'Error = ', err
write(*,*)
write(*,'(A)',advance='no')'Do you wish to try again (Y/N)? '
read(*,*)Check
if (trim(adjustl(Check)) == 'N' .or. trim(adjustl(Check)) == 'n' .or. trim(adjustl(Check)) ==
'no' .or. trim(adjustl(Check)) == 'NO' .or. trim(adjustl(Check)) == 'No') then
stop
end if
end if
end do

! Set time when program starts calculations
call system_clock(hours,count_rate)
initial2 = real(hours)/real(count_rate)

! Create a file that keeps track of the names of the nodes and allocates them a number
err = 0
open(unit=14,file='NodeLabel.tmp',status='replace',action='readwrite',iostat=err)
if (err /= 0) then
write(*,'(Ix,A)')'Problem opening the temporary file. Program will be stopped.'
write(*,*)'err = ', err
stop
end if

! Write the names and index numbers of all the nodes to the temporary file
Order = 0_ip
err = 0
read(12,*)
do while (err == 0)
read(12,*,iostat=err)NewName

! If there is a change in name, add the new name to the list of names
if (trim(adjustl(OldName)) /= trim(adjustl(NewName))) then
Order = Order + 1_ip
OldName = trim(adjustl(NewName))
if (Order < 10_ip) write(14,'(A,Ix,I1)')trim(adjustl(NewName)), Order
if (Order < 100_ip .and. Order > 9_ip) write(14,'(A,Ix,I2)')trim(adjustl(NewName)), Order
if (Order < 1000_ip .and. Order > 99_ip) write(14,'(A,Ix,I3)')trim(adjustl(NewName)), Order
if (Order < 10000_ip .and. Order > 999_ip) write(14,'(A,Ix,I4)')trim(adjustl(NewName)), Order
if (Order < 100000_ip .and. Order > 9999_ip) write(14,'(A,Ix,I5)')trim(adjustl(NewName)),
Order
if (Order < 1000000_ip .and. Order > 99999_ip) write(14,'(A,Ix,I6)')trim(adjustl(NewName)),
Order
if (Order < 10000000_ip .and. Order > 999999_ip) write(14,'(A,Ix,I7)')trim(adjustl(NewName)),
Order
if (Order < 100000000_ip .and. Order > 9999999_ip) write(14,'(A,Ix,I8)')trim(adjustl(NewName)),
Order
if (Order < 1000000000_ip .and. Order > 99999999_ip)
write(14,'(A,Ix,I9)')trim(adjustl(NewName)), Order
end if
end do

```

```

! Allocate the sizes of the arrays
allocate(Name(0:Order))
allocate(Supply(0:Order))
allocate(Cost(0:Order,0:Order))
allocate(Flow(0:Order,0:Order))
allocate(Exist(0:Order,0:Order))
allocate(Upper(0:Order,0:Order))
allocate(Lower(Order,Order))

Flow = 0_ip
Cost = real(Large)
Exist = 0_ip
Lower = 0_ip
Upper = 0_ip
Supply = 0_ip
Name(0) = 'Source'

! Set up the array containing the names of the nodes
i = 0_ip
err = 0
rewind(12)
read(12,*)
do while (err == 0)
  read(12,*,iostat=err)NewName
  if (trim(adjustl(NewName)) /= trim(adjustl(OldName)) .and. err == 0) then
    i = i + 1_ip
    Name(i) = trim(adjustl(NewName))

    ! Check if a name already exists, if so, stop program
    if (i > 1_ip) then
      do j=1,i-1
        if (trim(adjustl(Name(i))) == trim(adjustl(Name(j)))) then
          write(*,'(A,A,I7,A)')trim(adjustl(Name(i))), ' has already appeared in position ', j, '.'
          write(*,'(A)')'The program will be stopped to allow you to make the necessary changes.'
          read(*,*)
          stop
        end if
      end do
      OldName = trim(adjustl(NewName))
    end if
  end do

! Read the input from the data file and store it in the arrays
i = 0_ip
err = 0
rewind(12)
read(12,*)
do while (err == 0)
  i = i + 1_ip
  h = 0_ip
  do while (h == 0_ip)
    read(12,*,iostat=err)Name(i), NewName, ArbSupply, ArbCost, ArbLower, ArbUpper

    if (err == 0) then
      ! Search for the destination vertex name and get its index value so that the arrays store the correct data
      if (trim(adjustl(NewName)) == 'END') then
        Supply(i) = nint(ArbSupply,ip)
        Cost(i,i) = real(Large)
        h = 1_ip
      else ! If it is not a node without arcs adjacent from it
        j = 0_ip
        k = 0_ip
        rewind(14)

        ! Read the list either until the node is found (k = 0) or the end of the file is reached (k > 0)
        do while (k == 0_ip)
          read(14,*,iostat=k)OldName, j

          ! Warn the user if the file they have entered contains a node that has no flow out of it, except the

```

```

sink nodes
if (k /= 0_ip) then
write(*,'(A,Ix,A,Ix,A)')'Cannot find', trim(adjustl(NewName)), 'in the list of nodes.'
write(*,*)
write(*,'(A,A)',advance='no')'Do you wish to continue (Y), or do you want to stop the
program (N)? '
read(*,*)Check
if (trim(adjustl(Check)) == 'N' .or. trim(adjustl(Check)) == 'n' .or.
trim(adjustl(Check)) == 'no' .or. trim(adjustl(Check)) == 'NO' .or.
trim(adjustl(Check)) == 'No') then
stop
end if
end if ! if (k /= 0_ip) then

! If the name is found, assign the values of the temporary variables to the arrays
if (trim(adjustl(NewName)) == trim(adjustl(OldName))) then
if (Exist(i,j) == 1_ip) then
write(*,'(A,A,A,A,A)')'The arc between ', trim(adjustl(Name(i))), ' and ',
trim(adjustl(Name(j))), ' already exists.'
write(*,'(A)')'The program will continue and the previous information will be overwritten.'
write(*,*)
end if
Exist(i,j) = 1_ip
Cost(i,j) = ArbCost
Supply(i) = nint(ArbSupply,ip)
Lower(i,j) = nint(ArbLower,ip)

! If there is no upper limit on an arc
if (ArbUpper == 'inf' .or. ArbUpper == 'Inf' .or. ArbUpper == 'INF') then
Upper(i,j) = Large
else ! If there is an upper limit

! Open a file to write the string number and read it as a number
open(unit=18,file='TempUpper.tmp',status='replace',action='readwrite',iostat=err)
if (err == 0) then
write(18,*)trim(adjustl(ArbUpper))
backspace(18)
read(18,*)Upper(i,j)
else
write(*,*)'There is a problem writing a file, the program cannot continue and will '
'be stopped.'
stop
end if ! if (err == 0) then
close(18,status='delete')
end if ! if (ArbUpper == 'inf' .or. ArbUpper == 'Inf' .or. ArbUpper == 'INF') then

k = 1_ip
end if ! trim(adjustl(NewName)) == trim(adjustl(OldName))
end do ! while (k == 0_ip)

rewind(14)
read(12,*)NewName

! If the next name is not the same as the current name, a new node is found
if (NewName /= Name(i)) h = 1_ip

backspace(12)
end if ! if (trim(adjustl(NewName)) == 'END') then

else ! if (err /= 0) then
h = 1_ip
end if ! if (err == 0) then

end do ! while (h == 0_ip)
end do ! while (err == 0)

! Check if there are any bi-directional graphs, existence would cause problems
do i=1,Order
do j=1,Order
if (Exist(i,j) == 1_ip .and. Exist(j,i) == 1_ip .and. i /= j) then

```

```

    write(*,'(Ix,A)') 'There is a bi-directional arc, please add an extra vertex.'
    write(*,'(Ix,A,A,A,A)') 'The problem occurred at the arc between ', trim(adjustl(Name(i))),
    ' and ', trim(adjustl(Name(j)))
    stop
  end if
end do
end do

! Find the order and size of the network
Minimum = 0_ip
do i=1,Order
  do j=1,Order
    Minimum = Minimum + Exist(i,j)
  end do
end do

write(*,'(A,I7)') 'Number of nodes =', Order
write(*,'(A,I7)') 'Number of arcs =', Minimum

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Start solving the minimum cost problem !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
if (MaxMin == 1) then

  ! Check if all the demand and supply match up
  Minimum = 0_ip
  do i=1,Order
    Minimum = Minimum + Supply(i)
  end do

  ! Inform the user that the supply and demand are not equal
  if (Minimum /= 0_ip) then
    write(*,*) 'The supply values do not cancel out, please make sure that they do.'
    write(*,*) 'The net supply is ', Minimum
    write(*,'(Ix,A)',advance='no') 'Do you wish to stop the program? '
    read(*,*) Check
    if (trim(adjustl(Check)) == 'Y' .or. trim(adjustl(Check)) == 'y' .or. trim(adjustl(Check)) ==
    'yes' .or. trim(adjustl(Check)) == 'YES' .or. trim(adjustl(Check)) == 'Yes') then
      stop
    end if
  end if

  ! Set time when program starts calculations
  call system_clock(hours,count_rate)
  initial1 = real(hours)/real(count_rate)

  ! Find the minimum cost
  call SuccessiveShortestPath(Order, Large, Exist, Cost, Lower, Upper, Supply, Flow, Name)

  ! Set time when program ends calculations
  call system_clock(hours,count_rate)
  final1 = real(hours)/real(count_rate)

  ! Write the output
  if (FullCond /= 3) then
    write(*,*)
    write(*,'(Ix,88A1)') char(201), (char(205),i=1,17), char(209), (char(205),i=1,17), char(209),
    (char(205),i=1,22), char(209), (char(205),i=1,19), char(209), (char(205),i=1,7), char(187)
    write(*,'(Ix,A1,Ix,A15,Ix,A1,Ix,A15,Ix,A1,Ix,A20,Ix,A1,Ix,A17,Ix,A1,Ix,A5,Ix,A1)') char(186),
    'From', char(179), 'To', char(179), 'min > flow > max', char(179), 'Cost Contribution',
    char(179), 'Limit', char(186)
    write(*,'(Ix,88A1)') char(199), (char(196),i=1,17), char(197), (char(196),i=1,17), char(197),
    (char(196),i=1,22), char(197), (char(196),i=1,19), char(197), (char(196),i=1,7), char(182)
    do i=1,Order
      do j=1,Order
        if (Exist(i,j) == 1_ip) then
          if (Flow(i,j) == Upper(i,j) .and. Upper(i,j) /= 0_ip) then
            write(*,'(Ix,A1,Ix,A15,Ix,A1,Ix,A15,Ix,A1,Ix,I6,A1,I6,A1,I6,Ix,A1,Ix,F17.2,Ix,A1,Ix,A5,Ix,
            A1)') char(186), trim(adjustl(Name(i))), char(179), trim(adjustl(Name(j))),
            char(179), Lower(i,j), char(242), Flow(i,j), char(242), Upper(i,j), char(179),

```

```

      Cost(i,j)*real(Flow(i,j)), char(179), 'Yes', char(186)
    else if (Flow(i,j) > 0 .and. FullCond >= 1 .and. Upper(i,j) /= Large) then
      write(*,'(1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,I6,A1,I6,A1,I6,1x,A1,1x,F17.2,1x,A1,1x,A5,1x,
      A1)') char(186), trim(adjustl(Name(i))), char(179), trim(adjustl(Name(j))),
      char(179), Lower(i,j), char(242), Flow(i,j), char(242), Upper(i,j), char(179),
      Cost(i,j)*real(Flow(i,j)), char(179), 'No', char(186)
    else if (Flow(i,j) == 0 .and. FullCond == 1 .and. Upper(i,j) /= Large) then
      write(*,'(1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,I6,A1,I6,A1,I6,1x,A1,1x,F17.2,1x,A1,1x,A5,1x,
      A1)') char(186), trim(adjustl(Name(i))), char(179), trim(adjustl(Name(j))),
      char(179), Lower(i,j), char(242), Flow(i,j), char(242), Upper(i,j), char(179),
      Cost(i,j)*real(Flow(i,j)), char(179), 'No', char(186)
    else if (Flow(i,j) > 0 .and. FullCond >= 1 .and. Upper(i,j) == Large) then
      write(*,'(1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,I6,A1,I6,A1,A6,1x,A1,1x,F17.2,1x,A1,1x,A5,1x,
      A1)') char(186), trim(adjustl(Name(i))), char(179), trim(adjustl(Name(j))),
      char(179), Lower(i,j), char(242), Flow(i,j), char(242), 'infty', char(179),
      Cost(i,j)*real(Flow(i,j)), char(179), 'No', char(186)
    else if (Flow(i,j) == 0 .and. FullCond == 1 .and. Upper(i,j) == Large) then
      write(*,'(1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,I6,A1,I6,A1,A6,1x,A1,1x,F17.2,1x,A1,1x,A5,1x,
      A1)') char(186), trim(adjustl(Name(i))), char(179), trim(adjustl(Name(j))),
      char(179), Lower(i,j), char(242), Flow(i,j), char(242), 'infty', char(179),
      Cost(i,j)*real(Flow(i,j)), char(179), 'No', char(186)
    end if
  end if
end do
end do
write(*,'(1x,88A1)')char(200), (char(205),i=1,17), char(207), (char(205),i=1,17), char(207),
(char(205),i=1,22), char(207), (char(205),i=1,19), char(207), (char(205),i=1,7), char(188)
end if

```

! Write the total minimum cost

ArbCost = 0.

do i=1,Order

do j=1,Order

if (Flow(i,j) > 0_ip) ArbCost = ArbCost + real(Flow(i,j))*Cost(i,j)

end do

end do

write(*,*)

if (ArbCost < 100.) write(*,'(1x,A,F5.2)')'The total minimum cost is ', ArbCost

if (ArbCost < 1000. .and. ArbCost >= 100.) write(*,'(1x,A,F6.2)')'The total minimum cost is ', ArbCost

if (ArbCost < 10000. .and. ArbCost >= 1000.) write(*,'(1x,A,F7.2)')'The total minimum cost is ', ArbCost

if (ArbCost < 100000. .and. ArbCost >= 10000.) write(*,'(1x,A,F8.2)')'The total minimum cost is ', ArbCost

if (ArbCost < 1000000. .and. ArbCost >= 100000.) write(*,'(1x,A,F9.2)')'The total minimum cost is ', ArbCost

if (ArbCost < 10000000. .and. ArbCost >= 1000000.) write(*,'(1x,A,F10.2)')'The total minimum cost is ', ArbCost

if (ArbCost < 100000000. .and. ArbCost >= 10000000.) write(*,'(1x,A,F11.2)')'The total minimum cost is ', ArbCost

if (ArbCost < 1000000000. .and. ArbCost >= 100000000.) write(*,'(1x,A,F12.2)')'The total minimum cost is ', ArbCost

if (ArbCost < 10000000000. .and. ArbCost >= 1000000000.) write(*,'(1x,A,F13.2)')'The total minimum cost is ', ArbCost

if (ArbCost < 100000000000. .and. ArbCost >= 10000000000.) write(*,'(1x,A,F14.2)')'The total minimum cost is ', ArbCost

if (ArbCost < 1000000000000. .and. ArbCost >= 100000000000.) write(*,'(1x,A,F15.2)')'The total minimum cost is ', ArbCost

if (ArbCost < 10000000000000. .and. ArbCost >= 1000000000000.) write(*,'(1x,A,F16.2)')'The total minimum cost is ', ArbCost

if (ArbCost < 100000000000000. .and. ArbCost >= 10000000000000.) write(*,'(1x,A,F17.2)')'The total minimum cost is ', ArbCost

if (ArbCost < 1000000000000000. .and. ArbCost >= 100000000000000.) write(*,'(1x,A,F18.2)')'The total minimum cost is ', ArbCost

if (ArbCost >= 10000000000000000.) write(*,'(1x,A,F24.2)')'The total minimum cost is ', ArbCost

write(*,*)

! Automatically create the output filename from the input filename

j = index(DataFile,Dot) ! Find where the extension begins


```

OldName = Datafile(:j-1) ! Create a string that has the name of the input file but without the extension
NewName = trim(OldName)//DotMin ! Add the extension
open(unit=16,file=trim(adjustl(NewName)),status='replace',action='write',iostat=err)
if (err /= 0) then
  write(*,*)
  write(*,(1x,A,A,A)')'There is a problem creating ', trim(adjustl(NewName)), ' perhaps there is
  insufficient disk space.'
  write(*,*)
  stop
end if

write(*,(1x,A,A)')'You can find the relevant report in a file called ', trim(adjustl(NewName))
write(*,*)

write(16,'(A88)')' _____'
write(16,'(A2,A15,A3,A15,A3,A20,A3,A18,A3,A5,A2)')'| ', 'From', '| ', 'To', '| ', 'min > flow > max',
'| ', 'Cost contribution', '| ', 'Limit', '| '
write(16,'(A88)')'| _____| _____| _____| _____|
do i=1,Order
  do j=1,Order
    if (Exist(i,j) > 0) then
      if (Flow(i,j) == Upper(i,j) .and. Upper(i,j) /= 0_ip) then
        write(16,'(A2,A15,A3,A15,A3,I6,A1,I6,A1,I6,A3,F17.2,A3,A5,A2)')'| ', trim(adjustl(Name(i))),
        '| ', trim(adjustl(Name(j))), '| ', Lower(i,j), '>', Flow(i,j), '>', Upper(i,j), '| ',
        Cost(i,j)*real(Flow(i,j)), '| ', 'Yes', '| '
      else if (Flow(i,j) > 0 .and. FullCond >= 1 .and. Upper(i,j) /= Large) then
        write(16,'(A2,A15,A3,A15,A3,I6,A1,I6,A1,I6,A3,F17.2,A3,A5,A2)')'| ', trim(adjustl(Name(i))),
        '| ', trim(adjustl(Name(j))), '| ', Lower(i,j), '>', Flow(i,j), '>', Upper(i,j), '| ',
        Cost(i,j)*real(Flow(i,j)), '| ', 'No', '| '
      else if (Flow(i,j) == 0 .and. FullCond == 1 .and. Upper(i,j) /= Large) then
        write(16,'(A2,A15,A3,A15,A3,I6,A1,I6,A1,I6,A3,F17.2,A3,A5,A2)')'| ', trim(adjustl(Name(i))),
        '| ', trim(adjustl(Name(j))), '| ', Lower(i,j), '>', Flow(i,j), '>', Upper(i,j), '| ',
        Cost(i,j)*real(Flow(i,j)), '| ', 'No', '| '
      else if (Flow(i,j) > 0 .and. FullCond >= 1 .and. Upper(i,j) == Large) then
        write(16,'(A2,A15,A3,A15,A3,I6,A1,I6,A1,A6,A3,F17.2,A3,A5,A2)')'| ', trim(adjustl(Name(i))),
        '| ', trim(adjustl(Name(j))), '| ', Lower(i,j), '>', Flow(i,j), '>', 'infy', '| ',
        Cost(i,j)*real(Flow(i,j)), '| ', 'No', '| '
      else if (Flow(i,j) == 0 .and. FullCond == 1 .and. Upper(i,j) /= Large) then
        write(16,'(A2,A15,A3,A15,A3,I6,A1,I6,A1,A6,A3,F17.2,A3,A5,A2)')'| ', trim(adjustl(Name(i))),
        '| ', trim(adjustl(Name(j))), '| ', Lower(i,j), '>', Flow(i,j), '>', 'infy', '| ',
        Cost(i,j)*real(Flow(i,j)), '| ', 'No', '| '
      end if
    end if
  end do
end do
write(16,'(A88)')'| _____| _____| _____| _____|

write(16,*)
if (ArbCost < 100.) write(16,'(A,F5.2)')'The total minimum cost is R', ArbCost
if (ArbCost < 1000. .and. ArbCost >= 100.) write(16,'(A,F6.2)')'The total minimum cost is R',
ArbCost
if (ArbCost < 10000. .and. ArbCost >= 1000.) write(16,'(A,F7.2)')'The total minimum cost is R',
ArbCost
if (ArbCost < 100000. .and. ArbCost >= 10000.) write(16,'(A,F8.2)')'The total minimum cost is R',
ArbCost
if (ArbCost < 1000000. .and. ArbCost >= 100000.) write(16,'(A,F9.2)')'The total minimum cost is R',
ArbCost
if (ArbCost < 10000000. .and. ArbCost >= 1000000.) write(16,'(A,F10.2)')'The total minimum cost is
R', ArbCost
if (ArbCost < 100000000. .and. ArbCost >= 10000000.) write(16,'(A,F11.2)')'The total minimum cost
is R', ArbCost
if (ArbCost < 1000000000. .and. ArbCost >= 100000000.) write(16,'(A,F12.2)')'The total minimum
cost is R', ArbCost
if (ArbCost < 10000000000. .and. ArbCost >= 1000000000.) write(16,'(1x,A,F13.2)')'The total
minimum cost is ', ArbCost
if (ArbCost < 100000000000. .and. ArbCost >= 10000000000.) write(16,'(1x,A,F14.2)')'The total
minimum cost is ', ArbCost
if (ArbCost < 1000000000000. .and. ArbCost >= 100000000000.) write(16,'(1x,A,F15.2)')'The total
minimum cost is ', ArbCost
if (ArbCost < 10000000000000. .and. ArbCost >= 1000000000000.) write(16,'(1x,A,F16.2)')'The total

```

```

    minimum cost is ', ArbCost
if (ArbCost < 1000000000000000. .and. ArbCost >= 1000000000000000.) write(16,'(1x,A,F17.2)')'The
total minimum cost is ', ArbCost
if (ArbCost < 1000000000000000. .and. ArbCost >= 1000000000000000.) write(16,'(1x,A,F18.2)')'The
total minimum cost is ', ArbCost
if (ArbCost >= 1000000000000000.) write(*,'(1x,A,F24.2)')'The total minimum cost is ', ArbCost

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Start solving the maximum flow problem !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
else if (MaxMin == 2) then

    ! This is used to check if there are any sink nodes
    j = 0
    do i=1,Order
        if (Supply(i) < 0_ip) then
            j = j + 1
        end if
    end do

    ! If there are no sink nodes
    if (j == 0) then
        write(*,*)'Could not detect any sink nodes, please ensure that the supply of at least one node is negative.'
        read(*,*)

        ! Deallocate the arrays
        deallocate(Name)
        deallocate(Cost)
        deallocate(Flow)
        deallocate(Exist)
        deallocate(Upper)
        deallocate(Lower)
        deallocate(Supply)

        close(12)
        close(14,status='delete')
        close(16)

        stop
    end if

    ! This links the nodes with a positive supply to the source node
    j = 0
    do i=1,Order
        if (Supply(i) > 0_ip) then
            j = j + 1
            Cost(0,i) = 0.
            Exist(0,i) = 1_ip
            Upper(0,i) = Large
        end if
    end do

    ! If there are no source nodes
    if (j == 0) then
        write(*,*)'Could not detect any source nodes, please ensure that the supply of at least one node is
        positive.'
        read(*,*)
        ! Deallocate the arrays
        deallocate(Name)
        deallocate(Cost)
        deallocate(Flow)
        deallocate(Exist)
        deallocate(Upper)
        deallocate(Lower)
        deallocate(Supply)

        close(12)
        close(14,status='delete')
        close(16)

```



```

    stop
end if

! Set time when program starts calculations
call system_clock(hours,count_rate)
initial1 = real(hours)/real(count_rate)

! Find the maximum flow
call ShortAugPath(Order,Large,Exist,Upper,Supply,Flow,Name)

! Set time when program ends calculations
call system_clock(hours,count_rate)
final1 = real(hours)/real(count_rate)

! Write output to screen and file
if (FullCond /= 3) then
  write(*,*)
  write(*,'(Ix,69A1)')char(201), (char(205),i=1,17), char(209), (char(205),i=1,17), char(209),
    (char(205),i=1,21), char(209), (char(205),i=1,7), char(187)
  write(*,'(Ix,A1,Ix,A15,Ix,A1,Ix,A15,Ix,A1,Ix,A9,Ix,A1,Ix,A7,Ix,A1,Ix,A5,Ix,A1)')char(186), 'From',
    char(179), 'To', char(179), 'flow', char(242), 'maximum', char(179), 'Limit', char(186)
  write(*,'(Ix,69A1)')char(199), (char(196),i=1,17), char(197), (char(196),i=1,17), char(197),
    (char(196),i=1,21), char(197), (char(196),i=1,7), char(182)
  do i=1,Order
    do j=1,Order
      if (Exist(i,j) == 1_ip) then
        if (Flow(i,j) == Upper(i,j) .and. Upper(i,j) /= 0_ip) then
          write(*,'(Ix,A1,Ix,A15,Ix,A1,Ix,A15,Ix,A1,Ix,I9,A1,I9,Ix,A1,Ix,A5,Ix,A1)')char(186),
            trim(adjustl(Name(i))), char(179), trim(adjustl(Name(j))), char(179), Flow(i,j),
            char(242), Upper(i,j), char(179), 'Yes', char(186)
          else if (Flow(i,j) > 0 .and. FullCond >= 1 .and. Upper(i,j) /= Large) then
            write(*,'(Ix,A1,Ix,A15,Ix,A1,Ix,A15,Ix,A1,Ix,I9,A1,I9,Ix,A1,Ix,A5,Ix,A1)')char(186),
            trim(adjustl(Name(i))), char(179), trim(adjustl(Name(j))), char(179), Flow(i,j),
            char(242), Upper(i,j), char(179), 'No', char(186)
          else if (Flow(i,j) == 0 .and. FullCond == 1 .and. Upper(i,j) /= Large) then
            write(*,'(Ix,A1,Ix,A15,Ix,A1,Ix,A15,Ix,A1,Ix,I9,A1,I9,Ix,A1,Ix,A5,Ix,A1)')char(186),
            trim(adjustl(Name(i))), char(179), trim(adjustl(Name(j))), char(179), Flow(i,j),
            char(242), Upper(i,j), char(179), 'No', char(186)
          else if (Flow(i,j) > 0 .and. FullCond >= 1 .and. Upper(i,j) == Large) then
            write(*,'(Ix,A1,Ix,A15,Ix,A1,Ix,A15,Ix,A1,Ix,I9,A1,A9,Ix,A1,Ix,A5,Ix,A1)')char(186),
            trim(adjustl(Name(i))), char(179), trim(adjustl(Name(j))), char(179), Flow(i,j),
            char(242), 'infinite', char(179), 'No', char(186)
          else if (Flow(i,j) == 0 .and. FullCond == 1 .and. Upper(i,j) == Large) then
            write(*,'(Ix,A1,Ix,A15,Ix,A1,Ix,A15,Ix,A1,Ix,I9,A1,A9,Ix,A1,Ix,A5,Ix,A1)')char(186),
            trim(adjustl(Name(i))), char(179), trim(adjustl(Name(j))), char(179), Flow(i,j),
            char(242), 'infinite', char(179), 'No', char(186)
          end if
        end if
      end do
    end do
  end do
  write(*,'(Ix,70A1)')char(200), (char(205),i=1,17), char(207), (char(205),i=1,17), char(207),
    (char(205),i=1,21), char(207), (char(205),i=1,7), char(188)
end if

Minimum = 0_ip
do i=1,Order
  do j=1,Order
    if (Flow(i,j) > 0 .and. Supply(j) < 0_ip) Minimum = Minimum + Flow(i,j)
  end do
end do

write(*,*)
if (Minimum < 100_ip) write(*,'(Ix,A,I2)')'The total maximum flow is ', Minimum
if (Minimum < 1000_ip .and. Minimum >= 100_ip) write(*,'(Ix,A,I3)')'The total maximum flow is ',
  Minimum
if (Minimum < 10000_ip .and. Minimum >= 1000_ip) write(*,'(Ix,A,I4)')'The total maximum flow is ',
  Minimum
if (Minimum < 100000_ip .and. Minimum >= 10000_ip) write(*,'(Ix,A,I5)')'The total maximum flow is ',
  Minimum
if (Minimum < 1000000_ip .and. Minimum >= 100000_ip) write(*,'(Ix,A,I6)')'The total maximum flow

```

```

is ', Minimum
if (Minimum < 10000000_ip .and. Minimum >= 10000000_ip) write(*,'(1x,A,I7)') 'The total maximum flow
is ', Minimum
if (Minimum < 100000000_ip .and. Minimum >= 100000000_ip) write(*,'(1x,A,I8)') 'The total maximum
flow is ', Minimum
if (Minimum < 1000000000_ip .and. Minimum >= 1000000000_ip) write(*,'(1x,A,I9)') 'The total maximum
flow is ', Minimum
if (Minimum >= 10000000000_ip) write(*,'(1x,A,I15)') 'The total maximum flow is ', Minimum
write(*,*)

! Automatically create the output filename from the input filename
j = index(DataFile, Dot) ! Find where the extension begins
OldName = Datafile(:j-1) ! Create a string that has the name of the input file but without the extension
NewName = trim(OldName)//DotMax ! Add the extension
open(unit=16, file=trim(adjustl(NewName)), status='replace', action='write', iostat=err)
if (err /= 0) then
  write(*,*)
  write(*,'(1x,A,A,A)') 'There is a problem creating ', trim(adjustl(NewName)), ', perhaps there is
insufficient disk space.'
  write(*,*)
  stop
end if

write(*,'(1x,A,A)') 'You can find the relevant report in a file called ', trim(adjustl(NewName))
write(*,*)

write(16,'(1x,65A1)') ('_',i=1,65)
write(16,'(A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1)') '|', 'From', '|', 'To', '|',
'# of units', '|', 'Limit', '|'
write(16,'(67A1)') '|', ('_',i=1,17), '|', ('_',i=1,17), '|', ('_',i=1,21), '|', ('_',i=1,7), '|'
do i=1,Order
  do j=1,Order
    if (Exist(i,j) > 0) then
      if (Flow(i,j) == Upper(i,j) .and. Upper(i,j) /= 0_ip) then
        write(16,'(A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1)') '|',
trim(adjustl(Name(i))), '|', trim(adjustl(Name(j))), '|', Flow(i,j), '|', Upper(i,j), '|',
'Yes', '|'
      else if (Flow(i,j) > 0 .and. FullCond >= 1 .and. Upper(i,j) /= Large) then
        write(16,'(A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1)') '|',
trim(adjustl(Name(i))), '|', trim(adjustl(Name(j))), '|', Flow(i,j), '|', Upper(i,j), '|',
'No', '|'
      else if (Flow(i,j) == 0 .and. FullCond == 1 .and. Upper(i,j) /= Large) then
        write(16,'(A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1)') '|',
trim(adjustl(Name(i))), '|', trim(adjustl(Name(j))), '|', Flow(i,j), '|', Upper(i,j), '|',
'No', '|'
      else if (Flow(i,j) > 0 .and. FullCond >= 1 .and. Upper(i,j) == Large) then
        write(16,'(A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1)') '|',
trim(adjustl(Name(i))), '|', trim(adjustl(Name(j))), '|', Flow(i,j), '|', 'infinite', '|',
'No', '|'
      else if (Flow(i,j) == 0 .and. FullCond == 1 .and. Upper(i,j) == Large) then
        write(16,'(A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1,1x,A15,1x,A1)') '|',
trim(adjustl(Name(i))), '|', trim(adjustl(Name(j))), '|', Flow(i,j), '|', 'infinite', '|',
'No', '|'
      end if
    end if
  end do
end do
write(16,'(67A1)') '|', ('_',i=1,17), '|', ('_',i=1,17), '|', ('_',i=1,21), '|', ('_',i=1,7), '|'

write(16,*)
if (Minimum < 100_ip) write(16,'(A,I2)') 'The total maximum flow is ', Minimum
if (Minimum < 1000_ip .and. Minimum >= 100_ip) write(16,'(A,I3)') 'The total maximum flow is ',
Minimum
if (Minimum < 10000_ip .and. Minimum >= 1000_ip) write(16,'(A,I4)') 'The total maximum flow is ',
Minimum
if (Minimum < 100000_ip .and. Minimum >= 10000_ip) write(16,'(A,I5)') 'The total maximum flow is ',
Minimum
if (Minimum < 1000000_ip .and. Minimum >= 100000_ip) write(16,'(A,I6)') 'The total maximum flow is ',
Minimum
if (Minimum < 10000000_ip .and. Minimum >= 10000000_ip) write(16,'(A,I7)') 'The total maximum flow

```

```

    is ', Minimum
  if (Minimum < 100000000_ip .and. Minimum >= 100000000_ip) write(16,'(A,I8)')'The total maximum flow
  is ', Minimum
  if (Minimum < 1000000000_ip .and. Minimum >= 1000000000_ip) write(16,'(A,I9)')'The total maximum
  flow is ', Minimum
  if (Minimum >= 1000000000_ip) write(16,'(Ix,A,I15)')'The total maximum flow is ', Minimum

end if

! Deallocate the arrays
deallocate(Name)
deallocate(Cost)
deallocate(Flow)
deallocate(Exist)
deallocate(Upper)
deallocate(Lower)
deallocate(Supply)

close(12)
close(14)
close(16)

! Set time when program ends calculations
call system_clock(hours,count_rate)
final2 = real(hours)/real(count_rate)

! Calculate the time taken to solve the problem
hours = floor((final1 - initial1)/3600.)
minutes = floor((final1 - initial1)/60.) - 60*hours
seconds = final1 - initial1 - 3600*hours - 60*minutes
if (MaxMin == 1) then
  write(*,'(Ix,A,I3,A,I2,A,F6.3,A)')'The minimum cost flow calculation took ', hours, ' hours ', minutes,
  ' minutes and ', seconds, ' seconds'
else if (MaxMin == 2) then
  write(*,'(Ix,A,I3,A,I2,A,F6.3,A)')'The maximum flow calculation took ', hours, ' hours ', minutes,
  ' minutes and ', seconds, ' seconds'
end if

write(*,*)
hours = floor((final2 - initial2)/3600.)
minutes = floor((final2 - initial2)/60.) - 60*hours
seconds = final2 - initial2 - 3600*hours - 60*minutes
write(*,'(Ix,A,I3,A,I2,A,F6.3,A)')'The whole program took ', hours, ' hours ', minutes, ' minutes and ',
seconds, ' seconds'

write(*,*)
if (MaxMin == 1) then
  write(*,'(Ix,A,F7.4,A)')'Successive shortest path algorithm took ', 100.*(real(final1) -
  real(initial1))/(real(final2) - real(initial2)), '% of total time'
else if (MaxMin == 2) then
  write(*,'(Ix,A,F7.4,A)')'Shortest augmenting path algorithm took ', 100.*(real(final1) -
  real(initial1))/(real(final2) - real(initial2)), '% of total time'
end if

write(*,*)
write(*,'(A)',advance='no')'Do you wish to repeat (Y/N)? '
read(*,*)Check
if (trim(adjustl(Check)) == 'Y' .or. trim(adjustl(Check)) == 'Yes' .or. trim(adjustl(Check)) ==
  'yes' .or. trim(adjustl(Check)) == 'YES' .or. trim(adjustl(Check)) == 'y') then
  Check = 'y'
else
  Check = 'n'
end if
end do

stop
end program fruit

```

G.2 The Program *Convert*

The program *Convert* was developed when the linear programming models became too large to solve via standard spreadsheet solvers. The models were initially created in a spreadsheet format as it was easiest to recognise patterns emerging in the construction of the coefficient matrix for the variables. Due to the fact that the models have more variables than constraints and spreadsheets only allow 256 columns per worksheet, the variables were listed along rows and the constraints along columns. Five separate worksheets had to be used for each of the models for the three scenarios. The worksheets have the form depicted in Table G.1, and may be found on the compact disc attached to this thesis.

		Constraint 1 Name	Constraint 2 Name	...
		>=, <= or ==	>=, <= or ==	...
Variable 1 Name	Coef. in Obj. Func.	Coefficient	Coefficient	...
Variable 2 Name	Coef. in Obj. Func.	Coefficient	Coefficient	...
⋮	⋮	⋮	⋮	⋮

Table G.1: Spreadsheet format for *Convert*.

The program does not read how many input files (spreadsheets) there are — instead, the user has to enter this value manually. The files all have the same name, save for the last digit, which must range from one to the number of input files, e.g. `input1.csv`, `input2.csv`, ..., `input5.csv`, if 5 spreadsheets are required to describe the model. Each file is required to have the same first two columns (the variable names and the coefficients of the variables in the objective function), but the constraints may be different for different files. The program requires the user to enter the part of the file name common to all input files, in the example case this would be “input.” It also requires the number of input files and the number of constraints per input file. To make it easier to enter the input into the program when it is running, load files may be created. These are simply text files with each input entered on a line. For example, the load file for the summer scenario has the form

```
5
SumLP
234
135
153
250
133,
```

where 5 is the number of input files, `SumLP` is the name common to the input files and the next five values are the numbers of constraints per file. The program then asks the user whether the number of arcs used for the maximum flow or minimum cost flow should be reduced. This feature was added to *Convert* when the solutions from *PCx* were seen to be difficult to reproduce in this thesis, as it assigned a value (flow) to almost every variable (arc) of the model. However, as binary variables are required to restrict the number of arcs used for the solution and *PCx* cannot accept binary variables, the *NEOS Server* was used to solve the model, as it has solvers capable of solving mixed integer problems. However, the results found by the *XPRESS* engine used the same number of arcs, whether the arc restrictions were applied or not. Thus the function was not used further, but has remained for the case where another solver may be used.

Once the *MPS* file has been created and used to solve the model, the program is able to convert the output from either *PCx* format or the *NEOS Server* format to a clearer spreadsheet format. For this functionality, the outputs from either solver must first be converted to a *CSV* format: the text file in which the output is found has to be opened from a spreadsheet program, and columns have to be created where some consecutive spaces are found (this can be done by importing the file into a program such as *Excel* with spaces set as the delimiters of columns). The information should then be arranged in columns. The file should then be saved in *CSV* format and its name entered into *Convert*. The program code is shown below and may be found on the compact disc attached to the thesis.

```

program convert
!! Program to convert a spreadsheet model to MPS format
!! Also converts PCx or NEOS output to spreadsheet file
  implicit none
  integer :: x
  integer :: TotalVariables ! The number of variables in the LP model
  integer :: TotalConstraints ! The number of constraints in the LP model
  character(1) :: CreateMPS ! Should the MPS file be created or not?
  character(4) :: OutputFormat ! Does the user want to convert PCx or NEOS output to a friendlier format
  character(20) :: GroupName ! The common name of all files
  character(30) :: NamesFile ! The file that has the names of all variables and constraints
  character(6), parameter :: names='.names'

  ! Ask whether an MPS format file must be created
  write(*,'(A)',advance='no')'Do you want to create the MPS file (Y/N)? '
  read(*,*)CreateMPS
  write(*,*)

  if (CreateMPS == 'Y' .or. CreateMPS == 'y') then
    call Excel2MPS()

  else ! Use existing names file to change format of output file
    write(*,'(A)',advance='no')'What is the name of the files? '
    read(*,*)GroupName ! All files should have the same names until their numbers
    write(*,*)

    NamesFile = trim(adjustl(GroupName))/names
    write(*,'(A,A,A)')'The program will use the existing ', trim(adjustl(NamesFile)), ', ensure it is the correct
    one.'
    write(*,*)

    ! Open the names file
    open(unit=13,file=trim(adjustl(NamesFile)),status='old',action='read',iostat=x)
    if (x /= 0) then
      write(*,'(A,A,A)')'There is a problem opening ', trim(adjustl(NamesFile)), ', program will be stopped.'
      stop
    end if

    ! Find out how large the model is
    write(*,'(A)',advance='no')'How many variables are there? '
    read(*,*)TotalVariables
    write(*,*)
    write(*,'(A)',advance='no')'How many constraints are there? '
    read(*,*)TotalConstraints
    write(*,*)

    ! Convert the PCx output to spreadsheet format
    write(*,*)
    write(*,'(A)',advance='no')'Do you wish to convert from PCx or NEOS format, or not at all (PCx/NEOS/x)?'
    read(*,*)OutputFormat
    if (trim(adjustl(OutputFormat)) == 'p' .or. trim(adjustl(OutputFormat)) == 'P' .or.
        trim(adjustl(OutputFormat)) == 'pcx' .or. trim(adjustl(OutputFormat)) == 'PCx') then
      call PCx2CSV(13,trim(adjustl(GroupName)),TotalVariables,TotalConstraints,'Y')
    else if (trim(adjustl(OutputFormat)) == 'NEOS' .or. trim(adjustl(OutputFormat)) == 'neos' .or.
        trim(adjustl(OutputFormat)) == 'NE' .or. trim(adjustl(OutputFormat)) == 'ne') then

```



```

    write(*,'(A)')'Did you choose to restrict the number of arcs (Y/N)? '
    read(*,*)CreateMPS
    call NEOS2CSV(13,trim(adjustl(GroupName)),TotalVariables,TotalConstraints,CreateMPS)
end if

! Close the names file as it is no longer required
close(13)

end if

write(*,*)
write(*,'(A)')'The program is complete...'
read(*,*)

stop

contains

subroutine Excel2MPS()
!! This subroutine converts a spreadsheet model to the MPS format
implicit none
real :: r
real, parameter :: HugeNum = HUGE(r) ! A very large number
real, allocatable, dimension(:) :: Cost ! The coefficients of the objective function
real, allocatable, dimension(:) :: NewRHS ! The RHS of the constraints
real, allocatable, dimension(:,:) :: RHS ! Pre-formatted RHS
real, allocatable, dimension(:,:) :: NewMatrix ! The coefficients of the variables in the constraints
real, allocatable, dimension(:,,:) :: Matrix ! Pre-formatted coefficients
integer :: i, j, k, err
integer :: NumVar ! Number of variables in the model
integer :: NumInput ! Number of input files describing the model
integer :: MaxColumns ! Maximum column count in each input file
integer :: TotalColumns ! The total number of columns
integer, allocatable, dimension(:) :: NumColumns ! Number of columns in each input file
character(1) :: NumWord ! The last character of the input filename indicating the order in which they are
character(1) :: LoadYesNo ! Is there a load file?
character(1) :: WriteBinary ! Restrict flow by binary variables
character(1) :: FoundNameBefore ! Check whether a name has been found before
character(4) :: Solver ! Which solver is used to solve the problem
character(8) :: EightName ! The 8 character version of a variable or constraint name
character(20) :: InputName ! The common name of the input files
character(20) :: NewName ! A place-holder for new names
character(20) :: FileName ! A place-holder for file names
character(20) :: LoadFile ! The name of the file that can be used to circumvent manual input of parameters
character(30) :: blank, blank2 ! Place-holders
character(30) :: Name
character(1), parameter :: WriteBinaryY = 'i'
character(1), parameter :: WriteBinaryN = 'r'
character(4), parameter :: dotin = '.csv'
character(4), parameter :: dotmps = '.mps'
character(6), parameter :: dotmodel = '.model'
character(6), parameter :: dotnames = '.names'
character(1), allocatable, dimension(:) :: NewEquality ! Greater than, equal to or less than
character(8), allocatable, dimension(:) :: VarName ! Names of the variables
character(8), allocatable, dimension(:) :: NewNameRHS ! Names of the RHS
character(8), allocatable, dimension(:) :: BinaryVariableName ! Name of the binary variable
character(8), allocatable, dimension(:) :: BinaryConstraintName ! Name of the binary constraint
character(2), allocatable, dimension(:,:) :: Equality ! Pre-processed equalities
character(30), allocatable, dimension(:,:) :: NameRHS ! Pre-processed RHS names

! Find the name of the load file
write(*,'(A)',advance='no')'What is the load file name (N if none)? '
read(*,*)LoadFile
write(*,*)

! If there is no load file
if (trim(adjustl(LoadFile)) == 'N' .or. trim(adjustl(LoadFile)) == 'n') then

    LoadYesNo = 'N'

```

```

! Find the number of input files representing the model
write(*,'(A)',advance='no')'How many input files are there? '
read(*,*)NumInput
write(*,*)

! Find out the group name of the input files
write(*,'(A)',advance='no')'What is the group name of the input files? '
read(*,*)InputName
write(*,*)

else ! If there is a load file

LoadYesNo = 'Y'

! Open the load file
open(unit=44,file=trim(adjustl(LoadFile)),status='old',action='read',iostat=err)
if (err /= 0) then
write(*,'(A,A,A)')'Problem opening ', trim(adjustl(LoadFile)),', will stop program.'
stop ! Stop the program if the load file cannot be opened
end if

read(44,*)NumInput ! First line indicates the number of input files
read(44,*)InputName ! Second line indicates the group name of the input files

end if

! Find out if the binary variables should be used to restrict the number of arcs used
write(*,'(A)',advance='no')'Do you wish to restrict the number of arcs used for flow? '
read(*,*)WriteBinary

! Allocate the required amount of memory to the number of constraints per input file
allocate(NumColumns(0:NumInput))
NumColumns(0) = 0

! Open the names file
if (WriteBinary == 'Y' .or. WriteBinary == 'y') then
FileName = trim(adjustl(InputName))//WriteBinaryY//dotnames
else
FileName = trim(adjustl(InputName))//WriteBinaryN//dotnames
end if
open(unit=13,file=trim(adjustl(FileName)),status='replace',action='readwrite',iostat=err)
if (err /= 0) then
write(*,'(A,A,A)')'Problem opening ', trim(adjustl(FileName)),', will stop program.'
stop
end if

if (NumInput > 1) then

! Open each of the input files
do i=1,NumInput

! Write the input file number to a string
write(NumWord,'(I1)')i

! Add the number to the string and add the .in suffix to create the input file names
FileName = trim(adjustl(InputName))//NumWord//dotin

! Open the input files
open(unit=13+i,file=trim(adjustl(FileName)),status='old',action='read',iostat=err)
if (err /= 0) then
write(*,'(A,A,A)')'Cannot open ', trim(adjustl(FileName)),', will stop program.'
stop
end if

! If there is a load file check how many constraints there are per file
if (LoadYesNo == 'Y') then
read(44,*)NumColumns(i)
else ! Otherwise ask the user
write(*,'(A,A,A)',advance='no')'How many constraints are there in input file ',
trim(adjustl(FileName)), '? '

```

```

    read(*,*)NumColumns(i)
    write(*,*)
end if

j = 0
err = 0

! Read the first three lines which are not yet important
read(13+i,*)
read(13+i,*)
read(13+i,*)

! Create names 8 characters long from names too long to be included in the MPS file
do while (err == 0)
  read(13+i,*,iostat=err)Name
  if (err == 0) then ! If the end of the file has not been reached
    j = j + 1
    if (i == 1) then
      if (LEN(trim(adjustl(Name))) == 0) then ! Inform the user if there is a name of zero length
        write(*,*)'Name to subroutine ChangeName has zero length'
        write(*,*)'i = ', i, 'j = ', j
        read(*,*)
      end if
      ! Call the subroutine once to count the number of variables
      call ChangeName(Name,EightName,FoundNameBefore)
      if (FoundNameBefore == 'Y') then
        write(*,*)trim(adjustl(Name)), j
      end if
    end if
    if (i == 2) then
      if (LEN(trim(adjustl(Name))) == 0) then
        write(*,*)'Name to subroutine ChangeName has zero length'
        write(*,*)'i = ', i, 'j = ', j
        read(*,*)
      end if
      ! Call the subroutine a second time to write the names into an array
      call ChangeName(Name,VarName(j),FoundNameBefore)
      if (FoundNameBefore == 'Y') then
        write(*,*)'Ignored'
      end if
    end if
  end if
end do

! Check whether the numbers of variables are consistent in each input file
if (NumVar /= j .and. i > 1) then
  write(*, '(A,A,A)') 'There seem to be different variable numbers in ', FileName, ' and the one before.'
  stop
end if

NumVar = j

! Allocate the required memory for the variable names
if (i == 1) allocate(VarName(1:NumVar))
end do

else ! If there is only one input file

! Add the .in suffix to create the input file names
FileName = trim(adjustl(InputName))//dotin

! Open the input files
open(unit=14,file=trim(adjustl(FileName)),status='old',action='read',iostat=err)
if (err /= 0) then
  write(*, '(A,A,A)') 'Cannot open ', trim(adjustl(FileName)), ', will stop program.'
  stop
end if

```

```

! If there is a load file check how many constraints there are per file
if (LoadYesNo == 'Y') then
  read(44,*)NumColumns(i)
else ! Otherwise ask the user
  write(*,'(A,A,A)',advance='no')'How many constraints are there in input file ',
    trim(adjustl(FileName)), '? '
  read(*,*)NumColumns(i)
  write(*,*)
end if

j = 0
err = 0

! Read the first three lines which are not yet important
read(14,*)
read(14,*)
read(14,*)

! Create names 8 characters long from names too long to be included in the MPS file
do while (err == 0)
  read(14,*,iostat=err)Name
  if (err == 0) then ! If the end of the file has not been reached
    j = j + 1
    if (LEN(trim(adjustl(Name))) == 0) then ! Inform the user if there is a name of zero length
      write(*,*)'Name to subroutine ChangeName has zero length'
      write(*,*)'i = ', i, ' j = ', j
      read(*,*)
    end if

    ! Call the subroutine once to count the numbers of variables
    call ChangeName(Name,EightName,FoundNameBefore)
    if (FoundNameBefore == 'Y') then
      write(*,*)'C ', trim(adjustl(Name)), j
    end if
  end if
end do

NumVar = j

! Allocate the required memory for the variable names
if (i == 1) allocate(VarName(1:NumVar))

rewind(14)

! Read the first three lines which are not yet important
read(14,*)
read(14,*)
read(14,*)

do i=1,NumVar
  read(14,*,iostat=err)Name

  ! Call the subroutine a second time to write the names into an array
  call ChangeName(Name,VarName(j),FoundNameBefore)
end do

end if

TotalColumns = SUM(NumColumns)
if (WriteBinary == 'Y' .or. WriteBinary == 'y') then
  write(*,*)'There are ', 2*NumVar, ' variables and ', NumVar+TotalColumns, ' constraints in the model.'
else
  write(*,*)'There are ', NumVar, ' variables and ', TotalColumns, ' constraints in the model.'
end if

! Check if there are any variable names that are the same
do i=1,NumVar
  do j=i+1,NumVar
    if (VarName(i) == VarName(j)) then
      write(*,'(A,A,I5,I5)')'Two variables have the same name: ', VarName(i), i, j
    end if
  end do
end do

```

```

    end if
  end do
end do

! Allocate the required memory for the various arrays
MaxColumns = MAXVAL(NumColumns)
allocate(Cost(1:NumVar))
allocate(NameRHS(1:NumInput,1:MaxColumns))
allocate(RHS(1:NumInput,1:MaxColumns))
allocate(Equality(1:NumInput,1:MaxColumns))
allocate(Matrix(1:NumInput,1:NumVar,1:MaxColumns))
RHS = -1.*HugeNum

! Read the input files into memory
do i=1,NumInput
  rewind(13+i)
  read(13+i,*)blank, blank2, (NameRHS(i,j),j=1,NumColumns(i))
  read(13+i,*)blank, blank2, (RHS(i,j),j=1,NumColumns(i))
  read(13+i,*)blank, blank2, (Equality(i,j),j=1,NumColumns(i))
  do j=1,NumVar
    read(13+i,*,iostat=err)Name, Cost(j), (Matrix(i,j,k),k=1,NumColumns(i))
    if (err /= 0) then
      write(NumWord,'(I)')i
      FileName = trim(adjustl(InputName))/NumWord//dotin
      write(*,*)'Problem reading from ', trim(adjustl(FileName)), '.'
      read(*,*)
    end if
  end do
end do

! Allocate memory to new arrays
allocate(NewNameRHS(1:TotalColumns))
allocate(NewRHS(1:TotalColumns))
allocate(NewEquality(1:TotalColumns))
allocate(NewMatrix(1:NumVar,1:TotalColumns))

do i=1,NumInput
  do k=1,NumColumns(i)

    ! Ensure that there is a name for all constraints
    if (LEN(trim(adjustl(NameRHS(i,k)))) == 0) then
      write(*,*)'Name to subroutine ChangeName has zero length'
      write(*,*)'i = ', i, ' j = ', j
      read(*,*)
    end if

    ! Begin converting 2 dimensional arrays into vectors
    err = 0
    do j=1,i
      err = err + NumColumns(j-1) ! Count how many columns have already been converted
    end do

    ! Convert equality symbols to letters required by MPS format
    NewRHS(k+err) = RHS(i,k)
    if (Equality(i,k) == '==' .or. Equality(i,k) == '=') then
      NewEquality(k+err) = 'E'
    else if (Equality(i,k) == '<=' .or. Equality(i,k) == '<') then
      NewEquality(k+err) = 'L'
    else if (Equality(i,k) == '>=' .or. Equality(i,k) == '=>') then
      NewEquality(k+err) = 'G'
    else
      write(*,*)'Unknown equality in input file', i, ' and column ', k
    end if

    ! Change the names for the RHS and write them into a new vector
    call ChangeName(NameRHS(i,k),NewNameRHS(k+err),FoundNameBefore)
    if (FoundNameBefore == 'Y') then
      write(*,*)'E ', trim(adjustl(Name)), j
    end if
  end do
end do

```

```

    ! Convert 3 dimensional matrix to 2 dimensional matrix
    do j=1,NumVar
      NewMatrix(j,k+err) = Matrix(i,j,k)
    end do
  end do
end do

! Close all the input files as they are no longer required
do i=1,NumInput
  close(13+i)
end do

! Open the file into which the MPS information will be written
if (WriteBinary == 'Y' .or. WriteBinary == 'y') then
  NewName = trim(adjustl(InputName))//WriteBinaryY//dotmps
else
  NewName = trim(adjustl(InputName))//WriteBinaryN//dotmps
end if
open(unit=11,file=trim(adjustl(NewName)),status='replace',action='write',iostat=err)
if (err /= 0) then
  write(*,'(A,A,A)')'Problem opening ', trim(adjustl(NewName)), ', please try again.'
  stop
end if

! Write the name of the model from the input group name
EightName = InputName
write(11,'(A4,8x,A8)')'NAME', trim(adjustl(EightName))

! If the arc restriction is to be applied, find names for the new restrictions
if (WriteBinary == 'Y' .or. WriteBinary == 'y') then
  allocate(BinaryVariableName(1:NumVar))
  allocate(BinaryConstraintName(1:NumVar))
  do i=1,NumVar
    call BinaryName(VarName(i),BinaryVariableName(i),'v')
    call BinaryName(VarName(i),BinaryConstraintName(i),'c')
  end do
end if

! List all the names of the constraints
write(11,'(A)')'ROWS'
write(11,'(1x,A1,6x,A)')'N', 'Cost'
do k=1,TotalColumns
  write(11,'(1x,A1,2x,A)')NewEquality(k), NewNameRHS(k)
end do
if (WriteBinary == 'Y' .or. WriteBinary == 'y') then
  do i=1,NumVar
    write(11,'(1x,A1,2x,A)')'G', BinaryConstraintName(i)
  end do
end if

! Write the names of the variables, the constraints in which they appear and their coefficients in the constraints
write(11,'(A)')'COLUMNS'
do j=1,NumVar
  call WriteValue(11,VarName(j),'Cost',Cost(j),'N')
  do k=1,TotalColumns
    if (NewMatrix(j,k) >= .0000001 .or. NewMatrix(j,k) <= -.0000001) then
      call WriteValue(11,VarName(j),NewNameRHS(k),NewMatrix(j,k),'N')
    end if
  end do
end do

! Write the names of the binary variables and their constraints to the MPS file
if (WriteBinary == 'Y' .or. WriteBinary == 'y') then
  do i=1,NumVar
    call WriteValue(11,BinaryVariableName(i),'Cost',-0.01,'N')
    call WriteValue(11,BinaryVariableName(i),BinaryConstraintName(i),9999999.,'N')
  end do
end if

! Write the names of the constraints and their right hand side values

```

```

write(11,'(A)')'RHS'
do k=1,TotalColumns
  call WriteValue(11,'RHS1',NewNameRHS(k),NewRHS(k),'Y')
end do

! Write the names of the binary constraints and the RHS to the MPS file
if (WriteBinary == 'Y' .or. WriteBinary == 'y') then
  do i=1,NumVar
    call WriteValue(11,'RHS1',BinaryConstraintName(i),0.,'Y')
  end do
end if

! Write the names of the constraints and their right hand side values
if (WriteBinary == 'Y' .or. WriteBinary == 'y') then
  write(11,'(A)')'BOUNDS'
  do i=1,NumVar
    write(11,'(1x,A2,1x,A4,6x,A8)')'BV', 'BND1', BinaryVariableName(i)
  end do
end if

! Write the closing line of an MPS file
write(11,'(A)')'ENDATA'

close(11)

! Convert the output from PCx to a spreadsheet format
write(*,*)
write(*,'(A)',advance='no')'Do you wish to convert from PCx or NEOS format, or not at all (PCx/NEOS/x)?'
read(*,*)Solver
if (trim(adjustl(Solver)) == 'p' .or. trim(adjustl(Solver)) == 'P' .or. trim(adjustl(Solver)) ==
  'pcx' .or. trim(adjustl(Solver)) == 'PCx') then
  call PCx2CSV(13,trim(adjustl(InputName)),NumVar,TotalColumns,'N')
else if (trim(adjustl(Solver)) == 'NEOS' .or. trim(adjustl(Solver)) == 'neos' .or.
  trim(adjustl(Solver)) == 'N' .or. trim(adjustl(Solver)) == 'n') then
  call NEOS2CSV(13,trim(adjustl(InputName)),NumVar,TotalColumns,WriteBinary)
end if

close(13)

! Remove all unnecessary information from the memory
deallocate(Cost)
deallocate(NewRHS)
deallocate(RHS)
deallocate(NewMatrix)
deallocate(Matrix)
deallocate(NumColumns)
deallocate(NewEquality)
deallocate(VarName)
deallocate(NewNameRHS)
deallocate(Equality)
deallocate(NameRHS)
if (WriteBinary == 'Y' .or. WriteBinary == 'y') then
  deallocate(BinaryVariableName)
  deallocate(BinaryConstraintName)
end if
deallocate(NewMatrix)

return

end subroutine Excel2MPS

subroutine ChangeName(Input,Output,Found)
!! Changes a string of unknown length to one of at most 8 characters
!! It accepts a string (Input) which will be reduced in length
!! It returns a string (Output) of length 8 that is unique to the model
!! It returns a string (Found) of length 1 that is 'Y' if the input has been encountered before, else 'N'
implicit none
integer :: z, error
character(*), intent(in) :: Input
character(8), intent(out) :: Output

```



```

character(1), intent(out) :: Found ! Check whether the name has been used before
character(1) :: NumChar1 ! Used to add a 1 digit number at the end of a string
character(2) :: NumChar2 ! Used to add a 2 digit number at the end of a string
character(6) :: Temp
character(6) :: AdjustedInput ! The input string shortened/lengthened to 6 characters
character(8) :: Name1
character(30) :: Name2
character(1), parameter :: zero = '0'

! Ensure that there are no empty strings used as inputs
if (LEN(trim(adjustl(Input))) == 0) then
  write(*, '(A)') 'DANGER: Zero length input into Subroutine ChangeName!'
  read(*,*)
end if

! Begin reading the name file from the beginning
rewind(13)
read(13,*, iostat=error)Name1, Name2
Found = 'N'
do while (Found == 'N' .and. error == 0)

  ! Check if the name has been previously used
  if (trim(adjustl(Name2)) == trim(adjustl(Input))) then
    Output = Name1
    Found = 'Y'
  else ! Otherwise read the next line
    read(13,*, iostat=error)Name1, Name2
  end if
end do

! Create the 8 character name of the input
z = 1
if (Found == 'N') then ! If the name has not yet been encountered
  error = LEN(trim(adjustl(Input)))
  if (error == 0) write(*,*)error, Input ! Ensure no zero input (for debugging)
  call NameLength(Input, AdjustedInput) ! Reduce to 6 characters
  rewind(13)
  read(13,*, iostat=error)Name1, Name2

  ! Count the number of times names with the same first 6 characters have been encountered
  do while (error == 0)
    Temp = trim(adjustl(Name1))
    if (AdjustedInput == Temp) then
      z = z + 1
    end if

    ! Stop the program if there are more than 100 names with the same 6 first characters
    if (z > 100) then
      write(*,*) 'Program must be stopped – more than 99 names with the same 6 first characters.'
      stop
    end if
    read(13,*, iostat=error)Name1, Name2
  end do

  ! Begin creating the new name that will be used for the MPS file
  if (LEN(trim(adjustl(Input))) <= 6) then
    Output = trim(adjustl(AdjustedInput))
  else if (LEN(trim(adjustl(Input))) > 6 .and. z < 10) then
    write(NumChar1, '(I1)')z
    Output = AdjustedInput//zero//NumChar1
  else if (LEN(trim(adjustl(Input))) > 6 .and. z >= 10 .and. z <= 99) then
    write(NumChar2, '(I2)')z
    Output = AdjustedInput//NumChar2
  else if (LEN(trim(adjustl(Input))) > 6 .and. z == 100) then
    Output = AdjustedInput//zero//zero
  end if
end if

backspace(13)
write(13, '(A,1x,A)')Output, trim(adjustl(Input))

```

```

    return
end subroutine ChangeName

subroutine NameLength(In,Out)
!! Change an input string to a string with at most 6 characters
!! It accepts a string (In) which must be reduced in length (if length is > 6)
!! It returns a string (Out) with length 6 or required spaces if shorter
implicit none
integer :: Length
character(*), intent(in) :: In
character(6), intent(out) :: Out
character(1), parameter :: space = ' '

    Length = len(trim(adjustl(In)))
    if (Length > 6) then
        Out = trim(adjustl(In))
    else if (Length == 6) then
        Out = trim(adjustl(In))
    else if (Length == 5) then
        Out = trim(adjustl(In))//space
    else if (Length == 4) then
        Out = trim(adjustl(In))//space//space
    else if (Length == 3) then
        Out = trim(adjustl(In))//space//space//space
    else if (Length == 2) then
        Out = trim(adjustl(In))//space//space//space//space
    else if (Length == 1) then
        Out = trim(adjustl(In))//space//space//space//space//space
    else
        write(*,*)'There seems to be a name shorter than 1 character.'
    end if

return
end subroutine NameLength

subroutine BinaryName(Input,Output,Prefix)
!! Changes a string of unknown length to one of at most 8 characters with a 'b' as prefix
!! It accepts a string (Input) which will be reduced in length
!! It returns a string (Output) of length 8 that is unique to the model
!! It accepts a string (Prefix) of length 1 to add to the front of the name
implicit none
integer :: z, error
character(*), intent(in) :: Input
character(1), intent(in) :: Prefix
character(8), intent(out) :: Output
character(1) :: Found ! Check whether the name has been used before
character(1) :: NumChar1 ! Used to add a 1 digit number at the end of a string
character(2) :: NumChar2 ! Used to add a 2 digit number at the end of a string
character(3) :: NumChar3 ! Used to add a 3 digit number at the end of a string
character(5) :: Temp
character(5) :: AdjustedInput ! The input string shortened/lengthened to 5 characters
character(8) :: Name1
character(30) :: Name2
character(1), parameter :: zero = '0'

! Ensure that there are no empty strings used as inputs
if (LEN(trim(adjustl(Input))) == 0) then
    write(*, '(A)') 'DANGER: Zero length input into Subroutine ChangeName!'
    read(*,*)
end if

Found = trim(adjustl(Input))
if (Found == 'v' .or. Found == 'c') then
    write(*, '(A,A,A,A)') 'Error, ', trim(adjustl(Input)), ' should not have this prefix ', Found
end if
call PrefixFour(Input,AdjustedInput,Prefix) ! Reduce to 6 characters
rewind(13)
read(13,*,iostat=error)Name1, Name2

```

```

! Count the number of times names with the same first 5 characters have been encountered
z = 1
do while (error == 0)
  Temp = trim(adjustl(Name1))
  if (Temp == AdjustedInput) then
    z = z + 1
  end if

! Stop the program if there are more than 1000 names with the same 6 first characters
if (z > 1000) then
  write(*,*)'Program must be stopped – more than 999 names with the same 6 first characters.'
  read(*,*)
  stop
end if
read(13,*,iostat=error)Name1, Name2
end do

! Begin creating the new name that will be used for the MPS file
if (LEN(trim(adjustl(Input))) < 5) then
  Output = trim(adjustl(AdjustedInput))
else if (LEN(trim(adjustl(Input))) >= 5 .and. z < 10) then
  write(NumChar1,'(I1)')z
  Output = AdjustedInput//zero//zero//NumChar1
else if (LEN(trim(adjustl(Input))) > 5 .and. z >= 10 .and. z <= 99) then
  write(NumChar2,'(I2)')z
  Output = AdjustedInput//zero//NumChar2
else if (LEN(trim(adjustl(Input))) > 5 .and. z >= 100 .and. z <= 999) then
  write(NumChar3,'(I3)')z
  Output = AdjustedInput//NumChar3
else if (LEN(trim(adjustl(Input))) > 5 .and. z == 1000) then
  Output = AdjustedInput//zero//zero//zero
end if

backspace(13)
Name2 = Prefix//trim(adjustl(Input))
write(13,'(A,Ix,A)')Output, trim(adjustl(Name2))

return
end subroutine BinaryName

subroutine PrefixFour(In,Out,Pre)
!! Change an input string to a string with at most 4 characters
!! It accepts a string (In) which must be reduced in length (if length is > 4)
!! It returns a string (Out) with length 5 or required spaces if shorter
implicit none
integer :: Length
character(*), intent(in) :: In
character(1), intent(in) :: Pre
character(5), intent(out) :: Out
character(1), parameter :: space = ' '

Length = len(trim(adjustl(In)))
if (Length > 4) then
  Out = Pre//trim(adjustl(In))
else if (Length == 4) then
  Out = Pre//trim(adjustl(In))
else if (Length == 3) then
  Out = Pre//trim(adjustl(In))//space
else if (Length == 2) then
  Out = Pre//trim(adjustl(In))//space//space
else if (Length == 1) then
  Out = Pre//trim(adjustl(In))//space//space//space
else
  write(*,*)'There seems to be a name shorter than 1 character.'
end if

return
end subroutine PrefixFour

subroutine WriteValue(Unit,String1,String2,Value,ShowZero)

```

```

!! Write a part of an MPS file where the value has as much accuracy as can be portrayed in an MPS file
!! It accepts an integer (Unit) that indicates to which file the information should be written
!! A string (String1) that will be written first
!! A string (String2) that will be written second
!! A real value (Value) that will represent the coefficient to which the two strings refer
!! A string (ShowZero) that tells the program whether zero values should be written to file
implicit none
real, intent(in) :: Value
integer, intent(in) :: Unit
character(1), intent(in) :: ShowZero ! 'Y' if zero values should be shown
character(*), intent(in) :: String1, String2

if (Value > 9999999. .or. Value < -9999999.) then ! If the value is too large to be shown
  write(*,*)'Value out of range, program must be stopped.'
  stop
else if ((Value <= 9999999. .and. Value > 999999.9) .or. (Value < -999999.9 .and. Value >=
-999999.)) then
  write(Unit, '(4x,A8,2x,A8,2x,F8.0)')String1, String2, Value
else if ((Value <= 999999.9 .and. Value > 99999.99) .or. (Value < -9999.99 .and. Value >=
-99999.9)) then
  write(Unit, '(4x,A8,2x,A8,2x,F8.1)')String1, String2, Value
else if ((Value <= 99999.99 .and. Value > 9999.999) .or. (Value < -999.999 .and. Value >=
-9999.99)) then
  write(Unit, '(4x,A8,2x,A8,2x,F8.2)')String1, String2, Value
else if ((Value <= 9999.999 .and. Value > 999.9999) .or. (Value < -99.9999 .and. Value >=
-999.999)) then
  write(Unit, '(4x,A8,2x,A8,2x,F8.3)')String1, String2, Value
else if ((Value <= 999.9999 .and. Value > 99.99999) .or. (Value < -9.99999 .and. Value >=
-99.9999)) then
  write(Unit, '(4x,A8,2x,A8,2x,F8.4)')String1, String2, Value
else if ((Value <= 99.99999 .and. Value > 9.999999) .or. (Value < -.999999 .and. Value >=
-9.99999)) then
  write(Unit, '(4x,A8,2x,A8,2x,F8.5)')String1, String2, Value
else if ((Value <= 9.999999 .and. Value > .9999999) .or. (Value <= -.000001 .and. Value >=
-.999999)) then
  write(Unit, '(4x,A8,2x,A8,2x,F8.6)')String1, String2, Value
else if (Value <= .9999999 .and. Value >= .0000001) then
  write(Unit, '(4x,A8,2x,A8,2x,F8.7)')String1, String2, Value

! Set very small values equal to zero
else if (Value < .0000001 .and. Value > -.000001 .and. ShowZero == 'Y') then
  write(Unit, '(4x,A8,2x,A8,2x,A8)')String1, String2, '0.'
end if

return
end subroutine WriteValue

subroutine PCx2CSV(NamesUnit, OutName, Variables, Constraints, Direct)
!! This converts the output from PCx to a spreadsheet format (CSV file)
!! It accepts an integer (NamesUnit) indicating which file contains the names and their 8 character equivalents
!! A string (OutName) that will be used to create the output filename
!! An integer (Variables) indicating how many variables there are in the problem
!! An integer (Constraints) indicating how many constraints there are in the LP
!! A string (Direct) which is 'N' when an MPS file has been created, anything else if only the PCx to CSV conversion
is wanted
implicit none
integer, intent(in) :: NamesUnit
integer, intent(in) :: Variables
integer, intent(in) :: Constraints
character(1), intent(in) :: Direct
character(*), intent(in) :: OutName
real :: Value1, Value2
real :: TempNum
integer :: e, i
character(1) :: Eq
character(8) :: OldName1
character(10) :: TenName
character(20) :: TempName
character(30) :: OldName2
character(4), parameter :: dotout = '.out'

```

```

character(4), parameter :: dotcsv = '.csv'

! Ask the user to use the MPS file created to solve the problem and copy the output file back to the working
! directory
if (Direct == 'N') then
  TempName = OutName//dotout
  write(*,*)
  write(*,'(A)')'The program has completed the MPS file.'
  write(*,'(A,A,A)')'Please solve the problem with PCx and copy ', trim(adjustl(TempName)), ' back to this
  ! folder.'
  write(*,*)
  read(*,*)
end if

! Check whether the user has copied the correct file back to the working directory
e = 1
do while (e /= 0)
  open(unit=12,file=trim(adjustl(TempName)),status='old',action='read',iostat=e)
  if (e /= 0) then
    write(*,'(A,A,A)')'There is a problem opening ', trim(adjustl(TempName)), '&
    ! ', please ensure you have copied the file.'
    read(*,*)
  end if
end do

! Open the file used for the spreadsheet model
TempName = OutName//dotcsv
open(unit=14,file=trim(adjustl(TempName)),status='replace',action='write',iostat=e)
if (e /= 0) then
  write(*,*)'There is a problem opening ', trim(adjustl(TempName)), ', program will be stopped.'
  stop
end if

! Read the PCx output to memory
read(12,*)
read(12,*)
read(12,*)
do e=1,Variables
  read(12,*)TempNum, TenName, Value1
  rewind(NamesUnit)
  i = 1

  ! Search for the name that has been read in the PCx output and find its long version
  do while (trim(adjustl(TenName)) /= trim(adjustl(OldName1)) .and. i <= Variables+Constraints)
    read(NamesUnit,*)OldName1, OldName2
    i = i + 1

    ! If the search has passed the end of the names file, stop, there is an error
    if (i == Variables+Constraints) then
      write(*,'(A,A,A)')'Could not find variable name ', trim(adjustl(TenName)), '.'
      stop
    end if
  end do
  write(14,'(A,A,F10.2)')OldName2, ', ', Value1
end do
read(12,*)
read(12,*)OldName2

! Ensure that there are no variables that disappeared and the constraint information begins at the right point
if (trim(adjustl(OldName2)) /= 'Constraints:') then
  write(*,*)'Instead of "Constraints:", ', trim(adjustl(OldName2)), ' has appeared.'
  stop
end if
read(12,*)

! Write the constraint information to the CSV file
write(14,*)
write(14,'(A)')'Constraints'
do e=1,Constraints
  read(12,*)TempNum, TenName, Eq, Value1, Value2

```



```

rewind(NamesUnit)
do while (TenName /= OldName1)
  read(NamesUnit,*)OldName1, OldName2
end do
do
  if (Eq == 'G') write(14,'(A,A,F10.2,A,F10.2)'trim(adjustl(OldName2)), ',', Value1, ',>=', Value2
  if (Eq == 'E') write(14,'(A,A,F10.2,A,F10.2)'trim(adjustl(OldName2)), ',', Value1, ',==', Value2
  if (Eq == 'L') write(14,'(A,A,F10.2,A,F10.2)'trim(adjustl(OldName2)), ',', Value1, ',<=', Value2
end do

close(12)
close(14)

return
end subroutine PCx2CSV

subroutine NEOS2CSV(NamesUnit,OutName,Variables,Constraints,Binary)
!! This converts the output from NEOS to a spreadsheet format (CSV file)
!! It accepts an integer (NamesUnit) indicating which file contains the names and their 8 character equivalents
!! A string (OutName) that will be used to create the output filename
!! An integer (Variables) indicating how many variables there are in the problem
!! An integer (Constraints) indicating how many constraints there are in the LP
!! A string (Binary) which is 'Y' when the file has been created to reduce the number of arcs
implicit none
integer, intent(in) :: NamesUnit
integer, intent(in) :: Variables
integer, intent(in) :: Constraints
character(1), intent(in) :: Binary
character(*), intent(in) :: OutName
real :: Value1, Value2
real :: TempNum
real :: SlackValue
real :: DualValue
real :: RHS
integer :: e, i, j
integer :: Vars, Cons
integer :: NumSpacesC, NumSpacesV
character(1) :: Eq
character(2) :: At
character(8) :: OldName1
character(8) :: EightName
character(20) :: TempName
character(30) :: OldName2
character(1), parameter :: BinaryY = 'i'
character(1), parameter :: BinaryN = 'r'
character(4), parameter :: dotout = '.out'
character(4), parameter :: dotcsv = '.csv'

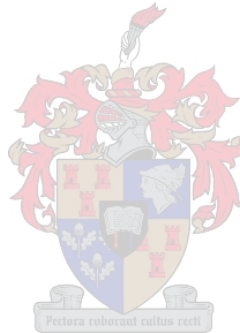
e = 1
do while (e /= 0)

  ! Ask the user to use the MPS file created to solve the problem and copy the output file back to the working
  ! directory
  write(*,*)
  write(*,'(A)',advance='no') 'What is the name of the NEOS output file? '
  read(*,*)TempName

  ! Check whether the user has copied the correct file back to the working directory
  open(unit=12,file=trim(adjustl(TempName)),status='old',action='read',iostat=e)
  if (e /= 0) then
    write(*,'(A,A,A)') 'There is a problem opening ', trim(adjustl(TempName)), ', please ensure you have
    the correct filename.'
    read(*,*)
  end if
end do

! Open the file used for the spreadsheet model
if (Binary == 'Y' .or. Binary == 'y') then
  TempName = trim(adjustl(OutName))//BinaryY//dotcsv
else
  TempName = trim(adjustl(OutName))//BinaryN//dotcsv

```



```

end if
open(unit=14,file=trim(adjustl(TempName)),status='replace',action='write',iostat=e)
if (e /= 0) then
  write(*,*)'There is a problem opening ', trim(adjustl(TempName)), ', program will be stopped.'
  stop
end if
write(14,'(A)')'Constraints'
write(14,'(A)')'Name,Value,SlackValue,ShadowPrice,RHS'

! Read the PCx output to memory
if (Binary == 'Y' .or. Binary == 'y') then
  Vars = 2*Variables
  Cons = Variables + Constraints
else
  Vars = Variables
  Cons = Constraints
end if
NumSpacesC = floor(real(Cons)/21.)
NumSpacesV = floor(real(Vars)/21.)
read(12,*)
read(12,*)
read(12,*)
do j=2,Cons+NumSpacesC
  if (j /= 22*nint(real(j)/22.)) then
    read(12,*,iostat=e)Eq, TempNum, EightName, At, Value1, SlackValue, DualValue, RHS
    if (e /= 0) then
      write(*,*)
      write(*,'(A)')'There is a problem reading from the NEOS input file'
      e = j + 2
      write(*,'(A,I4)')'Line number ', e
      write(*,*)Eq, TempNum, EightName, At, Value1, SlackValue, DualValue, RHS
      read(*,*)
    end if
    rewind(NamesUnit)

    ! Search for the name that has been read in the PCx output and find its long version
    i = 0
    do while (trim(adjustl(EightName)) /= trim(adjustl(OldName1)) .and. i <= Vars+Cons)
      read(NamesUnit,*)OldName1, OldName2

      ! If the search has passed the end of the names file, stop, there is an error
      if (i == Vars+Cons) then
        write(*,'(A,A,A)')'Could not find variable name ', trim(adjustl(EightName)), '.'
        stop
      end if
      i = i + 1
    end do
    write(14,'(A,A,F10.2,A,F10.2,A,F10.2,A,F10.2)')OldName2, ',', Value1, ',', SlackValue, ',', DualValue,
      ',', RHS
  else
    read(12,*)
  end if
end do
read(12,*)
read(12,*)OldName2

! Ensure that there are no variables that disappeared and the constraint information begins at the right point
if (trim(adjustl(OldName2)) /= 'ColumnsSection' .and. trim(adjustl(OldName2)) /= 'Columns') then
  write(*,'(A,A,A)')'Instead of "ColumnsSection" or "Columns", ', trim(adjustl(OldName2)), ' has
  appeared.'
  stop
end if
read(12,*)

! Write the constraint information to the CSV file
write(14,*)', '
write(14,'(A)')'Variables'
write(14,'(A)')'Name,Value,InputCost,ReducedCost'
do j=1,Vars+NumSpacesV
  if (j /= 22*nint(real(j)/22.)) then

```



```

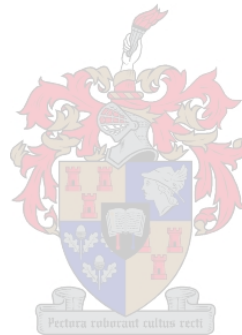
read(12,*,iostat=e)Eq,TempNum, EightName, At, Value1, Value2, RHS
if (e /= 0) then
  e = 2+Cons+NumSpacesC+3+j
  write(*,*)
  write(*, '(A,I4)') 'Problem reading line number ', e
  write(*,*) 'j = ', j, ' vars = ', Vars, ' SpacesV = ', NumSpacesV
  write(*,*) Eq,TempNum, EightName, At, Value1, Value2, RHS
  read(*,*)
end if
rewind(NamesUnit)
do while (EightName /= OldName1)
  read(NamesUnit,*) OldName1, OldName2
end do
write(14, '(A,A,F10.2,A,F10.2,A,F10.2)') OldName2, ',', Value1, ',', Value2, ',', RHS
else
  read(12,*)
end if
end do

close(12)
close(14)

return
end subroutine NEOS2CSV

end program convert

```



Appendix H

Compact Disc Contents

A compact disc has been added to this thesis to provide the reader with a better understanding of the layout of the *Fruitlog Scenario Model*, as well as the form that the graph theoretic and mathematical programming models took, once in spreadsheet format. The source code for the programs *FruitFly* and *Convert* are also included.

The spreadsheet forms of the graph theoretic models included on the compact disc, include the standard solution to the minimum cost flow summer and winter scenarios for all three subtropical cooling costs (in the folder **Minimum Cost Flow Models**). These are simply called **MinSumGT***.csv** and **MinWinGT***.csv**, where the ******* is replaced by the cooling cost of the subtropical fruit. Furthermore, the scenarios of unrestricted fruit flow through the ports are given in **MinSumUnres.csv** and **MinWinUnres.csv**. The summer scenario where fruit may only be exported via the port of Cape Town may be found in **MinSumCT.csv**. Recall that the last three models are for a subtropical cooling tariff of $R160$ per pallet. The files used to solve the maximum flow problems are called **MaxSumGT.csv** for the summer scenario and **MaxWinGT.csv** for the winter scenario and may be found in the folder **Maximum Flow Models**.

There are many more files for the mathematical programming approach. These include files used to solve the minimum cost flow standard summer scenario (for all three subtropical cooling tariffs — the file names have the form **SumMin(***)LP*.csv**), and those files used to solve the the models for scenarios where a larger weighting was placed on each fruit separately (which have the form ***SumLP*.csv**, where the first asterisk signifies which fruit has the additional weighting in the objective function). The minimum cost flow scenario where the summer peak is considered and fruit may only be exported via the port of Cape Town is contained in the files called **SumMinCTLP*.csv**. The minimum cost flow models for the winter scenario are named **WinMin(***)LP*.csv** for the standard solution. The scenario where the ratio of exported fruit via containers to break-bulk was forced to remain below a certain value is stored in the files with the name format **WinMinRatioLP*.csv**. Both the summer and winter peak weeks had scenarios where the terminal capacities were very large. These scenarios are saved in the files called **SumMinUnresLP*.csv** and **WinMinUnresLP*.csv** respectively. These files may all be found in the folder named **Minimum Cost Flow Models**). The maximum flow models are described in the files called **SumLP*.csv** for Scenario One in Chapter 5, while the file **Win1LP*.csv** is for Scenario Two and **Win2LP*.csv** for Scenario Three (in the folder called **Maximum Flow Models**). The load files for the program *Convert* are also all included.

