Applying tree knapsack approaches to general network design: A case study.

T. Baitshenyetsi, Hons. B.Sc.

Dissertation submitted in partial fulfilment of the requirements for the degree Magister Scientiae in Computer Science at the Potchefstroom campus of the North-West University

Supervisor: Prof H.A. Kruger

Co-supervisor: Prof J.M. Hattingh

November 2010
Potchefstroom

Contents

1 Introduction 1

1.1 Introduction 1

1.2 Problem statement 2

1.3 Objectives of the study 4

1.4 Research Methodology 4

1.5 Chapter outline 5

1.6 Conclusion 5

2 An overview of network flow models and knapsack problems 6

2.1 Introduction 6

2.2 Network flow models 6

2.2.1 Introduction 6

2.2.2 Transportation problems 8

2.2.3 Assignment problems 10

2.2.4 Transshipment problems 11

2.2.5 Shortest route problem 14

2.2.6 Maximum flow problem 16

2.2.7 Minimal spanning tree problem 17

2.2.8 Network design issues 18

2.3 The knapsack problem and extensions 22

2.3.1 Introduction 22

2.3.2 Types of knapsack models 23

2.3.3 Knapsack problem applications 32

2.4 Conclusion 33

3 An overview of the oil pipeline design problem 34

3.1 Introduction 34

3.2 The oil pipeline design problem 34

3.3 The model suggested by Brimberg et al. (2003) 36

3.4 Solution methods 38

3.5 Model results 39

3.6 Conclusion 39

4 Model development for the oil pipeline design problem 41

4.1 Introduction 41

4.2 Methodology and model development 41

4.2.1 Converting the pipeline network into a tree network structure 41

4.2.2 Model development 45

4.2.2.1 The objective function 45

4.2.2.2 Model constraints 48

4.2.2.3 The complete model 55

4.3 Conclusion 56

5 Empirical experiment and results 57

5.1 Introduction 57

5.2 Empirical experiment 57

5.2.1 Data used 59

5.2.2 Tree generation 60

5.2.3 The extended tree knapsack model 62

5.3 Results and discussion 64

5.4 Conclusion 69

6 Summary and conclusions 70

6.1 Introduction 70

6.2 Objectives of the study 70

6.3 Problems experienced 73

6.4 Possibilities for further research 73

6.5 Conclusion 74

Appendices

Appendix A: Arc distances 75

Appendix B: Production at each well site 76

Appendix C: Program pseudo code 77

Appendix D: Generated tree network 80

Appendix E: Computed cost incurred (a_{ij}) 82

List of Figures

Figure 2.1 Example of a network diagram........7

Figure 2.2 Network representation of a transportation problem........9

Figure 2.3 Network representation of a transshipment problem........12

Figure 2.4 Sample tree network........30

Figure 3.1 South Gabon oil field (Brimberg *et al*., 2003)........35

Figure 3.2 Subnetworks of South Gabon oil field........40

Figure 4.1 Illustrative network........42

Figure 4.2 Converting a network into a tree network........44

Figure 4.3 Graphical representation of cost function in Brimberg *et al* (2003)........46

Figure 4.4 Graphical representation of a fixed charges cost model........46

Figure 4.5 Example of the contiguity assumption constraint instance........48

Figure 4.6 An indexed tree network........49

Figure 4.7 An indexed tree network with production at nodes........52

Figure 4.8 Diagonal matrix with production........52

Figure 4.9 Node-arc incidence matrix........53

Figure 5.1 South Gabon oil field network........58

Figure 5.2 Abstract of tree representation........61

Figure 5.3 Computed cost incurred for arc 1-2........63

Figure 5.4 Tree knapsack solution for South Gabon oil pipeline network........67

Figure 5.5 Illustration of possible over and underestimation of costs for arc (1,4)........68

List of Tables

Table 5-1: Monetary units and pipe capacities. 60

Table 5-2: Calculated cost of each pipe 62

Table 5-3: Calculated cost for arc (1,2) 63

Table 5-4: Selected arcs for the solution 65

Table 5-5: Solution comparison 66

Bibliography 85

Abstract

There are many practical decision problems that fall into the category of network flow problems: numerous examples of applications can be found in areas such as telecommunications, logistics, distributions, engineering, computer science and so on. One of the most popular and valuable tools to solve network flow problems of a topological nature is the use of linear programming models. An important extension of these models is that of integer programming models that deal with problems where some, or all, of the variables are required to assume integer variables. A significant application in this class of problems is the knapsack problem that arises in different contexts such as loading containers in aircraft or satisfying the demand for various lengths of cloth which must be cut from fixed length bolts of fabric.

In this study, the feasibility of representing a network flow model in a tree network model and subsequently solving it using a tree knapsack approach is investigated. To compare and validate the proposed technique, a specific case study was chosen from the literature that can be used as a basis for the research project. The said study was an oil pipeline design problem, addressed by Brimberg *et al.* (2003). This focuses on the optimal design of an oil pipeline network for the South Gabon oil field in Africa. The objective was to reduce oil transportation costs to a major port. Following an overview of different network flow and knapsack models, an overview of the said matter is presented. A description of the proposed tree knapsack approach and the application of this approach to the given problem is given. Results have indicated that it is feasible to apply a tree knapsack approach to solve network flow problems.

Keywords: Linear programming models, integer programming, network flow, tree knapsack, oil pipeline network.

Uittreksel

Daar bestaan baie praktiese besluitnemingprobleme wat geklassifiseer kan word as netwerkvloei probleme. Voorbeelde hiervan kan gevind word in verskeie velde soos telekommunikasie, logistiek, ingenieurswese, rekenaarwetenskap, ens. Een van die mees waardevolle tegnieke om ‘n netwerkvloei probleem op te los is die gebruik van lineêre programmeringsmodelle. ‘n Belangrike uitbreiding van lineêre programmering modelle is heeltallige modelle waar sekere, of alle, veranderlikes heeltallige waardes het. ‘n Belangrike toepassing binne hierdie klas van probleme is die “knapsak” probleem wat in verskillende kontekste aangewend kan word, byvoorbeeld, die laai van houers in ‘n vliegting of die vraag na sekere lengtes materiaal wat gesny moet word. In hierdie studie word die moontlikheid en toepaslikheid van die gebruik van ‘n boom knapsak metode, om ‘n netwerkvloei probleem op te los, ondersoek.

Om hierdie metode te vergelyk, en die geldigheid daarvan te toets, is ‘n spesifieke gevallestudie uit die literatuur gekies. Die gevallestudie wat gekies is, handel oor die ontwerp van ‘n oliepypleiding probleem (Brimberg *et al.,* 2003). Die oliepypleiding probleem fokus op die optimale ontwerp van ’n oliepypnetwerk vir die Suid-Gaboen olie veld in Afrika. Die doel hiervan is om die vervoerkoste van olie na ‘n hawe te verminder. ‘n Oorsig van verskillende soorte netwerkvloei modelle asook knapsak modelle sal aangebied word. Die oliepypleiding probleem sal ook bespreek word. ‘n Beskrywing van die voorgestelde boom knapsak benadering asook die toepassing hiervan op die oliepypleiding probleem sal gegee word. Resultate het aangedui dat netwerkvloei probleme wel deur ‘n boom knapsak benadering opgelos kan word.

Sleutelwoorde: Lineêre programmering, heeltallige programmering, netwerkvloei, boom knapsak, oliepypleiding netwerk.

Acknowledgements

The writing and completing of this dissertation has been one of the greatest academic experiences that I have ever had to face. There were challenges that without the support, patience and guidance of the following people, this research study would not been possible. It is to them that I am heartily thankful to.

- Prof Hennie Kruger who undertook to act as my supervisor. His support, guidance and commitment to the highest standard made it possible to complete this study.

- Prof Giel Hattingh my co-supervisor without whom this research work will be worth nothing. Despite his health challenges and other commitments, his wisdom and knowledge inspired and motivated me to complete this work.

- My friends and colleagues who participated in this research with great interests, and my parents Mr and Mrs Baitshenyetsi who always supported and encouraged me in all my endeavours.

- I will fail as student if I do not put on record my deep appreciation to the one who kept my spirit alive, who gave me creative ideas and guided me through this research work, my heavenly father God, I bless His name

- It is always impossible to personally thank everyone who has facilitated the successful completion of this research project. To those who I did not specifically name, I also give my thanks for moving me towards my goal.

.

1

Introduction

1.1 Introduction

Many management decisions are focused on the best way to achieve certain objectives subject to certain restrictions. These may take the form of limited resources such as time, labour, energy, materials or money; or they could assume the form of restrictive guidelines such as a recipe or engineering specifications. Whenever a decision maker or researcher attempts to solve a type of problem by seeking an objective subject to restriction, a management science technique called linear programming is frequently used (Taylor, 2002).

It can be defined as a mathematical programming model with a linear objective and linear constraints. Mathematically such a model can be represented as follows (Weatherford and Moore, 2001)

Maximize (or minimize) $f(x_1,\ldots\ldots..x_n)$,

subject to the constraints that

$$g_1(x_1,\ldots\ldots..x_n) \;\; \geq, \; =, \leq \;\; b_{1,}$$

$$\vdots$$

$$g_m(x_1,\ldots\ldots..x_n) \;\; \geq, \; =, \leq \;\; b_{m,}$$

where f and $g_{1,}$ . . . . .,g_m are linear functions.

There are different types and extensions of general linear programming problems. Network problems, for example, constitute a large and special class of linear programming models and are used to solve a variety of problems such as shortest route, maximum flow, minimal spanning tree and the like. Another important extension of these models is found in integer programming models that deal with problems where some, or all, of the variables are required to assume integer variables. A significant application in this class of problems is the knapsack problem that arises in different contexts such as loading containers in aircraft or satisfying the demand for various lengths of cloth which must be cut from fixed length bolts of fabric.

The purpose of this chapter is to guide the reader into the research project by explaining the problem statement, objectives of the study and the methodology that will be followed. A layout of the study, explaining the purpose of each chapter, is also presented.

1.2 Problem statement

Network flow problems represent a huge number of practical decision problems; many examples of applications can be found in the area of logistics, distribution, engineering, computer science etc. One of the most popular and valuable tools to solve problems of this type is the use of linear programming models. Consider for example the case where the shortest distance between an origin and a specific destination point in a network has to be determined. The classical linear program formulation for this network problem can be represented by

$$\text{Minimize} \quad \sum_{i,j} c_{ij} x_{ij}, \tag{1.1}$$

subject to

$$\sum_{j=1}^{m} x_{ij} = 1, \qquad i = 1,.....,m \qquad \text{(origin node } i\text{)} \tag{1.2}$$

$$\sum_{i=1}^{m} x_{ij} - \sum_{j=1}^{n} x_{ij} = 0, \quad i = 1,.....,m \text{,} \; j = 1,......,n \text{ (transshipment nodes)} \qquad (1.3)$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1,......,n \quad \text{(destination node } j\text{)} \qquad (1.4)$$

where

c_{ij} = the distance, time or cost associated with the arc from node *i* to node *j*,

$x_{ij} = 1$ if the arc from node *i* to node *j* is on the shortest route, and

$x_{ij} = 0$ otherwise,

and *m* and *n* indicate the appropriate numbers of nodes.

A good exposition of the technical detail regarding how to construct and solve network flow models can be found in Moore and Weatherford (2001).

One of the challenges that researchers have to deal with is to constantly try and enhance the models or to try and improve the time taken to solve these types of models. Recently a study by Van der Merwe and Hattingh (2006) has applied a tree knapsack approach to local area telecommunications networks in order to try and address these issues. See also Van der Merwe and Hattingh (2010).

For this research study it was decided to investigate the feasibility of representing a network flow model as a tree network model and subsequently solve it using a tree knapsack approach in a similar way to that in which Van der Merwe and Hattingh (2006) solved the local area telecommunication network problem. Furthermore, to compare and validate the proposed technique, it was decided to choose a specific case study in the literature that could be used as a basis for this research project. The problem selected was an oil pipeline design problem addressed by Brimberg *et al.* (2003)

During 2003, these authors performed a research study in an attempt to design an optimal oil pipeline network for the South Gabon oil field in Africa. They formulated a standard mathematical network model to determine the “best path” for oil flow. The model was solved heuristically by Tabu Search and Variable Neighborhood Search methods. They also solved it exactly using a branch-and-bound method. In this proposed study the given oil pipe network will be transformed into a tree structure and thereafter solved by a tree knapsack model analogous to the tree knapsack model used by Van der Merwe (2002).

1.3 Objectives of the study

The primary objective of this study is consequently to investigate the feasibility of using an extended tree knapsack approach to solve a network flow problem. This will be accomplished by addressing the following secondary research objectives.

- Gain a clear understanding of and present an introductory overview of general network flow and tree knapsack models;

- Select and provide a suitable case study from the literature that can be used in the research project;

- Describe and formulate the tree knapsack and extended tree knapsack approach and model; and

- Describe and present the results of the tree knapsack approach when applied to the selected case study.

1.4 Research Methodology

The research study will start with a general literature survey that will be used to give an overview of network flow models, knapsack and tree knapsack models and examples of applications of these models. This will be followed by empirical work to formulate and apply a tree knapsack model to the selected network flow problem in order to test the feasibility of using the tree knapsack approach.

1.5 Chapter outline

This section explains the purpose of each chapter and how it is structured.

Chapter 2 will offer an overview of the network flow models, knapsack and tree knapsack models. The most important types of problems will be briefly reviewed and where appropriate, the mathematical formulation will also be provided. Chapter 3 will be devoted to a description of the chosen case study – the oil pipeline design problem – while Chapter 4 will concentrate on the research design and methodology followed to develop the tree knapsack model. In Chapter 5 the said model is applied to the oil pipeline design problem and the results will be presented and discussed. The last chapter, Chapter 6, will then summarize the goals set forth for the study and how they were achieved. Opportunities for further studies will also be pointed out.

The abovementioned chapters are supplemented by a set of appendices which contain details of work related to the study.

1.6 Conclusion

Chapter 1 served as an introduction to the research project and explained the problem statement, objectives of the study and the methodology that will be followed. A layout of the study, explaining the purpose of each chapter, was also furnished.

2

An overview of network flow models and tree knapsack problems

2.1 Introduction

The objective of this study, to investigate the feasibility of representing a network flow problem as a tree structure and to solve it using a tree knapsack approach, implies that two main areas of research will be involved in the project; namely network flow and knapsack models. To provide sufficient background and to gain an understanding of these two areas, this chapter presents an introductory overview of such models. The main types and applications of both models as well as a discussion on tree knapsack models will be furnished.

2.2 Network flow models

2.2.1 Introduction

A network is an arrangement of paths connected to various points through which one or more items move from one point to another (Taylor, 2002). Network models have become very popular and are used in a variety of application areas such as the transmission of information, transportation of people, distribution of goods etc. Another reason for the popularity of networks models is that they can be drawn as diagrams – this literally provides a picture of the system under analysis and enables a manager or researcher to visually interpret a system.

A network diagram consists of two main components; nodes and branches (arcs). Nodes, usually denoted by circles, represent junction points (e.g. cities, intersections, air or railroad terminals etc.) while arcs connect the nodes and allow the flow from one point in the network to another. The network shown in figure 2.1 is an example of a network diagram. The nodes are numbered from 1 to 5 while a branch is denoted by the pair (*i, j*): for example, the branch form node 2 to node 4 will be denoted by the pair (2, 4). c_{ij} denotes the cost of traversing the branch (*i, j*) and u_{ij} denotes the capacity along route (*i, j*)

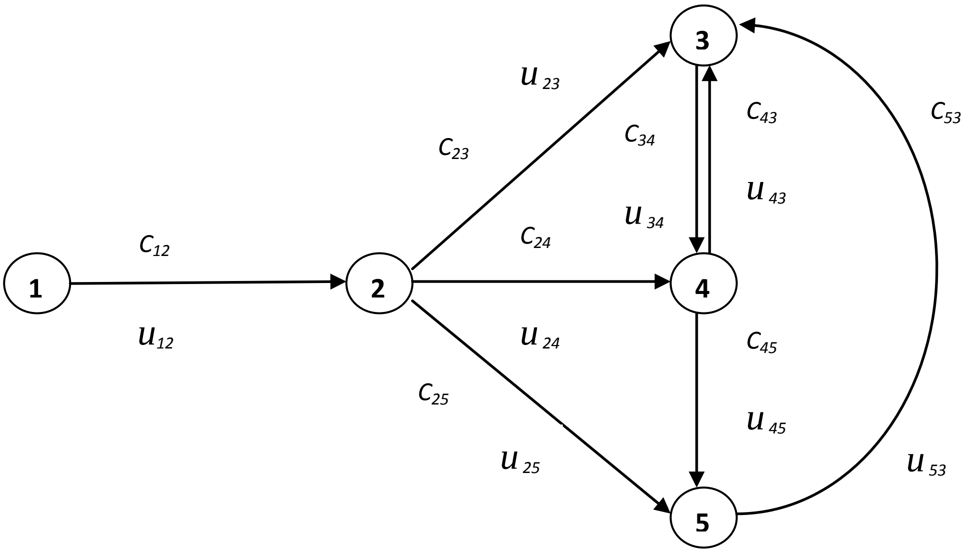


Figure 2.1 Example of a network diagram

According to Black and Tanenbaum (2010) a network or a graph can be described as a set of items connected by arcs. Each item is called a vertex or node and in terms of graph theory, the graph can be defined as a pair (*V*, *E*), where *V* is a set of vertices, and *E* is a set of edges between the vertices so that $E \subseteq \{(u,v) \mid u, v \in V\}$.

A number of practical decision problems fall into a general class of models known as network flow models. The most basic and familiar type of network models include projects to find the shortest path through a network (shortest-route), to establish the maximum flow of any quantity or substance through a network (maximum-flow) and to determine a path through a network that connects all the nodes while minimizing total distance or cost (minimal spanning tree). In addition to these models there are also three well known types of models involving sources and destinations that are members of the class of network flow models. These special types of models are known as transportation, transshipment and assignment problems.

The remainder of this section will present a brief overview of the network models mentioned above, starting with the three special cases.

2.2.2 Transportation problems

Transportation or shipping problems arise frequently in practice and involve determining the amount of goods or items to be transported from a number of origins (supply locations) to a number of destinations (demand locations). Typically the quantity of goods available at each origin is limited and the quantity needed at each destination is known. The objective is to minimize total shipping costs or distances.

The constraints in this type of problem deal with capacities at each origin and requirements at each destination. Figure 2.2 illustrates a network representation for a typical transportation problem with 3 origins and 4 destinations.

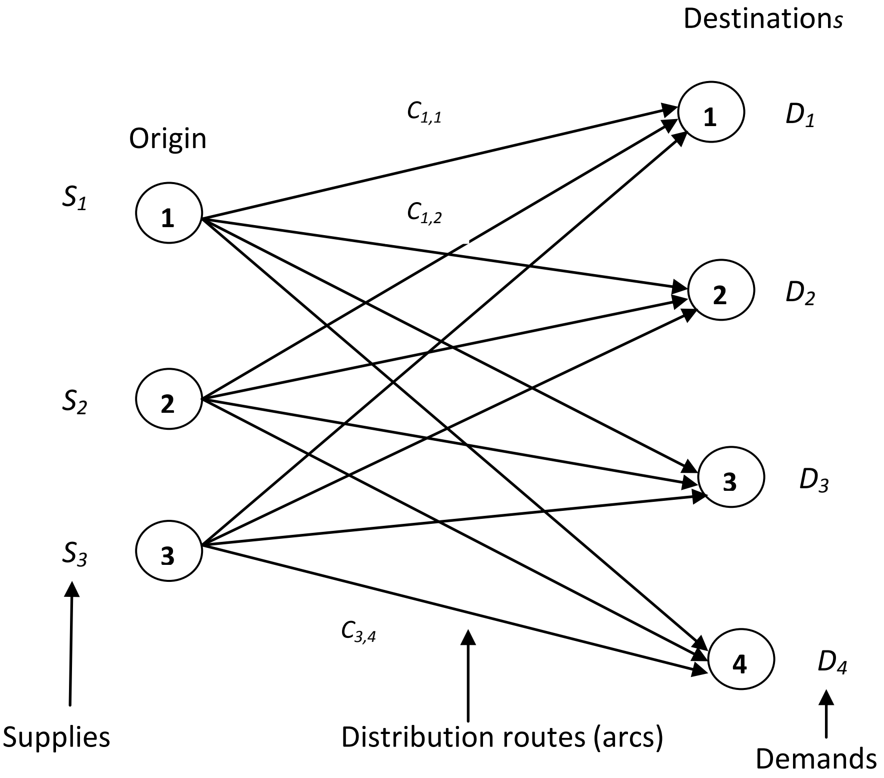


Figure 2.2 Network representation of a transportation problem

This network problem can now be solved by formulating it as a linear programming model, as follows.

Let x_{ij} = number of units shipped from origin i to destination j where i = 1, 2, 3 and j = 1, 2, 3, 4.

$$\text{Minimize} \quad \sum_{i=1}^{3}\sum_{j=1}^{4} c_{ij}x_{ij}, \tag{2.1}$$

subject to

$$\sum_{j=1}^{4} x_{ij} \le S_i, \qquad i = 1,2,3, \tag{2.2}$$

$$\sum_{i=1}^{3} x_{ij} = D_j, \qquad j = 1, 2, 3, 4, \tag{2.3}$$

$$x_{ij} \geq 0. \tag{2.4}$$

The above formulation is a basic model whereas in real world problems different variations of the model may occur. The following offer examples of situations where the model would need certain modifications:

- Total supply is not equal to total demand.
- Maximization of the objective function.
- Route capacities or route minima.
- Unacceptable routes.

A discussion on how to deal with these situations can be found in Anderson *et al*. (2009)

2.2.3 Assignment problems

Assignment problems involve determining the most efficient assignment of people to jobs, machines to tasks, police cars to city sectors, salespeople to territories etc (Render *et al.*, 2006). A distinguishing feature of the assignment problem is that one person is assigned to one and only one task. The problem can therefore be viewed as a special case of the transportation problem in which the supply at each origin and the demand at each destination is a single issue. The objective function in an assignment problem is usually to minimize time or costs or to maximize effectiveness and can be formulated as follows.

If *m* people need to be assigned to *n* tasks and c_{ij} denotes the cost of assigning person *i* to task *j*, then

let $x_{ij} = 1$ if person *i* is assigned to task *j*, 0 otherwise.

$$\text{Minimize} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij}, \tag{2.5}$$

subject to

$$\sum_{j=1}^{n} x_{ij} \leq 1, \qquad i = 1, 2, ...m, \tag{2.6}$$

$$\sum_{i=1}^{m} x_{ij} = 1, \qquad j = 1, 2, ...n, \tag{2.7}$$

$$x_{ij} \in \{0,1\} \qquad \text{for all } i \text{ and } j. \tag{2.8}$$

As with the transportation model, different variations are possible: e.g., an unacceptable assignment. A discussion on these variations can be found in Anderson *et al*. (2009).

2.2.4 Transshipment problems

A transshipment model is an extension of the transportation model. If items are being transported from a source through an intermediate point (called a transshipment point) before reaching a final destination, this is called a transshipment problem. Figure 2.3 depicts an example, with 2 origins, 2 transshipment nodes and 4 destinations.

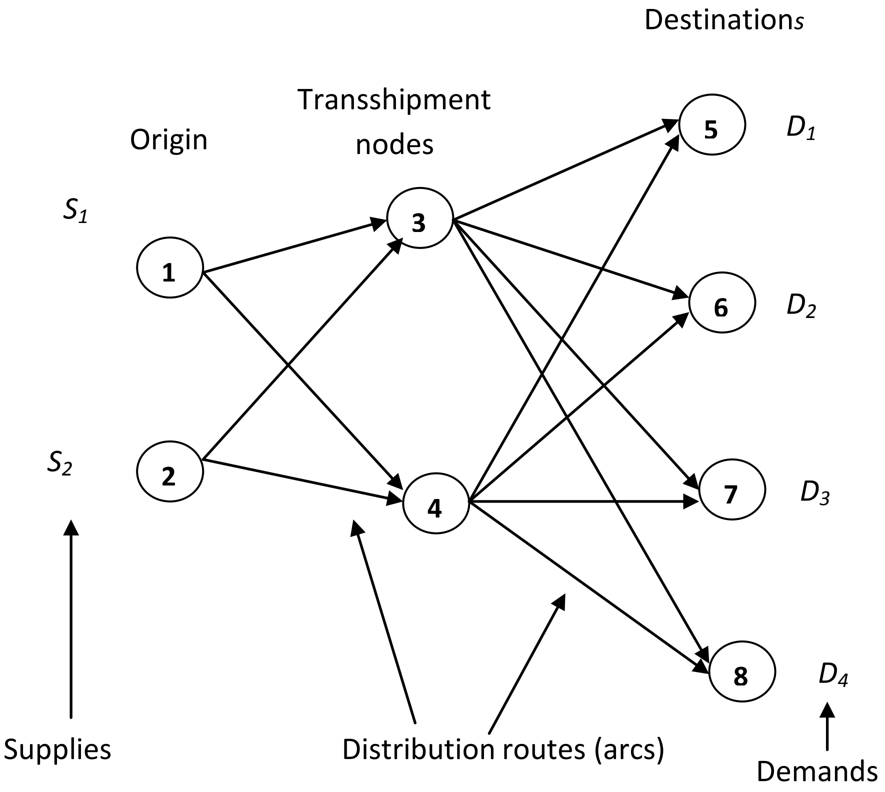


Figure 2.3 Network representation of a transshipment problem

The general linear programming model to solve a transshipment problem is formulated as follows.

Minimize $\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$, (2.9)

subject to

$$\sum_{j=1}^{n} x_{ij} \leq S_i, \qquad i=1,2,.....m, \tag{2.10}$$

$$\sum_{i=1}^{m} x_{ij} = D_j \qquad j=1,2,.....n, \tag{2.11}$$

$$\sum_{i=1}^{m} x_{ij} - \sum_{j=1}^{n} x_{ij} = 0, \tag{2.12}$$

$$x_{ij} \geq 0. \tag{2.13}$$

where

x_{ij} = number of units shipped from node *i* to node *j*,

c_{ij} = cost per unit shipping from node *i* to node *j*,

S_i = supply at origin node *i*,

D_j = demand at destination node *j*,

and *m* and *n* indicate the appropriate numbers of nodes.

As with transportation and assignment problems, transshipment problems may be formulated in terms of several variations, e.g. with route capacities. Anderson *et al*. (2009) discusses the various modifications required for different variations.

Transshipment problems frequently occur in management decision problems and many examples exist in the literature where these types of models were applied. For instance, Sharma and Jana (2009) describe a transshipment planning model for the petroleum refinery industry. The problem involves the transportation of refined oil from different refineries (origin) to various depots (transshipment nodes) and finally to various sales areas (destinations). The transportation mediums include pipelines, rail road and road tankers. This problem was solved by using a transshipment model and considering different objective functions such as minimization of costs, maximization of production capacity, minimization of oil storage at depots and so forth.

Other examples in the literature where a transshipment model approach was applied include the work of Klincewicz (1990) which focuses on solving a freight transport problem using facility location techniques. The objective of his model is to discover the minimum cost path either direct or indirect for shipments using shipping economics. Wee and Dada (2005) developed a formal model that focuses on the role of transshipment in a system of retailers who stock goods, while the work of Grahovac and Chakravarty (2001) focuses on transshipment of inventory in a supply chain with expensive low-demand items.

2.2.5 Shortest route problem

According to Weatherford and Moore (2001), a shortest route model refers to a network for which arc (*i*, *j*) has an associate number c_{ij}, which is interpreted as the distance (cost, time) from node *i* to node *j*. A route, or a path, between two nodes is any sequence of arcs connecting the two nodes. The objective, therefore, is to establish the shortest (least cost, least time) route from a specific node (origin) to another node (destination) in the network. This problem can be viewed and solved as a transshipment problem where the origin node has a supply of 1 and the destination node a demand of 1. All other nodes in the network have a demand (or supply) of 0. The formulation of the shortest route problem can be represented by the following.

$$\text{Minimize} \quad \sum_{i,j} c_{ij} x_{ij}, \tag{2.14}$$

subject to

$$\sum_{i=1}^{m} x_{ij} = 1, \qquad j = 1, 2, ...n, \quad \text{(origin node } i\text{)} \tag{2.15}$$

$$\sum_{i=1}^{m} x_{ij} - \sum_{j=1}^{n} x_{ij} = 0, \qquad \text{(transshipment nodes)} \tag{2.16}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1, 2, ...m, \quad \text{(destination node } j\text{)} \tag{2.17}$$

where

c_{ij} = the distance (time, cost) associated with the arc from node *i* to node *j*,

x_{ij} = 1 if the arc from node *i* to node *j* is on the shortest route,

x_{ij} = 0 otherwise,

and *m* and *n* indicate the appropriate numbers of nodes.

Shortest route problems have many applications; below are a few examples quoted from the literature to demonstrate the significance of these types of models.

Ragsdale (2007) stated that the equipment replacement problem is a common type of business issue that can be modelled as a shortest route problem. This involves determining the least costly schedule for replacing equipment over a specified length of time.

A school bus routing problem is described by Schittekat *et al*. (2006) where the shortest route model was applied to discover the optimal bus route. Some of the other recent applications of the shortest route model can be found in Lee *et al.* (2003) which focuses on optimal routing in non geostationary satellite ATM networks with inter satellite link capacity constraints. The model explores the routing of broadband communication services such as high definition TV (HDTV), video conferencing, high-speed data transfer and videophone on satellite asynchronous transfer mode (ATM) networks.

The work of Modarres and Zarei (2002) focuses on an application of network theory and the AHP (analytic hierarchy process) in urban transportation to minimize earthquake damage. Their objective is to determine the priority of trips, shortest paths, the fastest routes for daily trips, and the safest ones during an earthquake. Erkut and Verter (1998) modelled transport risks for hazardous materials with the objective of discovering the optimal path, taking into account the risk involved in transporting hazardous materials.

2.2.6 Maximum flow problem

In a maximal flow model the objective is to determine the maximum amount of flow (e.g. vehicles, messages, fluid etc.) that can enter and exit a network system in a given period of time. The amount of flow on each arc is usually limited by capacity restrictions, e.g. diameters of pipelines will limit the flow of oil in an oil distribution system. The maximum or upper limit on the flow in an arc is referred to as the flow capacity of the arc. Flow capacities for the nodes are not specified, the only requirement being that for each node (except the origin and destination nodes) the flow balance equation (flow into the node = flow out of the node) must be satisfied.

The formal model formulation for solving the maximal flow problem is provided by Weatherford and Moore (2001) as follows.

Let node 1 be the origin node and node *n* the destination and let x_{ij} denote the flow across the arc (*i*,*j*) connecting node *i* and node *j*. The model is then given as

$$\text{Maximize} \quad f, \tag{2.18}$$

subject to

$$\sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} f \text{ if } i = 1, \\ -f \text{ if } i = n, \\ 0 \text{ otherwise,} \end{cases} \tag{2.19}$$

$$0 \le x_{ij} \le n_{ij}, \qquad \text{for all arcs } (i, j) \text{ in the network,} \tag{2.20}$$

where

n_{ij} denotes the flow capacities on the various arcs.

The following examples from the literature show that the maximal flow problem is an important type of issue that frequently occurs in practice.

Gutierrez-Jarpa *et al*. (2009) made use of the maximal flow technique in a multi-objective model where one of the objectives is to maximize traffic on an existing bus route or railway line while minimizing the cost of the selected route. Stroup and Wollmer (1992) also incorporated the maximal flow technique in a fuel management model for the airline industry where the objective is to devise a minimum cost fuel tankering policy for an airline flight schedule based on fuel prices, station constraints and supplier constraints.

Other examples in the literature where the maximal flow technique was used can be found in the work of Fishman (1987) and Rosenthal (1981).

2.2.7 Minimal spanning tree problem

The minimal spanning tree problem is the last type of network problem that will be mentioned briefly in this introductory overview of these problems.

Ragsdale (2007) defines a minimal spanning tree as follows. Consider a network with *n* nodes: a spanning tree is a set of *n*-1 arcs that connects all the nodes and contains no loops. A minimum spanning tree involves determining the set of arcs that connects all the nodes in a network while minimizing the total length (cost) of the selected arcs.

According to Anderson *et al*. (2009) the minimum spanning tree problem is usually solved by a simple algorithm that employs a basic greedy heuristic. The algorithm is given by them as follows.

Step 1:

- Arbitrarily begin at any node and connect it to the closest node in terms of the criterion being used (e.g. time, cost or distance). The two nodes are referred to as connected nodes, while the remaining ones are referred to as unconnected nodes.

Step 2:

- Identify the unconnected node that is closest to one of the connected nodes. Break ties arbitrarily if two or more nodes qualify as the closest node. Add this new node to the set of connected nodes. Repeat this step until all nodes have been connected.

As with all the other network flow models, the minimal spanning tree problem has many applications in areas such as local area network design, communication services and the like. Some of these applications include the work of Kawatra and Bricker (1998) which focuses on a multi-period model for a capacitated minimal spanning tree problem. The objective of the problem was to minimize the total cost of transmission capacity. The work of Hage *et al.* (1996) deals with the minimum spanning tree problem in archaeology while the work of Jain and Mamer (1988) focused on approximations for a random minimal spanning tree in the design of communication networks.

2.2.8 Network design issues

Following the brief overview of some of the well known network models, it is important to note that certain network problems can be solved by linear programming models and relatively simple algorithms or heuristics such as Dijkstra’s algorithm (Kershenbaum, 1993) can be used for example to determine the shortest route through a network. Another advantage in certain network models is the integer property for specific cases (Weatherford and Moore, 2001). It is well known that linear programming models do not in general yield optimal solutions that have integer-valued solutions for the variables. The integer property is stated by Weatherford and Moore (2001) as follows.

If all the RHS [right hand side] terms and arc capacities are integers and the coefficient matrix of the constraints are unimodular, there will always be an integer-valued optimal solution to the model. The motivation for the above integer property can be found in the unimodularity of the coefficient matrices. Briefly this can be explained as follows.

Consider the following problem

Maximize $\mathbf{Z} = \boldsymbol{cx}$,

subject to

$$\boldsymbol{Ax} = \boldsymbol{b},$$

$$\boldsymbol{x} \geq \mathbf{0},$$

where ***A*** is a *m* x *n* matrix and all the other vectors are of appropriate size.

The matrix ***A*** is said to be unimodular if the determinant of every square sub matrix of ***A*** is 0, +1 or -1. This means that ***A*** is unimodular if and only if every submatrix is unimodular. According to Salkin and Mathur (1989) the following is then true. If the coefficient matrix ***A*** is unimodular and ***b*** is integer-valued then every basic feasible solution and therefore the optimal linear programming solution will be integer. Examples of network models which may have a unimodular coefficient matrix, and thus can be solved by a linear programming model, include classical transportation problems, assignment problems and minimum cost network flow problems.

A good technical exposition of the integer property and unimodularity, as well as examples, can be found in Salkin and Mathur (1989).

Another aspect relevant to the discussion of networks models is network design. The models and examples discussed so far described problems related to existing networks. For example, determine the shortest path, or maximum flow through an existing network. Network design problems usually involve decisions regarding network topology and capacity planning to satisfy certain demands or requirements. A good example of a network design problem can be found in Terblanche (2008) where a study was performed that deals with solving a survivable network design problem by considering uncertainty in traffic requirements.

The network design problem is, to a certain extent of particular interest in this study where a selection from different pipe capacities for chosen links in an oil field has to be made. This section is therefore concluded with a brief overview of general network design issues that need to be considered by a network designer. The discussion is based on the work of Kershenbaum (1993).

Some of the major issues in network design include the following:

Justifying a network

The most basic question is whether a network is justified at all. In certain instances needs may be satisfied with a simple point-to-point connection while other applications may require a more sophisticated network for specific needs.

Scope

The scope of a network is usually bounded by the communication facilities in the network as well as by the type of applications which it interconnects. The geographic scope of a network is another important aspect. In some instances domestic or international networks may be required. The volume of traffic may also have an impact on the scope of a network.

Manageability

A network comprised of many different types of facilities and specialized control procedures may be cost effective but it may be difficult to manage in terms of, for example, constant tuning. Networks that are homogeneous and as simple as possible may be easier to manage.

Network architecture

Network architecture presents a number of issues related to the overall “shape” of the network. Issues such as the type of node or type of link need to be considered. A decision is also necessary on whether the network should be decomposed into subnetworks for the sake of design and operation.

Switching mode

One of the main reasons for building a network, as opposed to giving each application its own dedicated facilities, is to share resources, specifically transmission facilities. There are a number of different ways of doing this, e.g. packet switching, circuit switching, random access etc.

Node placement and sizing

In theory, it is possible to place nodes anywhere. In practice, the placement of nodes is usually limited to a finite set of candidate sites. The selection of network node sites is seen as a fundamental problem and encompasses problems such as determining which sources and destinations should be part of the network, where to place the nodes, type and size of devices etc.

Link topology and sizing

Link topology and sizing involves selecting the specific links interconnecting the nodes. This is where the architecture of the network is determined and also the specific number and types of links

Routing

Routing involves selecting paths for each requirement and involves aspects such as selecting the routing procedure, type of routes, protocol selection etc.

Solving general network models

When modeling aspects that address the above network planning decisions, it is often necessary to model the situation as an integer linear program. These problems are often very difficult to solve and the planner is often forced to be satisfied with approximate solutions. See Terblanche (2008).

Section 2.2 (network flow models) has provided a brief introductory overview of some of the most basic and commonly known network flow models. This section is by no means an exhaustive review as there exist many variations of the models discussed. Furthermore, other network flow models are not discussed here: e.g. network analysis techniques known as CPM and PERT that are primarily used in project management tasks. The next section (section 2.3) will present a brief discussion of knapsack models.

2.3 The knapsack problem and extensions

2.3.1 Introduction

The knapsack or rucksack problem derives its name from an exercise where soldiers had to fill their knapsacks by selecting from a variety of objects that could be included. De Villiers (2004) describes the problem as a scenario where a hiker who carries a knapsack with him must choose objects to fill the bag. Each object he selects has a weight and associated value/ profit. The goal is to choose those objects that will yield the maximum total value/profit, subject to the weight capacity of the knapsack.

This zero-one version of the knapsack problem can be mathematically formulated (Van Der Merwe, 2002) by numbering the possible objects from 1 to *n* and introducing a vector set of binary variables $x_j(j = 1, \dots . n.)$ which is defined as follows.

$$x_j = \begin{cases} 1 \text{ if object } j \text{ is selected,} \\ 0 \text{ otherwise.} \end{cases}$$

Let p_j be the profit (value) assigned to object *j* and w_j the weight of the specific object. Let *c* be the weight capacity of the knapsack. The problem is then formulated as

$$\text{Maximize} \quad \sum_{j=1}^{n} p_j x_j , \tag{2.21}$$

subject to

$$\sum_{j=1}^{n} w_j x_j \leq c. \tag{2.22}$$

$$\text{and} \quad x_j \in \{0,1\} \quad \text{for } j = 1, 2, \dots, n. \tag{2.23}$$

The above formulation represents a specific type of knapsack problem whereas several other types of knapsack problems also exist. The following section will cover some of the common ones.

2.3.2 Types of knapsack models

In this section a brief overview of various types of knapsack models will be given. The discussion is based on the work of Van Der Merwe (2002) and certain sections are quoted from this work without referencing the source again.

2.3.2.1 Zero one knapsack

The 0 – 1 binary knapsack problem has been given in paragraph 2.3.1 above and is applicable where the decisions involve the selection (or not) of specific items for the knapsack. According to Martello and Toth (1990) the 0 – 1 knapsack problem is the most important type of knapsack problem and also one of the most frequently studied discrete programming problems.

2.3.2.2 Bounded knapsack

Knapsack problems where there are certain item types with only a limited number of items available of each type are referred to as bounded knapsack problems. The challenge is to choose the combination of items of each type that maximizes the total profit while a capacity constraint is not violated. Assume that there are *n* types of items and also that,

p_j = the profit of an item type j,

w_j = the weight of an item of type j,

b_j = the upper bound on the number of items of type j, and

c = the weight capacity of the knapsack.

The problem is based on selecting a number x_j $(j = 1, \ldots., n.)$ of items of each type to maximize the total profit. This is formulated as the following integer programming (IP) model:

$$\text{Maximize} \quad Z = \sum_{j=1}^{n} p_j x_j, \tag{2.24}$$

subject to

$$\sum_{j=1}^{n} w_j x_j \le c, \tag{2.25}$$

$$0 \le x_j \le b_j \quad \text{and integer}, \ j = 1, \ldots., n. \tag{2.26}$$

It is usually assumed that:

p_j, w_j, b_j and c are positive integers,

$$\sum_{j=1}^{n} b_j w_j > c, \tag{2.27}$$

$$b_j w_j \le \text{c}, \qquad j = 1, \ldots., n. \tag{2.28}$$

2.3.2.3 The multi-dimensional knapsack problems

Hill and Hiremath (2000) describe a multi-dimensional knapsack problem as a type of knapsack problem where a set of n items are packed in m knapsacks with capacities c_i (i =1, …m). The formulation of the multi-dimensional knapsack problem is as follows.

Assume that

p_j = profit of item j,

w_{ij} = weight associated with item j in knapsack i, and

c_i = capacity of the i-th knapsack,

then the mathematical formulation is given by

$$\text{Maximize} \quad Z = \sum_{j=1}^{n} p_j x_j, \tag{2.29}$$

subject to

$$\sum_{j=1}^{n} w_{ij} x_j \leq c_i, \qquad i = 1, \ldots .m, \tag{2.30}$$

$$x_j \in \{0,1\}. \tag{2.31}$$

2.3.2.4 Stochastic knapsack

According to Carraway *et al*. (1993) stochastic knapsacks occur in situations where the cost associated with each item is known with certainty but the return from including an item is uncertain. In this case returns are modelled as independent, normally distributed random variables. The objective is to maximize the probability that the total return is equal to or exceeds a specified goal value where *n* object classes exist.

Let

c_j = the stochastic return gained by including one item of type j,

w_j = the cost for including one item type j, W = the total cost constrained or the capacity of the knapsack,

C = a specified target that the overall returns should exceed or equal, and

P = probability of return.

The stochastic linear knapsack can now be formulated as follows:

$$\text{Maximize } \mathrm{P}\left[\sum_{j=1}^{n} c_j x_j \geq \mathrm{C}\right], \tag{2.32}$$

$$\text{subject to } \sum_{j=1}^{n} w_j x_j \leq \mathrm{W}, \tag{2.33}$$

$$x_j \in \{0,1,2,\ldots\}, \qquad j = 1,2 \ldots .n. \tag{2.34}$$

2.3.2.5 Multiple–choice knapsack

Van der Merwe (2002) explains multiple-choice knapsack problems in this manner. The multiple-choice knapsack problem is formulated as having a set of *n* items and *m* knapsacks (*m* < *n*),

where

p_j = profit for including item *j*,

w_{ij} = weight associated with item *j* in knapsack i, and

c_i = capacity of the *i*-th knapsack.

The challenge is to select *m* disjoint subsets of items so that the total profit is maximized and that none of the individual knapsack capacities are exceeded. The following is an IP formulation of the problem:

Maximize $\sum_{i=1}^{m}\sum_{j=1}^{n} p_j x_{i\,j}$, (2.35)

subject to

$$\sum_{j=1}^{n} w_j x_{i\,j} \le c_i, \qquad i = 1, \dots . m, \tag{2.36}$$

$$\sum_{i=1}^{m} x_{i\,j} \le 1, \qquad j = 1, \dots . n, \tag{2.37}$$

$$x_{ij} = 0 \text{ or } 1, \qquad i = 1, \dots . m, \qquad j = 1, \dots . n, \tag{2.38}$$

where

$$x_{ij} = \begin{cases} 1 \text{ if item } j \text{ is assigned to knapsack } i, \\ 0 \text{ otherwise.} \end{cases}$$

Assume that the weights w_{ij} are positive integers and also assume the following:

p_j and c_i are positive integers,

$$\sum_{j=1}^{n} w_{ij} > c_i, \qquad i = 1, \dots . m. \tag{2.39}$$

2.3.2.6 Tree Knapsack

The tree knapsack problem is of special interest in this study because it will be explored as an alternative for solving a specific network flow model. A brief overview of a tree knapsack based on the work of Van der Merwe and Hattingh (2006) will be given here.

A tree knapsack problem can be regarded as choosing a subtree of a tree and is described as follows by Van der Merwe and Hattingh (2006). Given an undirected tree T = (V, E) with n nodes rooted at node 0, V = {0, 1 …n-1} is the set of nodes that can be labeled in either depth or breadth first fashion while E denotes the defined edges. Also assumed are the following,

d_i= demand used by including node i in the subtree,

c_i = profit gained by including node i in the subtree,

p_i= the predecessor or parent of node i, and

H = the total capacity of the knapsack.

The task then is to find the subtree T' = (V', E') of T rooted at node 0 such that

$$\sum_{i \in V'} d_i \leq H,$$

and

$$\sum_{i \in V'} c_i$$

is maximized.

Let $$x_j = \begin{cases} 1 \text{ if node } j \text{ is chosen,} \\ 0 \text{ otherwise.} \end{cases}$$

The problem can then be formulated as an integer linear programming problem.

Maximize Z = $$\sum_{j=0}^{n} c_j x_j, \qquad (2.40)$$

subject to

$$x_{p_j} \geq x_j, \qquad j = 1, \ldots.n, \qquad (2.41)$$

$$\sum_{j=0}^{n} d_j x_j \leq H, \qquad (2.42)$$

$$x_j \in \{0,1\}, \qquad (2.43)$$

In the subtree a node can only be included if the parent of the node is also included in the subtree; see (2.41) above. This can also be stated in such a way that if a node *i* is to be included, all the nodes in the unique path between node *i* and the root node 0 must be included – this is referred to as the contiguity assumption (Van der Merwe, 2002). It may be noted that the zero-one knapsack is a special case where the tree consists only of the root node and one level of leaves of the tree.

To understand the contiguity assumption, refer to figure 2.4 below which is a representation of a sample tree with nodes labeled in a breadth first manner. Note that node 8, for example, cannot be included in a subtree without including nodes 0, 1, and 4.

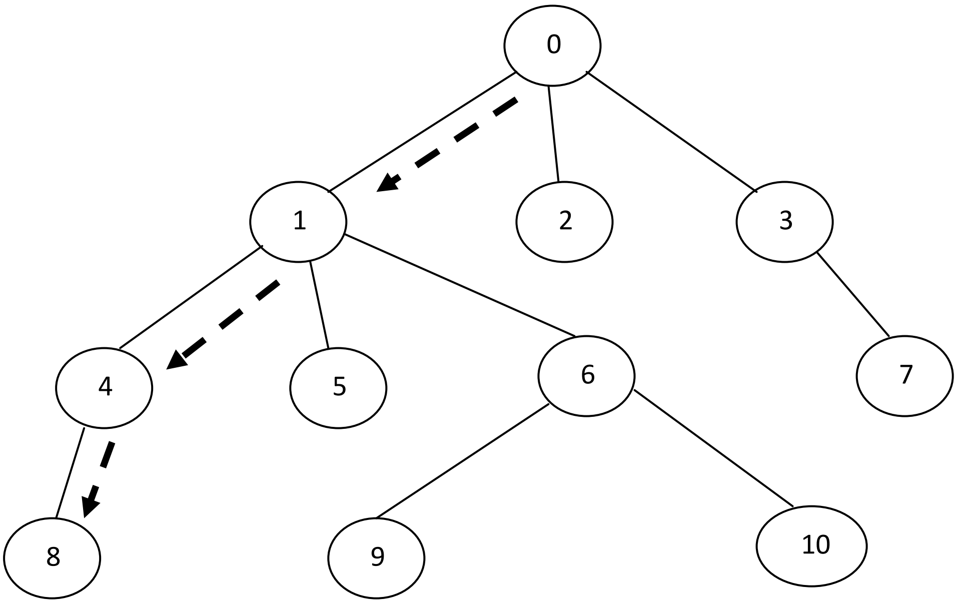


Figure 2.4 Sample tree network

It is clear from figure 2.4 that multiple subtrees may exist connected to node 0. The choice of which subtree to consider depends on the objective of the knapsack problem and is normally based on either maximizing a profit function or minimizing a cost function.

2.3.2.7 Extended tree knapsack

The extended tree knapsack is a more general form of the tree knapsack model discussed in the previous section. The model is discussed in detail in Shaw (1997) and is only briefly introduced here as the same principles will be applied to the case study under investigation in this research project.

In the extended tree knapsack model there is also a cost involved in transmitting y_i units from node i to predecessor p_i, say $f_i(y_i)$ where f_i is an arbitrary function that satisfies the condition that $f_i\,(0) = 0$. Van Der Merwe (2002) defines the model as follows.

Let $T = (V, E)$ be an undirected tree with n nodes labeled in depth-first order rooted at 0 and with $V = \{0, 1,.., n\text{-}1)$. Let $\hat{T} = (V, A)$ be a directed out-tree derived from T, where an out-tree is a tree derived by giving direction to undirected edges in the initial tree. Let the set A be defined as follows, $A = \{(p_i, i) \mid i \in V\}$. Define B as an n x n node-arc incidence matrix of $\hat{T}$, excluding the row corresponding to the root node (node 0). In B, each row corresponds to a node and each column corresponds to an arc. This means that the ith column of B has entries of zero, except in row i and row p_i ($\neq 0$), which have values of respectively 1 and -1. Let y_i be the amount of traffic sent from node i to its parent p_i and let

$$x_i = \begin{cases} 1 \text{ if node } i \text{ is selected to be served,} \\ 0 \text{ otherwise,} \end{cases}$$

let $y = (y_1, y_2, ..., y_n) \in \mathbb{R}^n$. Define the matrix D as, $D = diag(d_j)$. The extended tree knapsack problem can now be formulated as follows, where H is the capacity of the knapsack:

$$\text{Maximize Z} = \sum_{j=0}^{n-1} c_j x_j - \sum_{j=0}^{n-1} f_j(y_j), \tag{2.44}$$

Subject to

$$x_j \leq x_{p_j}, \qquad j = 1, 2, ..., n\text{-}1, \tag{2.45}$$

$$Dx - By = 0, \tag{2.46}$$

$$\sum_{j=1}^{n-1} d_j x_j \leq H, \tag{2.47}$$

$$y \geq 0, \tag{2.48}$$

$$x_j \in \{0,1\}, \qquad j = 0, 2, ..., n\text{-}1, \tag{2.49}$$

where **0** denotes the n-dimensional zero vector.

2.3.3 Knapsack problem applications

A wide variety of problems exhibit the structure of the knapsack problem (Bretthauer and Shetty, 2002). The purpose of this section is to provide only a few examples from the literature where the knapsack approach was used to solve specific real world problems.

The work performed by Bretthauer and Shetty (2002) furnishes a survey of algorithms and applications for nonlinear knapsack problems. They also mentioned the work of other researchers in this area; e.g. the application of the knapsack problem to a manufacturing capacity problem and a health care capacity planning problem. In both cases there are costs to be minimized, subject to an upper limit budget value. Wang and Hu (2010) devised a quadratic knapsack type public-key cryptosystem and showed that, using this approach, a system can be secured against brute-force attacks.

Other recent applications included areas such as mining, resource allocation scheduling and network problems. Moreno *et al.* (2010) discussed an algorithm using a precedence constrained knapsack approach to address an open-pit mine production scheduling problem while Kolliopoulos and Steiner (2007) made use of a partially ordered knapsack approach to address specific scheduling problems.

In the area of resource allocation Vanderster *et al.* (2009) formulates an allocation problem as a variant of the 0-1 multi-choice multidimensional knapsack problem while Melachrinoudis and Kozanidis (2002) described a mixed integer knapsack model for allocating funds to highway safety improvements.

Knapsack approaches are also often considered for network problems and Song *et al.* (2008) described the use of a multiple multidimensional knapsack problem applied to cognitive radio networks. The work of Shaw and Cho (1996) and Shaw *et al.* (1997) focused on the use of a tree knapsack problem as part of designing a local access telecommunication network. This work was further considered by Van der Merwe (2007), Van der Merwe and Hattingh (2006) and Van der Merwe and Hattingh (2010).

Other examples of knapsack problems and applications in the literature can be found in Patterson and Rolland (2002) where a knapsack approach is used in the hybrid fiber coaxial network design while Kuah and Perl (1989) focused on a feeder bus network design problem using a knapsack approach. More examples can be found in Ceri *et al*. (1982), Morita *et al.* (1989) and Jang and Wang (1993).

2.4 Conclusion

Chapter 2 supplied an introductory overview of some network and knapsack models with extensions. Aspects covered included a review of the basic and familiar types of network models, a discussion of the various types of knapsack models and examples from the literature.

The next chapter will offer an overview of the pipeline design problem that was chosen as a case study for this research project.

3

An overview of the oil pipeline design problem

3.1 Introduction

In order to investigate the feasibility of solving a network design problem using a tree knapsack approach, it was decided to select a specific case study from the literature that could be used as a basis for the research project. The case study selected describes the optimal oil pipeline design for the South Gabon oil field (Brimberg *et al*., 2003). The aim of this chapter is therefore to furnish a brief description of the chosen case study as well as the models and solution suggested in the case study. The complete discussion in this chapter is based on the work of Brimberg *et al*. (2003) and some of the sections are quoted from this source without referencing it again.

3.2 The oil pipeline design problem

In this case study Brimberg *et al.* (2003) explore a specific real world problem.

The project considers a set of offshore platforms and onshore wells, each producing a known or estimated amount of oil that needs to be connected to a port. These connections may take place directly between platforms, well sites and the port, or may go through connection points at given locations. The objective of the pipeline system is to try and reduce the cost of transporting oil to a specific port in order to allow for expansion of production to enable increased profitability – this implies that the configuration of the network and sizes of pipes must be chosen to minimize construction cost. Figure 3.1 is a representation of the South Gabon oil field network.

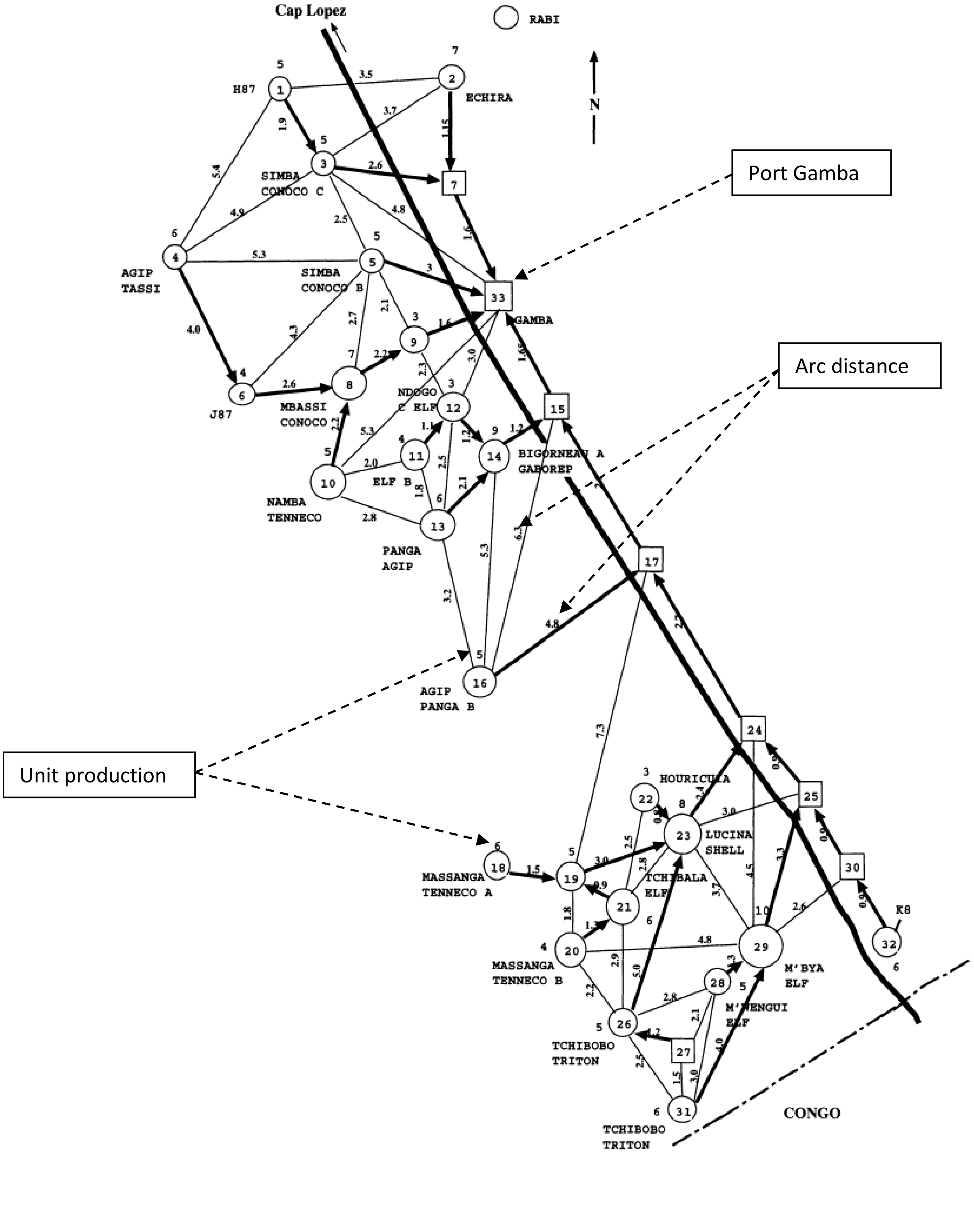


Figure 3.1 South Gabon oil field (Brimberg *et al.*, 2003)

From figure 3.1 it is evident that the South Gabon oil field network consists of 33 nodes. These represent the offshore platforms, onshore wells (both represented by circles in figure 3.1), seven connection points (represented by squares in figure 3.1) and one port called Gamba (node 33). The number inside each circle and square identifies the node while the numbers adjacent to the circles are the production rates at those sites. There are 129 potential arcs and the numbers on the arcs indicate the distance between the nodes. All the oil production in this region is transported to Gamba, from where it is then exported by sea.

3.3 The model suggested by Brimberg et al. (2003)

The pipeline design problem was formulated by Brimberg *et al*. (2003) as a mixed integer program. The flow of oil in the pipeline system was modelled by a network (*N*, *A*) with node set *N* and arc set *A*.

The node set $N = \{i \mid i = 1, 2, ..., n\}$ corresponds to the wells, i.e. both those on offshore platforms and at onshore sites, as well as to potential connection points between pipeline segments and to the port.

The arc set *A* corresponds to potential layouts of pipeline segments between offshore platforms, onshore production sites, connection points and the port. Arcs are assumed to be oriented, but, in some of the arcs, flow may be sent either from *i* to *j* or from *j* to *i*. This means that there are two potential directions for the flow, only one of which will be chosen in the optimal solution.

A set of pipe diameters is associated with each arc (*i*, *j*) of *A* and it is assumed that the pipe capacity is fixed once the diameter is fixed. The capacities are chosen among a given set which may vary with the pair of nodes *i* and *j* – not more than 5 or 6 capacities were considered by Brimberg *et al*. (2003).

The mathematical model of the pipeline design problem considered by Brimberg *et al*. (2003) was formulated as follows:

Let the flow in arc (i, j) be denoted by $f_{\mathrm{ij}} \geq 0$, for all $(i, j) \in A$ and let $y_{ij}^{k} = 1$ if a pipe with the k^{th} capacity is placed between nodes i and j, and 0 otherwise, for all (i, j) and k values. The model formulation is represented by

$$\text{Minimize} \sum_{(i,j)\in A} \sum_{k} E_{ij}^{k} y_{ij}^{k}, \qquad (3.1)$$

subject to

$$\sum_{j\in N} \sum_{k} y_{ij}^{k} = 1, \qquad \text{for all } i \text{ except the port node ,} \qquad (3.2)$$

$$\sum_{j\in N} f_{ji} + p_i = \sum_{j\in N} f_{ij}, \qquad \text{for all } i \text{ except the port node,} \qquad (3.3)$$

$$f_{ij} \leq \sum_{k} C_{ij}^{k} y_{ij}^{k}, \qquad \forall\ (i, j) \in A\ , \qquad (3.4)$$

$$f_{\mathrm{ij}} \geq 0, \qquad \forall\ (i, j) \in A, \qquad (3.5)$$

$$y_{ij}^{k} \in \{0, 1\}, \qquad \forall\ (i, j) \in A, \text{ and all } k\text{-values.} \qquad (3.6)$$

The objective function (3.1) is the sum of costs $\mathrm{E}_{\mathrm{ij}}^{\mathrm{k}}$ for all pipes. The first set of constraints (3.2) consists of multiple-choice constraints while constraints (3.3) express the conservation of flow; i.e. the total flow entering node i plus the flow due to production (p_i) at the node should equal the leaving flow. Constraints (3.4) express limitations on flow due to pipe capacity ($\mathrm{C}_{\mathrm{ij}}^{\mathrm{k}}$) while constraints (3.5) ensure that flows are non negative. Constraints (3.6) indicate that pipes are set up entirely or not at all.

3.4 Solution methods

Two heuristics (tabu search and variable neighborhood search) and an exact algorithm (branch and bound) were proposed by Brimberg *et al.* (2003) to solve the problem suggested in section 3.3. The two heuristics are used to obtain an upper bound for the branch and bound procedure while lower bounds are obtained by means of linear relaxation as well as through the use of two new types of inequalities suggested by these authors. A complete step by step implementation of the two heuristics as well as the description of the new inequalities can be found in Brimberg *et al.* (2003).

Another approach suggested by Brimberg *et al.* (2003) is the possible decomposition of the problem according to geographical considerations. They describe and motivate this decomposition as follows. Pipeline design problems possess a particular geometric structure which may sometimes be exploited to simplify their solution. For instance, reservoirs may be geographically dispersed, which induces some natural decomposition. If the network (N, A) has a cut vertex, i.e., a vertex j the suppression of which disconnects the network, the problem can be solved for the subnetwork(s) so obtained and not containing the port, considering vertex j as a port. Then a smaller problem is obtained by deleting these subnetworks except node j and adding their production to that at j. A less powerful decomposition scheme may be used in the case where (N, A) has a small disconnecting set of nodes, say $\{i, j\}$; then the subproblem corresponding to the different distribution of flow at i and j must be considered.

The third approach to solving the problem involves the branch and bound procedure. An interactive approach was suggested by the researchers in order to exploit the geographic structure of the matter at hand. In brief, the proposed interactive method entails decomposing the problem and applying the tabu search and the variable neighbourhood search heuristics to the decomposed problem. Branching and the control of bounds are then handled by the heuristics results as well as the newly suggested inequality mentioned earlier. A complete description of the different steps in the proposed interactive branch and bound procedure can be found in Brimberg *et al.* (2003).

3.5 Model results

Different pipe capacities were considered and (as it was done in Brimberg *et al.* (2003)), the total cost of a section of pipe was obtained by multiplying the arc distance by the unit price for each pipe capacity. Applying the heuristics to the problem, it was found that both the tabu as well as the variable neighborhood search yielded a heuristic solution value for the objective of 1423; this solution is also shown in figure 3.1 (bold printed arcs).

The oil field network was then decomposed into two subnetworks, a northern and southern part with node 17 as the articulation or junction point. See figure 3.2 (at the end of the chapter) for a graphical representation of the decomposed network. The problem involving the southern subnetwork has nodes 18 to 32 with node 17 (the articulation point) being a port. Solving this subproblem using CPLEX 7.0, an optimal value of 672 was obtained.

The second problem, which is the northern subnetwork, consists of nodes 1 to 17 with node 33 as a port. The connection point 17, being the articulation point of the northern and southern subnetworks, was given a total production equal to the sum of all productions in the southern subproblem. Applying the suggested inequalities to control the bounds of the problem, it was found that the same value as the heuristic solution was obtained, or in some cases larger or infeasible solutions were generated. This led to the conclusion of Brimberg *et al.* (2003), that the heuristics solution must be optimal for the case under consideration.

3.6 Conclusion

This chapter provided some brief background information on the case study selected as a basis for the research project. The discussion focuses on the design problem, the suggested model and possible solution methods. A complete technical discussion which is beyond the scope of this research report can be found in Brimberg *et al.* (2003).

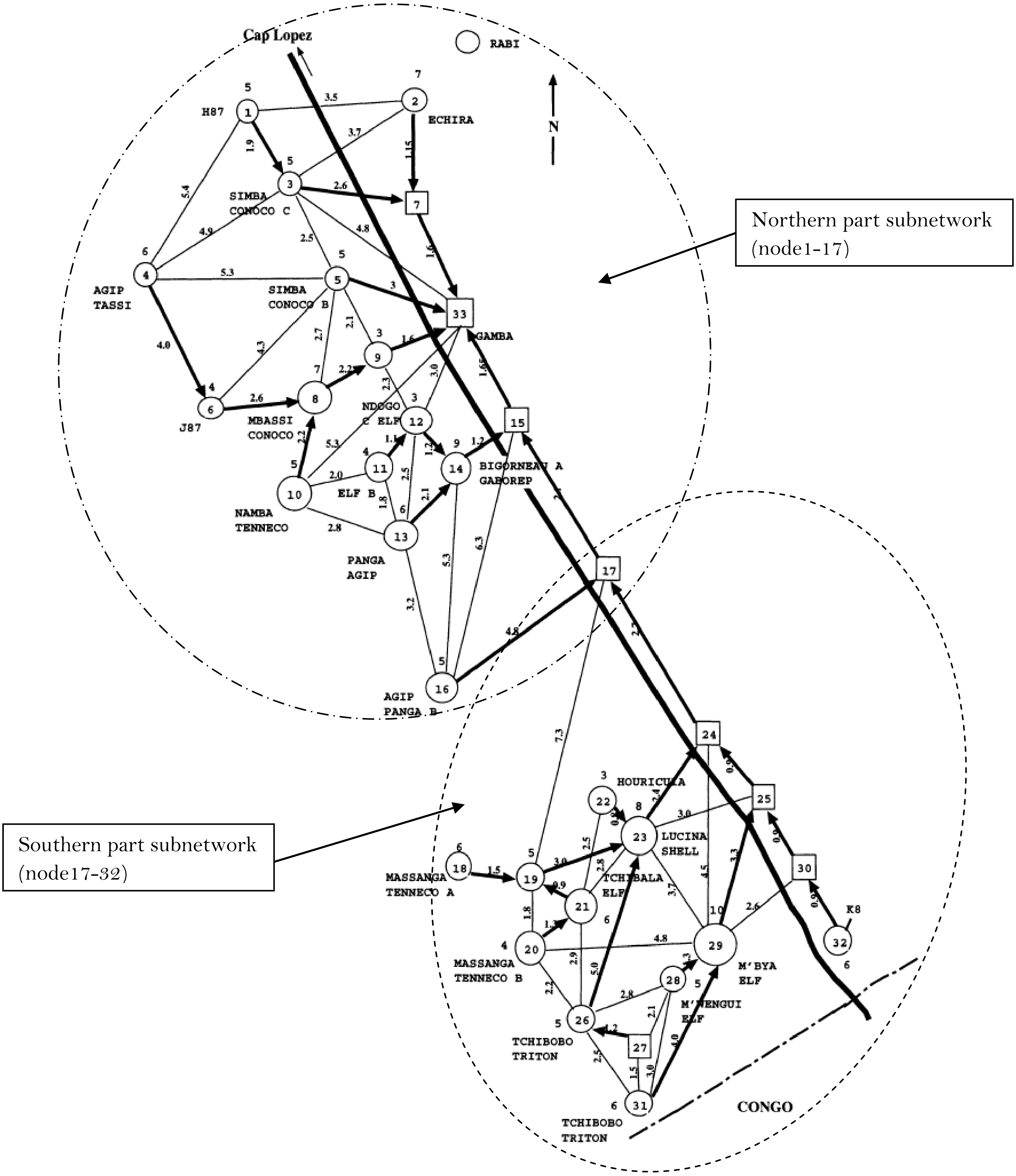


Figure 3.2 Subnetworks of South Gabon oil field

4

Model development for the oil pipeline design problem

4.1 Introduction

This chapter concentrates on the research design and methodology followed to develop a tree knapsack model that can be used to test the feasibility of solving the pipeline design in an alternative way. The tree knapsack model described in this chapter is based on the extended tree knapsack problem considered by Van der Merwe (2007) and explained in chapter 2 (section 2.3.2.7). The next chapter will thereafter focus on the application of the model to the pipeline design problem and the results obtained.

4.2 Methodology and model development

The methodology followed in this study comprises two main steps. In the first, the network representation of the pipeline design problem was converted into a tree structure to facilitate the use of a tree knapsack method as a solution. Second, a mathematical programming model based on an extended tree knapsack model was then formulated and solved in order to be able to express an opinion on the feasibility of the proposed methodology. The following two sections (section 4.2.1 and 4.2.2) will describe the details of the two steps.

4.2.1 Converting the pipeline network into a tree network structure

Prior to model development and applications, the South Gabon oil field network (see chapter 3) had to be converted into a tree structure. This process involves a series of steps i.e. identification of the root node, creating adjacent node lists for each node in the network, and finally building a tree network structure by creating paths based on the adjacency list. To illustrate this process, consider the following small network in figure 4.1.

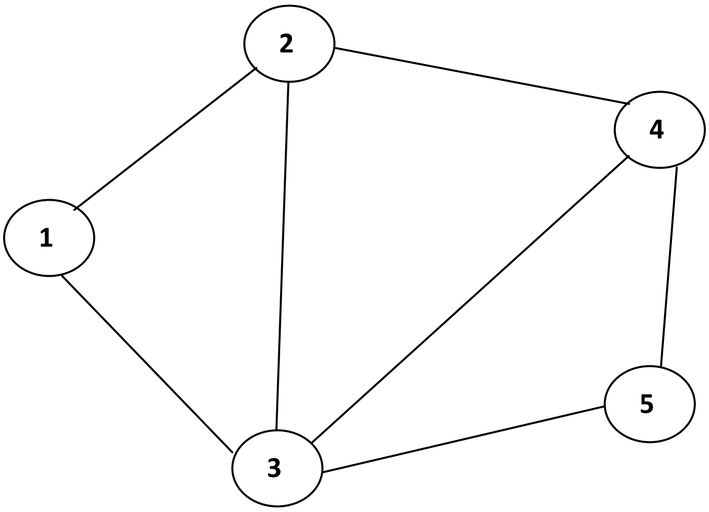


Figure 4.1 Illustrative network

Identifying the root node

Suppose that in figure 4.1 the root node of the network is node 5 and assume that it is the final destination or sink node.

Creating adjacent nodes list

An adjacent node list for a specific node is a set of nodes that are directly connected to that specific node. From figure 4.1 the following adjacency node lists can be generated.

Node	*Adjacency list*
5	{3, 4}
4	{2, 3, 5}

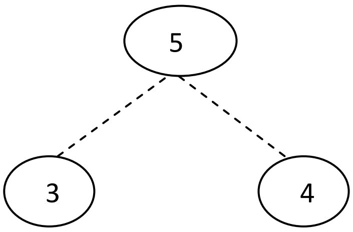
3	{1, 2, 4, 5}
2	{1,3, 4}
1	{2, 3}

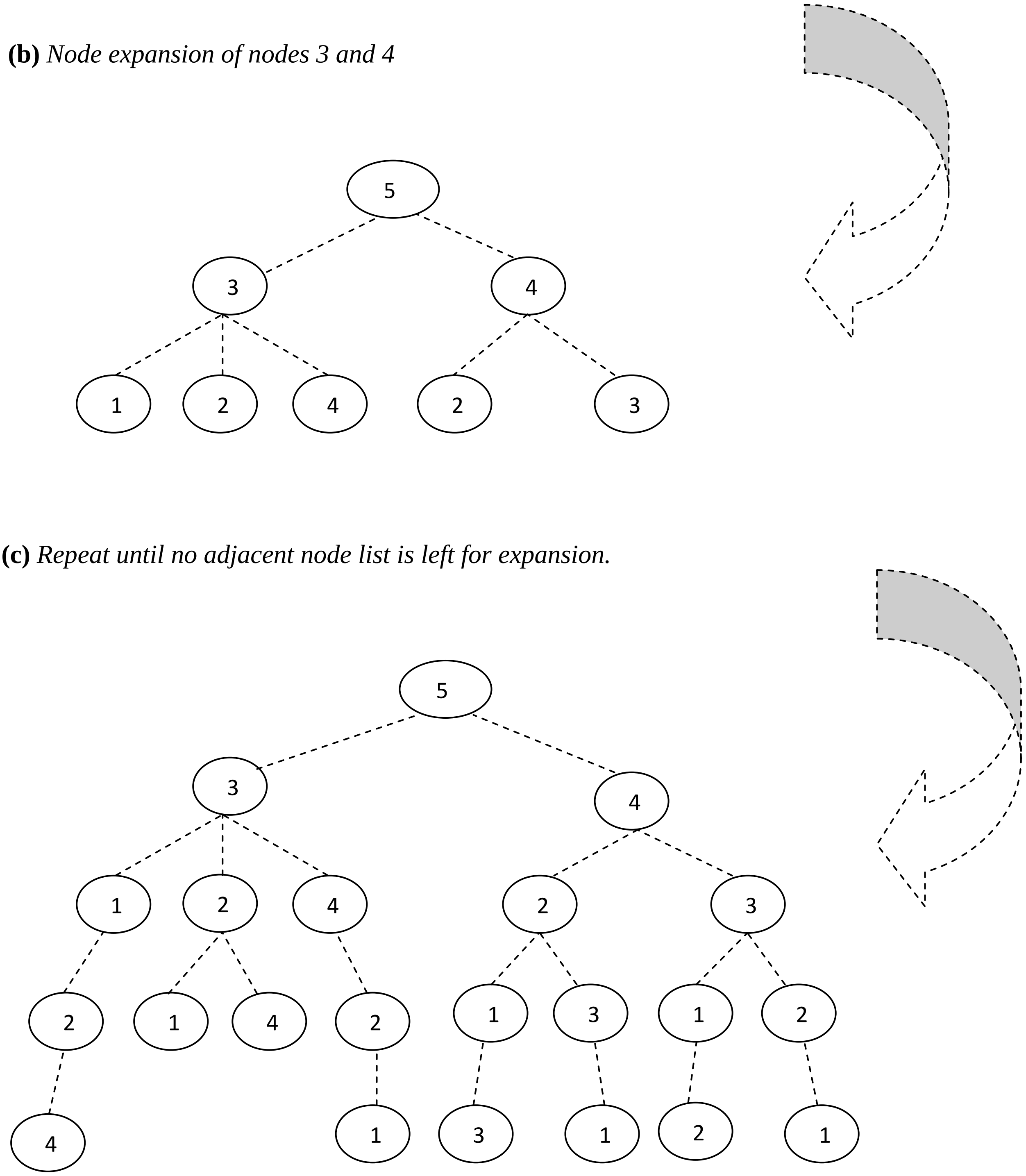
Building the tree network

Once the root node has been identified and all the appropriate adjacency node lists have been constructed, the tree can be completed. This is done as follows:

- First, all the adjacency nodes of the root node are added to the tree (see figure 4.2 (a)).
- Next, the child nodes of the root node are now expanded and added to the tree (see figure 4.2 (b)).
- Subsequently, the child nodes of nodes added in the previous step are expanded and added to the tree – nodes already included in a specific path must not be added to the tree again (see figure 4.2 (c)). Continue this process until there are no nodes left to expand in the adjacency node list.

(a) *Adjacency nodes of the root node are expanded and added to the tree*





(b) *Node expansion of nodes 3 and 4*

(c) *Repeat until no adjacent node list is left for expansion.*

Figure 4.2 Converting a network into a tree network

4.2.2 Model development

The next step in the proposed methodology is to formulate a mathematical programming model which is based on an extended tree knapsack model, and which will be used to solve the tree structure constructed in the first step.

4.2.2.1 The objective function

The objective function in the mathematical model suggested by Brimberg *et al*, (2003) was given in section 3.3 (Chapter 3) as

Minimize $\sum_{i,j \in A} \sum_{k} E_{ij}^{k} y_{ij}^{k}$,

where

A is the set of arcs (i,j),

E_{ij}^{k} = Cost to place a pipe of capacity k between nodes i and j,

$$y_{ij}^{k} = \begin{cases} 1 \text{ if a pipe with capacity } k \text{ is placed between nodes } i \text{ and } j, \\ 0 \text{ otherwise.} \end{cases}$$

The cost in the above objective function was based on the different pipe capacities i.e. a certain f_{ij}^{1} flow would require a cost of E_{ij}^{1} (capacity k = 1). If the flow increases above capacity 1, a new capacity (k = 2) pipe is required. This implies that for flow f_{ij}^{2} a cost of E_{ij}^{2} is required. This process then repeats itself every time flow rises above current capacity. Figure 4.3 shows a graphical representation of this cost function.

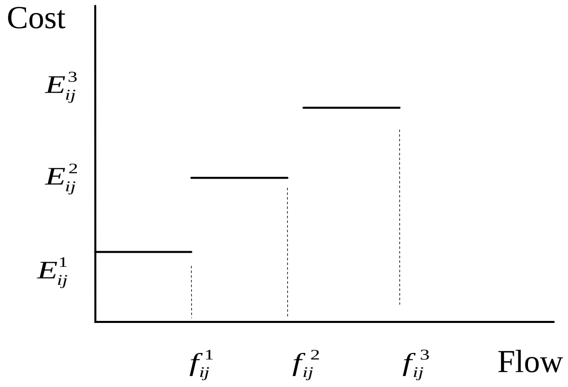


Figure 4.3 Graphical representation of cost function in Brimberg *et al* (2003)

To be able to make use of the extended tree knapsack approach, detailed in section 2.3.2.7 in Chapter 2, it was decided to make use of a fixed charges cost model as opposed to the cost model described above. This cost model can be explained as follows.

Assume that a flow of f_{ij} is sent from node *i* to node *j* (the predecessor of node *i*). Before any flow can be sent on a link, certain fixed costs need to be paid. Whenever the flow increases, a variable cost, a_{ij}, is incurred. If the fixed cost is represented by E_{ij} then the fixed charges cost model can graphically be explained as follows in figure 4.4

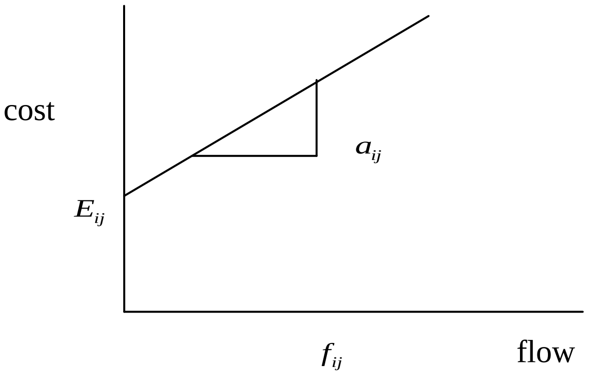


Figure 4.4 Graphical representation of a fixed charges cost model

The slope of the graph represents the variable cost. In the next chapter, chapter 5, a detailed description of how the specific a_{ij} values were determined will be given.

A more detailed description of this cost model can be found in Van der Merwe (2007) and Shaw (1997).

The objective function for the oil pipeline design problem in this research project was then formulated as follows.

$$\text{Minimize} \qquad \sum_{(i,j)} E_{ij}\delta_{ij} + \sum_{(i,j)} a_{ij} f_{ij}, \tag{4.1}$$

where

E_{ij} is the fixed cost associated with each arc (*i*,*j*),

a_{ij} is the cost incurred of a unit of the flow f_{ij} (the slope of the straight line in the fixed charge model)

f_{ij} represents the flow between node *i* and its predecessor node *j*. This flow comprises of two parts such that $f_{ij} = f_{ij_1} + f_{ij_2}$ where f_{ij_1} represents the flow which is less than ε_{ij} and f_{ij_2} represents the flow greater than ε_{ij} (a small positive number).

$$\delta_{ij} = \begin{cases} 1 \text{ if arc } i, j \text{ in the tree is chosen,} \\ 0 \text{ otherwise.} \end{cases}$$

The above objective function is interpreted as the minimization of the sum of costs of selected pipe links plus a variable cost determined by the flow above capacity level. For more detail see Van der Merwe (2007).

4.2.2.2 Model constraints

Several different constraints form part of the model; each set of constraints will be discussed in this section.

Contiguity constraints

The tree knapsack model was introduced in Chapter 2 (section 2.3.2.6). In that section the contiguity assumption was explained as follows. If node *j* is included in a tree all the nodes on the unique path between node *j* and the root node must also be included. This is accomplished by adding the following constraint.

$$x_j - x_{p_j} \leq 0 \qquad j = 1, 2, \ldots, n\text{-}1, \qquad (4.2)$$

where p_j is the predecessor or parent of node *j* and x_j is an indicator with a value of 1 if node *j* is selected and zero otherwise.

With respect to the oil pipeline network, assume that the tree structure in figure 4.5 below represents a flow from the root node (node 0) to nodes 1, 2, 3 and 4. The above constraint will now ensure that flow from the root node to a specific node will pass through that node's predecessors before it reaches its final destination. For example, flow from node 0 to node 4 will pass through node 2 (the predecessor of node 4) before reaching node 4. These same considerations will also ensure a path for flow from any node to the root node.

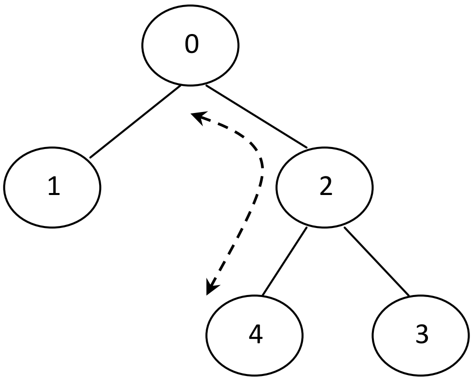


Figure 4.5 Example of the contiguity assumption constraint instance

Node selection constraints

A set of constraints is needed to ensure that nodes in the tree structure are chosen only once and thus avoid duplicate paths. To achieve this, a 0 – 1 variable is used as follows.

Suppose x_i = 0 or 1 for i = 1, 2,…..,n. The constraint $x_1 + x_2 + \ldots\ldots x_n = k$ implies that exactly k alternatives of n possibilities can be selected.

This general logical condition constraint was introduced in the following manner to the pipeline network problem.

Consider a five node network that was expanded into a nine node tree. (See figure 4.6 below.) Since the original number of nodes in the network has increased to nine in the tree, it is clear that duplicate nodes on some of the paths in the tree must exist. To overcome this problem, index numbers are allocated to the nodes in the tree; these numbers are then used to ensure that duplicated nodes are selected only once. Each original node thus has a corresponding set of index numbers.

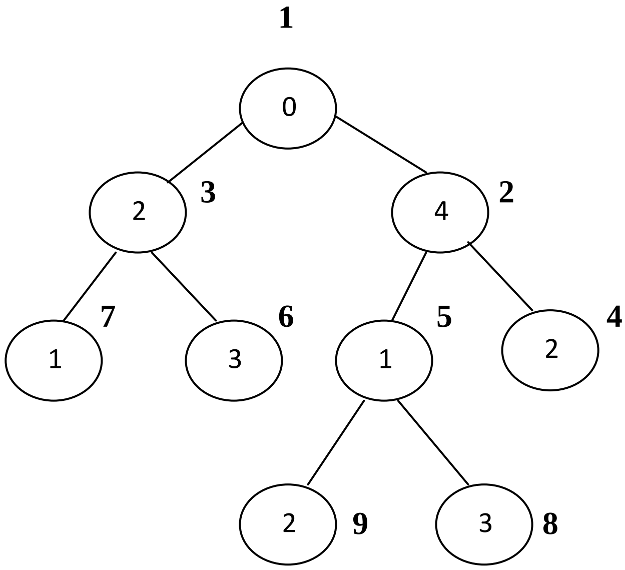


Figure 4.6 An indexed tree network

Original node 1 in figure 4.6 has for example the set of index numbers {5, 7} associated with it. In the case of the example depicted in figure 4.6, a set of original nodes was constructed: say S_0, S_1, S_2, S_3 and S_4. Each one of these sets consists of different indexed nodes. For example,

$S_0 = \{1\}$,

$S_1 = \{5, 7\}$,

$S_2 = \{3, 4, 9\}$,

$S_3 = \{6, 8\}$,

$S_4 = \{2\}$.

If $$x_j = \begin{cases} 1 \text{ if node } j \text{ is selected,} \\ 0 \text{ otherwise,} \end{cases} \qquad j = 1, 2, 3, \ldots\ldots 9,$$

then the summation of a specific indexed node set equal to one, will ensure that the node will only be selected once. For example, to ensure that the original node number 2 is selected only once, the following constraints related to S_2 can be added, i.e.

$x_3 + x_4 + x_9 = 1$.

The node selection constraint for the model is then formulated as follows.

$$\sum_{i \in S_j} x_i = 1, \qquad j = 1, 2, 3, \ldots\ldots n, \qquad S_j = \{\, i \mid N(i) = j\}, \tag{4.3}$$

where

i is the index number allocated to the original node number *N*(*i*).

Flow balance constraints

To ensure flow balance in the tree network, a set of flow balance constraints is necessary. The said constraints are of the form

total flow out of node *j* = total flow into node *j* + production at node *j*.

The following set of constraints for the pipeline design problem will ensure flow balance in the tree network.

$$Dx - B(f_1 + f_2) = 0, \tag{4.4}$$

where

- *D* is a diagonal matrix with diagonals d_j giving the production at node *j* and *B* is a node-arc incidence matrix.
- In the vectors f_1 and f_2, f_{ij_1} represents the flow below pipe capacity and f_{ij_2} represents the flow above existing pipe capacity. If no pipeline exists between *i* and *j* both these flows must be zero. In the case of a network expansion problem one may visualize f_{ij_1} as a flow that is available at zero cost (up to the capacity). If no pipeline exists, this capacity is zero.

B and *D* are two matrices used to create the set of constraints and can be explained as follows. Consider the indexed tree network with a production for each node in figure 4.7 below. Production at the nodes is indicated in the square boxes next to each node, except the root node which has a production of zero.

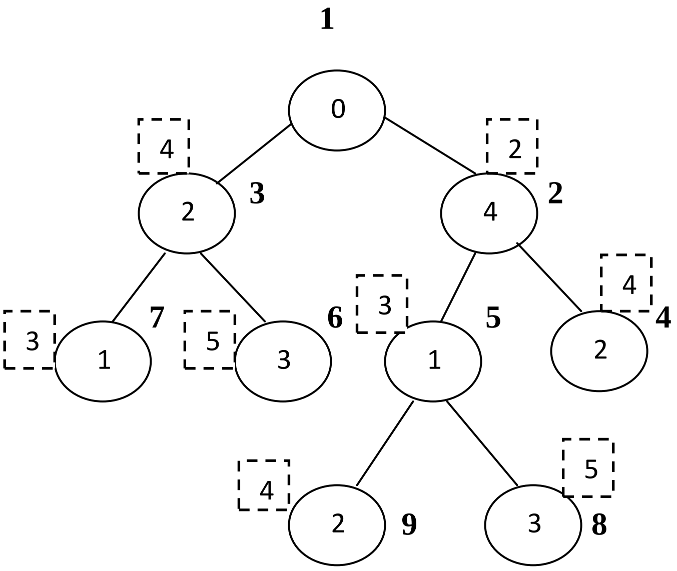


Figure 4.7 An indexed tree network with production at nodes

A diagonal matrix *D*, can now be constructed showing the production at each node. See figure 4.8 below.

	1	2	3	4	5	6	7	8	9
1	-30								
2		2							
3			4						
4				4					
5					3				
6						5			
7							3		
8								5	
9									4

Production at node 9

Figure 4.8 Diagonal matrix with production

Matrix *B* is called a node – arc incidence matrix and exhibits the following structure. See figure 4.9.

	2-1	3-1	4-2	5-2	6-3	7-3	8-5	9-5
1	-1	-1						
2	1		-1	-1				
3		1			-1	-1		
4			1					
5				1			-1	-1
6					1			
7						1		
8							1	
9								1

Figure 4.9 Node-arc incidence matrix

The incidence matrix not only shows nodes linked to other nodes but also the node at which flow originates and where it terminates. For example in figure 4.9 above, flow through arc 2-1 originates at node 2, hence a 1, and terminates at node 1 (the root node) indicated by a -1.

The two matrices are combined in the form $Dx - B(f_1 + f_2) = 0$ to construct the necessary flow constraints.

When looking at row 2 (node 2) and row 5 (node 5) for example, the following flow constraints can be derived.

- Node 2 flow constraint: $2x_2 - (f_{2\text{-}1,1} + f_{2\text{-}1,2}) + (f_{4\text{-}2,1} + f_{4\text{-}2,2}) + (f_{5\text{-}2,1} + f_{5\text{-}2,2}) = 0$.

- Node 5 flow constraint: $3x_5 - (f_{5\text{-}2,1} + f_{5\text{-}2,2}) + (f_{8\text{-}5,1} + f_{8\text{-}5,2}) + (f_{9\text{-}5,1} + f_{9\text{-}5,2}) = 0$.

This implies that the flow generated and sent from node 5 or 2 to its parent should be equal to the flow generated in itself, plus the flow being sent from its child nodes – in this manner the flow balance is ensured in the tree network.

Production constraints

A set of constraints to ensure that flow generated at nodes does not exceed the total capacity of the tree network is also necessary. In the example above, the total production at all nodes is 30 units (from the diagonal matrix *D*) and therefore total production of flow at the nodes should not exceed 30. For the pipeline design model, the constraints are formulated as

$$f_{ij_1} + f_{ij_2} \leq \delta_{ij} C, \tag{4.5}$$

where *C* is the total capacity at the root of the tree network and $\delta_{ij} \in \{0,1\}$.

Flow bound constraints

There are two sets of flow bounds incorporated in the oil pipeline design problem model. Flow bound constraints are necessary to ensure that flow remains within capacity. For the oil pipeline design problem these constraints are formulated as

$$0 \leq f_{ij_1} \leq P_{ij}\,, \tag{4.6}$$

where P_{ij} is the existing pipe capacity on arc (*i*,*j*) and f_{ij_1} is the flow less than the capacity. In the case where no pipeline exists, P_{ij} may be taken as a very small positive number (close to zero).

A second set of constraints for the flow greater than capacity is formulated as

$$f_{ij_2} \geq 0. \tag{4.7}$$

Binary constraints

The following binary constraints are needed.

For arc selection:

$$\delta_{ij} = \begin{cases} 1 \text{ if arc } i, j \text{ is selected,} \\ 0 \text{ otherwise,} \end{cases} \qquad \text{for all arcs.} \tag{4.8}$$

For node selection:

$$x_i = \begin{cases} 1 \text{ if node } i \text{ is chosen,} \\ 0 \text{ otherwise,} \end{cases} \qquad i = 1,2,3,\ldots n. \tag{4.9}$$

4.2.2.3 The complete model

Following the discussion and explanation presented in sections 4.2.2.1 and 4.2.2.2, the complete model for the pipeline design problem can be formulated as follows.

$$\text{Minimize} \quad \sum_{(i,j)} E_{ij}\delta_{ij} + \sum_{(i,j)} a_{ij}(f_{ij_1} + f_{ij_2}) \tag{4.10}$$

subject to

$$x_j - x_{p_j} \le 0, \qquad \text{for all } j, \tag{4.11}$$

$$\sum_{i \in S_j} x_i = 1, \qquad j = 1, 2, 3,..n, \qquad S_j = \{\, i \mid N(i) = j\}, \tag{4.12}$$

$$Dx \text{ -} B(f_1 + f_2) = 0, \tag{4.13}$$

$$f_{ij_1} + f_{ij_2} \le \delta_{ij} C \tag{4.14}$$

$$0 \le f_{ij_1} \le P_{ij} \qquad \text{for all arcs } (i,j), \tag{4.15}$$

$$f_{ij_2} \geq 0, \qquad \text{for all arcs } (i,j), \tag{4.16}$$

$$\delta_{ij} = \begin{cases} 1 \text{ if arc } i, j \text{ is selected,} \\ 0 \text{ otherwise,} \end{cases} \qquad \text{for all arcs } (i,j). \tag{4.17}$$

$$x_i = \begin{cases} 1 \text{ if node } i \text{ is chosen,} \\ 0 \text{ otherwise,} \end{cases} \qquad \text{for all nodes } i, \tag{4.18}$$

where

E_{ij} is the fixed cost for arc (*i*,*j*) and a_{ij} is a variable cost of flow.

4.3 Conclusion

Chapter 4 offered an overview of the research design and methodology followed to develop a tree knapsack model for the oil pipeline problem. A description of how to convert the pipeline network into an extended tree knapsack structure, as well as a formulation of the mathematical programming model to solve the issue, was given. The next chapter will describe the results of the model applied to the pipeline design problem.

5

Empirical experiment and results

5.1 Introduction

In the previous chapter an overview of the research design and methodology followed in this research, was provided. This chapter is devoted to the application of the model to the oil pipeline problem (Brimberg *et al*., 2003). The empirical experiment as well as the results will be described. The chapter will then be concluded with a discussion of the results obtained.

5.2 Empirical experiment

As explained in chapter 1, a specific case study was selected from the literature to investigate the feasibility of solving a network flow model using a tree knapsack approach. It will be recalled that the case study selected and discussed in chapter 3 is an oil pipeline design for the South Gabon oil field (Brimberg *et al*., 2003). The purpose of this section is to describe how the proposed model (described in chapter 4) was applied to this specific case. For easy reference and completeness' sake, the oil pipeline network is again illustrated below in figure 5.1.

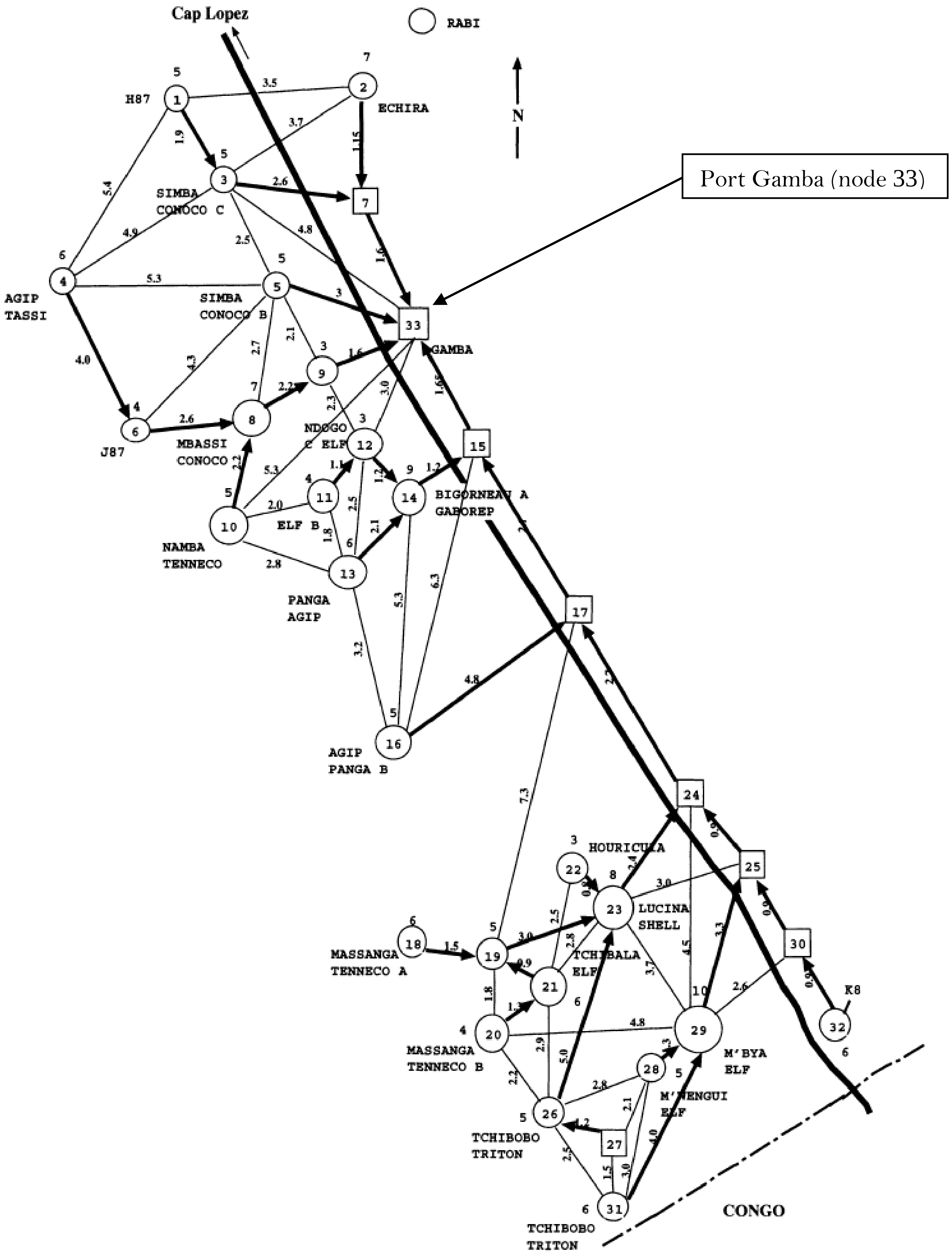


Figure 5.1 South Gabon oil field network

5.2.1 Data used

The following basic data required by the proposed model can be obtained directly from figure 5.1.

- The oil field consists of 33 nodes with 129 possible arcs.

- The root node (in a tree structure) is node 33 which is a port called Gamba.

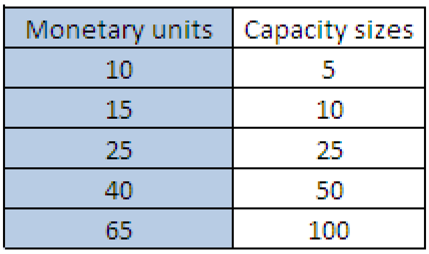
- The distance between each pair of connected nodes is indicated by the numbers on the arcs, e.g. the distance from node 1 to node 2 is 3.5. These distances are the pipe lengths required to connect the platforms and wells (see appendix A for a complete list of pipe lengths).

- The oil production at each site (node) is indicated by the number adjacent to each node, e.g. the production at node 1 is 5 (see appendix B for a complete list of production at each site).

It is unclear from the article that describes the case study what the units of measure were for arc distances and oil production. For the purpose of this study the term “units” will be used for both arc distances and oil production/flow: e.g. the distance from node 1 to node 2 is 3.5 *units* and the oil production at node 1 is 5 *units*.

In addition to the data mentioned above, there was also a cost associated with each pipe connecting the different nodes. The total cost of a section pipe is obtained by multiplying the arc length by the unit price for each pipe capacity. The following monetary units and corresponding pipe capacities were given for the oil field.

Table 5-1: Monetary units and pipe capacities.



Monetary units	Capacity sizes
10	5
15	10
25	25
40	50
65	100

5.2.2 Tree generation

In order to set up and solve the tree knapsack model, the network was first converted into a tree structure as explained in chapter 4. Using node 33 (the port of Gamba) as the root node and following the steps detailed in section 4.2.1 of chapter 4, the South Gabon oil network was converted into a tree network. It should be noted that the representation assumes that there will not be flow splitting or loops in the network.

A program was written in C++ to perform the tree generation. Appendix C contains the pseudo code for the tree generation procedure while the complete program can be found on the CD at the back of the dissertation.

The result of this step was an indexed tree consisting of 7030 nodes and 4183 paths. Figure 5.2 below represents a very small abstract of the generated tree for illustrative purposes. Appendix D provides a small sample list of the nodes generated from the program. A complete list can be found on the CD attached to this dissertation.

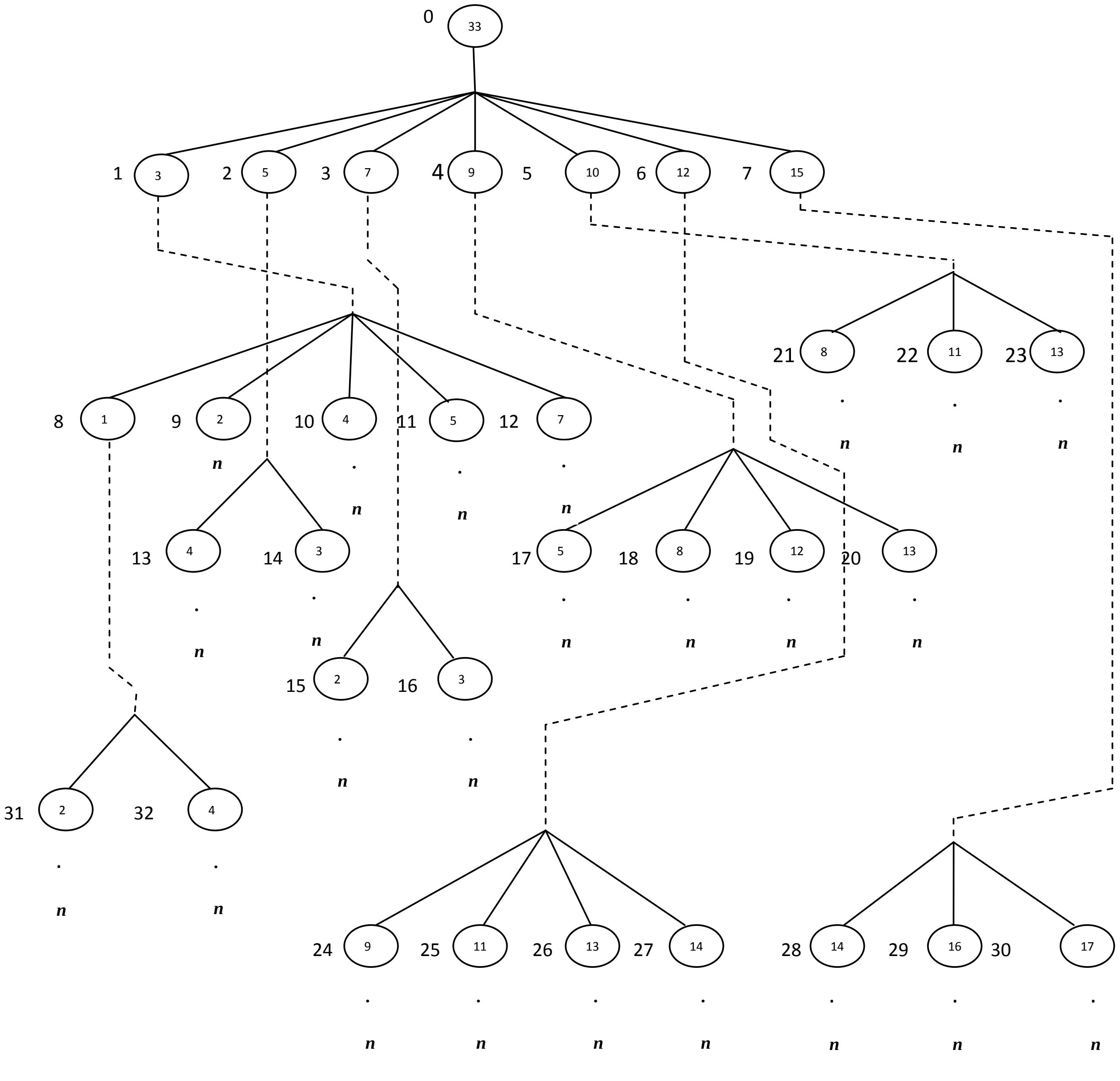


Figure 5.2 Abstract of tree representation

5.2.3 The extended tree knapsack model

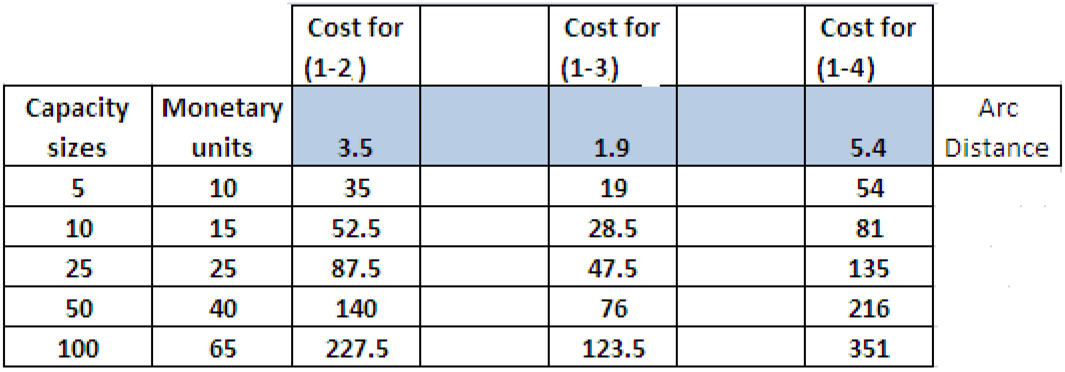
The objective function for the extended tree knapsack model was formulated in Chapter 4 (section 4.2.2.1) as

$$\text{Minimize} \qquad \sum_{(i,j)} E_{ij}\delta_{ij} + \sum_{(i,j)} a_{ij} f_{ij} . \qquad (4.1)$$

The E_{ij} represents the fixed cost of installing a pipeline on arc (i,j). The a_{ij} represents a unit of the variable cost of installing a pipeline of larger capacity on arc (i,j). The a_{ij} is found by first calculating the cost of each pipe capacity for the arc (i,j). This cost is the product of the monetary unit associated with each pipe capacity multiplied by the length of the arc (i,j).

Table 5.2 below shows the calculation of cost for the three arcs (1-2), (1-3) and (1-4)

Table 5-2: Calculated cost of each pipe



		Cost for (1-2)		Cost for (1-3)		Cost for (1-4)	
Capacity sizes	Monetary units	3.5		1.9		5.4	Arc Distance
5	10	35		19		54	
10	15	52.5		28.5		81	
25	25	87.5		47.5		135	
50	40	140		76		216	
100	65	227.5		123.5		351	

The a_{ij} and E_{ij} values used in this dissertation were then calculated as follows.

Consider the following table (table 5.3) with data for arc (1-2) that was taken from table 5.2.

Table 5-3: Calculated cost for arc (1-2)

		Cost for (1-2)
Capacity sizes	Monetary units	3.5
5	10	35
10	15	52.5
25	25	87.5
50	40	140
100	65	227.5
	slope	1.9938

A variable cost value for a_{12} is now computed by fitting a straight line, of the form $y = mx + c$, to the capacity versus computed cost values in the table above. This resulted in the following equation: $y = 1.9938x + 32.736$. The value $m = 1.9938$ now represents the variable cost value for a_{12}. The 32.736 is taken to be the fixed cost E_{12} to install the pipeline on arc (1-2). It should be noted that this method gives a relaxation of the real problem. The discrete costs are replaced by a fixed charge function. Figure 5.3 is a graphical representation of the calculation of the a_{12} value.

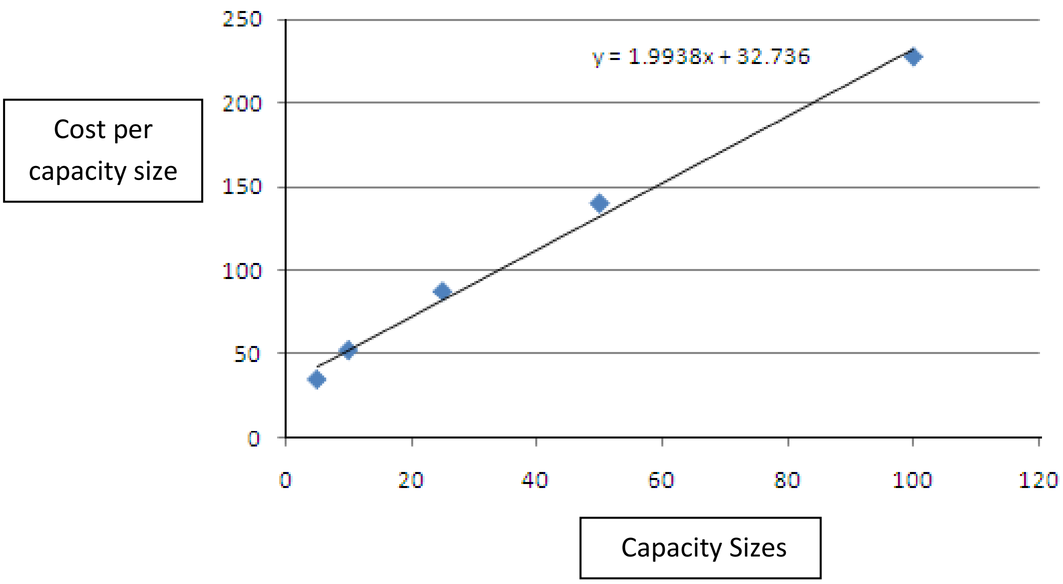


Figure 5.3 Computed cost incurred for arc 1-2

The above procedure was carried out for each arc (i,j) in order to obtain all the a_{ij} values. A complete list of all arcs and their associated a_{ij} values can be found in Appendix E.

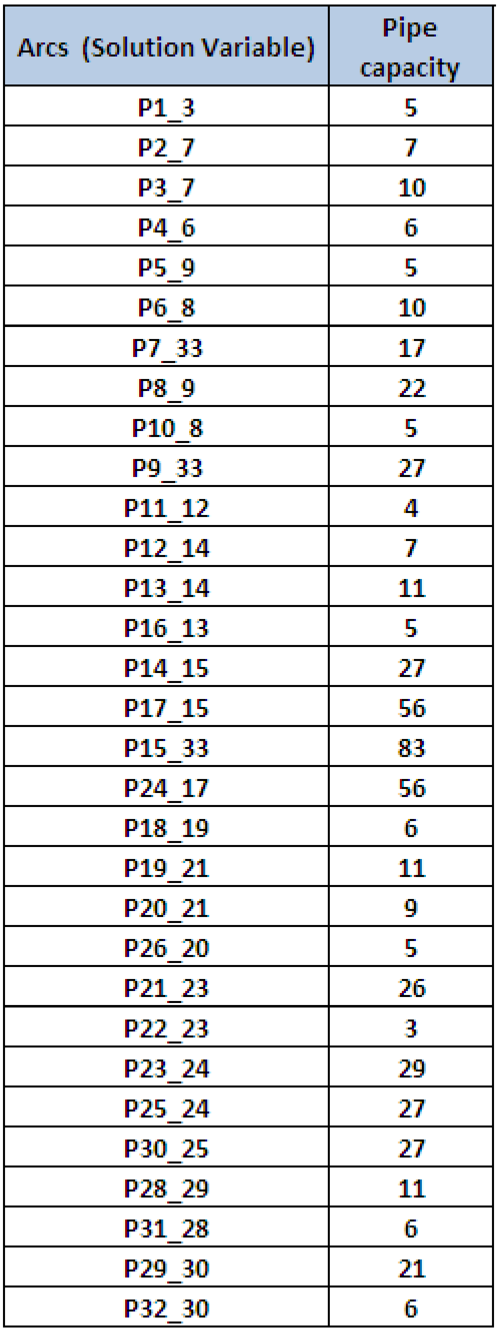
The model was then formulated as explained in chapter 4 with the data from the Brimberg *et al*. (2003) case study with the a_{ij} values described above. The complete formulation and programs to run and solve the model are included on the CD at the back of this dissertation. The model was solved using CPLEX software; the results will be discussed in the next section.

5.3 Results and discussion

Solving the extended tree knapsack model in CPLEX, a solution was obtained after 70.15 seconds. Table 5.4 shows the selected arcs in the solution. The pipe capacities suggested are given in the last column.

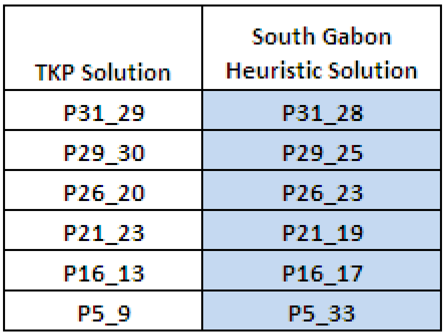
Comparing the results of the extended tree knapsack model with the results presented in the case study, it was found that in 6 instances the tree knapsack solution has chosen different arcs from those in the case study. The differences are reported in table 5.5.

Table 5-4: Selected arcs for the solution



Arcs (Solution Variable)	Pipe capacity
P1_3	5
P2_7	7
P3_7	10
P4_6	6
P5_9	5
P6_8	10
P7_33	17
P8_9	22
P10_8	5
P9_33	27
P11_12	4
P12_14	7
P13_14	11
P16_13	5
P14_15	27
P17_15	56
P15_33	83
P24_17	56
P18_19	6
P19_21	11
P20_21	9
P26_20	5
P21_23	26
P22_23	3
P23_24	29
P25_24	27
P30_25	27
P28_29	11
P31_28	6
P29_30	21
P32_30	6

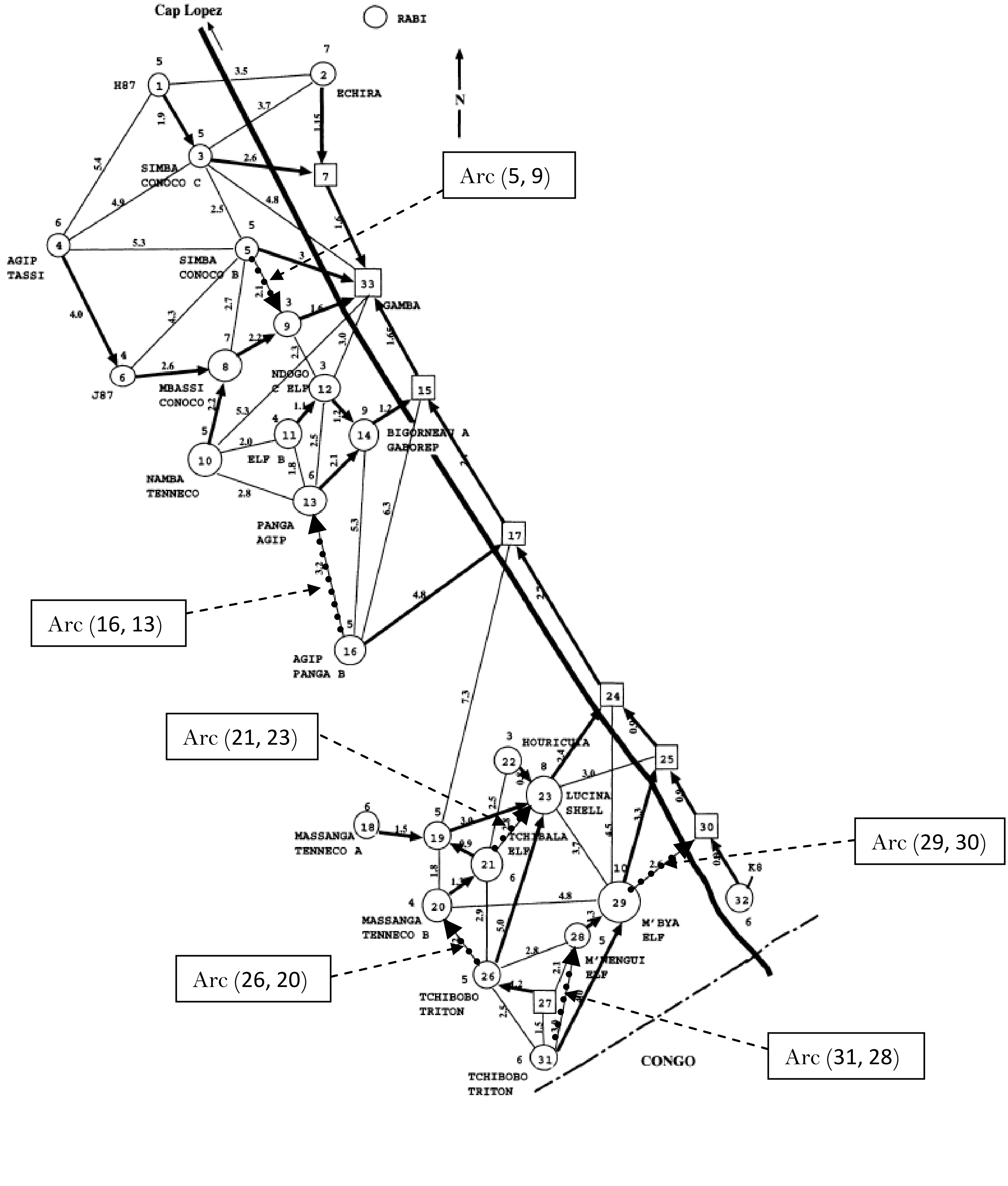
Table 5-5: Solution comparison



TKP Solution	South Gabon Heuristic Solution
P31_29	P31_28
P29_30	P29_25
P26_20	P26_23
P21_23	P21_19
P16_13	P16_17
P5_9	P5_33

Figure 5.4 depicts the oil pipeline network indicating the original solution from the Brimberg *et al*. (2003) study as well as that obtained by solving the tree knapsack model. The arcs shown by dotted lines connecting the nodes are the arcs obtained with the tree knapsack model.

The difference in the two solutions can be attributed to the relaxation used for the objective function that was used in the extended tree knapsack model (See 4.2.2.1). The relaxation in general results in an over or under estimation of the costs. Figure 5.5 illustrates this situation for arc (1, 4)


Arc (5, 9)

Arc (16, 13)

Arc (21, 23)

Arc (29, 30)

Arc (26, 20)

Arc (31, 28)

Figure 5.4 Tree knapsack solution for South Gabon oil pipeline network

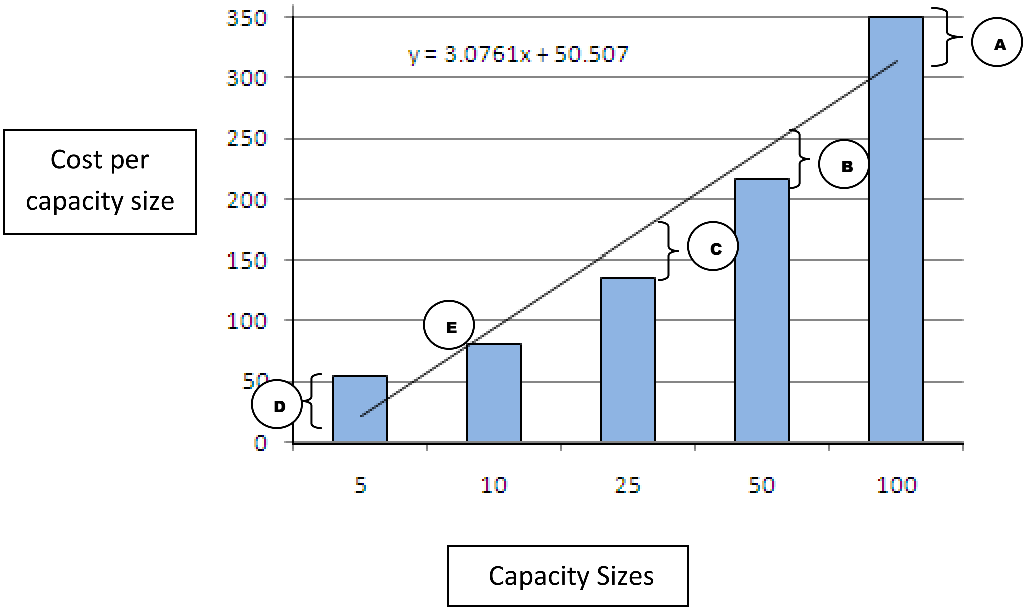


Figure 5.5 Illustration of possible over and underestimation of costs for arc (1,4)

In figure 5.5 it can be seen that A and D show an underestimation of the cost incurred above capacity level while B and C indicate an overestimation. E indicates the precise cost incurred for a capacity of 10 units.

To compare this study's results meaningfully with those of Brimberg *et al*. (2003), the cost was calculated using the actual pipe costs for the solution obtained with the extended tree knapsack model.

For example:

The flow from node 1 to node 3 is 5 (see figure 5.4) which requires a pipe capacity of at least 5. The cost of a pipe with capacity 5 was given as 10 units and the length of the pipe is 1.9. The cost for arc 1,3 is therefore 10 x 1.9 = 19 units.

The flow from node 3 to node 7 is 5 (coming through arc 1-3) plus 5 (generated at node 3). This flow of 10 requires a pipe capacity of 10 with an associated cost of 15 and the pipe length between 3 and 7 is 2.6. The cost for arc 3, 7 is therefore 15 x 2.6 = 39 units.

The cost for all selected arcs in both solutions was calculated in this way. This resulted in a cost of 1423 units for the Brimberg *et al*. (2003) study and 1461 units for the tree knapsack model – a deviation of 2.6 %.

The results of the study can be summarized as follows.

- The result of this study was within 2.6 % of that in the Brimberg *et al*. (2003) study. The low percentage deviation proves that it is definitely feasible to use a tree knapsack approach to solve network flow problems. By refining the cost function used in the tree knapsack model the 2.6% gap could be further reduced or may even be more efficient than the original case study.

- The feasibility is further proved by the relatively short time it took to solve the model (70.15 seconds using CPLEX).

- The case study investigation was a fairly large network comprising 33 nodes and 129 arcs. With modern software, e.g. CPLEX, large network flow problems may be solved in a reasonable time using the tree knapsack approach.

5.4 Conclusion

A detailed description of the empirical experiment and results was furnished in this chapter. The data used, the tree generation process and the tree knapsack model were discussed. This was followed by a consideration of the results which showed that it is feasible to use a tree knapsack approach for solving a network flow problem.

6

Summary and conclusions

6.1 Introduction

Chapter 6 presents the final comments and concluding remarks of the study. The objectives of the study and how they were achieved will be summarised. New problems and opportunities for further study that presented themselves during the research project will also be outlined.

6.2 Objectives of the study

Chapter 1 stated that the primary objective of this study was to investigate the feasibility of using an extended tree knapsack approach to solve a network flow problem. To accomplish this, a list of four secondary research objectives was defined that needed to be achieved:

- Gain a clear understanding of and present an introductory overview of general network flow and tree knapsack models;

- Select and provide an overview of a suitable case study from the literature that can be used in the research project;

- Describe and formulate the tree knapsack and the extended tree knapsack approach and model; and

- Describe and present the results of the tree knapsack approach when applied to the selected case study.

A summary of how these objectives were achieved is now provided.

Gain a clear understanding of and present an introductory overview of general network flow and tree knapsack models

Network flow problems are an important class of linear programming models and can be modelled in a variety of ways. Equally important are integer programming models of which the knapsack problem is an important application. In this study a network flow problem was solved using a tree knapsack approach and therefore it is imperative to gain a clear understanding both of network flow models as well as of knapsack models.

The objective was achieved by describing network flow models and the important and familiar applications such as transportation, assignment, transshipment, shortest route, maximum flow and minimum spanning tree problems (Chapter 2, section 2.2.1 – 2.2.7). An overview of knapsack models was supplied (Chapter 2, sections 2.3.2.1 – 2.3.2.7).

Select and provide an overview of a suitable case study from the literature that can be used in the research project

To compare and validate the proposed technique, a specific case study was chosen from the literature and was solved using the extended tree knapsack approach. The case study selected was an oil pipeline design problem by Brimley *et al.* (2003), which aimed at designing an optimal oil pipeline network for the South Gabon oil field.

The objective was achieved by affording an overview of the oil pipeline design problem (Chapter 3, section 3.2). A survey of the models suggested in the literature (Chapter 3, section 3.3) as well as solution methods and results was also presented (Chapter 3, sections 3.4 – 3.5).

Describe and formulate the tree knapsack and extended tree knapsack approach and model

It is important to obtain a clear understanding of the research design and methodology followed to develop a tree knapsack model that can be used to assess the feasibility of solving the pipeline network problem in an alternative way.

This objective was achieved by providing a comprehensive discussion on the methodology followed in this study. First, a description was given of how to convert the pipeline network into a tree structure (Chapter 4, section 4.2.1). This was followed by an explanation of the mathematical programming model that was used to solve the tree structure (Chapter 4, section 4.2.2).

Describe and present the results of the tree knapsack approach when applied to the selected case study

In order to express an opinion on the feasibility of the suggested tree knapsack approach, the proposed model was applied to a specific case study.

This objective was achieved by performing an empirical experiment using the oil pipeline design data in the proposed tree knapsack model (Chapter 5, section 5.2). The results and a comparison with the solution presented in the literature were also furnished (Chapter 5, section 5.3). All information concerning this final objective was considered in chapter 5.

To summarise, all objectives set forth in chapter 1 were achieved. Based on the results and discussion presented in chapter 5, it was concluded that,

- It is feasible to solve network flow problems by employing a tree knapsack approach.

- Relatively large problems can be solved with this approach using modern software such as CPLEX.

- These types of problems can be solved in a relatively short time.

6.3 Problems experienced

To compare a model's results with the results of other models is one way of validating work that was performed. There is a lack of case studies (containing sufficient data) such as the oil pipeline problem that could be used to strengthen the empirical evidence for the proposed tree knapsack approach.

Another difficulty (to a lesser extent) experienced was the lack or unavailability of existing software that can be used to convert network flow models into tree networks. For larger networks, like the oil pipeline network, the indexed tree structure may become fairly large and care must be taken when writing programs to handle the tree structure: memory management may be especially difficult.

In this study an approximation was used to represent the objective function. The approximation can be made more precise at the expense of introducing more discrete variables. This approach was not pursued further in this dissertation.

6.4 Possibilities for further research

Experiments with different cost functions, used in the objective function of the proposed model, could be performed in an effort to refine the model and to further improve results.

To strengthen the empirical evidence for the feasibility of using a tree knapsack approach in a network flow problem, other case studies might also be investigated.

Techniques to improve and enhance model performance, such as those suggested by Van der Merwe and Hattingh (2006), could be incorporated in this study to provide solutions for even larger network flow models.

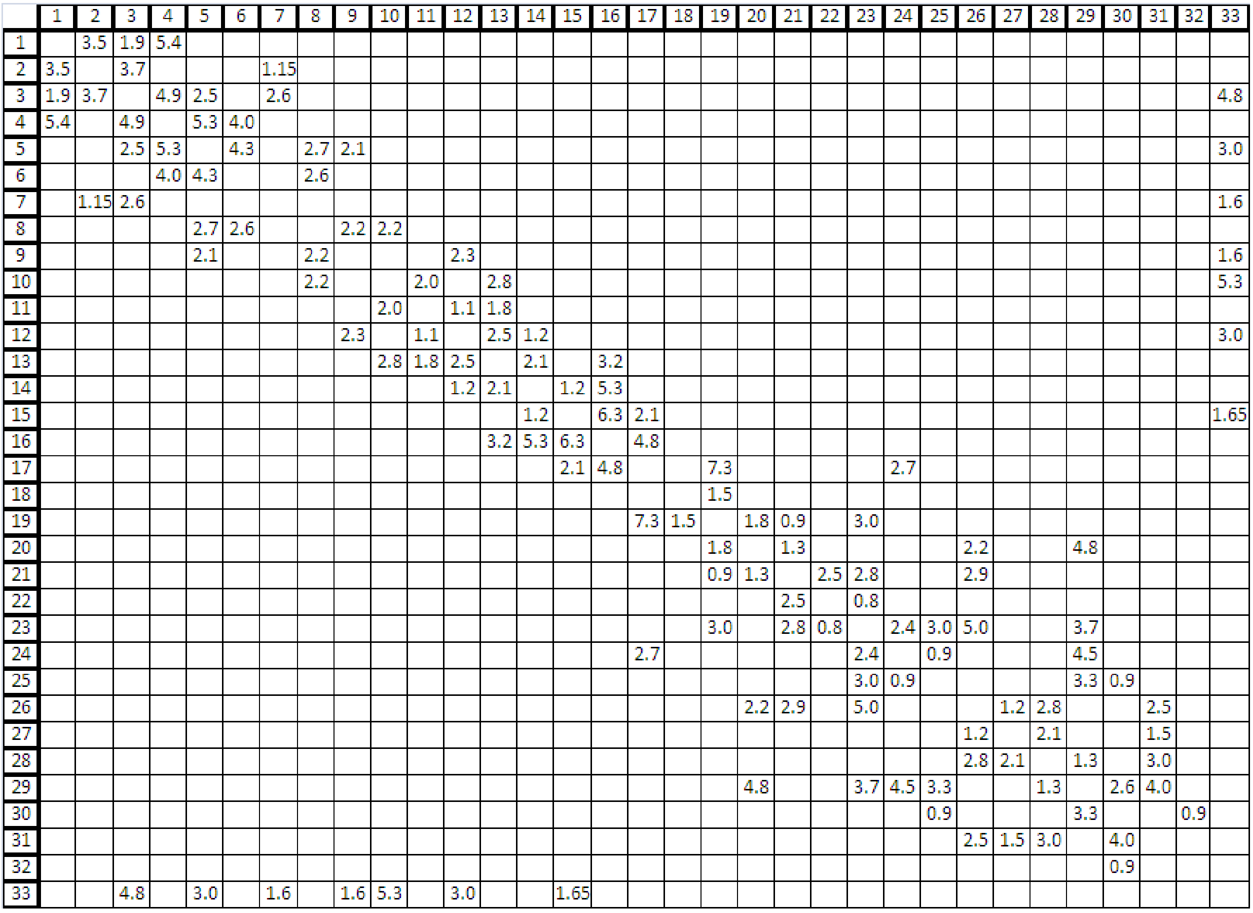
6.5 Conclusion

Chapter 6 is the final chapter of this study. It furnished a summary of the initial objectives and of how they were achieved. In the conclusion, problems and possible future research opportunities were outlined.

Appendix A

ARC DISTANCES

This appendix shows all the arc distances between some well sites of the South Gabon oil field. For example, the length of arc (2, 3) is 3.7.

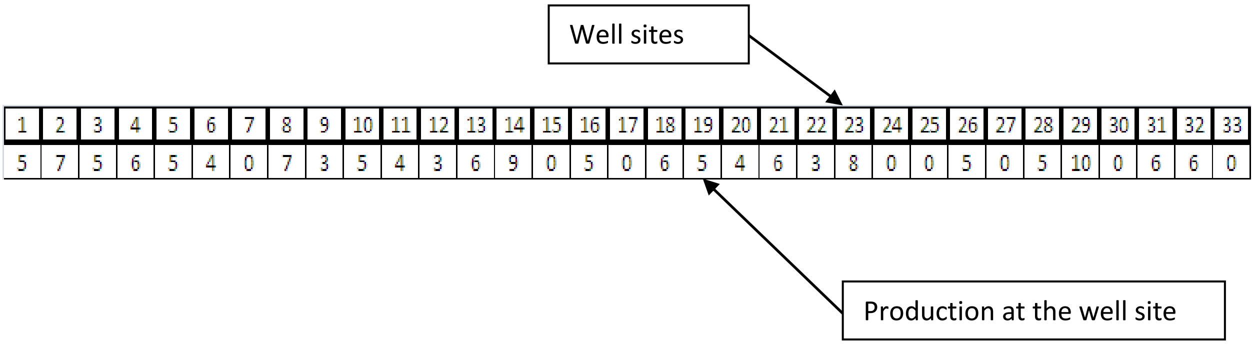


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1		3.5	1.9	5.4																													
2	3.5		3.7				1.15																										
3	1.9	3.7		4.9	2.5		2.6																										4.8
4	5.4		4.9		5.3	4.0																											
5			2.5	5.3		4.3		2.7	2.1																								3.0
6				4.0	4.3			2.6																									
7		1.15	2.6																														1.6
8					2.7	2.6			2.2	2.2																							
9					2.1			2.2				2.3																					1.6
10								2.2			2.0		2.8																				5.3
11										2.0		1.1	1.8																				
12									2.3		1.1		2.5	1.2																			3.0
13										2.8	1.8	2.5		2.1		3.2																	
14												1.2	2.1		1.2	5.3																	
15														1.2		6.3	2.1																1.65
16													3.2	5.3	6.3		4.8																
17															2.1	4.8			7.3					2.7									
18																			1.5														
19																	7.3	1.5		1.8	0.9		3.0										
20																			1.8		1.3					2.2			4.8				
21																			0.9	1.3		2.5	2.8			2.9							
22																					2.5		0.8										
23																			3.0		2.8	0.8		2.4	3.0	5.0			3.7				
24																	2.7						2.4		0.9				4.5				
25																							3.0	0.9					3.3	0.9			
26																				2.2	2.9		5.0				1.2	2.8			2.5		
27																										1.2		2.1			1.5		
28																										2.8	2.1		1.3		3.0		
29																				4.8			3.7	4.5	3.3			1.3		2.6	4.0		
30																									0.9				3.3			0.9	
31																										2.5	1.5	3.0		4.0			
32																														0.9			
33			4.8		3.0		1.6		1.6	5.3		3.0			1.65																		

Appendix B

PRODUCTION AT EACH WELL SITE

Appendix B shows the amount of oil produced at each well site in South Gabon oil field. It is not clear what the units of measure used for the production at each well site, were.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
5	7	5	6	5	4	0	7	3	5	4	3	6	9	0	5	0	6	5	4	6	3	8	0	0	5	0	5	10	0	6	6	0

Well sites
Production at the well site

Appendix C

PROGRAM PSEUDO CODE TO GENERATE A TREE NETWORK

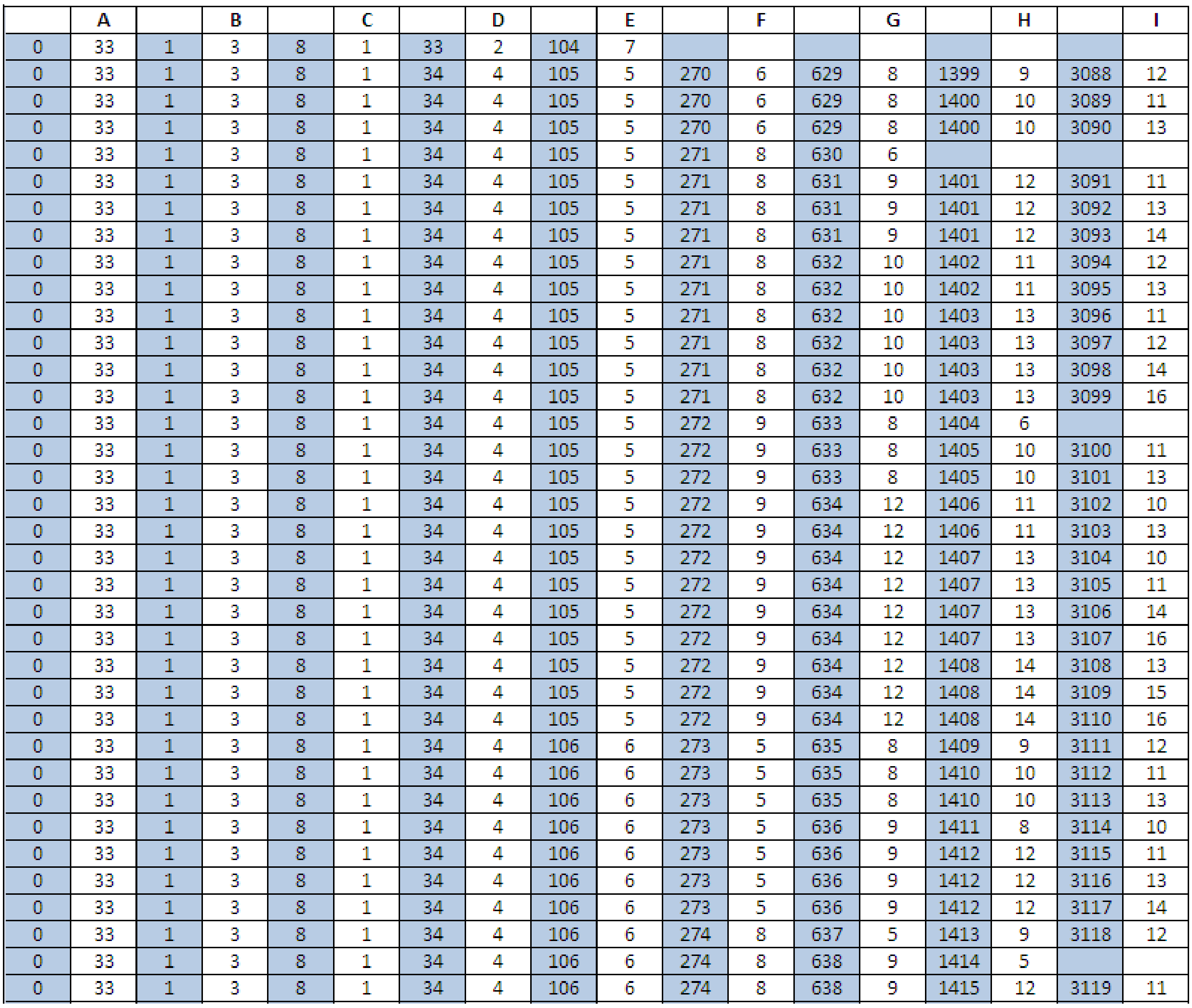
```
Begin
Set class of Node {
  Set public member as;
       Initialize node n, lvl
       level (lvl) {
        Set number to n
        Increment total Nodes Created;
    }
   Set Copy constructor for all members
}
   Initialize number, level, and index
   Set total Nodes Created as static
   Set pointers to adjacent node
   Friend bool operator is greater than (First level, second level)   {
            If pointer to the first level is great than pointer to the second level then
                              Return true;
                    Else
                              Return false;
            }
    Free Nodes {
      Decrement total Nodes Created
    }
Create class tree {
  Set public members as;
    Vector of network
    Pointer to the root node
    Pointer to the output file
   A string of output File Name
    Pointer to the construct Tree
    Tree (Initialize vector net, n, set file Name as string)
    Set empty list Paths (initialized destination Node)
    Set attach Index ();
    Set myBubbleSort ();
        Index all node incenses
        Keeps track of the allocated memory;
};
Initialize the static variable   {
  If there is no output exit
   Else
      Attach Index      }
If current Node exists {
  If the current node in the list of nodes visited exist then
   Return 0
  Else
  Insert the current node into the list of nodes visited
}
Control the depth of the tree
}
```

```
Go down one level in tree {
  Record all the memory allocated
    Optimize indexing and move the correct row
     }
 Go up one level {
      Return previous Node
  }
 Open the file containing the output file once done
   Print paths
Sort the level data member {
   Attach the index
     Return
}
Initialize Main () {
  Set length to 33
  Raw input {
     Enter the input file
      If no file found exit
     Else
       Read the file
 }
Enter the name of the output file;
   Enter the source node and enter the destination node
     Return 0
}
End
```

Appendix D

GENERATED TREE NETWORK

This appendix contains a small sample of the tree network results obtained from the execution of the tree generation program. Columns A to I represent the actual nodes from the South Gabon oil field network.

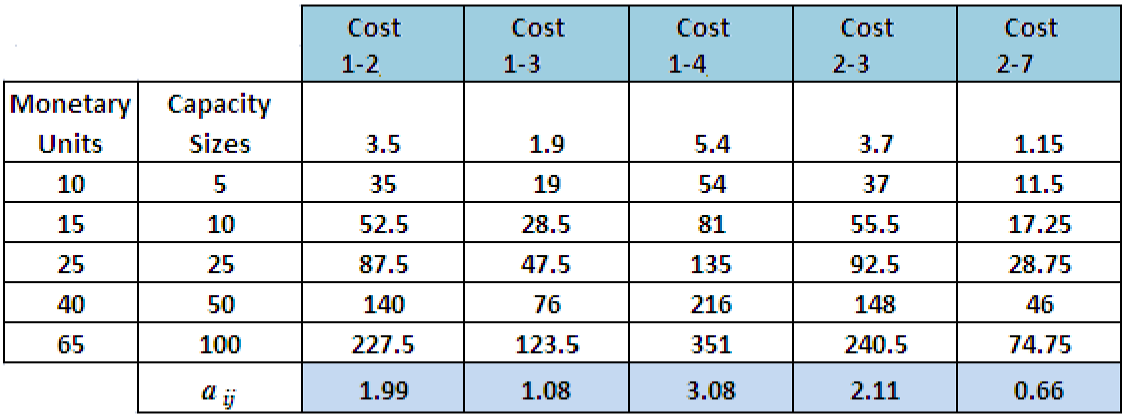


	A		B		C		D		E		F		G		H		I
0	33	1	3	8	1	33	2	104	7								
0	33	1	3	8	1	34	4	105	5	270	6	629	8	1399	9	3088	12
0	33	1	3	8	1	34	4	105	5	270	6	629	8	1400	10	3089	11
0	33	1	3	8	1	34	4	105	5	270	6	629	8	1400	10	3090	13
0	33	1	3	8	1	34	4	105	5	271	8	630	6				
0	33	1	3	8	1	34	4	105	5	271	8	631	9	1401	12	3091	11
0	33	1	3	8	1	34	4	105	5	271	8	631	9	1401	12	3092	13
0	33	1	3	8	1	34	4	105	5	271	8	631	9	1401	12	3093	14
0	33	1	3	8	1	34	4	105	5	271	8	632	10	1402	11	3094	12
0	33	1	3	8	1	34	4	105	5	271	8	632	10	1402	11	3095	13
0	33	1	3	8	1	34	4	105	5	271	8	632	10	1403	13	3096	11
0	33	1	3	8	1	34	4	105	5	271	8	632	10	1403	13	3097	12
0	33	1	3	8	1	34	4	105	5	271	8	632	10	1403	13	3098	14
0	33	1	3	8	1	34	4	105	5	271	8	632	10	1403	13	3099	16
0	33	1	3	8	1	34	4	105	5	272	9	633	8	1404	6		
0	33	1	3	8	1	34	4	105	5	272	9	633	8	1405	10	3100	11
0	33	1	3	8	1	34	4	105	5	272	9	633	8	1405	10	3101	13
0	33	1	3	8	1	34	4	105	5	272	9	634	12	1406	11	3102	10
0	33	1	3	8	1	34	4	105	5	272	9	634	12	1406	11	3103	13
0	33	1	3	8	1	34	4	105	5	272	9	634	12	1407	13	3104	10
0	33	1	3	8	1	34	4	105	5	272	9	634	12	1407	13	3105	11
0	33	1	3	8	1	34	4	105	5	272	9	634	12	1407	13	3106	14
0	33	1	3	8	1	34	4	105	5	272	9	634	12	1407	13	3107	16
0	33	1	3	8	1	34	4	105	5	272	9	634	12	1408	14	3108	13
0	33	1	3	8	1	34	4	105	5	272	9	634	12	1408	14	3109	15
0	33	1	3	8	1	34	4	105	5	272	9	634	12	1408	14	3110	16
0	33	1	3	8	1	34	4	106	6	273	5	635	8	1409	9	3111	12
0	33	1	3	8	1	34	4	106	6	273	5	635	8	1410	10	3112	11
0	33	1	3	8	1	34	4	106	6	273	5	635	8	1410	10	3113	13
0	33	1	3	8	1	34	4	106	6	273	5	636	9	1411	8	3114	10
0	33	1	3	8	1	34	4	106	6	273	5	636	9	1412	12	3115	11
0	33	1	3	8	1	34	4	106	6	273	5	636	9	1412	12	3116	13
0	33	1	3	8	1	34	4	106	6	273	5	636	9	1412	12	3117	14
0	33	1	3	8	1	34	4	106	6	274	8	637	5	1413	9	3118	12
0	33	1	3	8	1	34	4	106	6	274	8	638	9	1414	5		
0	33	1	3	8	1	34	4	106	6	274	8	638	9	1415	12	3119	11

Appendix E

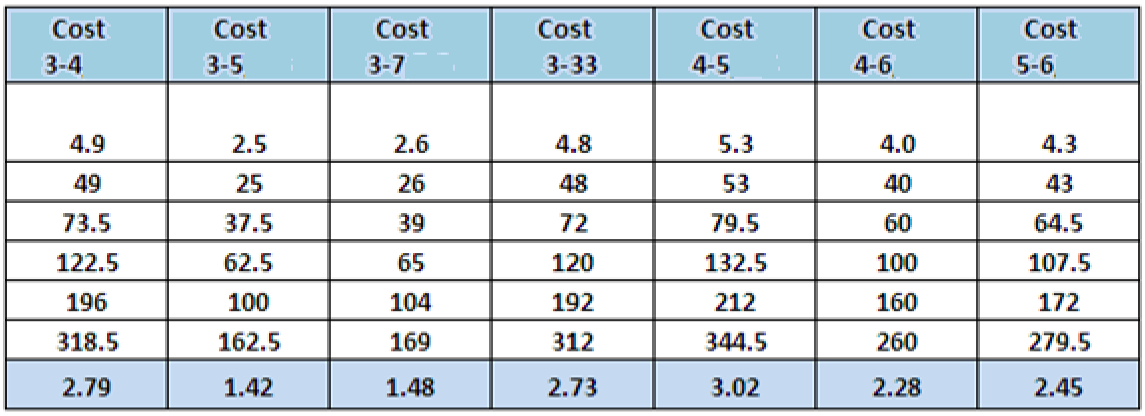
COMPUTED COST INCURRED (a_{ij})

Appendix E shows a complete list of all arcs and their associated a_{ij}.



		Cost 1-2	Cost 1-3	Cost 1-4	Cost 2-3	Cost 2-7
Monetary Units	Capacity Sizes	3.5	1.9	5.4	3.7	1.15
10	5	35	19	54	37	11.5
15	10	52.5	28.5	81	55.5	17.25
25	25	87.5	47.5	135	92.5	28.75
40	50	140	76	216	148	46
65	100	227.5	123.5	351	240.5	74.75
	a_{ij}	1.99	1.08	3.08	2.11	0.66

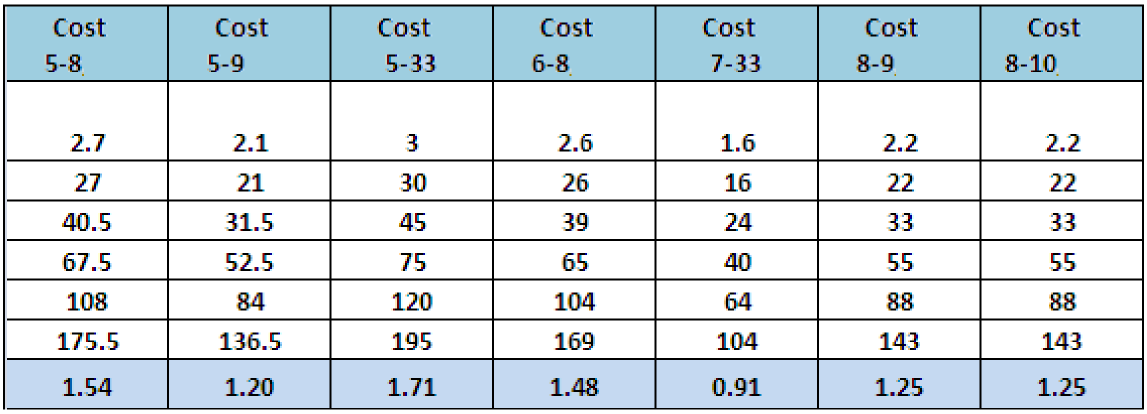
← **Arc distance**



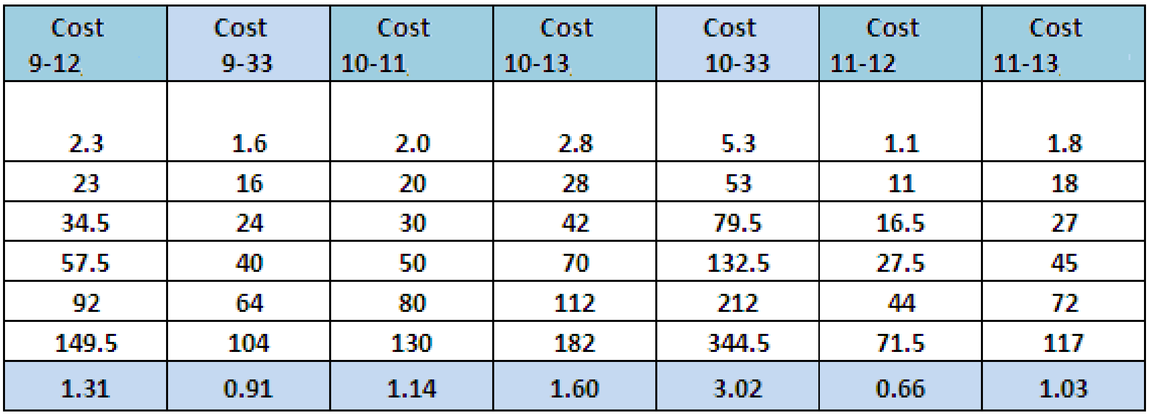
Cost 3-4	Cost 3-5	Cost 3-7	Cost 3-33	Cost 4-5	Cost 4-6	Cost 5-6
4.9	2.5	2.6	4.8	5.3	4.0	4.3
49	25	26	48	53	40	43
73.5	37.5	39	72	79.5	60	64.5
122.5	62.5	65	120	132.5	100	107.5
196	100	104	192	212	160	172
318.5	162.5	169	312	344.5	260	279.5
2.79	1.42	1.48	2.73	3.02	2.28	2.45

← **Arc distance**

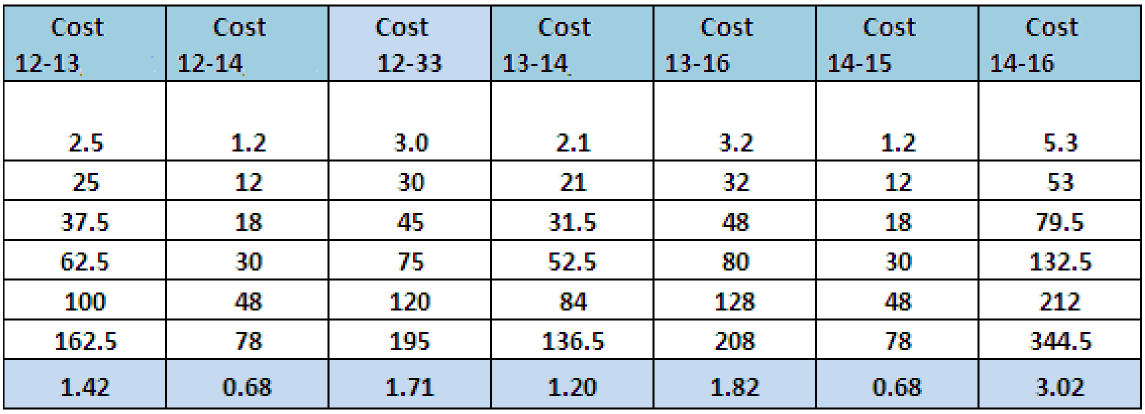
a_{ij} →



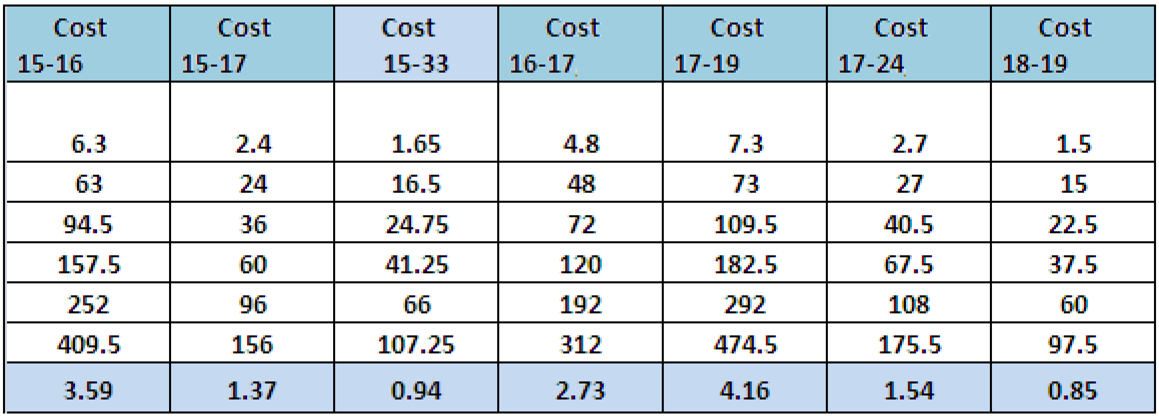
	Cost 5-8	Cost 5-9	Cost 5-33	Cost 6-8	Cost 7-33	Cost 8-9	Cost 8-10
Arc distance	2.7	2.1	3	2.6	1.6	2.2	2.2
	27	21	30	26	16	22	22
	40.5	31.5	45	39	24	33	33
	67.5	52.5	75	65	40	55	55
	108	84	120	104	64	88	88
	175.5	136.5	195	169	104	143	143
a_{ij}	1.54	1.20	1.71	1.48	0.91	1.25	1.25



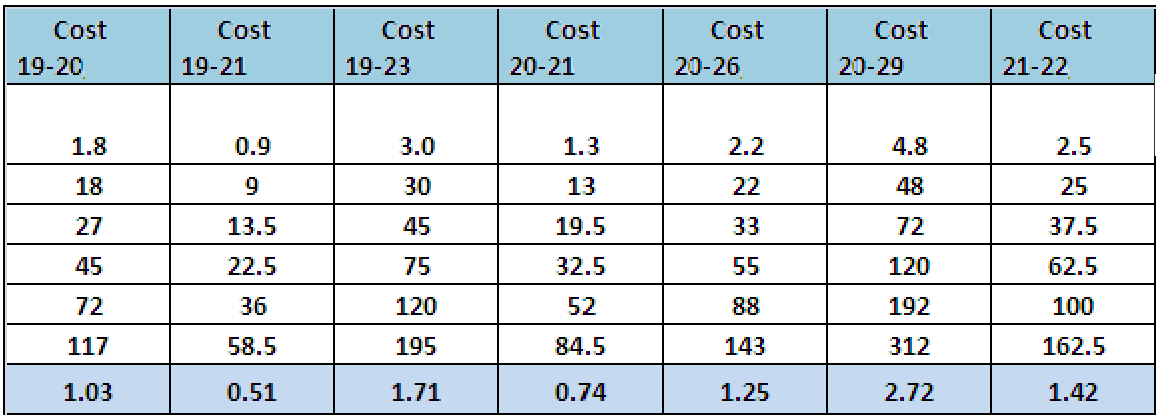
	Cost 9-12	Cost 9-33	Cost 10-11	Cost 10-13	Cost 10-33	Cost 11-12	Cost 11-13
Arc distance	2.3	1.6	2.0	2.8	5.3	1.1	1.8
	23	16	20	28	53	11	18
	34.5	24	30	42	79.5	16.5	27
	57.5	40	50	70	132.5	27.5	45
	92	64	80	112	212	44	72
	149.5	104	130	182	344.5	71.5	117
a_{ij}	1.31	0.91	1.14	1.60	3.02	0.66	1.03



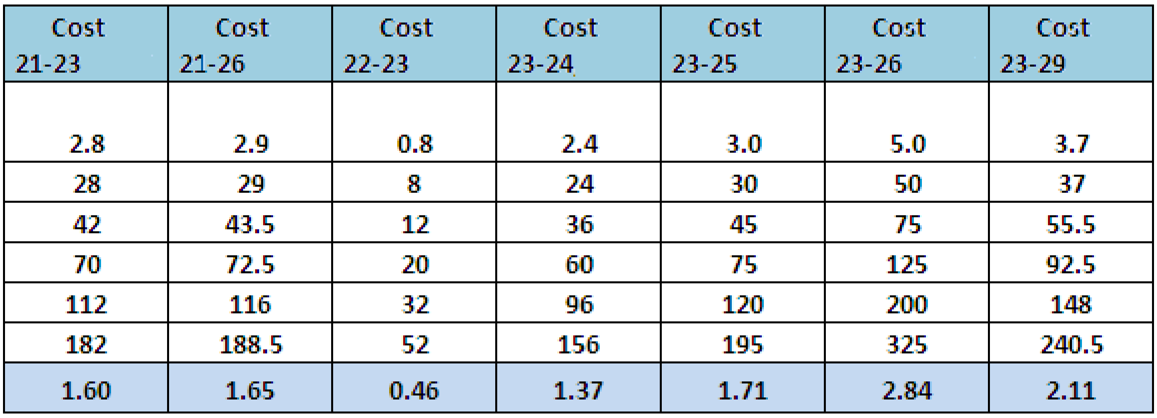
	Cost 12-13	Cost 12-14	Cost 12-33	Cost 13-14	Cost 13-16	Cost 14-15	Cost 14-16
Arc distance	2.5	1.2	3.0	2.1	3.2	1.2	5.3
	25	12	30	21	32	12	53
	37.5	18	45	31.5	48	18	79.5
	62.5	30	75	52.5	80	30	132.5
	100	48	120	84	128	48	212
	162.5	78	195	136.5	208	78	344.5
a_{ij}	1.42	0.68	1.71	1.20	1.82	0.68	3.02



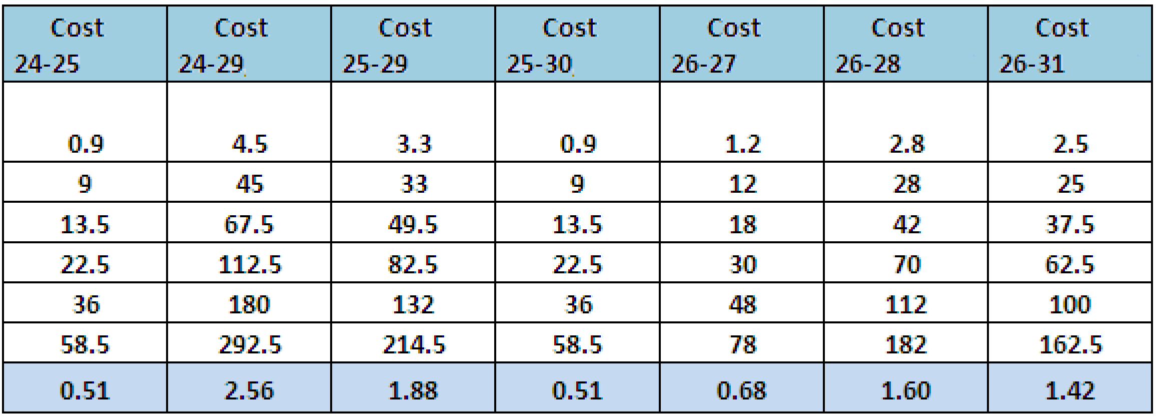
	Cost 15-16	Cost 15-17	Cost 15-33	Cost 16-17	Cost 17-19	Cost 17-24	Cost 18-19
Arc distance	6.3	2.4	1.65	4.8	7.3	2.7	1.5
	63	24	16.5	48	73	27	15
	94.5	36	24.75	72	109.5	40.5	22.5
	157.5	60	41.25	120	182.5	67.5	37.5
	252	96	66	192	292	108	60
	409.5	156	107.25	312	474.5	175.5	97.5
a_{ij}	3.59	1.37	0.94	2.73	4.16	1.54	0.85



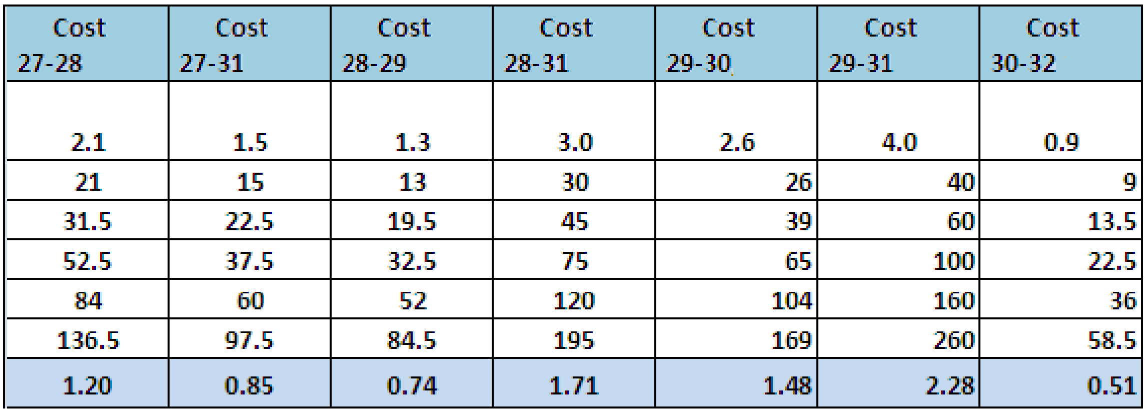
	Cost 19-20	Cost 19-21	Cost 19-23	Cost 20-21	Cost 20-26	Cost 20-29	Cost 21-22
Arc distance	1.8	0.9	3.0	1.3	2.2	4.8	2.5
	18	9	30	13	22	48	25
	27	13.5	45	19.5	33	72	37.5
	45	22.5	75	32.5	55	120	62.5
	72	36	120	52	88	192	100
	117	58.5	195	84.5	143	312	162.5
a_{ij}	1.03	0.51	1.71	0.74	1.25	2.72	1.42



	Cost 21-23	Cost 21-26	Cost 22-23	Cost 23-24	Cost 23-25	Cost 23-26	Cost 23-29
Arc distance	2.8	2.9	0.8	2.4	3.0	5.0	3.7
	28	29	8	24	30	50	37
	42	43.5	12	36	45	75	55.5
	70	72.5	20	60	75	125	92.5
	112	116	32	96	120	200	148
	182	188.5	52	156	195	325	240.5
a_{ij}	1.60	1.65	0.46	1.37	1.71	2.84	2.11



	Cost 24-25	Cost 24-29	Cost 25-29	Cost 25-30	Cost 26-27	Cost 26-28	Cost 26-31	
	0.9	4.5	3.3	0.9	1.2	2.8	2.5	Arc distanc
	9	45	33	9	12	28	25	
	13.5	67.5	49.5	13.5	18	42	37.5	
	22.5	112.5	82.5	22.5	30	70	62.5	
	36	180	132	36	48	112	100	
	58.5	292.5	214.5	58.5	78	182	162.5	
a_{ij}	0.51	2.56	1.88	0.51	0.68	1.60	1.42	



	Cost 27-28	Cost 27-31	Cost 28-29	Cost 28-31	Cost 29-30	Cost 29-31	Cost 30-32	
	2.1	1.5	1.3	3.0	2.6	4.0	0.9	Arc distance
	21	15	13	30	26	40	9	
	31.5	22.5	19.5	45	39	60	13.5	
	52.5	37.5	32.5	75	65	100	22.5	
	84	60	52	120	104	160	36	
	136.5	97.5	84.5	195	169	260	58.5	
a_{ij}	1.20	0.85	0.74	1.71	1.48	2.28	0.51	

Bibliography

ANDERSON, D.R., SWEENEY, D.J., WILLIAMS, T.A., & WISNIEWSKI, M. 2009. *An introduction to management science*. London: Cengage-Learning. 864p

BLACK, P.E., & TANENBAUM, P.J. 2010. Graphs-National Institute of Standards and Technology. [Available online], http://www.itl.nist.gov/div897/sqg/dads/HTML/graph.html Date of access 16 Sep. 2010.

BRETTHAUER, K.M. & SHETTY, B. 2002. The nonlinear knapsack problem: algorithms and applications. *European Journal of Operational Research,* 138: 459-472.

BRIMBERG, J., HANSEN, P., LIH, K. W., MLADENOVIC, N., & BRETON, M. 2003. An oil pipeline design problem. *Operational Research,* 51(2): 228-239.

CARRAWAY, R.L., SCHMIDT, R.L. & WEATHERFORD, L.R. 1993. An algorithm for maximizing target achievement in the stochastic knapsack problem with normal returns. *Naval Research Logistics,* 40: 161-173.

CERI, S., MARTELLA, G., & PELAGATTI, G. 1982. Optimal file allocation in a computer network: a solution method based on the knapsack problem. *Computer Networks,* 6(5): 345-357.

DE VILLIERS, D. 2004. Optimization of network mesh topologies and link capacities for congestion relief. PU for CHE (Dissertation –M.Sc.). 102p.

ERKUT, E., & VERTER, V. 1998. Modeling of transport risk for hazardous materials.*INFORMS: Operation Research and Management Science,* 46(5): 625-642.

FISHMAN, G.S. 1987. The distribution of maximum flow with applications to multistate reliability systems. *INFORMS: Journal on Operations Research*, 35(4): 607-618.

GRAHOVAC, J., & CHAKRAVARTY, A. 2001.Sharing and lateral transshipment of inventory in a supply chain with expensive low-demand items. *INFORMS: Journal on Management Science*, 47(4): 579-594.

GUTIERREZ-JARPA, G., DONOSO, M., OBREQUE, C. & MARIANOV, V. 2009. Minimum cost path location for maximum traffic capture. *Computers & Industrial Engineering*, 58: 332-341.

HAGE, P., HARARY, F., & JAMES, B. 1996. The minimum spanning tree problem in archaeology. *Society for American Archaeology*, 61(1): 149-155.

HILL, R., & HIREMATH, C.S. 2000. Generation methods for multidimensional knapsack problems and their implications. [Available online].. *http://www.iiisci.org/journal/CV$/sci/pdfs/P903080.pdf* Date of access: 09 Sep. 2010.

JAIN, A., & MAMER, J.W. 1988. Approximations for the random minimal spanning tree with application to network provisioning. *INFORMS:Journal on Operations Research*, 36(4): 575-584.

JANG, J., & WANG, S.J. 1993. A dynamic access control scheme based upon the knapsack problem. *Computers & Mathematics with Applications*, 26(12): 75-86.

KAWATRA, R. & BRICKER, D. 1998. A multiperiod planning model for the capacitated minimal spanning tree problem. *European Journal of Operational Research*, 121: 412-419.

KOLLIOPOULOS, S. G., & STEINER, G. 2007. Partially ordered knapsack and applications to scheduling. *Discrete Applied Mathematics,* 155(8): 889-897.

KERSHENBAUM, A. 1993. *Telecommunication network design algorithm.* New York: McGraw-Hill, Inc. 368p.

KLINCEWICZ, J. G. 1990. Solving a freight transport problem using facility location techniques. *INFORMS: Journal on Operations Research*, 38: 99-109.

KUAH, G. K. & PERL, J. 1989. The feeder bus network design problem. *The Journal of the Operational Research Society*, 40: 751-767.

LEE, H., AHN, D.H. & KIM, S. 2003. Optimal routing in non geostationary satellite ATM networks with intersatellite link capacity constraints. *The Journal of the Operational Research Society*, 54: 401-409.

MARTELLO, S & TOTH, P. 1990. Algorithms for knapsack problems. *Annals of Discrete Mathematics*, 31: 213-258.

MELACHRINOUDIS, E. & KOZANIDIS, G. 2002. A mixed integer knapsack model for allocating funds to highway safety improvements. *Transportation Research Part A*, 36: 789-803.

MODARRES, M., & ZAREI, B. 2002. Application of network theory and AHP in urban transportation to minimize earthquake damages. *The Journal of the Operational Research Society*, 53: 12.

MORENO, E., ESPINOZA, D., & GOYCOOLEA, M. 2010. Large-scale multi-period precedence constrained knapsack problem: A mining application. *Discrete Mathematics*, 36: 407-414.

MORITA, H., ISHIII, H., & NISHIDA, T. 1989. Stochastic linear knapsack programming problem and its application to a portfolio selection problem. *European Journal of Operational Research*, 40(3): 329-336.

PATTERSON, A. R. & ROLLAND, E. 2002. Hybrid fiber coaxial network design. *INFORMS: Journal on Operations Research*, 50: 538-551.

RAGSDALE, C.T. 2007. *Managerial decision modeling*. Mason, USA: Thomson, South-Western. 820p.

RENDER, B., STAIR R.M., & HAUNA, M.E. 2006. *Quantitative analysis for management.* 9th Edition. Upper Saddle River: Prentice Hall. 731p.

ROSENTHAL, R.E. 1981. A nonlinear network flow algorithm for maximization of benefits in a hydroelectric power system. *INFORMS: Journal on Operations Research*, 29(4): 763-786.

SALKIN, H.M. & MATHUR, K. 1989. Foundations of integer programming. New York: North-Hollond. 755p.

SCHITTEKAT, P., SEVAUX, M. & SORENSEN, K. 2006. A mathematical formulation for a school bus routing problem. *Service Systems and Service Management*, 1552 – 1557.

SHARMA, D. K. & JANA, R.K. 2009. A hybrid genetic algorithm model for transshipment management decisions. *International Journal of Production Economics*, 122: 703–713.

SHAW, D.X & CHO, G. 1996. A depth-first dynamic programming procedure for the tree knapsack problem. *Journal on Computing*, 9: 431- 438.

SHAW, D.X. & CHO, G. 1996. The critical item, upper bounds and branch-and-bound algorithm for the tree knapsack problem. *Journal on Computing*, 31:205- 216.

SHAW, D.X. 1997. A depth-first dynamic programming procedure for the extended tree knapsack problem in local access network design telecommunication systems. *Journal on Computing*, 27: 29-43.

SONG, Y., ZHANG, C., & FANG, Y. 2008. Multiple multidimensional knapsack problem and its application in cognitive radio networks. *Military Communications Conference*, Florida, 1 – 7.

STROUP, J.S., & WOLLER, D.R. 1992. A fuel management model for the airline industry. *INFORMS: Journal on Operations Research and Management Science*, 40(2): 229-237.

TAYLOR, B. W. 2002. *Introduction to Management Science.* 7th Edition. Upper Saddle River , New Jersey: Prentice Hall. 764p.

TERBLANCHE, S.E. 2008. Contribution towards survivable network design with uncertain traffic requirements. North-West University, Potchefstroom (Thesis-Ph.D). 114p.

VAN DER MERWE, D. J. 2002. Solving tree knapsack problems for local access network design applications. North-West University, Potchefstroom (Dissertation-M.Sc). 115p.

VAN DER MERWE, D.J. 2007. The use of partitioning strategies in local access telecommunication network problem and other applications. School of Computer, Statistical and Mathematical Sciences. North-West University, Potchefstroom (Thesis-Ph.D). 112p.

VAN DER MERWE, D. J., & HATTINGH, G. 2006. Tree knapsack approaches for local access network design. *European Journal of Operational Research* , 174: 1968-1978.

VAN DER MERWE, D. J., & HATTINGH, G. 2010. Solving the extended tree knapsack problem with fixed cost flow expansion functions. ORiON: *Journal of the Operational Research Society of South Africa*, 26(1): 1-16.

VANDERSTER, D.C., DIMOPOULOS, N.J., HERNANDEZ, R. P., & SOBIE, R.J. 2009. Resource allocation on computational grids using a utility model and the knapsack problem. *Future Generation Computer Systems*, 25: 35-50.

WANG, B., & HU, Y. 2010. Quadratic compact knapsack public key cryptosystem. *Computers and Mathematics with Applications*, 59(1): 194-206.

WEATHERFORD, L. R., & MOORE, J.H. 2001. *Decision modeling with Microsoft Excel.* 6th Edition. Upper Saddle River, New Jersey: Prentice Hall. 693p.

WEE, K. E., & DADA, M. 2005. Optimal policies for transshipping inventory in a retail network. *INFORMS:Journal on Management Science,* 51(10): 1519-1533.