Modification, Development, Application and Computational Experiments of some selected Network, Distribution and Resource Allocation models in Operations Research

by

PHILIMON NYAMUGURE

THESIS

Submitted in fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

STATISTICS

in the

FACULTY OF SCIENCE AND AGRICULTURE
(School of Mathematical and Computer Sciences)

at the

UNIVERSITY OF LIMPOPO

PROMOTER: Prof. M Lesaoana

CO-PROMOTERS: Prof. S Kumar (University of Melbourne, Australia)
Prof. E Munapo (North-West University)

June 2017

Declaration

I, **Philimon Nyamugure**, declare that the thesis which is hereby submitted for the qualification of Doctor of Philosophy in Statistics at the University of Limpopo, is my own independent work and has not been handed in before for a qualification at/in another University/Faculty/School. I further declare that all sources cited or quoted are indicated and acknowledged by means of a comprehensive list of references. I further cede copyright of the thesis to the University of Limpopo.

Signature..

Date: 19 June 2017

Abstract

Operations Research (OR) is a scientific method for developing quantitatively well-grounded recommendations for decision making. While it is true that it uses a variety of mathematical techniques, OR has a much broader scope. It is in fact a systematic approach to solving problems, which uses one or more analytical tools in the process of analysis. Over the years, OR has evolved through different stages. This study is motivated by new real-world challenges needed for efficiency and innovation in line with the aims and objectives of OR – the science of better, as classified by the OR Society of the United Kingdom. New real-world challenges are encountered on a daily basis from problems arising in the fields of water, energy, agriculture, mining, tourism, IT development, natural phenomena, transport, climate change, economic and other societal requirements. To counter all these challenges, new techniques ought to be developed. The growth of global markets and the resulting increase in competition have highlighted the need for OR techniques to be improved. These developments, among other reasons, are an indication that new techniques are needed to improve the day-to-day running of organisations, regardless of size, type and location.

The principal aim of this study is to modify and develop new OR techniques that can be used to solve emerging problems encountered in the areas of linear programming, integer programming, mixed integer programming, network routing and travelling salesman problems. Distribution models, resource allocation models, travelling salesman problem, general linear mixed integer programming and other network problems that occur in real life, have been modelled mathematically in this thesis. Most of these models belong to the NP-hard (non-deterministic polynomial) class of difficult problems. In other words, these types of problems cannot be solved in polynomial time (P). No general purpose algorithm for these problems is known. The thesis is divided into two major areas namely: (1) network models and (2) resource allocation and distribution models. Under network models, five new techniques have been developed: the minimum weight algorithm for a non-directed network, maximum reliability route in both non-directed and directed acyclic network, minimum spanning tree with index less than two, routing through $'k'$ specified nodes, and a new heuristic to the travelling salesman problem. Under the resource allocation and distribution models section, four new models have been developed, and these are: a unified approach to solve transportation and assignment problems, a transportation branch and bound algorithm for the generalised assignment problem, a new hybrid search method over the extreme points for solving a large-scale LP model with non-negative coefficients, and a heuristic for a mixed integer program using the characteristic equation approach. In most of the nine approaches developed in the thesis, efforts were done to compare the effectiveness of the new approaches to existing techniques. Improvements in the new techniques in solving problems were noted. However, it was difficult to compare some of the new techniques to the existing ones because computational packages of the new techniques need to be developed first. This aspect will be subject matter of future research on developing these techniques further. It was concluded with strong evidence, that development of new OR techniques is a must if we are to encounter the emerging problems faced by the world today.

Key words: NP-hard problem, Network models, Reliability, Heuristic, Large-scale LP, Characteristic equation, Algorithm.

Dedication

To the entire Nyamugure family

Acknowledgments

First and foremost, I would like to thank my academic promoters: Professor Santosh Kumar, Professor 'Maseka Lesaoana and Professor Elias Munapo for their immense contributions towards this thesis. Their patience, inspiration and guidance throughout my studies, have given birth to this thesis. I am greatly indebted to them all.

I would like to thank the Almighty God for being with me throughout my existence. I am grateful to Him for everything that I have achieved in life. To my wonderful wife and lovely children, I thank you for your patience and support throughout the study period. I would also like to express my heart-felt gratitude to Mr Thandolwethu Silongwe, Doc Daniel Maposa and Doc Gerald Marewo for their assistance. To all those who provided support and offered comments, your assistance is greatly appreciated, thank you.

I would also want to thank the National University of Science and Technology (NUST) for allowing me to study at the University of Limpopo, South Africa. I would also want to thank the University of Limpopo for giving me an opportunity to study with this great University. I really felt at home whenever I entered the university premises. A special thanks also go to NRF for the special grant that they offered me. The grant enabled me to learn comfortably and I managed to attend conferences and several workshops that I might not have attended had they not extended their generous gesture. To all the staff members of the Departments of Statistics and Operations research both at NUST and UL, I salute you even though I did not mention you by name.

It would be unfair to forget the various reviewers who reviewed papers in this thesis, their comments not only helped in improving those particular papers, but also greatly improved the quality of the thesis and widened my thinking horizon. I am grateful to all of them.

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

ABC	Artificial Bee Colony
ACF	Autocorrelation Function
ACO	Ant Colony Optimisation
AHP	Analytic Hierarchy Process
ANP	Analytic Network Process
ATSP	Asymmetric Travelling Salesman Problem
BB	Branch and Bound
BC	Branch and Cut
CE	Characteristic Equation
COP	Constrained Optimisation Problem
CPM	Critical Path Method
CPU	Central Processing Unit
DE	Differential Evolution
DEMATEL	Decision-Making Trial and Evaluation Laboratory technique
DS	Decision Science
EA	Evolutionary Algorithms
FTP	Fuzzy Transportation Problem
GA	Genetic Algorithm
GAP	Generalised Assignment Problem
GIS	Geographic Information System

IP	Integer Programming
ILP	Integer Linear Programming
LB	Lower Bound
LP	Linear Programming
LIPSOL	Linear-Programming Interior-Point SOLvers
MABC	Modified Artificial Bee Colony
MAUT	Multi-Attribute Utility Theory
MCDM	Multi-Criteria Decision making Methods
MCNFP	Minimum-Cost Network Flow Problem
MIP	Mixed Integer Programming
MILP	Mixed Integer Linear Program
MRGAP	Multi-Resource Generalised Assignment Problem
MS	Management Science
MST	Minimum Spanning Tree
NP-hard	Non-deterministic Polynomial-hard
OA	Operational Analysis
OR	Operations Research
OM	Operations Management
PIP	Pure Integer Program
SMART	Simple Multi-Attribute Ranking Technique
SOMA	Self Organising Migrating Algorithm
TDVRPTW	Time-Dependent Vehicle Routing Problem with Time Window
TSP	Travelling Salesman Problem
TST	Travelling Salesman Tour
UB	Upper Bound
VRP	Vehicle Routing Problem
VRPHTW	Vehicle Routing Problem with Hard Time Windows
VRPSTW	Vehicle Routing Problem with Soft Time Windows
ZOGP	Zero-One Goal Programming

Research Outputs

The following gives a list of research outputs from this thesis.

Peer Reviewed Journal Publications

1. Santosh Kumar, Elias Munapo, 'Maseka Lesaoana and **Philimon Nyamugure** (2017). A minimum spanning tree based heuristic for the travelling salesman problem. *OPSEARCH* doi:10.1007/s12597-017-0318-5.

2. Santosh Kumar, Elias Munapo, 'Maseka Lesaoana and **Philimon Nyamugure** (2017). A Hybrid Strategy for Reducing Feasible Convex Space and the Number of Variables for Solving a Conventional Large LP Model. *International Journal of Mathematical, Engineering and Management Sciences* 2(4), 213-230. ISSN: 2455-7749.

3. Santosh Kumar, Elias Munapo, 'Maseka Lesaoana and **Philimon Nyamugure** (2016). Is the travelling salesman actually NP-hard? Chapter 3 in *Engineering and Technology: Recent Innovations and Research, (An International Edition), Editor Ashok Mathani, Intenational Research Publication House*. 37-55. ISBN-978-93-86138-06-4.

4. **Philimon Nyamugure**, Elias Munapo, 'Maseka Lesaoana and Santosh Kumar (2016). A heuristic for mixed integer program using the characteristic equation approach. *International Journal of Mathematical, Engineering and Management Sciences*. 2(1), 1-16: ISSN 2455-7749.

5. Santosh Kumar, Elias Munapo, Maseka Lesaoana and **Philimon Nyamugure** (2016). Identification and Application of Virtual Directions in a Non-Directed Network: A Labelling Method for Determination of Maximum Reliability and the Route. *Communications in Dependability and Quality Management (CDQM), An international Journal*. 19(1), 85-95.

6. Elias Munapo, Santosh Kumar, Lesaoana 'Maseka and **Philimon Nyamugure** (2016). A Minimum Spanning Tree with node index ≤ 2 . *The Australian Society for Operations Research (ASOR) Bulletin*. 34(1), 1-14. www.asor.org.au

7. Elias Munapo, Lesaoana 'Maseka, **Philimon Nyamugure** and Santosh Kumar (2015). A transportation branch and bound algorithm for solving the generalised assignment problem. *International journal System Assurance Engineering and Management*. 6(3), 217-223. DOI 10.1007/s13198-015-0343-9 Springer. http://www.springerlink.com/openurl.asp?genre=article&id-doi10.1007/s13198-015-0343-9.

8. Elias Munapo, Santosh Kumar, Lesaoana 'Maseka and **Philimon Nyamugure** (2014). Solving a large-scale LP model with non-negative coefficients: A hybrid search over the extreme points and the normal direction to the given objective function. *The Australian Society for Operations Research (ASOR) Bulletin*. 33(1), 11-24.

9. Santosh Kumar, Elias Munapo, Maseka Lesaoana and **Philimon Nyamugure** (2014). A Minimum Spanning Tree Approximation to the Routing Problem through 'K' Specified Nodes. *Journal of Economics*. 5(3), 307-312.

10. Santosh Kumar, Elias Munapo, Ozias Ncube, Caston Sigauke and **Philimon Nyamugure** (2013). A minimum weight labelling method for determination of a shortest route in a non-directed network. *International Journal*

of Systems Assurance Engineering and Management. 4(1), 13-18. DOI 10.1007/s13198-012-0140-7 Springer verlag.

Conferences

1. **Philimon Nyamugure**, Elias Munapo, Lesaoana 'Maseka and Santosh Kumar (2014). A spanning tree based approach to the Travelling Salesman Problem in a completely connected network. *Southern Africa Mathematical Sciences Association* ***(SAMSA)*** *conference,* ***Victoria Falls, Zimbabwe***.

2. **Philimon Nyamugure**, Elias Munapo, Lesaoana 'Maseka and Santosh Kumar (2015). A heuristic for solving the Travelling Salesman Problem. Faculty of Science and Agriculture Research Day, 1-2 October 2015, Extended Abstracts, Bolivia Lodge, **Polokwane, South Africa**.

Awards

1. Best PhD presenter in the School of Mathematical and Computer Sciences, Faculty of Science and Agriculture, Faculty Research Day 1-2 October 2015, Prize R7000.

2. Best Senior presenter, Faculty of Applied Sciences, National University of Science and Technology Research Day, 28 July 2016, Prize US$450.00.

Chapter 1

Introduction and Background

"To explain all nature is too difficult a task for any one man or even for any one age. It is much better to do a little with certainty and leave the rest for others that come after than to explain all things by conjecture without making sure of anything".

Isaac Newton

1.1 Introduction

Operations Research (OR) is a scientific method for developing quantitatively well-grounded recommendations for decision making. It is often considered to be a sub-field of mathematics (Agbadudu, 2006). The terms management science, operations management, operational analysis and decision science are sometimes used to describe this field. By utilising techniques and theories from other mathematical sciences such as mathematical modelling, mathematical optimisation, statistical analysis, and artificial intelligence, OR arrives at optimal or near optimal solutions to complex decision making problems. While it is true that it uses a variety of mathematical techniques, operations research has a much broader scope. It is in fact a systematic approach to solving problems, which uses one or more analytical tools in the process of analysis. It can be said to have been in existence since the beginning of mankind (Agbadudu, 2006). However, OR as a formal subject is more than 60 years old. Its origins can be traced back to the latter half of the World War I (Rajgopal, 2004). By the mid-1950s, as OR assumed the mantle of a profession, it began to adopt its own methodology that included a variety of emerging mathematical methods such as linear programming, inventory theory, search and set theory and queuing theory. Although OR is a distinct discipline in its own right, it has also become an integral part of the engineering profession because of its emphasis on human-technology interaction and also because of its focus on practical applications.

Over the years, OR has evolved through different stages. Magee (1973) reviewed the phases through which OR has developed. According to the scholar, OR has gone through three phases of growth: the primitive phase, the academic phase and the maturing phase. The **Primitive Phase** is between 1940 and 1960. At this stage, the problem solvers were interested in practical operational problems. These problems were well-defined and capable of being handled by the smaller, less sophisticated computers available. Furthermore, OR was in the process of developing into a separate professional field and the theoretical foundations of the discipline developed rapidly. However, only very few universities offered formal training in operations research during that time.

The **Academic Phase** developed in the early 1960s. The number of universities offering programmes in operations research grew over 500 percent. Magee (1973) pointed out that in this phase, people with some OR experience started featuring at the higher corporate levels in private enterprises. The increasing speed and availability of computers were of great help during this time. Magee (1973) noted that research during this phase tended to be academic, that is, it was more concerned with developing theory rather than finding workable applications. It was also during this time, that the limitations of OR became evident. Some of these limitations are listed in Section 1.4.

Maturing Phase was described by Magee (1973) as a time when balance between theory and practice was obtained. He argued that even though evidence of such concerns were noted several years before, the real thrust towards practice and applications did not come until the 1970s.

1.2 Motivation of the Study

This study was motivated by new real-world challenges needed for efficiency and innovation in line with the aims and objectives of OR – the science of better, as classified by the OR Society of the United Kingdom. According to Hammer and Champy (1993), process re-engineering is the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical contemporary modern measures of performance, such as cost, quality, service and speed. In this study, we took the process of re-engineering a step further by developing and modifying existing OR models. New real-world challenges arise on a daily basis from problems in water, energy, agriculture, mining, tourism, IT development, natural phenomena, transport, climate change, economic and other societal requirements. To counter all these challenges, new techniques ought to be developed.

Kumar and Munapo (2012) in their study entitled “Some lateral ideas and their applications for developing new solution procedures for a pure integer programming model”, came up with a model that has added value and greatly contributed to the field of integer programming. Munapo et al. (2008) modified the critical path method (CPM) by using the shortest route algorithm to determine the optimal crash limits for various activities in a CPM network. Their work is currently being used by several construction companies when scheduling activities for their projects, and large sums of money are being saved by using this technique (Kumar and Munapo, 2012). Nyamugure et al. (2011) successfully designed a new algorithm that combined different process capability indices and proposed a holistic algorithm that addressed the entire quality of a production process.

These developments, among other reasons, are an indication that new techniques are needed to improve the day-to-day running of organisations, regardless of size, type and location. Several methods have been used to approximate the travelling salesman problem (TSP) but the challenge is that these methods do not inform on the quality of the solution with respect to the optimal solution. As a result there is need for new techniques that address and present the supporting theory that makes the proposed algorithm more efficient and more user-friendly than existing methods.
Degeneracy in a linear programming (LP) model can cause difficulties, as the value of the objective function may not improve in successive degenerate iterations. Sometimes a solution may be optimal but the test of optimality fails to recognise optimality of that solution due to wrong selection of degenerate variables. Since the transportation and assignment models are degenerate LP models, where order of degeneracy varies from 1 to n in the context of a LP model, special methods were developed to deal with these special degenerate models. A feasible solution in a balanced assignment model of order n is a degenerate solution by order $(n-1)$ in the context of a LP model (Munapo et al., 2012). Perhaps the best known, most widely used, and most written about technique for solving the assignment problem is the Hungarian method, which

was first introduced by (Kuhn, 1955), with variants developed by Balinski and Gomory (1964). This research is motivated by Munapo et al. (2012) who came up with a unified approach to solve both transportation and assignment models, which is independent of degeneracy. The proposed unified approach takes advantage of the Hungarian method of assignment and makes it more versatile so that degeneracy can be handled. This unified approach, is applicable to both assignment and transportation problems. Further, the process does not depend on the number of allocated cells which in the transportation method must be equal to $(m + n - 1)$ in independent cells. The new unified method is likely to prove more efficient in solving all transportation and assignment problems.

1.3 Purpose of the Study

1.3.1 Hypothesis

The central hypothesis of the study is that the occurrences of new problems and challenges the world is facing today demand that we develop, improve and modify existing models in order to counter these challenges.

1.3.2 Research aim

The principal aim of this study is to modify and develop new OR techniques that can be used to solve emerging problems encountered in the areas of linear programming, integer programming, mixed integer programming, network routing and travelling salesman problems.

1.3.3 Objectives

The specific objectives of this research are to:

1. Develop new techniques for solving mixed integer linear programming problems and a new approach to the transportation problem by modifying the assignment approach, which is capable of dealing with the degenerate problems.

2. Modify and improve existing shortest route model and maximum reliability route in a non-directed network in order to derive the optimal solution faster, cheaper and more accurately.

3. Combine holistically the minimum spanning tree algorithm and node index technique to form a new heuristic to the travelling salesman problem.

4. Apply the modified and developed models to solve distribution problems, resource allocation problems, travelling salesman problem, reliability problems and shortest route problems.

1.4 Research Problem

OR encompasses a wide range of problem solving techniques and methods, which are applied in pursuit of improved decision making and efficiency (Mangan et al., 2004). Such techniques and methods include simulation, mathematical optimisation, queueing theory and other stochastic-process models, Markov decision processes, econometric methods, data envelopment analysis, neural networks, expert systems, decision analysis, and the analytic hierarchy process. Nearly all of these techniques involve the construction of mathematical models that attempt to describe the system. Because of the computational and statistical nature of most of these fields, OR also has strong ties to computer science and analytics. Today, operations research is a mature, well-developed field with a sophisticated array of techniques that are used routinely to solve problems in a wide range of application areas.

As the OR field evolved, there is also an increase in the development of mathematical models that were used to model, improve, and even optimise real-world systems. As the global environment becomes fiercely competitive, OR has gained significance in applications. The growth of global markets and the resulting increase in competition have highlighted the need for OR techniques to be improved as well. During the twentieth century the pace of development of fundamentally new OR methodologies somewhat slowed down (Kirby, 2007). However, there has been a rapid expansion in the breadth of problem areas to which OR has been applied, and in the magnitudes of the problems that can be addressed using OR methodologies. The importance and numerous applications of OR models have led and stimulated research in this area. For a long time, OR and Artificial Intelligence could be seen as alternative approaches to solve real life optimisation problems such as scheduling or resource allocation. It now seems clear that we can have the best of both worlds by coming up with some hybrid techniques that combine the two fields. There are several emerging technologies which contain some of the most prominent ongoing developments, advances and innovations in various fields of modern technology. Some of these fields include: closed door ecology systems, 3D printing and bio-printing, femtotechnology and picotechnology, magnetic refrigeration, energy harvesting, wireless energy transfer, genetic engineering and artificial gravity (Aggarwa and Khetrapal, 2014).

Distribution models, resource allocation models, travelling salesman problem, general linear mixed integer programming (MIP) and other network problems that occur in real life have been modelled mathematically. Most of these models belong to the NP-hard (non-deterministic polynomial) class of difficult problems (Baker et al., 1975). No general purpose algorithm for these problems is known. In other words, these types of problems cannot be solved in polynomial time (P). The importance of resolving NP-hard problems is reflected by the fact that the problem of P versus NP is one of the seven Millennium Prize Problems in mathematics that were stated by the Clay Mathematics Institute in 2000 (Devlin, 2003; Carlson et al., 2006; Wiles, 2006). A correct solution to any of the seven problems will earn a US $1 million prize (also called a Millennium Prize) being awarded by the Institute. Out of the seven very difficult problems, only one was solved by the Russian mathematician Grigori Perelman in 2003 (Wiles, 2006). For more on the latest developments on NP-hard models, see Wang et al. (2013), Chagwiza et al. (2015), Lefever et al., (2016), and Polyakovskiy et al. (2016). This study has addressed a few NP-hard problems and tried to come up with solutions to these types of problems.

With all these technological advancements, there is need to continuously design new OR models and modify existing techniques in order to solve the ever emerging challenges our world is facing. The spread of infectious diseases like HIV/AIDS, malaria and currently Ebola and the Zika virus, depends on interactions of different types of diseases, which need OR techniques to understand and possibly come up with solutions. Occurrences of natural disasters and recent catastrophic events such as earthquakes, tsunamis, heat waves, landslides, forest fires, drought and floods, all need a holistic approach to understand, predict and mitigate the impact of these events. Current OR techniques need to be extended or revamped in order to deal with these new emerging problems our world is facing.

This thesis is an attempt to address problems that can be classified according to the selected areas in the next five subsections.

1.4.1 Routing models in non-directed networks

The shortest path problem determines the shortest path between two vertices of a graph such that the sum of the weights of the constituent edges is minimised. Many industrial problems can be formulated and analysed as a shortest path problem or its variants in a network. This has been of interest to researchers and practitioners in many network related disciplines like transportation, logistics, vehicle routing, airport tours, installation of power cables, operations research, geographic information system (GIS) technology, and emergency services. Many approaches have been developed to find the shortest path from a given node to all other nodes and appropriate computer codes have been developed for implementation of their algorithms. The problem of shortest route has been studied in directed as well as in non-directed networks (Dijkstra, 1959; Dantzig 1960; Zhan, 1997; Zhan and Noon, 1998; among others).

Many challenges with existing methods remain unsolved to date. Some of the challenges include being stuck into a local optimum solution instead of global optimum solution, and challenges created by non-directed graphs and failure to obtain an optimal solution. In this research a new approach for determining a shortest path for a non-directed network will be formulated. A minimum spanning tree approach to determine a route through 'k' specified nodes is also formulated. The path through 'k' specified nodes is a difficult problem for which no good solution procedure is known (Gomes et al., 2015). The proposed method determines an optimal or a near optimal path.

1.4.2 Minimum spanning tree based models for solving some NP-hard problems

Determination of the travelling salesman problem (TSP) is an NP-hard combinatorial optimisation problem (Kahng and Reda, 2004) that has applications in OR and many other fields, for example, computer science, genetics, electronics and logistics. The TSP finds a path of moving from an origin node to all the other nodes in the network and return back to the origin in such a way that each and every node is visited once and the total distance travelled is minimal. Several methods have been used to approximate the TSP but the problem is that these methods do not tell us about the quality of the solution with respect to the optimal solution.

In computational complexity theory the TSP belongs to the class of NP-complete problems, which means that in the worst-case running time for the TSP may increase exponentially with the number of cities (Mitchell, 2001; Nadef, 2002). Currently we are not aware of any efficient exact method for the TSP model. Heuristics have been used to approximate the TSP, but these heuristics do not tell us about the quality of the solution with respect to the optimal solution (Wolsey, 1980; Berman and Karpinski, 2006; Cowen, 2011; Razali and Geraghty, 2011). The TSP has so many variants and applications in real life that it has demanded attention of many researchers (Gutin and Punnen, 2006).

This research presents a heuristic to find the travelling salesman tour (TST) in a connected network through the minimum spanning tree (MST), thus converting the NP-hard problem to a relatively easier form.

1.4.3 Transportation and assignment problems

The generalised assignment problem (GAP) is the problem of assigning n jobs to m agents in such a way that the total cost is minimal and that each job is assigned to exactly one agent and the agent's capacity is also satisfied. GAP is an NP-hard problem and many approaches have been proposed in the past 50 years. The GAP can be relaxed to become an ordinary transportation problem. GAP is a branch and bound technique in which the sub-problems are solved by the available efficient transportation techniques rather than the usual simplex based approaches. A transportation model is easy to handle and efficient solution methods such as network approaches have been formulated. The GAP model can also be treated as a general case of the assignment problem in which the number of jobs and agents are equal in size, and the cost associated with each job-agent combination may have different values. GAP has many applications in real life and these include vehicle routing (Toth and Vigo, 2001), resource allocation (Winston and Venkataramanan, 2003), supply chain (Yagiura, 2006), machine scheduling and location, among others. In this research a transportation branch and bound algorithm for solving the GAP will be developed. This is a branch and bound (BB) technique in which the sub-problems are solved by the available efficient transportation techniques rather than the usual simplex based approaches. This technique also selects branching variables that minimise the number of sub-problems.

1.4.4 Large-scale linear programming

Many solution procedures have been developed for large-scale LP models (Kachiyan, 1979; Karmarkar, 1984) and many variants of these approaches have been discussed (Roos et al., 2005). Recently Munapo and Kumar (2013), considered a LP model with non-negative coefficients, and developed an iterative procedure to solve a large-scale LP by transforming the given 'n' variable LP to a '2' variable LP. Computational experiments indicated that their approach performed better with regard to a large number of randomly generated large LP problems. However, more needs to be done to improve their approach. As a result, a large-scale LP model with non-negative coefficients was reconsidered under a new strategy, that is, an iterative hybrid process. The approach uses the conventional simplex iterations for search along the extreme points of the convex region, generates an interior point using these extreme points, and moves from the interior point in the direction of the normal to the given objective function hyper-plane until an optimal solution has been identified. Authors have reconsidered a conventional LP model with no restriction on coefficients.

1.4.5 Integer programming and mixed integer programming

An integer linear program is a linear program which is further constrained by integer restrictions on some or all variables. When all variables are integer restricted, it is called a pure integer program (PIP) model, and when only some of the variables are restricted to integer values, it becomes a mixed integer program model. Integer programming (IP) models frequently arise in human resource planning, facility location, assignment problems, production planning, time-tabling, warehouse location, scheduling and capital budgeting, just to mention a few.

While most LP problems can be solved in polynomial time, PIP and mixed integer program are NP-complete problems, which have no known polynomial time algorithms to solve them. Generally, MIP problems have been solved using the LP-based BB solvers or with stochastic search-based solvers (Razali and Geraghty, 2011). In reality MIP solvers have implemented more sophisticated versions denoted by branch and cut (BC) algorithms (Sen and Sherali, 2006). With the increase in the application of both PIP and MIP, it is of paramount importance to generate methods that are capable of finding a global optimal solution. The major disadvantages of existing methods are that there are round off errors, creation or emergence of many sub-problems (branches), the time taken to obtain the optimal solution and failure to obtain global optimal solutions. All these shortcomings justify the need to find better approaches for MIP problems.

In this research a heuristic method for MIP using the characteristic equation hsa been formulated. The proposed method generates a good feasible solution with bounds, and it eliminates rounding off errors and dealing with sub-problems as is commonly required in existing BB methods.

All these problems have resulted in the need to develop new OR techniques to address the shortfalls of the current methodologies and try to solve emerging problems in order to improve the efficiency with which organisations operate.

1.5 Limitations of OR

The following are some of the limitations of OR:

- In the quantitative analysis of operations research, certain assumptions and estimates are made for assigning quantitative values to factors involved. If such estimates are wrong, the result would be equally misleading.
- Most management problems do not lend themselves to quantitative measurement and analysis. Intangible factors of any problem concerning human behaviour cannot be quantified accurately and all the patterns of relationships among the factors may not be covered. Accordingly, the outward appearance of scientific accuracy through the use of numbers and equations becomes unrealistic.
- The quantitative methods of OR are costly in many cases, elaborate and sophisticated in nature. Although complex problems are fit for analysis by tools of operations research, relatively simple problems have no economic justification for this type of quantitative analysis.
- Knowledge of some concepts of mathematics and statistics is a prerequisite for the adoption of quantitative analysis by managers. According to the present training and experience of most managers, the actual use of these tools may have natural hesitation by some managers.

- OR is not a substitute to the entire process of decision making and it does not relieve managers from their task of decision making. In one phase of decision making viz., selection of best solution through the evaluation of alternatives, OR comes into the picture.

These limitations justify the need to continuously develop and modify existing OR techniques in order to come up with new models that can solve these problems holistically. Despite the relevance of OR to organisations, Griffen (1987) argued that quantitative techniques alone cannot fully account for intangible or qualitative factors in decision making. Qualitative or intangible factors are factors that are difficult to measure numerically. Mangan et al. (2004) highlighted the benefits which can result from combining qualitative and quantitative methodologies in logistic research. Modern challenges associated with a global economy and the growth of technology have increased the complexity of the business environment. Modern corporations often strive to serve a global, rather than a regional customer base, hence they must be prepared to face worldwide competition.

1.6 Significance of the Study

The use of hybrid methods, (that are formed by combining existing techniques) is in response to the rising number of problems that need to be solved by OR. Under the unpredictable and turbulence environment, classical and traditional approaches are partially able to obtain a complete solution with certain degree of satisfaction. The modified technique must be generic, flexible, robust and versatile for solving complex problems. The use of these new methods can provide useful insights where the analytical approach has a shortfall. Therefore, new OR methods are required to handle these problems holistically.

The use of new and improved methods improves the way organisations run their business and saves millions of dollars. Two recent examples are: (1) the world's largest logistics company redesigned its overnight delivery network which was estimated to yield savings of more than 270 million United States dollars (2) and a global automobile manufacturer streamlined its prototype vehicle testing, saving 250 million United States dollars annually. These examples and more are available on the internet (Informs, 2004).

The results of this research will serve as a benchmark for comparison with other models that are currently in use. If scientists put their heads together we strongly feel that the results of their studies will make this world a better place to live in.

1.7 Scientific Contributions of the Study

This study aims to strengthen interdisciplinary ties by combining different techniques from different fields and to initiate new joint ventures in research and education. The major contribution of this thesis is in coming up with new techniques and methods of solving emerging problems and applying these new techniques to real life situations. The specific contributions are as follows:

1. Developed a new labelling method for a probabilistic directed network that identifies existence of virtual directions in a non-directed network. These directions are used for developing a labelling method for the non-directed network. The approach can easily be used for finding a minimum delay path, widest communication band width etc., which have applications in operations research, robotics and transportation of communication signals.

2. Developed a minimum spanning tree approach to determine a route through $'k'$ specified nodes. The path through $'k'$ specified nodes is a difficult prob-

lem for which no good solution procedure was known before. The proposed method determines the route, which may be either an optimal or a near optimal path and has several applications in telecommunication, distribution management and transportation.

3. Developed a new method to find the minimum spanning tree path, such that the node index of each node is less than or equal to 2. The method obtained is such that, such a spanning tree may can be used to determine the NP-hard travelling salesman tour which have applications in Computer Wiring, DNA, Logistics, Baseball etc.

4. Developed a transportation branch and bound algorithm for solving the generalised assignment problem. This is a branch and bound technique in which the sub-problems are solved by the available efficient transportation techniques rather than the usual simplex based approaches. This technique also select branching variables that minimise the number of sub-problems.

5. Modified and created a hybrid search process for a large-scale LP problem. The hybrid approach uses the normal simplex iterations for search over the extreme points of the convex region, then generates an interior point using these extreme points, and moves from the interior point in a known direction, which is normal to the given objective function. This approach is suitable for a large-scale LP and it has several real life applications.

6. Formulated a new heuristic for solving MIP problems using the characteristic equation. The new method does not create round off errors which leads to non-optimal or non-feasible solutions. The heuristic also does not create sub-problems as is in the case with BB or BC methods. It also searches the optimal solution using the simplex algorithm, but moves over the integer polyhedron.

1.8 Outline of the Thesis

The structure of this thesis is arranged such that Chapter 1 gives an introduction to the research study. This chapter also presents a comprehensive research problem, motivation of the study, hypothesis of the study, principal aim, objectives of the study, significance of the study and scientific contributions of the study. The Chapter highlights the origins of OR, the advantages of using OR, the growth of the subject since the 1950s and its numerous applications. It also motivates for the need to continuously modify and develop new techniques in order to solve the ever-emerging challenges faced in the world.

Chapter 2 reviews the general literature on the research areas that are covered in this thesis, that is, routing models, travelling salesman problem, transportation and assignment problems, large-scale linear programming problem and mixed integer programming problem. The Chapter provides the literature on the developments of the OR field, and the capabilities and challenges that the current models are facing in solving the emerging problems. The Chapter also reviews current OR research and how it is benefiting mankind.

The technical chapters of this thesis are arranged such that they fall into two major sections. Section I is on **Network models**, and consists of three chapters, namely (1) Routing models in non-directed networks, (2) Minimum spanning tree based models for solving some NP-hard problems, and (3) Travelling salesman problem. Section II is on **Resource allocation and distribution models**, and consists of two chapters, namely (1) Transportation and assignment problems, and (2) Linear programming based models for solving some NP-hard problems.

Chapters 3 to 7 are based on 9 papers as per details given as foot note on pages 18, 19 and 20 and have been submitted individually for publication purposes.

Each of these chapters explains how the new techniques have been formulated. A detailed analysis of the methodology, data analysis, results, conclusion and recommendations will be done in each of these chapters. Although these chapters are self-contained, there will be consistency in notation throughout the thesis so as not to confuse the reader. Also the chapters will be written in such a way that if there is a technique or method that is used in another chapter, then the two chapters will follow each other.

Chapter 3: **Routing Models in Non-directed networks** has been divided into two sections. Section 1[1] describes how a new minimum weight labelling method for determining the shortest route is developed. It also gives a numerical example to illustrate the new method. Its advantages are also outlined. Section 2[2] presents a new and efficient labelling approach for the determination of a maximum reliability route in a non-directed network. The proposed method finds the reliability and the corresponding path from an origin node to all other nodes. Practical use includes waste management, where recycling reduces the bulk of solid waste and provides cheap resource to industry. Similarly, information recycling is intended to minimise unnecessary computations when that information can be extracted by earlier computations.

Chapter 4: **Minimum Spanning Tree based Models for solving some NP-hard Problems** is divided into two sections. Section 1[3] outlines how a new method to find the minimum spanning tree path such that the node index of each node is less than or equal to 2, was developed. The method developed

[1]A minimum weight labelling method for determination of a shortest route in a non-directed network (Kumar et al., 2013) *International Journal of Systems Assurance Engineering and Management*. 4(1),13-18. 2013. DOI 10.1007/s13198-012-0140-7 Springer verlag.

[2]Identification and Application of Virtual Directions in a Non-Directed Network: A Labelling Method for Determination of Maximum Reliability and the Route (Kumar et al., 2016). *Communications in Dependability and Quality Management (CDQM) Journal* 19(1):85-95. 2016

[3]A Minimum Spanning Tree with node index $\leq$2 (Munapo et al., 2016). *The Australian Society for Operations Research (ASOR) Bulletin* 34(1): 1-14. 2016

is then used to determine the NP-hard travelling salesman problem. Section 2[1] develops a minimum spanning tree approach to determine a route through $'k'$ specified nodes. In this section, the concept of the specified nodes has been discussed in connection with the shortest route from a given origin node to the destination node. For a feasible solution, one has to visit the specified node within the specified time window. This simple mathematical concept has applications in patient routing in the hospitals, where time windows arise due to availability of beds, theatre, doctor, radiology results etc.

Chapter 5: **Travelling salesman problem**, is divided into two sections. Section 1 describes the method of obtaining the travelling salesman tour (TST) through the MST technique. The proposed method does not generate sub-problems that can explode as is the case with most of the branch and bound related methods. The method can result either in an optimal solution or bounds on the TSP tour. Section 2[2] describes how a new heuristic to the TSP is formulated. The new method is then compared to another leading TSP heuristic that used the Christofides algorithm (Cowen, 2011). The new method was found to produce better results than those obtained using the Christofides algorithm.

Chapter 6: **Transportation and Assignment Problem**[3] deals with the development of a transportation branch and bound algorithm for solving the generalised assignment problem. The proposed approach has the advantage that the individual knapsack objective values can be found independently, thus allowing the much needed use of parallel processors. The sub-problems resulting

[1]A Minimum Spanning Tree Approximation to the Routing Problem through $'k'$ Specified Nodes (Kumar et al., 2014). *Journal of Economics*, 5(3), 307-312. 2014

[2]A minimum spanning tree based heuristic for the travelling salesman problem in a connected network (Kumar et al. 2017(a)). Revised manuscript submitted to **Opsearch**, 2017

[3]A transportation branch and bound algorithm for solving the generalised assignment problem (Munapo et al., 2015).*International Journal of System Assurance Engineering and Management* 2015, 6(3):217-223. DOI 10.1007/s13198-015-0343-9 Springer. http://www.springerlink.com/openurl.asp?genre=article&id-doi10.1007/s13198-015-0343-9.

from the search trees are transportation models that can be solved efficiently by the available network approaches. The sub-problems that result from the usual branch and bound related approaches are NP-hard integer models which are very difficult to solve. The only disadvantage to this approach is that like the simplex based approaches it is also not spared by degeneracy. The degeneracy drawback can be alleviated by noting all alternate optimal solutions at every node and then branch in such a way that the objective value does not remain static.

Chapter 7: **Large-Scale Linear Programming**, is divided into two sections. Section 1[1] modifies and creates a hybrid search process for a large-scale LP problem. The hybrid approach uses the normal simplex iterations for search over the extreme points of the convex region, then generates an interior point using these extreme points, and moves from the interior point in a known direction, which is normal to the given objective function. This approach is suitable for a large-scale LP with coefficients of the form greater or equal to zero, and it has several real life applications. A conventional LP with no restriction on coefficients is reconsidered with strategies to reduce feasible space and number of constraints.

In this same Section[2] strategies for reducing the feasible space, and the number of variables when solving a large-scale LP model is also discussed. When the feasible space is reduced in a way discussed in this Section, the optimal solution can be found faster than in the current methods.

[1]Solving a large-scale LP model with non-negative coefficients: A hybrid search over the extreme points and the normal direction to the given objective function (Munapo et al., 2014). *The Australian Society for Operations Research (ASOR) Bulletin* 2014 33(1), 11-24.

[2]Strategies for reducing feasible space and the number of variables for solving a LP model (Kumar et al., 2017(b)). To appear in *International Journal of Mathematical, Engineering and Management Sciences*

Section 2[1] describes a new heuristic for solving MIP problems using the characteristic equation was developed. The new method has several advantages over existing ones in that it does not create round off errors which leads to non-optimal or non-feasible solutions, and also it does not create sub-problems as is in the case of branch and bound (BB) or branch and cut (BC) methods. A small and large-scale MIP problems were solved and compared to the BB and the BC methods respectively and the results clearly shows that our method performs better. The only drawback is that computational experiments on these new methods need to done.

Chapter 8: **Summary, Conclusions and Recommendations**, summarises and concludes the thesis, and it also gives overall recommendations of the study. The Chapter also outlines the limitations of the study and covers areas of future research directions.

[1]A heuristic for mixed integer program using the characteristic equation (Nyamugure et al., 2016). *International Journal of Mathematical, Engineering and Management Sciences*. 2(1):1-16: ISSN 2455-7749

Chapter 2

General Literature Review

"Research is to see what everybody else has seen, and to think what nobody else has thought".

Albert Szent-Gyorgyi

2.1 Introduction

This thesis presents research carried out in two major areas of: (a) network models, and (b) resource allocation and distribution models. Literature review is therefore limited to the models that have been developed in this thesis. The review highlights the developments that have been carried out in these two types of models, their application, limitations and areas that need improvements or further developments.

Operations research encompasses a wide range of problem-solving techniques and methods applied in the pursuit of improved decision making and efficiency, such as simulation, mathematical optimisation, queuing theory, stochastic-process models, Markov decision processes, econometric methods, data envelopment analysis, neural networks, expert systems, decision analysis, and the analytic hierarchy process (Fasika et al., 2013; Jinn-Tsong et al., 2013).

Most researchers in the area of big data solve a problem by combining different techniques (Gradisar and Trkman, 2013). Emerging problems in most cases are so complicated that a single method might not solve the problem to optimality. When analysing multi-criteria decision making (MCDM) methods, Velasquez and Hester (2013) concurred with this idea, and established that the newest trend with respect to MCDM method use, is to combine two or more methods to make up for shortcomings in any particular method. Konidari and Mavrakis (2007) utilised several methods to evaluate climate change mitigation policy instruments. In addition to utilising analytic hierarchy process (AHP) to define coefficients for criteria, the researchers used a combination of multi-attribute utility theory (MAUT) and a simple multi-attribute ranking technique (SMART) process to assign grades to the instruments. Zabeo et al. (2011) assessed the risk and vulnerability of soil contamination in Europe by selecting a vulnerability assessment framework. They did this by combining multi-criteria decision analysis techniques and spatial analysis.

2.2 Network Models

Kumar et al. (2013) proposed a new minimum weight labelling method for determining the shortest route in a non-directed network from a source node to a destination node. Kumar et al. (2014) developed a minimum spanning tree method for the determination of a route through $'k'$ specified nodes in a connected network with n nodes, where $0 \leq k \leq n-2$. The strength of the new approach lies in the fact that the proposed method provides the exact TSP optimal tour through the minimum spanning tree (MST), which is computationally easy to solve and the iteration used in the present context does not explode, as is the case with most available branch and bound type approaches.

Gnedenko et al. (1999) described a system as a directed network consisting of nodes and arcs where one node is defined as the source and another node is defined as the sink. Each component of the network is identified as an arc passing from one node to another. A failure of a component is equivalent to an arc being removed or cut from the network. The system is successful if there exists a successful path from the source to the sink. The system fails if no such path exists. The reliability of the system is the probability that there exist at least one successful path from the source to the sink. Leitch (1995) stated that the structure function of network reliability can be constructed with knowledge of the set of minimal cuts of the network. If C is a set of components comprising a minimal cut, then the event that all components in the cut will fail is $\prod(1 - X_i)$ and the event that all the components in the cut works is $1 - \prod(1 - p_i)$, where p_i is the probability that component i is working. If the network has $C_1, C_2, ..., C_k$ collections of minimal cuts, then the reliability $R_C(X)$ that the network is functioning (if it functions only if all cuts function) is given by:

$$R_C(X) = \left[1 - \prod_{i \in C_1} (1 - X_i)\right] \left[1 - \prod_{i \in C_2} (1 - X_i)\right] ...... \left[1 - \prod_{i \in C_k} p_i \, (1 - X_i)\right] \quad (2.1)$$

The reliability upper bound is obtained by computing the probability that at least one minimal path is successful with the added assumption that paths fail independently. The reliability upper bound R_U, according to Leitch (1995), is given by:

$$R_U(X) = 1 - \left[1 - \prod_{i \in P_1} p_i\right] \left[1 - \prod_{i \in P_2} p_i\right] ...... \left[1 - \prod_{i \in P_k} p_i\right] \quad (2.2)$$

where $P_1, P_2, ..., P_k$ are the minimal cuts probabilities. The reliability lower bound $R_L(X)$ is obtained by computing the probability that every minimal cut is successful, with the added assumption that paths fail independently and is given by:

$$R_L(X) = \left[1 - \prod_{i \in C_1} (1 - q_i)\right] \left[1 - \prod_{i \in C_2} (1 - q_i)\right] ...... \left[1 - \prod_{i \in C_k} p_i (1 - q_i)\right] \quad (2.3)$$

where $C_1, C_2, ..., C_k$ are the minimal cuts probabilities of the network.

Ant colony optimisation (ACO) is a heuristic algorithm which has been proven to be successful in solving the travelling salesman problem (TSP). It has been applied to a number of combinatorial optimisation problems, and is also considered as one of the high performance computing methods for the TSP (Hlaing and Khine, 2011). According to the researchers, ACO has a very good search capability for optimisation problems, but it still remains a computational bottleneck in that the ACO algorithm takes too much time to convergence. In trying to find an optimal solution for the TSP, ACO also gets trapped in local optima instead of converging to the global optima. They proposed an improved ACO algorithm by adopting a candidate set strategy to increase the convergence speed. They also included a dynamic updating rule for the heuristic parameter, based on entropy to improve the performance of their algorithm in solving the TSP. Their new method was tested on benchmark problems and their results showed that the proposed algorithm performed better than the conventional ACO algorithm.

Stutzle and Doringo (1999) gave an overview of the available ACO algorithms for the TSP. They highlighted that the first ACO algorithm called Ant System has been applied to the TSP and several improvements of the basic Ant algorithm have been proposed. When studying ACO algorithms for the TSP, Stutzle and Doringo (1999) noted that after all the ants have constructed their tours, the pheromone trails are updated. This is done by first lowering the pheromone strength on all arcs by a constant factor, and then allowing each ant to add a pheromone on the arcs it has visited. This is achieved using the following equation:

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{2.4}$$

where $0 < \rho \leq 1$ is the pheromone trail evaporator, $\Delta\tau_{ij}^{k}(t)$ is the amount of pheromone ant k puts on the arcs it has visited. Strutzle and Doringo (1999) then highlighted that the global best tour was used to update the pheromone trails. The global best solution which gives the strongest feedback, is given weight w. The r^{th} best ant of the current iteration contributes to pheromone updating with a weight given by $max\{0, w-r\}$. Thus, the modified updated rule was then given by:

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \sum_{k=1}^{m}(w-r).\Delta\tau_{ij}^{k}(t) + w.\Delta\tau ij^{gb}(t) \tag{2.5}$$

where $\Delta\tau_{ij}^{r}(t) = i/L^{r}(t)$ and $\Delta\tau_{ij}^{gb}(t) = 1/L^{gb}$, $L^{r}(t)$ is the length of the r^{th} ant's tour and L^{gb} is the length of the global tour. According to Strutzle and Doringo (1999), the modified updated rule (equation (2.5)) proved to be superior to the genetic algorithm and the simulated annealing procedures.

2.3 Resource Allocation and Distribution Models

He et al. (2015) developed a mixed integer linear program (MILP) model that addressed the dynamic resource allocation problem for transportation evacuation planning on large-scale networks. The model is built on the earliest arrival flow formulation that significantly reduced the problem size. In their study, the MILP model was decomposed into two sub-problems: the restricted master problem, which first identifies a feasible dynamic resource allocation plan; and the auxiliary problem, which models the dynamic traffic assignment in the evacuation network given a resource plan. Their results showed that the decomposition algorithm can solve problems to optimality efficiently on a large-scale network.

A wide range of problems can be modelled as mixed integer programming (MIP) problems using standard formulation techniques. However, in some cases the resulting MIP problem can be either too complicated or too large to be effectively solved by the current solvers (Vielma, 2015). In such cases either new MIP models need to be developed or existing methods need to be combined holistically in order to solve those problems.

Integer programming (IP) models frequently arise in human resource planning, facility location, assignment problems, production planning, time-tabling, warehouse location, scheduling and capital budgeting, just to mention a few. While most LP problems can be solved in polynomial time, pure integer programming (PIP) and MIP problems are NP-complete for which there are no known polynomial time algorithms to solve them (Williams, 2009). Generally, MIP problems have been solved using the LP-based branch and bound (BB) solvers or with stochastic search-based solvers (Vielma et al., 2007). The advantage of the first approach (BB solvers) is that it provides rigorous lower and upper bounds on the solution, which in turn provides the optimal solution. During the search the upper and lower bounds are used to prune branches of the tree (Mavrotas and Diakoulaki, 1998). BB algorithms, however, may lead to unwieldy situation due to large number of sub-problems that may have to be solved, particularly when the LP relaxation is poor. In reality MIP solvers have implemented more sophisticated versions denoted by branch and cut (BC) algorithms (Razali and Geraghty, 2011). In these algorithms, valid inequalities denoted by cutting planes are added to the linear relaxations in order to reduce the size of the feasible space without eliminating any feasible integer solution. One of the objectives of this research is to come up with a new method for solving a MIP, hoping that it may provide insight into the scope of MIP applications.

Li et al. (2008) developed an improved genetic algorithm of the vehicle routing problem (VRP) that they applied to the distribution of fruits and vegetables. Their method included the vehicle routing problem with hard time windows (VRPHTW) and vehicle routing problem with soft time windows (VRPSTW). According to Li et al. (2008), the mathematical model based on the total lowest cost for optimising target C were formulated as follows:

$$\min C = a_0 m + \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{m} a_1 \delta_{ij} d_{ij} x_{ijk} + \sum_{j=1}^{n} a_2(T_j).(\max\{(T_{Ej} - T_j, 0\} + \max\{(T_j - T_{Lj}), 0\} \tag{2.6}$$

with x_{ijk}={1, if vehicle k travels from i to j; 0, otherwise}, a_0 is fixed cost per vehicle, a_1 is the running unit cost per vehicle and $a_2(T_j)$ is unit penalty cost if the time goes beyond the time windows, q_i is demand of customer i, δ_{ij} represents evaluation coefficient of road surface evenness between customer i and customer j, d_{ij} is distance between customer i and customer j, $[T_{Ej}, T_{Lj}]$ denotes time windows of receiving for customer j, and T_j is travel time between customer i and customer$j$, the variable m is the number of delivery vehicles and n is the number of customers.

The model (2.6) was improved using the sweep algorithm by dividing the customers into different groups and ensuring that each group meets all of the constraints conditions (Li and Guo, 2010). A penalty strategy was introduced to define the fitness function Z as:

$$\begin{aligned} Z &= C + M_1 \sum_{k=1}^{m} max(\sum_{i=1}^{k}\sum_{j=1}^{k} x_{ijk}q_i - Q,\ 0) + M_2 \sum_{k=1}^{m} max(\sum_{i=1}^{k}\sum_{j=1}^{k} x_{ijk}d_i - L),\ 0) \\ &= a_0 m + \sum_{j=0}^{n}\sum_{k=1}^{m} a_1\delta_{ij}d_{ij}x_{ijk} + \sum_{j=1}^{n} a_2(T_j).(\max\{(T_{Ej} - T_j, 0\} \\ &+ \max\{(T_j - T_{Lj}), 0\} + M_1 \sum_{k=1}^{m} max(\sum_{i=1}^{k}\sum_{j=1}^{k} x_{ijk}q_i - Q,\ 0) \\ &+ M_2 \sum_{k=1}^{m} max(\sum_{i=1}^{k}\sum_{j=1}^{k} x_{ijk}d_i - L,\ 0) \end{aligned} \tag{2.7}$$

where C is the optimising target of the proposed model. In the improved Genetic model (2.7), penalty factors were added where M_1 is the penalty factor for being overweight and M_2 is the penalty factor for being over the maximum travel distance. Li et al. (2008) concluded that the improved genetic algorithm was superior to four other methods that they made comparison to. As proposed by (Calvete et al., 2007), the VRPHTW and the VRPSTW have been proven to be NP-hard, and only relatively small problems of this nature have been solved to optimality due to their huge computational requirements.

2.4 Development of new Techniques

Solving integer programming optimisation problems, that is, finding an optimal solution to such kind of problems, can be a difficult task. To solve a non-convex integer programming problem could be an algorithmically unsolvable task (Britton, 1979; Khachiyan, 1982). The convex non-linear IP problems belong to the class of NP-hard problems (Garey and Johnson, 1979; Arora and Barak, 2009). There are few exact algorithms which can solve these problems in polynomial time, depending on the nature of the problem, input data, length or size of the problem. The linear IP problems are easier to solve than convex non-linear integer programming problems. It should be noted, however, that there are many special cases of problems that belong to the class P (solvable in polynomial time) (e.g. matching, node packing on appropriately restricted classes of graphs, and some optimisation problems), i.e. there exist algorithms with polynomial time computational complexity, which can solve them. The difficulty in solving IP problems arises from the fact that unlike LP problems whose feasible region is a convex set, for IP problems, one must search for a lattice of feasible integer points to find an optimal solution.

Unlike LP problems where due to the convexity of the problem, we can exploit the fact that any local solution is a global optimum. IP problems have many local optima. Finding a global optimum to the problem requires one to prove that a particular solution dominates all the feasible points by arguments other than the calculus-based derivative approaches of convex programming with continuous variables. For this reason, the approximate algorithms for solving integer programming optimisation problems are widely used.

To date, only a few algorithms have been developed that are able to compute a bi-level problem whose lower level problem has discrete variables (DeNegre, 2011; Xu and Wang, 2014). Nevertheless, those algorithms either (i) heavily depend on enumerative BB strategies based on a rather weak relaxation, or (ii) involve complicated operations that are problem-specific and challenging for most researchers and practitioners. Hence, existing methods are of very limited computational capability. As a consequence, there is no commonly accepted approach and little support is available to transform bi-level problems into a decision making tool for real system practice (Xu, 2012).

The intrinsic complexity of interior-point method limits the size of the problems that they can handle. In the recent years, new and cheaper approaches have been developed to attack much larger semi-definite problems. These algorithms, called first-order methods, are very sophisticated extensions of old gradient methods. Their scope of applicability and the impressive acceleration possibilities that they offer are not yet fully understood. Many development strategies can still be investigated for further improvement in order to make them an indispensable tool to tackle huge-size semi-definite problems, or to leverage their versatility in the context of mixed integer convex optimisation.

Over the last two decades, many sophisticated evolutionary algorithms have been introduced for solving constrained optimisation problems (COPs). Elsayed et al. (2011) noted that due to the variability of characteristics in different COPs, no single algorithm performs consistently over a range of problems. The scholars introduced an algorithm framework that uses multiple search operators in each generation. The appropriate mix of the search operators, for any given problem in their method, is determined adaptively. Their algorithm framework was tested by implementing two different algorithms and the performance of the algorithms was judged by solving 60 test instances taken from two constrained optimisation benchmark sets from specialised literature. The first algorithm, which is a multi-operator based genetic algorithm (GA), showed a significant improvement over different versions of GA (each with a single one of these operators). The second algorithm, using differential evolution (DE), also confirmed the benefit of the multi-operator algorithm by providing better and consistent solutions. The overall results demonstrated that both GA and DE based algorithms show competitive, if not better performance than the state-of-the art algorithms.

Balseiro et al. (2011) developed an ant colony system algorithm hybridised with insertion heuristics for the time-dependent vehicle routing problem with time windows (TDVRPTWs). In their research a fleet of vehicles delivered goods to a set of customers and time window constraints of the customers were respected. They also took into account the fact that travel time between two points depended on the time of departure. The latter assumption was particularly important in an urban context where traffic plays a significant role. They highlighted that the shortcoming of ant colony algorithms for capacitated routing problems was that, at the final stages of the algorithm, ants tended to create infeasible solutions with unrouted clients. Hence, they enhanced the algorithm with an aggressive insertion heuristic relying on the minimum delay metric. Computational results confirmed the benefits of involving the insertion heuristics. Moreover, the resulting algorithm turned out to be competitive, matching or improving the best known results in several benchmark problems.

2.5 Improving Existing Techniques

Wang (2012) proposed a hybrid multi-criteria decision making (MCDM) model combining analytic network process (ANP) and decision making trial and evaluation laboratory technique (DEMATEL). Utilising this hybrid method, Wang (2012) applied a framework of decision making to international trade practices in Taiwan. Tsai et al. (2010) took this one step further, although not directly building upon Wang's (2012) research. They combined ANP, DEMATEL, and zero-one goal programming (ZOGP). They applied the new method to apply to a sourcing decision about (i) keeping IT functions in-house or (ii) contracting to a third party provider. Due to certain shortcomings in (AHP), ANP has seen an increase in usage, especially in combination with other MCDM methods.

Artificial bee colony (ABC) algorithm is a relatively new optimisation technique which has been shown to be competitive to other population-based algorithms. However, Gao and Liu (2012) highlighted that ABC still needs to be modified in terms of its solution search equation, which is good at exploration but poor at exploitation. Inspired by differential evolution (DE), Gao and Liu (2012) proposed an improved solution search equation. They stated that ABC searches only around the best solution of the previous iteration to improve the exploitation. In order to make full use and balance the exploration of the solution search equation of ABC and the exploitation of the proposed solution search equation, they introduced a selective probability P to obtain the new search mechanism. In addition, to enhance the global convergence when producing the initial population, both chaotic systems and opposition-based learning methods were employed. Gao and Liu (2012) came up with the modified ABC (MABC), which includes the probabilistic selection scheme and scout bee phase. They conducted experiments on a set of 28 benchmark functions, and their results demonstrated good performance of MABC in solving complex numerical optimisation problems, compared with two ABC-based algorithms.

Vidal et al. (2013) presented an efficient hybrid genetic search with advanced diversity control for a large class of time-constrained vehicle routing problems, introducing several new features to manage the temporal dimension. They proposed new move evaluation techniques, which accounted for penalised infeasible solutions with respect to time-window and duration constraints. The hybrid technique would allow evaluation of moves from any classical neighbourhood based on arc or node exchanges in constant time. Furthermore, geometric and structural problem decompositions were developed to address efficiently large problems. Their proposed algorithm outperforms all current state-of-the-art approaches on classical literature benchmark instances for any combination of periodic, multi-depot, site-dependent, and duration-constrained vehicle routing problem with time windows.

The VRP with time windows is a complex combinatorial problem with many real-world applications in transportation and distribution logistics (Garcia-Najera and Bullinaria, 2011). Its main objective is to find the lowest distance set of routes to deliver goods to customers with service time windows, using a fleet of identical vehicles with restricted capacity. However, there are other objectives, and having a range of solutions representing the trade-offs between objectives is crucial for many applications. Previous researchers used evolutionary methods for solving this problem, and they rarely concentrated on the optimisation of more than one objective, and hardly ever explicitly considered the diversity of solutions (Garcia-Najera and Bullinaria, 2011). They came up with an improved multi-objective evolutionary algorithm, which incorporates methods for measuring the similarity of solutions, to solve the multi-objective problem. The algorithm is applied to a standard benchmark problem set that showed that when the similarity measure is used appropriately, the diversity and quality of solutions are higher than when it is not used. Their algorithm achieves highly competitive results compared with previously published studies and those from a popular evolutionary multi-objective optimiser.

Job shop scheduling problem is a typical NP-hard problem (Qing and Wang, 2012). To solve the job shop scheduling problem more effectively, Qing and Wang (2012) designed some new genetic operators. In order to increase the diversity of the population, a mixed selection operator based on the fitness value and the concentration value was given. Qing and Wang (2012) made full use of the characteristics of the problem itself, by specifically designing a new crossover operator based on the machine and mutation operator based on the critical path. They presented a new algorithm to find the critical path from the schedule. Furthermore, a local search operator was designed, which improved the local search ability of GA greatly. Based on all these, a hybrid genetic algorithm was presented and its convergence was proved. Computer simulations were made on a set of benchmark problems and the results demonstrated the effectiveness of the proposed algorithm.

Roberti and Toth (2012) surveyed the most effective mathematical models and exact algorithms for finding the optimal solution of the asymmetric travelling salesman problem (ATSP). They extended the fundamental integer linear programming (ILP) problem proposed by Dantzig et al. (1954) by first deriving its classical relaxations and combining the techniques with BB and BC algorithms. They defined z_{ij}^h for $(i, j, h = 1, ..., n)$ to be a binary variable that is equal to 1 if arc (i, j) is traversed in position h in the optimal tour, or zero otherwise and r_{ij}^{hk} for $(i = 1, ..., n; j = 1, ..., n; k = 2, ..., n; h = 1, ..., n-1)$ to be a binary variable that is equal to 1 if arc (i, j) is traversed in position h in the first part of the original tour that links vertex 1 to vertex k, or zero otherwise. Their formulations involve $n^4 - 4n^3 + 9n^2 - 7n$ variables and $2n^3 - 9n^2 + 15n - 8$ constraints and the model was formulated as follows:

$$\text{Minimise} \sum_{i \in V} \sum_{j \in V} c_{ij} \sum_{h=1}^{n} z_{ij}^h$$

subject to

$$\begin{aligned}
&\sum_{j \in V} r_{1j}^{1k} = 1 \qquad k \in V\{1\} \\
&\sum_{j \in V\{1\}} r_{ij}^{h+1,k} - \sum_{j \in V\{k\}} r_{ji}^{hk} = 0 \qquad h = 1, ..., n-2, k, i \in V\{1\} : i \neq k \\
&\sum_{j \in V\{k\}} z_{kj}^{h+1} - \sum_{j \in V\{k\}} r_{jk}^{hk} = 0 \qquad h = 1, ..., n-1, k = 2....n \\
&\sum_{j \in V\{k\}} (z_{ij}^{h+1} - r_{ij}^{h+1,k}) - \sum_{j \in V\{1\}} r_{jk}^{hk}(z_{ji}^h - r_{ji}^{hk}) = 0 \qquad h = 2, ..., n-1, k, i \in V\{1\} : i \neq k \\
&z_{ij}^h \in \{0, 1\} \quad i, j, h = 1, ..., n \\
&r_{ij}^{hk} \in \{0, 1\} \quad j, k \in V\{1\}, i \in V\{k\}, h = 1, ..., n-1.
\end{aligned} \tag{2.8}$$

Biological networks look at the physical interactions between proteins, metabolic networks which encode biochemical reactions and transcriptional regulatory networks that describe the regulatory interactions between different genes (Ravasz et al., 2002). If for example, protein A interacts with and influences protein B, and B influences protein C, then A has an indirect influence on C. The shorter the chain of interaction, the stronger the influence. This knowledge may then be used to find treatments for diseases by identifying drug targets (Newman, 2002).

Mangan et al. (2004) noted the advantages derived from combining quantitative and qualitative methodologies in logistic research. They also highlighted that quantitative and qualitative methodologies are generally associated with the two principle research paradigms which are generally labelled as positivism and phenomenology, respectively.

Munapo and Kumar (2013) developed a method to solve a large scale linear programming (LP) model with non-negative coefficients. Their method is an iterative approach in which search points move from one boundary of the convex region to an improved point on the boundary of the LP convex region. Their method is suited for large-scale LP models because of its advantage of turning an $'n'$ variable problem into a two-variable problem. In their method they defined a surrogate constraint which they generate from a point $P_0\ (x_1 = \varphi_1, x_2 = \varphi_2, ..., x_n = \varphi_n)$ for some constants $(\varphi_1, \varphi_2, ..., \varphi_n)$. The point P_0 is a feasible point of the given LP model. The surrogate constraint was defined as:

$$(a_{11} + a_{21} + ... + a_{r1})\, x_1 + (a_{12} + a_{22} + ... + a_{r2})\, x_2 + ... + (a_{1n} + a_{2n} + ... + a_{rn})\, x_n = \beta_1 + \beta_2 + ... + \beta_n \tag{2.9}$$

An improved point P_1 in the feasible region was obtained by moving from the initial point P_0 in the normal direction to the surrogate by $P_1 = P_0 + \mu D + \lambda C^T$, where D is the direction of travel and C^T is the transpose of the objective function coefficients. The method was then compared to the simplex method and it proved to be more effective. The method also does not use slack variables as is the case with the simplex method. Munapo and Kumar (2013) highlighted that this method is most suitable for solving large-scale LP problems. They also noted that it would be desirable to extend the $'n'$ to 2 transformation for the model to cater for a situation where all the elements are not restricted to be non-negative.

2.6 Conclusion

The combination of multiple methods addresses deficiencies that may be seen in certain existing methods. These combined methods, along with the methods in their original forms, can be extremely successful in their applications, but only if their strengths and weaknesses are properly assessed. The most important challenge in some optimisation problems is central processing unit (CPU) time. The original existing optimisation algorithms have been modified in order to increase the convergence speed of most of these optimisation algorithms. The proposed modifications have also been tested on several optimisation problems to evaluate the effect of these modifications on the convergence speed of the new algorithms, and in most cases, the modified algorithms are faster than the original algorithms, and the results of these proposed algorithms have been verified with analytical results.

The performance characteristics of the modified models and algorithms have been compared with those of the original models, and algorithms and results obtained have shown that the modified algorithms are able to converge to optimality and find the optimum point faster compared to the original algorithms. These modifications have been applied not only to new problems, but also to the old problems in order to assess and compare whether or not there are real improvements in these modified methods.

The OR field was developed to find optimal solutions in a holistic manner by taking into consideration all the factors that affect the problem in question. In this modern world these factors are enormous, and expressing them as quantitative models and establishing relationships among them require voluminous calculations, which can be handled only by machines. There is need, therefore, to combine OR with other disciplines like computer science, in order to improve the decision making process.

As shown in this chapter, it is of paramount importance that new techniques be developed. In the literature that have been reviewed in this thesis, it has been shown that new algorithms outperform the old ones. The development of new methods or a combination of existing ones to come up with a better method, is one reason why this research, aimed at developing new techniques, has been undertaken. As the world evolves, new challenges are being encountered on a daily basis. In order to overcome or counterbalance these new challenges, the development of new techniques becomes a necessity. One good example is the data revolution coupled with new form of data, including big data, that has compelled institutions of higher learning and industry to offer e-Research, data science and/or data analytics. Climate change research offers yet another example.

In general, literature reviewed in this chapter has revealed several gaps in theory and application of OR. Several authors have advocated for the improvement of the existing methods and the need to evaluate the new methods adopted in other developed countries, with the view that these methods may also work for the developing countries. This may, however, not be the case given different operating characteristics and climatic conditions.

Part I

Network Models

Chapter 3

Routing Models in Non-directed Networks

> *"I have not failed. I have successfully discovered 10 000 things that won't work".*
>
> *Thomas Edison*

3.1 Introduction

The shortest path problem deals with the determination of a shortest path between two vertices of a graph such that the sum of the weights of the constituent edges is minimised. Shortest path algorithms are applied to automatically find directions between physical locations, such as driving directions on web mapping websites like Map-Quest or Google Maps. For this application fast specialised algorithms are available (Sanders, 2009). If one represents a non-deterministic abstract machine as a graph where vertices describe states and edges describe possible transitions, shortest path algorithms can be used to find an optimal sequence of choices to reach a certain goal state, or to establish lower bounds on the time needed to reach a given state. For example, if vertices represent the states of a puzzle like a Rubik's Cube and each directed edge corresponds to a single move or turn, shortest path algorithms can be used to find a solution that uses the minimum possible number of moves.

Many industrial problems can be formulated and analysed as a shortest path problem or its variants in a network. This has been of interest to researchers and practitioners in many network related disciplines like transportation, operations research, geographic information system (GIS) technology, and emergency services like supply of blood or reporting of ambulance at the accident scene. Many approaches have been developed to find the shortest path from a given node to all other nodes and appropriate computer codes have been developed for implementation of their algorithms (Dijkstra, 1959).

Many variants to this problem have been discussed in literature (Righin and Salani, 2008; Zhu and Wilhelm, 2008). In a networking or telecommunications mindset, the shortest path problem is sometimes called the min-delay path problem, and is usually tied with a widest path problem. For example, the algorithm may seek the shortest (min-delay) widest path, or widest shortest (min-delay) path. A more light-hearted application is the games of "six degrees of separation" that try to find the shortest path in graphs like movie stars appearing in the same film. Applications of shortest path also arise in many different areas where a problem can be viewed as a network problem, and therefore, network methods can be applied to solve such problems.

The majority of communications applications, from cellular telephone conversations to credit card transactions, assume the availability of a reliable network. Reliability is an attribute of any computer-related component (software or hardware), that consistently performs according to its specifications. Reliability has long been evaluated as one of three related attributes that must be considered when making, buying, or using a computer product or component (Ball, 1980). Reliability, availability, and serviceability are considered to be important aspects to design into any system. In theory, a reliable product is totally free of technical errors. In practice, however, vendors frequently express a product's reliability quotient as a percentage. Evolutionary products (those that have evolved through numerous versions over a significant period of time) are usually considered to become increasingly reliable, since it is assumed that bugs have been eliminated in earlier releases.

In a general network, the reliability computation problem is NP-hard (Ball, 1980). The evolution of communication technologies has resulted in continuous increase in capacities and higher concentration of traffic on relatively fewer elements. The failures of these high capacity elements affect the quality of service provided by the network. Reliability is a very complex measure of networks which is difficult to define and/or evaluate. Some of the most critical problems concerning network reliability modelling and analysis are in the determination of possible states of a network with extremely large number of elements subject to failure, and also the determination of the impact of failures on reliability measures in the presence of several applied multi-layer protection techniques.

The problem of evaluating and optimising the reliability of networks deserves attention. The network reliability analysis problem consists of measuring the global probability of the whole network value given failure/operation probabilities for each link and node. To remain competitive, the guarantee of high system reliability at low cost is essential. Computing system reliability is usually not sufficient because it would also provide mechanisms to optimise the reliability taking into account budgetary constraints and parameters which could vary in real-time.

In this chapter, two new variant methods: one for the shortest route problem in a non-directed network, and the other for the determination of maximum reliability in a non-directed network, are formulated. A worked example is given in each case. The advantages and practical usage of the proposed methods are also outlined.

The rest of the chapter is arranged such that Section 3.2 discusses the literature review of the two methods. A minimum weight labelling method for determining the shortest route is discussed in Section 3.3. The methodology, analysis, results and application of this method are also presented in this section. Section 3.4 presents a new and efficient labelling approach for the determination of a maximum reliability route in a non-directed route. Section 3.5 gives the summary of the chapter.

3.2 Literature Review

The problem of shortest route has been studied in directed as well as in non-directed networks (Dijkstra, 1959; Dantzig, 1960; Zhan, 1997; Zhan and Noon, 1998; among others). Many variants of the shortest path problem have been formulated and algorithms developed (Banasal and Kumar, 1977; Kumar et al., 1999; Munapo et al., 2008; among others).

In graph algorithms, the widest path problem is the problem of finding a path between two designated vertices in a weighted graph, and maximising the weight of the minimum-weight edge in the path. The widest path problem is also known as the bottleneck shortest path problem or the maximum capacity path problem. It is possible to adapt most shortest path algorithms to compute widest paths by modifying them to use the bottleneck distance instead of path length (Pollack, 1960). However, in many cases even faster algorithms are possible. For instance, in a graph that represents connections between routers in the Internet, where the weight of an edge represents the bandwidth of a connection between two routers, the widest path problem deals with determination of an end-to-end path that has the maximum possible bandwidth between two internet nodes (Shacham, 1992). The smallest edge weight on this path is known as the capacity or bandwidth of the path. Schulze (2011) described the widest path problem as an important component of the Schulze's method for deciding the winner of a multi-way election. It has also been applied to digital composition (Fernandez et al., 1998), metabolic pathway analysis (Ullah et al., 2009), and the computation of maximum flows (Ahuja et al., 1993).

In an undirected graph, a widest path may be found as the path between the two vertices in the maximum spanning tree of the graph, and a minimax path may be found as the path between the two vertices in the minimum spanning tree (Malpani and Chen, 2002). In any graph, directed or undirected, there is a straightforward algorithm for finding a widest path once the weight of its minimum-weight edge is known: simply delete all smaller edges and search for any path among the remaining edges using breadth first search or depth first search. Based on this test, there also exists a linear time algorithm for finding a widest path in an undirected graph, that does not use the maximum spanning tree. The main idea of the algorithm is to apply the linear-time path-finding algorithm to the median edge weight in the graph and then either to delete all smaller edges or contract all larger edges, depending on whether a path does or does not exist (Kaibel and Peinhardt, 2006).

In graph theory and theoretical computer science, the longest path problem is the problem of finding a simple path of maximum length in a given graph. A path is called simple if it does not have any repeated vertices. The length of a path may either be measured by its number of edges, or (in weighted graphs) by the sum of the weights of its edges. In contrast to the shortest path problem which can be solved in polynomial time in graphs without negative-weight cycles, the longest path problem is NP-hard, meaning that it cannot be solved in polynomial time for arbitrary graphs, unless P = NP (Karger et al., 1997). However, the longest path problem has a linear time solution for directed acyclic graphs, which has important applications in finding the critical path in scheduling problems.

The NP-hardness of the unweighted longest path problem can be shown using a reduction from the Hamiltonian path problem (Schrijver, 2003). A graph G has a Hamiltonian path if and only if its longest path has length $n - 1$, where n is the number of vertices in G. Because the Hamiltonian path problem is NP-complete, this reduction shows that the decision version of the longest path problem is also NP-complete. In this decision problem, the input is a graph G and a number k; the desired output is "yes" if G contains a path of k or more edges, and "no" otherwise (Schrijver, 2003).

The critical path method for scheduling a set of activities involves the construction of a directed acyclic graph in which the vertices represent project milestones and the edges represent activities that must be performed after one milestone and before another. Each edge is weighted by an estimate of the amount of time the corresponding activity will take to complete. In such a graph, the longest path from the first milestone to the last one is the critical path, which describes the total time for completing the project (Sedgewich and Wayne, 2011).

A network formulation is an abstraction of a physical real-life situation which is made for better understanding, analysis and for making informed decisions with respect to that situation as explained in Ahuja (1994) and Chenkassky (1996). One of the many problems that have attracted attention is that of the shortest route joining two nodes of the given network. For shortest route algorithms, see Bellman (1958), Dijkstra (1959) and Dantzig (1960); and for algorithms and applications, see Zhan (1997), and Abraham et al. (2010). A network is comprised of a collection of nodes and links or arcs that joins nodes together. The nodes usually represent points of interest and the links joining these nodes represent relationships between the nodes. In many situations these relationships can be quantified and represented either by a number or by a probabilistic estimate. A network can be directed or non-directed depending on the original physical real-life situation that is being investigated. Many variants of the shortest path problems have been studied in the literature; see for example, Arora and Kumar (1993), Kumar and Bapoo (1999) and Munapo (2004).

In a directed network, each link has an associated direction, which helps in analysing the given situation. The same analysis in a non-directed network becomes more demanding in the absence of directions. A labelling technique for the shortest path problem for a deterministic situation was developed by Munapo et al. (2008) and applied to a critical path analysis, which commonly arises in a project scheduling situation. Later a labelling method for the shortest path in a non-directed network was presented by Kumar et al. (2013). In both cases the network was assumed to have associated link-weights as a deterministic quantity representing distance, time or cost.

Reliability or dependability describes the ability of a system or component to function under stated conditions for a specified period of time (O'Connor

2002). For any system, one of the first tasks of reliability engineering is to adequately specify the reliability and maintainability requirements allocated from the overall availability needs, and more importantly, derived from proper design-failure analysis or preliminary prototype test results. Clear requirements should constrain the designers from designing particular unreliable items. Setting only availability, reliability, testability, or maintainability targets (e.g., maximum failure rates) is not sufficient. Reliability modelling is the process of predicting or understanding the reliability of a component or system prior to its implementation.

The practical definition of reliability is the probability that service will be continuously available over a given period of time (Xin et al., 2013). In their paper, the researchers highlighted that there is a gap between the state of service execution in terms of reliability and what the network reliability data suggest. Their study proposed a network service reliability analysis method based on service in order to bridge the gap. Their study satisfies these needs in the following ways: (1) analysing various factors affecting reliability of the network system; (2) establishing the reliability block diagram of service operating process and calculating the reliability of the equipment by the diagram; and (3) analysing a case utilising the model in (2). Their method did not overestimate the availability of the network system, but enhanced some previous studies by utilising the analysis based on service. With their method, they highlighted that the gap between the service experience of users in terms of reliability and the result reliability of the method will be bridged.

3.3 Shortest Route in a Non-directed Network

3.3.1 Research methodology

This section describes how a new minimum weight method for the determination of the shortest route in a non-directed network is formulated. The minimum weight label method for the determination of the shortest route in a directed network was developed by Munapo et al. (2008). In their method, they changed the weight associated with directed links entering a node by subtracting the minimum weight from all the incoming links and then adding the same minimum weight to the weights of all links departing from that node. This leaves the total weight unaffected along any path from the source node to the destination node.

In a directed network, a link (i, j) joins the nodes i and j. The node i indicates the start of the activity and the node j indicates the end of that activity, and it is further possible to assume, without any loss of generality, that $i < j$, i.e. the nodes of the network are to be numbered according to topological sort of the nodes. Let the source node be denoted by node 1 and the destination node be denoted by node n. All nodes of the network, other than the source node and the sink node, have at least one incoming link and at least one outgoing link. These intermediate nodes will be represented by 2, 3, ..., $(n - 1)$. A label to a node assigns two values: a number m representing its sequential position in the network as was assigned to it by the topological sort, and a weight w representing the shortest distance from the source node to that node. These two numbers associated with a node form its label. The shortest distance from the source node to that node is denoted by w, and it is known. The path that leads to the shortest distance from the source node to that node can also be easily traced out with the help of the label m.

The minimum weight label algorithm discussed in Munapo et al. (2008), consists of the following steps:

Step 1: Label the source node as (1,0), the first number indicating the node where we are coming from and the second number indicating the distance from the source node to that particular node.

Step 2: Set $k = 2$, and go to Step 3.

Step 3: At node k, find the minimum weight,

$w_k = \min[w_{l1,k}; w_{l2,k}; .....w_{lk,k}]$.

The assumption is that at node k, there are l_k number of incoming links. Using the minimum weight w_k, modify the associated weights with all incoming links by subtracting w_k from their existing weights and by adding the same weight w_k to the existing weights of the outgoing links from the node k. Go to Step 4.

Step 4: If $k < (n-1)$, set $k = k+1$ and return to Step 3.

Step 5: Label node n and determine w_n, the minimum weight associated with node n to conclude the shortest distance between the source node and node n. Note that w_n is the minimum weight associated with all the incoming links with the destination node n. The corresponding path can be traced by back-tracking with the help of the element of the label at node n.

3.3.2 Shortest path in a non-directed network

In the case of a non-directed network, the above steps are no longer possible even if we convert the non-directed links into directed links as the very structure of the network undergoes unmanageable changes. For example, a non-directed link (i, j) can be replaced by equivalent directed links $(i \rightarrow j)$, $(j \rightarrow l)$ and $(l \rightarrow i)$. Here node l is an extra node and link (j, l) and link (l, i) are extra links. The weight associated with one of the extra links can be zero, and the weight with the other link can be the original weight associated with the link (i, j). This means that each non-directed link conversion to directed link is achieved by introducing two additional links and one extra node. The network $N(M, L)$, where M is the node set and L is the link set, will transform into a network $N'(M + L, 3L)$. This makes the transformed network unwieldy. In addition to the property that each directed link $(i \rightarrow j)$ will be such that the property $i < j$ will also not be satisfied. Furthermore, the aspect that non-directed connected networks are cyclic will further add complications to the network. In light of all these problems, other properties of non-directed networks have to be exploited.

3.3.3 Mathematical background of non-directed networks

Let $N(M, L)$ be a non-directed network comprising of a node set $\{M\}$ and the link set L. Let also the cardinality of these two sets be denoted by $|M|$ and $|L|$ respectively. Let the origin node be denoted by O and the destination node be denoted by D. Then the number of the other intermediate nodes in the network is given by $(|M| - 2)$. It is assumed that the given situation is represented by a single-source, single-destination non-directed network with non-negative link distances. The following definitions are necessary for the development of the algorithm.

Definition 1: A Node Label

A three-element label is assigned to a node k when the shortest route and the corresponding shortest distance from the origin node O to node k is known. The three-element label at node k is denoted by $k(i, j, d)$, where the number i indicates the order of labelling. Thus, i will have integer values 0, 1, 2, ..., $(M - 1)$. Alternatively, i also indicates the order of the distance of node k from the origin node O. The number j indicates the node on the path leading to the node k. This information helps us to trace the shortest path from the origin node to the specific node k. Finally, the weight d indicates the shortest distance from the origin node O to the node k.

Initially, only the origin node O can be labelled, as the distance from the node O to itself is zero. The path from node O to itself means that travel has not yet commenced. The value of i is zero as it is the nearest node to the origin node O. Thus, initially the label at the origin node O will be $O(0, O, 0)$ and all other nodes will be unlabelled. The third element of the label on a node indicates the total shortest distance between the origin node and that particular node, meaning that the value of the third element d is also zero at the origin node.

Definition 2: The Sets of Nodes and Links

Let L_0 be the set of labelled nodes. Initially the set $L_0 = \{0\}$, i.e. an empty set.

Let L_1 be the set of links that are directly joining labelled nodes to unlabelled nodes. Initially it will be comprised of links (O, k), where a node k is a directly connected node to the origin node O. The links in this set L_1 will be considered for determination of the minimum weight.

L_2 is the set of links which will never participate in the shortest path. Initially this is an empty set. The links in this set will be a collection of those links that exist between any two labelled nodes with positive weight. Travel along the link with positive weight between two labelled nodes will only result in an inferior route. The elements in this set can be one way or two way links.

$L_3(O, k)$ is a set of links that give rise to the shortest path between the nodes O and k. This path gets unfolded with the help of the second element j of the label.

Definition 3: Minimum Weight

Since a label on the source node can be easily created, the set L_0 will not be an empty set. Thus, the set L_1 will also not be an empty set since the origin node will be joined to at least one other node. Hence a minimum weight will always exist, which may not be unique. Once a minimum weight has been obtained and a node k has been labelled, it will result in modified weights of all links joining the node k to other nodes of the network. Modified weight associated with the link (k, j) reflects the total distance from the origin node O to the node j via the node k. The modified weight with links (k, r) will represent the distance from the origin node O to the node r and the distance from the origin node O to the node k will be zero.

The following observations are also necessary for the development of the algorithm.

Observation 1

The origin node O is the start node. The shortest path and the corresponding distance is required from the node O to the destination node D. The first link on the required path will always be one of the links that directly joins the origin node O to some other node j of the given network in the direction $(O \to j)$.

Observation 2

Since all links are non-directed, theoretically one can return to the origin node O from any other node j which is directly connected with node O, but such a path from any node j to the node O will only increase the total distance between node O and node D. Thus, such a revisit to node O from any node j will not be appropriate with regards to the shortest path.

Proof Let the length of the link (j, O) be denoted by $d_{(j,O)} > 0$ which is a positive quantity. Let the shortest path distance between the nodes (O, D) be denoted by $sd_{(O,D)}$. The length of the path from the node j via the node O to the destination will be given by $d_{(j,O)} + sd_{(O,D)}$ which will be longer compared to the distance $sd_{(O,D)}$, which by definition is the shortest distance.

Observation 3

If a node k has been labelled, the shortest path and the shortest distance from the origin node O to the labelled node k is known. From Observation 2, it is clear that returning to a labelled node from any unlabelled node can only increase the distance. Thus, for purposes of the shortest path, all non-directed links from the labelled node k to unlabelled nodes i will be operating as directed links $(k \to i)$, where node k is a labelled node and node i is an unlabelled node.

Observation 4

Since initially the origin node O is a labelled node, the links contained in the set L_1 are the directly connected links originating from the node O to all other nodes. These links may temporarily be treated as directed links from O to those nodes j. Thus, the minimum weight of the links in the set L_1 can be easily determined. Suppose the minimum weight is associated with the link $(O \to k)$. If this minimum weight is subtracted from this minimum weight link, the altered associated weight will be 0 on link $(O \to k)$ and if the same minimum weight is added to all links from the node k in the direction $(k \to j)$, the new weight in the direction $(k \to O)$ will be twice the original weight. Note that one more node has been labelled and this is one iteration of the labelling process.

Observation 5

Since in each iteration of the labelling process, one more unlabelled node is labelled, the process must terminate in at most $(|M| - 1)$ iterations. Alternatively, as soon as the destination node is labelled and becomes a member of the set $\{L_0\}$, the process terminates.

3.3.4 The minimum weight algorithm for the non-directed network

Taking into consideration the above mentioned definitions and observations, the minimum weight algorithm for the non-directed network consists of the following steps:

Step 1:
Label the origin node O as $O(0, O, 0)$, and let the set L_1 will be comprised of all the links that emanate from the node O and linking all the nodes j, which are directly connected to node O. These links will be temporarily treated as directed links from the node O to the node j for determination of the minimum weight. The set L_2, which comprises of the links which will never be part of the shortest path, is an empty set at this point.

Find the minimum weight associated with a link that belongs to the set L_1. Let the corresponding node associated with the minimum weight be denoted by node j and the minimum weight be denoted by $w(O, j)$. Now label the node j that corresponds to the minimum weight as $j(1, O, w_{(O.j)})$. The number 1 in the label on node j indicates that it is the first labelled node after the initial labelled node O. The minimum distance to this node j from the origin node O is $w(O, j)$. The shortest path from the origin node O to the node j is formed by the link (O, j). All other links from the node j will have different weights in

two directions as the minimum weight will have to be added in the direction $(j \to k)$.
Go to Step 2.

Step 2:
Based on the weight value $w(O, j)$, identified in Step 1, modify link weights as follows:

1. The new weight associated with the link (O, j) will be $w(O, j) - w(O, j) = 0$.
2. New weights associated with all directly connected links (j, k) will be given by $w(j, k) + w(O, j)$. Thus, the weight on the link in the direction $(j \to O)$ will be twice the original weight.
3. New weights of all the remaining links not connected with the node j will remain unaltered.
4. Upgrade the sets L_0, L_1 and L_2.

Step 3:
Now the nodes O and j are labelled nodes. Using the Observation 3, one can once again temporarily assume that all undirected links are connected with the nodes O and j as directed links going away from them. Once again find the minimum weight associated with the directed links and label one more node as the new labelled node.

Step 4:
Is the node D labelled? If yes, stop and the shortest path has been determined. If not, go to Step 2.

Note that at each iteration, the total distance of a path joining the origin node O to the destination node D remains unaltered, and all associated link distances are non-negative quantities. Since the sum of the modified distances from the origin node O to the destination node D will be comprised of zero total distance, it will constitute the shortest path.

3.3.5 Analysis and results

The analysis and implementation of the above new technique will be done by considering a non-directed 6-node network shown in Table 3.1. The network diagram of the corresponding network is shown in Figure 3.1. Here the node O is the origin node, the node D is the destination node and the nodes 2, 3, 4 and 5 are the intermediate nodes. Since the non-directed links (O, j) will be used only in the direction $(O \to j)$, it is indicated in Table 3.2, the direction for each entry. Initially the set $L_0 = \{0\}$.

Table 3.1: Link distances

Nodes	O	**2**	**3**	**4**	**5**	**D**
O	–	6	12	∞	∞	∞
2	6	–	5	10	8	∞
3	12	5	–	3	2	∞
4	∞	10	3	–	7	9
5	∞	8	2	7	–	11
D	∞	∞	∞	9	11	–

The link distances are shown in Table 3.2. The set $L_1 = (\{\mathbf{O} \to 2), (O \to 3)\}$, and set $L_2 = \{\}$ is still an empty set. The minimum weight $w(O, j) = \min(6, 12) = 6$ and this corresponds to the link $(O \to 2)$, which will result in a label at the node 2. This label also results in the path set $L_3(O \to 2)$ as the link with minimum distance $w(O, 2) = 6$. The link distances in Table 3.2 will have to be modified by subtracting the minimum weight $w(O, 2) = 6$ from the link $(O \to 2)$ and added to all links joining the node 2 to all other nodes, i.e. links $[(2 \to O), (2 \to 3), (2 \to 4), (2 \to 5)]$.

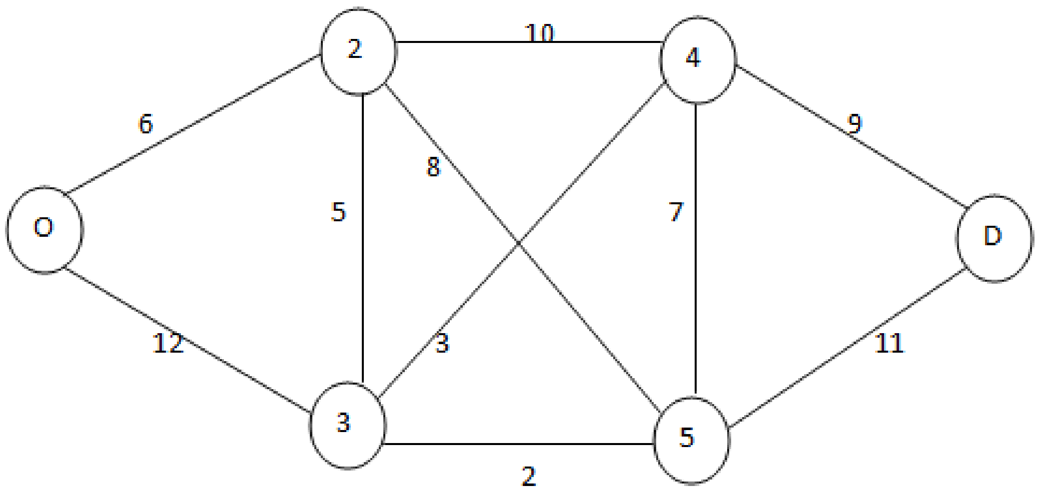

Figure 3.1: Network diagram of numerical example

Table 3.2: Modified link distances

Nodes	O	**2**	**3**	**4**	**5**	**D**
$O_{(0,O,0)}$	$-$	$6_{(O,2)}$	$12_{(O,3)}$	∞	∞	∞
2	$6_{(2,0)}$	$-$	$5_{(2,3)}$	$10_{(2,4)}$	$8_{(2,5)}$	∞
3	$12_{(3,0)}$	$5_{(3,2)}$	$-$	$3_{(3,4)}$	$2_{(3,5)}$	∞
4	∞	$10_{(4,2)}$	$3_{(4,3)}$	$-$	$7_{(4,5)}$	$9_{(4,D)}$
5	∞	$8_{(5,2)}$	$2_{(5,3)}$	$7_{(5,4)}$	$-$	$11_{(5,D)}$
D	∞	∞	∞	$9_{(D,4)}$	$11_{(D,5)}$	$-$

Implementation of the first iteration will result in Table 3.3. The updated sets are: $L_0 = \{O, 2\}$, $L_1 = \{(O \to 3), (2 \to 3), (2 \to 4), (2 \to 5)\}$, and $L_2 = \{(2 \to O)\}$. The minimum weight will be given by $\min(12, 11, 16, 14) = 11$, which leads to the label on the node 3 as shown in Table 3.4. The path set $L_3(O \to 3) = \{O \to 2 \to 3\}$ with minimum distance between the nodes O and 3 is equal to 11.

The distance 11 will be subtracted from the link $(2 \to 3)$ and added to links $(3 \to O)$, $(3 \to 2)$, $(3 \to 4)$ and $(3 \to 5)$, which are shown in Table 3.4.

From Table 3.4, the updated sets are: $L_0 = \{O, 2, 3,\}$; $L_1 = \{(2 \to 4), (2 \to 5), (3 \to 4), (3 \to 5)\}$; and $L_2 = \{(O \to 3), (2 \to O), (3 \to 2)\}$.

The minimum weight will be given by $\min(16, 14, 14, 13) = 13$. This minimum

Table 3.3: Modified link distances after iteration 1

Nodes	O	**2**	**3**	**4**	**5**	**D**
$O_{(0,O,0)}$	–	$0_{(O,2)}$	$12_{(O,3)}$	∞	∞	∞
$2_{(1,O,6)}$	$12_{(O,2,0)}$	–	$11_{(O,2,3)}$	$16_{(O,2,4)}$	$14_{(O,2,5)}$	∞
3	$12_{(3,0)}$	$5_{(3,2)}$	–	$3_{(3,4)}$	$2_{(3,5)}$	∞
4	∞	$10_{(4,2)}$	$3_{(4,3)}$	–	$7_{(4,5)}$	$9_{(4,D)}$
5	∞	$8_{(5,2)}$	$2_{(5,3)}$	$7_{(5,4)}$	–	$11_{(5,D)}$
D	∞	∞	∞	$9_{(D,4)}$	$11_{(D,5)}$	–

Table 3.4: Modified link distances after iteration 2

Nodes	O	**2**	**3**	**4**	**5**	**D**
$O_{(0,O,0)}$	–	$0_{(O,2)}$	$12_{(O,3)}$	∞	∞	∞
$2_{(1,O,6)}$	$12_{(0,2,0)}$	–	$0_{(0,2,3)}$	$16_{(0,2,4)}$	$14_{(0,2,5)}$	∞
$3_{(2,2,11)}$	$23_{(O,2,3,0)}$	$16_{(O,2,3,2)}$	–	$14_{(O,2,3,4)}$	$13_{(O,2,3,5)}$	∞
4	∞	$10_{(4,2)}$	$3_{(4,3)}$	–	$7_{(4,5)}$	$9_{(4,D)}$
5	∞	$8_{(5,2)}$	$2_{(5,3)}$	$7_{(5,4)}$	–	$11_{(5,D)}$
D	∞	∞	∞	$9_{(D,4)}$	$11_{(D,5)}$	–

weight corresponds to the node 5 by using the link $(3 \to 5)$. This means that 13 will have to be subtracted from the link weight $(3 \to 5)$ and added to link weights $(5 \to 2)$, $(5 \to 3)$, $(5 \to 4)$ and $(5 \to D)$. This is shown in Table 3.5.
From Table 3.5, the updated sets are $L_0 = \{O, 2, 3, 5\}$; $L_1 = \{(2 \to 4)$, $(3 \to 4)$, $(5 \to 4)$, $(5 \to D)\}$; and $L_2 = \{(O \to 3)$, $(2 \to O)$, $(3 \to 2)$, $(5 \to 2)$, $(5 \to 3)\}$. The minimum weight will be given by min(16, 14, 20, 24) = 14, which corresponds to node 4. Labelling node 4 and updating the network and the associated weights, we get Table 3.6.

The minimum distance path from the origin node O to the node 4 is given by $\{O \to 2 \to 3 \to 4)\}$ with minimum distance 14. From Table 3.6, the updated sets are $L_0 = \{O, 2, 3, 4, 5\}$, $L_1 = \{4 \to D)$, $(5 \to D)\}$, and $L_2 = \{(O \to 3)$, $(2 \to 4)$, $(4 \to 2)$, $(5 \to 2)\}$. The minimum weight will be given by min(23, 24) = 23, which corresponds to the destination node D. Labelling the node D and

Table 3.5: Modified link distances after iteration 3

Nodes	O	**2**	**3**	**4**	**5**	**D**
$O_{(0,O,0)}$	–	$0_{(O,2)}$	$12_{(O,3)}$	∞	∞	∞
$2_{(1,O,6)}$	$12_{(O,2,0)}$	–	$0_{(0,2,3)}$	$16_{(0,2,4)}$	$14_{(0,2,5)}$	∞
$3_{(2,2,11)}$	$23_{(O,2,3,0)}$	$16_{(O,2,3,2)}$	–	$14_{(O,2,3,4)}$	$13_{(O,2,3,5)}$	∞
4	∞	$10_{(4,2)}$	$3_{(4,3)}$	–	$7_{(4,5)}$	$9_{(4,D)}$
$5_{(3,3,13)}$	∞	$21_{(O,2,3,5,2)}$	$15_{(O,2,3,5,3)}$	$20_{(O,2,3,5,4)}$	–	$24_{(O,2,3,5,D)}$
D	∞	∞	∞	$9_{(D,4)}$	$11_{(D,5)}$	–

Table 3.6: Modified Link distances after iteration 4

Nodes	O	**2**	**3**	**4**	**5**	**D**
$O_{(0,O,0)}$	–	$0_{(O,2)}$	$12_{(O,3)}$	∞	∞	∞
$2_{(1,O,6)}$	$12_{(O,2,0)}$	–	$0_{(0,2,3)}$	$16_{(0,2,4)}$	$14_{(0,2,5)}$	∞
$3_{(2,2,11)}$	$23_{(O,2,3,0)}$	$16_{(O,2,3,2)}$	–	$14_{(O,2,3,4)}$	$13_{(O,2,3,5)}$	∞
$4_{(4,3,14)}$	∞	$24_{(O,2,3,4,2)}$	$17_{(O,2,3,4,3)}$	–	$21_{(O,2,3,4,5)}$	$23_{(O,2,3,4,D)}$
$5_{(3,3,13)}$	∞	$21_{(O,2,3,5,2)}$	$15_{(O,2,3,5,3)}$	$20_{(O,2,3,5,4)}$	–	$24_{(O,2,3,5,D)}$
D	∞	∞	∞	$23_{(O,2,3,4,D)}$	$24_{(O,2,3,5,D)}$	–

updating the network and the associated weights, we get Table 3.7.

Table 3.7: Modified link distances after iteration 5

Nodes	O	**2**	**3**	**4**	**5**	**D**
$O_{(0,O,0)}$	–	$0_{(O,2)}$	$12_{(O,3)}$	∞	∞	∞
$2_{(1,O,6)}$	$12_{(O,2,0)}$	–	$0_{(0,2,3)}$	$16_{(0,2,4)}$	$14_{(0,2,5)}$	∞
$3_{(2,2,11)}$	$23_{(O,2,3,0)}$	$16_{(O,2,3,2)}$	–	$14_{(O,2,3,4)}$	$13_{(O,2,3,5)}$	∞
$4_{(4,3,14)}$	∞	$24_{(O,2,3,4,2)}$	$17_{(O,2,3,4,3)}$	–	$21_{(O,2,3,4,5)}$	$23_{(O,2,3,4,D)}$
$5_{(3,3,13)}$	∞	$21_{(O,2,3,5,2)}$	$15_{(O,2,3,5,3)}$	$20_{(O,2,3,5,4)}$	–	$24_{(O,2,3,5,D)}$
$D_{(5,4,23)}$	∞	∞	∞	$23_{(O,2,3,4,D)}$	$24_{(O,2,3,5,D)}$	–

Table 3.8: Distance and optimal path from origin node O to other nodes

From node O to node j	**Min distance**	**Shortest Path**
$j = O$	0	$O \to O$
$j = 2$	6	$O \to 2$
$j = 3$	11	$O \to 2 \to 3$
$j = 4$	14	$O \to 2 \to 3 \to 4$
$j = 5$	13	$O \to 2 \to 3 \to 5$
$j = D$	23	$O \to 2 \to 3 \to 4 \to D$

Table 3.8 shows the optimal solution using our method. Table 3.9 shows the solution of the same problem using the Dijkstra Algorithm (Dijkstra, 1959). The solution obtained by minimum weight labelling method compares well with the one obtained using Dijkstra's Algorithm. The major advantage of this method is that for an m node network the method finds an optimal solution in at most $m-1$ iterations. Dijkstra's Algorithm does a blind search by looking at all nodes that can be reached from a node that has just been permanently labelled, thereby consuming a lot of time and wasting the necessary resources (Shivan et al. 2013). The algorithm cannot handle negative edges as well, and this leads to acyclic graphs, and most often cannot obtain the optimal shortest path. The distance is calculated as a 1-to-many basis in the Dijkstra's Algorithm when a many-to-many should be better, and this makes the algorithm to have limited memory due the 1-to-many approach.

Table 3.9: Solution of the problem using Dijkstra's Algorithm (Dijkstra, 1959).

From node O to node j	**Min distance**	**Shortest Path**
$j = 2$	6	$O \to 2$
$j = 3$	11	$O \to 2 \to 3$
$j = 4$	14	$O \to 2 \to 3 \to 4$
$j = 5$	13	$O \to 2 \to 3 \to 5$
$j = D$	23	$O \to 2 \to 3 \to 4 \to D$

3.4 Maximum Reliability Route in a Non-directed Reliability Network

3.4.1 Research methodology

In this section we reconsider a problem similar to the shortest path, but assume that link weights are positive quantities between 0 and 1 representing probabilities. For example, the weight associated with a link (i, j) is r_{ij} where $0 \leq r_{ij} \leq 1$ represents reliability of a successful intended operation between the nodes i and j. For a perfect node reliability, R_{ii} = R_i = 1, $\forall i$ (i.e. the reliability of a node is perfect).

We first develop a labelling method for a probabilistic directed network and identify existence of virtual directions in a non-directed network. These directions are used for developing a labelling method for the non-directed network. The proposed approach can easily be used for finding a minimum delay path, widest communication band width etc., which have applications in operations research, robotics and transportation of communication signals. The labelling method developed is used for directed as well as non-directed reliability networks.

3.4.2 Maximum reliability route in a directed acyclic network

The network and associated assumptions

Let $N(M, L)$ be an acyclic directed network N with M nodes and L links or arcs. We will assume that the following conditions hold:

1. Each link (i, j) in the acyclic directed network joining the nodes i and j can be such that $i \leq j$ if the direction of the link (i, j) is from the node i to the node j. In other words, the nodes are numbered in a topological sort of the nodes. The reliability r_{ij} associated with the link (i, j), is such that $0 \leq r_{ij} \leq 1$ and nodes are perfect, i.e. R_{ii} = R_i = 1, $\forall i$.

2. Nodes of the given network after the topological sort are numbered 1 to M, where 1 is the origin node and M is the destination node.

3. All nodes of the network, other than the source and the destination, have at least one incoming and one outgoing link associated with them. The origin node has only the outgoing links and the destination node has only the incoming links.

4. In a progressive way, each node i is assigned a two-tuple label, where the first value in that label is a number m representing a sequential position of a node as per the topological sort from where one has arrived to the node i, and the second number between 0 and 1 is the reliability of successful operation joining the origin node to the current position node i. This number is a probability represented by r_i, where $0 \leq r_i \leq 1$. The path that corresponds to the reliability r_i from the origin node 1 to the node i can easily be traced with the help of the first value of the two-tuple label. A suffix associated with each label is just an indicator of the order of the label.

5. It is assumed that if r_{ij} is the reliability of the link (i, j) and the reliability of any link (j, k) is r_{jk}, then the reliability of the path formed by the two links from i to k passing through the node j is given by r^j_{ik}=$r_{ij}.r_{jk}$.

6. The problem considered in this section is to find the value of $r_{(1,m)} = r_m$, $\forall m$, where $(m = 1, 2, \ldots, M)$.

7. In a directed network, a node j can be labelled only when all nodes directly connected to node j from node i have been labelled with respect to that node j.

Algorithm for finding the maximum reliability route in the directed network

The problem considered in this section is to find in the directed network $N(M, L)$, the maximum reliability $r_{(1,m)}$=r_m, for $m = 1, 2, \ldots, M$ and the corresponding path that will give rise to the reliability r_m.

The following algorithm will be used for a directed network.

Step 1. Carry out the topological sort and call the origin node 1 and the destination node M. Initially the origin node will be the only labelled node and its label will be given by $(1, 1)_1$, indicating that the route to node 1 from the origin node 1 is through itself and its reliability is 1, since the node is perfect. It means that r_1=$R_1 = 1$. The problem is to find the value of $r_{(1,m)}$=r_m, $\forall$ $m = 2, 3, \ldots, M$. Set $k = 1$, and go to Step 2.

Step 2. Find the maximum probability associated with the outgoing links from the labelled node k, and let this reliability be denoted by $r_{(k,j)}$ where $k < j$ and node j qualifies for labelling. The reliability of the route from the origin node to the node j will be given by:

$$r_{(1,j)} = r_{1k}.r_{kj}. \tag{3.1}$$

where the reliability of the route from the origin node to the node j is given by r_j and it is the product of the reliability of the route from the origin node to the node $k(r_k)$ and the reliability of the link joining the node k to the node j.

Step 3. If $k < M - 1$, set $k = k + 1$. Go to Step 2.

Step 4. All nodes have been labelled except the destination node. This means that the maximum reliability and the corresponding route from the origin to the destination can now be determined by using the equation (3.1).

3.4.3 Analysis and results

For the analysis of this method we considered an 8-node directed acyclic network shown in Figure 3.2. Direct link probabilities are also shown.

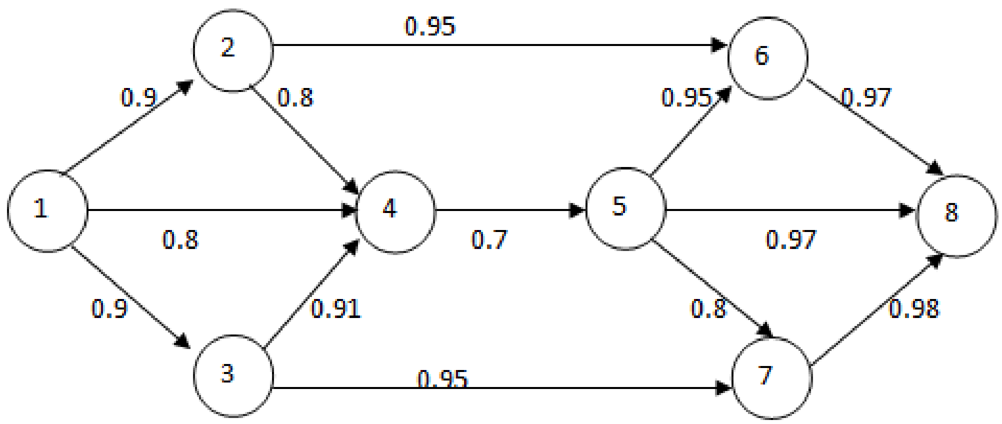

Figure 3.2: Directed network with link probabilities

Link probabilities and the corresponding labels obtained by using equation (3.1) are entered in Table 3.10.

Notes that explain labels shown in Table 3.10

1. We start by labelling node 1, the origin node as $(1, 1)_1$. Although there are three links going out of the origin node 1, only two links will satisfy the labelling requirement as the origin of these links has been labelled. Two links entering the node 4 have their starting point unlabelled. Since node

1 is a labelled node, only links (1,2) and (1,3) will qualify for the next labelling. After arbitrarily selecting the node 2 (because of the tie) the label at this node will be $(1, 0.9)_2$. The next set of links that will qualify for consideration of labelling will be the link (1,3) only. This will result in a label at node 3 as $(1, 0.9)_3$ as shown in Table 3.10.

2. When nodes 1, 2 and 3 have been labelled, the next set of links can be (1,4), (2,4) and (3,4). Calculation details for the next label are:
Max $\{(1, 4) \Rightarrow 1$x0.8 **= 0.8**, $(2, 4) \Rightarrow 0.9$x0.8**= 0.72**, $(3, 4) \Rightarrow 0.9$x0.91 **= 0.82**$\}$ = 0.82, hence node 4 is labelled as $(3, 0.82)_4$.

3. Now nodes 1, 2, 3 and 4 have been labelled. The only link joining the labelled node to an unlabelled node will be (4,5). Hence node 5 will be labelled from node 4 and the label will be $(4, (0.82$x$0.7)$=$0.57)_5$.

4. Note that nodes 1, 2, 3, 4 and 5 have been labelled. The links that will qualify will be given by: (2,6), (5,6), (3,7) and (5,7). Corresponding calculations for the next label are:
Max $\{(2, 6) \Rightarrow 0.9$x$0.95 = 0.86$, $(5, 6) \Rightarrow 0.57$x$0.95 = 0.54$,$(3, 7) \Rightarrow 0.9$x$0.95 = 0.86$, $(5, 7) \Rightarrow 0.57$x$0.8 = 0.46\}$ **= 0.86**, i.e. label either node 6 or node 7. Let us select node 6 arbitrarily. Then the label at the node 6 will be $(2, 0.86)_6$.

5. When nodes 1 to 6 have been labelled, the links for consideration of the next label will be (3,7) and (5,7). Associated calculations have been given in note 4 above, hence the label at node 7 will be $(3, 0.86)_7$.

6. Thus, the final label to the node 8 will be based on links (5,8), (6,8) and (7,8). It will be determined by:
Max $\{(5, 8) \Rightarrow 0.57x0.97 = 0.55$, $(6, 8) \Rightarrow 0.86$x$0.97 = 0.82$, $(7, 8) \Rightarrow 0.86$x$0.98 = 0.84\}$ **= 0.84** from the node 7. Thus, the label at the node 8 will be $(7, 0.84)_8$ and the path will be $1 \rightarrow 3 \rightarrow 7 \rightarrow 8$. The required maximum reliability is 0.84.

Table 3.10: Link reliabilities and labels

i/j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	Links for next label	label	Reliability and Path
1	1	0.9	0.9	0.8	*	*	*	*	$(1,2)$, $(1,3)$	$(1,1)_1$	**1,** $1 \to 1$
2		1	*	0.8	*	0.95	*	*	$(1,3)$	$(1,0.9)_2$	**0.9,** $1 \to 2$
3			1	0.91	*	*	0.95	*	$(1,4)$, $(2,4)$, $(3,4)$	$(1,0.9)_3$	**0.9,** $1 \to 3$
4				1	0.7	*	*	*	$(4,5)$	$(3,0.82)_4$	**0.82,** $1 \to 3 \to 4$
5					1	0.95	0.8	0.97	$(2,6)$, $(5,6)$, $(3,7)$, $(5,7)$	$(4,0.57)_5$	**0.57,** $1 \to 3 \to 4 \to 5$
6						1	*	0.97	$(2,6)$, $(5,6)$, $(3,7)$, $(5,7)$	$(2,0.86)_6$	**0.86,** $1 \to 2 \to 6$
7							1	0.98	$Note5$	$(3,0.82)_7$	**0.82,** $1 \to 3 \to 7$
8								1	$(5,8)$, $(6,8)$, $(7,8)$	$(7,0.84)_8$	**0.84,** $1 \to 3 \to 7 \to 8$

In Table 3.10[1]

[1] A * indicates that there is no direct connection between that pair of nodes; and a blank space indicates opposite direction where information flow is not permitted

3.4.4 Reliability route in a non-directed network

In the case of a non-directed network, the steps of the algorithm described above are no longer valid, even if each non-directed link is replaced by two directed links, as the network will cease to be an acyclic network and will just become unmanageable. Therefore, the only choice is to exploit other features, which may help. Here are a few observations.

Some useful observations

Observation 1: Two numbers p and q representing probabilities between 0 and 1, will satisfy the inequalities $0 \leq p.q \leq \min(p, q)$.
This observation can be easily proved. Let $\min(p, q) = p$. Note that $p - (p.q) = p.(1 - q) \geq 0$ since both quantities on the LHS are positive quantities. A similar result will hold for the quantity q.

Observation 2: Although the given network has non-directed links, a revisit to the origin node from any other node will simply make its reliability less than 1, the current reliability. Hence for the origin node that can be labelled as $(1, 1)_1$, the label will also determine virtual directions from the node 1 to all directly connected nodes k and the direction will be from node 1 to k. For node 1 which has a reliability 1 and is labelled as $(1, 1)_1$, only directions from this node to all other nodes directly connected to it are permissible.

Virtual direction theorem: Once a node has been labelled, it creates virtual directions for all links connected from that node to all unlabelled nodes. The direction will be from the labelled node to an unlabelled node. Thus, the origin node, which initially is the only labelled node, will give rise to virtual directions to all links from the origin, and the direction will be going away from the origin. Thus, all links directly connecting a labelled node to an unlabelled node have virtual directions from the labelled node towards the unlabelled node. Once a labelled node has been identified, one can easily generate a new label from that node. Thus, at each iteration, a new label is created, and the process will terminate in at most M iterations when the given destination node in the non-directed network $N(M, L)$ has been labelled.

The algorithm for the non-directed network

Since there are no directions associated with the links, the destination node can be labelled before all other nodes are labelled. This means that, one has a choice to either stop when the destination node is labelled, or continue labelling until all other nodes are labelled. The steps for the non-directed network will be as follows:

Step 1: Label at the origin node is $(1, 1)_1$, meaning that the path to node 1 is through the node 1 and it is a perfectly reliable node.

Step 2: Find all feasible links that join the labelled node to an unlabelled node by a direct connection. This will be the set of links originating from the origin node. Select the link with a maximum reliability and call this link $(1, b)$, joining the origin node to the node b. This will be the second label, hence we label the node b as $(1, r_{1b})_2$, indicating that the path is the link $1 \to b$ and the reliability associated with this path is r_{1b}. Node b becomes the second labelled node in the network. Let there be a parameter k, and set $k = 2$.

Step 3: If $k \leq M - 1$, set $k = k + 1$ and go to Step 4, else go to Step 5.

Step 4: Find the set of qualifying direct links joining the labelled node to unlabelled nodes. Find the next labelled node and the corresponding maximum reliability path joining the origin node 1 to the selected node. Label the selected node and return to Step 3.

Step 5: Since all nodes have been labelled except one node, find the qualifying links that can label this remaining node. Find the reliability, the label associated with this node and the reliability path.

3.4.5 Analysis and results

Let us reconsider the network in Figure 3.2 as a non-directed network, formed by disregarding the directions as shown in Figure 3.3.

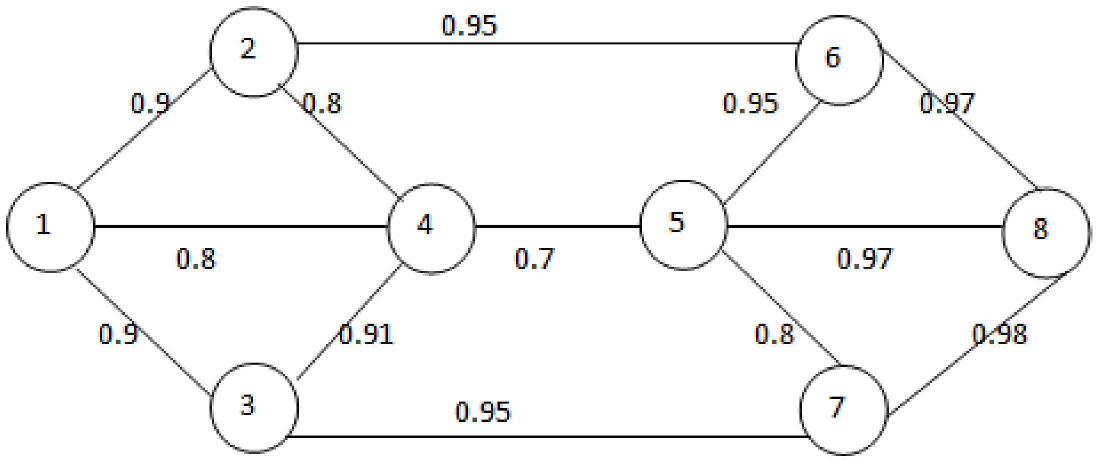

Figure 3.3: Undirected network with link probabilities

Since the network is a non-directed network, the link reliabilities will be the same from i to j as for j to i. This is given in Table 3.11. Calculation details are given as notes below.

Notes on Calculations

1. After an initial label, feasible links joining the labelled node directly to unlabelled nodes are links {(1,2), (1,3), (1,4)} and the associated reliabilities are {0.9, 0.9, 0.8}. The maximum reliability is with respect to node 2 or node 3. We arbitrarily select node 3. This label is entered in Table 3.11.

2. Since the labelled nodes are nodes 1 and 3, feasible links will be (1,2), (1,4), (3,4) and (3,7) for the next label. Corresponding reliabilities are:

Table 3.11: Link reliabilities and labels from Figure 3.3

i/j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	label	Path	Remarks
1	1	0.9	0.9	0.8	$*$	$*$	$*$	$*$	$(1,1)_1$	$1 \to 1$	Initial label 1
2	0.9	1	$*$	0.8	$*$	0.95	$*$	$*$	$(1,0.9)_3$	$1 \to 2$	See note 3
3	0.9	$*$	1	0.91	$*$	$*$	0.95	$*$	$(1,0.9)_2$,	$1 \to 3$	See note 1
4	0.8	0.8	0.91	1	0.7	$*$	$*$	$*$	$(3,0.82)_7$	$1 \to 3 \to 4$	See note 6
5	$*$	$*$	$*$	0.7	1	0.95	0.8	0.97	$(8,0.814)_8$	$1 \to 3 \to 7 \to 8 \to 5$	See note 7
6	$*$	0.95	$*$	$*$	0.95	1	$*$	0.97	$(2,0.86)_4$	$1 \to 2 \to 6$	See note 3
7	$*$	$*$	0.95	$*$	0.8	$*$	1	0.98	$(3,0.86)_5$	$1 \to 3 \to 7$	See note 4
8	$*$	$*$	$*$	$*$	0.97	0.97	0.98	1	$(7,0.84)_6$	$1 \to 3 \to 7 \to 8$	See note 5

Max $\{(1,2) \to 1$x0.9 = **0.9**, $(1,4) \to 1$x0.8= **0.8**, $(3,4) \to 0.9$x0.91 = **0.82**, $(3,7) \to 0.9$x0.95 = **0.86** $\}$ = 0.9. Therefore, node 2 qualifies for the next label and the path is $1 \to 2$.

3. Now nodes 1, 2 and 3 have been labelled. Potential links are $\{$(1,4), (2,4), (2,6), (3,4), (3,7)$\}$ and the reliabilities and the corresponding paths are: Max $\{(1,4) \to 1$x0.8 = **0.8**, $(2,4) \to 0.9$x0.8= **0.72**, $(2,6) \to 0.9$x0.95 = **0.86**, $(3,4) \to 0.9$x0.91 = **0.82**, $(3,7) \to 0.9$x0.95 = **0.86**$\}$ = 0.86 corresponding to node 6 or node 7. We arbitrarily we label node 6 next. The label on node 6 will be $(2,0.86)_4$, which is entered in Table 3.11. The path is $1 \to 2 \to 6$.

4. The labelled nodes are $\{$1, 2, 3 and 6$\}$ and the corresponding feasible links are (1,4), (2,4), (3,4), (3,7), (6,5) and (6,8). Corresponding reliability evaluations are given by:
Max $\{(1,4) \to 1$x0.8 = **0.8**, $(2,4) \to 0.9$x0.8= **0.72**, $(3,4) \to 0.9$x0.91 = **0.82**, $(3,7) \to 0.9$x0.95 = **0.86**, $(6,5) \to 0.86$x0.95 = **0.66**, $(6,8) \to 0.86$x0.97 = **0.83**$\}$ = 0.86, suggesting that node 7 be labelled, generating the path $1 \to 3 \to 7$.

5. The Labelled nodes are $\{$1, 2, 3, 6 and 7$\}$ and node 8 is still unlabelled. Feasible links at this stage are $\{$(1,4), (2,4) (3,4), (6,5), (6,8), (7,5) and

(7,8)}. These possibilities are now evaluated for reliabilities, which are given by:

Max $\{(1,4) \to 1$x0.8 = **0.8**, $(2,4) \to 0.9$x0.8= **0.72**, $(3,4) \to 0.9$x0.91 = **0.82**, $(6,5) \to 0.86$x0.95 = **0.66**, $(6,8) \to 0.86$x0.97 = **0.83**, $(7,5) \to 0.86$x0.8 = **0.69**, $(7,8) \to 0.86$x0.98 = **0.84** $\}$ = 0.84, suggesting that node 8 be labelled from node 7. The path is $1 \to 3 \to 7 \to 8$. The maximum reliability of the path joining the nodes 1 to 8 is 0.84, and the corresponding path is $1 \to 3 \to 7 \to 8$.

Here we have a choice to stop if the objective is to find the maximum reliability path joining the origin node to the destination node; or We can continue labelling as all nodes have not been labelled, and we do not know the maximum reliability route from the origin node to these unlabelled nodes. Currently the set of labelled nodes is {1, 2, 3, 6, 7 and 8}.

6. Links joining the labelled to unlabelled nodes are (1,4), (2,4), (3,4), (6,5), (7,5) and (8,5). Reliability values from node 3 are given by:

   Max $\{(1,4) \to 1$x0.8 = **0.8**, $(2,4) \to 0.9$x0.8 = **0.72**, $(3,4) \to 0.9$x0.91 = **0.82**, $(6,5) \to 0.855$x0.95 = **0.812**, $(7,5) \to 0.86$x0.8 = **0.69**, $(8,5) \to 0.84$x0.97 = **0.8148** $\}$ = 0.82. Node 4 is labelled as $(3, 0.82)_7$.

7. All nodes are labelled except node 5. Links to be considered are (4,5), (6,5) (7,5) and (8,5). The associated calculations are:

   Max $\{(4,5) \to 0.82$x0.7 = **0.574**, $(6,5) \to 0.855$x0.95 = **0.812**, $(7,5) \to 0.86$x0.8 = **0.69**, $(8,5) \to 0.84$x0.97 = **0.8148** $\}$ = 0.814 from node 8. Thus the label on node 5 is $(8, 0.814)_8$.

Comparison between directed and non-directed networks

For a given directed network, reconsidered as a non-directed network, the maximum reliability between any pair of nodes will always be greater than or equal to the reliability between the same pair of nodes under the case when the net-

Table 3.12: Reliabilities comparison, paths and the labelling order

From Origin to node	**Reliability and the path for Directed network**	**Reliability and the path for Non-directed network**
1	1, $1 \to 1$, initial	1, $1 \to 1$ initial
2	0.9, $1 \to 2$	0.9, $1 \to 2$
3	0.9, $1 \to 3$	0.9, $1 \to 3$
4	0.82, $1 \to 3 \to 4$	0.82, $1 \to 3 \to 4$
5	0.57, $1 \to 3 \to 4 \to 5$	0.814, $1 \to 3 \to 7 \to 8 \to 5$
6	0.86, $1 \to 2 \to 6$	0.86, $1 \to 2 \to 6$
7	0.86, $1 \to 3 \to 7$	0.86, $1 \to 3 \to 7$
8	0.84, $1 \to 3 \to 7 \to 8$	0.84, $1 \to 3 \to 7 \to 8$

work is assumed to be directed. This is due to loss of independence in the movement from one node to the other. The fact that links have been reduced in a directed network means that the choice for links that gives higher reliabilities has also been reduced resulting in a greater or equal reliability in a non-directed. For comparison, see Table 3.12.

3.5 Summary of the Chapter

A new minimum weight labelling method for determination of the shortest route in a non-directed network was formulated. The method was used to solve a 6-node network problem. This method can be used on both the directed and non-directed networks. This new method has its motivation from the method developed by Munapo et al. (2008), which solves problems on directed networks. The major contribution of this method for determining the shortest route in a non-directed network is that, for an m-node network, the algorithm developed finds an optimal solution in at most $m-1$ iterations. The solution of the method was also found to be similar to the one obtained using the Dijkstra's Algorithm (1959).

For large networks (with more than 50 nodes and 100 links) this method will be more preferred than the traditional methods because of its ability to compare the weights of all the adjacent nodes to the one recently permanently labelled. The algorithm enables the algorithm to search backwards for any possible shortest paths, guaranteeing that all possible paths have been searched. Just like the Dijkstra's Algorithm all nodes of the network will be labelled so that it can be easy to identify the shortest path from the start node to any other node on the network. This concept of the algorithm has several applications in real life, with examples in telecommunication, transportation, logistics and distribution management.

As is the case with any new methodology, there are some areas for possible extension and improvement. For example, the determination of set L_2 (which is a set of links that will never participate in the shortest path), can lead to the determination of the second best, third best etc., shortest paths. The second best shortest path has several applications in real life, the common application being in disaster management. If the best shortest path method cannot be used, then the second best can be implemented. Xu et al. (2012), went further to evaluate the K shortest paths in a schedule-based network, an algorithm that has several applications in computer science.

In a non-directed reliability network, virtual directions for a specified purpose do exist, and they have been used to find the maximum reliability from the origin node to all other nodes. In the case of a non-directed network, the order of label indicates that the path reliabilities are in non-increasing order. Since virtual directions are dependent on labels, this approach can be used for the determination of all reliability paths from a given node to all other nodes in that network. In any real-life application for a given situation, it may be desirable to consider all variations that might occur in the input data, before accepting the outcome of an analysis. The occurrence of these variations in the input data was referred to as a *protean system* by Kumar (1995), Kumar and Arora (1995) and Kumar et al. (1999), which can easily be accommodated.

Information recycling is useful for protean networks. The protean system deals with changes in the model, and recycling deals with extracting information that may be available from the system before occurrence of a change. In waste management, recycling reduces the bulk of solid waste and provides cheap resource to industry. Similarly, information recycling is intended to minimise unnecessary computations when that information can be extracted by earlier computations. These situations can arise also in reliability networks. Information recycling concept has been applied to mathematical programming models (Kumar, 1995).

Traditionally, directed networks are relatively easy to analyse compared to non-directed networks, as directions have inbuilt additional information that has been exploited from time to time in various forms (Bellman, 1958; Pollack and Weibenson, 1960; Beckwith, 1961). A large number of applications of directed graphs have been presented and analysed in Hastings (1973). All these cases discussed by Hastings (1973) form a directed network, and were analysed using the dynamic programming technique for the directed networks. The same dynamic programming analysis becomes very demanding for a non-directed network. Munapo et al. (2008) developed a labelling technique for the directed network by link-weight modification and solved the shortest route problem in a directed network. Their method is simple and easy to implement, and it found strong applications in the critical path method (CPM) analysis. However, when a network is non-directed, all those properties used by Hastings (1973) or Munapo et al. (2008) are no longer applicable. Loss of directions results in an increase of computational effort as illustrated by Beckwith (1961).

In this chapter, we have attempted to use other properties of the given network and identified virtual directions based on those other properties. We used those virtual directions to establish a labelling method when link weights are deterministic values representing cost, distance or time. Using these virtual directions, a labelling technique was developed and illustrated. Similarly, in a probability network where link weights are represented by probabilities, the network has been analysed for directed and non-directed networks for finding the maximum reliability and the route in these reliability networks.

Since the proposed method concludes in $n-1$ iterations where n represents the number of nodes in the given network, the computational requirement remains under control, even for the non-directed network. The concept of identifying virtual directions is a challenge which is worth further investigations for other variants of routing problems, and this will be the subject of subsequent investigations.

Chapter 4

Minimum Spanning Tree based Models for Solving Some NP-hard Problems

To do successful research, you don't need to know everything, you just need to know of one thing that isn't known.
Arthur Schawlow

4.1 Introduction

The minimum spanning tree (MST) is one of the most well known problems in combinatorial optimisation (Graham, 1985). According to Graham (1985), the MST is the shortest distance that is used to connect all the nodes in a network. Zhaocai et al. (2013) defined the MST as a problem of finding the minimum edge connected subsets containing all the nodes of a given undirected graph. Zhaocai et al. (2013) came up with a new and fast algorithm for solving the

MST problem based on the computation of DNA molecules. Concurring with both Graham (1985) and Zhaocai et al. (2013), Peppino et al. (2013) also defined the MST as the problem of finding a spanning tree with minimum total cost such that each non-leaf node in the tree has a degree of at least d, $(d > 2)$. While the MST previously used to perform more comprehensive studies of asset returns correlations, it can also be used to deduce the underlying ownership structure with reasonable accuracy (Rosovsky et al., 2014). The Euclidean MST-based evolutionary algorithm to solve multi-object optimisation problems was proposed by Li et al. (2014).

The purpose of this chapter is to develop two new techniques that make use of the MST of a network graph. The two techniques are developed in such a way that the node index of each node n_i satisfies the condition that $1 \leq n_i \leq 2$, for all i. It is anticipated that such a spanning tree may have several applications, including determination of the travelling salesman tour (TST).

This chapter also presents a MST approach to determine a route through $'k'$ specified nodes. The path through $'k'$ specified nodes is a difficult problem for which no good solution procedure is known. The proposed method determines the route, which may either be an optimal path or a near optimal path. The complexity of this problem depends on the number of specified nodes.

The rest of the chapter is arranged as follows: Section 4.2 reviews the literature of the MST model and the route through $'k'$ specified nodes. Section 4.3 presents the MST with node index which is less than or equal to 2. The theorems that led to the reduction in the node index are also discussed in this section. Section 4.4 presents the route through $'k'$ specified nodes, its applications and the complexity of the problem as the number of specified nodes increases. Section 4.5 gives a summary of the chapter.

4.2 Literature Review

Jayawant and Glavin (2009) highlighted that the MST problem originated in the 1920s when Boruvka in 1926, identified and solved the problem during the electrification of Moravia. In the 1950s, many authors contributed to the MST problem, among them were Kruskal (1956) and Prim (1957), whose algorithms are very widely used today. The algorithm now known as Prim's algorithm was in fact discovered earlier by Jarnik in 1930. Jayawant and Glavin (2009) presented a variant of Boruvka's algorithm and compared it to the algorithms given by Boruvka, Prim and Kruskal which have been central to the history of the problem.

Anupam (2015) defined the MST problem as a classic (and important) problem, which has been tackled many times. The author gave a brief history of the problem and stated that Boruvka's algorithm, formulated in 1926, was the first MST algorithm. Jarnik gave his algorithm in 1930 and Kruskal gave his in 1956. Prim rediscovered Jarnik's algorithm in 1957 and Dijkstra gave his algorithm in 1959. According to Anupam (2015), all these algorithms can be easily implemented in $O(mlogn)$ time where n is the number of nodes and m is the number of edges. Yao's (1975) algorithm was formulated in 1975 and it achieved a run time of $O(loglogn)$. Anupam further highlighted that Karger, Klein and Tarjan got an algorithm with a run time of $O(m)$ time but it was a randomised algorithm, so the search for a deterministic linear-time algorithm continued. Dynamic programming formulation for the MST was given by Garg and Kumar (1968).

According to Graham and Hell (1985), the MST is one of the most typical problems of combinatorial optimisation. The problem has generated important ideas of modern combinatorics and has played a central role in the design of computer algorithms. The MST has several applications that include

designing of computer communication networks, power and leased lined telephone networks, wiring connections, links in transportation network, piping in a flow network and several others (Graham and Hell, 1985). The MST offers methods of solutions to other problems to which it applies less directly, such as network reliability, surface homogeneity tests, picture processing, automatic speech recognition clustering and classification problems (Kumar et al., 2016)
.

Given an undirected network with positive edge costs and a positive integer $d > 2$, the minimum-degree constrained minimum spanning tree problem is the problem of finding a spanning tree with minimum total cost such that each non-leaf node in the tree has a degree of at least d (Akgun and Tansel, 2010). According to the authors, this problem is new to the literature while the related problem with upper bound constraints on degrees is well studied. Mixed integer programs proposed for either type of problem are composed, in general, of a tree-defining part and a degree-enforcing part. In their formulation of the minimum-degree constrained minimum spanning tree problem, Akgun and Tansel (2010) stated that the tree-defining part is based on the Miller-Tucker-Zemlin constraints while the only earlier paper available in the literature on this problem used single and multi-commodity flow-based formulations that are well studied for the case of upper degree constraints. They proposed a new set of constraints for the degree-enforcing part that lead to significantly better solution than earlier approaches when used in conjunction with Miller-Tucker-Zemlin constraints.

Chazelle (2000) agreed with other authors and pointed out that the history of the MST problem goes as far back as Boruvka's work in 1926, and the author also pointed out that the MST is perhaps the oldest open problem in computer science. Chazelle (2000) presented a deterministic algorithm for computing a

MST of a connected graph. The algorithm had a running time of $O(m\alpha(m,n))$, where α is the classical functional inverse of Ackermann's function, n is the number of vertices (nodes) and m is the number of edges. The algorithm used pointers and not arrays, and it made no numeric assumption on the edge cost.

Ishii and Matsutomi (1995) considered a P model version of stochastic spanning tree problems with random edge costs. Parameters of underling probability distribution of edge costs were unknown, and estimated by a confidence region from statistical data. The problem was first transformed into a deterministic equivalent problem with a minimax type objective function and a confidence region of means and variances, since they assumed normal distributions with respect to random edge costs. Their model reflects the situation that the maximum possible damage due to an unknown parameter should be minimised. They also showed that the problem can be reduced to the deterministic equivalent problem of another stochastic spanning tree problem, which they had previously investigated. Thus, they found an optimal spanning tree of the original problem very efficiently by this reduction technique.

According to Gomes et al. (2015), there are very few works that address the problem of calculating the shortest path from a source node to a target node that visits a specified set of nodes (it is assumed that the source and target nodes are different). The first known work is from Saksena and Kumar (1966) who developed an exact algorithm using the principle of optimality, for calculating the shortest path (possibly with cycles) that visits a specified set of nodes. They named their method $SK66$. de Andrade (2013) noted that if the set of specified nodes to be visited is made of all nodes in the graph, excluding the source and target nodes, this corresponds to finding an Hamiltonian path of minimum cost, which is NP-hard.

Ibaraki (1973) considered separately the problem of calculating the shortest loop-less path that visits a specified set of nodes and the shortest path (possibly with cycles) that visits a specified set of nodes. The scholar proposed two approaches for the calculation of the shortest loop-less path that visits a specified set of nodes, one based on dynamic programming and the other based on the branch and bound principle. Computational results indicated that the algorithm based on branch and bound principle was more efficient than the algorithm based on dynamic programming.

Vardhan et al. (2009) presented an algorithm to find a simple path in the given network with multiple must-include nodes. They highlighted that the problem of finding a simple path with only one must-include node can be solved in polynomial time using lower bound max-flow approach. However, including multiple nodes in the path has been shown to be NP-complete. This problem may arise in network areas such as forcing the route to go through particular nodes, which have wavelength converter (optical), monitoring provision (telecom), gateway functions, or are base stations.

In their research, Vardhan et al. (2009) formulated a heuristic algorithm to find a simple path between a pair of terminals which has a constraint to pass through a certain set of other nodes. The algorithm was divided into two main steps: (1) considering a pair of nodes in sequence from source to destination as a segment and then computing candidate paths between each segment, and (2) combining paths, one from each segment in order to form a simple path from the source to the destination. The max-flow approach was used to find candidate paths which provided maximum number of edge disjoint paths for individual segments. The second step of their algorithm used backtracking algorithm for combining paths. The time complexity of the first step of their algorithm is $O(k\,|V|\,|E|^2)$, where k is the number of must-include nodes, V is the number of vertices and E is the number of edges. The time complexity of step (2) depends upon the total number of candidate paths which are not touching any one of the candidates of other segments. So, the worst-case time complexity of step (2) was $O(\lambda^k)$, where λ is the maximum nodal degree of the network. However, they showed that step (2) has minimal effect on the algorithm and does not grow exponentially with k in this application. Their experimental results showed that the algorithm is successful in computing the near optimal path in reasonable time.

The problem of calculating the shortest path that visits a given set of nodes is at least as difficult as the travelling salesman problem, and it has not received much attention (Gomes et al., 2015). The authors formulated a heuristic whose results were compared to a previously efficient integer linear programming (ILP) formulation that solved this problem. The ILP formulation included a constraint that forced the model to obtain a path that will be protected by a node-disjoint path. Computational experiments, however, showed that this approach, in large networks, may fail to obtain a solution in a reasonable amount of time. Therefore, Gomes et al. (2015) proposed three versions of their heuristic, for which extensive computational results show that they are able to find a solution in most of the cases that they have considered. The calculated solutions using their method gave an acceptable relative error regarding the cost of the optimal active path. Furthermore, the CPU time required by their heuristics was significantly smaller than the one that is required by the ILP solver that they compared with.

4.3 Minimum Spanning Tree with index ≤ 2

The minimum spanning tree (MST) of a given graph can be obtained iteratively by any greedy approach, which is linear in time and converges in $n-1$ iterations, where n is the number of nodes. In a connected network, a spanning tree is a group of $n-1$ arcs that connects all the nodes of the network and contains no loops. A spanning tree connecting all nodes of a network becomes a MST when the sum of the selected arcs is smallest. Generally, the method of finding a MST, arbitrarily starts from any node and connects that node to the nearest node, forming a spanning tree of the two nodes. In the next iteration, one more node is selected which is nearest to one of them and also not forming a loop with already selected nodes. Ties are resolved arbitrarily. After $n-1$ such iterations, all nodes and the selected $n-1$ links form a MST of the given graph.

The index of a node in the MST is given by the number of arcs joining this node to other nodes. The index value for each node will be at least 1 and at most $n-1$ the maximum being realised when all nodes are connected to the same node.

4.3.1 The problem statement and the mathematical support

For a given graph $G(n, L)$, where n is the number of nodes and L is the number of links or arcs, the MST obtained by any known greedy approach will have node indexes n_i, where $1 \leq n_i \leq (n-1)$ for node i, and hence such a MST will have to be modified to satisfy the condition $1 \leq n_i \leq 2$, for all i. In this thesis, it is assumed that the $'n'$-node network is a connected graph where each node has at least two arcs emanating from it.

Definitions

Basic arc: An arc connecting two nodes i and j is said to be **basic** if $x_{ij} = 1$, that is if it belongs to the MST solution. If $x_{ij} \neq 1$, then the arc is said to be **non-basic**.

Index of a node: The index of a node in an MST graph is given by the number of basic arcs emanating from that node. Since the total number of selected arcs in a MST will be $n - 1$, the total node index value of these selected arcs in a MST will be $2(n - 1)$. In an extreme case, the total node index number can be distributed such that $n_i = 1$, for $n - 1$ nodes; and for another node, the node index $n_i = n - 1$, and this happens when all the nodes are connected to one node. When the node index has to satisfy the condition $n_i \leq 2$, for all i, however, the selection of arcs forming the minimum connected graph will have to be readjusted. Since the MST will be comprised of all nodes and $n - 1$ selected arcs, the number of nodes with index 2 will be at most $n - 2$, and the remaining two nodes will have the node index of 1 to get the total node index value of $2(n - 1)$. Therefore, the selected arcs joining nodes with node index greater than 2 will have to be replaced by other arcs to balance out the index requirement on each node. The network modification theorem given later in this chapter attains even distribution requirement of the node index values.

High and low index nodes: Since the number of basic arcs emanating from a node gives its index value, a node is called a high index node if its number of basic arcs is greater than 2, and a low index node if the number of basic arcs is 1. In this thesis, we require a MST, where the index n_i at node i satisfies the condition $1 \leq n_i \leq 2$, for all i.

Neighbouring arcs: These are arcs that emanate from neighbouring nodes. A node i is said to be a neighbour to node j if the two nodes i and j are connected by a single arc. In a completely connected graph, all nodes are neighbouring nodes since all pairs of nodes are directly connected by single arcs.

A string in a MST: A string in a MST is a collection of arcs where the degree of all intermediate nodes is 2 and the end nodes have a degree 1 with respect to that string. In other words, more than one string may start from the same node and more than one string may have a part of the string common to them. Thus, the MST of a given graph may have several strings.

Balancing node index by arc weight modification

Theorem 4.1

Adding or subtracting the same constant μ to all the arc-distances emanating from the same node does not change the relative merit of any given tour with respect to other tours. Note that there are $(n-1)!$ tours in an $'n'$-node completely connected network.

Proof

In a completely connected graph, each node has $n-1$ arcs emanating from it. Suppose there are $n-1$ arcs emanating from node i as shown in Figure 4.1.

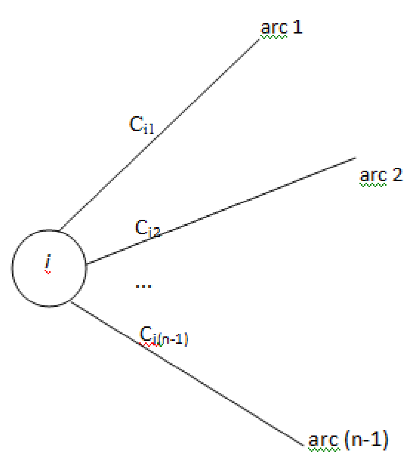

Figure 4.1: Arcs emanating from a node i

Adding a constant μ to all the arcs in Figure 4.1 generates the modified distances shown in Figure 4.2.

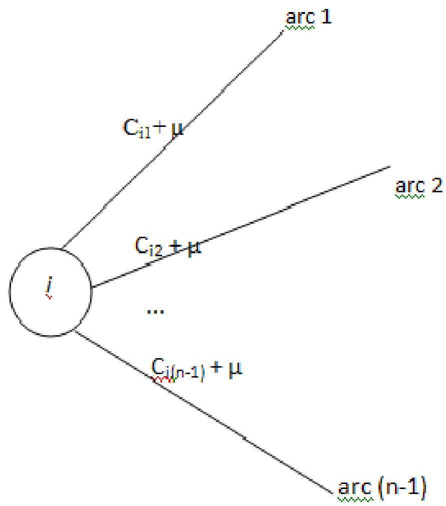

Figure 4.2: After adding a constant μ to every arc

When an arc length is changed by adding a constant quantity μ to each arc emanating from that node, it will have consequences on both sides of the arc, i.e. arc-weight distribution of two nodes will be affected. Therefore, two separate considerations are required at both ends of the arc. Consider that adding a constant modifies the arcs emanating from the node i. The motivation for this modification is to create an alternative for the MST. When the node i is a candidate for modification as the index n_i is greater than 2; let the arc (i, p), from node i to node p be currently a MST member that is causing the imbalance (i.e. to have node index at $n_i \geq 2$. If length of the arc (i, p) is increased to be equal to an arc (p, l), which at present is not a member of the MST since the arc-weight (i, p) is smaller than the arc-weight (p, l). After adding a constant μ, the arc-weight (i, p) is made equal to arc weight (p, l); the arc (i, p) can now be replaced by the arc (p, l) in the MST. Therefore, altering an arc-weight of (i, p) brings the corresponding index value change at node i as well as at node p. The index at node i goes down by 1 and the index at node p goes up by 1.

Consideration at node i

Let the optimal MST be of length $(L[\tau_0])$. This length is a sum of $(n-1)$ selected arc-weights in the given n-node network. The MST under the index restriction can have at most two arcs emanating from node $'i'$, one of them will give entry to that node and the other will provide exit from that node. Since $(L[\tau_0])$ is minimum, the same MST will remain minimum in the modified network as shown by equation (4.1).

$$L[\tau_0] + 2\mu \leq \min L^{'}[\tau_k] + 2\mu \tag{4.1}$$

where $L^{'}[\tau_k]$ represents the set of lengths of other MSTs excluding the minimum length. Note that equation (4.1) holds since equation (4.2) is true by definition.

$$L[\tau_0] \leq \; L^{'}[\tau_k] \tag{4.2}$$

The constant μ is a positive quantity that was used to create an alternative without changing the relative merit of a given MST.

Consideration at the other end of the arc (i, p)

Since in a connected graph all arcs emanating from a node $'i'$ are changed, we have also changed arc weights from other nodes $'p'$ to this node $'i'$. Thus, relative merits of the arcs from the node $'p'$ are changing as well. However, the affected arcs have no place in the MST as these other arcs did not belong to the MST, but were only creating alternative routes. Thus, arc weights can be modified in the above manner, resulting in an equivalent network with alternative MSTs.

Balancing node index by arc weight modification

Theorem 4.2

For any given MST solution, the number of basic arcs emanating from node i can be altered by adding a constant μ to all the arcs emanating from that node.

Proof

Let any two neighbouring nodes of node i be node j and node k as shown in Figure 4.3.

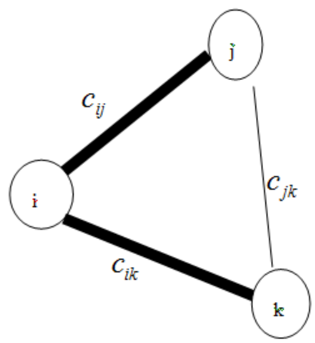

Figure 4.3: Neighbouring nodes

In Figure 4.3 the arcs $(i;j)$ and $(i;k)$ are basic. A positive constant μ is added to each arc emanating from node i as shown in Figure 4.4.

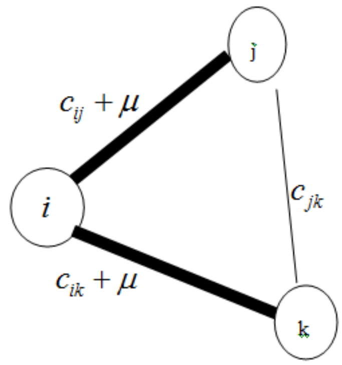

Figure 4.4: Adding a positive constant μ

If μ is a positive quantity such that $C_{ij}+\mu \geq C_{jk}$ or $C_{ik}+\mu \geq C_{jk}$, then the new MST solution becomes as shown in Figure 4.5 or in Figure 4.6.

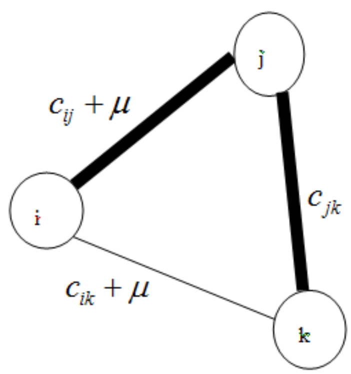

Figure 4.5: New MST when $C_{ik}+\mu \geq C_{jk}$

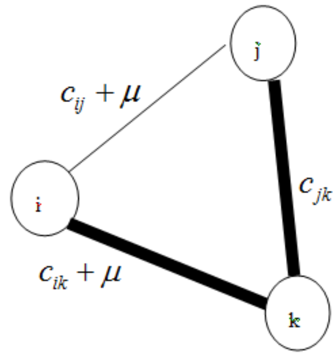

Figure 4.6: New MST when $C_{ij} + \mu \geq C_{jk}$

These diagrams show ways of reducing the number of basic arcs emanating from a given node i. Only three nodes and three arcs are used to illustrate the theorem. A method for any number of nodes greater than three is similar to the above case. The question is how to find the quantity μ.

The purpose of including an additional quantity to the existing arc weights is to create alternative arcs that can qualify to become basic as a member of the new MST. Thus, one can alter the number of basic arcs from a given node. The value of μ is obtained by looking at the minimum difference between the basic arc-weight and the incoming non-basic arc-weight so as to create an alternative arc to form a new MST.

4.3.2 MST path

The MST under the node index condition $1 \leq n_i \leq 2$, for all i, is a path. When $n-2$ nodes have index 2, the remaining two nodes will have index 1, then they form a path. This path may have several applications like finding the TST of a network. Such a path will also be useful for the situation when a single truck is being used to deliver seeds to various centres and deliveries must be done before the season starts. It is assumed that seeds are being sourced from a supplier who is willing to deliver to any desired starting point. In this case the MST path will give a far better solution when seeds are delivered from the node with index value 1 to all other nodes. Note that Theorem 4.1 and Theorem 4.2 are applicable to any high index node. Their repeated applications can modify the node index value to a desired value, which in this case is 2.

An example of a MST path is given in Figure 4.7.

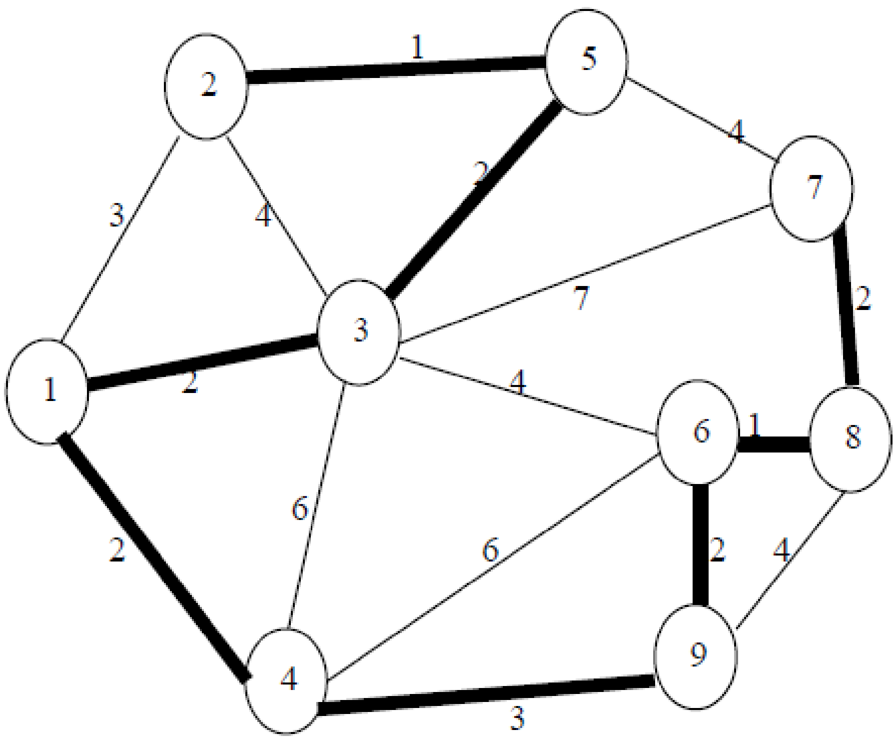

Figure 4.7: An MST path between nodes 2 and 7

The MST solution in Figure 4.7 is a path. If the supplier is willing to deliver to any of the nine centres (represented by nodes), then it makes sense to have the deliveries moving from node 2 passing through all the other centres until we reach node 7. Theorems 4.1 and 4.2 can be applied to obtain a MST path from any given source node i to any destination node j. If the associated weights in Figure 4.7 are in hours then the shortest time in which deliveries can be made to all nodes is given in equation (4.3).

$$L[\tau_0] = 2 + 1 + 2 + 3 + 2 + 2 + 2 + 1 = 15hrs \tag{4.3}$$

Note that there is no other delivery time less than 15hrs.

The Algorithm

The algorithm to find the MST with node index restriction can be described as follows:

Step 1

Find a MST of the given graph by any known method. If in the process of arc selection, a tie is experienced, always select the arc that does not increase the degree of a node beyond 2. Go to Step 3.

Step 2

Find the MST of the modified network and go to Step 3. Once again ties are resolved as in Step 1. As arc lengths are modified, more and more ties will be observed. Always select an arc that does not increase the index of a node beyond 2, if feasible.

Step 3

Check if the MST obtained satisfies index conditions. Do all nodes have node index less than or equal to 2? If the answer is “no”, go to Step 4, else go to Step 5.

Step 4

Select a node $'i'$ with node index 3 or more. With the help of the neighbouring arcs, find the minimum value μ that can be used to reduce the index at the high index node $'i'$. Reduction in index is achieved by adding an appropriate minimum quantity μ to all arcs emanating from the selected node $'i'$. Doing this will result in the creation of an alternative basic arc that will reduce index of the node $'i'$, and increase the node index of a low index node. Go to Step 2.

Step 5

The optimal MST is obtained when all index conditions are satisfied.

4.3.3 Analysis and results

We consider the 6-node completely connected network used by Cowen (2011) for a TSP. It is given in Table 4.1. The objective is to find an index restricted MST.

Table 4.1: Arc-weights considered by Cowen (2011)

From\To	1	**2**	**3**	**4**	**5**	**6**
1	–	11	9	9*	15	16
2	11*	–	14	10	10	15
3	9	14	–	6	13	11*
4	9	10	6*	–	9	10
5	15	10*	13	9	–	8
6	16	15	11	10	8*	–

The corresponding network diagram is shown in Figure 4.8.

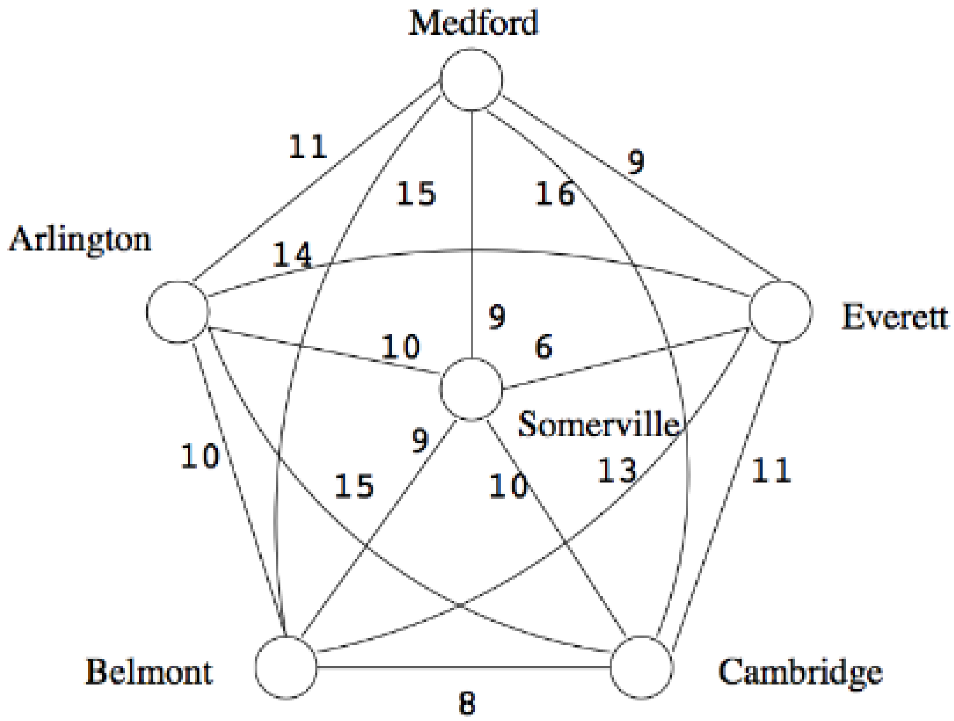

Figure 4.8: Arc-weights considered by Cowen (2011)

In Figure 4.8 Medford is node 1, Arlington node 2, Everett node 3, Somerville node 4, Belmont node 5 and Cambridge node 6. The optimal tour obtained by Cowen (2011) is comprised of the following arcs: $\{(1,4),(4,3),(3,6),(6,5),(5,2),(2,1)\}$. These arcs have been indicated by a star mark in Table 4.1. The optimal tour length as obtained by Cowen (2011), is given by {9+6+11+8+10+11 = 55}.

Since, for an MST, we can start from any node, we commence arbitrarily from node 6 and select the first arc $(6,5)$ as part of the MST. The next selected arc will be either from the node 5 or the node 6, which is arc $(5,4)$. We continue similarly, and select the third arc as $(4,3)$. At the next stage, we have a tie. Two possibilities arise. They are arc $(3,1)$ or arc $(4,1)$. Note that the arc $(4,1)$ together with the existing selected arcs will create three basic arcs from node 4, whereas the arc (3,1) does maintain index balance. Thus, arc $(3,1)$ is selected. Selected arcs so far are: $\{(6,5)_1,(5,4)_2,(4,3)_3,(3,1)_4\}$. One more arc has to be selected to connect node 2, which still is an isolated node. This is either link $(4,2)$ or $(5,2)$. If $(4,2)$ is selected, it will increase the number of basic links at node 4 and similarly if the link $(5,2)$ is selected, it will increase the number of basic links at node 5. Hence in either case imbalance of basic arcs will arise at nodes 4 or 5. Therefore, all links emanating from nodes 4 and 5 are altered by adding 1. These modified arc lengths are shown in Table 4.2.

Table 4.2: Modified arc-lengths in rows 4 and 5

From\To	1	**2**	**3**	**4**	**5**	**6**
1	–	11	9	$10M4$	$16M5$	16
2	11	–	14	$11M4$	$11M5$	15
3	9	14	–	$7M4$	$14M5$	11
4	$10M4$	$11M4$	$7M4$	–	$11M4,5$	$11M4$
5	$16M5$	$11M5$	$14M5$	$11M4,5$	–	$9M5$
6	16	15	11	$11M4$	$9M5$	–

Once again, starting from node 6, the MST from Table 4.2 will be comprised of the arcs: $\{(6,5)_1,(5,4)_2,(4,3)_3,(3,1)_4\}$. Now there are three possibilities to connect node 2, for it to be part of the MST. They are links $(1,2)$ or $(4,2)$ or $(5,2)$. Note once again that the link $(4,2)$ will increase the number of basic arcs at node 4, and the link $(5,2)$ will increase the number of basic arcs at node 5, hence the link selected for the MST is $(1,2)$. Thus, the required MST will be given by: $\{(6,5)_1,(5,4)_2,(4,3)_3,(3,1)_4,(1,2)_5\}$. These selected arcs in the MST will give rise to the MST path as follows: $2 \longrightarrow 1 \longrightarrow 3 \longrightarrow 4 \longrightarrow 5 \longrightarrow 6$. Note that some of this path is contained in the tour obtained by Cowen (2011).

4.3.4 Concluding remarks

Due to the numerous applications of minimum spanning trees to communications and transportation networks, it is important to have efficient algorithms to find minimum spanning trees in weighted connected graphs. In this section, arc-weight modification theorems have been developed to create alternative MSTs in a network. The node index was restricted to a value which is less than or equal to 2. A shortest path in a non-directed network has an alternative interpretation that it gives rise to an MST of index that is less than or equal to 2 of the nodes on the shortest route. This path comprises of selected nodes which lie on the shortest path. These MST paths may have an application in the determination of the travelling salesman tour (TST). Since the MST approach is linear and the TST is NP-hard, the TST obtained through the MST is likely to reduce the complexity of the TST. Obtaining a TST through the MST will be the subject matter of Chapter 5 in this thesis.

4.4 Routing through $'k'$ Specified Nodes

The routing problem pertains to the search for a shortest route in a network connecting two designated nodes, generally called the origin and the destination. The path through $'k'$ specified nodes is a difficult problem for which no good solution procedure is known. The method considers the path from the origin node to the destination node, which must visit a set of specified nodes en route before arriving to the destination node. When the set of specified nodes is a null set, the problem reduces to an ordinary shortest route problem (Ahuja et al., 1990). However, when the set of specified nodes is not a null set, the required route is such that it has to pass through the set of specified nodes before arriving at the destination node. Such a route may have loops. Another extreme case of the $'k'$ specified node problem is when the set of specified nodes contains all nodes and one is required to return to the origin node after visiting all nodes. The problem reduces to a conventional travelling salesman problem (Bellman and Dreyfus, 1962). Complexity of this problem depends on the number of specified nodes. Saksena and Kumar (1966) solved the general routing problem through the $'k'$ specified nodes by using the functional equation technique of dynamic programming. They assumed that $0 \leq k \leq n$, where n is the number of nodes in the given network with non-negative link distances.

The requirement for a path to pass through $'k'$ specified nodes arises when one may be interested in either saving a separate trip to the given specified node or attempting to take care of a future eventuality that is likely to arise in that situation. The path through $'k'$ specified nodes is a mathematical simplification of much general situations encountered in all walks of life. Guided Tours is a multi-billion dollar business all over the world. Tours are planned and programmed meticulously. For any tour company operating in any country, a common feature is that with all major tours, they always have suggestions for a few pre-tours or post-tours, providing choice to the tourist to cover those destinations. This is an attractive offer for the tourist and also equally good for the tour company to create an additional business from the same tourist and the route through $'k'$ specified nodes concept can be used in this case.

National planning can serve as another illustration. For example, a national decision is generally not intended to address the immediate situation faced today but solution is intended to address also the likely situation going to arise tomorrow or in the near future. To be specific, consider a decision to add an extra lane to an existing road to address traffic congestion. This additional lane is not just intended to meet the current traffic requirements, but the number of additional lanes should also address future likely traffic requirements on the same sector.

4.4.1 The problem

Let the given network consist of $n+1$ nodes, which for convenience are denoted in any order by a sequence of numbers 0, 1, 2, ..., $n-1$, n. Here 0 denotes the origin node and n denotes the destination node. Let the set of specified nodes (through which the route must pass) contain k elements, where $0 \leq k \leq n$. For a case where k = 1, i.e. there is one specified node, let this node be denoted by r_j. In this case the problem reduces to two shortest route problems, i.e. finding the shortest route from the node 0 to the node r_j and shortest route from the node r_j to the final destination node n. The sum of these two shortest routes will give the required shortest path from 0 to n passing through the node r_j. One can easily see that the combinations will increase when $k > 1$.

Saksena and Kumar (1966) developed a functional equation using the principle of optimality, which is briefly presented here. They defined $D^r(i, j)$ to be the distance between the ordered pair (i, j), where i denotes the starting node and j denotes the destination, and the index r indicates the specified nodes that occur on the optimal route, r = 0, 1, 2,... They also defined f_1^{ξ} as the minimum distance from the specified node i to the final destination, passing through at least ξ distinct specified nodes (the initial node i is not to be counted as one of the specified nodes even if it is repeated en route). These definitions together with the principle of optimality result in the functional equation (4.4).

$$f_i^{\xi} = \min\{D^r(i,j) + f_j^{\xi-r-1}\} \tag{4.4}$$

for j=1, 2, 3,... and $j \neq i$. Also f_i^0 is the minimum distance from the specified node i to the final destination, without passing through any other specified node. The initial calculation of the minimum distance was given by equation (4.5).

$$f_i^1 = \min\{D(i,j) + f_j^0\} \tag{4.5}$$

Arora and Kumar (1993) reconsidered the problem of passing through $'k'$ specified edges.

Problem complexity

Similar to the travelling salesman problem, the path through $'k'$ specified nodes has complexity as a function of the number of specified nodes. This is explained in Table 4.3.

Table 4.3: $'k'$ specified node patterns

No. of specified nodes (r) **and their identification**	**Possible routes**	**No of shortest route prob. solved/Total No of new prob. solved/No. of evaluations**	**Complexity pattern 2(No. of specified nodes)** $r + ^rC_2$
$1; r_1$	$O \rightarrow r_1 \longrightarrow n$	$2/2/1 = 1!$	2(1) + 0=2
$2; r_1, r_2$	$O \rightarrow r_1 \longrightarrow r_2 \longrightarrow n$ $O \rightarrow r_2 \longrightarrow r_1 \longrightarrow n$	3 $2/5/2 = 2!$	 $2(2) +^2 C_2 = 5$
$3; r_1, r_2, r_3$	$O \rightarrow r_1 \longrightarrow r_2 \longrightarrow r_3 \longrightarrow n$ $O \rightarrow r_1 \longrightarrow r_3 \longrightarrow r_2 \longrightarrow n$ $O \rightarrow r_2 \longrightarrow r_1 \longrightarrow r_3 \longrightarrow n$ $O \rightarrow r_2 \longrightarrow r_3 \longrightarrow r_1 \longrightarrow n$ $O \rightarrow r_3 \longrightarrow r_1 \longrightarrow r_2 \longrightarrow n$ $O \rightarrow r_3 \longrightarrow r_2 \longrightarrow r_1 \longrightarrow n$	4 2 1 1 1 $0/9/6 = 3!$	$2(3) +^3 C_2 = 9$
...	...	...	...
$r; r_1, r_2, ..., r_r$	$O \longrightarrow r_1 ... \longrightarrow r_r \longrightarrow N$	$(r + 1)...0/2(r) + [^rC_2]/r!/r!$	$2(r) +^r C_2$

From Table 4.3, it is can be seen that as the value of k increases, the complexity of the problem increases with respect to the number of shortest route problems solved and also the number of evaluations required before arriving at the solution. The number of evaluations increases in a factorial manner. Therefore, a good approximate solution in a linear time would be desirable in many practical situations.

MST based approach to find the route through the specified nodes

Let the network be denoted by $N\{n, L\}$, where n is a set of nodes whose elements are $O, 1, 2, \ldots, n$ and L is an arc set with elements $\{L_{ij}\}$. Let s be the set of specified nodes, and let also this set s be a sub-set of the set of n nodes. The steps of the algorithm are as follows:

Step 1

Find the shortest path without imposing specified nodes condition by any known method. Let the links on this path be denoted by the set SP.

Step 2

Check if the shortest path obtained from Step 1 has visited all nodes in the set s? If "yes", terminate the search process and go to Step 6. Else redefine the set of specified nodes which have not been covered by the shortest path. Let the specified nodes that are not on the shortest path be denoted by s', where $s' \leq s$. Go to Step 3.

Step 3

Find the MST of the given network, starting from a node in the set s. Go to Step 4. This connected graph will be comprised of all nodes, those specified and non-specified ones. Let the links in the MST be denoted by the set 'MST'.

Step 4

Find the union of the two sets (SP and 'MST') obtained from Steps 1 and Step 3. The set of links in the union set will have links forming a path joining the origin and the destination nodes. This union set will contain all the specified nodes, non-specified nodes and the links. All nodes and links in the union set may not be required.

Step 5

Rearrange the links in the union set to form strings. For example, the links $\{(1,2),(1,7),(2,3),(3,4)$, and $(2,4)\}$ will form three strings: $\{1 \longrightarrow 2 \longrightarrow 3 \longrightarrow 4, 1 \longrightarrow 7$ and $1 \longrightarrow 2 \longrightarrow 4\}$. One of these strings will be the shortest path joining the origin and destination.

Step 6

Delete a string if it does not contain any node from the set of specified nodes and does not contain both the origin node and the destination node. Remove loops if it is beneficial to do so.

Step 7

Prepare the final path as a string joining the origin node to the destination node through the set of specified nodes.

4.4.2 Analysis and results

Let us reconsider the example solved by Saksena and Kumar (1966). Figure 4.9 shows the network diagram and the problem is to find the shortest path joining the origin node O to node 9, passing through the set of specified nodes $\{2,4,6,8\}$.

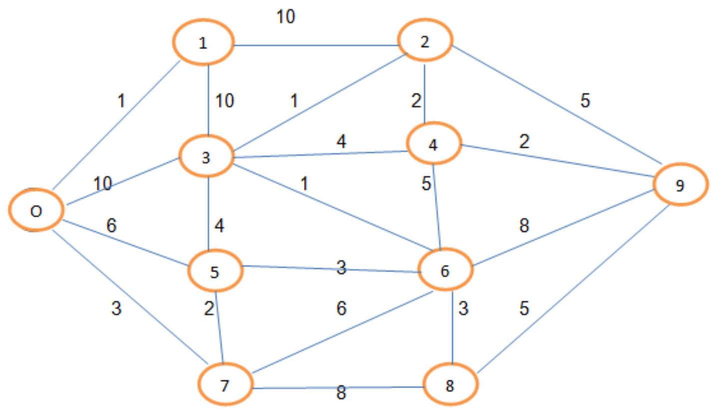

Figure 4.9: A connected graph and arc distances (Saksena and Kumar, 1966)

The shortest path using the **Step 1** is given by:

$\{O \longrightarrow 7 \longrightarrow 5 \longrightarrow 6 \longrightarrow 3 \longrightarrow 2 \longrightarrow 4 \longrightarrow 9\}$. The length of the above string is 14. Note that there are alternative paths, for example, the path:

$\{O \longrightarrow 7 \longrightarrow 5 \longrightarrow 3 \longrightarrow 2 \longrightarrow 4 \longrightarrow 9\}$ is of the same length, but we chose the previous one since it has $\{2, 4, 6\}$ three specified nodes and the alternative path has only two specified nodes $\{2, 4\}$.

Step 2: The specified node not visited by the selected shortest path is node 8. Thus the set $k = \{8\}$.

From **Step 3**, the MST starting from the node 8 will be comprised of the following links in the order of selection:

$\{(8, 6), (6, 3), (3, 2), (2, 4), (4, 9), (6, 5), (5, 7), (7, O), (O, 1)\}$.

From **Step 4**, the union of SP and 'MST' is given as:

$\{(8, 6),$ **(6,3),(3,2),(2,4), (4,9),(6,5), (5,7),(7,O)**, $(O, 1)\}$, where the common links are in bold.

Using **Step 5**, starting from the node O, we form as many strings as possible. These strings are given as follows:

String 1: $O \longrightarrow 1$

String 2: $O \longrightarrow 7 \longrightarrow 5 \longrightarrow 6 \longrightarrow 3 \longrightarrow 2 \longrightarrow 4 \longrightarrow 9$.

String 3: $O \longrightarrow 7 \longrightarrow 5 \longrightarrow 6 \longrightarrow 8$

From **Step 6**, we notice that the string 1 can be deleted as the node 1 is neither the specified node nor the destination node. The other node O has been covered in other strings. String 3 is common to string 2, except node 8. Therefore, we have a choice of either forming a loop $6 \to 8 \to 6$ or merging the node 8 with the path of string 2. In this case both possibilities will increase the cost equally. Thus there are two answers to the problem: A path with a loop, or path without a loop.

A path with a loop is: $O \to 7 \to 5 \to 6 \to 8 \to 6 \to 3 \to 2 \to 4 \to 9$, with a cost of 14 +6 = 20

A path without a loop is: $O \to 7 \to 8 \to 6 \to 3 \to 2 \to 4 \to 9$, with a cost of 14 – 5 + 8 +3 = 20.

Another Consideration

Suppose node 1 is also a specified node. In that case string 1 will not be discarded. Two alternative paths would be given as:

Alternative 1: $O \to 1 \to O \to 7 \to 5 \to 6 \to 8 \to 6 \to 3 \to 2 \to 4 \to 9$, and the cost will be 20 + 2 = 22.

Alternative 2: $O \to 1 \to O \to 7 \to 8 \to 6 \to 3 \to 2 \to 4 \to 9$, and the cost will be 20 +2 = 22.

Note that node 1 cannot be absorbed on any path as the cost associated with a loop is only two units against 10 for moving out of node 1 other than the node O.

4.4.3 Concluding remarks

In this section, we considered the routing problem through $'k'$ specified nodes. The requirement of including multiple nodes in the computation of end-to-end routing paths has many applications in real-life networks. For example, in optical, Ethernet, and mobile networks. The Canadian traveller problem and the stochastic shortest path problem are generalisations where either the graph is not completely known to the mover, changes over time, or where actions (traversals) are probabilistic. The need for the calculation of a path from a source node to a target node, which must visit a given set of nodes may arise due to network management constraints. It is easy to see that in the proposed approach, one can easily establish lower and upper bounds on path length. For example, the lower bound on the path length is the unconstrained path length, and the upper bound is the length of the path one can get from the union set in Step 5. It is desirable to establish a procedure of moving the bounds, and thus, establishing optimality of the solution.

4.5 Summary of the Chapter

In this chapter we have developed two algorithms which are key in solving some of the NP-hard problems. The key point of these algorithms was to reduce the node index n_i to a number which is less than or equal to 2. The underlining theorems that enable the node index to be changed were presented. A MST path was defined and its applications were also highlighted. Alternative interpretation of the MST-path is a shortest route passing through all the nodes. This has also been identified in this chapter. Numerical examples that illustrate the two algorithms were presented and the results were found to be in line with the results obtained by other researchers. The route through $'k'$ specified nodes algorithm was also formulated in this chapter. The complexity of this problem depends on the number of specified nodes. This problem has several applications in real life, which include the TSP and the Canadian traveller's problem all of which have several applications in real life.

Chapter 5

The Travelling Salesman Problem

The world is a book, and those who do not travel read only one page.

Saint Augustine

5.1 Introduction

The travelling salesman problem (TSP) is an NP-hard combinatorial optimisation model that has applications in OR and many other fields, for example computer science, genetics, electronics and logistics (Garg and Shah, 2011; Saranya and Vaijayanthi, 2014). The TSP belong to the NP (non-deterministic polynomial) class of difficult problems. No efficient general purpose algorithm for this problems is known. In other words, these types of problems cannot be solved in polynomial time (P). The TSP is one of the problem of P versus NP which is one of the seven Millennium Prize Problems in mathematics that were stated by the Clay Mathematics Institute in 2000 (Delvin, 2003; Calson et al., 2006). A correct solution to any of the seven problems will earn a US $1M prize (sometimes called a Millennium Prize) being awarded by the Institute. Out of the seven very difficult problems, only one was solved by the Russian mathematician Grigori Perelman in 2003 (Wiles, 2006).

Even though the TSP is computationally difficult, many heuristic and exact methods have been developed. In some instances, models have been solved involving tens of thousands of cities. In computational complexity theory the TSP belongs to the class of NP-complete problems, which means that in the worst-case, running time for the TSP algorithm may increase exponentially with the number of cities (Nadef, 2002). Currently we are not aware of any efficient exact method for the TSP model. Heuristics have been used to approximate the TSP, but the problem is that heuristic approaches do not tell us about the quality of the solution with respect to the optimal solution (Razali and Geraghty, 2011). The TSP has so many variants and so many applications in real life that it has demanded attention of many researchers.

5.1.1 The problem

Könisberg was a town in Prussia which was divided into four land regions by the river Pregel. The regions were connected with seven bridges as shown in Figure 5.1. The problem was to find a tour through the town that crosses each bridge exactly once. Leonhard Euler gave a formal solution for the problem and as it is believed, established the graph theory field in mathematics. The TSP is a problem of finding a way of moving from an origin node and return to it in such a way that each and every node is visited once and the total distance travelled is minimal. The TSP can also be represented as a mathematical programming model. A tour is a loop that connects all nodes in a travelling salesman model. The loop becomes an optimal tour if the total length is the

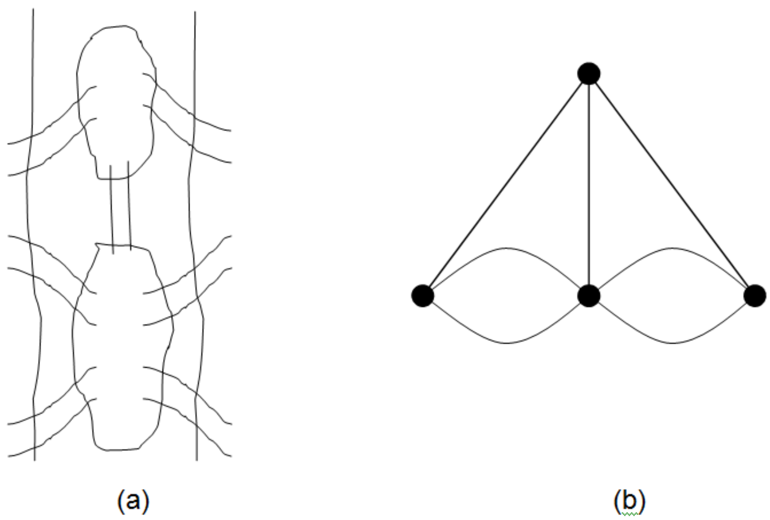

Figure 5.1: The seven bridges of Könisberg (a) and corresponding graph (b)

smallest of all possible tours. The tour is sometimes called a Hamiltonian cycle, and finding the optimal tour is known to be NP-complete.

5.2 Literature Review

Saranya and Vaijayanthi (2014) highlighted that the TSP is an NP-hard problem in combinatorial optimisation and it is important in research and theoretical computer science. The authors used Bio-inspired algorithms such as ant colony optimisation, bee colony optimisation and cuckoo search optimisation to obtain a solution to the TSP. Their main objective was to find a cyclic permutation that minimises the cost of visiting every node only once and reduce the complexities faced in existing techniques in providing the optimal solution for the TSP.

Doringo and Gambardella (1997) described an artificial ant colony capable of solving the TSP. They highlighted that ants of the artificial colony were able to generate successively, shorter feasible tours by using information accumulated in the form of a pheromone trail deposited on the edges of the TSP graph. In their research, computer simulations demonstrated that the artificial ant colony is capable of generating good solutions to both symmetric and asymmetric instances of the TSP. Their method, just like simulated annealing, neural networks and evolutionary computation, showed the successful use of a natural metaphor to design an optimisation algorithm.

Cickova et al. (2008) described the application of self-organising migrating algorithm (SOMA) to the TSP. According to the authors, SOMA was a relatively new optimisation method that was based on Evolutionary Algorithms (EAs) that are originally focused on solving non-linear programming problems that contain continuous variables. The use of EA to solve the TSP requires some special approaches to guarantee feasibility of solutions. In their article, two concrete examples were given to demonstrate the practical use of SOMA. Firstly, they applied the penalty approach as a simple way to guarantee feasibility of the solution. Then, they presented a new approach that works only on feasible solutions. The results of their study showed that SOMA gave relatively good solution for large TSPs.

5.3 The TSP through MST

Given the TSP 'n' node completely connected network, a spanning tree is a group of arcs that connects all the nodes of the network and contains no loops. A spanning tree connecting all nodes of a network becomes a minimum spanning tree (MST) when the sum of the selected arcs is smallest. There are many algorithms for finding the MST of a given network as explained in Section 4.3 of this thesis. The advantage of using the MST within the context of a TSP is that the MST can be obtained in linear time and its convergence is guaranteed.

Theorem 4.1 and Theorem 4.2 will be applied in this section in order to reduce or increase the node index (degree) of any node. An MST path is a MST such that the number of basic arcs on all nodes is less than or equal to 2. Note that Theorem 4.1 is applicable to every node, hence when this theorem is applied to a high degree node in the original MST of the given network; degree of that node can be reduced. Thus, by repeated applications of this theorem, given arc lengths can be modified so that an MST path is generated in a modified network. An MST in the form of a path is said to be desirable if the start and finish of the path is separated by a single arc, which is called a jumber (λ). If the jumber is the smallest of all the non-basic arcs, then it is called a bridge. Figure 5.2 is an example of a desirable path and bridge. Thus, when the bridge is combined with the desirable path, it results in a very important outcome, i.e. it can be used to solve the TSP model. Repeated applications of the two theorems can help to create a desirable MST path. The MST in Figure 5.2

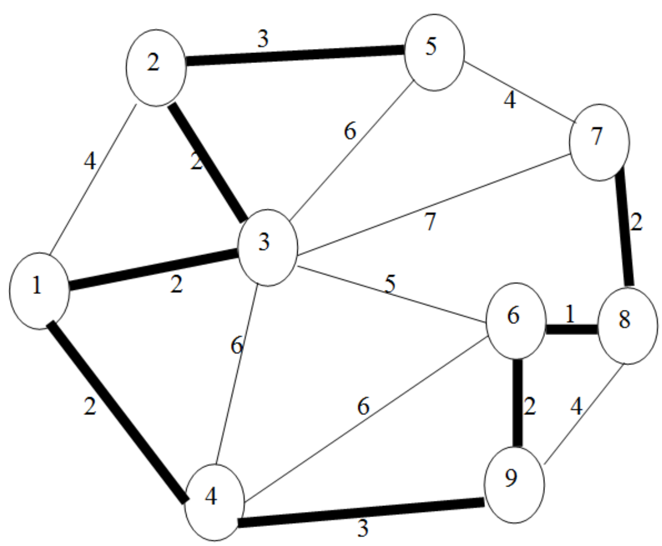

Figure 5.2: Desirable MST path, jumber and bridge.

is an example of a desirable path where the head (node 7) is separated from the tail (node 5) by the jumber (arc (5,7)). Note that there is no non-basic arc smaller than the arc (5,7), hence it becomes a bridge, which implies that an optimal tour is available. Bridges are not readily available. Theorem 4.1 has to be applied to certain nodes to make all non-basic arcs bigger than or equal to the length of the jumber arc.

5.3.1 Optimality of the tour - Theorem 5.1

The optimality of the tour obtained as above can be established by combining the desirable MST path and the bridge.

Proof by contradiction

Suppose there exists τ_0 such that:

$$L[\tau_0] \leq L[MST] + L[\lambda] \tag{5.1}$$

$L[MST]$ is the smallest connection of all the nodes. There is no other connection smaller than this. The length of the bridge is the smallest length which is not part of the MST. Then by contradiction

$$L[\tau_0] \geq L[MST] + L[\lambda] \tag{5.2}$$

Thus both equation (5.1) and (5.2) can hold only when

$$L[\tau_0] = L[MST] + L[\lambda] \tag{5.3}$$

The TSP tour can be obtained when it is possible to create a desirable MST path and a bridge; otherwise only bounds are possible.

Reduction in arc length representing the jumber – Theorem 5.2

When an MST path has been obtained and the jumber arc-length is longer than the arc-lengths that are non-basic, optimality of the solution is not established. In that case we can alter the arc lengths by subtracting a constant μ from all the arcs emanating from that node, such that the altered arc lengths are non-negative. Proof is similar to theorem 5.1.

5.3.2 The TSP tour algorithm based on the MST

The algorithm to find the TSP tour for a given network is comprised of the following steps.

Step 1:

Find the MST of the given TSP network diagram by any known method. Since MST determination can be initiated from any node of the network, apply the MST algorithm starting from the node that contains the largest arc length. In the MST development process, if there is a tie, always select the arc that does not increase degree of a node beyond 2. Go to Step 3.

Step 2:

Find the MST of the modified network and go to Step 3. Once again as in Step 1, the MST is initiated and ties are resolved as in Step 1. As arc lengths are modified, more and more ties are observed. Always select an arc that does not increase the degree of a node beyond 2, if possible.

Step 3:

Is the MST in the form of a path (i.e. all nodes in the MST have degree less than or equal to 2)? If the answer is “no”, go to Step 4 else go to Step 5.

Step 4:

Select a node ‘k′ with degree 3 or more. With the help of the neighbouring arcs, find the minimum value μ to reduce the degree at the node ‘k′. Reduction in degree is achieved by adding an appropriate minimum quantity μ to all arcs emanating from the selected node ‘k′. This way one can create an alternative and change the basic arc to reduce the degree of the node ‘k′ in the new MST. Go to Step 2.

Step 5:

The arc joining the head and tail is known as a jumber. Find if this jumber is smallest of all non-basic arcs? If the answer is “yes”, go to Step 6, else, go to Step 7.

Step 6:

Establish if the required TSP tour in the modified network is comprised of arcs forming the MST path and the bridge. Go to Step 8.

Step 7:

Since the jumber is not minimum, the bounds on the optimal TSP tour $L[\tau_0]$ can be established as follows:

$L[MST] \leq L[\tau_0] \leq L[MST] + L[Jumber]$

The length of the MST and the jumber are obtained with reference to the original data before any modification. Now reduce the length of the jumber arc by subtracting a constant so that the modified lengths are non-negative. Return to Step 2 if bounds are improved, else go to Step 9.

Step 8:

Find arc lengths of the selected arcs from the original TSP network and the length of this tour is the sum of the arc lengths of these selected arcs.

Step 9:

If in any iteration a cycle is observed, select arbitrarily the row with highest arc length and select two smallest elements. Build the connected graph on those two selected arcs. Re-establish the bounds. If bounds have been improved, terminate the process and give the best possible tour obtained so far with its bounds.

5.3.3 Results and analysis

Reconsider the 6-node completely connected network used by Cowen (2011). The corresponding network diagram is shown in Figure 5.3.

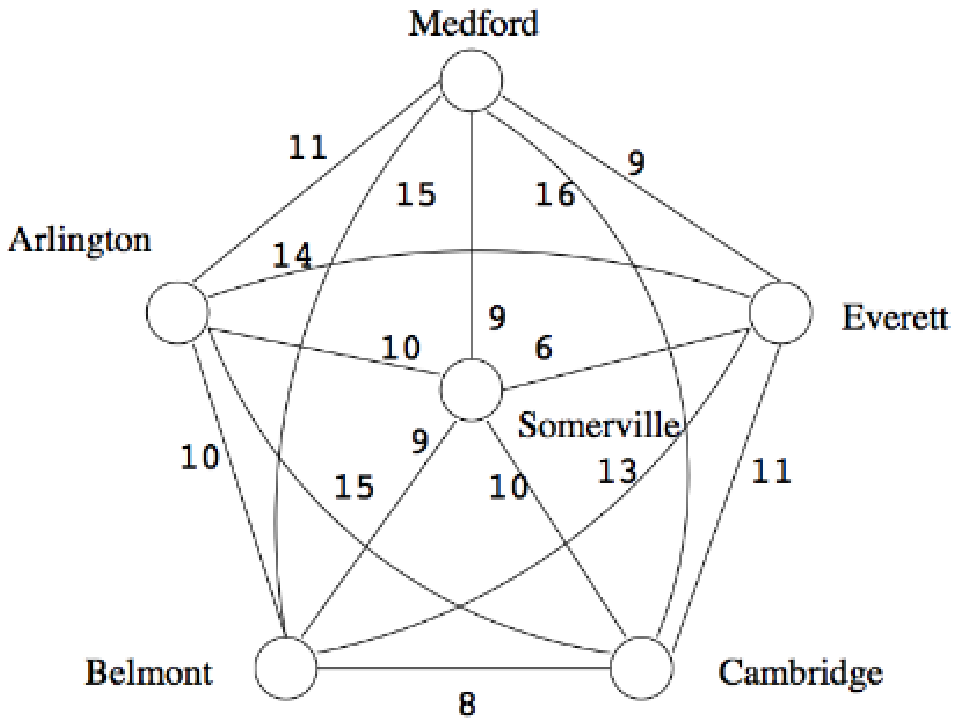

Figure 5.3: Arc-distances of the example considered by Cowen (2011)

The corresponding table is shown in Table 5.1. The optimal tour as obtained

Table 5.1: Arc-distances of the example considered by Cowen (2011)

From\To	1	**2**	**3**	**4**	**5**	**6**
1	–	11	9	9*	15	16
2	11*	–	14	10	10	15
3	9	14	–	6	13	11*
4	9	10	6*	–	9	10
5	15	10*	13	9	–	8
6	16	15	11	10	8*	–

by Cowen (2011) was as follows:

$1 \to 4 \to 3 \to 6 \to 5 \to 2 \to 1$ and the optimal tour is given by $\{9 + 6 + 11 + 8 + 10 + 11 = 55\}$.

Now the algorithm discussed in this paper is applied to this problem in Table

5.1. The largest arc length is 16, so we connect nodes 6 and 1. If we start from node 6, the first arc to be selected as part of the MST will be the arc (6,5). Next selected arc will be either from node 5 or node 6. The next selected arc is (5,4). We continue similarly, and select the third arc as (4,3). At the next stage, we have a tie. Two possibilities arise, they are arc (3,1) or arc (4,1). Note the arc (4,1) together with the existing selected arcs will create three basic arcs (degree 3) from node 4, whereas the arc (3,1) does maintain balance of basic arcs i.e. degree of 2. Hence the arc (3,1) is selected. Thus the selected arcs so far are: $\{(6,5)_1, (5,4)_2, (4,3)_3, (3,1)_4\}$. One more arc has to be selected to connect node 2, which is still an isolated node. This is either link (4,2) or (5,2). If (4,2) is selected, it will increase the number of basic arcs at node 4, and similarly if the link (5,2) is selected, it will increase the number of basic arcs at node 5. Hence in either case imbalance of basic arcs will arise at nodes 4 or 5. Therefore, all links emanating from nodes 4 and 5 are altered by adding 1. These modified arc lengths are shown in Table 5.2. Once again, starting from node 6, the MST from

Table 5.2: Modified arc lengths in rows 4 and 5

From\To	1	**2**	**3**	**4**	**5**	**6**
1	–	11	9	$10M4$	$16M5$	16
2	11	–	14	$11M4$	$11M5$	15
3	9	14	–	$7M4$	$14M5$	11
4	$10M4$	$11M4$	$7M4$	–	$11M4,5$	$11M4$
5	$16M5$	$11M5$	$14M5$	$11M4,5$	–	$9M5$
6	16	15	11	$11M4$	$9M5$	–

Table 5.2 will be comprised of the arcs: $\{(6,5)_1, (5,4)_2, (4,3)_3, (3,1)_4\}$. Now there are three possibilities to connect node 2 to be part of the MST. They are links (1,2), (4,2) or (5,2). Note once again that link (4,2) will increase the number of basic arcs at node 4, and the link (5,2) will increase the number of basic arcs at node 5, hence the link selected for the MST is (1,2). Thus, the required MST will be given by: $\{(6,5)_1, (5,4)_2, (4,3)_3, (3,1)_4, (1,2)_5\}$. These selected arcs in the

MST will give rise to the MST path as follows:

$2 \to 1 \to 3 \to 4 \to 5 \to 6$. The arc (2,6) is the jumber with length 15. Thus, the bounds are:

$L[MST] \leq L[\tau_0] \leq L[MST] + L[Jumber]$ that is to say, TSP tour is greater than the MST, which is 43 and the TSP is less than the MST+ Jumber, which is 58. The minimum non-basic arc has length 10 units. Hence we modify the jumber row, i.e. row 2 and reduce each element by 5 units to bring the jumber length equal to the minimum non-basic arc length. This is given in Table 5.3.

Table 5.3: Modified arc lengths from Table 5.2

From\To	1	**2**	**3**	**4**	**5**	**6**
1	–	11	9	$10M4$	$16M5$	16
2	$6M2$	–	$9M2$	$6M4,2$	$6M5,2$	$10M2$
3	9	$9M2$	–	$7M4$	$14M5$	11
4	$10M4$	$6M4,2$	$7M4$	–	$11M4,5$	$11M4$
5	$16M5$	$6M4,2$	14	$11M4,5$	–	9
6	16	$10M2$	11	$11M4$	9	–

The first two elements of the MST from Table 5.3 are given by: $\{(5,6)_1, (5,2)_2\}$. The next element has a tie, i.e. it can be either (2,1) or (2,4). Thus, three elements of the two alternative MSTs will be given by $\{(5,6)_1, (5,2)_2, (2,1)_3\}$ and $\{(5,6)_1, (5,2)_2, (2,4)_3\}$. For the fourth element, we consider two cases separately.

Case 1

Consider the string $\{(5,6)_1, (5,2)_2, (2,1)_3\}$. The fourth element in the above string will be given by (2,4). Thus the updated string will be $\{(5,6)_1, (5,2)_2, (2,1)_3, (2,4)_4\}$, and the final element to be added to form the MST will be (4,3); resulting in $\{(5,6)_1, (5,2)_2, (2,1)_3, (2,4)_4, (4,3)_5\}$. This selection of arcs in the MST does not form a path. There are three basic arcs at node 2. Before investigating it further, let us consider the other string.

Case 2

Consider the string $\{(5,6)_1, (5,2)_2, (2,4)_3\}$. The fourth element will be (2,1) and the final element will be (4,3). The full string will be $\{(5,6)_1,(5,2)_2,(2,4)_3,(2,1)_4,(4,3)_5\}$. Both alternatives have resulted in the same MST. The nearest neighbour is arc (1,4). Adding 4 to all arcs emanating from node 2 gives the results shown in Table 5.4. The MST is comprised of the following arcs: $\{(6,5),(6,4),(4,3),(3,1)\}$

Table 5.4: Modified arc lengths from Table 5.3

From\To	1	**2**	**3**	**4**	**5**	**6**
1	–	11	9	$10M4$	$16M5$	16
2	$11M2$	–	$13M2$	$10M4,2$	$10M5,2$	$14M2$
3	9	$13M2$	–	$7M4$	$14M5$	11
4	$10M4$	$10M4,2$	$7M4$	–	$11M4,5$	$11M4$
5	$16M5$	$10M4,2$	$14M5$	$10M5$	–	$9M5$
6	16	$14M2$	11	$11M4$	$9M5$	–

The last link has a tie between (4,2) and (5,2). The link (5,2) is selected as it will give rise to an MST path. The MST path is given by:
$1 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 2$ with a length of $(9+6+10+8+10=43)$. The jumber is (1,2) with length 11. The length of the TSP is given by $L[MST]+L[Jumber]$ where MST length = 43 and and length of the jumber = 11. Therefore, the TSP tour length = 54. Our solution is less that the solution obtained by Cowen which was 55.

5.3.4 Concluding remarks

The approach proposed in this thesis is likely to give new directions for research in search for an efficient algorithm to solve the difficult TSP model. The strength of the proposed approach is that it is based on the MST tours. In addition, the proposed method does not generate sub-problems that can explode as is the case with most of the branch and bound related methods (Padberg and Rinaldi, 1991). The proposed approach uses only MST and addition/subtraction operations which are very simple. The MST path used in this approach also has a very important real life application. It can be used to find quickest routes when time is the most important factor. It was assumed that the given network is a completely connected network. It may be desirable to extend this idea to connected graphs. The method discussed in this section can result either in an optimal solution or bounds on the TSP tour. It would be desirable to develop a procedure to improve bounds, when optimal solution is not known.

5.4 A Heuristic for the TSP based on the MST Technique

Let a connected network of n nodes and m edges be denoted by $G(n, m)$. After excluding a node p and all edges that emanates from the node p, the remaining network is denoted by $G'(n-1, m-d_p)$, where d_p is the number of arcs that are emanate from the node p. Note that the network $G'(n-1, m-d_p)$ is a $(n-1)$-node connected network. The MST of the network $G'(n-1, m-d_p)$ will be comprised of $(n-2)$ arcs. Let the length of the MST be denoted by $LMSTG'(n-1, m-d_p)$.
Observation 1: The MST can be converted to a MST path by repeated applications of the index-balancing theorem discussed in Section 4.3 of this thesis. After i iterations of the index-balancing theorem $(i = 1, 2, \ldots, l)$, let the length of the MST be denoted by $L^iMSTG'(n-1, m-d_p)$, where:

$$LMSTG'(n-1, m-d_p) \leq L^1MSTG'(n-1, m-d_p) \leq ... \leq L^iMSTG'(n-1, m-d_p) \leq \\ ... \leq L^lMSTG'(n-1, m-d_p) \tag{5.4}$$

i.e. each iteration of the index-balancing theorem leads to an increase in the MST length. At the l^{th} iteration, when the index at each node is ≤ 2, the MST becomes a path. The imbalance of index numbers in the MST with regard to the MST path will always be an even number, i.e. the high and low index values will be equal and the total index value will be $2(n-2)$.

Observation 2: The number of arcs in a TST in $G(n,m)$ will be n. The sum of the two selected arcs together with the MST of $G'(n-1,m-d_p)$ will constitute a collection of n arcs, but a feasible TST solution will be realised only when the length of the selected two arcs is added to the MST path that starts and finishes at the nodes q and k. Since we are interested in a path between the nodes q and k, which passes through all the other remaining nodes, we can set the link $(q,k)=\infty$, if it exists. Simply, the link (q,k) will form a loop with the links (p,q) and (p,k). Let the length of this MST path be $L^{l}MSTG'(n-1,m-d_p)$. Thus, a feasible TST will be formed by the two selected links and the MST path giving an upper bound on the TST.

Observation 3: After establishing a feasible TST for a particular q and k node combination, the search for a better tour commencing from any other combination of two links emanating from the node p can be fathomed at the i^{th} iteration $(i=0,1,2,\ldots,l)$. The index-balancing theorem gives:

$$L(p,q)+L(p,k)+L^{i}MSTG'(n-1,m-d_p)\geq LTSTG(n,m) \qquad \textbf{(5.5)}$$

where $LTSTG(n.m)$ is the current upper bound.

5.4.1 The three components of the heuristic

Component One: Determination of a MST path joining nodes q and k

For the given network $G(n,m)$, the node $p\in n$ and two associated links $\{(p,q),(p,k)\}\in m$, the focus is to find the MST path joining the nodes q and k passing through all the remaining nodes in the network $G'(n-1,m-d_p)$. Note that the shortest path joining the nodes q and k can be determined by any method, and that path has an alternative interpretation of the MST path (i.e. all intermediate nodes on the path have an index 2 and the nodes q and k have index 1). Let the number of nodes on the shortest path be given by K, then the number of isolated nodes of the network $G'(n-1, m-d_p)$ are given by $(n-3-K)$. Using the greedy approach, all these isolated nodes can be connected to the shortest path between the nodes q and k, thus forming a connected tree. Since each node in the shortest path is balanced with regard to its index value when an additional arc is connected to it; it will give rise to index imbalance by making it a high degree node with index value less than 2 for all nodes other than q and k, and for the nodes q and k, imbalance will arise if node index is less than 1. All these high index nodes will have to be treated one by one to reduce the node index so that the node index for the nodes q and k is 1 and for all other nodes, the index is 2. This objective can be achieved in many ways. One simple procedure is described next:

Step 1:

In the network $G'(n-1, m-d_p)$, set the link $(q, k) = \infty$, if it exists .

Step 2:

Find the MST of the network $G'(n-1, m-d_p)$, which will be comprised of $(n-2)$ links. The sum of these edges gives the MST length, denoted by $LMSTG'(n-1, m-d_p)$.

Step 3:

Using the index-balancing theorem, convert this spanning tree to a MST path joining the nodes q and k, passing through all the remaining nodes in $G'(n-1, m-d_p)$.

Step 4:

The TST length will be: $LTSTG(n, m) = L(p, q) + L(p, k) + L^l MSTG'(n-1, m-d_p)$. Since this is a feasible TST, it becomes an upper bound. Note that it may not be an optimal TST solution.

Component two: Identification of the node p

The basic arc have already been defined as the arc that belongs to the TST or MST otherwise it is non-basic. We have also defined the index of a node as the number of basic arcs emanating from that node in the given TST or MST. The node with lowest index value in $G(n, m)$ is selected as the node p. If lowest index nodes are more than one, we calculate the penalty associated with all the tied nodes and select lowest index and highest penalty node.

Since only two arcs emanating from a node will be members of the TST (or basic) and the rest will be non-basic, the best combination from any particular node will be to include the minimum and the second minimum arcs in the TST. However, since the TST has to satisfy many other conditions, it would not be feasible to include the two minimum from each node in the TST. In that case, one has to include arcs of lengths greater than the minimum and the second minimum. Thus, the third minimum arc length from that node will be an alternative that can be used for minimising an increase in the total length. When the third best arc is used, the penalty p_{n_i} associated with the node n_i is defined as follows:

$$p_{n_i} = 3rd\ Minimum\ arc\ length - 2nd\ Minimum\ arc\ length. \tag{5.6}$$

Thus, given the network, the penalties can be easily calculated for each node and it will be infinity if a node has only two arcs emanating from it. In other words, the two arcs must belong to the TST. A node associated with lowest index and high penalty is a potential candidate for the proposed TST, yet there is no guarantee that it will lead to the optimal tour. The selected node p and the two associated links with minimum cost are denoted by (p, q) and (p, k). The complexity of the proposed approach will depend on the index value of the selected node p. If the lowest index value in the given network is r, where $r \geq 2$, the optimum TST will require determination of ${}^{r}C_2$ number of travelling salesman tours for different combinations of two arcs from the node p. Thus, in a completely connected network $G(n, m)$, the maximum number of problems that will be solved is given by ${}^{(n-1)}C_2$.

Identification of the two links from the node p is done by establishing all possible combinations and arranging them in an increasing order with respect to the cost. Combination 1 will be associated with the two minimum. If there is a tie, we arrange them in non-decreasing order and call them $1, 2, \ldots, l$.

Component three: The index balancing theorem

The MST of an $(n-1)$ node network will have $(n-2)$ links, and a total of $2(n-2)$ index values. In a MST, the index values can be a number such that each node can have index between $1 \leq n_i \leq (n-1)$ and the total will be equal to $2(n-2)$. An application of the index balancing theorem can decrease the index value at a high index value node and increase the index value at a node of low index value. According to Theorem 4.1, adding the same constant to all arcs emanating from the same node does not change their relative merit, but can create alternatives for the MST. Thus, additional quantity can create alternatives to obtain new MSTs which balance the indexing.

5.4.2 Determination of an optimal TST in the network $G(n, m)$

The following steps are followed.

Step 1:

For the given network, identify the following:

1. The node p.

2. The index of the node p, let it be r, for $r \geq 2$.

3. The number of combinations, two at a time, is given by ${}^{r}C_2 = 1, 2, \ldots, l$. Let us denote as a function of length, these combinations by $C_1, C_2, \ldots, C_l$ and their corresponding lengths by $L(C_1) \leq L(C_2) \leq .... \leq L(C_l)$.

4. Identify the nodes q and k associated with the least cost combination, C_1.

5. Set the link $(q, k) = \infty$ in the network $G'(n-1, m-d_p)$.

6. Find the MST of the network $G'(n-1, m-d_p)$, where the link $(q, k) = \infty$.

7. The length of the MST is denoted by $LMSTG'(n-1)$.

8. The number of index-imbalances in the $MSTG'(n-1)$ is denoted by $NoMSTG'(n-1) = N$, for $i = 1, 2, \ldots, N$.

9. Apply the index-balancing until the MST becomes the MST path between the nodes q and k for the combination C_1.

10. Find a feasible TST and denote its length by $LTSTG(n) = L(p, q) + L(p, k) + LMSTPG'(n-1)$.

11. The UB = $LTSTG(n)$.

12. Set $k = 1$.

Step 2:
Set $k = k + 1$. If $k + 1 > k$, the current UB is the required optimal TST.

Step 3:
For the k^{th} combination from the node p, identify:

1. The two associated arcs with the k^{th} combination.

2. The two nodes that will have the index 1 in this k^{th} combination. Call these two nodes q and k.

3. Set the link $(q, k) = \infty$ in the network $G'(n-1, m-d_p)$.

4. Find the MST of the $G'(n-1, m-d_p)$, from Step 3.3.

5. The number of index imbalances in 4 above.

6. Set $i = 0$

7. For the current combination, check if:
$LArc1 + LArc2 + L^i MSTG'(n-1) \leq LTSTG(n)$ If satisfied, go to 8. If not, go to Step 2.

8. Set $i = i + 1$

9. Apply the i^{th} index balancing.

10. If the MST is a path satisfying the node index requirement, check Step 3.7

Step 4:
Terminate the search for this combination and go to Step 5.
Step 5:
If a feasible TST is obtained and it is less than the current upper bound, replace the existing UB by the new value and return to step 2.

5.4.3 Results and analysis

Example 1

To find the TST for the network $G(9, 15)$ given in Figure 5.4, we apply the techniques highlighted in Section 5.4. For the network given in Figure 5.4, we calculate node index of each node. These index values are given in Table 5.5. From Table 5.5, the minimum index is on node 7, hence node 7 is selected as the node p. Although penalties are not required, however, just for completeness, these penalties are as given in Table 5.6. Once again, note that node 7 is associated with a penalty of infinity. The two links will be $(7, 5)$, $(7, 8)$. Thus,

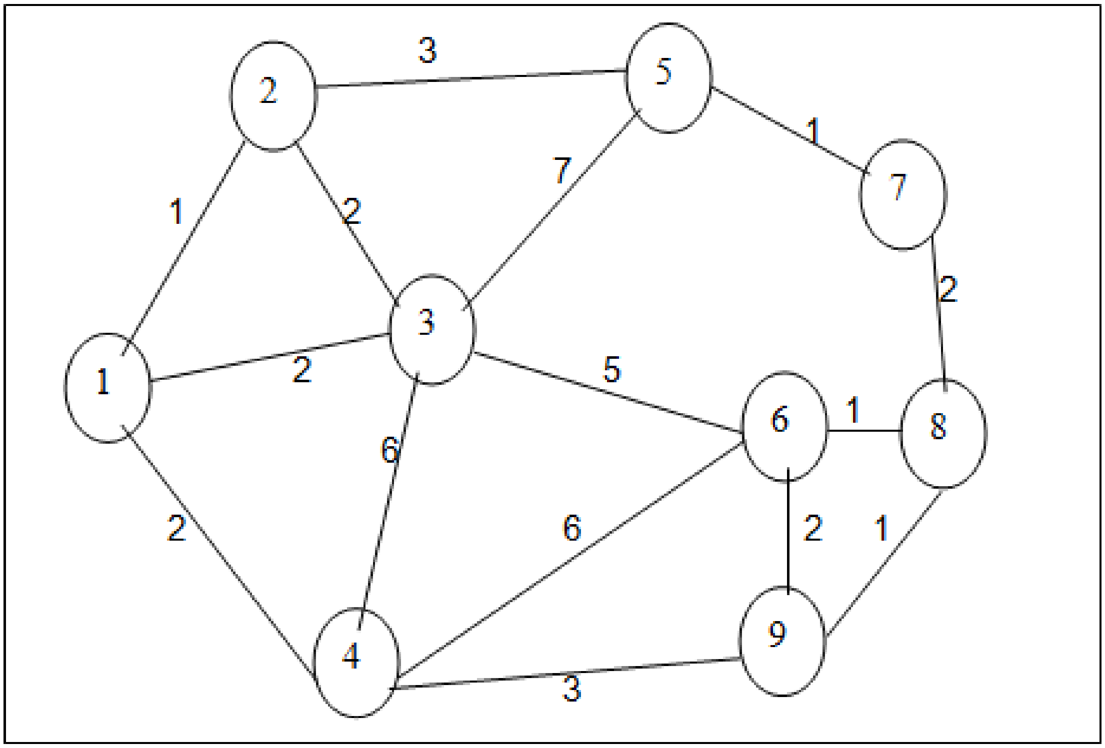

Figure 5.4: Network to determine the TST

Table 5.5: Index values of nodes

Node	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Index	3	3	5	4	3	4	2	3	3

Table 5.6: Penalties associated with each node

Node	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Penalty	0	1	3	3	4	3	∞	1	1

the network $G'(8, 13)$ is given in Figure 5.5. Note that the link $(5, 8)$ does not exist.

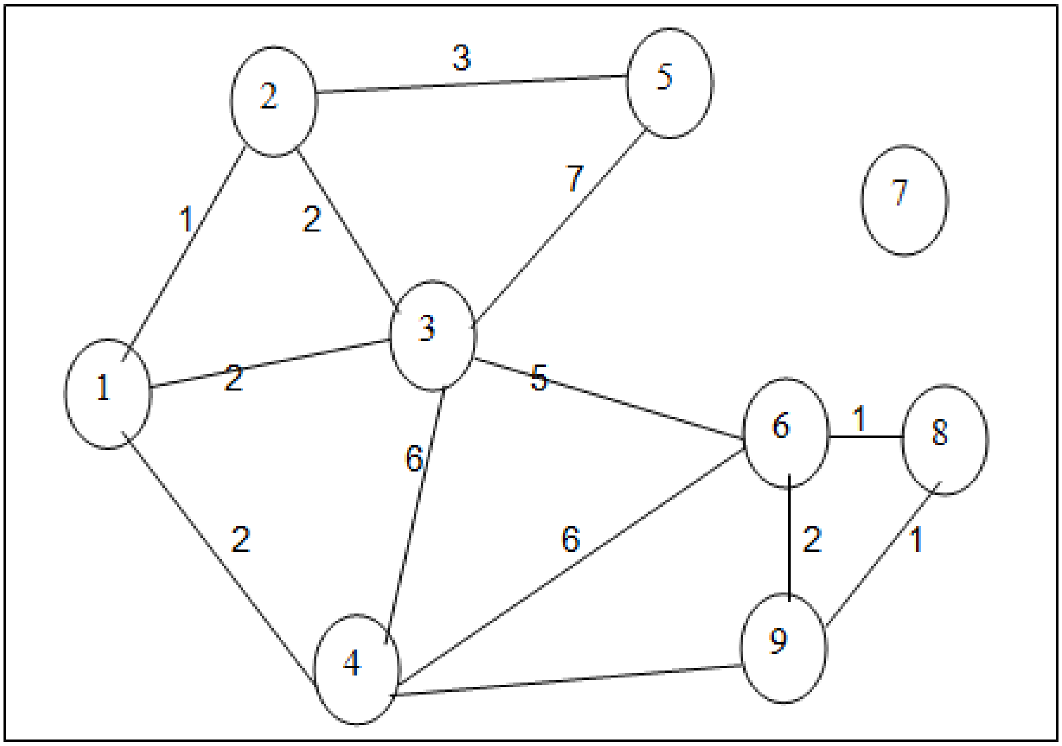

Figure 5.5: The network $G'(8, 13)$, that excludes node 7

The problem is to find the MST path in Figure 5.5 joining the nodes 5 and 8, and passing through nodes 1, 2, 3, 4, 6 and 9. The MST of the network in Figure 5.5 will be as shown in Figure 5.6. The length of this MST, denoted by $LMST = 13$. Hence the $LB = 13 + 1 + 2 = 16$, which is not feasible. From the MST network in Figure 5.6, it is clear that nodes 1 and 8 are high degree nodes. The index at node 1 is 3, which should be 2 and the index at node 8 is 2, which should be 1 as it is a terminal node. Nodes 3 and 6 are low degree nodes as their current degree is 1.

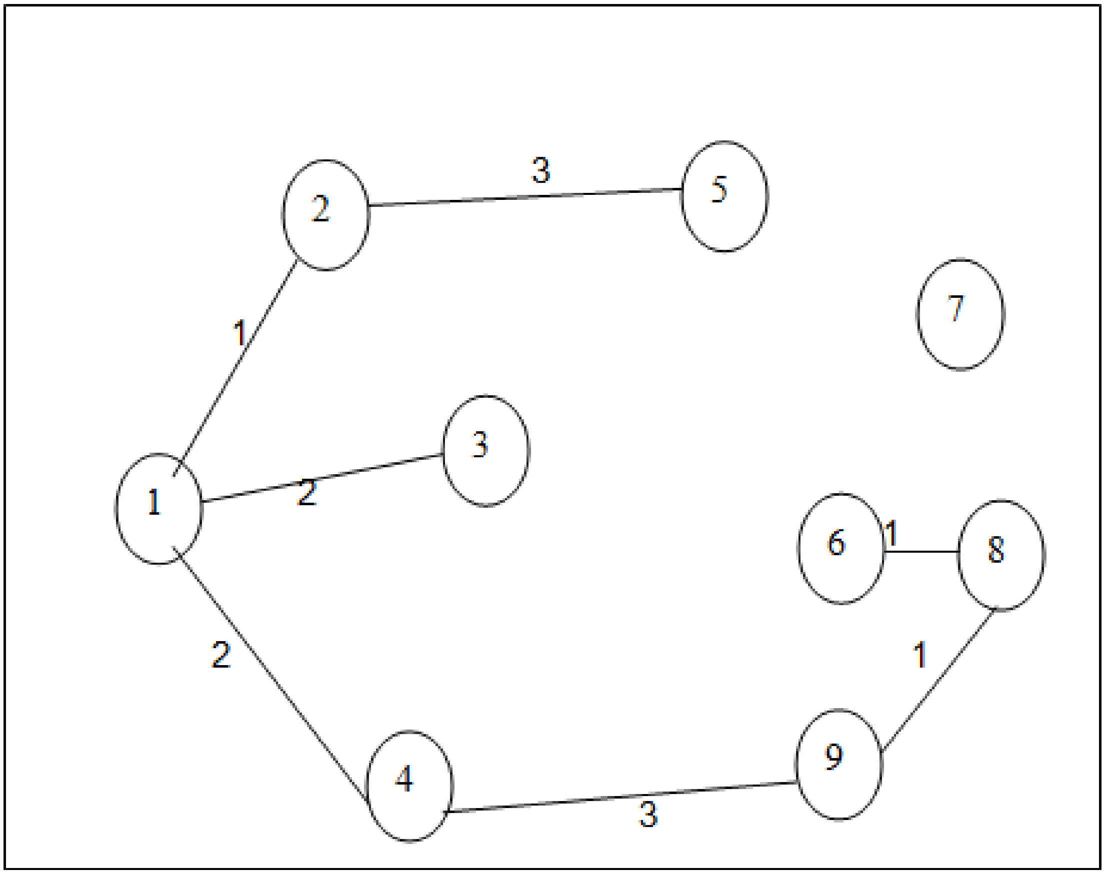

Figure 5.6: The MST of $G'(8, 7)$

Adding 1 to all arcs emanating from node 1 will change the MST selection of arc (2,1) to arc (2,3) and balance of degree at nodes 1 and 3 will be satisfied. Similarly, adding 1 unit to all arcs emanating from node 8 will change the MST selection to the arc (9,6) replacing the arc (9,8). This change will fix up the degrees at nodes 6 and 8. Thus, the final selection will be as shown by dark solid lines in Figure 5.7.

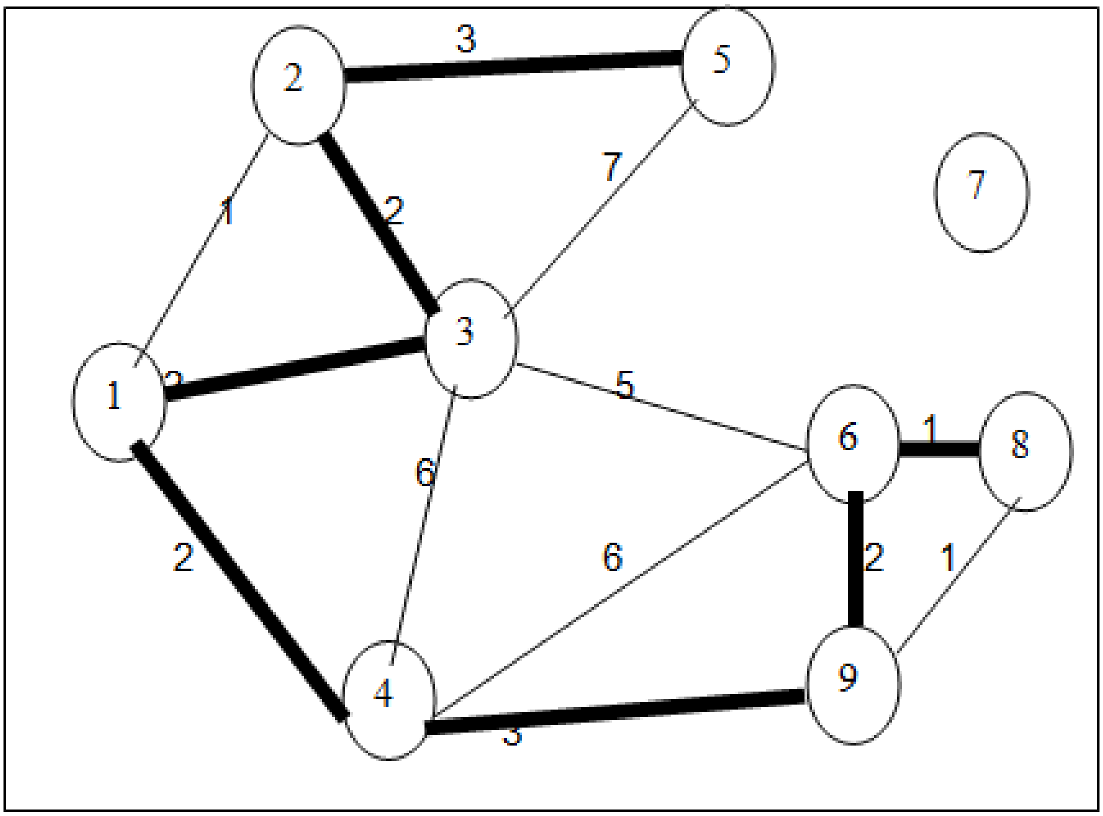

Figure 5.7: The modified MST path is shown in dark lines

The TST will be comprised of arcs $\{(7,5),(7,8)\}$ giving the TST as $\{(5,2),(2,3),(3,1),(1,4),(4,9),(9,6),(6,8)\}$. The length of the TST will be 18, which is optimal and is shown in Figure 5.8.

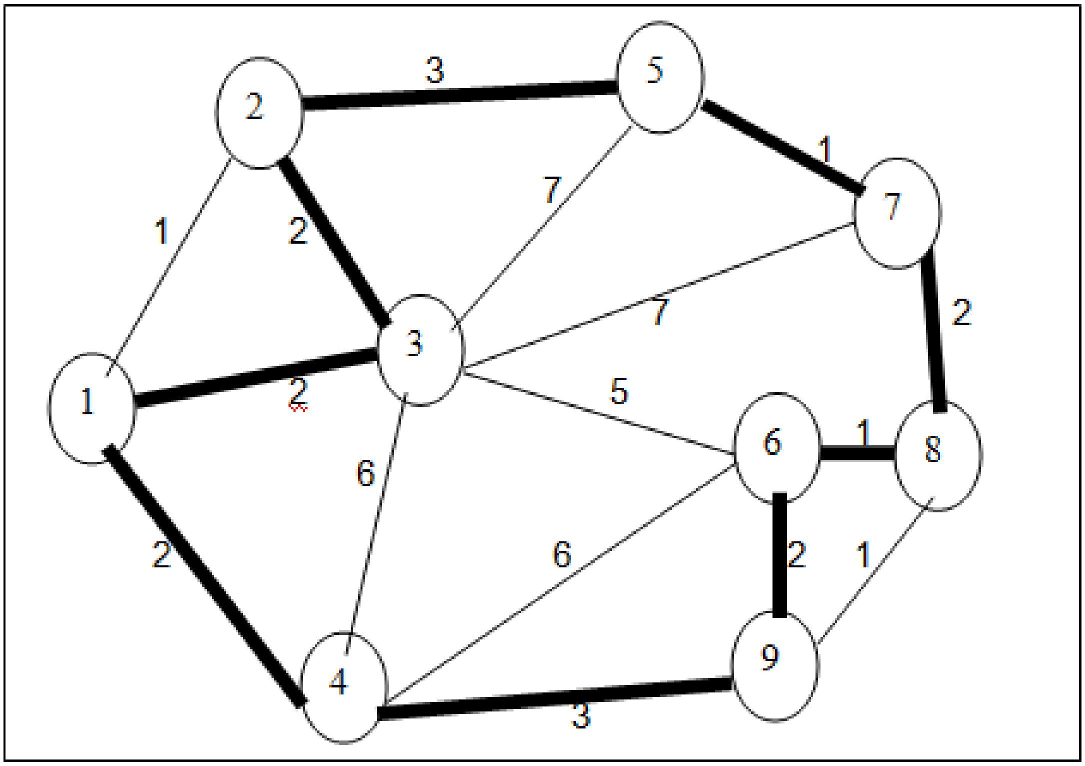

Figure 5.8: The optimal TST shown in dark lines

Example 2

Let us consider again the example that was considered in Section 5.3.3 which was also solved by Cowen (2011). The network diagram is as shown in Figure 5.9.

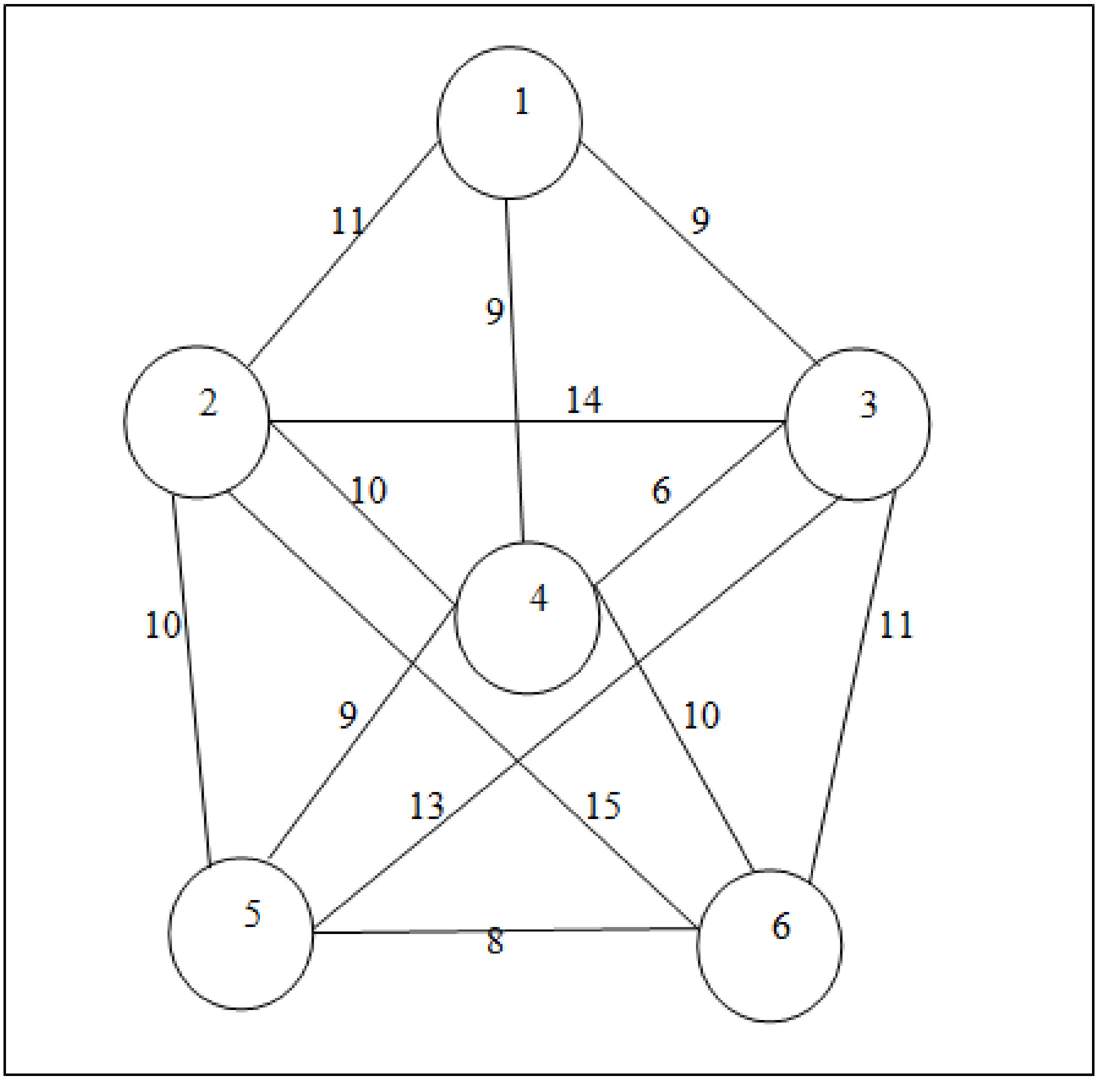

Figure 5.9: The network considered by Cowen (2011)

The node index and associated penalties for the network in Figure 5.9 is given in Table 5.7. The index is lowest at node 1, therefore it is selected as the

Table 5.7: Node index and penalties for the network in Figure 5.9

Node	**1**	**2**	**3**	**4**	**5**	**6**
Index	3	5	5	5	4	4
Penalty	2	1	2	1	1	1

starting node p. For full investigation, the number of combinations will be 3 given by $\{(1,3).(1,4)\}$ with distance 18; $\{(1,2),(1,3)\}$ with distance 20; and $\{(1,2),(1,4)\}$ with distance 20. The network in Figure 5.10 is obtained after removing the node 1 and all the arcs emanating from this node. Since arcs (1,4) and (1,3) are the two minimum arcs from node 1, we first set the link (3,4) to be infinity and find the MST path joining nodes 3 and 4 passing through all the remaining nodes, which are nodes 2, 5 and 6.

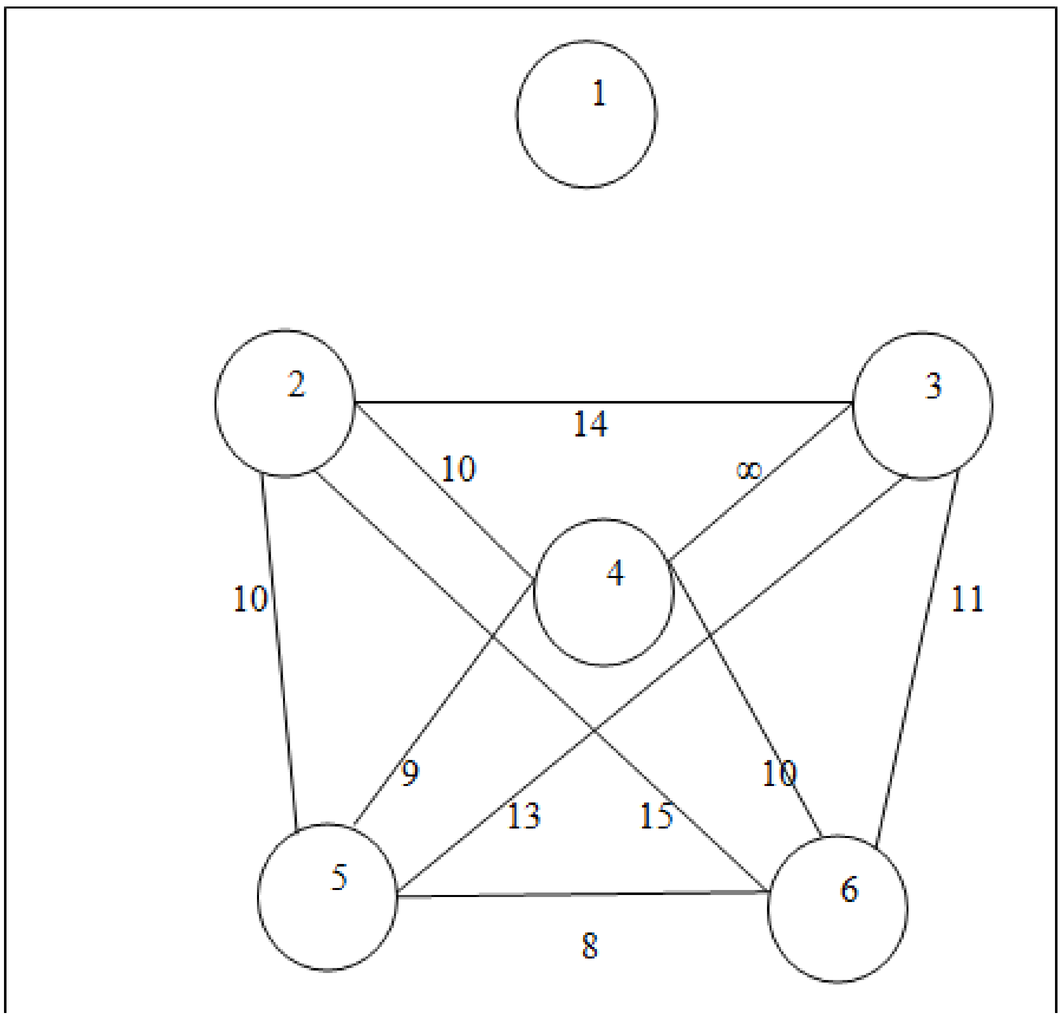

Figure 5.10: Network $G'(5,10)$

The MST is comprised of links $\{(4,5)_1, (5,6)_2, (5,2)_3$ and $(6,3)_4\}$. The total distance is 38. Note that node 5 is a high index node as its index value is 3, which should be 2. Similarly, node 2 is a low index node with index value 1, which should be 2. If we add 1 unit to all links emanating from node 5, an alternative will be created and we can select $(2,4)$, replacing the link $(4,5)$. Thus, the MST will be $(4,2), (2,5), (5,6)$ and $(6,3)$. Total distance will be 39, which is a feasible TST. The length of this TST will be $18 + 39 = 57$, resulting in the $UB = 57$. For optimality, one has to investigate the remaining two more combinations, i.e. $\{(1,2), (1,3)\}$ with a distance of 20 and $\{(1,2), (1,4)\}$ with a distance of 20.

Investigate $\{(1,2), (1,3)\}$ with a distance of 20. Set the link $(2,3)$ equal to infinity and find the MST, which will be formed of links $(2,5), (5,6), (6,4)$ and $(4,3)$ as shown in Figure 5.11.

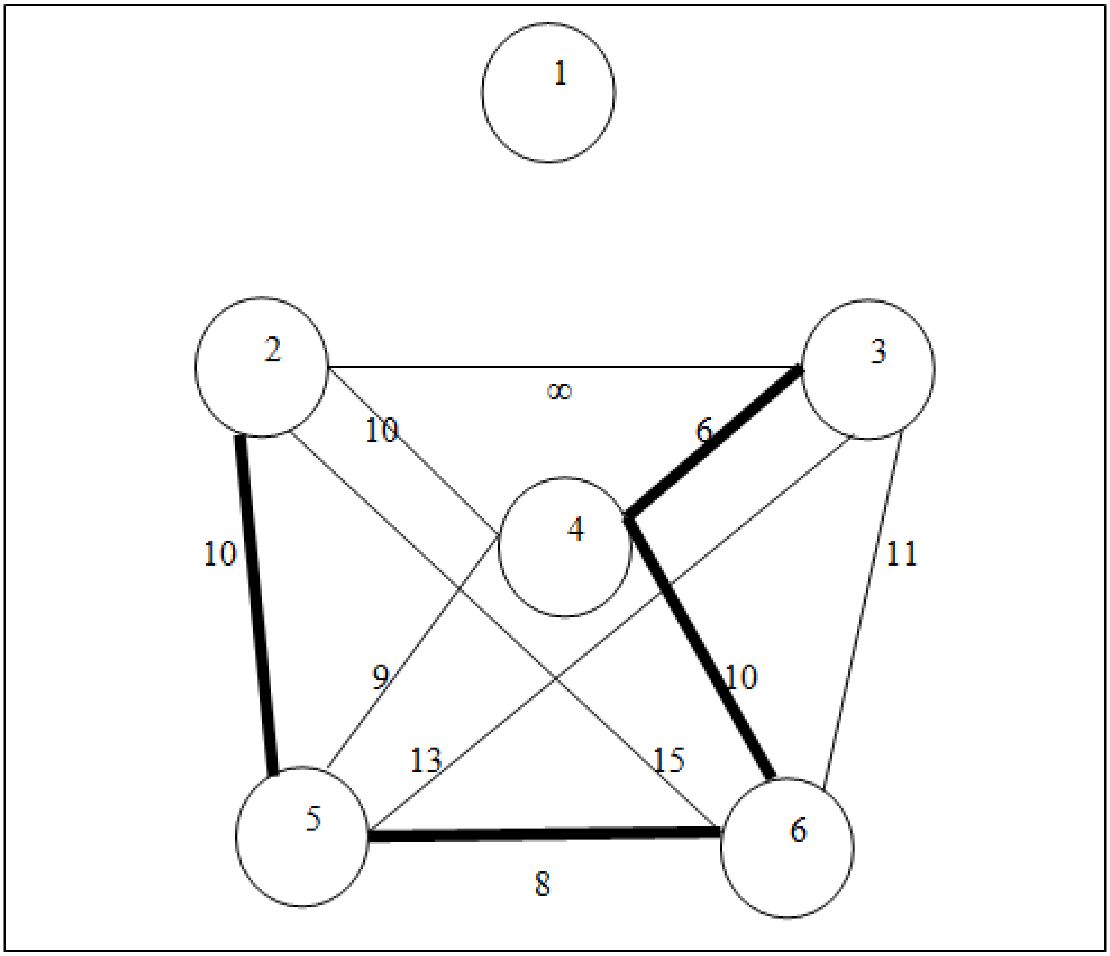

Figure 5.11: The MST path

The TSP will be given by $L(1,3)+L(1,2)+MST\ path$. This will be $(1,3)$, $(1,2)$, $(2,5)$, $(5,6)$, $(6,4)$, $(4,3)$ with length $9+11+10+8+10+6=54$. Thus, the UB is replaced by its new value of 54. This solution is as shown in Figure 5.12 with the TSP solution as $1-2-5-6-4-3-1$ giving a total distance of 54.

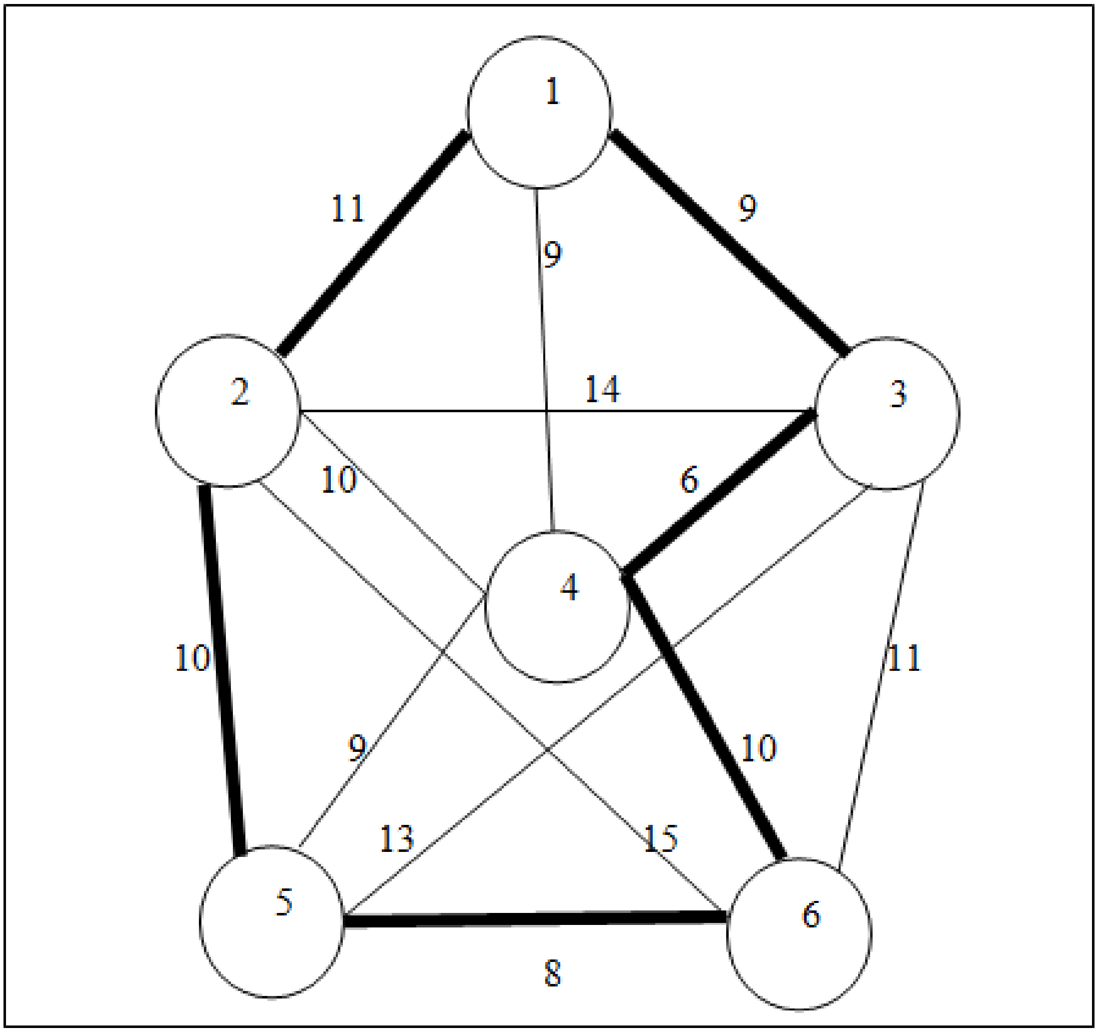

Figure 5.12: The TST solution

5.4.4 Concluding remarks

The index value plays a major role in the proposed approach. The proposed approach is best when at least one node has a low index value. If the lowest index value is m, the number of sub-problems solved will be given by ${}^{m}C_2$. In a completely connected n node network, the worst case will have ${}^{(n-1)}C_2$ combinations. The proposed heuristic converts the problem in three parts to establish an upper bound. Real test for the proposed approach is to develop a software and apply it to a large number of randomly generated data and compare the proposed approach with existing approaches. The approach discussed in this section uses link-weight modification to obtain alternative MSTs, which eventually have a TST interpretation. The link-weight modification was used earlier by Munapo et al. (2008) for a directed network, and later Kumar et al. (2013) applied the link-weight modification idea for the determination of a shortest path in a non-directed network.

5.5 Summary of the Chapter

This chapter presented an MST approach that can solve the TSP. The index value of a node played a major role in the proposed approach. The proposed approach is best when at least one node has a low index value. If the lowest index value is m, then the number of sub-problems solved will be given by ${}^{m}C_2$. In a completely connected n node network, the worst case will have ${}^{(n-1)}C_2$ combinations. The approach discussed in this chapter uses link-weight modification to obtain alternative MSTs, which eventually have a TST interpretation. The worst case situation will arise in the case of a completely connected n node network when each node will have an index value of $n-1$. The proposed approach will work more efficiently when a node in the given network happen to have a low index value.

The MST approach was extended and a heuristic to find the travelling salesman tour (TST) in a connected network was formulated. The approach first identifies a node and two associated arcs that are desirable for inclusion in the required TST. If we let this node be denoted by p and two selected arcs emanating from this node be denoted by (p, q) and (p, k), then we find a path joining the two nodes q and k passing through all the remaining nodes of the given network. A sum of these lengths, i.e. length of the links (p, q) and (p, k) along with the length of the path that joins the nodes q and k passing through all the remaining nodes will result in a feasible TST. A simple procedure was outlined to identify: (1) the node p, (2) the two corresponding links (p, q) and (p, k), and (3) the path joining the nodes q and k passing through all the remaining nodes. The approach is based on the MST; hence the complexity of the travelling salesman tour is reduced. The network in the present context has been assumed to be a connected with at least two arcs emanating from each node.

Part II

Resource Allocation and Distribution Models

Chapter 6

Transportation and Assignment Problems

To cross the seas, to traverse the roads, and to work machinery by galvanism, or rather electro-magnetism, will certainly, if executed, be the most noble achievement ever performed by man.
Alfred Smee

6.1 Introduction

The assignment problem is one of the fundamental combinatorial optimisation problems which has been modified, extended and applied many times (Pentico, 2007). Degeneracy in a linear programming (LP) model can cause difficulties, as the value of the objective function may not improve in successive degenerate iterations. Sometimes a solution may be optimal but the test for optimality fails to recognise optimality of that solution due to wrong selection of degenerate variables. Since the transportation and assignment models are degenerate

LP models, where order of degeneracy varies from 1 to n (n is the number of supply or demand points) within the context of a LP model. Special methods were developed to deal with these special degenerate models. A feasible solution in a balanced assignment model of order n is a degenerate solution by order $n-1$ in the context of a transportation model.

Degeneracy in LP problems has been extensively studied since it can cause cycling in simplex-type algorithms, unless special rules are enforced (Hung et al., 1986). Degeneracy has been considered a 'bad' phenomenon in the folklore of LP. Degeneracy is also found even in specialised areas of LP, such as network flow problems, transportation problems and assignment problems. Hung et al. (1986) pointed out that the more degenerate a transportation problem is the fewer extreme points (vertices) it contains in its polytope. The authors further stated that the more degenerate a transportation problem is the more feasible bases it contains. Therefore, degeneracy may affect the efficiency of a primal simplex algorithm. Thus far, however, there has been no systematic study of the effects of degeneracy on a primal code. To isolate the effects of degeneracy on an algorithm, it is necessary to generate problems with a controlled amount of degeneracy. This however has proved difficult. Chandrasakaran et al. (1982) proved that verifying whether a given transportation problem is degenerate is itself an NP-complete problem. Perhaps the best known, most widely used, and most written about method for solving the assignment problem is the Hungarian method. Originally suggested by Kuhn (1955), variants of the Hungarian method were developed by Balinski and Gomory (1964). Therefore, one objective of this thesis is to present a method for solving degenerate problems.

Over the past decade, company operations have increasingly emphasised demand and revenue management, as firms seek to exploit sources of supply and demand flexibility to increase profit margins. This has led to a number of new models that focus on profit maximisation by accounting for both the costs and revenue implications associated with operations decisions. Models for profit-maximising extension of the generalised assignment problem (GAP) in which both the assignments of jobs to available agents and the degree of resource consumption associated with each assignment must be determined, have been developed. Sales and advertising planning involves similar trade-offs between revenue generation, resource constraints and costs. In sales-force planning contexts, for example, the sales-force serves as a set of resources, where each salesperson has a limited amount of time and/or effort that they can allocate to customers. It is often the case that the greater the amount of effort a salesperson allocates to a given customer, the greater the return from that customer in terms of sales (Rainwater et al., 2009). The planning phase therefore involves determining the assignment of sales-force to customers and the degree of effort a salesperson should devote to each assigned customer in order to maximise the total return from customers (or expected return, when the relationship between effort and sales is not deterministic). Sales setting may be interpreted more generally as applying to a set of available marketing instruments, where an allocation of capacity-constrained marketing instruments to customers must be determined in order to maximise profit.

This chapter developed two models that dealt with resource allocation and distribution with the aim of minimising costs. The models developed can also be used to maximise revenue by considering optimal transportation combinations and optimal assignments. The major problem that traditional transportation and assignment problems faced is the problem of degeneracy. The two models that are formulated in this chapter have tried to eliminate this problem of degeneracy.

6.2 Literature Review

Shmoys and Tardos (1993) came up with an algorithm for the generalised assignment problem that extended the assignment problem to a variant problem that solved a range of possible processing times for each machine job pair. They concluded that the cost linearly increased as the processing time decreased. The main result from their study presented a polynomial-time algorithm that, given values of C and T, the algorithm finds a schedule of cost at most C and make-span of at most $2T$, if a schedule of cost C and make-span T exist.

Caggiani et al. (2012) proposed a meta-heuristic dynamic traffic assignment algorithm that was used in conjunction with a bee colony optimisation (BCO) that is capable of solving high-level combinatorial problems with fast convergence performances. Their model allowed overcoming classical demand-flow relationships drawbacks. They concluded that reliability and effectiveness of traffic assignment models depend on other important elements such as origin-destination $O - D$ travel demand, which is the core input of traffic assignment models. Other researchers like Cascetta and Postorino (2001) and Yang et al. (2001), pointed out that the most general form of solving the estimation of $O - D$ matrix using traffic counts, is to formulate the problem as an optimisation problem.

Munapo et al. (2010) revisited the GAP and came up with an ascending hyperplane approach through network flows. The GAP model was first relaxed to form a transportation model which is easier to handle than the original model. The relaxed model was then formulated as a minimum-cost network flow problem (MCNFP) and an efficient network simplex method was applied to solve the relaxed problem. The optimal solution of the relaxed model gave a lower bound (LB) to the given GAP. The LB becomes an optimal solution to the GAP, if all resource constraints are satisfied. However, if any resource constraint is not satisfied, that violation is used to determine the new LB which is greater than the previous one, and hence the ascending hyper-plane approach. These violated resource constraints, which are in the given GAP model, are used to modify the MCNFP diagram before resolving the flow problem. This procedure is repeated until all resource constraints are satisfied in the original GAP model. The proposed method is efficient for the GAP.

Kumar and Murugesan (2012) presented a modified revised simplex method for minimising a fuzzy transportation problem in which the supplies and demands are triangular fuzzy numbers. A fuzzy transportation problem (FTP) is a transportation problem in which the transportation cost, supply and demand quantities are fuzzy quantities. The objective of the FTP is to determine the shipping schedule that minimises the total fuzzy transportation cost while satisfying fuzzy supply and demand limits. They obtained an optimal solution of FTP in which the number of constraints equalled the number of occupied cells.

Alaei et al. (2012) presented a competitive algorithm for the online stochastic GAP under the assumption that no item can take up more than a fraction of the capacity of any bin. Items arrived online, and each item had a value and a size. Upon arrival, an item can be placed in a bin or discarded. The objective was to maximise the total value of the placement. Both the value and size of an item may depend on the bin in which the item is placed. The size of an item was revealed only after it had been placed in a bin. Distribution information about the value and size of each item was available in advance. However items arrived in adversarial order (non-adaptive adversary). The authors presented an application of their results to subscription-based advertising where each advertiser, if served, required a given minimum number of impressions. The proposed algorithm initially computed an optimal solution for a linear program corresponding to a fractional expected instance. In the online stage, the algorithm tentatively assigned each item upon arrival to one of the bins at random with probabilities proportional to the fractional LP solution. This ensured that the expected total size of the items that were assigned tentatively to each bin does not exceed its capacity. However, once a bin becomes full, any item which gets tentatively assigned to that bin will have to be discarded.

According to Rainwater et al. (2009), the GAP seeks an allocation of jobs to capacitated resources at minimum total assignment cost, assuming a job cannot be split among multiple resources. They considered a generalisation of this broadly applicable problem in which each job must not only be assigned to a resource, but its resource consumption must also be determined within job-specific limits. In their profit-maximising version of the GAP, a higher degree of resource consumption increased the revenue associated with a job. Their model permits a job's revenue per unit resource consumption to decrease as a function of total resource consumption, which allowed modelling quantity discounts. The objective of their study was to determine job assignments and resource consumption levels that maximise total profit. The authors then developed a class of heuristic solution methods, and demonstrated the asymptotic optimality of this class of heuristics in a probabilistic sense. Rainwater et al. (2009) computational study demonstrated that their heuristic performed very well, particularly for large ratios of the number of jobs to the number of agents. When additional improvement strategies that they proposed in their research were considered, the heuristic was successful on instances with smaller ratios as well. They observed that the time required to obtain solutions of comparable quality was considerably less for their heuristic than for the commercial solver CPLEX. The fact that their heuristic obtained quality solutions so quickly was encouraging for further research directions. Specifically, they believe that the heuristic may be very valuable when solving more general related optimisation problems for which the GAP arises as a sub-problem that needs to be solved repeatedly.

Motivated by practical applications, various generalisations of GAP have been proposed. The multi-resource generalised assignment problem (MRGAP), in which more than one resource constraint were considered for each agent, is a natural generalisation of the GAP. The MRGAP has many practical applications, for example, in distributed computer systems and in the trucking industry (Gavish and Pirkul, 1986).

Adlakha and Arsham (1998) proposed a single unified algorithm that solves both the transportation problem and the assignment problem. The algorithm provides useful information to perform cost-sensitivity analysis to a decision maker. Similar to the simplex method and its variants, their algorithm was pivotal. The algorithm initiates the solution with a warm-start and does not require any slack/surplus variables. Unlike the Hungarian method, the algorithm can solve higher than a 2-dimensional assignment problem. Their proposed solution algorithm also facilitated incorporation of side constraints which are frequently encountered in real life. Their algorithm revealed the full power of LP's sensitivity analysis extended to handle an optimal degenerate solution. In contrast to other methods, the method proposed by Adlakha and Arsham (1998) provided ranges for which the current solution remains optimal, for simultaneous dependent or independent changes of the cost coefficients from their normal values. The computational results from their algorithm demonstrated that the algorithm is more efficient than the simplex method in terms of the number of iterations and size of the tableaux.

In this chapter we came up with a unified approach to solve transportation and assignment problems. A mathematical support for the development of the unified approach is outlined step by step. A transportation branch and bound algorithm for solving the generalised assignment problem is also outlined in this chapter. Numerical examples have also been presented that explain and demonstrate the applicability of the two techniques that are developed in this chapter.

6.3 A Unified Approach to Solve Transportation and Assignment Problems

In this section a unified approach to solve both the transportation and assignment models, which is independent of degeneracy, is presented. The proposed unified approach is similar to the Hungarian method of assignment, where solutions are not subject to any test of optimality, but it has the distinct advantage of dealing with degeneracy. The new method is a unified pivotal solution algorithm designed for both transportation and assignment problems. The algorithm is free of pivotal degeneracy which may cause cycling and does not require any extra variables such as slack, surplus or artificial variables that are used in dual and primal simplex methods. The algorithm allows higher order assignment problems and side constraints. The proposed algorithm has the further advantages of being computationally practical, being easy to understand and providing useful information for decision makers.

6.3.1 Mathematical support for development of the Unified Approach

A mathematical model for a balanced transportation is given by:

$$\left.\begin{aligned}
&\text{Minimise } Z_0 = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij},\\
&\text{subject to,}\\
&\sum_{j=1}^{n} X_{ij} = a_i \text{ for } i = 1,2,...,m,\\
&\sum_{i=1}^{m} X_{ij} = b_i \text{ for } j = 1,2,...,n,\\
&\sum_{i=1}^{n} a_i = \sum_{j=1}^{m} b_j = K,\\
&x_{ij} \geq 0 \;\forall\; i,j.
\end{aligned}\right\} \tag{6.1}$$

Note that K is a known constant integer value. A feasible solution to an assignment model has many degenerate basic variables in the context of a LP and transportation model. The m sources and n destinations assignment model is degenerate by order n in an LP context and by order $n-1$ in the context of a transportation model. This means that there are n or $(n-1)$ degenerate basic variables in the context of the LP and transportation model, respectively. These degenerate basic variables do cause wasted iterations when applying the simplex or the transportation iterations. In this section a special algorithm has been proposed that tackles these problems and converge to a solution faster and more accurately. Without any loss of generality, it has been assumed that the transportation model satisfies equation (6.2)

$$\sum_{i=1}^{q} a_i = \sum_{i=1}^{q}\sum_{j=1}^{p} X_{ij} = \sum_{i=1}^{p} b_j \tag{6.2}$$

A balanced transportation problem can be viewed as a balanced assignment problem by duplicating the row i for a_i number of rows, where $i = 1, 2, 3, ..., \ q$ and after that duplicating also the column j for b_j number of columns, $j = 1, 2, 3, ..., \ p$. Thus, the problem will convert to a balanced assignment problem of dimension K. Let us call this equivalent assignment matrix as an enlarged matrix. With this background we now discuss a few concepts in the context of a transportation cost matrix.

Concept of a line in a transportation cost matrix

In a transportation cost matrix, a horizontal line in row i represents a_i number of lines and a vertical line in column j represents b_j number of lines. A zero in cell (i, j) represents a sub-matrix of dimension (a_i, b_j) with a_i x b_j number of zero elements. These zero elements can be covered either by a_i number of horizontal lines or by b_j number of vertical lines.

Counting zeros corresponding to more than one zero element in row i

Suppose row i has p number of zero elements in columns $c_1, c_2, ..., c_p$. Then the number of zeros in an equivalent enlarged assignment matrix is equal to $\sum_{i=1}^{p} b_i$. Similarly, if column j has q number of zero elements in rows $r_1, r_2, ..., r_q$, then the number of zero elements in this column is equivalent to $\sum_{i=1}^{q} a_i$.

Covering zero elements with minimum number of lines

For each row $i = 1, 2, ..., q$ and for each column $j = 1, 2, ..., p$, let the number of zeros calculated as discussed above be represented by: $z_1^r, z_2^r, ..., z_q^r$ and $z_1^c, z_2^c, ..., z_p^c$ respectively. Thus z_i^r and z_j^c are integer values greater or equal to zero for all i and j. The procedure of covering all zero elements in the transportation cost matrix is as follows:

Find maximum $\{z_1^r, z_2^r, ..., z_q^r$ and $z_1^c, z_2^c, ..., z_p^c\}$.

(i) If the maximum corresponds to a row, then draw through the zero elements, a horizontal line.

(ii) If the maximum corresponds to a column, then draw a vertical line through the zero elements.

(iii) If there is a tie, resolve it arbitrarily. Delete the row or the column corresponding to the vertical or horizontal line and recalculate the number of zeros in the remaining rows and columns, and repeat the above steps until all zero elements are covered. Here we are not making any allocation, hence the values of a_i and b_j remain the same in un-deleted rows and columns for counting the number of zero elements in each row and column.

6.3.2 Steps of the algorithm

The unified method is similar to the Hungarian method for assignment problem, the notable difference being its ability to handle the problem of degeneracy. Without any loss of generality, all transportation and assignment models can be assumed balanced, and all transportation models can, in principle, be seen as an assignment model. The algorithm consists of four steps, which are as follows:

Step 1

Reduce the given transportation matrix in such a way that each row and each column has at least one zero element. This is achieved in a fashion to the usual Hungarian method of assignment.

Step 2

Cover the zero elements with a minimum number of lines. If the number of lines are equal to K, then an optimal solution is available. Go to Step 4. If the minimum number of lines is less than K, then proceed to Step 3.

Step 3

In the reduced cost matrix, find the smallest non-zero element that is uncovered by a line and represent this value by $h > 0$. Add this value to every intersection of lines, and subtract h from all elements that are not covered by a line. Other elements that are covered by a line remain unchanged. Return to Step 2.

Step 4

Since the total number of lines is equal to K, it is possible to establish a feasible allocation restricted to zero elements only. By arguments similar to the Hungarian method, that solution is optimal.

Justification

In this method, we are viewing the transportation matrix as a version of a large assignment matrix where several rows and columns are identical. Since the relative cost matrix of assignment does not change by performing Step 3, we are generating an equivalent cost matrix with more number of independent zeros. Hence convergence of this process is guaranteed in a finite number of steps. Solution is optimal because allocation is restricted to zero elements only. This unified approach, which is a modification of the Hungarian method, is applicable to both the assignment and transportation problems. Furthermore, the process does not depend on the number of allocated cells which in the transportation method must be equal to $(m + n - 1)$ in independent cells. Thus, the proposed unified method is efficient to solve all transportation and assignment models.

6.3.3 Analysis and results

Consider the transportation problem in Table 6.1 whose objective is to minimise costs. The objective is to find the minimum cost solution for the problem. Note that the problem, if solved by the transportation method, results in degeneracy. Thus, the test of optimality will require one more independent cell to be basic with $'0'$ allocation. However, we illustrate that the proposed unified approach has no such requirement.

Table 6.1 illustrates the transportation problem.

Table 6.1: Illustrative Transportation Problem

Source		**Costs**		**Total Supply**
	1	2	3	
A	3	6	7	60
B	8	5	7	30
C	4	9	11	30
Total Demand	35	55	30	120

Applying the row reduction, we get the 3×3 matrix.

$$\begin{pmatrix} 0 & 3 & 4 \\ 3 & 0 & 2 \\ 0 & 5 & 7 \end{pmatrix}$$

After column reduction, we get the matrix.

$$\begin{pmatrix} 0 & 3 & 2 \\ 3 & 0 & 0 \\ 0 & 5 & 5 \end{pmatrix}$$

Table 6.2 gives the results of using the concept of covering the zero elements.

Table 6.2: Determining number of zeros

Row or Column where the zero appears	**Row or Column where the zeros are found**	**Number of zeros in equivalent enlarged matrix**
R_1	C_1	35
R_2	C_2, C_3	$55 + 30 = 85$
R_3	C_1	35
C_1	R_1, R_2	$60 + 30 = 90$
C_2	R_2	30
C_3	R_2	30

To cover all the zeros in Table 6.2, the maximum number of lines are found in C_1 (90) and R_2 (85). We then draw draw a horizontal line in row 2 and a vertical line in column 1. The total number of lines used is $35 + 30 = 65 < 120$, so we go to Step 3. The minimum un-lined element is 2. Application of Step 3 results in the following 3×3 matrix:

$$\begin{pmatrix} 0 & 1 & 0 \\ 5 & 0 & 0 \\ 0 & 3 & 3 \end{pmatrix}$$

Table 6.3 gives the results of using the concept of covering the zero elements.

Table 6.3: Results of covering zeros

Row or Column where the zero appears	**Row or Column where the zeros are found**	**Number of zeros in equivalent enlarged matrix**
R_1	C_1, C_3	$35 + 30 = 65$
R_2	C_2, C_3	$55 + 30 = 85$
R_3	C_1	35
C_1	R_1, R_3	$60 + 30 = 90$
C_2	R_2	30
C_3	R_1, R_2	$60 + 30 = 90$

To cover all the zeros using Table 6.3, the maximum number of lines are found in C_1, C_3 and R_2. We then draw a horizontal line in row 2 and vertical lines in column 1 and column 3. The total number of lines used is $35+30+30 = 95 < 120$, and we go to Step 3. The minimum unlined element is 1. Application of Step 3 results in the following matrix:

$$\begin{pmatrix} 0 & 0 & 0 \\ 6 & 0 & 0 \\ 0 & 2 & 3 \end{pmatrix}$$

Table 6.4 gives the results of the concept of covering the zero element.

Table 6.4: Final table for covering zeros

Row or Column where the zero appears	**Row or Column where the zeros are found**	**Number of zeros in equivalent enlarged matrix**
R_1	C_1, C_2, C_3	$35 + 55 + 30 = 120$
R_2	C_2	55
R_3	C_1	35
C_1	R_1, R_3	$60 + 30 = 90$
C_2	R_1, R_2	$60 + 30 = 90$
C_3	R_1	60

To cover all the zeros using Table 6.4, the maximum number of lines are found in R_1, C_1 and C_2. We then draw a horizontal line in row 1, vertical lines in column 1 and column 2. The total number of lines used is $120 + 90 + 90 > K$, hence an optimal solution can now be found. We therefore, stop and determine the feasible solution confined to zero elements, which is as displayed in Table 6.5. The total transportation cost is 645.

Table 6.5: Results of the final transportation problem

Source	**Costs in $**			**Total Supply**
	1	2	3	
A	5	25	30	60
B		30		30
C	30			30
Total Demand	35	55	30	120

6.3.4 Concluding remarks

The proposed approach was able to deal with a transportation model in which the first p_1 rows (where $p_1 \leq p$) have equal supply r, and similarly q_1 (where $q_1 \leq q$) have the same equal demand r. It could also be seen that the proposed approach was able to deal with the modification that was made to the transportation problem, but conventional transportation approaches will face difficulties due to degeneracy. It is desirable to develop a software for the proposed approach and test the proposed approach on larger problems to gauge its real computational efficiency.

6.4 A Transportation Branch and Bound Algorithm for Solving the Generalised Assignment Problem

6.4.1 Introduction

This section presents a transportation branch and bound algorithm for solving the GAP. This is a branch and bound technique in which the sub-problems are solved by the available efficient transportation techniques rather than the usual simplex-based approaches. The GAP is the problem of assigning n jobs to m agents such that the total cost is minimal and that each job is assigned to exactly one agent and the agent's capacity is also satisfied. GAP is NP-hard (Nauss, 2004) and has had many approaches proposed in the past 60 years. The GAP model is the general case of the assignment problem in which both jobs and agents have an equal size and the cost associated with each job-agent combination may have different values. GAP has many applications in real life. These include vehicle routing (Toth and Vigo, 2001), resource allocation (Winston and Venkataramanan, 2003), supply chain (Yagiura, 2004, 2006), machine scheduling and facility location (Munapo et al., 2010) among others.

6.4.2 Generalised assignment problem

A mathematical formulation of the GAP may be represented as shown in equation (6.3).

$$\left.\begin{aligned}
&\text{Minimise } Z_{GAP} = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij},\\
&\text{subject to,}\\
&\sum_{j=1}^{n} r_{ij}x_{ij} \leq b_i \text{ for } i = 1,2,...,m,\\
&\sum_{i=1}^{m} x_{ij} = 1 \text{ for } j = 1,2,...,n,\\
&x_{ij} \geq 0 \; \forall \; i,j.
\end{aligned}\right\} \tag{6.3}$$

where $i = 1,2,...m$ is a set of agents, $j = 1,2,...,n$ is a set of jobs, c_{ij} is the cost of assigning agent i to job j, r_{ij} is the resource needed by agent i to do job j, and b_i is the resource available to agent i.

6.4.3 Relaxing the GAP

The GAP can be relaxed to become an ordinary transportation problem. The GAP constraints representing resource restrictions are given in equation (6.4).

$$\sum_{j=1}^{n} r_{ij}x_{ij} \leq b_i \; \forall \; i. \tag{6.4}$$

The GAP model can be relaxed by replacing these constraints with other forms of inequalities given in equation (6.5).

$$\sum_{j=1}^{n} x_{ij} \leq \gamma_i \; \forall \; i. \tag{6.5}$$

where γ_i is obtained by solving the knapsack problem in equation (6.7). Thus the model becomes a transportation model as presented in equation (6.6).

$$\left.\begin{aligned}
&\text{Minimise } Z_{GAP} = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij},\\
&\text{subject to,}\\
&\sum_{j=1}^{n} x_{ij} \leq \gamma_i \text{ for } i = 1,2,...,m,\\
&\sum_{i=1}^{m} x_{ij} = 1 \text{ for } j = 1,2,...,n,\\
&x_{ij} \geq 0\ \forall\ i,j.
\end{aligned}\right\} \tag{6.6}$$

The values of γ_i are obtained by solving the following knapsack problem:

$$\left.\begin{aligned}
&\gamma_i = \text{Maximise } \sum_{j=1}^{n} x_{ij},\\
&\text{subject to,}\\
&\sum_{j=1}^{n} r_{ij}x_{ij} \leq b_i\\
&x_{ij} = 0\ or 1
\end{aligned}\right\} \tag{6.7}$$

Solving the knapsack problem

The optimal solution to the knapsack solution can be obtained by arranging the resource coefficients in row i in ascending order, i.e., $r'_{i1}, r'_{i2}, ..., r'_{in}$ where $r'_{i1} \leq r'_{i2} \leq ... \leq r_{i\gamma_i} ... \leq r'_{in}$ are the arranged coefficients. The knapsack objective value γ_i, is the largest integral value such that $b_i \geq r'_{i1} + r'_{i2} + ... + r'_{i\gamma_i}$, where $1 \leq \gamma_i \leq n$. The integral value γ_i is now the supply in the transportation model.

6.4.4 The transportation model

The optimal solution to the transportation model will act as a lower bound to the GAP and is usually infeasible to the original GAP model. The relaxed problem is shown in Table 6.6.

Table 6.6: Transportation problem

					Supply
	...	...	...		γ_1
			...	...	γ_2
	...	...	...	...	...
	...	...	...	...	γ_m
Demand	1	1	...	1	

This transportation problem is not a balanced model. In most cases $\sum_{i=1}^{m} \gamma_i \neq n$. If $\sum_{i=1}^{m} \gamma_i < n$, then equation (6.3) becomes infeasible, that is, at least one of the constraints $\sum_{i=1}^{m} = 1$ will be violated. If $\sum_{i=1}^{m} \gamma_i = n$, then the relaxed model can be solved directly without balancing. The solution to the relaxation model is optimal if it satisfies equation (6.3).

If $\sum_{i=1}^{m} \gamma_i > n$, then the relaxed model requires balancing before applying transportation techniques. To balance the transportation problem, a dummy column is added when we have inequality of the form $(>)$. When the transportation is balanced, then the optimal solution can be found by using network codes for transportation models. These are efficient and recommended, and the subproblems are not solved from scratch. The current solutions are used as starting solutions in the next iterations. Lagrangian or linear programming (LP) relaxations are not readily useful to this procedure. With this approach, it is only possible to branch if the relaxation gives an integer optimal solution, and this is not possible with LP or Lagrangian relaxations.

Branch and bound approach

A branch and bound method can be used to ascend from the lower bound to an optimal solution of the GAP. The lower bound obtained by solving the relaxed model is usually infeasible to equation (6.3). A row i that is not feasible can be selected, a clique inequality generated and used to create branches. Suppose from row i, the following variables are basic and they make up an infeasible solution: $x_{if_1}, x_{if_2}, ..., x_{if_l}$, where x_{if_j} is a basic variable. We also have $r_{if_1} + r_{if_2} + ... + r_{if_l} > b_i$, where r_{if_j} is its corresponding resource coefficient with $j = 1, 2, ..., l$. From the last inequality, we deduce that some of these basic variables are not supposed to be basic. One or more of these basic variables may not be the required basic feasible solutions and the exact number is only known for the specific given problem. Branching does not necessarily mean that the transportation sub-problem has to be resolved from scratch. The sub-problem is solved by improving the current solution. The previous solution is used as a starting solution in the next iteration.

The order of branching is very important as it can affect the size of the search tree. Strategies are required to determine a branching order that results in the smallest search tree. In this thesis it is recommended that branching starts with those rows that have the least number of choices. In other words, the most restricted rows are used in creating branches (Kumar et al., 2007). Thus the branching starts with the most restricted row, which in this thesis is defined as the row where the least number of branches can be generated.

6.4.5 Transportation branch and bound algorithm for GAP

The transportation branch and bound algorithm for the GAP consists of the following steps;

Step 1

Relax GAP to obtain a lower bound.

Step 2

Select the most restricted row to come up with branching variables.

Step 3

Branch using the selected variables. Return to Step 2 until the best transportation solution is feasible.

Best solution: A solution is said to be the best solution if it is the smallest optimal solution available.

Optimality

Suppose that the terminal nodes are given as $Z_1^T, Z_2^T, ..., Z_n^T$. The upper bound is selected from the node giving the best solution so far. Then

$$Z_{GAP} = \min[Z_1^T, Z_2^T, ..., Z_n^T] \tag{6.8}$$

Thus, Z_{GAP} is optimal. In the branching tree, a node is said to be a terminal node if (i) an optimal solution to the transportation model is feasible to the original GAP model, (ii) transportation model does not have a feasible optimal solution or (iii) an optimal solution to the transportation model is bigger than a given upper bound.

NB: Generation of clique inequalities and using them as cuts is not a new idea. Clique constraints used in this research are in fact a simple type of knapsack constraints generated from single constraints of the original problem. Knapsack constraint generators are very common in modern MIP solvers. What is new is the fashion of using these inequalities to form branches and solving the sub-problems generated as transportation problems. This is effective for GAP models. Branching is done on a node with the smallest objective value.

6.4.6 Analysis and results

Table 6.7 shows a transportation model that was used to illustrate the GAP model.

Table 6.7: Transportation model for numerical illustration

	Costs					**Resources**
	$(28, 24)$	$(76, 38)$	(M, L)	$(52, 22)$	$(28, 36)$	56
	$(98, 12)$	(M, L)	$(40, 22)$	$(92, 30)$	$(98, 36)$	56
	(M, L)	$(90, 20)$	$(32, 28)$	$(20, 44)$	(M, L)	56
Jobs	1	1	1	1	1	

The letter M is a large cost to discourage assignment and L shows that an assignment is not possible in that cell.

The transportation model is given in equation (6.9).

$$\left.\begin{array}{l}
\text{Minimise } Z_{GAP} = 28x_{11} + 76x_{12} + 52x_{14} + 28x_{15} + ... + 32x_{33} + 20x_{34}. \\
\text{ subject to,} \\
24x_{11} + 38_{12} + 22x_{14} + 36x_{15} \leq 56 \\
12x_{21} + 22x_{23} + 30x_{24} + 36x_{25} \leq 56 \\
20x_{32} + 28x_{33} + 44x_{34} \leq 56 \\
x_{11} + x_{21} = 1 \\
x_{12} + x_{32} = 1 \\
x_{23} + x_{33} = 1 \\
x_{14} + x_{24} + x_{34} = 1 \\
x_{15} + x_{25} = 1 \\
x_{ij} = 0 \; or \; 1, \text{for all } i, j.
\end{array}\right\} \quad (6.9)$$

Arranging the resource coefficients of constraints (first three constraints of equation (6.9)) in ascending order, we have $\{22, 24, 36, 38\}$, $\{12, 22, 30, 36\}$ and $\{20, 28, 44\}$. The γ_i values are easily calculated from equation (6.9) and the arranged resource coefficients generate equation (6.10).

$$\left.\begin{aligned} 22 + 24 = 46 \leq 56 \ \Rightarrow \gamma_1 = 2 \\ 12 + 22 = 34 \leq 56 \ \Rightarrow \gamma_2 = 2 \\ 20 + 28 = 48 \leq 56 \ \Rightarrow \gamma_3 = 2 \end{aligned}\right\} \tag{6.10}$$

The transportation model becomes as shown in Table 6.8.

Table 6.8: Transportation model for numerical illustration

						Supply
	28	76	L	52	28	2
	98	L	40	92	98	2
	L	90	32	20	L	2
Demand	1	1	1	1	1	

A dummy column is introduced to balance the transportation problem as shown in Table 6.9.

Table 6.9: Balancing the transportation model (by adding a dummy column)

							Supply
	28	76	L	52	28	0	2
	98	L	40	92	98	0	2
	L	90	32	20	L	0	2
Demand	1	1	1	1	1	1	

Any efficient transportation technique can be used to solve the model, and an optimal solution to the relaxed model is obtained as presented in Table 6.10.

The solution in Table 6.10 is a second order degenerate solution. The optimality solution can be easily verified by using cells $(1,6)$ and $(3,3)$ as basic with zero allocation. Using the resource constraints, one can easily verify that row one is infeasible, since $24+36=60>56$, that is to say $x_{11}+x_{15}\leq 1$. This implies that, either $x_{11}=0$ or $x_{15}=0$.

Table 6.10: Optimal solution to the relaxed model (lower bound)

							Supply
	28[1]	76	L	52	28[1]	0	2
	98	L	40[1]	92	98	0[1]	2
	L	90[1]	32	20[1]	L	0	2
Demand	1	1	1	1	1	1	

From Table 6.10, $Z_{relaxed}=206$. Similarly the third row is also infeasible because $90+20=110>56$, that is $x_{32}+x_{34}\leq 1$. This implies that either $x_{32}=0$ or $x_{34}=0$. We select the branches from row 3 as it is more restricted compared to row 1. This results in Figure 6.1.

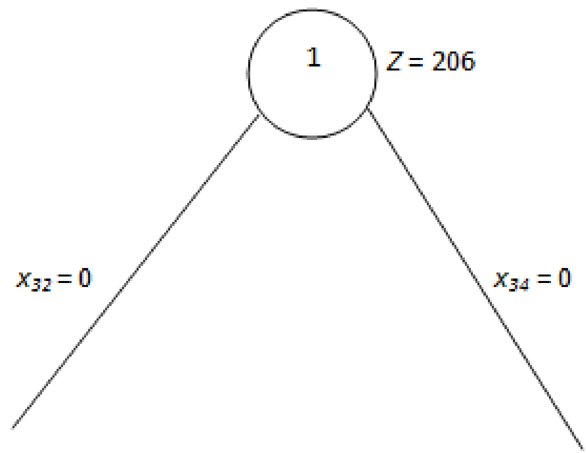

Figure 6.1: Initial branching with respect to cells (3,2) and (3,4)

Now let us consider the case when $x_{32} = 0$. The transportation problem shown in Table 6.10 is modified by replacing the assignment cost of 90 in the cell $(3, 2)$ by L. This is given in Table 6.11. Once again, for the above solution, the resource constraint 3 is not satisfied because $28 + 44 = 72 > 56$. Hence, either $x_{33} = 0$ or $x_{34} = 0$. This will lead to nodes 4 and 5, respectively.

Table 6.11: Node corresponding to the restriction of no assignment in cell (3,2)

							Supply
	28[1]	76[1]	L	52	28	0	2
	98	L	40	92	98[1]	0[1]	2
	L	L	32[1]	20[1]	L	0	2
Demand	1	1	1	1	1	1	

$Z_{relaxed} = 254$.

Similarly, at node 3 we deal with the restriction that allocation in the cell (3,4) is restricted to zero, in other words, we modify Table 6.10 and replace the cost element in the cell (3,4) by L. This is shown in Tables 6.12, 6.13, 6.14, 6.15, 6.16 and 6.17.

Table 6.12: Node corresponding to the restriction that allocation in cell (3, 4) is zero

							Supply
	28[1]	76	L	52	28[1]	0	2
	98	L	40[1]	92	98	0[1]	2
	L	90[1]	32	L	L	0	2
Demand	1	1	1	1	1	1	

$Z_{relaxed} = 270.$

Table 6.13: Node 4

							Supply
	28[1]	76[1]	L	52	28	0	2
	98	L	40[1]	92	98[1]	0	2
	L	L	L	20[1]	L	0[1]	2
Demand	1	1	1	1	1	1	

$Z_{relaxed} = 262$

Table 6.14: Node 5

							Supply
	28[1]	76[1]	L	52	28	0	2
	98	L	40	92[1]	98[1]	0	2
	L	L	32[1]	L	L	0[1]	2
Demand	1	1	1	1	1	1	

$Z_{relaxed} = 326$

Table 6.15: Node 6

							Supply
	28[1]	76	L	52[1]	L	0	2
	98	L	40	92	98[1]	0[1]	2
	L	90[1]	32[1]	L	L	0	2
Demand	1	1	1	1	1	1	

$Z_{relaxed} = 300$

Table 6.16: Node 7

							Supply
	L	76	L	52[1]	28[1]	0	2
	98[1]	L	40	92	98	0[1]	2
	L	90[1]	32	L	L	0	2
Demand	1	1	1	1	1	1	

$Z_{relaxed} = 300$

Table 6.17: Node 9

							Supply
	28	76[1]	L	52	28[1]	0	2
	98[1]	L	40[1]	92	L	0	2
	L	L	L	20[1]	L	0[1]	2
Demand	1	1	1	1	1	1	

$Z_{relaxed} = 262$

From the solution obtained in Table 6.12, it is noted that the resource constraint 1 is not satisfied, since for the allocation in cells $(1, 1)$ and $(1, 5)$ resource requirement is $24 + 36 = 60 > 56$. Hence branching from node 3, we will get either $x_{11} = 0$ or $x_{15} = 0$. These restrictions lead to nodes 10 and 11 in Figure 6.2. We have given the transportation cost tables under various restrictions as shown in the tree diagram in Figure 6.2. These results have been summarised in the tree diagram of Figure 6.2, where the following interpretations have been used:

(i) A terminal node is said to be feasible if the optimal solution to the transportation sub-problem is feasible to the original GAP problem.

(ii) A terminal node is said to be infeasible if the optimal solution to the transportation is infeasible to the GAP model.

(iii) DNE means the transportation sub-problem does not have a feasible optimal solution.

(iv) The numbers in the circles denote the order of solution.

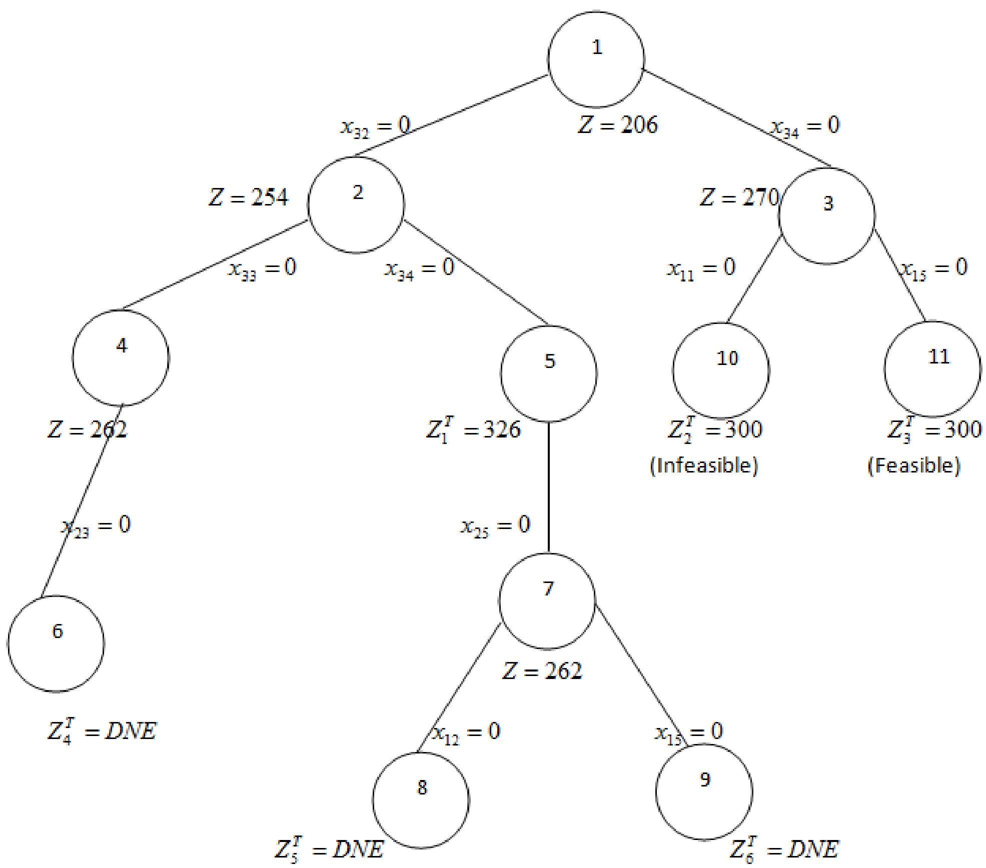

Figure 6.2: Full search tree for the given numerical illustration

From the search tree given in Figure 6.2 the optimal solution to the GAP problem is given as shown in equation (6.11) and (6.12).

$$Z_{GAP} = \min[Z_1^T, Z_2^T, Z_3^T, Z_4^T, Z_5^T, Z_6^T] = 300 \tag{6.11}$$

The minimum value of 300 emanates from Z_3^T which corresponds to Table 6.15. The corresponding decision variables from this table are as follows;

$$\left.\begin{array}{l} x_{11} = x_{14} = x_{25} = x_{32} = x_{33} = 1 \\ \text{and,} \\ x_{12} = x_{15} = x_{21} = x_{23} = x_{24} = x_{34} = 0 \end{array}\right\} \tag{6.12}$$

Node 1 is an optimal solution to the relaxed model, and is given in Table 6.10. Thus, a total of 10 nodes (starting from node 2) are required to verify the optimal solution value.

6.4.7 Concluding remarks

The proposed approach has the advantage that the individual, γ_i values can be found independently allowing the much needed use of parallel processors. The sub-problems resulting from the search trees are transportation models and can be solved efficiently by the available network approaches. The sub-problems that result from the usual branch and bound related approaches are NP-hard integer models which are very difficult to solve. The only nuisance to this approach is that like the simplex-based approaches, it is also not spared by degeneracy. It may be desirable to use the approach discussed by the authors in an earlier publication (Munapo et al., 2012). In the search tree diagram given in Figure 6.2, it may be noted that there is no change in the objective value from node 4 to node 7. The degeneracy drawback can be alleviated by noting all alternate optimal solutions at every node and then branching in such a way that the objective value does not remain static. Attempts will be made in future to use cuts in branching, compare its efficiency with the available approaches and explore for better strategies that can significantly improve the selection of branching variables.

6.5 Summary of the Chapter

This chapter studied at two new approaches that can be used to solve the transportation and assignment problems. The major advantage of these two techniques is that they take the problem of degeneracy into consideration. The algorithm for the unified approach to solve the transportation and assignment problems, fully exploits the sub-problem's structure and has very favourable re-optimisation capabilities. Both these properties are necessary for achieving optimality. This unified approach, which is a modification of the Hungarian method is applicable to both the assignment and transportation problems. Furthermore, the process does not depend on the number of allocated cells which in transportation method must be equal to $(m + n - 1)$ in independent cells. Thus, the proposed unified method is efficient to solve all transportation and assignment models. This approach is free of pivotal degeneracy which may cause cycling, and does not require any extra variables such as slack, surplus or artificial variables, that are used in dual and primal simplex methods.

The generalised assignment problem (GAP), deals with assigning a set of n items to a set of m knapsacks, where each item must be assigned to exactly one knapsack and there are constraints on the availability of resources for item assignment. The GAP is a classical combinatorial optimisation problem that models a variety of real world applications including flexible manufacturing systems, facility location and vehicle routing problems. The GAP is known to be NP-hard, since the partition problem of a given set of positive integers into two equal sized subsets can be reduced to GAP with m=2.

In the method proposed in this thesis, the GAP was relaxed to become an ordinary transportation problem by replacing the ordinary constraints with inequalities obtained by solving the knapsack problem. In the new method the current solutions to the transportation problem are used as starting solutions in the next iterations. With this approach, it is only possible to branch if the relaxation gives an integer optimal solution, and this is not possible with LP or Lagrangian relaxations. The order of branching is very important as it can affect the size of the search tree. In this method, we recommended that branching starts with those rows that have the least number of choices, which we defined as the row where the least number of branches can be generated from. What is new is the fashion of using these inequalities to form branches, and solving the sub-problems generated as transportation problems. This is effective for GAP models.

Chapter 7

Linear Programming based Model for solving some NP-hard Problems

Good judgement comes from experience; experience comes from bad judgement

F. Brooks

7.1 Introduction

The chapter presents two linear programming (LP) based techniques for solving some NP-hard problems. Section 7.2 reviews some literature that is related to the work that is covered in this chapter, that is, large-scale LP problems and the mixed integer problems. Section 7.3 considered a large-scale LP model with non-negative coefficients and develops a new strategy that is an iterative hybrid process. The approach uses the conventional simplex iterations for search along the extreme points of the convex region. The procedure generates an interior point using these extreme points and moves from the interior point in the direction of the normal to the given objective function hyper-plane. The approach is a modification of the Munapo-Kumar strategy (Munapo and Kumar, 2013) where they determined two unknown distances to be travelled from a current position on the boundary of a constraint in two known directions in the convex polyhedron space of the given LP model.

Section 7.4 presents a method for the general large-scale LP model which has extended the method discussed in Section 7.3 by considering a general large-scale LP without restrictions on non-negative coefficients.

In Section 7.5 we have developed a heuristic for solving a mixed integer programming (MIP) problem using the characteristic equation (CE) approach. While most LP problems can be solved in polynomial time, pure integer programming (PIP) and MIP problems are NP-hard, hence there are no known polynomial time algorithms to solve them (Bertacco et al., 2007). The method discussed in this section makes use of the CE which is obtained from the final tableaux of the relaxation problem. The CE proved to be a very useful and efficient method for solving MIP problems. The major advantage of using the CE is that convergence to the optimal solution is guaranteed, and the CE can be used to obtain ordered optimal solutions, even for NP-hard problems.

7.2 Literature Review

Bixby et al. (1991) formulated a procedure that combined the simplex method, an interior point method and a hybrid interior point/simplex approach. The procedure solved a 12 753 313-variable LP relaxation of a set partitioning problem arising from an airline crew scheduling application. Their method discussed the characteristics of the set partitioning problems, the sifting procedure, a new pricing rule and the experience with the sifting procedure using the simplex method and an interior point method. Their method illustrated the power of an interior point/simplex method combination for solving very large-scale linear programs.

Munapo and Kumar (2013) developed a method for solving large-scale LP problems with non-negative coefficients. Their method was an iterative one, in which the search-point move from one boundary of the convex region to an improved point on the boundary of the LP convex region. Their method was suited for large-scale LP problems. The advantage of the procedure was that the n-variable problem was reduced to a two-variable LP problem, which was easier to solve.

According to Lesaja (2009), the introduction and development of interior-point methods have had a profound impact on optimisation theory as well as practice, influencing the field of operations research and related areas. Development of these methods has quickly led to the design of new and efficient optimisation codes particularly for LP problems. The author further highlighted that there has been an increasing need to introduce theory and methods of this new area in optimisation into the appropriate undergraduate and first year graduate courses such as introductory operations research and/or linear programming courses, industrial engineering courses and mathematical modelling courses.

Zhang (1996) described an implementation of a primal-dual infeasible-interior-point algorithm for large-scale LP under the MATLAB environment, and came up with a software called Linear-programming Interior-Point SOLvers (LIPSOL). Under the MATLAB environment, LIPSOL inherited a high degree of simplicity and versatility in comparison to its counterparts in Fortran or C language. Their extensive computational results demonstrated that LIPSOL also attains an impressive performance comparable with that of efficient Fortran or C codes in solving large-scale problems. The author further discussed in detail a technique for overcoming numerical instability in Cholesky factorisation at the end-stage of iterations in interior-point algorithms.

Sherali and Adams (1994) came up with a tight equivalent representation for mixed integer zero-one programming problems. For the linear case, they proposed a technique which first converts the problem into a non-linear, polynomial mixed integer zero-one problem by multiplying the constraints with some suitable d-degree polynomial factors involving the n binary variables, for any given $d \in (0, \ldots, n)$, and subsequently linearises the resulting problem through appropriate variable transformations. As d varies from zero to n, they obtained a hierarchy of relaxations spanning from the ordinary LP relaxation to the convex hull of feasible solutions. The facets of the convex hull of feasible solutions in terms of the original problem variables were available through a standard projection operation. They also suggested an alternate scheme for applying this technique which gives a similar hierarchy of relaxations, but involving fewer "complicating" constraints. Techniques for tightening intermediate level relaxations, and insights and interpretations within a disjunctive programming framework, are also presented. The methodology readily extends to multilinear mixed integer zero-one polynomial programming problems in which the continuous variables appear linearly in the problem.

According to Codato and Fischetti (2006) (MIPs) involving logical implications modelled through big-M coefficients are notoriously among the hardest to solve. In their paper, they proposed and analysed computationally, an automatic problem reformulation of quite general applicability, aimed at removing the model dependency on the big-M coefficients. Their solution scheme defined a master

ILP with no continuous variables, which contains combinatorial information on the feasible integer variable combinations that can be "distilled" from the original MIP model. The master solutions were sent to a slave linear program, which validates them and possibly returns combinatorial inequalities to be added to the current master ILP. The inequalities were associated with minimal (or irreducible) infeasible subsystems of a certain linear system, and were separated efficiently in case the master solution is integer. The overall solution mechanism closely resembled the Benders' Cuts, but the cuts produced by their method were purely combinatorial and did not depend on the big-M values used in the MIP formulation. This produces an LP relaxation of the master problem which can be considerably tighter than the one associated with original MIP formulation. Computational results on two specific classes of hard-to-solve MIPs indicated that the new method produced a reformulation which can be solved with some orders of magnitude faster than the original MIP model.

7.3 Solving a Large-scale LP Model with Non-negative Coefficients: A Hybrid Search over the Extreme Points and the Normal Direction to the Given Objective Function

This section develops a hybrid search process for a large-scale LP problem. The hybrid approach uses the normal simplex iterations for search over the extreme points of the convex region, generates an interior point using these extreme points, and moves from the interior point in a known direction, which is normal to the given objective function. This approach is suitable only for a large-scale LP model.

Consider the problem given by equation (7.1).

$$\left.\begin{array}{l} \text{Maximise } Z = CX, \\ \text{subject to}, \\ AX \leq B. \end{array}\right\} \tag{7.1}$$

where

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ \dots \\ b_m \end{bmatrix} C = \begin{bmatrix} c_1 & \dots & c_n \end{bmatrix} X = \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix}$$

$x_{ij} \geq 0 \ \forall \ i = 1, 2, ..., m, \ j = 1, 2.., n \ b_i \geq 0 \ \forall \ i$, and $c_j \geq 0 \ \forall \ j$.

7.3.1 Preliminary work, Munapo-Kumar strategy

Before we discuss the proposed hybrid approach, it would be desirable to briefly review the Munapo-Kumar strategy (Munapo and Kumar, 2013). The authors determined two unknown distances to be travelled from a current position on the boundary of a constraint in two known directions in the convex polyhedron space of the given LP in equation (7.1). One of the two unknowns was the optimal distance moved in the direction of the normal to the objective function, and the other distance was the normal to the plane containing the point at the present location. The normal direction of the given objective function was the best direction for attaining maximum increase in the value of the objective function. For a known feasible point in the space defined by the given LP in (7.1), the two directions were known. The unknowns were the optimal distances to be moved in those two known directions. The known coefficients of the given unknown n variables in the given LP gave them all the required normal directions. The optimal value of these two unknowns was determined by formulating and solving a '2' variable LP model from the given n variable LP model in (7.1), hence transforming a given n unknown LP into a two-variable LP. The optimal solution to the two-variable LP, gave them the required optimal distances to be moved in those two known directions. The two moves in two known directions, take the search point from the initial point P_i on the surface of a constraint to an improved point P_{i+1}, which was also on the surface of another constraint. The purpose of this journey from P_i to P_{i+1} was to improve the value of the objective function as much as possible. The value of the objective function at the point P_{i+1} has improved compared to its value at the point P_i. If optimality conditions were not satisfied at the point P_{i+1}, then this point takes the role of the point P_i and the search for a new improved point P_{i+2} will be carried out. Thus, the process was an iterative one.

7.3.2 The hybrid strategy

Let the initial extreme point be denoted by IP_0. We propose two options:

Option one is to move from the point IP_0 in the direction normal to the objective function. The intention is to move as far as possible within the feasible region. Let this new position be denoted by IP_1, which will either be an extreme point of the feasible region or alternatively it may be a boundary point of the convex region of the LP in equation (7.1).

Option two is to carry out a few simplex iterations in search of an optimal solution and if an optimal solution has not been identified at these extreme points, it is proposed to use these extreme points to generate an interior point in the feasible region. From the interior point we move as far as possible in the direction of the normal to the given objective function.

Whichever approach we take, after moving in the direction of the normal to the objective function, we arrive at a known point. Since location of the new point is known, one can easily find the value of the objective function at that point. Let us denote the new point by the symbol IP_1, and the corresponding value of the objective at this point be denoted by Z_{IP_1}. If IP_1 is not an optimal solution, an additional condition (2) can be added to the LP in (7.1) for the optimal solution. The condition is:

$$Z = \sum_{j=i}^{n} c_j x_j \geq Z_{IP_1} \tag{7.2}$$

A consequence of the equation (7.2) is that the feasible search region is reduced. One can easily return back to the conventional simplex iterations for determination of the optimal solution of the LP in equation (7.1) by using the LP given by equation (7.3).

$$\left.\begin{aligned} &\text{Maximise } Z = CX,\\ &\text{ subject to},\\ &AX \leq B\\ &\sum_{j=1}^{n} c_j x_j \geq Z_{IP_1} \end{aligned}\right\} \tag{7.3}$$

where A, B, C and X are as defined in equation (7.1). The process terminates at any point when simplex optimality conditions are satisfied, else one has to carry out at least three simplex iterations to obtain at least three new feasible extreme points. Using these extreme points one can find an interior feasible point IP_i for $(i = 2, 3, ...)$ of the reduced convex polyhedron given by equation (7.3). Once again from the interior point IP_i, one moves in the direction of the normal to the hyper-plane of the objective function. Thus, repeated applications of the above process will lead us to the optimal extreme point where optimality conditions are satisfied. Note that the problem (7.3) will change each time the constraint in (7.2) is replaced by a new constraint. Thus, one can accommodate the new constraint in (7.2) in the simplex tableau by using the ideas of information recycling discussed by Kumar (2005, 2006).

Without loss of generality, if we let the initial point be denoted by IP_i, and the point at the end of a move in the normal direction by IP_{i+1}. The corresponding value of the objective function at that point is denoted by Z_{IP_i}. The additional constraint in the equation (7.3) is replaced by the constraint in equation (7.4).

$$CX \geq Z_{IP_{i+1}} \text{ for } i = 0, 1, 2, ... \tag{7.4}$$

Since $Z_{IP_0} < Z_{IP_1} < ... < Z_{IP_r}$, we are dealing with sub-sets of the feasible diminished or reduced size, and the process must terminate after a finite number of such iterations of linear move in the normal direction together with the simplex iterations.
The above process is summarised as follows:

1. Either find at least three extreme points by the simplex method and develop an interior point or from the initial feasible point IP_0 move as far as possible within the feasible region in the normal direction of the given objective function at the improved point IP_1.

2. Find the coordinated of the point IP1 and the corresponding value of the objective function denoted by Z_{IP_i}.

3. If IP_1 is not the required optimal solution, find at least three new feasible extreme points of the convex polyhedral of the problem given by equation (7.3). These points are obtained by using the conventional simplex method and exploiting the fact that the active constraints at the point IP_1 are known to us.

4. If the optimal solution was not identified while developing the extreme points; an interior point is established, a feasible move in the normal direction is carried out and the feasible space is further reduced.

5. The process is repeated unless simplex optimality conditions are satisfied.

Note that the constraint in equation (7.2) increases the size of the basis from m to $m+1$, where m is the number of constraints in the given LP of equation (7.1).

Thus, the constraint in equation (7.2) in one way helps to reduce the feasible search region, but at the same time increases the number of extreme points, which may not be the extreme points of the original convex polyhedron of the given LP generated by equation (7.1). When the optimal solution is identified, the additional constraint will not be an active constraint, hence back to the extreme point defined by equation (7.1). However, the other role of extreme points is to locate an interior feasible point. Therefore, it has no significance whether the extreme points used for locating an interior point is of the original convex polyhedron (7.1) or that of the modified problem given by (7.3). Until the optimal solution has been identified, the role of the extreme point is to reduce the search region. Our detailed procedure is discussed in Section 7.3.3.

7.3.3 The hybrid process

Some observations

A few simple observations are presented here, which are useful for developing the hybrid process for solving large-scale LP models of the type shown in equation (7.1). **Observation 1: Location of a feasible interior point**
An interior feasible point can be easily generated by averaging $r \geq 3$ extreme points of the convex polyhedron of a given LP. Let these r extreme points be represented by EP_i for $(i = 1, 2, ..., r$. The interior point as a function of these r extreme points will be given by equation (7.5).

$$IP_k = \frac{1}{r} \sum_{i=1}^{r} EP_i \tag{7.5}$$

The extreme points EP_i can easily be located by r iterations of the simplex method. Let the coordinates of the i^{th} known extreme point of the convex polyhedron be given by:

$$EP_i = \begin{bmatrix} x_1^i \\ ... \\ x_n^i \end{bmatrix}$$

where x_j^i, for $i = 1, 2, ..., r$ and $j = 1, 2, ..., n$ are known values. Note that the location given by equation (7.5) will be an interior feasible point of the convex polyhedron defined by equation (7.3), since the location given by equation (7.5) is a convex combination of known r extreme points of the convex space generated by the LP shown in equation (7.3). Furthermore, the purpose of equation (7.5) is to generate a known interior feasible point where each element value can be rounded up or down to avoid dealing with fractions, provided the rounded location still remains feasible and continue to represent a feasible interior point.

Observation 2: A consequence of the rounding process

Note that rounding up to next integer point may not always lead to a feasible point, hence when the interior point is rounded up, it must be checked for feasibility. However, rounding down will always lead to a feasible point.

Proof

Let the co-ordinates of the interior point as obtained from equation (7.5) be denoted by $x_j = (\alpha_j + \beta_j)$, where α_j is an integer part and β_j is the fractional part of the value of the variable x_j, for $j = 1, 2, ...$. Since an interior point is a feasible point, all constraints at this point must be satisfied. Note that the process of rounding down means the β_j values are set equal to zero for all $j = 1, 2..., n$, and since $\beta_j \geq 0$, all constraints in equation (7.3) would still hold. However, it may not always be of advantage to round these values down. This process is discussed in the numerical illustration in Section 7.3.5.

Observation 3: Maximise the value of the objective function

Since the location of IP_k or its rounded version is a known interior point, the value of the objective function at this point is also known. The known value of the objective is denoted by Z_{IP_k}. In order to achieve maximum increase in the value of the objective function, it is proposed to move from this location in the normal direction of the given objective hyperplane. Thus the new location is given by equation (7.6). Note that C^T is the normal direction of the given objective function.

$$IP_k + \alpha C^T \geq 0 \tag{7.6}$$

where $\alpha \geq 0$. The scalar α has to be calculated as the largest value such that $Z = CX$ is maximised subject to $A(IP_k + \alpha C^T) \leq B$, and $(IP_k + \alpha C^T) \geq 0$ or equivalently as shown in equation (7.7).

$$\left.\begin{array}{l} \text{Maximise } Z = C(IP_k + \alpha C^T) \\ \text{ subject to}, \\ \alpha AC^T \leq B - AIP_k \\ \alpha C^T \geq -IP_k \end{array}\right\} \tag{7.7}$$

The problem in equation (7.7) is a one-variable LP, where the value of α is given by:

$$\alpha = \text{minimum}\left\{\frac{(B - AIP_k)}{AC^T}\right\}. \tag{7.8}$$

Once the value of α is known the new point with an improved value of the objective function is a known point. The new improved feasible location may be an extreme point or may be a point on the boundary of a constraint of the given LP in equation (7.1). It is denoted by IP_{k+1}, and can be determined from equations (7.6) and (7.8). It is expressed in equation (7.9).

$$IP_{k+1} = IP_k + \alpha C^T \tag{7.9}$$

The new LP given in equation (7.10) is developed, and the process is repeated.

$$\left.\begin{array}{l}\text{Maximise } Z = CX \\ \text{ subject to,} \\ \alpha AX \leq B \\ Z \geq Z_{IP_{k+1}} \\ X \geq 0\end{array}\right\} \quad (7.10)$$

Here $X = IP_{k+1}$.

Note that if α as obtained from (7.8) is equal to zero, it means that one has reached to the end of the feasible space in the normal direction. If the optimal solution has not been identified, one has to change the search direction, which is easily achieved by the simplex method.

Observation 4: The new point

The new point $X = IP_{k+1}$ will always be an extreme point of the convex space defined by equation (7.10). In other words, the problem defined by equation (7.10) has a feasible region.

Proof Note that the point defined by equation (7.9) will remain feasible to equation (7.10) as all constraints of equation (7.1) are satisfied at this point, and the additional constraint in equation (7.2) is also satisfied. Furthermore, it may be noted that at this point at least one of the constraints in equation (7.1) is active, and also the additional constraint in equation (7.2) is active. Thus, it is an extreme point at the intersection of at least two active constraints of the model in equation (7.10).

7.3.4 The hybrid algorithm for solving large-scale LP with non-negative coefficients

From the above discussion, one can generate the following steps for the hybrid algorithm for solving a large-scale LP of the equation (7.1) type.

Step 1

Obtain an initial feasible point IP_0 given by:

$$IP_0 = \begin{bmatrix} x_1 \\ ... \\ x_n \end{bmatrix} = \begin{bmatrix} 0 \\ ... \\ 0 \end{bmatrix}$$

and move as far as possible in the direction of the normal to the objective function. Find the new feasible point with an improved value of the objective function. In other words, find the value of α such that $\alpha AC^T \leq B$ and $\alpha \geq 0$. Alternatively develop $r(r \geq 3)$ extreme points by using the simplex method, find an interior point and move as far as possible in the direction of the normal to the given objective function. The initial feasible point IP_0 will be given by equation (7.5).

Step 2

For $i = 1$, find IP_1 and Z_{IP_1}. If optimal, go to Step 7, else go to Step 3.

Step 3

Formulate the problem:

$$\left.\begin{aligned} &\text{Maximise } Z = CX \\ &\text{ subject to,} \\ &\alpha AX \leq B \\ &CX \geq Z_{IP_1} \\ &X \geq 0 \end{aligned}\right\} \quad (7.11)$$

Set $i = 1$, and find a feasible solution to equation (7.11). If optimal, go to Step 7, else go to Step 3.1.

Step 3.1

Set $i = i + 1$, and go to Step 3.2.

Step 3.2

If $i < 3$, find the improved feasible solution using the simplex method. If optimal, go to Step 7, else return to Step 3.1. If $i = 3$, go to Step 4.

Step 4

Set $i = i+1$. Using the last three feasible extreme points, find the interior point using equation (7.5). Call it interior point IP_i.

Step 4.1

Round up all co-ordinates to integer values and check for feasibility. If the rounded up solution is feasible, find the improved point in the direction of the normal. If not feasible, go to Step 4.2.

Step 4.2

Round all values down, which have been obtained from equation (7.5). This point will always be feasible but check if rounding is of advantage. (This point has been clearly explained in Section 7.3.5. Once again using equation (7.8), find the value of α and the new location, IP_{k+1}, else go to Step 4.3.

Step 4.3

Retain the point IP_k as was determined from equation (7.5), and determine the value of α from equation (7.9) and the new location, IP_{k+1}.

Step 5

If the solution is optimal at the new point IP_{k+1}, then go to Step 7, else go to Step 6.

Step 6

Find the value of the objective function at the point IP_{k+1}, and replace the additional constraint in equation (7.3). However, before returning back in search of three new extreme points, also check the dual simplex values. If the solution is not optimal due to one or two shadow prices, it may be worth continuing with the simplex method as the solution is likely to conclude to the optimal solution in a couple of iterations. Return to Step 3.

Step 7

Conclude the search process as the optimality condition has been satisfied.

7.4 Development of the Method for a General Large-scale LP Model

Let the number of variables and the number of constraints in the LP model of equation (7.1) be denoted by n and m respectively. Without any loss of generality, let us assume that n_1 of the n variables $(n_1 < n)$ have positive coefficients in the objective function and the remaining $(n - n_1)$ coefficients in the objective function have negative values. We further assume, for ease of presentation, that the positive coefficients are associated with the first n_1 variables and the negative coefficients are associated with the remaining variables $(n_1 + 1, n_1 + 2, \ldots, n.)$. Furthermore, it is assumed that the variables $(n_1 + 1, n_1 + 2, \ldots, n_2)$ are such that the corresponding constraint columns A_j are such that one of the element $a_{ij} < 0$ for at least one i, $(i = 1, 2, \ldots, m$ and the columns A_j for the variables $(n_2 + 1, n_2 + 2, \ldots, n)$ are such that $a_{ij} \geq 0$ for all i. Thus, without any loss of generality, the LP model of equation (7.1), can be expressed as given by equation (7.12).

$$\left.\begin{array}{l}
\text{Maximise } Z = \sum_{j=1}^{n_1} c_j x_j - \sum_{j=n_1+1}^{n_2} c_j x_j - \sum_{j=n_2+1}^{n} c_j x_j \\
\text{subject to,} \\
\sum_{j=1}^{n_1} A_j x_j + \sum_{j=n_1+1}^{n_2} A_j x_j + \sum_{j=n_2+1}^{n} A_j x_j \leq B \\
x_j \geq 0, \quad \forall \, \jmath, \\
c_j \geq 0 \ \text{ for } j = 1, 2, ..., n_1, \\
c_j \leq 0 \ \text{ for } \jmath = n_1 + 1, n_1 + 2, ..., n_2, n_2 + 1, ..., n, \\
A_j \geq 0 \ \text{ for } \ j = n_2, n_2 + 1, ..., n
\end{array}\right\} \tag{7.12}$$

where

$$A = \begin{bmatrix} a_{11} & ... & a_{1n} \\ ... & ... & ... \\ a_{m1} & ... & a_{mn} \end{bmatrix} = \begin{bmatrix} A_1 & A_2 & ... & A_n \end{bmatrix}, B = \begin{bmatrix} b_1 \\ ... \\ b_m \end{bmatrix}, C = \begin{bmatrix} c_1 & ... & c_n \end{bmatrix}, X = \begin{bmatrix} x_1 \\ ... \\ x_n \end{bmatrix}$$

From equation (7.12) a sub-problem is developed as given in equation (7.13).

$$\left.\begin{aligned} &\text{Maximise } Z = \sum_{j=1}^{n_1} c_j x_j \\ &\text{subject to,} \\ &A_{n_1} X_{n_1} \leq B \\ &\sum_{j=1}^{n_1} c_j x_j \geq 0, \quad \text{for J} = 1, 2, ..., n_1, \\ &x_j \geq 0 \ \text{ for } j = 1, 2, ..., n_1, \end{aligned}\right\} \qquad (7.13)$$

where

$$A = \begin{bmatrix} a_{11} & ... & a_{1n_1} \\ ... & ... & ... \\ a_{m1} & ... & a_{mn_1} \end{bmatrix} B = \begin{bmatrix} b_1 \\ ... \\ b_m \end{bmatrix}, C = \begin{bmatrix} c_1 & ... & c_{n_1} \end{bmatrix}, X_{n_1} = \begin{bmatrix} x_1 \\ ... \\ x_{n_1} \end{bmatrix}$$

Note that in the sub-problem of equation (7.13), all objective function coefficients are non-negative. Therefore the non-negative requirement is satisfied for the model in equation (7.13). A movement in the direction of the normal to the objective function of model (7.13) will be confined to the positive quadrant in the n_1 dimensional space. Also note that when we deal with the model in equation (7.13) instead of the model in equation (7.1), we are dealing with less number of variables.

Let us use the symbol CR to denote the convex region defined by the linear constraints of equation (7.1). In the hybrid approach, one finds the interior point by averaging three successive extreme points obtained by the usual simplex iterations. These three interior points will also form a convex region. Let us call the first convex region obtained from the first set of three extreme points as CR_1. It may also be noted that subsequent interior points will be based on other three extreme points, therefore, in a similar way we denote these subsequent convex region by CR_i, for $i = 1, 2, \ldots$. All these convex regions may not have any relationship among them but they all share a common relationship with the CR that is each CR_i *is a sub-space of* CR. The direction of the normal, which is a function of the given coefficients of the objective function, will increase the value of the objective function only if it can cross through the hyper-plane that shares the boundary with feasible region on either side. This is possible by searching at least three extreme points which are all adjacent to the current extreme point in different directions.

Thus, instead of finding three consecutive extreme points from the current location, we find r number of extreme points from the same location, where $r \geq 3$. Note that these extreme points from the same location in r random directions can be obtained in much less computational effort. Also the normal will also be facing the boundary, which is feasible on either side.

7.4.1 A few observations from the sub-problem of (7.13)

Observation 1: Since the coefficients c_j in equation (7.13) are non-negative, the normal direction to the hyper-plane representing the objective function of the sub-problem (7.13) will be confined to the positive quadrant in the n_1-dimensional space defined by the equation (7.13).

Observation 2: Consider that a feasible extreme point of the convex region defined by the given LP of equation (7.13) is a known extreme point. If that point is not optimal the non-optimality may be reflected by a large number of non-basic variables, each such variable reflecting non-optimality can generate a feasible extreme point of the convex region of (7.13). A random selection of r variables can generate r extreme points in different directions. An average of the selected extreme points will give rise to an interior point from where the direction of the normal will always be in the positive quadrant increasing the value of the objective function. Thus, instead of successive extreme points (like in the simplex approach), it is better to generate two or more extreme points, which are adjacent to the same feasible extreme point. Selection of these entering variables is random among those which qualify for entry to the basis. Also note that the computational effort required to generate the interior point in the above manner will require much less computational effort.

Observation 3: A known result in LP is that an LP is unbounded if the minimum positive ratio does not exist and $c_j > 0$. A similar, but inverse property of a LP could be that if a variable x_j is such that $A_j \geq 0$ and $c_j \leq 0$, then the optimal value of the variable $x_j = 0$. In other words, that variable will consume resources without giving any positive return. Thus, it will never qualify to enter the basis. Let us label this variable as a **permanent non-basic** variable which can can be removed from the LP model. Therefore, some variables in the equation (7.1) may qualify to be **permanent non-basic** variables.

Observation 4: Since in the LP model in equation (7.13), the requirement of non-negative coefficients on the constraints set is not imposed, these coefficients may be positive or negative quantities, hence rounding up as well as rounding down will have to be checked for feasibility. Thus, rounding in this general model may not be desirable.

Observation 5: Let the location of a known interior point be IP_k and assume that the value of the objective function at this point is also known, which may be denoted by Z_{IP_k}. In order to achieve maximum increase in the value of the objective function, it is proposed to move from this location in the normal direction of the given objective hyper-plane. Thus, the new location is given by $IP_k + \alpha C^{T_{n_i}}$, where $\alpha \geq 0$. Note that $C^{T_{n_i}}$ is the normal direction of the given objective function of the model in equation (7.13). The scalar $\alpha \geq 0$ has to be calculated as the largest value such that $Z = C^{T_{n_i}} X_{n_i}$ is maximised subject to $A_{(m \times n_i)}(IP_k + \alpha C^{T_{n_i}}) \leq B$ and $IP_k + \alpha C^{T_{n_i}} \geq 0$ where:

$$IP_{k+1} = IP_k + \alpha C^{T_{n_i}} \tag{7.14}$$

The constant α is given by:

$$\alpha = \min \left\{ \frac{B - A_{(m \times n_i)} IP_k}{AC^{T_{n_i}}} \right\} \tag{7.15}$$

and also $\alpha C^{T_{n_i}} \geq -IP_k$.

Once the value of α is known the new point with an improved value of the objective function is a known point. The new improved feasible location may be an extreme point or may be a point on the boundary of a constraint of the given LP equation (7.13). It is denoted by $IP_{(k+1)}$ and it can be determined from equation (7.16).

$$IP_{k+1} = IP_k + \alpha C^{T_{n_i}} \tag{7.16}$$

The new LP is given by equation (7.17).

$$\left.\begin{array}{l} \text{Maximise } Z = C^{T_{n_i}} X_{n_i} = IP_{k+1} \\ \text{subject to,} \\ A_{(m \times n_1)} X_{n_1} \leq B \\ Z \geq Z_{IP_{k+1}} \\ X_{n_i} \geq 0 \end{array}\right\} \tag{7.17}$$

Here $X_{n_i} = IP_{k+1}$. Note that if α as obtained from (7.14) is equal to zero, it means one has reached to the end of the feasible space in the normal direction of the objective function. If an optimal solution has not been identified, one has to change the search direction, which is easily achieved by:

1. Identifying the current extreme point of the convex space of (7.17). This can be achieved by Phase 1 calculations of the simplex method on the model (7.17).
2. From this extreme point, either continue the usual simplex iterations to reach the optimal point, or repeat the above steps to reduce the feasible space to the next improved point IP_{k+2} from the point IP_{k+1}.

Observation 6: The new point $X = IP_{k+1}$ will always be the extreme point of the convex space defined by (7.17). In other words, the problem defined by (7.17) has a feasible region.

Proof: Note that the point defined by equation (7.14) will remain feasible to the model of equation (7.13) as all constraints of equation (7.13) are satisfied at this point and the additional constraint in the LP model of equation (7.13) is also satisfied. Furthermore, it may be noted that at this point at least one of the constraint of (7.13) is active, and also the additional constraint in the model (7.13) is also active. Thus, it is an extreme point at intersection of at least two active constraints of the model (7.13).

Observation 7: Once an optimal solution to the LP in equation (7.13) is established, that solution can be used to establish the optimal solution to the model of equation (7.1) by the application of column generation rule for the rest of the variables with negative coefficients in the objective function.

Observation 8: For a given m constraint n_1 variable LP, the optimal solution of the LP comprised of n_1 variables where $n_1 \leq n$, will form a lower bound on the optimal value of the given LP. The proof is obvious as any other additional given variable not among the n_1 variables can either increase the value of the objective function, or remain non-basic at zero value.

7.4.2 An algorithm for solving the general large-scale LP model

From the above discussion, one can generate an algorithm for solving a general large-scale LP.

Step 1

Consider an LP model of equation (7.12). Check the objective function for positive coefficients. If all C_j's are non-negative, then $n_1 = n$ and $n - n_2 = 0$. Go to Step 2. If some of these coefficients are negative quantities, rearrange the given LP model in the structure of LP model (7.12), i.e. separating positive and negative coefficients of the objective function and rearranging the constraints accordingly, to develop a sub-LP model of the form (7.12), and then develop the LP model of equation (7.13). Go to Step 2.

Step 2

Set $k = 0$.

Step 3

Obtain an initial feasible point IP_k of the LP, which is given by

$$X = \begin{bmatrix} x_1 \\ ... \\ x_{n_1} \end{bmatrix} = \begin{bmatrix} 0 \\ ... \\ 0 \end{bmatrix}$$

If the number of entering variables are more than a pre-assigned value $r \geq 3$, create at random, r feasible extreme points, which lead to an improved value of the objective function. From these r extreme points, develop an interior point by averaging their values. Call this point IP_{k+1}.

Step 4

Move from the initial point IP_{k+1} in the direction of the normal to the objective function within the feasible space as far as possible. Let this new feasible point be denoted by IP_{k+2}, and the value of the objective function at this point is denoted by $Z_{IP_{k+2}}$.

Step 5

Formulate a new LP problem of equation (7.17) i.e.

$$\left.\begin{array}{l} \text{Maximise } Z = C^{T_{n_i}} X_{n_i} \\ \text{subject to,} \\ A_{(m\times n_1)} X_{n_1} \leq B \\ C^{T_{n_1}} X_{n_i} \geq Z_{IP_{k+2}} \\ X_{n_i} \geq 0 \end{array}\right\} \tag{7.18}$$

where $IP_{k+2} = IP_{k+1} + \alpha C^{T_{n_1}}$

Step 6

Set $k = k + 1$

Step 7

Using any of the existing LP software, find a feasible extreme point IP_{k+2}, and return to Step 3. Develop the simplex tableau corresponding to the point IP_{k+2} as a basic feasible solution. Check the number of variables that may qualify for entry to the basis. If the number of non-basic variables that qualify for entry is greater or equal to the pre-assigned value r, continue with the simplex iterations, and when an optimal solution has been identified, go to Step 8, else go to Step 3.

Step 8

Conclude the search process as the optimality condition has been satisfied. Print the optimal solution to model (7.12). Consider the original LP model of (7.12) and select a variable x_j for which $c_j < 0$ for $j = n_1 + 1, \ldots, n$.

Step 9

Since $c_j \leq 0$, use column generation to find $Z_j - c_j = C_B B^{-1} A_j - c_j$. If $Z_j - c_j < 0$, carry out the usual simplex iteration, and if $Z_j - c_j \geq 0$, go to Step 10.

Step 10

Set $j = j + 1$. If $j < n$, go to Step 9, else go to Step 11.

Step 11

Print the optimal solution to the given LP model.

7.4.3 Analysis and results

Example 1 (See Page 196 for example 2)

Consider the Klee-Minty Cube example given as:

$$\left.\begin{aligned} &\text{Maximise } Z = \sum_{j=1}^{n} 10^{n-j} x_j \\ &\text{subject to,} \\ &2\sum_{j=1}^{i-1} 10^{i-j} x_j + x_i \leq 100^{i-1} \\ &x_j \geq 0, \quad \text{for } i = 1, 2, ..., n \text{ and } j = 1, 2, ..., n \end{aligned}\right\} \tag{7.19}$$

For $n = 3$ the problem will be given by:

$$\left.\begin{aligned} &\text{Maximise } Z = 100x_1 + 10x_2 + x_3 \\ &\text{subject to,} \\ &x_1 \leq 1, \quad 20x_1 + x_2 \leq 100 \\ &200x_1 + 20x_2 + x_3 \leq 10\ 000 \\ &x_1,\ x_2,\ x_3 \geq 0, \end{aligned}\right\} \tag{7.20}$$

In general, the simplex method can solve the Klee-Minty model in (2^{n-1}) iterations. For $n = 3$, it will require seven iterations. The optimal solution can be easily verified as: $x_3 = 10\ 000$, $x_1 = x_2 = 0$ and $z = 10\ 000$. Now, we apply the proposed method discussed in this thesis. The initial extreme point is

$$EP_0 = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Let $r = 3$. The three possible extreme points from the initial point that can be reached are given by:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 100 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 10000 \end{bmatrix}$$

The interior point generated by these three points will be given by:

$$IP_0 = \begin{bmatrix} \frac{1}{3} \\ \frac{100}{3} \\ \frac{10000}{3} \end{bmatrix}$$

and the objective value at this point will be given by:

$$Z_{IP_0} = \frac{11100}{3} = 3700.$$

To find the value of α, we have:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 20 & 1 & 0 \\ 200 & 20 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 100 \\ 10000 \end{bmatrix}, C^T = \begin{bmatrix} 100 \\ 10 \\ 1 \end{bmatrix}, IP_1 = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{100}{3} \\ \frac{10000}{3} \end{bmatrix}$$

We therefore obtain

$$AIP_0 = \begin{bmatrix} \frac{1}{3} \\ 40 \\ 7400 \end{bmatrix}, AC^T = \begin{bmatrix} 2010 \\ 100 \\ 20201 \end{bmatrix}, B - AIP_0 = \begin{bmatrix} 0.67 \\ 60 \\ 2600 \end{bmatrix}$$

Resulting in

$$\alpha = \text{min} \begin{bmatrix} 100\alpha \leq 0.67 \\ 2010\alpha \leq 60 \\ 20201\alpha \leq 2600 \end{bmatrix} = \text{min} \begin{bmatrix} 0.0067 \\ 0.0298 \\ 0.1287 \end{bmatrix}$$

= 0.0067.

Therefore, the improved interior point is:

$$IP_2 = IP_1 + \alpha C^T = \begin{bmatrix} \frac{1}{3} \\ \frac{100}{3} \\ \frac{10000}{3} \end{bmatrix} + 0.0067 \begin{bmatrix} 100 \\ 10 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.0 \\ 33.4 \\ 3333.34 \end{bmatrix}$$

The new constraint to be added will be:

$100x_1 + 10x_2 + x_3 \geq Z_{IP_1} = 3767.34.$

Thus, the modified problem becomes:

$$\left.\begin{aligned} &\text{Maximise } Z = 100x_1 + 10x_2 + x_3 \\ &\text{subject to,} \\ &x_1 \leq 1, \quad 20x_1 + x_2 \leq 100 \\ &200x_1 + 20x_2 + x_3 \leq 10\,000 \\ &100x_1 + 10x_2 + x_3 \geq 3767.34 x_1,\; x_2,\; x_3 \geq 0, \end{aligned}\right\} \tag{7.21}$$

The feasible solution to the model of equation (7.21) can be obtained by any method and it is given in Table 7.1.

For the optimal solution, one more pivotal iteration will give the optimal solution shown in Table 7.2, which is $x_3 = 10000$, $x_1 = x_2 = 0$ and $Z = 10000$

Table 7.1: Feasible solution to model in equation (7.21)

i\j	x_1	x_2	x_3	S_1	S_2	S_3	S_4	RHS
s_1	1	0	0	1	0	0	0	1
x_2	20	1	0	0	1	0	0	100
x_3	-100	0	0	0	-10	1	1	5332.66
x_3	-100	0	1	0	-10	0	-1	2767.34
$Z - c_j$	0	0	0	0	0	0	-1	3767.34

Table 7.2: Feasible solution to model in equation (7.21)

i\j	x_1	x_2	x_3	S_1	S_2	S_3	S_4	RHS
s_1	1	0	0	1	0	0	0	1
s_2	20	1	0	0	1	0	0	100
x_3	0	20	1	0	0	1	0	10000
s_4	100	10	0	0	0	1	1	2189.7
$Z - c_j$	0	10	0	0	0	0	0	10000

Example 2

Consider a LP model as given in equation (7.22)

$$\left.\begin{array}{l}
\text{Maximise } Z = 2x_1 + 5x_2 + 3x_3 + 4x_4 + 2x_5 + 3x_6 + 2x_7 + x_8 + 2x_9 - x_{10} - 2x_{11} - 3x_{12} \\
\text{subject to}, \\
x_1 + 0x_2 - x_3 + 8x_4 - 2x_5 + 4x_6 + 0x_7 + 3x_8 + x_9 + 4x_{10} - 2x_{11} + 0x_{12} \leq 24 \\
2x_1 + 2x_2 + 0x_3 + 3x_4 + 3x_5 + 0x_6 + 0x_7 + 4x_8 + 0x_9 + 0x_{10} + x_{11} + x_{12} \leq 32 \\
4x_1 + 5x_2 + 0x_3 + 0x_4 + 0x_5 + 4x_6 + 0x_7 + 0x_8 + 2x_9 + 0x_{10} + 5x_{11} + x_{12} \leq 30 \\
3x_1 + 0x_2 + 3x_3 + 2x_4 + 3x_5 - 2x_6 + 4x_7 + 2x_8 + 0x_9 + 1x_{10} + x_{11} + 0x_{12} \leq 30 \\
0x_1 + 4x_2 + 0x_3 - 2x_4 0x_5 - 0x_6 + 0x_7 - 2x_8 + 0x_9 + 2x_{10} + 3x_{11} + x_{12} \leq 22 \\
2x_1 + 1x_2 + 4x_3 + 1x_4 + 2x_5 + 02x_6 + 5x_7 + x_8 + 0x_9 + 0x_{10} + 2x_{11} + 3x_{12} \leq 12 \\
0x_1 + 0x_2 + x_3 + 0x_4 + 1x_5 + 2x_6 + 0x_7 + 4x_8 + 5x_9 + 2x_{10} + 2x_{11} + 0x_{12} \leq 28 \\
1x_1 + 0x_2 + 0x_3 + 3x_4 + 0x_5 + x_6 + x_7 + 2x_8 + 6x_9 + 1x_{10} + x_{11} + 2x_{12} \leq 36 \\
3x_1 + 2x_2 + 2x_3 + x_4 - 2x_5 + 0x_6 - x_7 + 0x_8 + 3x_9 + 0x_{10} + 4x_{11} + 2x_{12} \leq 18 \\
x_1, x_2, ..., x_{12} \geq 0
\end{array}\right\} \tag{7.22}$$

Using a LP package, the optimal solution to the above LP is given by $Z = 46.64$, with $x_2 = 4.72, x_4 = 3.12, x_5 = 2.08, x_9 = 3.2$ and $x_1 = x_3 = x_6 = x_7 = x_8 = x_10 = x_{11} = x_{12} = 0$. We can now apply the method developed in this chapter. Note that variables with positive coefficients in the objective function are given by: $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$ and negative coefficients variables are (x_{10}, x_{11}, x_{12}). The value of n_1 is 9. A sub-problem of nine or less variables can be developed. Let us consider all these nine variables to form a LP model similar to equation (7.13). This is given by equation (7.23) as follows:

$$\left.\begin{array}{l}\textbf{Maximise } Z = 2x_1 + 5x_2 + 3x_3 + 4x_4 + 2x_5 + 3x_6 + 2x_7 + x_8 + 2x_9 \\ \textbf{subject to}, \\ x_1 + 0x_2 - x_3 + 8x_4 - 2x_5 + 4x_6 + 0x_7 + 3x_8 + x_9 \leq 24 \\ 2x_1 + 2x_2 + 0x_3 + 3x_4 + 3x_5 + 0x_6 + 0x_7 + 4x_8 + 0x_9 \leq 32 \\ 4x_1 + 5x_2 + 0x_3 + 0x_4 + 0x_5 + 4x_6 + 0x_7 + 0x_8 + 2x_9 \leq 30 \\ 3x_1 + 0x_2 + 3x_3 + 2x_4 + 3x_5 - 2x_6 + 4x_7 + 2x_8 + 0x_9 \leq 30 \\ 0x_1 + 4x_2 + 0x_3 - 2x_4 0x_5 - 0x_6 + 0x_7 - 2x_8 + 0x_9 \leq 22 \\ 2x_1 + 1x_2 + 4x_3 + 1x_4 + 2x_5 + 02x_6 + 5x_7 + x_8 + 0x_9 \leq 12 \\ 0x_1 + 0x_2 + x_3 + 0x_4 + 1x_5 + 2x_6 + 0x_7 + 4x_8 + 5x_9 \leq 28 \\ 1x_1 + 0x_2 + 0x_3 + 3x_4 + 0x_5 + x_6 + x_7 + 2x_8 + 6x_9 \leq 36 \\ 3x_1 + 2x_2 + 2x_3 + x_4 - 2x_5 + 0x_6 - x_7 + 0x_8 + 3x_9 \leq 18 \\ x_1, x_2, ..., x_9 \geq 0 \end{array}\right\} \quad (7.23)$$

For the LP model (7.22), we have:

$$A = \begin{bmatrix} 1 & 0 & -1 & 8 & 5 & -2 & 4 & 3 & 1 \\ 2 & 2 & 0 & 3 & 3 & 0 & 0 & 4 & 0 \\ 4 & 5 & 0 & 0 & 0 & 4 & 0 & 0 & 2 \\ 3 & 0 & 3 & 2 & 3 & -2 & 4 & 2 & 0 \\ 0 & 4 & 0 & -2 & 0 & 0 & 0 & -2 & 0 \\ 2 & 1 & 4 & 1 & 2 & 0 & 5 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 2 & 0 & 4 & 5 \\ 1 & 0 & 0 & 3 & 0 & 1 & 1 & 2 & 6 \\ 3 & 2 & 2 & 1 & -2 & 0 & -1 & 0 & 3 \end{bmatrix}, B = \begin{bmatrix} 24 \\ 32 \\ 30 \\ 30 \\ 22 \\ 12 \\ 28 \\ 36 \\ 18 \end{bmatrix}, C^T = \begin{bmatrix} 2 \\ 5 \\ 3 \\ 4 \\ 2 \\ 3 \\ 2 \\ 1 \\ 2 \end{bmatrix}, X^T = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix}$$

The initial extreme point is

$$EP_0 = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Let $r = 3$. From this initial extreme point, we generate three extreme points with respect to variables x_2, x_3 and x_4. These points, expressed in variables $(x_1, x_2, \ldots, x_9)$, are given by:

$$EP_{x_2} = \begin{bmatrix} 0 \\ 5.2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, EP_{x_3} = \begin{bmatrix} 0 \\ 0 \\ 3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, EP_{x_4} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

resulting in

$$IP_0 = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 1.8 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The objective value at this point is $Z_{IP_0} = 1.8(5) + 1(3) + 1(4) = 16$. Since C^T is the normal direction of the given objective function, we need to find the value of α. We first find AIP_0, AC^T and $B - AP_0$ as:

$$AIP_0 = \begin{bmatrix} 1 & 0 & -1 & 8 & 5 & -2 & 4 & 3 & 1 \\ 2 & 2 & 0 & 3 & 3 & 0 & 0 & 4 & 0 \\ 4 & 5 & 0 & 0 & 0 & 4 & 0 & 0 & 2 \\ 3 & 0 & 3 & 2 & 3 & -2 & 4 & 2 & 0 \\ 0 & 4 & 0 & -2 & 0 & 0 & 0 & -2 & 0 \\ 2 & 1 & 4 & 1 & 2 & 0 & 5 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 2 & 0 & 4 & 5 \\ 1 & 0 & 0 & 3 & 0 & 1 & 1 & 2 & 6 \\ 3 & 2 & 2 & 1 & -2 & 0 & -1 & 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1.8 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 7 \\ 6.6 \\ 9 \\ 5 \\ 5.2 \\ 6.8 \\ 1 \\ 3 \\ 6.6 \end{bmatrix},$$

and

$$AC^T = \begin{bmatrix} 1 & 0 & -1 & 8 & 5 & -2 & 4 & 3 & 1 \\ 2 & 2 & 0 & 3 & 3 & 0 & 0 & 4 & 0 \\ 4 & 5 & 0 & 0 & 0 & 4 & 0 & 0 & 2 \\ 3 & 0 & 3 & 2 & 3 & -2 & 4 & 2 & 0 \\ 0 & 4 & 0 & -2 & 0 & 0 & 0 & -2 & 0 \\ 2 & 1 & 4 & 1 & 2 & 0 & 5 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 2 & 0 & 4 & 5 \\ 1 & 0 & 0 & 3 & 0 & 1 & 1 & 2 & 6 \\ 3 & 2 & 2 & 1 & -2 & 0 & -1 & 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 5 \\ 3 \\ 4 \\ 2 \\ 3 \\ 2 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 48 \\ 36 \\ 49 \\ 33 \\ 10 \\ 40 \\ 25 \\ 33 \\ 26 \end{bmatrix}$$

Also

$$B - AP_0 = \begin{bmatrix} 24 \\ 32 \\ 30 \\ 30 \\ 22 \\ 12 \\ 28 \\ 36 \\ 18 \end{bmatrix} - \begin{bmatrix} 7 \\ 6.6 \\ 9 \\ 5 \\ 5.2 \\ 6.8 \\ 1 \\ 3 \\ 6.8 \end{bmatrix} = \begin{bmatrix} 17 \\ 25.4 \\ 21 \\ 25 \\ 16.8 \\ 5.2 \\ 27 \\ 33 \\ 11.2 \end{bmatrix}$$

Since

$$\alpha = \mathbf{min}\{(\frac{B - IP_k}{AC^T})\} = \begin{bmatrix} 17/48 \\ 25.4/36 \\ 21/49 \\ 25/33 \\ 16.8/10 \\ 5.2/40 \\ 27/25 \\ 33/33 \\ 11.2/26 \end{bmatrix} = \begin{bmatrix} 0.35 \\ 0.71 \\ 0.43 \\ 0.76 \\ 1.68 \\ 0.13 \\ 1.08 \\ 1.00 \\ 0.43 \end{bmatrix} = 0.13$$

Using equation (7.15) and moving in the normal direction of the objective hyperplane, the improved interior point is:

$$IP_1 = IP_0 + \alpha C^T = \begin{bmatrix} 0 \\ 1.8 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0.13 \begin{bmatrix} 2 \\ 5 \\ 3 \\ 4 \\ 2 \\ 3 \\ 2 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0.26 \\ 2.45 \\ 1.39 \\ 1.52 \\ 0.26 \\ 0.39 \\ 0.26 \\ 0.13 \\ 0.26 \end{bmatrix}$$

and $Z_{IP_1} = 2(0.26)+5(2.45)+3(1.39)+4(1.52)+2(0.26)+3(0.39)+2(0.26)+1(0.13)+2(0.26) = 26.88$.

Thus, a basic feasible solution to equation (7.23) is required in presence of the additional constraint given by equation (7.24) which is:

$$Z = 2x_1 + 5x_2 + 3x_3 + 4x_4 + 2x_5 + 3x_6 + 2x_7 + x_8 + 2x_9 \geq 26.88(Z_{IP_1}) \qquad \textbf{(7.24)}$$

The 9-variable problem of equation (7.23) was solved using the LIPS package and the results were, as expected, the same as for the given 12-variable problem of equation (7.22). These results are: $x_2 = 4.72, x_4 = 3.12, x_5 = 2.08, x_9 = 3.2$ and $x_1 = x_3 = x_6 = x_7 = x_8 = 0$ giving $Z_{opt} = 46.64$. Although the solution is optimal, to establish the optimality of equation (7.22), we need inverse of the basic matrix of the 9-variable problem. From the LIPS package, the inverse of the matrix B and set of basic variables were obtained as given in Table 7.3.

Table 7.3: Basic variables, solution and inverse of matrix B

Basic	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	S_{18}	S_{19}	S_{20}	S_{21}	RHS
S_{17}	$2/75$	0	$-4/3$	0	1	$68/75$	0	0	0.89	9.36
S_{14}	$-31/150$	1	$-1/6$	0	0	$-229/150$	0	0	0.18	6.96
x_4	0.12	0	0	0	0	0.08	0	0	-0.04	3.12
S_{16}	0.02	0	0.5	1	0	-1.82	0	0	-0.34	17.52
x_2	$4/75$	0	$1/3$	0	0	$-14/75$	0	0	-0.24	4.72
x_5	$-13/150$	0	$-1/6$	0	0	$83/150$	0	0	0.14	2.08
S_{19}	$113/150$	0	$11/6$	0	0	$-433/150$	1	0	-3.14	9.92
S_{20}	0.44	0	2	0	0	-3.04	0	1	-3.48	7.44
x_9	$-2/15$	0	$-1/3$	0	0	$7/15$	0	0	0.6	3.2
$Z_j - c_j$	$23/75$	0	$2/3$	0	0	$107/75$	0	0	0.12	46.64

From Table 7.3, one can, for optimality, get values of $Z_{x_{10}} - C_{10} = 167/75, Z_{x_{11}} - C_{11} = 604/75, Z_{x_{12}} - C_{12} = 614/75$, which confirms optimality of the given problem of equation (7.22).

7.4.4 Concluding remarks

The above method is suitable for a large-scale LP of the type of equation (7.1). Advantages, if any, can be assessed only after developing an appropriate software and comparing computational efforts required by the simplex method and the proposed hybrid method. From the initial point, a linear move in the normal direction is desirable if the increase in the objective function value can reduce the number of simplex iterations. The proposed method is likely to converge to the solution faster than the simplex method by utilising movements in the normal direction to the objective function. A few iterations of moving in the normal direction give an approximate solution, which is not provided by any other method. Instead of dealing with the given problem of dimension $m \times (n+m)$, in the proposed method we are dealing with $n_1 \times (n_1+m)$ problem, thus reducing pivoting computations in each iteration.

7.5 A Heuristic for a Mixed Integer Program using the Characteristic Equation Approach

7.5.1 Introduction

An integer linear program is a linear program which is further constrained by integer restrictions on some or all variables. When all variables are integer restricted, it is called a pure integer program (PIP) model and when only some of the variables are restricted to integer values, it becomes a mixed integer programming model. Integer programming (IP) models frequently arise in human resource planning, facility location, assignment problems, production planning, time-tabling, warehouse location, scheduling and capital budgeting, just to mention a few. While most linear programming (LP) problems can be solved in polynomial time, PIP and MIP are NP-complete problems, which have no known polynomial time algorithm to solve them (Bertacco et al., 2007).

In this thesis, the PIP model is solved by using the characteristic equation (CE) with the hope that this approach may provide insight into MIP solution procedures and applications. Generally, MIP problems have been solved using the LP-based branch and bound (BB) solvers or with stochastic search-based solvers (Noraini and Geraghty, 2011). In reality MIP solvers have implemented more sophisticated versions denoted by branch and cut (BC) algorithms (Sen and Sherali, 2006). With the increase in the application of both PIP and MIP models, it is of paramount importance that methods sought are capable of finding a global optimal solution. The major disadvantages of existing methods, like round off errors, and creation or emergence of many sub-problems (branches), is the time taken to obtain the optimal solution and failure to obtain global optimal solutions. These justify the need to find better approaches for MIP problems.

In this thesis, a hybrid of the existing approaches for solving the LP and PIP models has been used for solving the MIP model. A PIP has been solved by using a descending hyper-plane that was developed by Kumar et al. (2007), which was later renamed a characteristic equation by Kumar and Munapo (2012). For the MIP model, the LP solution acts as an upper bound (UB) and the PIP solution as the lower bound (LB). In a MIP model, one has to deal with integer restricted variables as well as continuous variables, requiring distribution of the available resources for these two types of variables. This aspect of distribution has been addressed by the CE, since ordered optimal solutions can be obtained by the CE for a PIP model. The proposed method generates a good feasible solution with bounds, and eliminates rounding off errors and dealing with sub-problems, as is commonly required in existing BB methods.

7.5.2 Mathematical development for the proposed method

Consider a general mixed integer programming problem:

$$\left.\begin{array}{l} \text{Minimise or Maximise } Z = C^T X, \\ \text{ subject to}, \\ AX \leq / = / \leq B \\ X \geq 0 \\ X_I \in Z \end{array}\right\} \tag{7.25}$$

where C^T is $1 \times n$, X is $(n \times 1)$, A is $(m \times n)$ and B is $(m \times 1)$. Let $n_I < n$ represent integer restricted non-negative variables. The remaining $(n - n_I)$ variables are such that $x_j \geq 0$. In equation (7.25), X_I represents integer restricted non-negative variables in X, implying that $X_I \geq 0$. The MIP relaxation, i.e. LP of the model in equation (7.25) is given by:

$$\left.\begin{array}{l}\text{Minimise or Maximise } Z = C^T X,\\ \text{ subject to,}\\ AX \leq / = / \leq B\\ X \geq 0\end{array}\right\} \tag{7.26}$$

The pure integer programming model of (7.21) is given by:

$$\left.\begin{array}{l}\text{Minimise or Maximise } Z = C^T X,\\ \text{ subject to,}\\ AX \leq / = / \leq B\\ X \geq 0 \text{ and integer}\end{array}\right\} \tag{7.27}$$

After obtaining the PIP solution to (7.27), one can also develop a modified LP from the given MIP (equation 7.25) when all integer restricted variables are replaced by their values and the problem reduces to $(n - n_I)$ variables, where all these variables are non-negative restricted real variables. Here onwards, we will call it a modified LP model. The three problems (7.25), 7.26) and (7.27), and the modified LP have close relationships among themselves, which can be used to develop a method for solving the MIP model of (7.25). Some of these relationships that may be of immediate interest to us are discussed next.

Observation 1

The LP relaxation model (7.26) is a least constrained model among the three models: (7.25), (7.26) and (7.27), hence the LP optimal solution of model (7.26) will be an upper bound to the MIP model (7.25).

Observation 2

A feasible solution to the PIP model (7.27) will also be feasible to the MIP model (7.25).

Proof

Since a feasible solution X to model (7.27) will also satisfy the requirement for the MIP model (7.25), where n_I of the n variables are required to have integer restricted values, it then follows that all feasible solutions to the PIP model (7.27) will also be feasible to the MIP model (7.25). Therefore, the PIP optimal solution will act as a lower bound to the MIP model (7.25).

Observation 3

The MIP model involves a two-way distribution of resources. The first distribution of resources involves the division of the resource vector B among the integer and continuous variables, and the second distribution gives rise to values of the basic variables. The distribution of the resource vector B is achieved by determining ordered optimal solutions to the PIP model. These ordered optimal solutions can be obtained by using the CE developed by Kumar et al. (2007), and Kumar and Munapo (2012). The distribution of resources to continuous variables is obtained by the LP model.

Observation 4

The optimal PIP solution X_{PIPopt} has a property that if $x_j = \beta_j$ is an element of this optimal integer solution, then $x_j = \beta_j + 1$ along with other variables at their optimal values will always lead to an infeasible solution.

7.5.3 The problem

In the proposed approach, for the given MIP model, we first develop two parallel LP and PIP models. The optimal value of the relaxed objective is denoted by Z_{LP}, which acts as the upper bound to the given MIP model. For any optimisation problem Z, let $F(Z)$ denote its set of feasible solutions. The only requirement for Z_{LP} to be a valid relaxation of Z is $F(Z) \subseteq F(Z_{LP})$ which is true in this case. The relaxed problem is easier to solve than the original problem, and the Z_{LP} gives the upper bound (Geoffrion and Marsten, 1972).

From the optimal solution of Z_{LP} a characteristic equation (CE) is formed to resolve the PIP model (7.27), where all variables are restricted to integer values. The advantage of the CE is that it can provide the best, second best, third best solutions, etc. for the PIP model (7.27). The optimal PIP solution acts as the LB to the given MIP model. The gap between the UB given by Z_{LP} and the LB given by Z_{PIP} can be decreased by the LP modified model that can be formulated by substituting values of the integer restricted variables in the MIP model (7.25), and solving the remaining model as a LP model in $(n - n_I)$ variables. The combined solution will be a feasible solution to the MIP model (7.25). This will give rise to an improved LB, denoted by Z_{MIP}. If the difference $(Z_{LP} - Z_{MIP})$ is insignificant or is zero, then either the optimal solution or near optimal solution to the MIP is obtained.

7.5.4 The characteristic equation

The CE is obtained from the final tableau of the LP relaxation. It is a mapping of the integer hyper-plane on feasible integer points and the mapping of the objective function on interior integer points in a descending order. The CE is based on the following three basic ideas:

1.The objective value must be an **integer**.

2. Non-basic variables are either zero (as in the LP solution) or if some of them are not zero, they must be an **integer quantity**.

3. Basic variables are also functions of the non-basic variables, and for the non-zero non-basic variables, the basic variables must also become integer values in a PIP model.

The major advantage of using the CE is that convergence is guaranteed and it can be used to obtain ordered optimal solutions for the PIP problem. However, the solution of the CE can be a challenge, and also if there are more than one solution to the CE, all those solutions have to be tested for integer solutions. The LP extreme points are an intersection of at most m constraints and the LP optimal solution is an intersecting point of one more hyper-plane represented by the objective function. The CE is a mapping of the hyper-plane on feasible integer points. From the optimal relaxation LP solution, one can write the objective function row as:

$$\frac{D}{D}Z + \frac{\beta_1 s_1 + \beta_2 s_2 + ... + \beta_k s_k}{D} = \frac{R + iD}{D} \tag{7.28}$$

where k represents the number of non-basic variables, β_j represents the integer coefficients, D is the lowest common factor for all terms and R is the remainder in the RHS value. The CE is then given by:

$$\beta_1 s_1 + \beta_2 s_2 + ... + \beta_k s_k = R + iD \text{ for } i = 0, 1, 2, 3..... \tag{7.29}$$

When the LHS of equation (7.29) is equal to the RHS, the objective function Z is guaranteed to be an integer value. Since we are looking for an integer solution to the CE, we also got the condition (2) in Section 7.5.4 satisfied. For the solution to be an acceptable feasible solution, the solution of the CE solution must also convert the values of the basic variables to non-negative integer values. The next best solution is obtained by further reduction in the value of the objective function, which is possible by increasing the value of i. It may also be noted that when the PIP model of equation (7.27) is solved for the optimal solution, all variables have non-negative integer values. We substitute from this solution, values of n_1 integer restricted values in the MIP model (7.25) and get a modified LP in $(n - n_1)$ real variables, which is a LP model and can be solved by any known method. The combined integer restricted values from the PIP and LP solutions for the modified LP gives a feasible solution to the MIP, hence acts as a LB.

7.5.5 Algorithmic steps of the method

The method is comprised of the following steps:

Step 1:

Solve the relaxed LP model (7.26), and find the value of Z_{LP} which will be an UB to the given problem.

Step 2:

Obtain the characteristic equation (CE) from the solution of Z_{LP}. Set $k = 1$.

Step 3:

Solve the CE and obtain k^{th} best integer solution for minimum i. This will be a solution to the PIP, i.e. when all are integer restricted variables.

Step 4:

The PIP solution from Step 3 will be a lower bound (LB) to the given MIP. Let this LB be denoted by Z_{PIP}.

Step 5:

If $Z_{LP} - Z_{PIP}$ is equal to zero, or approximately equal to zero, go to Step 10. Else go to Step 6.

Step 6:

Substitute the integer solution for the integer restricted variables obtained from Step 3 into the original MIP model (7.25), and get a modified LP in $(n-n_I)$ variables.

Step 7:

Solve the LP in $(n - n_I)$ real variables obtained at Step 6.

Step 8:

Combine the integer solution of Step 3 for the integer variables and real solution for the real variables from Step 7 to get a feasible solution to the MIP. This value is likely to be less than the UB and more than the LB, and if the difference is insignificant, one can stop the search; else one has to improve the feasible solution. Check if this solution can be declared as the optimal solution. If "yes", go to Step 10, else go to Step 9.

Step 9:

Set $k = k + 1$ and go to Step 3.

Step 10:

Conclude the search process as the optimality condition has been satisfied.

7.5.6 Analysis and results

This example is taken from Hillier and Lieberman (2001).

$$\left.\begin{array}{l}\text{Maximise } Z = 4x_1 - 2x_2 + 7x_3 - x_4, \\ \text{ subject to,} \\ x_1 + 5x_3 \leq 10 \\ x_1 + x_2 - x_3 \leq 1 \\ 6x_1 - 5x_2 \leq 0 \\ -x_1 + 2x_3 - 2x_4 \leq 3 \\ x_1, x_2, x_3 \geq 0 \text{ and integer} \\ x_4 \geq 0 \end{array}\right\} \tag{7.30}$$

Using LIPS program the LP relaxation solution found is shown in Table 7.4 and the final simplex solution is shown in Table 7.4.

Table 7.4: Relaxed solution to problem in equation (7.30)

Variable	**Value**	**Obj. cost**	**Reduced cost**
x_1	1.25	4	0
x_2	1.5	-2	0
x_3	1.75	7	0
x_4	0	-1	1

For the solution $Z_{LP} = 14.5$, it means the $UB = 14.25$, for the given MIP problem of equation (7.30). Now solve (7.30) again as a PIP model, for which the CE will be required, and it is obtained from equation (7.31):

Table 7.5: Final table of the simplex iterations for solution to problem in equation (7.30)

Basis	x_1	x_2	x_3	x_4	s_5	s_5	s_7	s_8	RHS
x_2	0	1	0	0	0.1	0.5	-0.1	0	1.50
S_8	0	0	0	-2	$-\frac{17}{60}$	$\frac{7}{12}$	$\frac{7}{60}$	1	0.75
x_1	1	0	0	0	$\frac{1}{12}$	$\frac{5}{12}$	$\frac{1}{12}$	0	1.25
x_3	0	0	1	0	$\frac{11}{60}$	$-\frac{1}{12}$	$-\frac{1}{60}$	0	1.75
$Obj.$	0	0	0	-1	$-\frac{17}{12}$	$-\frac{1}{12}$	$-\frac{5}{12}$	0	14.25

$$12x_4 + 17s_5 + s_6 + 5s_7 = 3 + 12i \text{ for } i = 1, 2, ..., 14 \tag{7.31}$$

The non-basic variables x_4, s_5, s_6 and s_7 have to move from the current value of zero to some non-negative integer value such that (7.31) remains satisfied for the minimum i and also converts the current basic variables from the real values to integer values. From Table 7.5, the relations between the basic and non-basic variables are given by:

$$\left.\begin{aligned} x_1 &= 1,25 - (\frac{1}{12})s_5 - (\frac{5}{12})s_6 - (\frac{1}{12})s_7 \\ x_2 &= 1.5 - 0.1s_5 - 0.5s_6 + 0.1s_7 \\ x_3 &= 1.75 - (\frac{11}{60})s_5 - (\frac{1}{12})s_6 - (\frac{1}{60})s_7 \\ s_8 &= 0.75 + 2x_4 + (\frac{17}{60})s_5 - (\frac{7}{12})s_6 - (\frac{7}{60})s_7 \end{aligned}\right\} \tag{7.32}$$

For $i = 0$, equation (7.27) becomes:

$$12x_4 + 17s_5 + s_6 + 5s_7 = 3 \tag{7.33}$$

The only integer solution from equation (7.33) is $x_4 = 0, s_5 = 0, s_6 = 3$ and

$s_7 = 0$. Putting this solution in (7.32), we have $x_1 = x_2 = x_4 = 0, x_3 = 2, s_8 = -1$, which is not feasible, hence one has to try the next value for i which is $i = 1$. The CE is:

$$12x_4 + 17s_5 + s_6 + 5s_7 = 15 \tag{7.34}$$

Equation (7.30) has five integer solutions given in Table 7.6. From Table 7.6

Table 7.6: Relaxed solution to problem in equation (7.30)

Solution	**Comment**
$s_5 = 0, s_6 = 15, s_7 = 0, x_4 = 0$	No feasible integer solution
$s_5 = 0, s_6 = 5, s_7 = 2, x_4 = 0$	$x_1 = x_2 = 0, x_3 = 2, x_4 = 1$ and $Z = 13$
$s_5 = 0, s_6 = 15, s_7 = 0, x_4 = 0$	No feasible integer solution
$s_5 = 0, s_6 = 10, s_7 = 1, x_4 = 0$	No feasible integer solution
$s_5 = 0, s_6 = 0, s_7 = 3, x_4 = 0$	No feasible integer solution

the best integer solution gives a Z value of 13. This value forms a LB. The UB is 14.25. Substituting the integer values for the integer restricted variables from the above solution $(x_1 = x_2 = 0, x_3 = 2)$ into MIP problem of (7.30) we obtain a modified LP in x_4 given by:

$$\left.\begin{aligned} &\text{Maximise } Z = 14x_1 - x_4,\\ &\text{subject to},\\ &4 - 2x_4 \leq 3\\ &x_4 \geq 0 \end{aligned}\right\} \tag{7.35}$$

Solving (7.35) trivially gives $x_4 = \frac{1}{2}$. Combining the integer solution and the above continuous solution we have an improved solution to the given MIP model. It is given by $x_1 = x_2 = 0, x_3 = 2, x_4 = 0.5$ and $Z = 13.5$.

The above solution can be concluded as an optimal solution to the given MIP, as currently we have $13 \leq Z_{MIP} \leq 14.25$ and a feasible solution of 13.5. Note the only variation possible is to increase the values of the variables x_1, x_2 and x_4

which will exceed the current UB. Similarly, increase in x_4 will will worsen the current solution. Hence an optimal solution has been obtained and the search is terminated.

7.5.7 Concluding remarks

The proposed method will have several advantages over existing techniques and some of them include, but are not limited to:

- There are no round off errors since rounding leads to non-optimal or non-feasible solutions.
- The dimension of the matrix A, where A is the coefficients of the decision variables in the constraints, remains unchanged.
- There are no sub-problems as is in the case of BB or BC methods.
- The method does not depend on the initial relaxed solution, but it uses the relaxed solution as the upper bound.
- The method searches the optimal solution using the simplex method, but moves over the integer polyhedron.
- This approach is suitable for changes in the input values, and utilises the characteristic equation to adjust the lower bound.

The solution of a CE can be demanding for larger values of i. Further work is required on how to solve a CE. The method uses the concept of ordered solutions, but for most of the ordered solutions, the value of i will be high. For higher values of i, a solution to CE becomes more demanding. In the BB approach, for a MIP, when a large number of variables are integer restricted, the number of sub-problems can be very high. Sometimes even a feasible solution is not easily obtainable. However, the proposed approach guarantees determination of a feasible solution, and also gives its bounds. The proposed approach works better when integer restricted variables are relatively large. Branch and bound may work better if integer restricted variables are only a few in number. In fields like machine learning, computer vision, advertising, and statistics, it is quite common to encounter MIP formulations with millions of binary decision variables. These algorithms can yield good results in practice, but do not offer any theoretical bounds on runtime and solution quality. The proposed method may prove to be useful for these problems.

7.6 Summary of the Chapter

This chapter concentrated on two linear programming based methods for solving NP-hard problems. Section 7.3 modified the Munapo and Kumar (2013) strategy which considered a LP model with non-negative coefficients, and developed an iterative procedure to solve a large-scale LP by transforming the given 'n' variable LP to a '2' variable LP. The new method developed in Section 7.3 reconsidered a similar large-scale LP model with non-negative coefficients and developed a new strategy that is an iterative hybrid process. The approach uses the conventional simplex iterations for search along the extreme points of the convex region; generates an interior point using these extreme points and moves from the interior point in the direction of the normal to the given objective function hyper-plane. The new procedure calculates a scalar $\alpha \geq 0$ that is used to identify a new location with an improved Z value. If $\alpha = 0$, it means that one has reached to the end of the feasible space in the normal direction to the objective function. If an optimal solution has not been identified, one has to change the search direction, which is easily achieved by the simplex method. The new point identified will always be an extreme point of the convex space. In other words, the problem defined has a non-void feasible region. The approach concludes the search when an optimal solution has been identified. Computational experiments indicated that the approach performed better with regard to a large number of randomly generated large LP problems.

Section 7.4 further improved the procedure discussed in Section 7.3 by removing the restriction of non-negative coefficients. A general large-scale LP has been considered without any restriction on the coefficients, since in many real-life applications of the LP model, the condition of non-negative coefficients may not always be satisfied.

In Section 7.5 a new heuristic for a mixed integer program using the characteristic equation was formulated. This new method is a hybrid of the existing approaches for solving the LP and PIP models, and it has been developed to solve MIP problems. The characteristic equation which was defined as a mapping of the integer hyper-plane on feasible integer points and the mapping of the objective function on interior integer points in a descending order, was used to obtain the LB. The proposed new method has several advantages over existing techniques, some of which include, eradication of round off errors since rounding leads to non-optimal or non-feasible solutions; there are no sub-problems as is in the case of branch and bound or branch and cut methods; and the approach is also, suitable for changes in the input values. The proposed approach guarantees determination of a feasible solution, and gives its bounds. The new approach also works better when integer restricted variables are relatively large.

Chapter 8

Summary, Conclusion and Recommendations

After all is said and done, more needs to have been done than said.

Neil Mason

8.1 Summary and Conclusion

This thesis has managed to modify and develop eight new techniques of solving different types of problems in line with the principal aim of modifying and developing new OR techniques that can be used to solve emerging problems encountered in the areas of linear programming, integer programming, mixed integer programming, network routing and travelling salesman problems. Five of the new models were in the section of network models and the other three are in the section of resource allocation and distribution models.

A new minimum weight labelling method for determination of the shortest route in a non-directed network was formulated. The major contribution of this method for determining a shortest route in a non-directed network is that, for an m-node network, the algorithm finds an optimal solution in at most $(m-1)$ iterations. For large networks, this method is likely to perform better than the traditional methods because of its convergence property. A few important and desirable investigations for further research could be:

1. Develop an appropriate software for the proposed method.
2. Solve many large randomly generated problems and compare with existing methods.
3. Investigate if ideas can be extended to generate ordered optimal solution, as they have applications in disaster management.

If the best shortest path cannot be used, then the second best can be implemented. Xu et al. (2012), went further to evaluate the K shortest paths in a schedule-based network, an algorithm that has several applications in computer science.

Another model that was formulated was that of the calculation of maximum reliability in both directed and non-directed networks. In the case of a non-directed network, the order of label indicated that the path reliabilities are in non-increasing order. Since virtual directions are dependent on labels, the formulated approach can be used for the determination of all reliability paths from a given node to all other nodes in that network. Using these virtual directions, a labelling technique was developed and illustrated. Information recycling is useful for protean networks. The protean system deals with changes in the model and recycling deals with extracting information that may be available from the system before occurrence of a change. In waste management, recycling reduces the bulk of solid waste and provides cheap resource to industry. Similarly, information recycling is intended to minimise unnecessary computations when that information can be extracted by earlier computations. These situations can arise also in reliability networks and when possible, one should take advantage.

Traditionally directed networks are relatively easy to analyse compared to non-directed networks as directions have inbuilt additional information that has been exploited from time to time in various forms. Using the formulated maximum reliability technique, we have attempted to use other properties of the given network and identified virtual directions based on those other properties of the given network. We used those virtual directions to establish a labelling method when link weights are deterministic values representing cost, distance or time. Similarly, in a probability network where link weights are represented by probabilities, the network has been analysed for directed and non-directed networks for finding the maximum reliability and the route in these reliability networks. Since the proposed method concludes in $n-1$ iterations, where n represents the number of nodes in the given network, the computational requirement remains under control, even for the non-directed network.

The concept of identifying virtual directions is a challenge which is worth further investigations for other variants of routing problems, and this will be the subject of subsequent investigations.

Two algorithms which are key in solving some of the NP-hard problems like the TSP were developed. The algorithms are; MST with index less than or equal to 2, and routing through $'k'$ specified nodes. The key point of these algorithms was to reduce the node index n_i to a number which is less than or equal to 2. The underlining theorems that enable the node index to be changed were presented. A MST path was defined and its applications were highlighted. Alternative interpretation of the MST-path is a shortest route passing through all the nodes. Numerical examples that illustrate the two algorithms were presented, and the results were found to be in line with the results obtained by other researchers. The route through $'k'$ specified nodes algorithm was also formulated in this thesis. The requirement for a path to pass through $'k'$ specified nodes arises when one may be interested in either saving a separate trip to the given specified node or attempting to take care of a future eventuality that is likely to arise in that situation. The complexity of this problem depends on the number of specified nodes. This problem has several applications in real life, which include the TSP and the Canadian traveller's problem, all of which have several applications in real-life.

A heuristic for the TSP based on the MST technique was also developed. The index value plays a major role in the proposed approach. The proposed approach is best when the network has at least one node with a low index value. If the number of lowest index value is m, then the number of sub-problems solved will be given by ${}^{m}C_2$. In the completely connected n node network, the worst case will have ${}^{(n-1)}C_2$ combinations. The proposed heuristic converts the problem in three parts to establish an upper bound. The approach discussed used link-weight modification to obtain alternative MSTs, which eventually have a TST interpretation. The proposed heuristics was compared to some well-known algorithms, and it produced better solutions for problems dealing with networks with smaller number of nodes.

Two new approaches that can be used to solve the transportation and assignment problems were developed. The major advantage of these two techniques is that they can handle the problem of degeneracy without any special consideration. The algorithm for the unified approach to solve transportation and assignment problems fully exploits the sub-problem's structure and has very favourable re-optimisation capabilities, both these properties are necessary for achieving optimality. The unified approach, which is a modification of the Hungarian method is applicable to both the assignment and transportation problems. Furthermore, the process does not depend on the number of allocated cells, which in transport method must be equal to $(m + n - 1)$ in independent cells. Thus, the proposed unified method is efficient in solving all degenerate transportation models. This approach is free of pivotal degeneracy which may cause cycling and does not require any extra variables such as slack, surplus or artificial variables that are used in dual and primal simplex methods. The generalised assignment problem (GAP), deals with assigning a set of n items to a set of m knapsacks, where each item must be assigned to exactly one knapsack and there are constraints on the availability of resources for item assignment. In the method proposed in this thesis, the GAP was relaxed to become an ordinary transportation problem by replacing the ordinary constraints with inequalities obtained by solving the knapsack problem. In the new method the current solutions to the transportation problem are used as starting solutions in the next iterations. With this approach it is only possible to branch if the relaxation gives an integer optimal solution, and this is not possible with LP or Lagrangian relaxations. The GAP is a classical combinatorial optimisation problem that models a variety of real world applications including flexible manufacturing systems, facility location and vehicle routing problems. The GAP is known to be NP-hard, since the partition problem of a given set of positive integers into two equal sized subsets can be reduced to GAP with m=2 knapsacks.

The last chapter of the thesis looked at two LP based methods for solving NP-hard problems. The first method developed an iterative procedure to solve a large-scale LP by transforming the given 'n' variable LP to a '2' variable LP. The new method developed in this thesis reconsiders a similar large-scale LP model with non-negative coefficients and developed a new strategy that is an iterative hybrid process. The approach uses the conventional simplex iterations for search along the extreme points of the convex region, generates an interior point using these extreme points, and moves from the interior point in the direction of the normal to the given objective function hyper-plane. The new procedure calculates a scalar $\alpha \geq 0$ that is used to identify a new location with an improved Z value. If $\alpha = 0$, it means one has reached to the end of the feasible space in the normal direction to the objective function. If an optimal solution has not been identified, one has to change the search direction, which is easily achieved by the simplex method. We further improved the procedure by removing the restriction of non-negative coefficients and came up with a general large-scale LP without any restriction on the coefficients since in many real-life applications of the LP model, the condition of non-negative coefficients may not always be satisfied.

It will be desirable to develop an appropriate software and compare the proposed method with existing methods for many randomly generated problems.

Finally, a new heuristic for a mixed integer program using the characteristic equation was formulated. This new method is a hybrid of the existing approaches for solving the LP and PIP models and it has been developed to solve MIP problems. The characteristic equation which was defined as a mapping of the integer hyper-plane on feasible integer points and the mapping of the objective function on interior integer points in a descending order was used to obtain the lower bound. The proposed new method has several advantages over existing techniques which include the eradication of round off errors since rounding leads to non-optimal or non-feasible solutions. The new procedure does not create sub-problems as is in the case with branch and bound or branch and cut methods, and the approach is also suitable for changes in the input values. The proposed approach guarantees determination of a feasible solution and also gives its bounds. The approach is likely to work better when integer-restricted variables are relatively large in number.

8.2 Recommendations

This thesis recommends that as the global environment is gradually changing, there is need also to continuously come up with new techniques of solving emerging problems. The new techniques can be formed by developing, combining and modify existing models in order to encounter these challenges in terms of the speed at which the solution is obtained, or in terms of the costs of obtaining the solutions. The development of software packages to the proposed algorithms is key especially if we are to test them on larger problems and compare them with existing methods.

8.3 Considerations for Further Studies

Most of the techniques that have been developed in this thesis need to be programmed so that their comparisons with existing techniques can be compared at all levels of complexity. It was observed that all the new techniques developed are performing comparatively well with existing methods. For them to be effectively compared in solving large problems, computational experiments of all the new techniques must be carried out. This aspect has been left as further studies to these techniques.

Bibliography

ABRAHAM, I., FIAT, A., GOLDBERG, A. V., AND WEMEEK, R. F. (2010). Highway dimension, Shortest paths, and Provably Efficient Algorithms. *In Proceedings of the twenty-first annual ACM-SIAM Symposium on Discrete Algorithms*, 782–793.

ADLAKHA, V. AND ARSHAM, H. (1998). Managing cost uncertainties in transportation and assignment problems. *Journal of Applied Mathematics*, **2** (1), 65–104.

AGBADUDU, A. B. (2006). *Operations research, mathematics and social sciences: the link*. University of Benin, volume 86 of University of Benin inaugural lecture series.

AGGARWA, P. AND KHETRAPAL, P. (2014). Study on Utilization of Magnetic Levitation in 3-D Bio-Printing. *International Journal of Innovative Science, Engineering and Technology*, **1** (9), 5–7.

AHUJA, R. K., MAGNANTI, T. L., AND ORLIN, J. B. (1993). Network Flows: Theory, Algorithms and Applications. *Prentice Hall*, 210–212.

AHUJA, R. K., MEHLHORN, K., ORLIN, J. B., AND TARJAN, R. E. (1990). Faster algorithms for the shortest path problem,. *Journal of the Association of Computing Machinery*, **37**, 213–223.

AKGUN, I. AND TANSEL, B. C. (2010). Min-degreed constrained minimum spanning tree problem: New formulation via Miller-Tucker-Zemlin constraints. *Computers and Operations Research*, **37** (1), 72–82.

ALAEI, S., HAJIAGHAYI, M. T., AND LIAGHAT, V. (2012). The Online Stochastic Generalized Assignment Problem. *In FOCS*, http://www.cs.cornell.edu/saeed/archive/onlinestochasticgap.pdf.

ANUPAM, G. (2015). Deterministic Minimum Spanning Trees. *Advanced Algorithms, CMU, Spring*, 1–6.

APPLEGATE, D. L., BIXBY, R. E., CHVATAL, V., AND COOK, W. L. (2006). *The Travelling Salesman Problem: A Computational Study*. Princeton University Press.

ARORA, H. AND KUMAR, S. (1993). Path through k-specified edges in a linear graph. *Journal of Operations Research*, **30** (2), 163–173.

ARORA, S. AND BARAK, B. (2009). *Computational Complexity: A Modern Approach*. Cambridge University Press.

BAES, M., OERTEL, T., AND WEISMANTEL, R. (2016). Duality for mixed-integer convex minimization. *Mathematical Programming*, **8** (1), 9–17.

BAJAJ, C. P. (1970). Some constrained shortest route problems. *Unternehmensforschung*, **15** (1), 287–301.

BAKER, T. P., GILL, J., AND SOLOVAY, R. (1975). Relativizations of the P = NP question. *SIAM Journal of Computing*, **4** (4), 431–442.

BALINSKI, M. L. AND GOMORY, R. E. (1964). A Primal Method for the Assignment and Transportation Problems. *Management Science*, **10** (3), 578–593.

BALL, M. O. (1980). Complexity of network reliability computations. *Networks*, **10**, 153–165.

BALSEIRO, S. R., LOISEAU, I., AND RAMONET, J. (2011). An ant colony algorithm hybridized with insertion heuristics for the Time Dependent Vehicle Routing Problem with Time Windows. *Computers and Operations Research*, **38** (6), 954–966.

BANASAL, S. P. AND KUMAR, S. (1977). Maximum reliability route subject to M-improvements in a directed graph. *IEEE Trans Reliab R*, **26** (2), 103–105.

BECKWITH, R. E. (1961). Dynamic Programming and network routing, an introduction to the technique of functional equations. *Proc. of the Dynamic Programming workshop, Boulder, Colorado*.

BELLMAN, R. E. (1958). On an optimal routing problem. *Q. Appl. Math.*, **16**, 87–90.

BELLMAN, R. E. AND DREYFUS, S. E. (1962). *Applied Dynamic Programming*. Princeton University Press.

BERMAN, P. AND KARPINSKI, M. (2006). 8/7- approximation algorithm for the (1,2)- TSP, Proc. *17th ACM-SIAM SODA conference*, 641–648.

BERNING, T. L. (2010). *Improved estimation procedures for a positive extreme value index*. PhD Thesis, SUNScholar, Stellenbosch University, Cape Town.

BERTACCO, L., FISCETTI, M., AND LODI, A. (2007). A feasibility pump heuristic for general mixed-integer problems. *Discrete Optimisation*, **4**, 63–76.

BIXBY, R. E., GREGORY, J. W., LASTIG, I. J., MARSTEN, R. E., AND SHANNO, D. F. (1991). Very Large-Scale Linear Programming: A case study in combining Interior Point and Simplex Methods. *TR91-11*, 1–18.

BORIA, N. AND PASCHOS, V. T. (2010). Fast Re-optimization for the Minimum Spanning Tree problem. *Journal of Discrete Algorithms*, **8** (3), 296–310.

BRITTON, J. K. (1979). Integer solution of system of quadratic equations. *Mathematical proceedings of the Cambridge Philosophical Society*, **86**, 385–389.

CAGGIANI, L., MAURO, D., MARINELLI, M., AND OTTOMANELLI, M. (2012). A meta-heuristic dynamic traffic assignment model for O-D matrix estimation using aggregated data:. *Procedia-Social and Behavioral Sciences*, Available at www.Sciencedirect.com.

CALVETE, H. I., GALE, C., OLIVEROS, M. J., AND SANCHEZ-VALVERDE, B. (2007). A goal programming approach to vehicle routing problems with soft time windows. *European Journal of Operational Research*, **177** (3), 1720–1733.

CARLSON, J., JAFFE, A., AND WILES, A. (2006). The Millennium Prize Problems. Providence, RI. *American Mathematical Society and Clay Mathematics Institute*, ISBN 978–0–8218–3679–8.

CASCETTA, E. AND POSTORINO, M. N. (2001). Fixed Point Approaches to the Estimation of O/D Matrices Using Traffic Counts on Congested Networks. *Transportation Science*, **35**, 134–147.

CHAGWIZA, G., JONES, B. C., HOVE-MUSEKWA, S. D., AND MTISI, S. (2015). A generalised likelihood uncertainty estimation mixed-integer programming model: application to a water resource distribution network. *Cogent Mathematics*, **2**, 1048076.

CHANDRASAKARAN, R., KABADIE, S. N., AND MURTY, K. G. (1982). Some NP-complete problems in linear programming. *Operations Research Letters*, **1**, 101–104.

CHAZELLE, B. (2000). A Minimum Spanning Tree Algorithm with Inverse-Ackermann Type Complexity. *Journal of the ACM*, **47** (6), 1028–1047.

CHENKASSKY, B. V., GOLDBERG, A. V., AND RADZIK, T. (1996). Shortest paths algorithms: Theory and Experimental Evaluation, Mathematical Programming. *Ser. A DOI1016/0025-5610 (95) 00021-6, MR 1392160.*, **73** (2), 129–174.

CHRISTOFIDES, N. (1976). *Worst case analysis of a new heuristic for the traveling salesman problem*. Graduate School of Industrial Administration, Carnegie-Mellon University, Pitts-burgh, PA.

CICKOVA, Z., BREZINA, I., AND PEKAR, J. (2008). Alternative Method for Solving Traveling Salesman Problem by Evolutionary Algorithm. *Management Information Systems*, **3** (2008), 017–027.

CODATO, G. AND FISCHETTI, M. (2006). Combinatorial Benders' Cuts for Mixed-Integer Linear Programming. *Operatiuons Research*, **54** (4), 756–766.

COLES, S. (2001). *An introduction to statistical modelling of extreme values*. Springer-Verlag, London.

COOK, W. L. (2010). *"History of the TSP". The Traveling Salesman Problem.* Georgia Tech.
URL: *http://www.tsp.gatech.edu/index.html*

COWEN, L. (2011). The travelling salesman problem, Lecture 4, Comp. 260. *Advanced Algorithms, Tufts University*, http://www.cs.tufts.edu/comp/260/Old/ lecture4.pdf.

DANTZIG, G., FULKERSON, D., AND JONJSON, S. (1954). Solutions of a large-scale traveling salesman problem. *Operations Research*, **2** (4), 393–410.

DANTZIG, G. B. (1960). On the shortest route through a network. *Manag Sci*, **6** (2), 187—-190.

DE ANDRADE, R. C. (2013). Elementary shortest-paths visiting a given set of nodes. *In Simposio Brasileiro de Pesquisa Operacional*, 16–19.

DENEGRE, S. (2011). *Interdiction and discrete bi-level linear programming*. PhD thesis, Lehigh University.

DEVLIN, K. J. (2003). *The Millennium Problem: The Seven Greatest Unsolved Mathematical Puzzles of our time*. New York: Basic Books. ISBN 0-465-01729-0.

DIJKSTRA, E. W. (1959). A note on two problems in connection with graphs. *Numer. Math.*, **1**, 269—-271.

DORINGO, M. AND GAMBARDELLA, L. M. (1997). Ant colonies for the travelling salesman problem. *Biosystems*, **43** (2), 73–81.

ELSAYED, S. M., SARKER, R. A., AND ESSAM, D. L. (2011). Multi-operator based evolutionary algorithms for solving constrained optimization problems. *Computers and Operations Research*, **38** (12), 1877–1896.

FASIKA, B. G., KLAUS-DIETER, T., AND MARCUS, S. (2013). Implementing the SCOR Model Best Practices for Supply Chain Improvement in Developing Countries. *International Journal of u- and e-Service, Science and Technology*, **6** (4), 13–27.

FAZIO, P., RANGO, F. D., AND SANTAMARIA, C. S. A. F. (2013). Routing optimization in vehicular networks: A new approach based on multi-objective metrics and minimum spanning tree. *International Journal of Distributed Sensor Networks*, **2013**, http://dx.doi.org/10.1155/2013/598675.

FERNANDEZ, E., GARFINKEL, R., AND ARBIOL, R. (1998). Mosaicking of aerial photographic maps via seams defined by bottleneck shortest paths. *Operations Research*, **46** (3), 293–304.

GAO, W. F. AND LIU, S. Y. (2012). A modified artificial bee colony algorithm. *Computers and Operations Research*, **39** (3), 687–697.

GARCIA-NAJERA, A. AND BULLINARIA, J. A. (2011). An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Computers and Operations Research*, **38** (1), 287–300.

GAREY, M. R. AND JOHNSON, D. S. (1979). *Computers Intractability: A guide to the theory of NP-Completeness*. San Francisco, W. H. Freeman.

GARG, D. AND SHAH, S. (2011). Ant Colony Optimization for Solving Traveling Salesman Problem. *International Journal of Computer Science and System Analysis*, **5** (1), 23–29.

GARG, R. C. AND KUMAR, S. (1968). Shortest Connected Graph through dynamic programming technique. *Math's Magazine, USA*, **41** (4), 170–173.

GARG, R. C., KUMAR, S., DASS, P., AND SEN, P. (1968). Generalised travelling salesman problem through N sets of nodes in a competitive market. *AKOR, Ahlanf and Planning for schung, Band II, Heft 2*, 170–173.

GAVISH, B. AND PIRKUL, H. (1986). Computer and database location in distributed computer systems. *IEEE Transactions on Computers*, **35**, 583–590.

GEOFFRION, A. M. AND MARSTEN, R. E. (1972). Integer programming algorithms: A framework and state-of-the art survey. *Management Science*, **18** (9), 465–491.

GHARAN, S. O., SABERI, M., AND SINGH, M. (2012). *A Randomized Rounding Approach to the Traveling Salesman Problem*. Stanford University.

GNEDENKO, B., PAVLOV, I., AND USHAKOV, I. (1999). *Statistical Reliability Engineering*. John Wiley and Sons, New York.

GOMES, T., MARQUES, S., MARTINS, L., PASCOAL, M., AND TIPPER, D. (2015). Protected Shortest Path visiting specified nodes. *Conference paper*. **URL:** *https://www.researchgate.net/publication/283270344*

GRADISAR, M. AND TRKMAN, P. (2005). A Combined Approach for Solution of General One-Dimensional Cutting Stock Problem. *Computers and Operations Research*, **32** (7), 1793–1807.

GRAHAM, R. L. AND HELL, P. (1985). On the History of the Minimum Spanning Tree. *Annals of the History of Computing*, **7** (1), 43–57.

GRIFFEN, R. (1987). *Management*. Boston: Houghton Mifflin Company.

GUTIN, G. AND PUNNEN, A. P. (2006). *The Travelling Salesman Problem and its Varients*. Springer.

HAMMER, M. AND CHAMPY, J. (1993). *Re-engineering the Corporation*. Harper Collins, New York.

HASTINGS, N. A. J. (1973). *Dynamic Programming with Management Applications*. Butterworth Press.

HE, X., ZHENG, H., AND PEETA, S. (2015). Model and a Solution Algorithm for the Dynamic Resource Allocation Problem for Large-Scale Transportation Network Evacuation. *Transportation Research Procedia*, **7**, 441–458.

HILLIER, F. S. AND LIEBERMAN, G. J. (2001). *Introduction to Operations Research*. McGraw Hill, New York.

HLAING, Z. C. S. AND KHINE, M. A. (2011). Solving Traveling Salesman Problem by using Improved Ant Colony Optimization Algorithm. *International Journal of Information and Education Technology*, **1** (5), 404–409.

HUNG, M. S., ROM, W. O., AND WAREN, A. D. (1986). Degeneracy in Transportation Problems. *Discrete Applied Mathematics*, **13**, 223–237.

IBARAKI, T. (1973). Algorithms for obtaining shortest paths visiting specified nodes. *SIAM Review*, **15** (2), 309–317.

ISHII, H. AND MATSUTOMI, T. (1995). Confidence Regional Method of Stochastic Spanning Tree Problem. *Mathl.Comput. Modelling*, **22** (10-12), 77–82.

JAYAWANT, P. AND GLAVIN, K. (2009). Minimum Spanning Trees. *INVOLVE*, **2** (4), 439–450.

JINN-TSONG, T., JIA-CEN, F., AND JYH-HORNG, C. (2013). Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. *Computers and Operations Research*, **40** (12), 3045–3055.

JONES, E. M. B. C. AND KUMAR, S. (2008). A minimum incoming weight label method and its application in CPM networks. *Orion*, **24** (1), 37–48.

KACHIYAN, L. G. (1979). A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, **20**, 191–194.

KAHNG, A. B. AND REDA, S. (2004). Match Twice and Stitch: A New TSP Tour Construction Heuristic. *Operations Research Letters*, **32** (6), 499–509.

KAIBEL, V. AND PEINHARDT, A. F. M. (2006). On the bottleneck shortest path problem. *ZIB-Report 06-22, Konrad-Zuse-Zentrum für Informationstechnik Berlin*.

KALABA, R. (1960). On some Communication Network Problems, Combanatorial Analysis, Proc. Symp. *Applied Mathematics*, **10**, 261–280.

KARGER, D., MOTWANI, R., AND RAMKUMAR, G. D. S. (1997). Approximating the longest path in a graph. *Algorithmica*, **18** (1), 82–98.

KARMARKAR, N. (1984). A new polynomial algorithm for linear programming. *Combinatorica*, **4**, 373–395.

KHACHIYAN, L. (1982). Convexity and Algorithmic solving polynomial programming problems. *Technitcheskaya Kibernetika, (in Russian)*, **6**, 47–56.

KIRGY, M. W. (2007). Paradigm Change in Operations Research: Thirty years of debate. *Operations Research*, **55** (1), 1–13.

KONIDARI, P. AND MAVRAKIS, D. (2007). A multi-criteria evaluation method for climate change mitigation policy instruments. *Energy Policy*, **35** (12), 6235–6257.

KRUSKAL, J. B. (1956). On the Shortest Spanning Sub-tree of a graph and the Travelling Salesman Problem. *Proc. Amer. Math Soc.*, **7**, 48–50.

KUHN, H. W. (1955). The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, **2** (1955), 83–97.

KUMAR, B. R. AND MURUGESAN, S. (2012). On fuzzy transportation problem using triangular fuzzy numbers with modified revised simplex method. *International Journal of Engineering Science and Technology*, **4** (1), 285–294.

KUMAR, S. (2005). Information recycling mathematical methods for protean systems: A path-way approach. *South African Journal of Industrial Engineering*, **16** (2), 81–101.

KUMAR, S. (2006). Information recycling mathematical method: A path-way approach to Geometric Programs. *South African Journal of Industrial Engineering*, **17** (2), 127–143.

KUMAR, S., ARORA, H., AND BAPPOO, R. (1999). Kth best solution of a system. *J Appl Cybern SCIMA*, **26** (1), 3–19.

KUMAR, S. AND BAPOO, R. (1999). Deployment of the refuse vehicles in protean environment. *Opsearch*, **36** (3), 242–259.

KUMAR, S. AND MUNAPO, E. (2012). Some lateral ideas and their applications for developing new solution procedures for a pure integer programming model. *Proceedings of the Heber International Conference on Applications of Mathematics and Statistics*, 13–21.

KUMAR, S., MUNAPO, E., AND JONES, B. C. (2007). An integer equation controlled descending path to a protean pure integer program. *Indian Journal of Mathematics*, **49**, 211–237.

KUMAR, S., MUNAPO, E., LESAOANA, M., AND NYAMUGURE, P. (2014). A Minimum Spanning Tree Approximation to the Routing Problem through 'k' Specified Nodes. *J Economics*, **5** (3), 307–312.

KUMAR, S., MUNAPO, E., LESAOANA, M., AND NYAMUGURE, P. (2016a). Identification and Application of Virtual Directions in a Non-Directed Network: A Labelling Method for Determination of Maximum Reliability and the Route. *Communications in Dependability and Quality Management, An International Journal*, **19** (1), 85–95.

KUMAR, S., MUNAPO, E., LESAOANA, M., AND NYAMUGURE, P. (2016b). *Is the travelling salesman actually NP-hard?. Chapter 3 in Engineering and Technology: Recent Innovations and Research (An International Edition)*. International Research Publication House. ISBN-978-93-86138-06-4.

KUMAR, S., MUNAPO, E., LESAOANA, M., AND NYAMUGURE, P. (2017a). A minimum spanning tree based heuristic for the travelling salesman problem. *Revised manuscript submitted to Opsearch*.

KUMAR, S., MUNAPO, E., LESAOANA, M., AND NYAMUGURE, P. (2017b). Strategies for reducing feasible space and the number of variables for solving a lp model. *To appear in International Journal of Mathematical, Engineering and Management Sciences*.

KUMAR, S., MUNAPO, E., NCUBE, O., SIGAUKE, C., AND NYAMUGURE, P. (2013). A minimum weight labelling method for determination of a shortest route in a non-directed network. *Int. J. Syst. Assur. Eng. Manag 4(1):13–18*, **4** (1), 13–18.

LEFEVER, W., AGHEZZAF, E., AND HAMOU, K. H. (2016). A convex optimization approach for solving the single-vehicle cyclic inventory routing problem. *Computers and Operations Research*, **72** (2016), 97–106.

LEITCH, R. D. (1995). *Reliability Analysis for Engineers: An introduction*. Oxford University Press, New York.

LESAJA, G. (2009). Introducing Interior-Point Methods for Introductory Operations Research Courses and/or Linear Programming Courses. *The Open Operational Research Journal*, **3**, 1–12.

LI., J. AND GUO, Y. H. (2010). *Theories and Methods in Optimization of Logistic Distribution*. China Material Press.

LI, M., YANG, S., ZHENG, J., AND LIU, X. (2014). A euclidean minimum spanning tree-based evolutionary algorithm for multi-objective optimisation. *Evolutionary Computation*, **22** (2), 189–230.

LI, P., HE, H., ZHENG, D., HUANG, Y., AND FAN, C. (2008). Vehicle Routing Problem with soft time windows based on improved genetic algorithm for fruits and vegetables distribution. *Discrete Dynamics in Nature and Society*, **2015**, http://dx.doi.org/10.1155/2015/483830.

MAGEE, J. F. (1973). Progress in the Management Sciences. *TIMS, Interfaces*, **3**, 59–61.

MALPANI, N. AND CHEN, J. (2002). A note on practical construction of maximum bandwidth paths. *Information Processing Letters*, **83** (3), 175–180.

MANGAN, J., LALWANI, C., AND GARDNER, B. (2004). Combining quantitative qualitative methodologies in logistic research. *International Journal of Physical Distribution and Logistic Management*, **34** (7), 565–578.

MAVROTAS, G. AND DIAKOULAKI, D. (1998). A branch-and-bound algorithm for mixed zero-one multiple objective linear programming. *Eur.J. Oper. Res.*, **107** (3), 530–541.

MILLER, C., TUCKER, A., AND ZEMLIN, R. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM*, **7**, 326–329.

MITCHELL, J. E. (2001). *Branch and cut algorithms for integer programming: In Christodous, A. Floudas and M. Panos and M. Pardalos (eds.), Encyclopedia of Optimization*. Kluwer Academic Publishers.https:// homepages.rpi.edu mitchj bioJEM.pdf.

MUNAPO, E., JONES, B. C., AND KUMAR, S. (2008). Minimum incoming weight label method and its application in CPM networks. *ORiON*, **24** (1), 37–48.

MUNAPO, E. AND KUMAR, S. (2013). Solving large-scale linear optimising problems with non-negative coefficients by transforming into 2-variable LP problem. *The Australian Society for Operations Research (ASOR) Bulletin*, **32** (1), 1–12.

MUNAPO, E., KUMAR, S., LESAOANA, M., AND NYAMUGURE, P. (2014). Solving a large-scale LP model with non-negative coefficients: A hybrid search over the extreme points and the normal direction to the given objective function. *The Australian Society for Operations Research (ASOR) Bulletin*, **33** (1), 11–24.

MUNAPO, E., KUMAR, S., LESAOANA, M., AND NYAMUGURE, P. (2016). A Minimum Spanning Tree with node index less than or equal to 2. *The Australian Society for Operations Research (ASOR) Bulletin*, **34** (1), 1–14.

MUNAPO, E., KUMAR, S., AND MUSEKWA, S. D. (2010). A revisit to generalized assignment problem: An ascending hyper-plane approach through network flows. *International Journal of Management Prudence*, **2** (1), 7–14.

MUNAPO, E., LESAOANA, M., NYAMUGURE, P., AND KUMAR, S. (2015). A transportation branch and bound algorithm for solving the generalised assignment problem. *International journal of System Assurance Engineering and Management*, **6** (3), 217–223.

MUNAPO, E., NYAMUGURE, P., LESAOANA, M., AND KUMAR, S. (2012). A note on unified approach to solving the transportation and assignment models in operations research. *International Journal of Mathematical Modeling, Simulation and Applications*, **5** (2), 140–149.

NADEF, D. (2002). Polyhedral theory and branch and cut algorithms for the symmetric TSP: In G. Gutin and G. Punnen (eds). *Kluwer, Dorcdrecht*, 29–116.

NAUSS, R. N. (2004). The elastic generalized assignment problem. *Journal of the Operations Research Society*, **55** (12), 1333–1341.

NESETRIL, J. AND NESETRILOVA, H. (2010). The origins of Minimum Spanning Tree Algorithms-Boruvka and Jarnk. *Documentra Mathematica: Extra volume ISMP*, 127–141.

NEWMAN, M. (2002). The spread of epidemic diseases on networks. *V-Physical Review E*, **66** (1), 1551–1555.

NYAMUGURE, P., MAPOSA, D., SIGAUKE, C., CHIYAKA, E., AND MUNJERI, D. (2011). A holistic application of process capability indices. *African Journal of Business Management*, **5** (28), 11413–11424.

NYAMUGURE, P., MUNAPO, E., LESAOANA, M., AND KUMAR, S. (2016). A heuristic for mixed integer program using the characteristic equation approach. *International Journal of Mathematical, Engineering and Management Sciences*, **2** (1), 1–16.

O'CONNOR, P. D. T. (2002). *Practical Reliability Engineering, Forth Edition*. John Wiley and Sons, New York.

PADBERG, M. AND RINALDI, G. (1991). A branch and cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *Siam Review*, **33** (1), 60–100.

PAPADIMITRIOU, C. H. AND VEMPALA, S. (2006). On the approximability of the traveling salesman problem. *Combinatorica*, **26** (1), 29–116.

PENTICO, D. W. (2007). Assignment Problem: A Golden Anniversary Survey. *European Journal of Operation Research*, **176**, 774–793.

PEPPINO, F., FLORIANO, R., CESARE, S., AND AMILCARE, F. S. (2013). Routing Optimization in Vehicular Networks: A New Approach Based on Multiobjective Metrics and Minimum Spanning Tree. *International Journal of Distributed Sensor Networks*, http://dx.doi.org/10.1155/2013/598675.

POLLACK, M. AND WEIBENSON, W. (1960). Solution of the shortest route problem-A review. *Operations Research*, **8**, 224–230.

POLYAKOVSKIYA, S., BERGHAMMERB, R., AND NEUMANNA, F. (2016). Solving hard control problems in voting systems via integer programming. *European Journal of Operational Research*, **250** (2016), 204–213.

PRIM, R. C. (1957). Shortest Connection Networks and Some Generalizations. *Bell System Tech. J.*, **36**, 1389–1401.

QING-DAO-ER-JI, R. AND WANG, Y. (2012). A new hybrid genetic algorithm for job shop scheduling problem. *Computers and Operations Research*, **39** (10), 2291–2299.

RAINWATER, C., GEUNES, J., AND ROMEIJN, H. E. (2009). The generalized assignment problem with flexible jobs. *Discrete Applied Mathematics*, **157**, 49–67.

RAJGOPAL, J. (2004). *Principles and Applications of Operations Research*. Maynard's Industrial Engineering Handbook, 5th Edition.

RAVASZ, E., SOMERA, A. L., MONGRU, D. A., OLTVAI, Z. N., AND BARABÁSI, A. L. (2002). Hierarchical organization of modularity in metabolic networks. *Science*, **297** (5586), 1551–1555.

RAZALI, N. M. AND GERAGHTY, J. (2011). Genetic Algorithm Performance with different Selection Strategies in Solving TSP. *Proceedings of the World Congress on Engineering, July 6-8, Londan, UK*.

RESOVSKY, M., GROF, M., HORVATH, D., AND GAZDA, V. (2014). Revealing ownership structure of funds using minimum spanning tree. *ACTA PHYSICA POLONICA A*, **126** (6), 1322–1326.

RIGHIN, G. AND SALANI, M. (2008). New Dynamic Programming Algorithms for resource constrained elementary shortest path problem. *Networks*, **51** (3), 155–170.

ROBERTI, R. AND TOTH, P. (2012). Models and algorithms for Asymmetric Traveling Salesman Problem: An experimental comparison. *EURO Journal on Transportation and Logistics*, **1** (1), 113–133.

ROOS, C., TERLAKY, T., AND VAL, J. (2005). Interior point methods for Linear Optimization. *Springer*.

RUIZ, E., ALBAREDA-SAMBOLA, M., FERNANDEZ, E., AND RESENDE, M. G. (2015). A biased random key genetic algorithm for the capacitated minimum spanning tree problem. *Computers and Operations Research*, **57**, 95–108.

SAKSENA, J. P. AND KUMAR, S. (1966). The routing problem with 'k' specified nodes. *Operations Research*, **14** (5), 909–913.

SANDERS, P. (2009). Fast route planning. *Google Tech Talk. http://www.youtube.com/watch v=-0ErpE8tQbw*.

SARANYA, S. AND VAIJAYANTHI, R. P. (2014). Traveling Salesman Problem Solved Using Bio Inspired Algorithms (ABC). *International Journal of Innovative Research in Computer and Communication Engineering*, **2** (1), 223–237.

SCHRIJVER, A. (2003). *Combinatorial Optimisation Polyhedra and Efficiency*. Springer.

SCHULZE, M. (2011). A new monotonic, clone-independent, reversal symmetric, and Condorcet-consistent single-winner election method. *Social Choice and Welfare*, **36** (2), 267—303.

SEDGEWICK, R. AND WAYNE, D. K. (2011). *Algorithms (4th ed.)*. Addison-Wesley Professional ISBN 9780321573513.

SEN, S. AND SHERALI, H. D. (2006). Decomposition with branch-and-cut approaches for two stage stochastic mixed-integer programming. *Mathematical Programming 106(2), 203 223*, **106** (2), 203—-223.

SHACHAM, N. (1992). Multicast routing of hierarchical data. *IEEE International Conference on Communications (ICC '92)*, **3**.

SHERALI, H. D. AND ADAMS, W. P. (1994). A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems. *Discrete Applied Mathematics*, **52**, 83–106.

SHIVANI, S., LEEN, J., AND BHARTI, K. (2013). Shortest Path Algorithm. *International Journal of Application or Innovation in Engineering and Management (IJAIEM)*, **2** (7), 316–320.

SHMOYS, B. D. AND TARDOS, E. (1993). An approximation algorithm for the generalized assignment problem. *Journal of Mathematical Programming*, **62**, 461–475.

SINHA, S. M. AND BAJAJ, C. P. (1970). The Maximum Capacity Route through a set of Specified Nodes. *Operations Research*, **7**, 96–114.

SRIVASTAVA, S. S., KUMAR, S., GARG, R. C., AND SEN, P. (1969). Generalized Travelling Salesman Problem through 'K' sets of nodes. *Canadian Operations Research Society Journal*, **7** (2), 97–101.

STUTZLE, T. AND DORIGO, M. (1999). ACO Algorithms for the Traveling Salesman Problem. *Evolutionary Algorithms in Engineering and Computer Science*.

TOTH, P. AND VIGO, D. (2001). The vehicle routing problem. *SIAM monographs on discrete mathematics and applications, Philadelphia*.

TSAI, W., LEU, J., LIU, J., LIN, S., AND SHAW, M. (2010). A MCDM approach for sourcing strategy mix decision in IT projects. *Expert Systems with Applications*, **37** (5), 3870–3886.

ULLAH, E., LEE, K., AND HASSOUN, S. (2009). An algorithm for identifying dominant-edge metabolic pathways. *IEEE/ACM International Conference on Computer-Aided Design (ICCAD 2009)*, 144–150.

VARDHAN, H., BILLENAHALLI, S., HUANG, W., RAZO, M., SIVASANKARAN, A., TANG, L., MONTIY, P., TACCA, M., AND FUMAGALLI, A. (2009). Finding a Simple Path with Multiple Must-include Nodes. *Technical Report UTD/EE/2/2009*, 1–21.

VELASQUEZ, M. AND HESTER, P. T. (2013). An analysis of multi-criteria decision making methods. *International Journal of Operations Research*, **10** (2), 56–66.

VIDAL, T., CRAINIC, T. G., GENDREAU, M., AND PRINS, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers and Operations Research*, **40** (1), 475–489.

VIELMA., J. P. (2015). Mixed integer linear programming formulation techniques. *SIAM Review*, **57** (2015), 3–57.

VIELMA, J. P., MURRAY, A., RYAN, D., AND WEINTRAUB, A. (2007). Improving computational capabilities for addressing volume constraints in forest harvest scheduling problem. *European Journal of Operational Research*, **176** (2007), 1246–1264.

WANG, T. (2012). The interactive trade decision-making research: An application of novel hybrid MCDM model. *Economic Modelling*, **29** (3), 926–935.

WANG, Z., HUANG, D., MENG, H., AND TANG, C. (2013). A new and fast algorithm for solving the minimum spanning tree problem based on DNA molecules and computation. *BioSystems*, **114** (1), 1–7.

WILES, A. (2006). ”the Birch and Swinnerton-Dyer Conjecture”. In Carlson, J., Jaffe A., Wiles A.. The Millennium Prize Problems. *American Mathematical Society*, **ISBN 978-0-8218-3679-8.**, 31–44.

WILLIAMS, H. P. (2009). Logic and integer programming. *International Series in Operations Research and Management Science*, **130**.

WILSON, R. AND WATKINS, J. J. (1990). *An introduction approach*. Wiley, New York. MR91b:05001 Zbl 0712.05001.

WINSTON, W. L. AND VENKATARAMANAN, M. (2003). *Introduction to mathematical programming, 4th edn.* Thomson Brooks/Cole, Pacific Grove.

WOLSEY, L. A. (1980). Heuristics analysis, linear programming and branch and bound. *Mathematical Programming Study*, **13**, 121–134.

XIN, J., GUO, L., HUANG, N., AND LI, R. (2013). Network service reliability analysis model,. *Chemical Engineering Transactions*, **33**, 511–516.

XU, P. (2012). *Three essays on bi-level optimization algorithms and applications*. PhD thesis, Lowa State University.

XU, P. AND WANG, L. (2014). An exact algorithm for the bi-level mixed integer programming problem under three simplifying assumptions. *Computers and Operations Research*, **41** (1), 309–318.

YAGIURA, M., IBARAKI, T., AND GLOVER, F. (2004). An ejection chain approach for the generalized assignment problem. *Inf. Journal of Computing*, **16**, 133–151.

YAGIURA, M., IBARAKI, T., AND GLOVER, F. (2006). A path re-linking approach with ejection chains for the generalized assignment problem. *European Journal of Operations Research*, **169**, 548–569.

YANG, H., MENG, Q., AND BELL, M. G. H. (2001). Simultaneous estimation of the origin destination matrices and travel-cost coefficient for congested networks in a stochastic user equilibrium. *Transportation Science*, **35**, 107–123.

YAO, A. C. (1975). An O(log (log n)) algorithm for finding minimum spanning tree. *Inf. Process. lett.*, **4** (1975), 21–23.

YAO, A. C. (1982). On constructing minimum spanning tree in k-dimensional spaces and related problems. *SIAM J. Comp.*, **11** (1982), 721–736.

ZABEO, A., PIZZOL, L., AGOSTINI, P., CRITTO, A., GIOVE, S., AND MARCOMINI, A. (2011). Regional risk assessment for contaminated sites Part 1: Vulnerability assessment by multi-criteria decision analysis. *Environment International*, **37** (8), 1295–1306.

ZHAN, F. B. (1997). Three shortest path algorithms on real road networks: Data structures and procedures. *J. Geogr. Inf. Decis. Anal.*, **1** (1), 69–82.

ZHAN, F. B. AND NOON, C. E. (1998). Shortest path algorithms: An evaluation using real road networks. *Transp Sci*, **32** (1), 65–73.

ZHANG, Y. (1996). Solving Large-Scale Linear Programs by Interior-Point Methods under the MATLAB Environment. *Technical Report TR96-01*, 1–30.

ZHU, X. AND WILHELM, W. E. (2008). Three-stage approaches for optimizing some variations of the resource constrained shortest-path sub-problem in a column generation context. *European Journal of Operations Research*, **183** (2), 564–577.