University of Alberta

Wide Area Telecommunication Network Design: Problems and Solution Algorithms with Application to the Alberta SuperNet

by

Edgar Alberto Cabral ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy

in

Management Science

Faculty of Business

Edmonton, Alberta
Spring 2005

0-494-08210-0

Your file Votre référence
ISBN
Our file Notre reference
ISBN

University of Alberta

Library Release Form

Name of Author: Edgar Alberto Cabral

Title of Thesis: Wide Area Telecommunication Network Design: Problems and Solution Algorithms with Application to the Alberta SuperNet

Degree: Doctor of Philosophy

Year this Degree Granted: 2005

Edgar Alberto Cabral

Al. Navarro de Andrade 444
Adamantina, SP
CEP 17800-000
Brazil

Date: ____________________

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled *Wide Area Telecommunication Network Design: Problems and Solution Algorithms with Application to the Alberta SuperNet* submitted by Edgar Alberto Cabral in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Management Science.

Dr. Erhan Erkut (Supervisor)

Dr. Gilbert Laporte (Co-Supervisor)

Dr. Ármann Ingólfsson

Dr. M. John Hodgson

Dr. Ray Patterson

Dr. David A. Schilling (external)

Date: ____________________

To my dear "Cris & Tiana",
and my little daughter Yara.

Abstract

As Internet technologies for long-distance transmission fall in price, and the volume of data transmission surpasses that of voice transmission, networks developed solely for the Internet are becoming the trend. However, long-distance Internet technologies add considerable delay to the signal and reduce its effectiveness for real-time communication, in particular, video-conferencing and broadband entertainment in wide area network design. Consequently, a mixture of technologies is necessary in designing a wide area Internet network that allows real-time communication.

For various reasons, existing operations research models used in telecommunication are inadequate for our network design problem, so we propose new models and solution algorithms that allow for specific characteristics of network design. Our models extend those in the literature by accounting for constraints on the distance between signal regenerators, and for the costs of sheltering this equipment.

We used the Alberta SuperNet project as benchmark for wide area Internet network design for our models. The Alberta SuperNet is a cooperation between the Alberta Provincial Government and the private sector, lead by Bell West Inc. Bell West designed a high-speed Internet network that connects 422 communities in Alberta through Gigabit Ethernet optical fibre connections. The province is the first in Canada to provide this service to smaller communities.

Acknowledgments

I would like to thank everybody who supported my work in various ways and enabled me to write this dissertation.

I would like to thank my supervisor, Dr. Erhan Erkut, and co-supervisor, Dr. Gilbert Laporte, who have provided all their support and guidance.

I also would like to thank Dr. M. John Hodgson, Dr. Ármann Ingólfsson, Dr. Ray Patterson, and Dr. David A. Schilling for their valuable comments on how to improve this dissertation.

I owe special thanks to Dr. Osman Alp and Dr. Stevanus Tjandra for their encouragement, patience and support. They always found time during their post-doctoral stage to help me crystallize the operations research concepts that are present in this dissertation. I would also like to thank Dr. Fatma Gzara for her clarification on designing column generation models.

Many thanks to Bell West Inc., who provided us the problem described in the dissertation, and provided funds for my research. Also, I owe special thanks to Mr. Daniel Haight, the head of the Centre for Excellence in Operations, for the excellent opportunities presented through the centre to develop my dissertation.

Many thanks to the kind donations from the Ziegler family, Domtar Inc., and TransCanada Pipelines Ltd. to the PhD Program at the School of Business. I hope to honour your support with this dissertation.

Special thanks to Dr. David Cooper, Ms. Jeanette Gosine, Ms. Keltie Tolmie, and other faculty and staff members at the School of Business for their support in my PhD program.

Most of all, I thank my wife, Dr. Cristiana S. Seixas, for her patience, encouragement and emotional support during this work. Without her support and wisdom, this project would never have been concluded.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Wide Area Internet Network Design

Network design is a field with many key applications in transportation and telecommunication (see, for example, Balakrishnan, Magnanti, and Mirchandani 1997, and Raghavan and Magnanti 1997). Telecommunication networks are generally classified according to their geographical span and include *Local Area Networks* (LANs), *Metropolitan Area Networks* (MANs), and *Wide Area Networks* (WANs). LANs connect small areas, usually a single building or a set of buildings, with adequate capacity and speed for communication between users inside the LAN. WANs are at the other extreme, connecting a large area composed of many cities or countries; the Internet is the best example. Consequently, WANs require higher investments in infrastructure. MANs are mid-way between LANs and WANs, covering a large city or a metropolitan area. Telecommunication networks are also classified according to their topology, i.e., the main pattern in the network design. Common topologies are star, tree, ring and mesh; we provide a succinct description of these topologies in the next section. In this thesis, we consider the problem of designing a WAN in a tree topology that provides Internet service to hundreds of communities. Presently, Internet technology is not specifically designed for wide area telecommunication networks, and a mixture of technologies is necessary to form an Internet wide area network. Such a mixture introduces complications that present models in the literature have not considered, and our goal is to identify and handle them.

Dividing a network into two or more hierarchical levels to satisfy capacity, guarantee protocol requirements, and obtain a cost-effective solution is a common procedure in telecommunication network design (see Klincewicz 1998). Our models consider two hierarchies, the *backbone network* and the *access networks*. Customers are grouped into access networks, each with one backbone node as a switching centre. The backbone network connects all backbone nodes. Any communication between two customers in different access networks must go through the backbone network, first from the communication's origin node to the origin's assigned backbone node, then to the destination's assigned backbone node, and finally to the communication destination's node.

This thesis is organized as follows. Chapter 2 provides a review of the network design literature that is related to our research, a description of wide area network design using Internet technology and of the Alberta SuperNet, and a summary of relevant issues missing in the literature that are related to this type of network design. In Chapter 3, we present a mathematical formulation of the telecommunication network design problem for the Alberta SuperNet. We also describe a methodology to solve this problem. Due to its complexity, we divide the problem into several sub-problems. Chapter 4 describes the sub-problem of defining a path with appropriate repeaters installed so that a signal between two points respects the maximum distance between repeaters. Chapter 5 describes the sub-problem of where to install fibre and equipment shelters and proposes algorithms to solve this problem. Chapter 6 describes the sub-problem of defining protocols, equipment, hardware and fibre use for a given network with fibres and equipment shelters. Finally, Chapter 7 concludes this thesis.

2 Literature Review

Network design permeates much of today's design literature, from the micro scale (electronic chip design) to the macro scale (world-wide communication networks). Each design may impose different characteristics, and identifying the similarities in design over different areas becomes impractical. We therefore concentrate on telecommunication network design, limiting the literature review to articles published in Operations Research journals in the fields of telecommunication and transportation. Yet, we expect the coverage to be sufficiently large to justify the work proposed in this thesis.

Among the numerous papers in telecommunication network design, we find those in the area of operations research applied to network optimization the most relevant to our research: Sexton and Reid (1992), Wu (1992), Sanso and Soriano (1999), and Doverspike and Saniee (2000).

In this thesis, we consider two main costs associated with the use of an arc to form the telecommunication network. The first cost, designated the *trenching cost* of an arc, is fixed and corresponds to the cost of installing optical fibres in the ground. In general, the cost is linearly related to the length of the arc and corresponds to the cost of digging a trench in the ground, installing the conduits and the fibre, and closing the trench. The second cost corresponds to all optical fibres installed along the arc that transport the signals. This cost depends on the number of fibres installed and is usually non-linear and convex. We define this as the *fibre cost* in an arc.

Similarly, we consider two main costs associated with the installation of equipment in a node. The first cost corresponds to the fixed cost of installing an equipment shelter, which can accommodate any combination of equipment and is called the *shelter cost* of a node. The second cost is for the communication equipment itself, and it depends on the type of equipment installed at the node to render communication feasible. This last cost is designated *equipment cost* at a node.

As mentioned before, networks can be classified by their topology. In a *star* topology, one node is considered the centre of the topology, and all other nodes are connected directly to it.

forming a network design similar to a star (see Figure 2-1). Each line represents a physical medium connecting the nodes to the centre node, for example, the installation of optical fibres underground, inside a pipeline. Usually, star networks are acceptable, if the cost to place the medium connecting the two nodes is irrelevant compared to all other necessary expenses (generally true for small area networks), or when the trench costs are irrelevant (usually the case in building installations, where conduits are available).

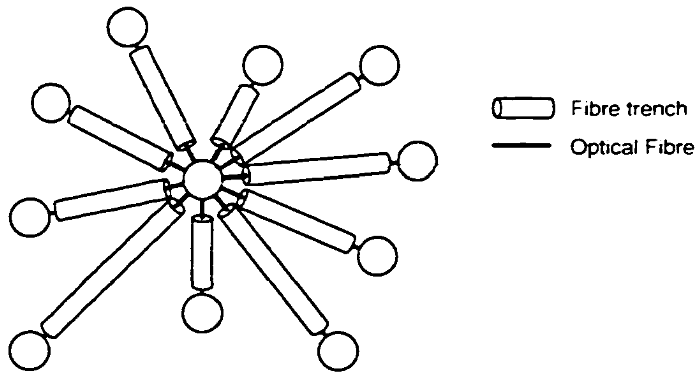

Figure 2-1: Star topology

However, as the span area increases, the trench cost becomes relevant, especially if the arcs do not provide proper fibre trenches. Thus, a *tree* topology may be more economical, as the trench costs to install fibres are shared by multiple communication links. Figure 2-2 shows a solution that may be less expensive than that in Figure 2-1 for the same set of nodes.

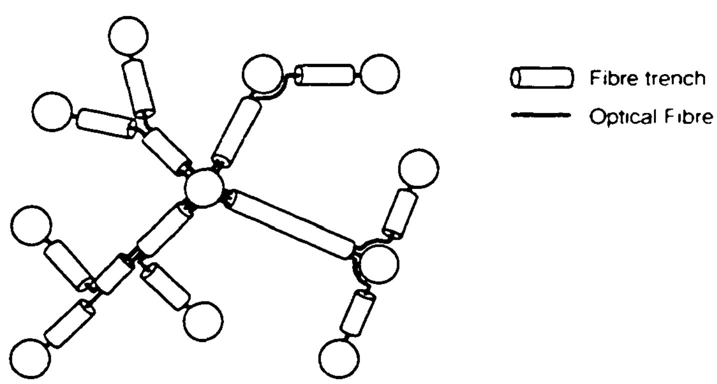

Figure 2-2: Tree topology

The major drawback for the star and the tree topologies is the absence of alternative paths between pairs of nodes. Any failure in a link or at a node can only be restored by fixing the failing element, which can take a long time and result in huge losses for network users (see MacDonald 1994).

Another common topology in network telecommunication is the *ring* structure. It became popular with the availability of telecommunication equipment that could automatically redirect the communication flow in the case of link or equipment failure. Communication rings that can automatically recover from a single failure (either of one link or of equipment located in one node) are termed *Self-healing rings* (SHR) in the literature. Although there are several types of SHR, depending on the reconfiguration procedure and type of protocol used, we classify according to the appropriate ring capacity for a given set of demands: the *unidirectional* SHR and the *bidirectional* SHR (as presented by Soriano et al. 1999). In the *unidirectional* SHR, all traffic is routed in the same direction along one fibre, the working fibre. The second fibre is the protection fibre, which is used only when a failure is detected (see Figure 2-3). Another common denomination for this configuration is *dedicated protection* SHR, as one fibre is dedicated to protect the network from possible communication failure.

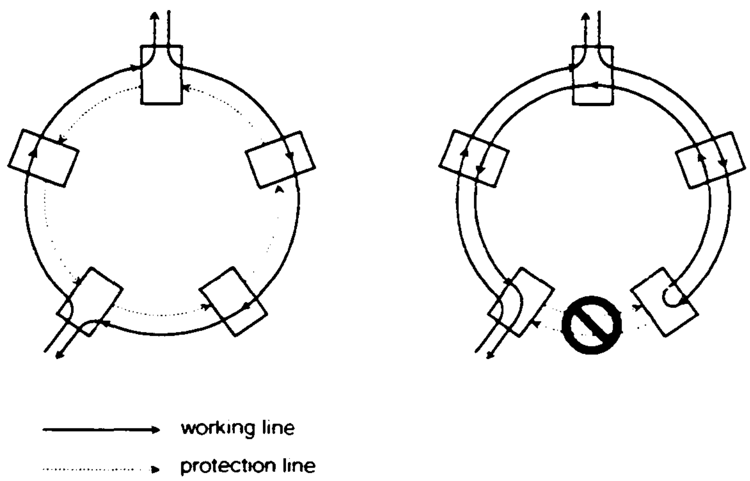

Figure 2-3: Unidirectional SHR

In the bidirectional SHR, instead of having a dedicated fibre for recovery, the existing communication bandwidths inside the fibres are divided into two portions: one for regular communication and another for protection. Therefore, in the case of a communication failure, the signal is redirected inside the fibre bandwidths. This configuration may be more efficient, capacity-wise, than the unidirectional SHR.

Finally, if the network cannot be classified as a star, a tree or a ring, we define it to be a *mesh* topology, as long as it contains alternative paths between a subset of nodes in the network. However, when considering different network hierarchies, one can combine two topologies to form one network design, for example, a backbone ring topology combined with tree access networks. To make a distinction between mesh structures and combinations of star, tree, and ring topologies, we call them *mixed* topologies.

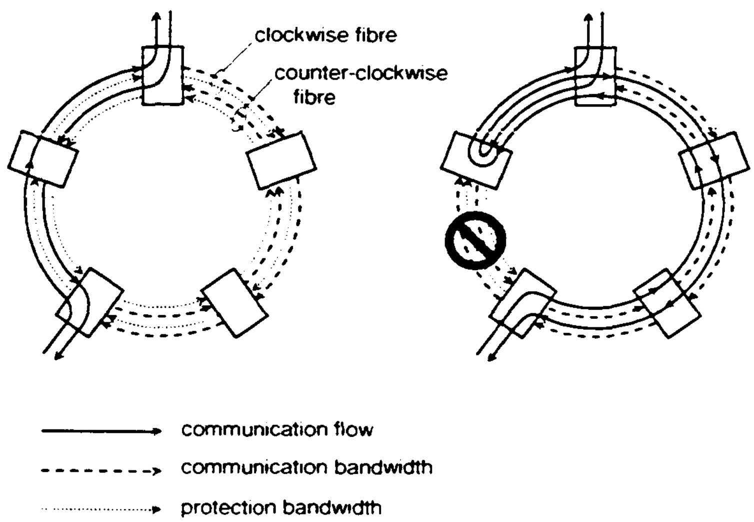

Figure 2-4: Bidirectional SHR

We created a classification method for selecting articles to use in our research, because many articles relate to telecommunication network design. In our table, one dimension is given by the network topology and the other is given by the type of *layer*. Topologies are classified as *tree*, *ring*, *mesh*, or a mixture of the three, called *mixed*. Articles that used the star topology are included in the mixed topology row, as star topology is usually used for the access network and can be combined with any of the other topologies. Layers can be classified as *backbone* or *access*, as in the literature, or a combination of both that we call *combined*. The latter also includes multi-hierarchical networks, which can be found in transportation literature.

Table 2-1 displays this classification. The tree and mixed topologies are of interest from Table 2-1. We discuss articles and book chapters related to the tree network structure, in the next sub-section and those related to mixed structures, which may or may not include tree topologies in their design, in sub-section 2.2.

		Backbone	Access	Combined
Topology	**Tree**	Esbensen (1995), Gendreau, Larochelle, and Sanso (1999), Gouveia (1993, 1995, 1999), Gouveia and Martins (1999, 2000), Hwang and Richards (1992), Hwang *et al.* (1992), Kapsalis *et al.* (1993), Koch and Martin (1998), Lee *et al.* (1996), Lucena and Beasley (1998), Magnanti and Wolsey (1995), Malik and Yu (1993), Mehlhorn (1988), Patterson *et al.* (1999), Polzin and Vahdati Daneshmand (2001a, 2001b), Rayward-Smith and Clare (1986), Takahashi and Matsuyama (1980)	Girard *et al.* (2001), Randazzo and Luna (2001)	
	Ring	Armony *et al.* (2000), Chamberland and Sanso (2000), Chang and Chang (2000), Chung *et al.* (1996), Grover *et al.* (1995), Laguna (1994), Lee *et al.* (2000), Luss *et al.* (1998), Sherali *et al.* (2000)		
	Mesh		Grover *et al.* (1997, 1999), Pickavet and Demeester (2000)	
	Mixed	Wu, Cardwell, and Boyden (1991)	Altinkemer and Yu (1992), Gavish (1991)	Balakrishnan, Magnanti, and Mirchandani (1994a, 1994b, 1998), Chamberland and Sanso (2001), Chamberland *et al.* (1996, 2000), Chardaire (1999), Chopra and Tsai (2002), Chung *et al.* (1992), Current *et al.* (1986), Gavish (1992), Kim *et al.* (1995), Labbé et al. (2001), Lee *et al.* (1993), Park *et al.* (2000), Rosenberg (2001)

Table 2-1. Classification of papers on network design problems

2.1 Tree Network

The major models for tree network design in networks are the *Steiner Tree Problem* (STP), the *Capacitated Minimum Spanning Tree Problem* (CMSTP), and the *Capacitated Tree Problem* (CTP). The articles presented in the (*tree, backbone*) cell in Table 2-1 are considered below.

The STP is closely related to our problem. It describes the search for a minimum cost-spanning tree that connects a *mandatory* subset of nodes in a graph, while minimizing the cost of edges included in the solution. *Non-mandatory* nodes that are included in the optimal solution are called *Steiner* nodes. This problem was proven NP-hard by Garey and Johnson (1979), and surveys are provided by Maculan (1987); Winter (1987); Hwang and Richards (1992); and Hwang, Richards, and Winter (1992). We group heuristic solutions as *classical* (see Takahashi and Matsuyama 1980, Rayward-Smith and Clare 1986, and Mehlhorn 1988) and as *modern* (see meta-heuristics such as the tabu search applied to the STP by Gendreau, Larochelle, and Sanso 1999 and the genetic algorithms applied to the STP by Kapsalis, Rayward-Smith, and Smith 1993 and by Esbensen 1995). Classical algorithms are faster than modern algorithms, but present inferior solutions. Examples of exact solution methods for the STP are given by Koch and Martin (1998), Lucena and Beasley (1998), and Polzin and Vahdati Daneshmand (2001b). When solving the STP, it is important to rely on reduction techniques for the problem, as described in Duin and Voss (1989), Esbensen (1995) and Polzin and Vahdati Daneshmand (2001b). Reduction techniques consist of eliminating nodes and edges that can be proven unnecessary from the graph to reduce problem size, and play a major role in present algorithms that find optimal solutions to the STP, as discussed in Polzin and Vahdati Daneshmand (2001a, 2001b)[1].

We present an example of an STP in Figure 2-5. Black nodes are the mandatory node subset that must be connected through the STP, whereas not all white nodes are required in the final solution.

[1] Reduction techniques could not be developed for our design problem during the process of this thesis. They may be considered in future work.

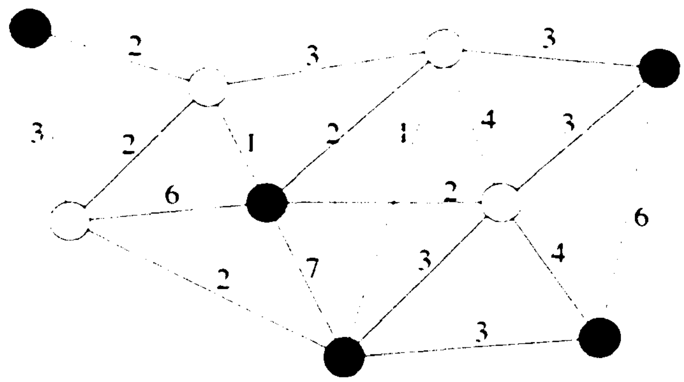

Figure 2-5: Example of the STP

Solving the *Minimum Spanning Tree* (MST) on this example provides a feasible solution to the problem, with a total trench cost of 16. The MST is solvable in polynomial time. One could extract sub-graphs from this example that contain mandatory nodes and a subset of non-mandatory nodes to evaluate their optimal MST solution. Using such a procedure, after evaluating a total of 16 different graphs, we obtained the optimal solution to the STP above, which has 12 as the total trench cost. We present this solution in Figure 2-6. Although non-mandatory nodes exist in the STP, it does not consider the communication signals necessary for signal reinforcement. Consequently, we use STP mainly to show that our network design problem is NP-hard.

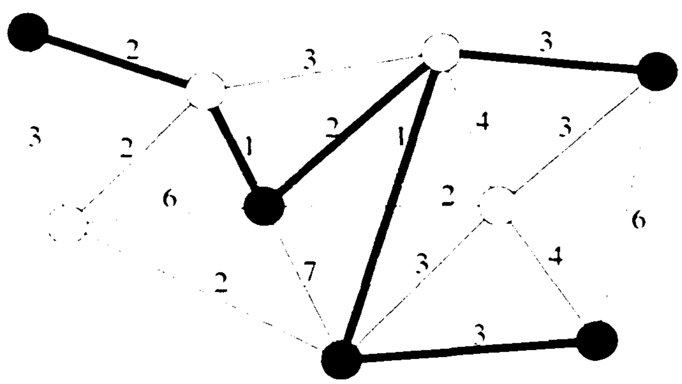

Figure 2-6: Solution of the STP from Figure 2-5

In the CMSTP, one searches for a minimum cost spanning tree, in which each subtree emanating from a given root node has a number of nodes smaller than Q, the maximum number of nodes allowed for each subtree. Although such a problem is common in telecommunication network design (see Malik and Yu 1993; Gouveia 1993; Gouveia 1995; Gouveia and Martins (1999, 2000); and Patterson, Rolland, and Pirkul 1999), it considers all nodes to be mandatory when designing the network, an impractical assumption for our problem. The same occurs in the case of the CTP, a generalization of the CMSTP problem, in which communication capacity is also considered with respect to the links of the network (see Lee, Park, and Park 1996).

The two articles presented in the (*tree, access*) cell of Table 2-1 were specifically published for access network design. Girard, Sanso, and Dadjo (2001) present the *Multi-Center Capacitated Spanning Tree* (MCCST), which incorporates the idea of having many center nodes in the access level. Randazzo and Luna (2001) present a modification of the STP, where linear costs for communication flow over the links are considered. Again, both models assume all nodes are mandatory.

2.2 Mixed Topology

The *mixed* topology articles described in Table 2-1 correspond to problems that did not enforce a particular topology or that had two or more hierarchies, one of which was a tree topology or no particular topology. For example, network design models that use flow conservation constraints to guarantee connectivity can create a mesh structure, but can also create a tree topology if the communication flow has an aggregated origin or destination. Therefore, they are capable of generating tree topologies and should be considered for our problem. The *backbone topology/access topology* notation represents two-hierarchical topologies; for example, the tree/star notation means that the backbone network follows a tree topology, and the access network follows a star topology.

Among the articles presented in the *mixed* topology row in Table 2-1, we can classify Gavish (1991, 1992), and Altinkemer and Yu (1992) as surveys on network design. In particular, Altinkemer and Yu consider communication flow delay through embedded queuing costs in its formulation. They suggest the use of Lagrangean Relaxation to obtain a lower bound and, consequently, feasible solutions to the network design problem.

Some articles explicitly enforce a tree structure in one of the levels. Chamberland, Sanso, and Marcotte (2000), for example, describe a tabu search algorithm that produces a two-level network of either a ring/star or a tree/star. They consider different capacities and different technologies per hierarchy, but the important point is that their model also considers constraints on the nodes, in particular equipment-switching capacity. An extension is presented in Chamberland and Sanso (2001). They include switch-fabric capacity at the nodes and *dual homing* for some clients, which consists of assigning access nodes to two backbone nodes instead of one, thus increasing the survivability of individual connections. Nonetheless, the algorithm presented by Chamberland and Sanso can only produce ring/star topology solutions.

Other algorithms in the literature can implicitly enforce a tree topology by assigning a common origin or destination to the communication flow. Examples are Chamberland, Marcotte, and Sanso (1996) and Rosenberg (2001); the latter is of particular interest because it considers the cost of inserting equipment for signal conversion. A dual ascent technique is used to produce feasible solutions.

Major models that produce a tree/tree topology from the *mixed* topology row in Table 2-1 are the *Hierarchical Network Design* (HND), the *Two-level Network Design* (TLND), and the *Multi-Level Network Design* (MLND).

The HND problem (Current, ReVelle, and Cohon 1986) finds a path/tree topology. It also finds a spanning tree where two primary nodes must be connected through a higher grade link path, and the other nodes may be connected through lower grade links.

The TLND can be seen as an extension to the HND problem, as the set of nodes to be connected can be divided into primary and secondary nodes. The TLND searches a tree spanning all nodes, where the primary nodes must be connected through a tree that uses higher grade links (and may contain secondary nodes), and the rest can be serviced through lower grade links. This problem was presented in Balakrishnan, Magnanti, and Mirchandani (1994a, 1994b); the first describes a heuristic, and the second presents a dual-based algorithm. Because the TLND is an extension to the STP, it is proven NP-hard.

Finally, the MLND problem is an extension of the TLND, where more than two hierarchies can be created. It was presented by Chopra and Tsai (2002): the problem was solved through a conversion to an STP combined with a branch-and-cut approach.

2.3 The Alberta SuperNet Project

This research was motivated by the Alberta SuperNet project, a partnership between the Alberta Provincial Government and a private consortium led by Bell West Inc. The goal is to provide broadband Internet access to 422 communities across Alberta. During the request for proposals phase, Bell contacted the Centre for Excellence in Operations (CEO) at the University of Alberta School of Business. They wanted to find a network design that would satisfy requirements from the initial proposal and minimize the overall implementation costs. After this initial contact, a research partnership was formed between Bell West Inc. and the CEO to conduct research on telecommunication network design problems.

When completed, the Alberta SuperNet will consist of two main parts: the backbone and the access network. The backbone will connect 27 major communities, while the access network will connect 395 smaller communities to the backbone. Each access network will have one switching center called a backbone node, which is defined as the *root node* for that access network. In a similar fashion, we also define a *root node* for the backbone network that switches all communication originating at the backbone nodes.

Optical fibres in the Alberta SuperNet will be installed along existing roads, and therefore, the design problem will use the road network as input. According to our GIS database, the Alberta road network is composed of approximately 280,000 arcs and 86,000 nodes (see Figure 2-7).

Three telecommunication transport technologies are considered in this thesis: *Gigabit Ethernet* (GE), *Synchronous Optical Network* (SONET) and *Dense Wavelength Division Multiplex* (DWDM). The first technology is directly compatible with the *Internet Protocol* (IP), while the last two are established technologies for wide area network designs. GE, SONET and DWDM use the same type of optical fibre[2]. But a GE fibre link can transmit only

---

[2] Single-mode fiber is chosen for long distance, high capacity transmission (see the Fiber Optic Technology tutorial available at www.iec.org).

2.5 Gbps (Gigabits per second) per fibre, compared to 10 Gbps per fibre for a SONET fibre link and 40 Gbps per fibre for a DWDM fibre link. GE equipment costs between $10,000 and $20,000, SONET equipment costs between $60,000 and $70,000, and DWDM equipment costs between $70,000 and $100,000. GE equipment introduces a delay to the communication signal traversing it, whereas SONET and DWDM introduce almost no signal delay. Therefore, depending on the maximum acceptable delay, the network may consist of a mix of these three technologies.

Each technology involves three types of hardware: *repeaters*, *multiplexers* and *switchers*. A *repeater* simply regenerates the signal(s) entering a node, and we assume that if a repeater is installed at a node, all signals are regenerated while maintaining their individuality. The signals are not bundled into the same communication stream. For example, if three SONET signals enter a node that has a SONET repeater with each signal using one fibre, there will be three regenerated signals leaving the node, each using one fibre.

A *multiplexer* bundles multiple signals into one, while regenerating them. Therefore, if three SONET signals enter a node, each using one cable, the output will be a single signal, requiring one, two, or three fibres, depending on the capacity required for each signal. If the three signals use less than one-third of a SONET fibre's communication capacity, the multiplexer can bundle all signals into one fibre. But if the three signals use almost all of one fibre's capacity, the resulting bundled signal should use three fibres to provide the desired communication demand. Finally, a *switcher* converts signals that use one technology/protocol into signals compatible with another technology/protocol. Switchers also bundle signals. For example, if three SONET signals enter a node that has a SONET to DWDM switcher, the output signal will likely be one DWDM signal along one fibre, because DWDM can combine 32 SONET fibres into one.

Gigabit Ethernet is the chosen *access level* communication technology for the Alberta SuperNet, because of its low implementation costs and its direct compatibility with IP. SONET and DWDM are the chosen *backbone level* communication technologies, because of their high bandwidth per optical fibre and virtual absence of communication delay, compared to GE technology. Nevertheless, SONET and DWDM can be used in the access networks to guarantee acceptable levels of signal delay.

The Alberta SuperNet spans a wide area, and therefore, some signals travel far enough that they need to be regenerated. GE signals can travel for approximately 70 km, while SONET and DWDM signals can travel without regeneration for approximately 85 km. For practical reasons, we use the maximum distance of 70 km for all technologies, so that decision makers can switch from one technology to another without needing to relocate shelters for signal regeneration.

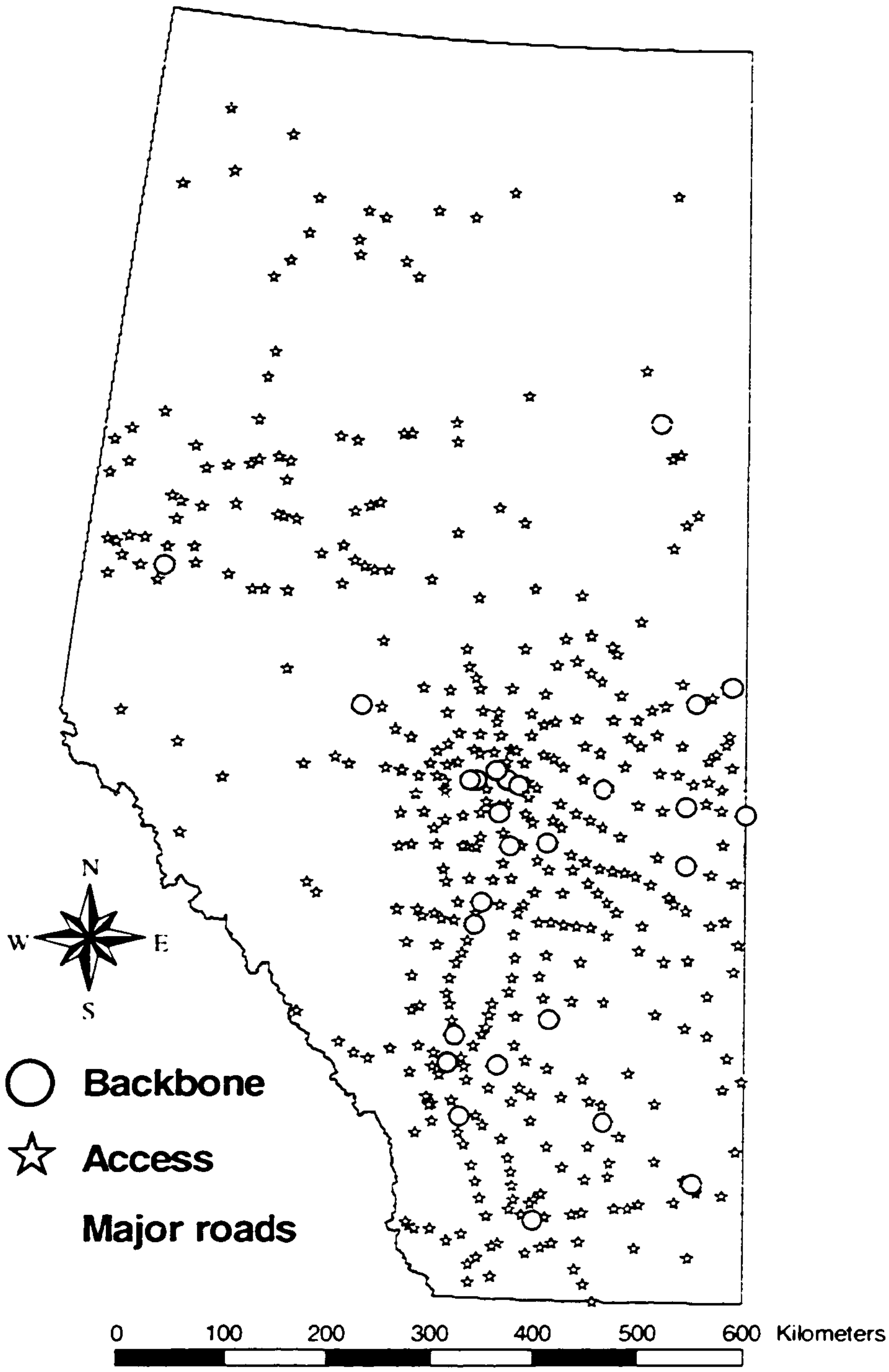

Figure 2-7. The Alberta SuperNet Project

To reduce the implementation costs, the planners of the Alberta SuperNet project decided that the network design should not have alternative paths (redundancy) for communication flows. Thus, the most cost effective topology is a *tree network structure*, in which *digging costs* and *fibre installation costs* are minimized, as suggested by Chamberland, Sanso, and Marcotte (2000). The Alberta SuperNet project also permits the co-existence of two or more technologies along the same cable in different strands of optical fibres. With such freedom, both backbone and access signals can travel in parallel, as long as there are a sufficient number of fibres available in the link for all signals.

The use of multiple technologies renders the telecommunication network design problem complex for small scale networks. We aim to design algorithms to automate the network design while minimizing costs and respecting all constraints imposed by the project. The constraints are summarized as follows:

- *The presence of mandatory and non-mandatory nodes.* As the road network is used as input, not all nodes are necessary in the final design, creating the necessity to distinguish between mandatory and non-mandatory nodes.

- *Constraints on distance between equipment for signal regeneration.* As the distance covered by the Alberta SuperNet is longer than signal transmission capacity without regeneration, we must consider signal regeneration and equipment location that is responsible for desired signal regeneration.

- *The necessity to limit signal delay caused by hardware.* As the Alberta SuperNet provides real-time communication capability between two communities for video conferencing and video broadcasting, different hardware and signal technology must be used to limit signal delay to the acceptable levels required for these services.

- *The possibility of carrying signals of different protocols in parallel.*

- *The option of either laying or leasing optical fibres* creates the necessity of accounting for fibre use explicitly for each communication signal flow.

Although, in this study, the technologies, protocols and hardware are mandated for the Alberta SuperNet project, the solutions extend to similar networks that involve large areas with fewer mandatory nodes than non-mandatory nodes.

2.4 Summary

No model presented in the previous section includes the following important constraints from the Alberta SuperNet project. Table 2-2 below summarizes the models presented (STP, CMSTP, CTP, MCCST, HND, TLND, MLND) according to each constraint added by the Alberta SuperNet project:

	STP	CMSTP	CTP	MCCST	HND	TLND	MLND
Non-mandatory nodes	Yes	No	No	No	No	No	No
Distance between repeaters	No	No	No	No	No	No	No
Signal delay	Yes/No[3]	No	No	No	No	No	No
Parallel signals	No	No	No	No	No	No	No
Lease: capacitated design	No	Yes	Yes	Yes	No	No	No

Table 2-2. Models from the literature and the Alberta SuperNet project

From this table, we see that a model that solves all problems together is not available in the literature. Moreover, simpler problems are hard to solve, and therefore, we should simplify our problem in order to render it tractable. In the next section, we mathematically formulate our problem and describe a solution method.

[3] Articles with hop-indexed notation for the STP can handle signal delay through limitation of the number of nodes traversed by the signal before reaching the root node (see Gouveia and Martins (1999, 2000)).

2.5 References

Altinkemer, K., Z. Yu. 1992. Topological Design of Wide Area Communication Networks. *Annals of Operations Research* **36** 365-382.

Armony, M., J.G. Klincewicz, H. Luss, M.B. Rosenwein. 2000. Design of Stacked Self-Healing Rings Using a Genetic Algorithm. *Journal of Heuristics* **6**(1) 85-105.

Balakrishnan, A., T.L. Magnanti, P. Mirchandani. 1994a. Modeling and Heuristic Worst-case Performance Analysis of the Two-level Network Design Problem. *Management Science* **40**(7) 846-866.

Balakrishnan, A., T.L. Magnanti, P. Mirchandani. 1994b. A Dual-based Algorithm for Multi-level Network Design. *Management Science* **40**(5) 567-581.

Balakrishnan, A., T.L. Magnanti, P. Mirchandani. 1996. Heuristics, LP's and Trees in Trees: Network Design Analysis. *Operations Research* **44** 478-496.

Balakrishnan, A., T.L. Magnanti, P. Mirchandani. 1997. Network Design. Dell'Amico, M., F. Maffioli, S. Martello, Eds. *Annotated Bibliographies in Combinatorial Optimization*. John Wiley & Sons, Chichester.

Balakrishnan, A., T.L. Magnanti, P. Mirchandani. 1998. Designing Hierarchical Survivable Networks. *Operations Research* **46**(1) 116-136.

Ball, M.O., T.L. Magnanti, G.L. Nemhauser. 1995. *Network Models*. Elsevier, North-Holland, Amsterdam.

Chamberland, S., B. Sanso, O. Marcotte. 2000. Topological Design Of Two-Level Telecommunication Networks With Modular Switches. *Operations Research* **48**(5) 745-760.

Chamberland, S., B. Sanso. 2000. Topological Expansion Of Multiple-Ring Metropolitan Area Networks. *Networks* **36**(4) 210-224.

Chamberland, S., B. Sanso. 2001. On the design problem of multitechnology networks. *INFORMS Journal on Computing* **13**(3) 245 - 256.

Chamberland, S., O. Marcotte, B. Sanso. 1996. On the Joint Topological, Dimensioning and Location Problem for Broadband Networks. *Proceedings of the International IFIP-IEE Conference on Broadband Communications*, 525-536.

Chang, M.R., S.Y. Chang. 2000. A Heuristic Method For Self-Healing Ring Design In A Single-Homing Cluster. *Telecommunication Systems* **14**(1-4) 175-195.

Chardaire, P. 1999. Hierarchical Two Level Location Problems. Sanso, B., P. Soriano, Eds. *Telecommunications Network Planning*. Kluwer Academic Publishers, Boston.

Chopra, S., C.-Y. Tsai. 2002. A branch-and-cut approach for minimum cost multi-level network design. *Discrete Mathematics* **242** 65-92.

Chung, S.H., H.G. Kim, Y.S. Yoon, D.W. Tcha. 1996. Cost-Minimizing Construction Of A Unidirectional SHR With Diverse Protection. *IEEE-ACM Transactions On Networking* **4**(6) 921-928.

Chung, S.H., Y.-S. Myung, D.-W. Tcha. 1992. Optimal Design of a Distributed Network with a Two-Level Hierarchical Structure. *European Journal of Operations Research* **62** 105-115.

Current, J., C. ReVelle, J.L. Cohon. 1986. The Hierarchical Network Design Problem. *European Journal of Operational Research* **27** 57-66.

Dell'Amico, M., F. Maffioli, S. Martello. 1997. *Annotated Bibliographies in Combinatorial Optimization*. John Wiley & Sons, Chichester.

Doverspike, R., I. Saniee. 2000. *Heuristic Approaches for Telecommunications Network Management, Planning and Expansion*. Kluwer Academic Publishers, Norwell MA.

Duin, C.W., S. Voss. 1989. Reduction Tests for the Steiner Problem in Graphs. *Networks* **19** 549-567.

Esbensen, H. 1995. Computing Near-Optimal Solutions to the Steiner Problem in a Graph Using a Genetic Algorithm. *Networks* **26** 173-185.

Frantzeskakis, L.F., H. Luss. 1999. The Network Redesign Problem For Access Telecommunications Networks. *Naval Research Logistics* **46**(5) 487-506.

Garey, M.R., D.S. Johnson. 1979. *Computers and intractability: A guide to the theory of NP-completeness*. W.H. Freeman, San Francisco.

Gavish, B. 1991. Topological Design of Telecommunication Networks - Local Access Design Methods. *Annals of Operations Research* **33** 17-71.

Gavish, B. 1992. Topological Design of Computer Communication Networks - The Overall Design Problem. *European Journal of Operations Research* **58** 149-172.

Gendreau, M., J.-F. Larochelle, B. Sanso. 1999. A Tabu Search Heuristic for the Steiner Tree Problem. *Networks* **34**(2) 162-172.

Girard, A., B. Sanso, L. Dadjo. 2001. A Tabu Search Algorithm for Access Network Design. *Annals of Operations Research* **106** 229-262.

Gouveia, L. 1993. A Comparison of Directed Formulations for the Capacitated Minimum Spanning Tree Problem. *Telecommunication Systems* **1** 51-76.

Gouveia, L. 1995. A 2n Constraint Formulation for the Capacitated Minimum Spanning Tree Problem. *Operations Research* **43** 130-141.

Gouveia, L. 1999. Using Hop-Indexed Models For Constrained Spanning and Steiner Tree Models. Sanso, B., P. Soriano, Eds. *Telecommunications Network Planning*. Kluwer Academic Publishers, Boston.

Gouveia, L., P. Martins. 1999. The Capacitated Minimum Spanning Tree Problem: An Experiment with Hop-Indexed Model. *Annals of Operations Research* **86** 271-294.

Gouveia, L., P. Martins. 2000. A hierarchy of hop-indexed models for the capacitated minimum spanning tree problem. *Networks* **35**(1) 1-16.

Grover, W.D., J.B. Slevinsky, M.H. MacGregor. 1995. Optimized Design of Ring-Based Survivable Networks. *Canadian Journal Of Electrical And Computer Engineering* **20**(3) 139-149.

Grover, W.D., R.R. Iraschko, Y. Zheng. 1999. Comparative Methods and Issues in Design of Mesh-Restorable STM and ATM Networks. Sanso, B., P. Soriano, Eds. *Telecommunications Network Planning*. Kluwer Academic Publishers, Boston.

Grover, W.D., V. Rawat, M.H. MacGregor. 1997. Fast heuristic principle for spare capacity placement in mesh-restorable SONET/SDH transport networks. *Electronics Letters* **33**(3) 195-196.

Hwang, F.K., D.S. Richards, P. Winter. 1992. *The Steiner Tree Problem*. North-Holland, Amsterdam.

Hwang, F.K., D.S. Richards. 1992. Steiner Tree Problems. *Networks* **22** 55-89.

Jorgensen, D.G., M. Meyling. 2002. A Branch-and-Price Algorithm for Switch-Box Routing. *Networks* **40**(1) 13-26.

Kapsalis, A., V.J. Rayward-Smith, G.D. Smith. 1993. Solving the Graphical Steiner Tree Problem Using Genetic Algorithms. *Journal Of The Operational Research Society* **44**(4) 397-406.

Kim, H.-J., S.-H. Chung, D.-W. Tcha. 1995. Optimal design of the two-level distributed network with dual homing local connections. *IIE Transactions* **27** 555-563.

Klincewicz, J.G. 1998. Hub Location in Backbone/Tributary Network Design: A Review. *Location Science* **6** 307 - 335.

Koch, T., A. Martin. 1998. Solving Steiner Tree Problems in Graphs to Optimality. *Networks* **32** 207-232.

Labbé, M., G. Laporte, I.R. Martín, J.J.S. González. 2004. The Ring Star Problem: Polyhedral Analysis and Exact Algorithm. *Networks* **43**(3) 177-189.

Laguna, M. 1994. Clustering for the Design of SONET Rings in Iteroffice Telecommunications. *Management Science* **40**(11) 1533-1541.

Lee, C.H., H.B. Ro, D.W. Tcha. 1993. Topological Design Of A Two-Level Network With Ring Star Configuration. *Computers and Operations Research* **20**(6) 625-637.

Lee, K., K. Park, S. Park. 1996. Design of Capacitated Networks with Tree Configurations. *Telecommunication Systems* **6** 1-19.

Lee, Y., H.D. Sherali, J. Han, S. Kim. 2000. A Branch-And-Cut Algorithm For Solving An Intraring Synchronous Optical Network Design Problem. *Networks* **35**(3) 223-232.

Lucena, A., J.E. Beasley. 1998. A Branch and Cut Algorithm for the Steiner Problem in Graphs. *Networks* **31**(1) 39-59.

Luss, H., M.B. Rosenwein, R.T. Wong. 1998. Topological Network Design for SONET Ring Architecture. *IEEE Transactions on Systems, Man, and Cybernetics-Part A* **28**(6) 780-790.

MacDonald, J.C. 1994. Public Network Integrity - Avoiding a Crisis of Trust. *IEEE Journal on Selected Areas in Communications* **12**(1) 5-12.

Maculan, N. 1987. The Steiner Problem in Graphs. *Annals of Discrete Mathematics* **31** 185-212.

Magnanti, T.L., L.A. Wolsey. 1995. Optimal Trees. Ball, M.O., T.L. Magnanti, G.L. Nemhauser, Eds. *Network Models*. Elsevier, North-Holland, Amsterdam.

Malik, K., G. Yu. 1993. A Branch-And-Bound Algorithm For The Capacitated Minimum Spanning Tree Problem. *Networks* **23**(6) 525 - 532.

Mehlhorn, K. 1988. A Faster Approximation Algorithm for the Steiner Problem in Graphs. *Information Processing Letters* **27** 125-128.

Park, K., K. Lee, S. Park, H. Lee. 2000. Telecommunication Node Clustering With Node Compatibility And Network Survivability Requirements. *Management Science* **46**(3) 363-374.

Patterson, R., E. Rolland, H. Pirkul. 1999. A Memory Adaptive Reasoning Technique for Solving the Capacitated Minimum Spanning Tree Problem. *Journal of Heuristics* **5** 159-180.

Pickavet, M., P. Demeester. 2000. A Zoom-In Approach To Design SDH Mesh Restorable Networks. *Journal of Heuristics* **6**(1) 107-130.

Polzin, T., S. Vahdati Daneshmand. 2001a. A comparison of Steiner Tree Relaxations. *Discrete Applied Mathematics* **112**(3) 241-261.

Polzin, T., S. Vahdati Daneshmand. 2001b. Improved Algorithms for the Steiner Problem in Networks. *Discrete Applied Mathematics* **112**(3) 263-300.

Raghavan, S., T.L. Magnanti. 1997. Network Connectivity. Dell'Amico, M., F. Maffioli, S. Martello, Eds. *Annotated Bibliographies in Combinatorial Optimization*. John Wiley & Sons, Chichester.

Randazzo, C.D., H.P.L. Luna. 2001. A Comparison of Optimal Methods for Local Access Uncapacitated Network Design. *Annals of Operations Research* **106** 263-286.

Rayward-Smith, V.J., A. Clare. 1986. On Finding Steiner Vertices. *Networks* **16** 283-294.

Rosenberg, E. 2001. Dual ascent for uncapacitated telecommunications network design with access, backbone, and switch costs. *Telecommunication Systems* **16**(3-4) 423 - 435.

Sanso, B., P. Soriano. 1999. *Telecommunications Network Planning*. Kluwer Academic Publishers, Boston.

Sexton, M., A. Reid. 1992. *Transmission networking: SONET and the synchronous digital hierarchy*. Artech House, Boston.

Sherali, H.D., J.C. Smith, Y. Lee. 2000. Enhanced Model Representations For An Intra-Ring Synchronous Optical Network Design Problem Allowing Demand Splitting. *INFORMS Journal On Computing* **12**(4) 284-298.

Soriano, P., C. Wynants, R. Seguin, M. Labbe, M. Gendreau, B. Fortz. 1999. Design and Dimensioning of Survivable SDH/SONET Networks. Sanso, B., P. Soriano, Eds. *Telecommunications Network Planning*. Kluwer Academic Publishers, Boston.

Takahashi, H., A. Matsuyama. 1980. An Approximate Solution for the Steiner Problem in Graphs. *Mathematica Japonica* **24**(6) 573-577.

Winter, P. 1987. Steiner Problem in Networks: A Survey. *Networks* **17** 129-167.

Wu, T-H. 1992. *Fiber Network Service Survivability*. Artech House, Boston/London.

Wu, T-H., R.H. Cardwell, M. Boyden. 1991. A Multi-period Design Model for Survivable Network Architecture Selection for SONET Interoffice Networks. *IEEE Transactions on Reliability* **40**(4) 417-427.

3 Problem Formulation and Solution Methodology

3.1 Problem Formulation

The requirement to keep track of distances so as to obey the upper bound on the distance between repeaters (defined as Λ) creates a serious complication in the formulation of this problem. We considered a simpler network design problem where only maximum distance constraints were considered, and attempted to use a network flow model where node labels represented the distances traveled by signals after the most recent repeater. This approach, however, turned out to be unsuitable for problems where the optimal solution includes a "detour," where a path enters and leaves a node along the same edge.

To illustrate this limitation, we present a formulation and apply it to the example in Figure 3-1, where the signal originates at node 1 and it is destined for node 4. Let N be the node set and A the arc set. Let X_{ij} be a binary variable representing communication flow through arc (i,j), Y_i be a binary variable representing a multiplexer installed at node i to regenerate the signal at node i, and Z_{ij} be a binary variable representing trench opening along arcs $(i,j)$and(j,i). Let variables λ_i and λ_i' represent distance travelled by the signal so far, λ_i being the distance right before the signal leaves node i, and λ_i' being the distance right before the signal enters node i. Both variables must be smaller than Λ. As constants, we have d_{ij} as the length of arc (i,j), h_i as the cost to install a multiplexer at node i, and c_{ij} as the cost to install a trench along arcs (i,j) and (j,i) (in other words, this cost is incurred if X_{ij} or X_{ji} are equal to 1).

Min

$$\sum_{\substack{(i,j)\in A \\ i<j}} c_{ij}Z_{ij} + \sum_{i\in N} h_i Y_i \qquad (3.1)$$

S.t.

$$\sum_{(i,j)\in A} X_{ij} - \sum_{(j,i)\in A} X_{ji} = \begin{cases} 1 & \text{if } i \text{ is origin} \\ -1 & \text{if } i \text{ is destination} \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in N \qquad (3.2)$$

$$\lambda_i + d_{ij} X_{ij} \le \lambda'_j \qquad \forall (i,j) \in A \qquad (3.3)$$

$$\lambda'_i - \mathcal{M} Y_i \le \lambda_i \qquad \forall i \in N \qquad (3.4)$$

$$\lambda'_i \le \Lambda \qquad \forall i \in N \qquad (3.5)$$

$$X_{ij} \le Z_{ij} \qquad \forall (i,j) \in A : i < j \qquad (3.6)$$

$$X_{ij} \le Z_{ji} \qquad \forall (i,j) \in A : i > j \qquad (3.7)$$

where

X_{ij}, Y_i, and Z_{ij} are binary variables

$\mathcal{M}$ is a large number

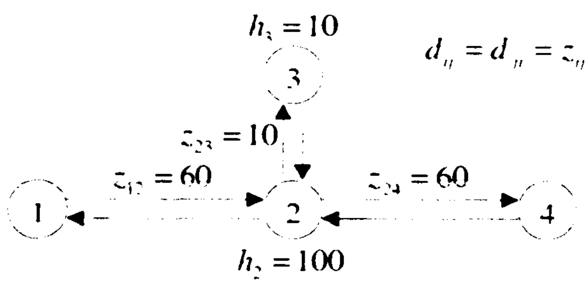

Figure 3-1: Example for the flow conservation model

Although the optimal solution for the example in Figure 3-1 is the path 1-2-3-2-4, with a multiplexer installed at node 3 and with 140 as the total implementation cost, the above formulation would return the path 1-2-3 with a repeater at node 2 and a total cost of 230. To understand the reason for such behaviour, let us assume that $X_{12} = X_{23} = X_{32} = X_{24} = 1$, $Z_{12} = Z_{23} = Z_{24} = 1$, and $Y_3 = 1$ (all other X, Y and Z variables are equal to 0). Consequently, constraint set (4.3) is $\lambda_1 + d_{12}X_{12} \le \lambda'_2 \Rightarrow \lambda_1 + 60 \le \lambda'_2$ for $(i,j) = (1,2)$ and $\lambda_3 + d_{32}X_{32} \le \lambda'_2 \Rightarrow \lambda_3 + 10 \le \lambda'_2$ for $(i,j) = (3,2)$. Therefore, considering that $\lambda'_1 = 0$ and $\lambda'_3 = 0$ (the first, because it is the origin of the signal, the second, because a repeater is installed at node 3), then $\lambda'_2 \ge 60$, and a repeater should be installed at node 2 to send the signal from 2 to 4, leading to a solution with a total cost of 240. Consequently, any solver will prefer the solution with path 1-2-3 and a repeater at node 2, and such formulation has proven to be sub-optimal.

The following formulation uses an approach based on the enumeration of all possible paths between two points in the network. Let $G=(N,E)$ be an undirected network on which the Alberta SuperNet is defined, with $n=|N|$ nodes and $m=|E|$ edges. Let $E'\subset E$ contain all edges available for leasing. Let P_{od} be the set of all possible paths between nodes o and d in N that are shorter than Λ, i.e., the summation of the lengths of individual edges in p is smaller than Λ. Therefore, if $P_{od}\neq\emptyset$, there is a path $p\in P_{od}$ on which a signal can travel from o to d without being regenerated. We define a_{ij}^{p} to be equal to 1 if a path p includes edge $(i,j)\in E$, 0 otherwise.

Let $T=\{1,2,3\}$ represent the technologies available for transmission: GE, SONET and DWDM, represented by 1, 2 and 3, respectively. The node subset $K\subset N$ represents the 422 communities in Alberta and is partitioned into K_1 and K_2; the former represents the 395 access communities, and the latter represents the 27 backbone communities (consequently, $K=K_1\cup K_2$ and $K_1\cap K_2$ is empty). We assume that there is an arbitrary "root node" $root\in K_1$ of the Alberta SuperNet, where all communication is switched. We further introduce the following constants:

c_{ij}^{t}	cost per fibre used along edge $(i,j)\in E$ for communication flow in technology $t\in T$
w^{t}	communication flow capacity per fibre for technology $t\in T$
$o^{tt'}$	communication flow switching capacity given by switch t to t', $t,t'\in T: t\neq t'$
y_{ij}	cost of digging along edge $(i,j)\in E$
x_{ij}	number of fibres available along edge $(i,j)\in E$ for leasing
h_{i}	cost of housing equipment at node $i\in N$

r^t cost of installing a multiplexer for technology $t \in T$

$s^{tt'}$ cost of installing a switcher from t to t', $t, t' \in T : t \neq t'$

δ^t delay added to a signal traversing a multiplexer of technology $t \in T$

$\delta^{tt'}$ delay added to a signal traversing a switch from t to t'

dem_i communication flow demand at node $i \in K \setminus \{root\}$

DEM total communication flow reaching the root node, equal to $\sum_{i \in K \setminus \{r\}} dem_i$

Δ_{max} maximum acceptable signal delay to root

The following are the variables used to model the Alberta SuperNet problem (ASP):

X_{ij}^t = number of fibres along edge $(i, j) \in E$ loaded with communication flow in technology $t \in T$

$Y_{ij} = \begin{cases} 1 & \text{if digging is necessary along edge } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$

$Z_p^t = \begin{cases} 1 & \text{if path } p \in P_{ij} \text{ transports a signal of technology } t \in T \text{ in the final solution} \\ 0 & \text{otherwise} \end{cases}$

$H_i = \begin{cases} 1 & \text{if equipment shelter is necessary at location } i \in N \\ 0 & \text{otherwise} \end{cases}$

W_p^t = communication flow through path $p \in P_{ij}$ for technology $t \in T$

$O_i^{tt'}$ = communication flow through a switch located at node i from technology t to t'

$R_i^t = \begin{cases} 1 & \text{if multiplexer for technology } t \in T \text{ is installed at node } i \in N \\ 0 & \text{otherwise} \end{cases}$

$S_i^{tt'} = \begin{cases} 1 & \text{if a switch from technology } t \text{ to technology } t' \text{ is installed at node } i \in N \\ 0 & \text{otherwise} \end{cases}$

θ_i^t = signal delay accumulated up to node $i \in N$, for technology $t \in T$, before entering multiplexers or switchers located at i

The following is the mathematical model that represents the ASP:

(IP1) Min

$$\sum_{\substack{t \in T \\ (i,j) \in E}} c_{ij}^t X_{ij}^t + \sum_{(i,j) \in E} y_{ij} Y_{ij} + \sum_{i \in N} h_i H_i + \sum_{\substack{i \in N \\ t \in T}} \left(r^t R_i^t + \varepsilon_t \theta_i^t \right) + \sum_{\substack{i \in N \\ t,t' \in T : t \neq t'}} s^{tt'} S_i^{tt'} \quad (3.8)$$

S.t.

$$\sum_{\substack{p \in P_{ji} \\ j \in N \setminus \{i\}}} W_p^t - \sum_{\substack{p \in P_{ij} \\ j \in N \setminus \{i\}}} W_p^t + \sum_{t' \in T \setminus \{t\}} O_i^{t't} - \sum_{t' \in T \setminus \{t\}} O_i^{tt'} = \begin{cases} dem_i & \text{if } t = 1, i \in K_1 \\ -DEM & \text{if } t = 1, i = root \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N, t \in T \quad (3.9)$$

$$\sum_{\substack{p \in P_{od} \\ o,d \in N, o \neq d}} a_{ij}^p W_p^t \leq w^t X_{ij}^t \qquad \forall (i,j) \in E, t \in T \quad (3.10)$$

$$O_i^{tt'} \leq o^{tt'} S_i^{tt'} \qquad \forall i \in N, t,t' \in T : t \neq t' \quad (3.11)$$

$$\sum_{t \in T} X_{ij}^t \leq \mathcal{M}_5 Y_{ij} \qquad \forall (i,j) \in E \quad (3.12)$$

$$X_{ij}^t \leq \mathcal{M}_6 \sum_{\substack{p \in P_{od} \\ o,d \in N, o \neq d}} a_{ij}^p Z_p^t \qquad \forall (i,j) \in E, t \in T \quad (3.13)$$

$$\sum_{t,t' \in T : t \neq t'} S_i^{tt'} \leq \mathcal{M}_7 H_i \qquad \forall i \in N \quad (3.14)$$

$$\sum_{t \in T} R_i^t \leq \mathcal{M}_8 H_i \qquad \forall i \in N \quad (3.15)$$

$$\theta_i^t + \delta^t R_i^t - \mathcal{M}_9 \left(1 - \sum_{\substack{p \in P_{od} \\ o,d \in N, o \neq d}} a_{ij}^p Z_p^t \right) \leq \theta_j^t \qquad \forall (i,j) \in E, t \in T \quad (3.16)$$

$$\theta_j^{t'} + \delta^{t't} - \mathcal{M}_{10} \left(1 - S_j^{t't} \right) \leq \theta_j^t \qquad \forall j \in N, t',t \in T : t \neq t' \quad (3.17)$$

$$\theta_i^t \leq \Delta_{max} \qquad \forall t \in T \quad (3.18)$$

$$\sum_{\substack{o \in N \setminus \{j, root\} \\ p \in P_{oj}}} Z_p^t \leq R_j^t + \sum_{t' \in T : t \neq t'} S_j^{t't} \qquad \forall j \in N \setminus \{root\}, t \in T \quad (3.19)$$

$$\sum_{t \in T} X_{ij}^t \leq x_{ij} \qquad \forall (i,j) \in E' \quad (3.20)$$

where

$W_{ij}^{t}, O_{i}^{tt'}, \theta_{i}^{t}$ are nonnegative real variables

$Y_{ij}, Z_{ij}^{t}, H_{i}, R_{i}^{t}, S_{i}^{tt'}$ are binary variables

X_{ij}^{t} are nonnegative integer variables

$\mathcal{M}_i$ are big numbers associated with constraints i

First, we describe the objective function. The first term, $\sum_{t \in T, (i,j) \in E} c_{ij}^{t} X_{ij}^{t}$, accounts for fibre costs, which may depend on the characteristics of each arc and the technology installed. In general, $c_{ij}^{t} = c \cdot d_{ij}$, i.e., the price is given by a constant times the length of the arc, independent of t. However, if fibre is already available in a trench, the cost should be zero. The second term, $\sum_{(i,j) \in E} y_{ij} Y_{ij}$, corresponds to the fixed cost of digging a trench to install fibres along arc $(i,j) \in E$. The third term, $\sum_{i \in N} h_i H_i$, corresponds to the cost of installing a shelter for equipment at node i. The fourth term, $\sum_{i \in N, t \in T} \left(r^{t} R_{i}^{t} + \varepsilon_1 \theta_{i}^{t} \right)$, is composed of two sub-terms. The first corresponds to the cost of installing a multiplexer for technology t at node i. The second forces signal delay to be minimal at each node; otherwise, the delay may assume the maximum limits dictated by Δ_{max}. The coefficient ε_1 must be as small as possible, to prevent it from influencing the network design outcome. Finally, the last term, $\sum_{i \in N, t,t' \in T, t \neq t'} s^{tt'} S_{i}^{tt'}$, corresponds to the cost of switchers in the final design.

Constraints (3.9) guarantee a flow balance at each node in the network. The first two terms on the *left hand side* (LHS) are for communication flow along path segments that transports the same technology signal. The last two terms on the LHS are for communication flow passing from one technology to another, through switchers. The *right hand side* (RHS) represents the flow demand for each node $i \in N$.

Constraints (3.10) guarantee that there is enough fibre along edge (i,j) (assigned to technology t) for all communication flow over the paths that use edge (i,j) and that transport signal over technology t. Similarly, constraints (3.11) guarantee that the total communication flow from technology t to technology t' at node i does not exceed the capacity of the switcher.

Moreover, the constraints enforce no switching communication, if a switcher is not installed at i (in other words, $O_i^{tt'} = 0$ if $S_i^{tt'} = 0$).

Constraints (3.12) guarantee that fibres will be available only at arcs where digging and fibre installation are accounted for, i.e., $X_{ij}^t > 0$ for any t, only if $Y_{ij} = 1$. Constraints (3.13) correlate variables X_{ij}^t with variables Z_p^t; this means that whenever $X_{ij}^t > 0$, there will be at least one path $p \in P_{od} : o, d \in N, o \neq d$ using edge $(i, j) \in E$ selected (i.e., $Z_p^t = 1$). Although Z_p^t variables are not present in the objective function, they are used in constraints (3.16) and (3.20) to calculate θ variables and to define the presence of repeaters and multiplexers.

Constraints (3.14) and (3.15) guarantee that switchers and multiplexers will be installed at node i, only if equipment shelters are also installed at i (in other words, if $H_i = 1$).

Variables θ_j^t represent the longest delay accumulated so far on signals of technology t reaching node j. Constraints (3.16) consider each node i reaching node j through an edge (i, j). If there is a communication flow from i to j in technology t over a path p that uses edge (i, j), then $Z_p^t = 1$. Consequently, (3.16) can be reduced to $\theta_i^t + \delta^t R_i^t \leq \theta_j^t$, and θ_j^t must be at least equal to the delay for t at node i plus any delay incurred by the signal, if a multiplexer for technology t is installed at node i. Otherwise, if there is no communication flow from i to j in t, $Z_p^t = 0$, and (3.16) is valid for any $\theta_j^t \geq 0$, as long as $\mathcal{M}_9$ is sufficiently large.

Constraints (3.17) work in a similar fashion. Each communication flow is switched from t' to t at node j through a switcher of type $(t', t) \in P$. Whenever switchers are installed ($S_j^{t't} = 1$), constraints (3.17) can be reduced to $\theta_j^{t'} + \delta^{t't} \leq \theta_j^t$. Also, θ_j^t must be at least equal to the delay of the signal in technology t', which enters node j and is converted to technology t, plus the delay incurred in the switching process, given by $\delta^{t't}$. Otherwise, (3.17) is valid for any $\theta_j^t \geq 0$, as long as $\mathcal{M}_{10}$ is sufficiently large.

Constraints (3.18) simply guarantee that no signal entering the *root* will violate the maximum communication delay Δ_{max}.

Constraints (3.19) enforce the presence of either a repeater for technology t or a switcher from t to t' at node j, if any path reaches node j that has been selected to transport a signal over technology t.

The last set of constraints (3.20) imposes a limit on the number of fibres available at each edge $(i, j) \in E$. This set of constraints can only be written for some edges, where fibres for leasing are available. Constraints (3.20) are not used for edges that represent the option of laying optical fibre.

The above formulation details the constraints discussed in Section 2.3. A summary is provided below:

- *the presence of mandatory and non-mandatory nodes* is handled by the flow balance constraints (3.9), as a non-mandatory node is not obliged to have any communication flow traversing it;

- *constraints on the distance between equipment for signal regeneration* is covered implicitly through the path sets P_{ij} and constraints (3.19);

- *the need to limit signal delay caused by using different types of hardware and protocols* is covered by constraints (3.16) to (3.18);

- *the possibility of carrying signals of different protocols in parallel* is handled by the model, because no constraint forces only one protocol on an output line; and

- *the option of either laying or leasing optical fibres* is considered by creating parallel arcs representing fibre available to lease by setting their y_{ij} equal to leasing costs and by using constraints (3.20) to limit the number of optical fibres available for the links. Constraints (3.20) are not used for arcs that represent the option of laying optical fibre.

3.2 Solution Methodology

Due to its complexity, we divide the Alberta SuperNet problem into simpler sub-problems: the *Network Design with Repeaters* (NDwR) and the *Network Loading and Technology Selection* (NLTS) problems. We then solve them in sequence to obtain a feasible solution to the ASP. Solving the NDwR involves solving a sub-problem called the *Constrained Shortest Path with Resource Regeneration Problem* (CSPwRRP).

Chapter 4 describes the CSPwRRP, which deals with finding a minimum cost path and repeater locations, while respecting the maximum constraint on distance given by Λ. A label correcting algorithm is proposed to solve the problem.

For the NDwR, described in detail in Chapter 5, we first deal with the problem of locating equipment and selecting which arcs to place fibre along, while accounting for the *constraint on distance between equipment for signal regeneration*. An *integer programming* model is formulated, in which three sets of variables are presented. The first two sets describe equipment and fibre installation; the last set represents communication flow paths with associated equipment installation patterns. Each path/repeater pattern is guaranteed to respect the distance between equipment constraints. This last set of variables grows exponentially, according to the size of the network, and therefore, a complete list is impractical. Instead, a Dantzig-Wolfe decomposition scheme is used, listing necessary variables and obtaining the optimal solution to the relaxed version of the IP model. Two heuristics are presented to solve the NDwR, with computational performance evaluated at the end.

The NLTS, described in detail in Chapter 6, deals with the *necessity of limiting signal delay caused by the usage of different types of hardware and protocols*, and the *possibility of carrying signals of different protocols in parallel*; it is an extension to the branch of network design called the *Network Loading Problem* (NLP). In this second sub-problem, we present a mathematical formulation and describe a tabu search algorithm to solve the NLTS problem. NLTS assumes that hardware installation is mandatory at the nodes, and therefore, a simplified graph can be generated using the solution presented by the NDwR.

Finally, in Chapter 7, we conclude this dissertation and present ideas for future work involving NDwR and NLTS to solve the ASP.

4 Constrained Shortest Path with Resource Regeneration Problem (CSPwRRP)

4.1 Introduction

In this chapter we consider the simplest version of the network design problem, where only two nodes are to be connected, and the signal must be regenerated. Although simple, this problem has not been identified in the research literature. The problem is to find a minimum cost path between two nodes, while respecting a constraint on the maximum distance between repeaters. We call it the *Constrained Shortest Path with Resource Regeneration Problem* (CSPwRRP). This problem appears primarily in telecommunication network design contexts, where a signal sent from an origin to a destination must be regenerated every Λ units of distance to continue its travel. Another application of the CSPwRRP may occur in transportation contexts, where either a vehicle needs to be refuelled given a limited autonomy, or certain services must be provided to the transported commodities after some travel time.

Regeneration *per se* is often disregarded in the telecommunication and transportation science literature. We identified only two articles that consider the recharge of resources: Kuby and Lim (2003); and Bapna, Thakur, and Nair (2002). The former considers the problem of optimally locating alternative-fuel stations. The latter models the dynamics involved in changing from leaded to unleaded fuel at gas stations. However, these articles simplify the CSPwRR problem. Despite the lack of references to the regeneration problem, the *Constrained Shortest Path Problem* (CSPP), has been extensively researched and is a special case of the CSPwRR problem. The CSPP finds the least-cost path between two specified nodes, such that the total length is less than Λ (a more general description can be found in Dumitrescu and Boland 2003, among others). The CSPP problem is NP-hard (Garey and Johnson 1979, p. 214), but can be solved in pseudo-polynomial time, according to Dumitrescu and Boland (2003). Consequently, the CSPwRRP is also NP-hard, but can be solved in pseudo-polynomial time, as we will present later on.

The major approaches to the algorithms developed for the CSPP are based on dynamic programming (Joksch 1966). These algorithms were supplanted more recently by vertex-labelling algorithms (Aneja, Aggarwal, and Nair 1983, Dumitrescu and Boland 2003), branch-and-bound algorithms using a Lagrangian-based bound (Beasley and Christofides 1989), and a Lagrangian relaxation coupled with K-shortest path enumeration algorithms (Handler and Zang 1980). Most recently, Dumitrescu and Boland (2003) presented a vertex-labelling algorithm with several pre-processing techniques, which efficiently solves the CSPP. Carlyle and Wood (2003) devise an algorithm based on Lagrangian Relaxation and compare it to Dumitrescu and Boland (2003), obtaining faster results for the smallest of the test sets. These are the major trends for solving the CSPP, and the reason that we based our algorithm on the vertex-labelling algorithms.

Let us now formally define CSPwRRP. In an undirected network $G=(N,E)$ with $n=|N|$ nodes and $m=|E|$ edges, repeater housing costs h_i are defined for each node $i\in N$; edge length d_{ij} and trenching cost c_{ij} are defined for each edge $(i,j)\in E$, with $d_{ij}\leq\Lambda$; finally, we define an origin node $o\in N$, a destination node $d\in N$, and a maximum distance between repeaters Λ. The CSPwRRP then searches for a path p from o to d and for a repeater node set $r\subseteq N\setminus\{o,d\}$ with nodes in p that minimizes edge and repeater installation costs, while respecting the constraints on the maximum distance between repeaters. One can easily notice that the CSPP is a special case of the CSPwRR problem by simply using $h_i=\infty$ for all $i\in N$. Consequently, if the CSPP has no feasible solution, the CSPwRRP (with $h_i=\infty$) will return ∞ as the value of an optimal solution. If CSPP is infeasible, then no o-d path has length $\leq\Lambda$, but there may be some sub-paths from i to j ($i,j\in N, i\neq o$ or $j\neq d$) that have lengths $\leq\Lambda$, i.e. CSPwRRP (with $h_i\leq\infty$) might still be feasible. Therefore, the infeasibility of CSPP does not imply infeasibility of CSPwRRP. However, the opposite is true.

A path in the network is a sequence of nodes $P=\{i_0,i_1,\ldots,i_p\}$ such that $(i_{k-1},i_k)\in E$ for all $k=1,\ldots,p$. In some instances, P will be considered a set of arcs instead, depending on the context of the formulation. A repeater pattern $R\subset P\setminus\{i_0,i_p\}$ can be associated with a given path P, representing repeaters installed along P that provide signal regeneration at nodes in R.

In that case, path P is composed of $|R|+1$ sub-paths P_t, $t=0,\ldots,|R|$, with P_0 being the sub-path from origin node o to the first repeater node in path P, P_1 being the sub-path from the first repeater node in P to the second repeater node in P, and so on. The last sub-path, P_R, is the segment from the last repeater in P to the destination node d. If R is empty, then by convention $P_0 = P$.

Two criteria are evaluated in the CSPwRRP context: the *length* of a path, and the *cost* of a path with associated repeater installations. We denote $f(P,R)=\sum_{(i,j)\in P} c_{ij} + \sum_{i\in R} h_i$ as the total cost of implementing path P combined with repeater pattern R, and $g(P)=\sum_{(i,j)\in P} d_{ij}$ as the total length of path P. We similarly define those functions for sub-paths, with $f(P_t)=\sum_{(i,j)\in P_t} c_{ij}$ as the total cost of implementing sub-path P_t, and $g(P_t)=\sum_{(i,j)\in P_t} d_{ij}$ as the total length of sub-path P_t.

Given these definitions, a path P combined with a repeater pattern R (*path/repeater pattern* P and R) is considered *feasible* if each sub-path $P_t \in P$ has a total length $g(P_t)\leq \Lambda$. If we were able to enumerate all feasible paths/repeater patterns P and R connecting o and d, an optimal solution to the CSPwRRP would be one for which $f(P,R)$ is minimal. However, as the number of paths grows exponentially with the graph size, enumerating all possible paths/repeater patterns becomes undesirable for large graphs.

A lower bound to the CSPwRRP can be obtained by solving the minimum cost path problem over G and disregarding the distance constraints, obtaining a path P'. If $g(P')\leq \Lambda$, then P' with no repeaters is the optimal solution to the CSPwRR problem. If $g(P')>\Lambda$, then P' with no repeaters is a lower bound to the CSPwRRP.

An upper bound can be obtained using $R'=P'\setminus\{o,d\}$. However, this upper bound may be poor, especially if link lengths on P' are short or repeaters costs along P' are expensive.

Although it is possible to improve this upper bound by dropping some of the repeaters, additional effort is required to guarantee feasibility.

To illustrate the CSPwRRP problem, we refer to example G in Figure 4-1. Our goal is to go from node 1 to node 6, using $\Lambda = 70$ and the simplifying assumption that $c_{ij} = d_{ij}$.

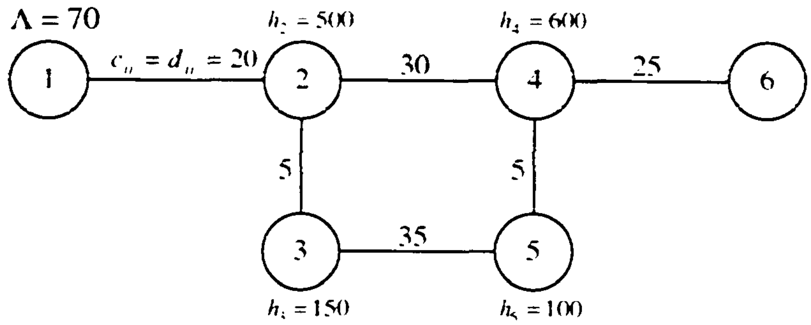

Figure 4-1: Graph *G*

With $c_{ij} = d_{ij}$, the shortest path from 1 to 6 is also the minimum cost path from 1 to 6. Therefore $P = \{1,2,4,6\}$ with $f(P,\varnothing)$ and $g(P)$ equal to 75. As $g(P) \geq \Lambda$, P is infeasible for the example in Figure 4-1, and a lower bound of 175 can be obtained by adding $g(P)$ to the smallest repeater housing cost in G, $h_5 = 100$ (as $\lceil 1.071 - 1 \rceil = 1$, only one repeater is necessary). To render P feasible, a repeater can be installed either at node 2 or at node 4. As $h_2 > h_4$, the best repeater pattern for $P = \{1,2,4,6\}$ is $R = \{2\}$, with a total cost of 575. Another feasible path/repeater pattern solution would be $P = \{1,2,3,2,4,6\}$, $R = \{3\}$ and $f(P,R) = 235$. Complete enumeration yields the best path/repeater pattern: $P = \{1,2,4,5,4,6\}$, $R = \{5\}$ and $f(P,R) = 185$. Note that node 4 appears twice in the path. In general, optimal paths are not limited to simple paths. However, as all costs are assumed positive, the formation of an infinite cycle in the final solution is avoided, and each cycle will appear only once along the optimal path. First, we prove that a cycle only appears if a repeater is installed in one of the nodes inside the cycle. This proof can be extended for cycles with more than 2 nodes, as long as one realizes that the repeater must be located exactly in the middle of the cycle. Otherwise the solution would be sub-optimal.

Proposition 1: If an optimal path/repeater pattern P and R has a cycle $\{i, j, i\}$, j must have a repeater (i.e., $j \in R$).

Proof: Let $P = \{o, \ldots, r_t, \ldots, i, j, i, \ldots, r_{t+1}, \ldots, d\}$ be the optimal path for the CSPwRRP from origin node $o \in N$ to destination node $d \in N$ defined in G, and nodes r_t and r_{t+1} have repeaters installed forming sub-path $P_t = \{r_t, \ldots, i, j, i, \ldots, r_{t+1}\}$ (t is an integer between 1 and $|R| - 1$). Let us also assume that $c_{ij} = c_{ji} > 0$. Sub-path P_t is feasible, i.e., $g(P_t) \leq \Lambda$, and has the cost

$$f(P_t) = f(\{r_t, \ldots, i\}) + c_{ij} + c_{ji} + f(\{i, \ldots, r_{t+1}\}).$$

However, sub-path $P_t' = \{r_t, \ldots, i, \ldots, r_{t+1}\}$ is also feasible and has the cost

$$f(P_t') = f(\{r_t, \ldots, i\}) + f(\{i, \ldots, r_{t+1}\}) \leq f(P_t).$$

Consequently, there must be a feasible path/repeater pattern $P' = \{o, \ldots, r_t, \ldots, i, \ldots, r_{t+1}, \ldots, d\}$ and R in G with a total cost smaller than or equal to P and R, contradicting the fact that P and R is an optimal path/repeater pattern. Hence, if an optimal path P contains a cycle $\{i, j, i\}$, node j must contain a repeater. ■

This proof can be extended for cases where r_t is replaced by o, and r_{t+1} is replaced by d. Now, we prove that an infinite cycle is prevented if costs are strictly positive.

Proposition 2: If an optimal path/repeater pattern P and R has a cycle $\{i, j, i\}$, this cycle appears only once in P.

Proof: Let $P = \{o, \ldots, i, r_t = j, i, \ldots, i, r_{t+1} = j, i, \ldots, d\}$ be the optimal path for the CSPwRRP in G, where cycle $\{i, j, i\}$ appears twice. We know from proposition 1 that node j must contain a repeater, and therefore, we have a sub-path $P_t = \{r_t = j, i, \ldots, i, r_{t+1} = j\}$, and we know that

$$f(P,R) = f(\{o,\ldots,r_t\},\{r_1,\ldots,r_{t-1}\}) + f(P_t) + h_{r_t} + f(\{r_{t+1},\ldots,d\},\{r_{t+2},\ldots,r_R\}).$$

However, one can see that path $P' = (o,\ldots,i,r_t = j,i,\ldots,d)$ is also feasible with the cost

$$f(P',R) = f(\{o,\ldots,r_t\},\{r_1,\ldots,r_{t-1}\}) + h_{r_t} + f(\{r_{t+1},\ldots,d\},\{r_{t+2},\ldots,r_{|R|}\}) \leq f(P,R),$$

contradicting the optimality of P and R. Therefore, if an optimal path P contains a cycle $\{i,j,i\}$, this cycle cannot appear more than once in P. ∎

4.2 Integer Programming formulation

To represent the CSPwRR problem as an *Integer Programming* (IP) model, the undirected graph $G = (N,E)$ must be transformed into a directed graph $H = (N,A)$. Each arc $(i,j) \in A$ represents the optimal CSPP solution going from node i to node j in G, with $i \neq d$ and $j \neq o$, because CSPwRRP solutions only leave the origin node o and only arrive at destination node d. Consequently, $A = \{(i,j) : \exists \text{ a CSPP solution from } i \text{ to } j, \text{ with } i \neq d \text{ and } j \neq o\}$. Traversal cost p_{ij} and length ℓ_{ij} are defined for each arc $(i,j) \in A$, which correspond to the CSPP path cost and length, respectively. According to Dumitrescu and Boland (2003), the CSPP can be solved by a label-setting algorithm with a computational complexity $O(|E|\Lambda)$, if appropriate data structures are used, and there are no zero cost values in G. Consequently, the algorithm that transforms G into H has the complexity $O(|N||E|\Lambda)$. Figure 4-2 presents the equivalent graph H for the example in Figure 4-1.

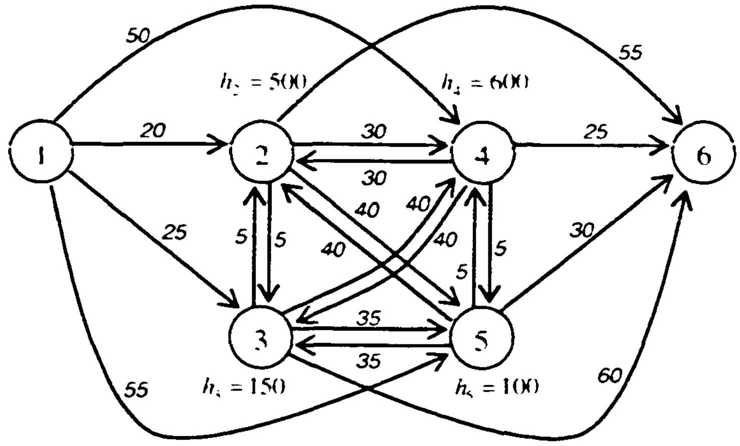

Figure 4-2: Graph H for the example in Figure 4-1

The following IP problem defined on graph H solves the CSPwRRP for graph G:

(IP1) Min

$$\sum_{(i,j)\in A} l_{ij} x_{ij} + \sum_{i\in N} h_i y_i \tag{4.01}$$

S.t.

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = \begin{cases} 1 & \text{if } i = o \\ -1 & \text{if } i = d \\ 0 & \text{o.w.} \end{cases} \qquad \forall i \in N \tag{4.02}$$

$$x_{ij} \le y_i \qquad \forall i \in N \setminus \{o,d\}, \forall (i,j) \in A \tag{4.03}$$

x_{ij} and y_i are binary variables

Variables x_{ij} are equal to 1, if an arc $(i,j)\in A$ is part of the minimum cost path with repeaters from o to d, 0 otherwise. Variables y_i are equal to 1, if a repeater is installed at node $i \in N$, 0 otherwise.

Constraints (4.02) guarantee a flow balance at each node of the network, assuring that there is a continuous path from o to d, if the graph contains a feasible path.

In the formulation above, constraints (4.03) force the installation of repeaters at each node of a path on H connecting o to d. One may argue that placing a repeater at every node in a path in H is sub-optimal; but we demonstrate that this is not so. We start by verifying that a given path P, where repeaters are installed in every node on the path in H, is a *feasible* path in G. As arcs in A represent constrained shortest paths that respect the distance constraint of Λ, placing repeaters at all nodes on the path guarantees that no path segment between a pair of adjacent repeaters will be longer than Λ. Now, we prove that the solution provided by IP1 would not place a repeater in the optimal path solution unnecessarily.

Proposition 3: If $x_{ij} = 1$ and $j \neq d$, then $y_j = 1$.

Proof: Let $P = \{o,...,i,j,k,...,d\}$ in H be the optimal path for the CSPwRRP from origin node $o \in N$ to destination node $d \in N$, i.e., P is the minimum cost path in H with a total distance between repeaters smaller than or equal to Λ. Assume that P has no repeater installed at node j. This means that $\ell_{ij} + \ell_{jk} \leq \Lambda$, and in this case, there must be an arc (i,k) in H with a total cost $p_{ik} \leq p_{ij} + p_{jk}$. Consequently, there must be a path $P' = (o,...,i,k,...,d)$ in H with a total cost smaller than or equal to P, contradicting the fact that P is optimal. Thus, if $x_{ij} = 1$ and $j \neq d$, then $y_j = 1$. ■

Proposition 3, therefore, guarantees that IP1 formulates CSPwRRP correctly.

4.3 Algorithms Proposed

In this section, we propose three algorithms to solve the CSPwRRP, one based on graph expansion and two using a label-correcting algorithm. The first algorithm (Section 4.3.1) extends the graph transformation that we presented previously for the IP formulation of the CSPwRRP. The second algorithm (Section 4.3.2) uses a label-correcting algorithm on the original network that maintains Pareto optimal solutions, while considering the repeaters in the network. The third algorithm (Section 4.3.3.) uses a label-correcting algorithm combined with a merge-sort structure that improves the theoretical computational complexity of the algorithm.

4.3.1 CSPP-based Algorithm

The first algorithm expands $H=(N,A)$ by splitting each node $i\in N\setminus\{o,d\}$ into two nodes, i' and i'', creating an equivalent graph $H'=(N',A')$, with $|N'|=2|N|-2$ nodes and $|A'|=|A|+|N|-2$ arcs. Arc set A' is composed of five distinguishable sets of arcs. The first set is formed by all arcs leaving the origin node $o\in N$; each arc $(o,j)\in A$ has an equivalent arc $(o,j')\in A'$. The second set is formed by all arcs reaching the destination node $d\in N$, where each arc $(i,d)\in A$ has an equivalent arc (i'',d) in A'. The third set is formed by all other arcs $(i,j)\in A$ where $i\neq d$ and $j\neq o$, with arc (i'',j') defined in A'. The fourth set of arcs in A' represents the cost of installing a repeater at node $i\in N\setminus\{o,d\}$. An arc (i',i'') is defined in A' with a length of zero and a cost of h_i associated to it. Finally, the last is the set of arc (o,d), if this arc is in A. Hence,

$$N':=\{o,d\}\cup\bigcup_{i\in N-\{o,d\}}\{i',i''\}$$

and

$$A':=\{(o,j'):(o,j)\in A\}\cup\{(i'',d):(i,d)\in A\}\cup\{(i'',j'):(i,j)\in A, i\neq o, j\neq d\}\cup$$
$$\{(i',i''):i\in N-\{o,d\}\}\cup\{(o,d):(o,d)\in A\}$$

The transformation from H to H' has a complexity of $O(|A|+|N|)$.

Figure 4-3 shows the equivalent graph H' for the example presented in Figure 4-2, where arcs in bold correspond to the fourth set defined in A'.

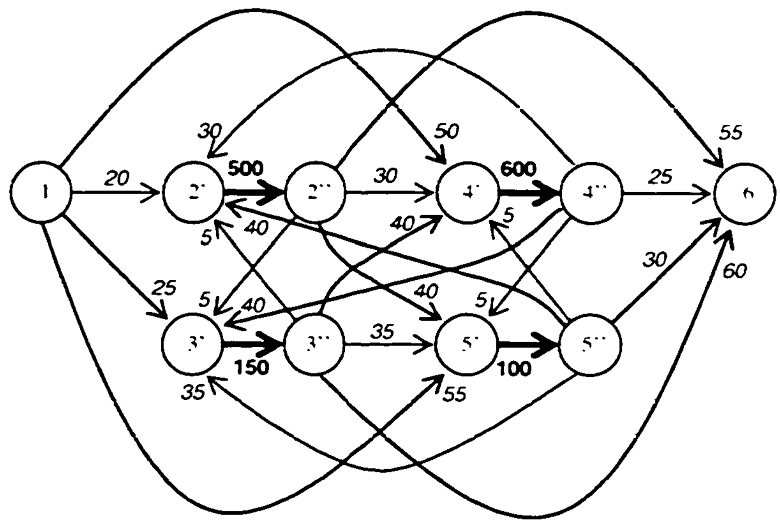

Figure 4-3: Graph H' for the example from Figure 4-1

Once graph H' is obtained, we solve the minimum cost path problem in H' to obtain the final solution to the CSPwRRP. This last step can be done using the Dijkstra algorithm, which has a complexity of $O(|A'| + |N'|\log|N'|)$, if implemented with the Fibonacci Heap (Ahuja *et al.* 1993, p. 122). Consequently, the overall complexity of the CSPP-based algorithm can be calculated as

$$O(|N||E|\Lambda + |A'| + |N'|\log|N'|) \equiv O(|N||E|\Lambda).$$

4.3.2 Label-Correcting Algorithm

The second proposed algorithm is based on a label-correcting algorithm, with a set of labels defined for each node. Each label for node i in N, $(H_i, \lambda_i, f_i, pred\_ptr_i)$, represents a different path from node o to node i and is composed of 4 attributes: H_i represents repeater installation $(H_i = 1)$ or not $(H_i = 0)$ at node i; λ_i is the total distance from the last repeater to node i (if $H_i = 1$, then $\lambda_i = 0$); f_i is the total cost of implementing this path from o to i, respecting Λ and including the repeater installation at i; and $pred\_ptr_i$ is a pointer to the label predecessor to this label.

Definition: We say that a label $(H'_j, \lambda'_j, f'_j, pred\_ptr'_j)$ is *dominated* by another label $(H''_j, \lambda''_j, f''_j, pred\_ptr''_j)$, if $\lambda'_j > \lambda''_j$ and $f'_j \geq f''_j$ or $\lambda'_j \geq \lambda''_j$ and $f'_j > f''_j$. In this case, we say that $(H''_j, \lambda''_j, f''_j, pred\_ptr''_j)$ *dominates* $(H'_j, \lambda'_j, f'_j, pred\_ptr'_j)$. Moreover, we say that a label $(H''_j, \lambda''_j, f''_j, pred\_ptr''_j)$ is *nondominated*, if no other label $(H'_j, \lambda'_j, f'_j, pred\_ptr'_j)$ dominates it.

During the algorithm's run, it maintains a set of non-dominated labels Γ_j for each node $j \in N$ and a list SE (scan eligible) of labels with the potential to improve the labels of at least one another node. At each iteration of the algorithm, a label called *current label* is removed from the SE list. This current label can be selected by following the First-In First-Out (FIFO) rule. Initially, the source label is defined as the current label. During the scanning process of the current node, the temporary labels of all successor nodes of the current node are calculated. As the algorithm progresses, the set of label Γ_j of each successor node j of the current node i is updated so that Γ_j contains only non-dominated labels. The cardinality of Γ_j may decrease or increase as the algorithm proceeds. If a temporary label enters a given Γ_j, it is inserted into the SE list. The algorithm terminates once an iteration is completed and the SE list is empty. Upon the termination, the set Γ_j contains the labels that denote all non-dominated optimal paths from the source node o to node j. The pseudocode of the algorithm is given in Table 4-1.

Step 0: $\Gamma_j := \begin{cases} \{(0,0,0,\varnothing)\} & \text{if } j = o \\ \varnothing & \text{o.w.} \end{cases}$; $SE := \{(o,(0,0,0,\varnothing))\}$

Step 1:

while $SE \neq \emptyset$ do

{ choose a current pair node and label $(i^*, \pi) = (i^*, (H_i^*, \lambda_i^*, f_i^*, pred\_ptr_i^*))$ from SE

$SE := SE - \{i^*, \pi\}$

for all $(i^*, j) \in E$ do // the scanning process

{ if $j = d$ or $h_j = \infty$ then $\Omega := \{0\}$

else $\Omega := \{0,1\}$

{ if $\lambda_i^* + d_{i^*j} \le \Lambda$ then

for all $H_j \in \Omega$ do

{ if $H_j = 1$ then $\lambda := 0$; else $\lambda := \lambda_i^* + d_{i^*j}$

$f_j := h_j H_j + c_{i^*j} + f_i^*$

$\pi_j = \left(H_j, \lambda_j, f_j, pred\_ptr_j\right)$

delete all $\left(H'_j, \lambda'_j, f'_j, pred\_ptr'_j\right) \in \Gamma_j$ dominated by π_j

if π_j is not dominated by any label in Γ_j, then

insert π_j in Γ_j and $\left(j, \pi_j\right)$ in SE

}

}

}

}

Table 4-1: Label-correcting algorithm to solve the CSPwRRP

The iteration process of the algorithm, when applied to the numerical example in Figure 4-1, is summarized as follows:

Initial stage:

$\Gamma_1 := \{(0,0,0,\varnothing)\}$; $SE := \{(1,(0,0,0,\varnothing))\}$

$\Gamma_2 = \Gamma_3 = \Gamma_4 = \Gamma_5 = \Gamma_6 = \varnothing$

At the end of iteration 1:

$\left(i^*, \left(H_i^*, \lambda_i^*, f_i^*, pred\_ptr_i^*\right)\right) = (1,(0,0,0,\varnothing))$

$\Gamma_1 = \{(1,(0,0,0,\varnothing))\}$

$\Gamma_2 = \{(1,0,520,(1,(0,0,0,\varnothing))),(0,20,20,(1,(0,0,0,\varnothing)))\}$

$\Gamma_3 = \Gamma_4 = \Gamma_5 = \Gamma_6 = \varnothing$

$SE = \{(2,(0,20,20,(1,(0,0,0,\varnothing)))),(2,(1,0,520,(1,(0,0,0,\varnothing))))\}$

At the end of iteration 2:

$\left(i^*,\left(H_i^*,\lambda_i^*,f_i^*,pred\_ptr_i^*\right)\right)=(2,(0,20,20,(1,(0,0,0,\varnothing))))$

$\Gamma_1=\{(1,(0,0,0,\varnothing))\}$

$\Gamma_2=\{(1,0,520,(1,(0,0,0,\varnothing))),(0,20,20,(1,(0,0,0,\varnothing)))\}$

$\Gamma_3=\{(1,0,175,(2,(0,20,20,\_))),(0,25,25,(2,(0,20,20,\_)))\}$

$\Gamma_4=\{(1,0,650,(0,20,20,\_)),(0,50,50,(0,20,20,\_))\}$

$\Gamma_5=\Gamma_6=\varnothing$

$SE=\{(2,(1,0,520,\_)),(3,(0,25,20,\_)),(3,(1,0,175,\_)),(4,(0,50,50,\_)),(4,(1,0,650,\_))\}$

After 18 iterations, we get:

$\left(i^*,\left(H_i^*,\lambda_i^*,f_i^*,pred\_ptr_i^*\right)\right)=(6,(0,30,185,\_))$

$\Gamma_1=\{(1,(0,0,0,\varnothing))\}$

$\Gamma_2=\{(1,0,520,(1,(0,0,0,\varnothing))),(0,5,180,(3,(1,0,175,\_))),(0,20,20,(1,(0,0,0,\varnothing)))\}$

$\Gamma_3=\{(1,0,175,(2,(0,20,20,\_))),(0,25,25,(2,(0,20,20,\_)))\}$

$\Gamma_4=\{(1,0,650,(2,(0,20,20,\_))),(0,5,160,(5,(1,0,155,\_))),(0,50,50,(2,(0,20,20,\_)))\}$

$\Gamma_5=\{(1,0,155,(4,(0,50,50,\_))),(0,55,55,(4,(0,20,20,\_)))\}$

$\Gamma_6=\{(0,25,675,(4,(1,0,650,\_))),(0,30,185,(4,(0,5,160,\_)))\}$

$SE=\varnothing$

The algorithm stops and the optimal label for node 6, with respect to the total cost, is $(0,30,185,(4,(0,5,160,\_)))$, representing an optimal path 0-1-2-4-5-4-6, with a repeater in node 5 and a total cost of 185. The optimal label path for node 2 is given by $(0,20,20,(1,(1,0,0,\_)))$, as it has the smallest cost of the other labels of node 2 in Γ_2. All other optimal label paths can be similarly obtained; notice that the Γ_i's are ordered in increasing λ values and decreasing $f(\ )$ values. Consequently, the optimal label for each node j is given by the last label in each Γ_j. This second algorithm has a complexity of $O\left(|N|^2\Lambda^2\right)$.

4.3.3 Merge Label-Correcting Algorithm

The third algorithm is a modification of the one presented in Section 4.3.2. This algorithm maintains a set of non-dominated labels for each node $j \in N$. However, instead of maintaining a Scan Eligible (SE) list of labels, it maintains a SE list of nodes that have the potential to improve the labels of at least one other node. This idea is based on the algorithm presented by Brumbaugh-Smith and Shier (1989) for the bicriterion shortest path problem, a simplified version of the CSPP. At the initialization step, only the source node is inserted in the SE list. The labelling process moves forward from the source to all other nodes.

At each iteration, a node i is removed from the SE list and becomes the current node for a label scanning process. In our implementation, we follow the First-In First-Out (FIFO) rule, although other implementations could have been used (see Brumbaugh-Smith and Shier 1989). During the scanning process of the current label, a temporary set of labels Γ_{temp} is created for each node j, which is a successor of i and uses all labels present in Γ_i as starting points. A label enters Γ_{temp}, only if there is no element inside Γ_{temp} that dominates the label. Moreover, labels in Γ_{temp} that are dominated by this candidate label are automatically eliminated from Γ_{temp}. At the end, Γ_{temp} contains all non-dominated labels for node j that can be calculated from Γ_i. After obtaining Γ_{temp}, the algorithm applies the process $\text{Merge}(\Gamma_j, \Gamma_{temp})$ described in Brumbaugh-Smith and Shier (1989), which has a computational complexity of $O(|\Gamma_j| + |\Gamma_{temp}|)$. Both Γ_j and Γ_{temp} are ordered according to the first criteria, as required by the Merge algorithm. The pseudo-code of the Merge Label-Correcting Algorithm is provided in Table 4-2.

Step 0: $\Gamma_j := \begin{cases} \{(0,0,0,\varnothing)\} & \text{if } j = o \\ \varnothing & \text{o.w.} \end{cases}$; $SE := \{o\}$

Step 1:

while $SE \neq \emptyset$ do

{ choose a current node i^* from SE

$SE := SE - i^*$

for all $(i^*, j) \in E$ do // the scanning process

```
{     Γ_temp := ∅
      for all labels (i*, π := (H_i*, λ_i*, f_i*, pred_ptr_i*)) ∈ Γ_i do
{     if j = d or h_j = ∞ then Ω := {0}
      else Ω := {0,1}
      {     if λ_i* + d_ij ≤ Λ then
      for all H_j ∈ Ω do
      {     If H_j = 1 then λ_j := 0; else λ_j := λ_i* + d_ij
            f_j := h_j H_j + c_ij + f_i*
            π_j = (H_j, λ_j, f_j, pred_ptr_j)
            delete all (H'_j, λ'_j, f'_j, pred_ptr'_j) ∈ Γ_temp dominated by π_j
            if π_j is not dominated by any label in Γ_temp, then
                  insert π_j in Γ_temp
      }
}
Γ'_j := Merge(Γ_j, Γ_temp)
if Γ'_j ≠ Γ_j then
{     Γ_j := Γ'_j
      SE := SE + {j}
}
   }
}
```

Table 4-2: Merge label-correcting algorithm

This third algorithm has total complexity of $O(|N||\Lambda|\log|\Lambda|)$, which is better than the complexity of the previous two algorithms.

4.4 Computational Results

All computational tests were carried out on a Sun Fire 480R station with four 900 MHz processors, 16 gigabytes of RAM and a Sun Solaris 5.7 operational system. The algorithms were coded in C++ and compiled with *Sun Forte Developer 7 C++* compiler.

The test networks are equivalent to those in Problem Class 4, defined by Dumitrescu and Boland (2003) and later used in Carlyle and Wood (2003). These test networks have a grid structure, with a rows and b columns, and randomly (uniform) generated integer values for length and costs. Origin and destination nodes o and d are defined outside the grid: o is connected to all nodes in the leftmost column, and d is connected to all nodes in the rightmost column of the grid through edges with zero length and zero cost. The adjacency degree adj dictates the number of pairs of edges adjacent to each node. When $adj = 2$, each node u inside the grid is connected by edges to its four adjacent nodes (left, up, right and down, whenever possible), with length and cost values selected from [10, 30] for vertical edges and horizontal edges. This produces the test network described by Dumitrescu and Boland (2003). When $adj = 3$, we add two more edges with length and cost values selected from [10, 30], diagonally, from upper left to lower right. When $adj = 4$, we add two more edges with length and cost values selected from [10, 30], diagonally, from upper right to lower left. In Figure 4-4 we present the test networks that we generated using $a = 3$, $b = 4$, and the three possible values of adj. Table 4-3 summarizes all tests for the three algorithms. Each row contains computational results for 50 instances for each a, b, adj, and Λ value and for a and b between 10 and 50. Table 4-4 expands the test set sizes for the last two algorithms, as their computational time was relatively low, compared to the CSPP-based algorithm, for a and b between 60 and 100. The sizes of the graphs are summarized by the number of nodes $|N|$ and the number of edges $|E|$.

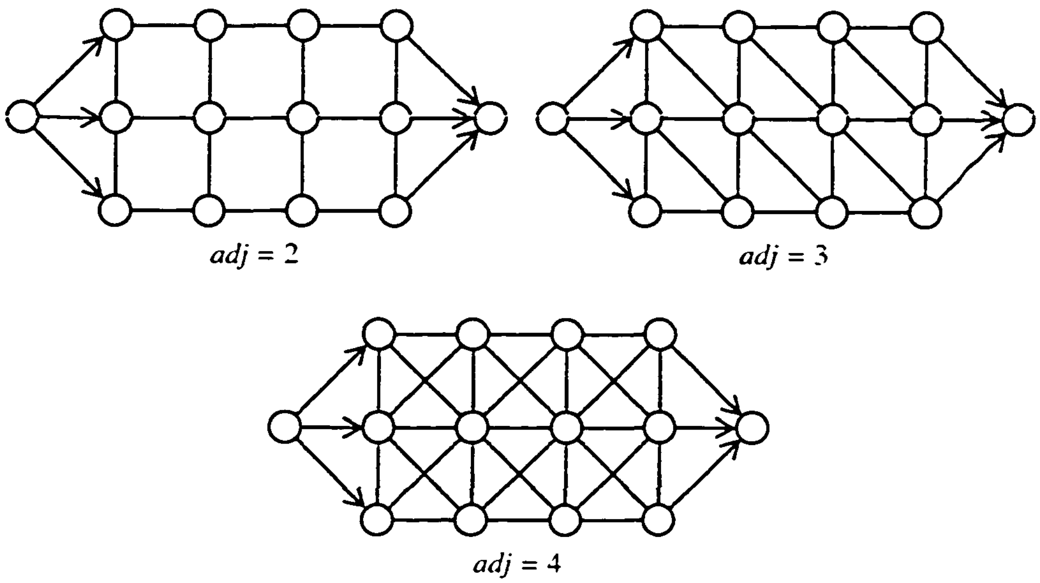

Figure 4-4: Examples of different *adj* values

						CSPP-based Time (sec)		Label-Correcting Time (sec)		Merge L.-C. Time (sec)	
a	*b*	*adj*	*A*	*|N|*	*|E|*	*avg*	*stdev*	*avg*	*stdev*	*avg*	*stdev*
10	10	2	50	102	180	0.1	0.0	0.0	0.0	0.0	0.0
			70	102	180	0.1	0.0	0.0	0.0	0.0	0.0
			90	102	180	0.1	0.0	0.0	0.0	0.0	0.0
			110	102	180	0.1	0.0	0.0	0.0	0.0	0.0
		3	50	102	261	0.1	0.0	0.0	0.0	0.0	0.0
			70	102	261	0.1	0.0	0.0	0.0	0.0	0.0
			90	102	261	0.1	0.0	0.0	0.0	0.0	0.0
			110	102	261	0.2	0.0	0.0	0.0	0.0	0.0
		4	50	102	342	0.1	0.0	0.0	0.0	0.0	0.0
			70	102	342	0.1	0.0	0.0	0.0	0.0	0.0
			90	102	342	0.2	0.0	0.0	0.0	0.0	0.0
			110	102	342	0.2	0.0	0.0	0.0	0.0	0.0
20	20	2	50	402	760	0.9	0.0	0.0	0.0	0.0	0.0
			70	402	760	0.9	0.0	0.0	0.0	0.0	0.0
			90	402	760	1.0	0.0	0.0	0.0	0.0	0.0
			110	402	760	1.1	0.0	0.0	0.0	0.0	0.0
		3	50	402	1121	0.9	0.0	0.0	0.0	0.0	0.0
			70	402	1121	1.0	0.0	0.0	0.0	0.0	0.0
			90	402	1121	1.2	0.0	0.0	0.0	0.0	0.0
			110	402	1121	1.4	0.0	0.0	0.0	0.0	0.0
		4	50	402	1482	1.0	0.0	0.0	0.0	0.0	0.0
			70	402	1482	1.2	0.0	0.0	0.0	0.0	0.0
			90	402	1482	1.5	0.0	0.0	0.0	0.0	0.0
			110	402	1482	1.9	0.1	0.0	0.0	0.0	0.0
30	30	2	50	902	1740	3.4	0.0	0.0	0.0	0.0	0.0
			70	902	1740	3.6	0.0	0.0	0.0	0.0	0.0
			90	902	1740	3.8	0.0	0.0	0.0	0.0	0.0
			110	902	1740	4.1	0.0	0.0	0.0	0.0	0.0
		3	50	902	2581	3.5	0.0	0.0	0.0	0.0	0.0
			70	902	2581	3.9	0.0	0.0	0.0	0.0	0.0
			90	902	2581	4.3	0.0	0.0	0.0	0.0	0.0
			110	902	2581	4.8	0.1	0.0	0.0	0.0	0.0
		4	50	902	3422	3.8	0.0	0.0	0.0	0.0	0.0
			70	902	3422	4.4	0.0	0.0	0.0	0.0	0.0
			90	902	3422	5.2	0.0	0.0	0.0	0.0	0.0
			110	902	3422	6.3	0.1	0.0	0.0	0.0	0.0
40	40	2	50	1602	3120	9.7	0.0	0.0	0.0	0.0	0.0
			70	1602	3120	10.1	0.0	0.0	0.0	0.0	0.0
			90	1602	3120	10.6	0.0	0.0	0.0	0.0	0.0
			110	1602	3120	11.2	0.1	0.0	0.0	0.0	0.0
		3	50	1602	4641	10.0	0.0	0.0	0.0	0.0	0.0
			70	1602	4641	10.7	0.1	0.0	0.0	0.0	0.0
			90	1602	4641	11.6	0.1	0.1	0.0	0.1	0.0
			110	1602	4641	12.8	0.1	0.1	0.0	0.1	0.0
		4	50	1602	6162	10.5	0.0	0.0	0.0	0.0	0.0
			70	1602	6162	11.8	0.1	0.1	0.0	0.1	0.0
			90	1602	6162	13.4	0.1	0.1	0.0	0.1	0.0
			110	1602	6162	15.8	0.1	0.1	0.0	0.1	0.0
50	50	2	50	2502	4900	22.6	0.1	0.1	0.0	0.1	0.0
			70	2502	4900	23.2	0.1	0.1	0.0	0.1	0.0
			90	2502	4900	23.8	0.1	0.1	0.0	0.1	0.0
			110	2502	4900	24.7	0.1	0.1	0.0	0.1	0.0
		3	50	2502	7301	23.1	0.1	0.1	0.0	0.1	0.0
			70	2502	7301	24.1	0.1	0.1	0.0	0.1	0.0
			90	2502	7301	25.4	0.1	0.1	0.0	0.1	0.0
			110	2502	7301	27.3	0.1	0.1	0.0	0.1	0.0
		4	50	2502	9702	23.9	0.2	0.1	0.0	0.1	0.0
			70	2502	9702	25.7	0.1	0.1	0.0	0.1	0.0
			90	2502	9702	28.4	0.2	0.1	0.0	0.1	0.0
			110	2502	9702	32.3	0.3	0.2	0.0	0.1	0.0

Table 4-3: Computational time averages and standard deviations for 50 runs per row

a	b	adj	A	|N|	|E|	Label-Correcting Time (sec) avg	Label-Correcting Time (sec) stdev	Merge L.-C. Time (sec) avg	Merge L.-C. Time (sec) stdev
60	60	2	50	3602	7080	0.1	0.0	0.1	0.0
			70	3602	7080	0.1	0.0	0.1	0.0
			90	3602	7080	0.1	0.0	0.1	0.0
			110	3602	7080	0.1	0.0	0.1	0.0
		3	50	3602	10561	0.1	0.0	0.1	0.0
			70	3602	10561	0.1	0.0	0.1	0.0
			90	3602	10561	0.2	0.0	0.1	0.0
			110	3602	10561	0.2	0.0	0.2	0.0
		4	50	3602	14042	0.1	0.0	0.1	0.0
			70	3602	14042	0.2	0.0	0.1	0.0
			90	3602	14042	0.2	0.0	0.2	0.0
			110	3602	14042	0.2	0.0	0.2	0.0
70	70	2	50	4902	9660	0.2	0.0	0.1	0.0
			70	4902	9660	0.2	0.0	0.2	0.0
			90	4902	9660	0.2	0.0	0.2	0.0
			110	4902	9660	0.3	0.0	0.2	0.0
		3	50	4902	14421	0.2	0.0	0.1	0.0
			70	4902	14421	0.2	0.0	0.2	0.0
			90	4902	14421	0.3	0.0	0.2	0.0
			110	4902	14421	0.3	0.0	0.2	0.0
		4	50	4902	19182	0.2	0.0	0.2	0.0
			70	4902	19182	0.3	0.0	0.2	0.0
			90	4902	19182	0.3	0.0	0.2	0.0
			110	4902	19182	0.3	0.0	0.2	0.0
80	80	2	50	6402	12640	0.2	0.0	0.2	0.0
			70	6402	12640	0.2	0.0	0.2	0.0
			90	6402	12640	0.3	0.0	0.3	0.0
			110	6402	12640	0.3	0.0	0.3	0.0
		3	50	6402	18881	0.2	0.0	0.2	0.0
			70	6402	18881	0.3	0.0	0.2	0.0
			90	6402	18881	0.3	0.0	0.3	0.0
			110	6402	18881	0.3	0.0	0.3	0.0
		4	50	6402	25122	0.2	0.0	0.2	0.0
			70	6402	25122	0.3	0.0	0.3	0.0
			90	6402	25122	0.3	0.0	0.3	0.0
			110	6402	25122	0.4	0.0	0.3	0.0
90	90	2	50	8102	16020	0.3	0.0	0.3	0.0
			70	8102	16020	0.4	0.0	0.3	0.0
			90	8102	16020	0.4	0.0	0.3	0.0
			110	8102	16020	0.5	0.0	0.4	0.0
		3	50	8102	23941	0.3	0.0	0.3	0.0
			70	8102	23941	0.4	0.0	0.3	0.0
			90	8102	23941	0.5	0.0	0.4	0.0
			110	8102	23941	0.5	0.0	0.4	0.0
		4	50	8102	31862	0.3	0.0	0.3	0.0
			70	8102	31862	0.4	0.0	0.3	0.0
			90	8102	31862	0.5	0.0	0.4	0.0
			110	8102	31862	0.5	0.0	0.4	0.0
100	100	2	50	10002	19800	0.5	0.0	0.3	0.0
			70	10002	19800	0.6	0.0	0.4	0.0
			90	10002	19800	0.6	0.0	0.5	0.0
			110	10002	19800	0.7	0.0	0.5	0.0
		3	50	10002	29601	0.5	0.0	0.3	0.0
			70	10002	29601	0.6	0.0	0.4	0.0
			90	10002	29601	0.6	0.0	0.5	0.0
			110	10002	29601	0.7	0.0	0.5	0.0
		4	50	10002	39402	0.5	0.0	0.3	0.0
			70	10002	39402	0.6	0.0	0.4	0.0
			90	10002	39402	0.6	0.0	0.5	0.0
			110	10002	39402	0.7	0.0	0.5	0.0

Table 4-4: Computational time averages and standard deviations for 50 runs per row

Figure 4-5, Figure 4-6, and Figure 4-7 show the computational time required for each algorithm, according to the number of nodes in each problem. Although the CSPP-based algorithm was not used to solve $|N| > 2502$, all three figures present the same range for $|N|$ to facilitate comparison. The CSPP-based algorithm is the slowest of the three algorithms. To facilitate comparison between the label-correcting and the merge label-correcting algorithms, the y-axis in Figure 4-6 and Figure 4-7 are similar. The merge label-correcting algorithm is faster than the label-correcting algorithm. Moreover, it seems that the computational times for the merge label-correcting algorithm increase linearly with the number of nodes of the network.

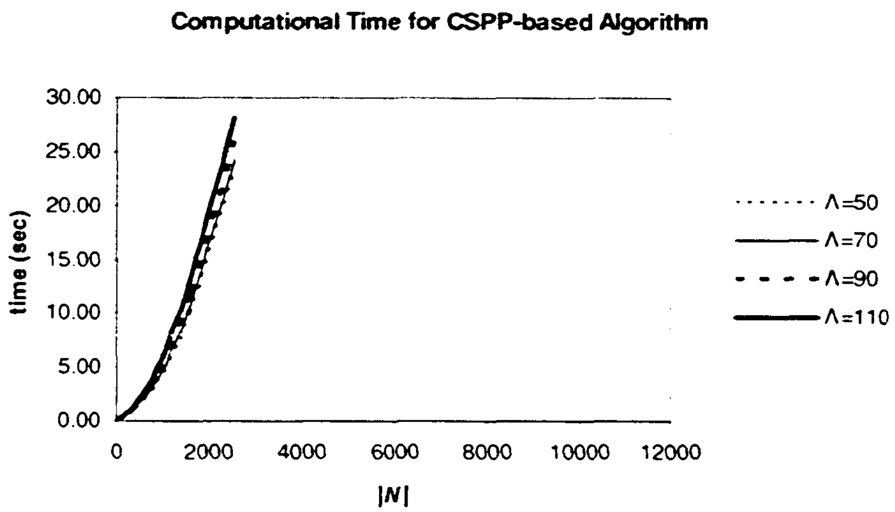

Figure 4-5: Computational time for CSPP-based algorithm

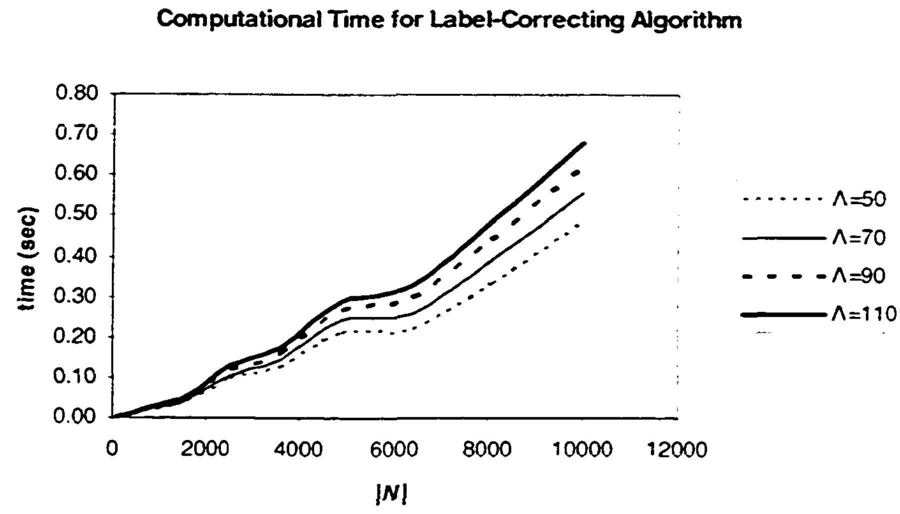

Figure 4-6: Computational time for label-correcting algorithm

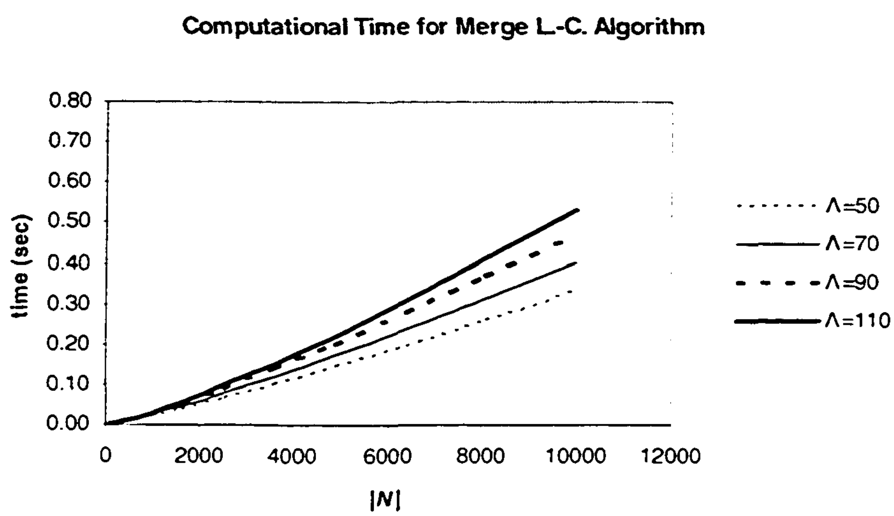

Figure 4-7: Computational time for merge l.-c. algorithm

Figure 4-8, Figure 4-9, and Figure 4-10, show computational time as a function of Λ. We can observe that the computational time for all algorithms increases with Λ.

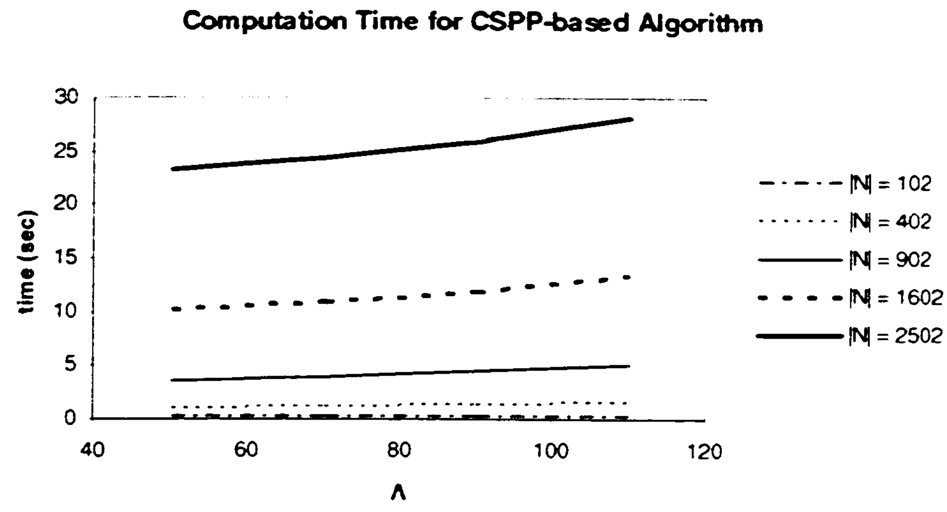

Figure 4-8: Computational time for CSPP-based algorithm (Λ in x-axis)

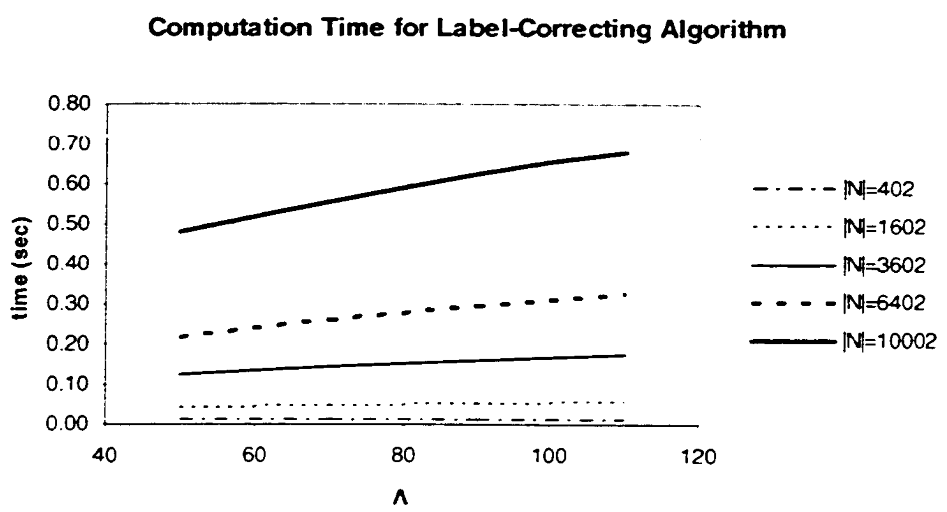

Figure 4-9: Computational time for label-correcting algorithm (Λ in x-axis)

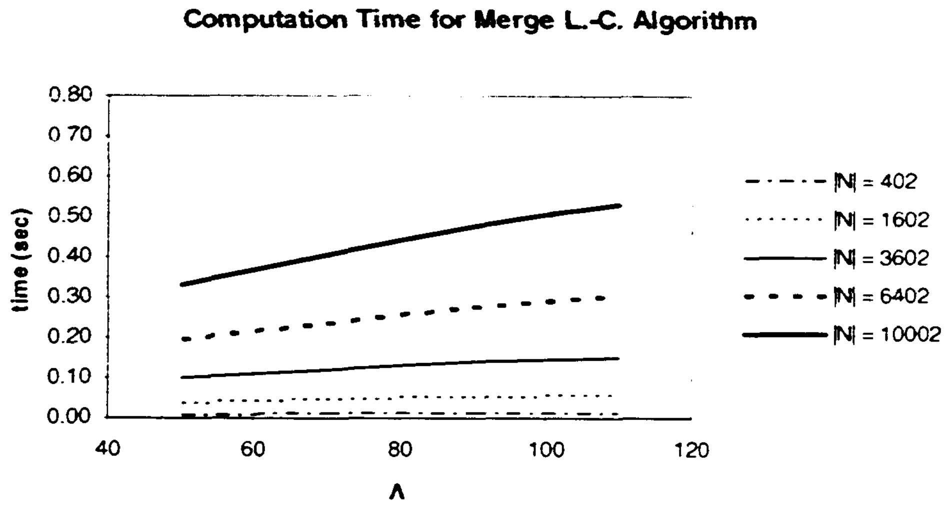

Figure 4-10: Computational time for merge L.-c. algorithm (Λ in x-axis)

To characterize the computational time for each algorithm, we applied a regression model, where the estimated time is equal to $\alpha(|N|^{\beta})(\Lambda^{\gamma})$. We also minimized the sum of squared errors, while fitting values of α, β, and γ. For the CSPP-based algorithm, the best fit was given by $10^{-10}(|N|^{3.00})(\Lambda^{0.64})$, with a $R^2 = 0.86$. For the label-correcting algorithm, the best fit was given by $4.52\times10^{-7}(|N|^{1.27})(\Lambda^{0.53})$, with a $R^2 = 0.98$. Finally, for the merge label-correcting algorithm, the best fit was given by $7.05\times10^{-7}(|N|^{1.23})(\Lambda^{0.47})$, with a $R^2 = 0.98$. These results are similar to our estimated algorithm complexities: namely $O(|N||E|\Lambda)$ for the first algorithm, $O(|N|^2\Lambda^2)$ for the second algorithm, and $O(|N||\Lambda|\log|\Lambda|)$ for the third algorithm.

A sample problem solved by the above algorithms is provided in Figure 4-11, and is constructed with $a = 5$, $b = 5$, $adj = 2$ and $\Lambda = 114$. For this example, $|N| = 27$ and $|E| = 50$. The solution obtained by both the CSPP-based and the label-correcting algorithm is $p = \{0, 2, 7, 8, 9, 8, 13, 14, 15, 20, 19, 24, 26\}$, with an associated repeater pattern $r = \{9, 13, 20\}$, and a total cost of implementation of 1822 units. The first sub-path has a total length of 104, the second sub-path has a total length of 97, and the last sub-path has a total length of 98.

Below, we have an example of a T-junction, where the signal will enter and leave the node by the same edge.

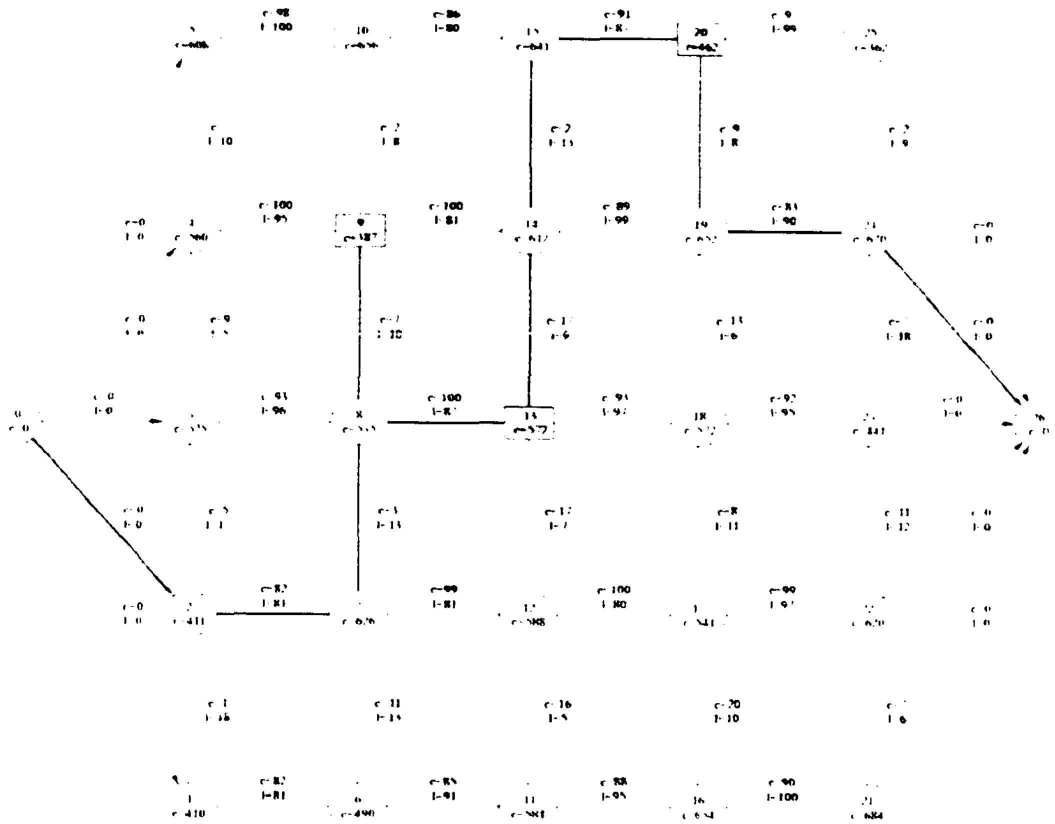

Figure 4-11: Example of CSPwRR problem

4.5 Conclusions and Comments

In this chapter, we presented and solved the *Constrained Shortest Path with Resource Regeneration Problem* (CSPwRRP). It consists of finding a feasible communication path between two points, while minimizing the cost of traversing edges in the graph and of installing appropriate repeaters to render the path distance feasible, with segments between origin, destination and repeaters shorter than Λ.

We presented three algorithms: one using the *Constrained Shortest Path Problem* (CSPP) as a sub-problem for the resolution of the CSPwRRP, and the other two using a label-correcting algorithm. The overall complexities were estimated as $O(|N||E|\Lambda)$ for the first algorithm (Section 4.3.1), $O(|N|^2\Lambda^2)$ for the second algorithm (Section 4.3.2), and $O(|N||\Lambda|\log|\Lambda|)$ for the third algorithm (Section 4.3.3). In our computational experiments, the third algorithm outperformed the other two algorithms, as one can observe from Table 4-3 and Table 4-4 in Section 4.4. Moreover, from the results presented in Figure 4-6 and Figure 4-7 and from the values obtained by the regression models, we can conclude that both the label-correcting algorithms (Sections 4.3.2 and 4.3.3) are promising for solving the CSPwRRP, and that the merge label-correcting algorithm is the best one.

4.6 References

Ahuja, R.K., T.L. Magnanti, J.B. Orlin. 1993. *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall, Englewood Cliffs, NJ.

Aneja, Y.P., V. Aggarwal, K.P.K. Nair. 1983. Shortest Chain Subject to Side Constraints. *Networks* **13** 295-302.

Bapna, R., L.S. Thakur, S.K. Nair. 2002. Infrastructure development for conversion to environmentally friendly fuel. *European Journal of Operational Research* **142** 480–496.

Beasley, J.E., N. Christofides. 1989. An Algorithm for the Resource Constrained Shortest Path Problem. *Networks* **19** 379-394.

Brumbaugh-Smith, J., D. Shier. 1989. An Empirical Investigation of Some Bicriterion Shortest Path Algorithms. *European Journal of Operational Research* **43** 216-224.

Carlyle, W.M., R.K. Wood. 2003a. Lagrangian Relaxation and Enumeration for Solving Constrained Shortest-Path Problems. *Proceedings of the 38th Annual ORSNZ Conference*.

Dumitrescu, I., N. Boland. 2003. Improved Preprocessing, Labeling and Scaling Algorithms for the Weight-Constrained Shortest Path Problem. *Networks* **42**(3) 135-153.

Garey, M.R., D.S. Johnson. 1979. *Computers and intractability: A guide to the theory of NP-completeness*. W.H. Freeman, San Francisco.

Handler, G.Y., I. Zang. 1980. A Dual Algorithm for the Constrained Shortest Path Problem. *Networks* **10** 293-309.

Joksch, H.C. 1966. The Shortest Route Problem with Constraints. *Journal of Mathematical Analysis and Applications* **14** 191-197.

Kuby, M., S. Lim. 2003. The Flow-Refueling Location Problem for Alternative-Fuel Vehicles. *Socio-Economic Planning Sciences*. Submitted.

5 The Network Design with Resource Regeneration Problem (NDwRRP)

5.1 Introduction

The problem discussed in this chapter is the design of a network composed of links and points of regeneration, in which commodities can flow from their origin to their destination and be regenerated with their maximum travel autonomy. We name this problem the *Network Design with Resource Regeneration Problem* (NDwRRP).

Applications of the NDwRRP are encountered in wide area telecommunication network design and in transportation network design, where refuelling stations are intrinsic to the desired network. For example, in telecommunication, fibre optic signals can travel up to 70 km without signal regeneration. They need reinforcement beyond this limit. Trenches for fibre and shelters for equipment must be installed to provide pathways for communication to flow. For each communication path, the distance between regeneration equipment (and related equipment shelters) must respect the 70 km limit. Our first contact with this problem was through the Alberta SuperNet project, which aimed to connect 422 communities in the province. Because these communities were far apart, the major decision was where to place repeaters in order to reinforce the signal.

Although the optimization literature is rich in network design problems, we found no model that included constraints on the distance between repeaters (see surveys by Magnanti and Wong (1984); Balakrishnan, Magnanti, and Mirchandani (1997); and Raghavan and Magnanti (1997)). Most of the research on network design focuses on the fixed cost of installing trenches for fibres, and disregards the fixed cost of nodes for equipment installation. Tcha and Yoon (1995) discuss fixed costs both along the edges and at the nodes of a network where hub candidates must be selected, and they provide for signal bundling and switching. To model the problem, the authors assume that each region must have one hub assigned to it, consequently forming a facility location model. They then propose a dual-based heuristic solution. Yoon, Baek, and Tcha (1998) present an extension to the work proposed in Tcha and Yoon (1995).

In a cooperative research study with Bell West Inc., we observed that although fibre installation is a major expense in the installation phase of telecommunication network design, equipment housing and further equipment renewal contributed to the overall cost of the network. Our literature review identified that present OR models focus on the costs at the edges. We thus decided to pursue a model that could incorporate costs at both the edges and nodes of the graph, in particular, for signal regeneration purposes. To the best of our knowledge, our research is the first to consider equipment-housing issues in network design. Although we use terminology specific to telecommunication network design to define the problem, the concepts we present can be extended to network design problems with similar characteristics.

This chapter is organized as follows. Section 5.2 presents a formal description of the NDwRRP. Section 5.3 contains two possible formulations for the NDwRRP, one using a formulation directly related to the problem definition (which is currently unsolvable), and another using an equivalent, larger formulation (which is solvable). We also describe in this section, a *Column Generation* (CG) framework for each formulation to obtain a lower bound for the NDwRRP. Section 5.4 details the implementation of two heuristics to solve the NDwRRP. Section 5.5 contains the computational experiments to evaluate the heuristics. Finally, a conclusion is provided in Section 5.6.

5.2 Problem Definition

Given an undirected graph $G=(N,E)$ composed of $n=|N|$ nodes and $m=|E|$ edges ($E=\{(i,j):i,j\in E \text{ and } i<j\}$, a set $K=\{(o(k),d(k))\}$ of communication node pairs defining communication flow in both directions between nodes $o(k)$ and $d(k)$ in N for each $k=1,\ldots,|K|$, and a maximum distance between points of regeneration Λ, the NDwRRP searches for a sub-graph $G'=(N'\subseteq N,E'\subseteq E)$ of fibre optic trenches and a subset $R\subseteq N'$ of equipment shelters. These must minimize *network implementation costs* and provide a *feasible path* for each communicating pair defined in K.

Network implementation costs are calculated as follows. Each node $i\in N$ has a fixed cost h_i, of creating an equipment shelter. This shelter will host the signal repeater, henceforth called *repeater*, at node i. Each edge $e=(i,j)$, $e\in E$ has a fixed cost $c_e \equiv c_{ij} \geq 0$, the cost of installing a trench that will accommodate optical fibres that transport signals in both directions along e (from i to j and from j to i). Consequently, given a solution G' and R, its implementation cost is given by $\sum_{e\in E'} c_e + \sum_{i\in R} h_i$. We assume that the number of fibres in each trench is sufficient to accommodate all communication flow in the network and that the cost of optical fibres is incorporated in c_e. We also assume that the signal regeneration equipment can regenerate all communication flow in the network and that equipment cost is included in h_i.

Path feasibility is defined as follows. Each edge $e\in E$ also has a length $d_e \geq 0$, which is the distance a commodity travels along edge e in either direction. Given G', R, and origin-destination pairs defined in K, a path for communication flow $k\in K$ in G' is a sequence of nodes $p=\{i_0 = o(k), i_1, \ldots, i_{|p|+1} = d(k)\}$ with $|p|+1$ nodes, such that $(i_{l-1}, i_l)\in E'$ for all $l=1,\ldots,|p|+1$. Associated regeneration points are defined as $r\subseteq R\cap p\setminus\{i_0, i_{|p|+1}\}$. In that case, we say that path p is composed of $|r|+1$ sub-paths p_t, $t=0,\ldots,|r|$. The subpath p_0 extends from the origin node $i_0 = o(k)$ to the first repeater node in path p, p_1 is the sub-path from the first repeater node in p to the second repeater node in p, and so on. The last sub-path, $p_{|r|}$, is the segment from the last repeater node in p to the destination node $i_{|p|+1} = d(k)$. A path $p=\{i_0 = o(k), \ldots, i_{|p|+1} = d(k)\}$ for a commodity flow $k\in K$ is feasible for given G' and R, if there is an associated repeater pattern $r\subseteq R$ and each sub-path p_t for $t=0,\ldots,|r|$ has a total length $\sum_{(i,j)\in p_t} d_{ij} \leq \Lambda$. Notice that a feasible path defined for communication from node $o(k)$ to node $d(k)$ is also feasible for the communication flow from $d(k)$ to $o(k)$.

To illustrate the NDwRRP, we refer to example G in Figure 5-1. Two communication pairs are established: one from node 1 to node 5, and another from node 1 to node 7. Therefore, $K=\{(1,5),(1,7)\}$. Suppose the maximum distance travelled by a signal without regeneration is $\Lambda = 84$.

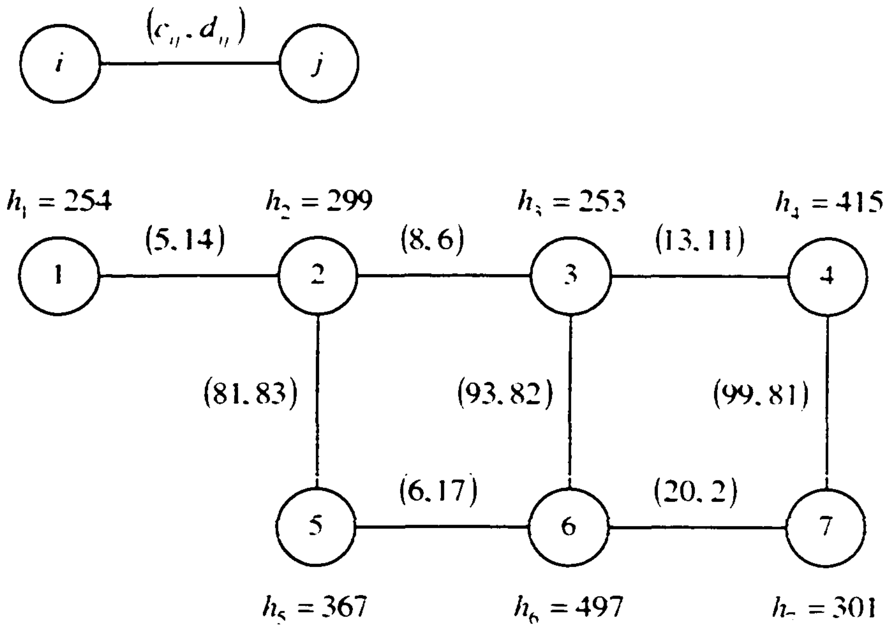

Figure 5-1: Graph *G*

One can easily arrive at a reasonable solution for the example in Figure 5-1. The path $p = \{1, 2, 5\}$ with repeater pattern $r = \{2\}$ provides a feasible communication path for communication pair $(1, 5) \in K$, at a total cost of 5+81+299 = 385. For the communication pair $(1, 7) \in K$, a possible path/repeater pattern is $p = \{1, 2, 3, 6, 7\}$ with a repeater pattern $r = \{3\}$, at a total cost of 5+8+93+20+253=379. The network design using the above mentioned path/repeater patterns would cost 385+379-5=759.

The problem solved in the previous chapter, the *Constrained Shortest Path with Resource Regeneration Problem* (CSPwRRP), is related to finding a feasible path in G. Actually, if $|K| = 1$, the NDwRRP is reduced to the CSPwRRP. When $|K| > 1$, communication paths can share links and shelters. Consequently, two or more paths that are sub-optimal in the CSPwRRP may lead to an optimal solution for the NDwRRP.

To prove that the NDwRRP is NP-hard, it suffices to evaluate a case where all commodity flow in K originates at a single node in N and where repeaters are unnecessary (which is equivalent to $\Lambda \cong \infty$). Thus, the NDwRRP reduces to the Steiner Tree Problem, which is NP-hard (see Garey and Johnson 1979 and Ortega and Wolsey 2003). Hence, a simplified version of the problem with a single origin and no repeaters is difficult to solve.

5.3 Formulation

This section is divided into four sub-sections. Sub-section 5.3.1 describes an *Integer Programming* (IP) model (named IP1) for the NDwRRP, which uses the undirected graph $G=(N,E)$ and the set K, as presented in the problem definition in Section 5.2. Sub-section 5.3.2 introduces the *Column Generation* (CG) framework, in which the IP1 is relaxed. A scheme to obtain a lower bound over the NDwRRP is presented. However, as the sub-problem to be solved in this CG framework is identified as unsolvable, a graph transformation from G to G' and another IP formulation are presented in sub-section 5.3.3. A second CG framework scheme, featuring the CSPwRRP as the CG sub-problem is then presented in sub-section 5.3.4.

5.3.1 IP Formulation for the Undirected Graph (IP1)

For each commodity flow $k \in K$, let P^k be the set of all paths in G with an origin in $o(k)$ and a destination in $d(k)$ that are feasible, given a proper set of repeater locations. Therefore, for each path $p \in P^k$, there is an associated set R^p of different repeater patterns such that p and r form a feasible path/repeater pattern over G. The following are the variables used in IP1:

$q^k_{p,r}$ = 1, if commodity $k \in K$ uses path $p \in P^k$ and repeater pattern $r \in R^p$, 0 otherwise.

x_e = 1, if edge $e \in E$ is included in E', 0 otherwise

y_i = 1, if node $i \in N$ is included in R, 0 otherwise

a^p_e and b^r_i are binary coefficients, where:

$$a^p_e = \begin{cases} 1 & \text{if edge } e \text{ is member of path } p \\ 0 & \text{otherwise} \end{cases}$$

and

$$b^r_i = \begin{cases} 1 & \text{if node } i \text{ is member of repeater pattern } r \\ 0 & \text{otherwise} \end{cases}$$

The mathematical formulation IP1 for the NDwRRP is:

(IP1) Min

$$\sum_{e\in E} c_e x_e + \sum_{i\in N} h_i y_i \qquad (5.1)$$

S.t.

$$\sum_{\substack{p\in P^k,\\ r\in R^p}} q^k_{p,r} = 1 \qquad \forall k\in K \qquad (5.2)$$

$$\sum_{\substack{p\in P^k,\\ r\in R^p}} a^p_e q^k_{p,r} - x_e \le 0 \qquad \forall e\in E, k\in K \qquad (5.3)$$

$$\sum_{\substack{p\in P^k,\\ r\in R^p}} b^r_i q^k_{p,r} - y_i \le 0 \qquad \forall i\in N, k\in K \qquad (5.4)$$

$q^k_{p,r}$, x_e and y_i are binary variables

This model contains an exponential number of variables and $|K|(1+m+n)$ constraints. The objective function (5.1) is composed of trench and equipment shelter costs. Constraints (5.2) impose exactly one path and one repeater pattern for each O-D pair. Constraints (5.3) and (5.4) guarantee that edges on selected paths have trenches and that nodes on selected repeater patterns have equipment shelters in the final design.

As the coefficients in (5.3)-(5.4) are limited to 0 or 1, the above model is highly degenerate. Cutting plane constraints can be added to reduce degeneracy by expanding constraints (5.3) and (5.4) so that each path and repeater is individually accounted for in terms of trench and repeater facility installation in the final design. However, this approach may create an exponential number of constraints.

In the next sub-section, a relaxed version of the IP1 formulation is considered. To avoid the problem of considering all feasible paths, we apply a column generation framework to form a sub-set of variables that provides the optimal solution to the relaxed version of IP1.

5.3.2 Column generation framework of IP1

In a column generation framework, a large *Linear Programming* (LP) problem is solved. This problem is defined as the *master problem* (LMP) and usually contains an exponential number of variables. Its dual is referred to as the DMP. A restricted master problem (RMP) considers only a subset of the feasible variables of the LMP, with DRMP being its dual. In the column generation framework, one starts with a set of variables in the RMP that renders the problem feasible and performs a series of iterations, in which new variables are included in the RMP, until no more variables can be added to improve the solution. To identify the variables to be inserted, the RMP is solved, and the dual values are calculated for the DRMP. Then, a sub-problem is sought by finding a constraint in the DRMP that can be violated, considering the present dual values. When such constraint violation is identified in the DRMP, the equivalent variable can be inserted in the RMP, improving the solution.

The master problem for IP1 (named LMP1) is presented below, where the integrality constraints of IP1 are relaxed:

(LMP1)Min

$$\sum_{e\in E} c_e x_e + \sum_{i\in N} h_i y_i \tag{5.5}$$

S.t.

$$\sum_{\substack{p\in P^k,\\ r\in R^p}} q^k_{p,r} = 1 \qquad \forall k\in K \tag{5.6}$$

$$\sum_{\substack{p\in P^k,\\ r\in R^p}} a^p_e q^k_{p,r} - x_e \le 0 \qquad \forall e\in E, k\in K \tag{5.7}$$

$$\sum_{\substack{p\in P^k,\\ r\in R^p}} b^r_i q^k_{p,r} - y_i \le 0 \qquad \forall i\in N, k\in K \tag{5.8}$$

$$q^k_{p,r}, x_e, y_i \ge 0$$

The corresponding dual of LMP1 is named DMP1. The restricted master problem (RMP1) considers only a subset of the feasible paths $\overline{P}^k \subseteq P^k$ and a subset of the feasible repeaters $\overline{R}^p \subseteq R^p$, defined for $p\in \overline{P}^k$. Consequently, RMP1 is:

(RMP1) Min

$$\sum_{e \in E} c_e x_e + \sum_{i \in N} h_i y_i \qquad (5.9)$$

S.t.

$$\sum_{\substack{p \in \bar{P}^k,\\ r \in \bar{R}^p}} q_{p,r}^k = 1 \qquad \forall k \in K \quad (5.10)$$

$$\sum_{\substack{p \in \bar{P}^k,\\ r \in \bar{R}^p}} a_e^p q_{p,r}^k - x_e \le 0 \qquad \forall e \in E, k \in K \quad (5.11)$$

$$\sum_{\substack{p \in \bar{P}^k,\\ r \in \bar{R}^p}} b_i^r q_{p,r}^k - y_i \le 0 \qquad \forall i \in N, k \in K \quad (5.12)$$

$$q_{p,r}^k, x_e, y_i \ge 0$$

for which the dual is defined as:

(RDMP1) Max

$$\sum_{k \in K} u_k \qquad (5.13)$$

S.t.

$$u_k + \sum_{e \in E} a_e^p v_e^k + \sum_{i \in N} b_i^r w_i^k \le 0 \qquad \forall k \in K, p \in \bar{P}^k, r \in \bar{R}^p \quad (5.14)$$

$$\sum_{k \in K} - v_e^k \le c_e \qquad \forall e \in E \quad (5.15)$$

$$\sum_{k \in K} - w_i^k \le h_i \qquad \forall i \in N \quad (5.16)$$

u^k unrestricted $\qquad \forall k \in K$

$v_e^k, w_i^k \le 0 \qquad \forall e \in E, i \in N, k \in K$

Let $(\bar{q}, \bar{x}, \bar{y})$ be an optimal solution to RMP1 and $(\bar{u}, \bar{v}, \bar{w})$ be the associated dual optimal solution to RMP1. As RMP is a restriction of LMP1, we can obtain a feasible solution to LMP1 from the solution of RMP1 by setting $q = \bar{q}$ for $p \in \bar{P}^k$ and $r \in \bar{R}^p$, and $q = 0$ for

$p\in \overline{P}^k$ and $R^p \setminus \overline{R}^p$. Hence, $z(RMP)\geq Z(LMP)$. Moreover, the solution to RMP1 is optimal to LMP1, only if the dual solution $(\overline{u},\overline{v},\overline{w})$ is also feasible for DMP1 (the dual of LMP1), i.e., if no constraint

$$\overline{u}_k + \sum_{e\in E} a_e^{pr}\overline{v}_e^k + \sum_{i\in N} b_i^r \overline{w}_i^k \leq 0 \qquad \forall k\in K, p\in P^k, r\in R^p$$

is violated.

Note that the other two dual constraints (5.15) and (5.16) are feasible, as they do not depend on the path p. So, if for every communication pair $k\in K$, associated path/repeater pattern $p\in P^k$, and $r\in R^p$ we have $u_k + \sum_{e\in E} a_e^{pr} v_e^k + \sum_{i\in N} b_i^r w_i^k \leq 0$, then $(\overline{q},\overline{x},\overline{y})$ is optimal for LMP1. To verify this condition, it suffices to solve the following sub-problem for each $k\in K$:

$$\min_{\substack{p\in P^k\\ r\in R^p}} \left\{ -\sum_{e\in E} a_e^{pr}\overline{v}_e^k - \sum_{i\in N} b_i^r \overline{w}_i^k \right\} \tag{5.17}$$

which follows from

$$\overline{u}_k + \sum_{e\in E} a_e^{pr}\overline{v}_e^k + \sum_{i\in N} b_i^r \overline{w}_i^k \leq 0$$

$$\max_{\substack{p\in P^k\\ r\in R^p}} \left\{ \overline{u}_k + \sum_{e\in E} a_e^{pr}\overline{v}_e^k + \sum_{i\in N} b_i^r \overline{w}_i^k \right\} \leq 0$$

$$\overline{u}_k + \min_{\substack{p\in P^k\\ r\in R^p}} \left\{ -\sum_{e\in E} a_e^{pr}\overline{v}_e^k - \sum_{i\in N} b_i^r \overline{w}_i^k \right\} \leq 0$$

Problem (5.17) can be described as the selection of edges in E and nodes in N forming a sub-graph, in which a feasible path/repeater pattern for communication flow $k\in K$ can be found. Therefore, it only considers the edges necessary for such a path. Unfortunately, the problem is difficult to solve, although it appears similar to the *Constrained Shortest Path with Resource Regeneration Problem* (CSPwRRP) described in the previous chapter. To illustrate the difference between problem (5.17) and the CSPwRRP, we refer to the example in Figure

5-2. The communication flow goes from node 1 to node 3, and the maximum distance between repeaters is $\Lambda = 70$. By formulating the problem as a CSPwRRP for the graph in Figure 5-2, the optimal solution would be path $p = \{1, 2, 3, 4\}$ with repeater pattern $r = \{2\}$, at a total cost of 55+10+50+100 = 215. The solution path $p = \{1, 3, 2, 3, 4\}$ with the repeater pattern $r = \{2\}$ would be sub-optimal, as the total cost would be 55+10+10+50+100 = 225 (edge (2, 3) would be counted twice). However, by formulating the problem as problem (5.17), the optimal solution would be solution path $p = \{1, 3, 2, 3, 4\}$ and repeater pattern $r = \{2\}$, because edge (2, 3) would be counted only once, for a total cost of 50+10+50+100 = 210.

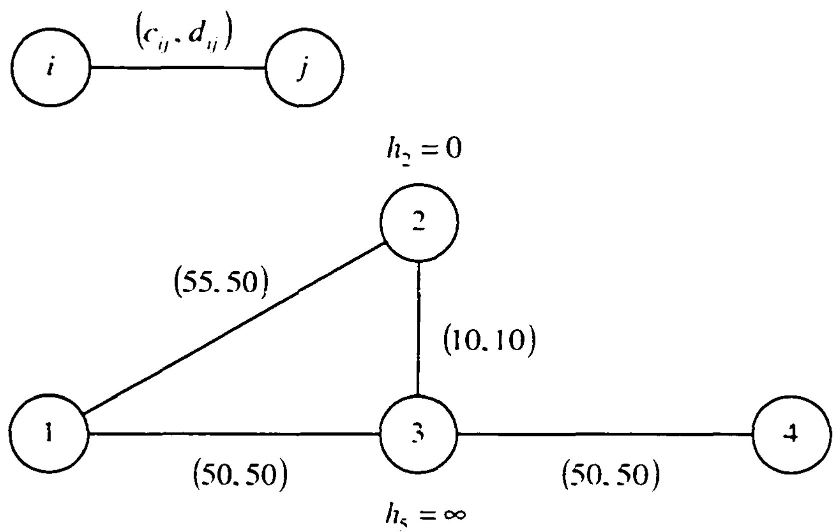

Figure 5-2: Example to distinguish problem (5.17) from CSPwRRP

Instead of trying to solve this problem as presented, one can perform a graph transformation and an equivalent formulation, for which the sub-problem is the CSPwRRP. Thus, the problem is solvable, as we will show in the next two sub-sections.

5.3.3 IP Formulation for the Directed Graph (IP2)

In formulation IP2, we use the directed graph $H = (N, A)$, instead of the undirected graph $G = (N, E)$ from the original problem formulation. Both G and H have the same set of nodes N; however, for each edge $(i, j) \in E$, two arcs in opposite directions are created in A.

Consequently, $A = \{(i,j),(j,i): \text{for each edge } (i,j) \in E\}$ with $2m$ arcs. Arcs (i,j) and (j,i) for a given edge $(i,j) \in E$ have associated trenching costs $\bar{c}_{ij}$ and $\bar{c}_{ji}$ equal to half the trenching cost of their respective edge (i.e., $\bar{c}_{ij} = \bar{c}_{ji} = c_{ij}/2$). With such transformation, we need to create a feasible communication path for each $k \in K$ in both directions, from $o(k)$ to $d(k)$ and from $d(k)$ to $o(k)$, to obtain a trenching cost equal to the one obtained via formulation IP1. Therefore, we redefine an alternative communication set K', where each communication pair $k \in K$ with an origin at $o(k)$ and a destination at $d(k)$ is represented by two communication pairs k' and k'' in K', with $o(k') = o(k)$, $d(k') = d(k)$, $o(k'') = d(k)$, and $d(k'') = o(k)$. Figure 5-3 presents the equivalent graph H for the example from Figure 5-1: $K' = \{(1,5),(1,7),(5,1),(7,1)\}$.

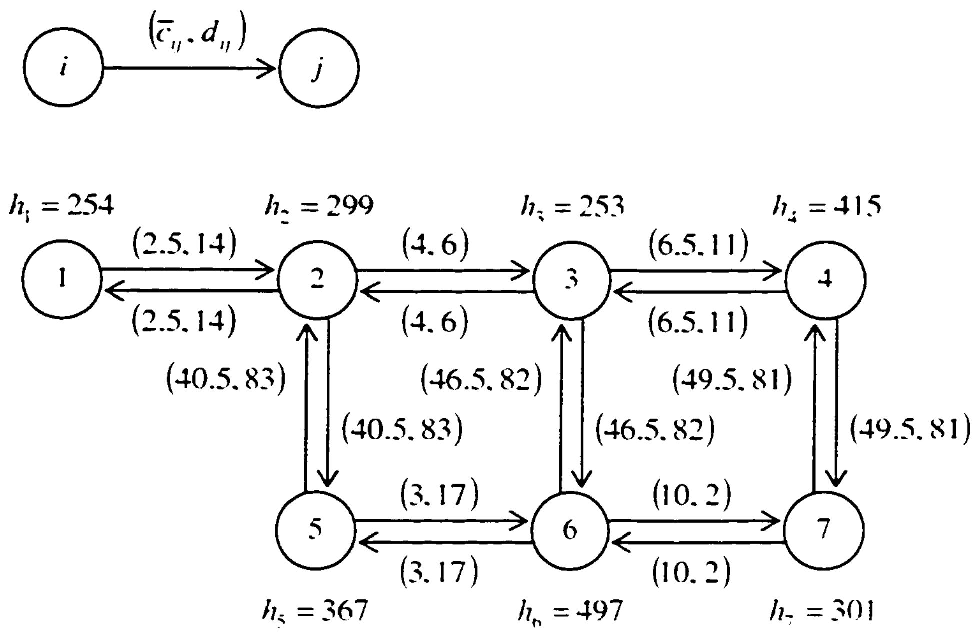

Figure 5-3: Graph H for the example from Figure 5-1

IP2 is:

(IP2) Min

$$\sum_{(i,j) \in A} \bar{c}_{ij} x_{ij} + \sum_{i \in N} h_i y_i \tag{5.18}$$

S.t.

$$\sum_{\substack{p \in P^k, \\ r \in R^p}} q_{p,r}^k = 1 \qquad \forall k \in K' \quad (5.19)$$

$$\sum_{\substack{p \in P^k, \\ r \in R^p}} a_{ij}^p q_{p,r}^k - x_{ij} \le 0 \qquad \forall (i,j) \in A, k \in K' \quad (5.20)$$

$$\sum_{\substack{p \in P^k, \\ r \in R^p}} b_i^r q_{p,r}^k - y_i \le 0 \qquad \forall i \in N, k \in K' \quad (5.21)$$

$q_{p,r}^k$, x_{ij} and y_i are binary variables

a_{ij}^p and b_i^r are binary coefficients, where

$$a_{ij}^p = \begin{cases} 1 & \text{if arc } (i,j) \in A \text{ is member of path } p \\ 0 & \text{otherwise} \end{cases}$$

and

$$b_i^r = \begin{cases} 1 & \text{if node } i \text{ is member of repeater pattern } r \\ 0 & \text{otherwise} \end{cases}$$

We now prove that the optimal solution to IP2 produces the same objective function value as the optimal solution to IP1.

Proposition 1: $z(\text{IP1}) = z(\text{IP2})$.

Proof: First, assume that $z(\text{IP1}) > z(\text{IP2})$. In that case, there must be a set of path/repeater patterns in IP2 not present in IP1. The total fixed cost on the arcs and nodes must be less than in IP1. However, if such a path/repeater pattern set exists and is feasible, an equivalent path/repeater pattern in IP1 could be selected instead. By contradiction, we show that $z(\text{IP1}) > z(\text{IP2})$ is not possible. The same reasoning can be used to show that $z(\text{IP1})$ cannot be smaller than $z(\text{IP2})$. ■

5.3.4 Column generation framework of IP2

The master problem for IP2, LMP2, is:

(LMP2) Min

$$\sum_{(i,j)\in A} \bar{c}_{ij} x_{ij} + \sum_{i\in N} h_i y_i \tag{5.18}$$

S.t.

$$\sum_{\substack{p\in P^k,\\ r\in R^p}} q^k_{p,r} = 1 \qquad \forall k \in K' \tag{5.19}$$

$$\sum_{\substack{p\in P^k,\\ r\in R^p}} a^p_{ij} q^k_{p,r} - x_{ij} \le 0 \qquad \forall (i,j)\in A, k\in K' \tag{5.20}$$

$$\sum_{\substack{p\in P^k,\\ r\in R^p}} b^r_i q^k_{p,r} - y_i \le 0 \qquad \forall i\in N, k\in K' \tag{5.21}$$

$$q^k_{p,r}, x_e, y_i \ge 0$$

a^p_{ij} and b^r_i are binary coefficients, where

$$a^p_{ij} = \begin{cases} 1 & \text{if arc } (i,j)\in A \text{ is member of path } p \\ 0 & \text{otherwise} \end{cases}$$

and

$$b^r_i = \begin{cases} 1 & \text{if node } i \text{ is member of repeater pattern } r \\ 0 & \text{otherwise} \end{cases}$$

The corresponding dual of LMP2 is named DMP2. The restricted master problem (RMP2) considers only a subset of the feasible paths $\bar{P}^k \subseteq P^k$ and a subset of the feasible repeaters $\bar{R}^p \subseteq R^p$, defined for $k \in K'$ and $p \in \bar{P}^k$. Consequently, RMP2 is:

(RMP2) Min

$$\sum_{(i,j)\in A} \bar{c}_{ij} x_{ij} + \sum_{i\in N} h_i y_i \tag{5.22}$$

S.t.

$$\sum_{\substack{p\in \bar{P}^k,\\ r\in \bar{R}^p}} q^k_{p,r} = 1 \qquad \forall k \in K' \tag{5.23}$$

$$\sum_{\substack{p\in\overline{P}^k,\\ r\in\overline{R}^p}} a_{ij}^r q_{p,r}^k - x_{ij} \le 0 \qquad \forall (i,j)\in A, k\in K' \quad (5.24)$$

$$\sum_{\substack{p\in\overline{P}^k,\\ r\in\overline{R}^p}} b_i^r q_{p,r}^k - y_i \le 0 \qquad \forall i\in N, k\in K' \quad (5.25)$$

$$q_{p,r}^k, x_{ij}, y_i \ge 0$$

for which the dual is defined as:

(RDMP2) Max

$$\sum_{k\in K'} u_k \quad (5.26)$$

S.t.

$$u_k + \sum_{(i,j)\in A} a_{ij}^r v_{ij}^k + \sum_{i\in N} b_i^r w_i^k \le 0 \qquad \forall k\in K', p\in\overline{P}^k, r\in\overline{R}^p \quad (5.27)$$

$$\sum_{k\in K'} -v_{ij}^k \le \overline{c}_{ij} \qquad \forall (i,j)\in A \quad (5.28)$$

$$\sum_{k\in K'} -w_i^k \le h_i \qquad \forall i\in N \quad (5.29)$$

u^k unrestricted $\qquad \forall k\in K'$

$v_{ij}^k, w_i^k \le 0 \qquad \forall (i,j)\in A, i\in N, k\in K'$

Let $(\overline{q},\overline{x},\overline{y})$ be an optimal solution to RMP2 and $(\overline{u},\overline{v},\overline{w})$ be the associated dual optimal solution to RMP2. As RMP2 is a restriction of LMP2, we can obtain a feasible solution to LMP2 from the solution of RMP2 by setting $q=\overline{q}$ for $p\in\overline{P}^k$ and $r\in\overline{R}^p$, and $q=0$ for $p\in\overline{P}^k$ and $R^p\setminus\overline{R}^p$. Hence, $z(RMP2)\ge z(LMP2)$. Moreover, the solution to RMP2 is optimal to LMP2, *if and only if* the dual solution $(\overline{u},\overline{v},\overline{w})$ is feasible for DMP2 (the dual of LMP2), i.e.

$$\overline{u}_k + \sum_{(i,j)\in A} a_{ij}^r \overline{v}_{ij}^k + \sum_{i\in N} b_i^r \overline{w}_i^k \le 0 \qquad \forall k\in K', p\in P^k, r\in R^p$$

Therefore, if for every communication pair $k \in K'$ and associated path/repeater pattern $p \in P^k$ and $r \in R^p$ we have $u_k + \sum_{(i,j) \in A} a_{ij}^p v_{ij}^k + \sum_{i \in N} b_i^r w_i^k \le 0$, then $(\bar{q}, \bar{x}, \bar{y})$ is optimal for LMP2. To verify this condition, it suffices to solve the following sub-problem for each $k \in K'$:

$$z(CSPwRRP_k) = \min_{\substack{p \in P^k, \\ r \in R^p}} \left\{ - \sum_{(i,j) \in A} a_{ij}^p \bar{v}_{ij}^k - \sum_{i \in N} b_i^r \bar{w}_i^k \right\} \tag{5.30}$$

The above sub-problem (5.30) is exactly the CSPwRRP described in the previous chapter. Each column identified by (5.30) is added to the RMP2; then dual values are calculated and new columns are identified. The procedure stops if no more columns can be identified.

When applied to the example given in Figure 5-1, the column generation procedure finds a lower bound of 686. Consequently, the first solution (with a cost of 759) is within 10.6% of the optimal solution.

However, identifying all necessary columns may be time consuming and unnecessary. Instead, we consider another stopping criterion using a lower bound on $z(RMP2)$. The summation of the CSPwRRP solutions for each $k \in K'$ forms the lower bound $\underline{z}(RMP2)$, stated as:

$$\underline{z}(RMP2) = \sum_{k \in K'} z(CSPwRRP_k)$$

and

$$gap = \frac{(z(RMP2) - \underline{z}(RMP2))}{\underline{z}(RMP2)}$$

With this approach, the algorithm stops when the gap between these two values is smaller than a user-defined parameter. We coded the algorithm for column generation and provided one example of upper and lower bounds to illustrate the column generation procedure behaviour, in Figure 5-4. The test in Figure 5-4 is a network design problem with 36 nodes, 48 edges, and 15 communication pairs. CG2% corresponds to the algorithm where the gap parameter is established at 2%. CG0% corresponds to the algorithm that stops only when new path/repeater pattern variables cannot be identified through the solution of the CSPwRRP for the DMP2. Each unit in the horizontal axis corresponds to an iteration of the column generation procedure, in which new variables are identified through the solution of the CSPwRRP over the dual values (violated constraints on RDMP2). The vertical axis corresponds to the total cost of implementing the network design.

In Figure 5-4, we present both the lower bounds (LB) and the upper bounds (UB) for these two algorithms. Each unit in the horizontal axis represents the solution of the RLMP formulation with CPLEX and the identification of violated constraints in the DLMP.

This small test was carried out in a Sun Fire 480R station with four 900 MHz processors and 16 gigabytes of RAM (more details are available at the beginning of Section 5.5). CG0% called CPLEX 449 times and stopped after 380 seconds, whereas CG2% called CPLEX 169 times and stopped after 133 seconds. The bounds of CG0% are identical to those of CG2% for the first 169 iterations.

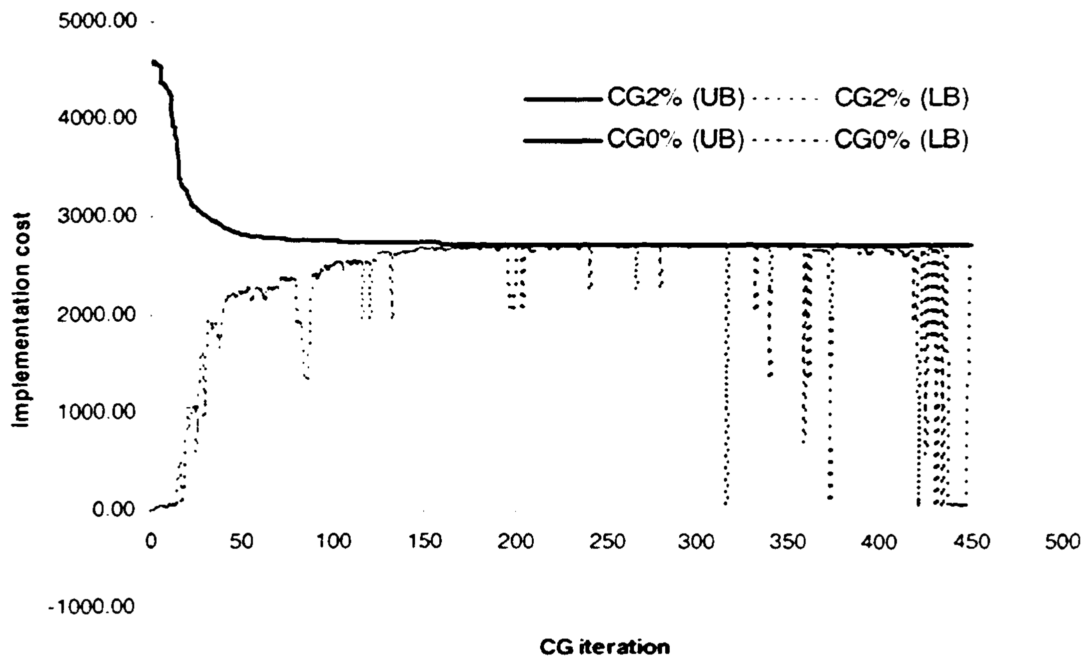

Figure 5-4: Example of convergence for the column generation framework

However, solving LPM2 to optimality does not mean that we have a solution to IP1. Nor do paths P^k and repeater patterns R^p, $p \in P^k$ for each $k \in K'$ found for LPM2 necessarily contain the paths and repeater patterns of the optimal solution to IP2. If we had the optimal paths and repeater patterns for LMP2, a branch-and-bound scheme would still be necessary to find the optimal solution to IP1. As we are unconvinced that this approach would be computationally feasible, especially for larger problems, we turn our attention to heuristic algorithms.

5.4 Heuristic Algorithms

We develop two heuristics to solve the NDwRRP. The first heuristic is similar to the algorithm developed by Takahashi and Matsuyama (1980) for the Steiner Tree Problem and consists of building the network one communication pair at a time. The second heuristic is based on a price perturbation scheme. An iteration of the algorithm involves calculating feasible paths and updating trench and shelter costs according to use. The most used receives a partial price that biases path selection towards the most used elements. Both algorithms use the algorithm developed in the previous chapter for the CSPwRRP.

5.4.1 First Heuristic

The Steiner Tree Problem is the search for a minimum spanning tree that connects a sub-set of mandatory nodes in a graph at a minimum cost. To solve this problem, Takahashi and Matsuyama (1980) propose a construction heuristic, in which one calculates the shortest paths that connect the present solution to mandatory nodes. The cheapest path is added to the solution. Every time a path is added, the edges along the path have their costs zeroed, so that the next shortest paths will take advantage of links already utilized in the solution.

The same idea is applied to the NDwRRP. This heuristic starts with empty G' and R and, at each iteration, adds trenches and shelters so that G' and R become feasible for one communication pair in K at a time. At the end of $|K|$ calls to the CSPwRRP algorithm, the final set G' and R form a feasible solution for the G and K given. The algorithm is detailed in Table 1, with the implementation cost of G' and R provided by variable z. The major drawback in this algorithm is that the result depends strongly on the sequence of trenches and shelters inserted into G' and R. A path may take advantage of trenches and shelters only if they have been inserted into the final solution. One should explore different sequences of communication pairs and choose the best solution.

Step 0: Let $G' = \varnothing$, $R = \varnothing$ and $z = 0$
Step 1:
for each $k \in K$ do
{ call CSPwRRP to find a path from $o(k)$ to $d(k)$; it returns path p and repeater set r
for each $(i,j) \in p$ do
{ $z = z + c_{ij}$; $c_{ij} = 0$; }
for each $i \in r$ do
{ $z = z + h_i$; $h_i = 0$; }
}

Table 5-1: First heuristic for he NDwRRP

For example, if we solved the example presented in Figure 5-1 first with $k = (1,5)$, the optimal CSPwRR would be $p = \{1,2,5\}$ and $r = \{2\}$, with a total cost of 5+81+299 = 385. Then, we solve the CSPwRRP for $k = (1,7)$, $p = \{1,2,3,6,7\}$, and $r = \{3\}$, with a total cost of 0+8+93+20+253 = 374. The total network design would be 759, which is the solution presented initially in Section 5.2.

However, if we first solved for $k = (1,7)$, the optimal CSPwRR would be $p = \{1,2,3,6,7\}$, with $r = \{3\}$ and a total cost of 5+8+93+20+253=379. Then, the optimal CSPwRR for $k = (1,5)$ would be $p = \{1,2,3,6,7,6,5\}$ and $r = \{3,7\}$, with a total cost of 0+0+0+0+6+301=307. The total network design cost would be 379+307 = 686, which (coincidentally) is the lower bound obtained by the column generation procedure. Consequently, this should be the optimal solution, represented below in Figure 5-5.

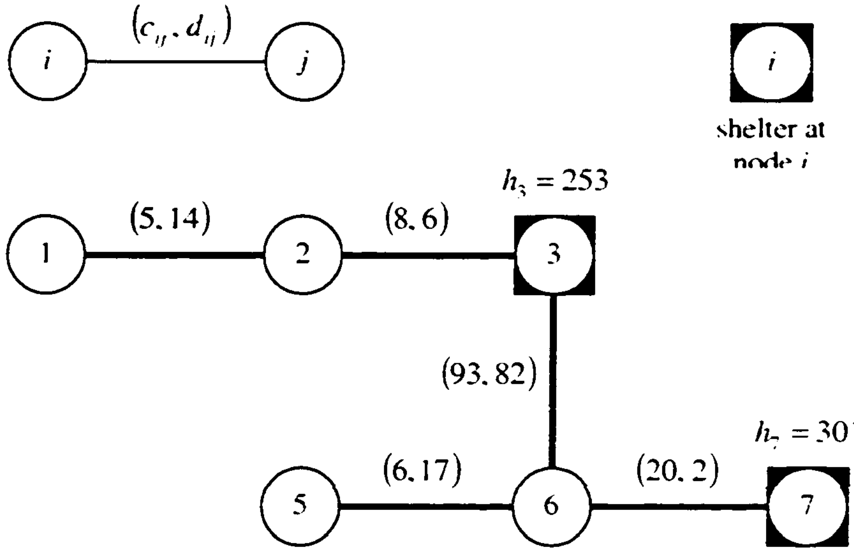

Figure 5-5: Optimal network design

5.4.2 Second Heuristic

The idea behind the second heuristic is to disturb the fixed costs of trenches and shelters, according to their common usage. When a set of feasible paths and repeater patterns are calculated for each $k \in K$ at each iteration t, for given installation costs c_{ij}^t and h_i^t, each edge $(i, j) \in E$ is linked with a counter Δ_{ij}^t. This indicates the number of feasible paths that use this edge, which can be twice per path (see cycle formation in Section 3.1). Each node $i \in N$ is linked with a counter δ_i, which indicates the number of repeater patterns that install a repeater at node i. Then, the installation prices for the next iteration are updated as $c_{ij}^{t+1} = \max((1 - \alpha\Delta_{ij}^t)c_{ij}^t, 0)$ and $h_i^{t+1} = \max((1 - \alpha\delta_i^t)h_i^t, 0)$, with $\alpha \in (0,1]$ being a coefficient (an algorithm parameter). Initial installation costs c_{ij}^0 and h_i^0 are set to c_{ij} and h_i, respectively. At each iteration, two implementation costs are calculated, one based on costs from the original problem, named z, and the other based on costs used for the iteration t, named z'. The algorithm stops at a given iteration $z' = 0$. The algorithm is detailed in Table 5-2.

Step 0: let $t = 0$, $c_{ij}^{0} = c_{ij}$ and $h_i^{0} = h_i$ for all $(i, j) \in E$ and $i \in N$

Step 1:

let $z = 0$, $z' = 0$, $\Delta_{ij}^{t} = 0$ and $\delta_i^{t} = 0$ for all $(i, j) \in E$ and $i \in N$

for each $k \in K$ do

{ call CSPwRRP to find a path from $o(k)$ to $d(k)$ using c_{ij}^{t} and h_i^{t} fixed costs;

it returns path p and repeater set r

for each $(i, j) \in p$ do

{ if $\Delta_{ij}^{t} = 0$ then $z = z + c_{ij}$ and $z' = z' + c_{ij}^{t}$

$\Delta_{ij}^{t} = \Delta_{ij}^{t} + 1$ }

for each $i \in r$ do

{ if $\delta_i^{t} = 0$ then $z = z + h_i$ and $z' = z' + h_i^{t}$

$\delta_i^{t} = \delta_i^{t} + 1$ }

}

if $z' > 0$ then Stop

else {

for all $(i, j) \in E$ do $c_{ij}^{t+1} = \max\left(\left(1 - \alpha\Delta_{ij}^{t}\right)c_{ij}^{t}, 0\right)$

for all $i \in N$ do $h_i^{t+1} = \max\left(\left(1 - \alpha\delta_i^{t}\right)h_i^{t}, 0\right)$

}

let $t = t + 1$

go to Step 1

Table 5-2: Second heuristic for the NDwRRP

5.5 Computational Results

All computational tests were carried out on a Sun Fire 480R station with four 900 MHz processors, 16 gigabytes of RAM and a Sun Solaris 5.7 operational system. The algorithms were coded in C++ and compiled with g++ 2.95.3. CPLEX 8.0 was used to solve the linear programs in the column generation procedure.

The test networks are similar to those used to test the CSPwRRP in Chapter 4. They have a grid structure, with a rows and b columns and uniform, randomly generated integer values for length and cost. Origin and destination nodes o and d are defined outside the grid: o is connected to all nodes in the leftmost column, and d is connected to all nodes in the rightmost column of the grid through arcs with zero length and zero cost. Each node u inside the grid is connected by edges to its adjacent nodes whenever possible (left, up, right and down). Length and cost values are selected from [10, 30] for vertical and horizontal edges. Factor *Rsize* determines the number of repeaters desired in the final solution. Factor *Ksize* determines the number of communication signals in K, where each origin $o(k)$ and destination $d(k)$ is defined randomly. The shelter costs are randomly generated as integers in $[\Lambda, 2\Lambda]$.

For our first results, we use the column generation framework to obtain a lower bound for a network design with resource regeneration. We experiment with three algorithms: CG0%, CG2%, and CG2%+. CG0% is the column generation algorithm that uses any feasible solution as its starting point and iterates until all necessary columns are inserted into the restricted formulation RLMP2. CG2% is similar to CG0%, but stops when $gap \leq 2\%$. CG2%+ also stops when $gap \leq 2\%$, however, it uses the best solution among ten trials of the first heuristic as the starting point. Table 5-3 shows the average computational time for 50 different tests of the same network size. The first two columns specify the graph size and the number of communication pairs. The next three columns present computational time in seconds for CG0%, CG2% and CG2%+. The last three columns present the ratio between CG0% and CG2%+, in regard to the computational time taken by CG2%. From these last three columns, one can observe that stopping the column generation within 2% of the optimal solution greatly increases the algorithm's speed for larger problems. However, using the best solution of the first heuristic as the starting point to the column generation actually worsens the computation time for finding a lower bound.

a x b	|K|	CG0% t (sec)	CG2% t (sec)	CG2%+ t (sec)	CG0% ratio	CG2% ratio	CG2%+ ratio
5 x 5	5	1.4	1.3	1.0	112.2%	100.0%	79.2%
	10	11.2	10.5	9.3	106.8%	100.0%	88.7%
	15	28.6	27.5	24.9	104.0%	100.0%	90.7%
6 x 6	5	33.3	13.0	29.8	255.4%	100.0%	229.1%
	10	307.2	124.3	187.1	247.2%	100.0%	150.5%
	15	1079.5	344.3	412.7	313.5%	100.0%	119.9%
7 x 7	5		181.7	145.6			
	10		2164.2	2224.9			
	15		7246.1	7693.0			

Table 5-3: Computational time for CG algorithm

Table 5-4 presents the average gaps from CG2% and CG2+ in relation to CG0%, calculated as

$$\text{CG2\% } gap = \frac{\text{CG2\% LB} - \text{CG0\% LB}}{\text{CG0\% LB}} \text{ and CG2\%} + gap = \frac{\text{CG2\%} + \text{LB} - \text{CG0\% LB}}{\text{CG0\% LB}}.$$

As one can see from Table 5-4, results from CG2% and CG2% are close to those obtained through CG0%.

a x b	|K|	CG2% gap	CG2%+ gap
5 x 5	5	0.09%	0.05%
	10	0.04%	0.05%
	15	0.02%	0.02%
6 x 6	5	0.19%	0.17%
	10	0.20%	0.22%
	15	0.20%	0.21%

Table 5-4: Gap from CG0%

Based on our limited testing, it seems that the most promising lower-bounding algorithm is CG2%, which is faster and generates higher quality solutions than CG2%+. However, its use is limited for graphs with more than 50 nodes, as we can see from the last line in Table 5-3.

Table 5-5 presents the computational time to solve problems using the two heuristics developed in section 5.4. H1 corresponds to the execution of the first heuristic presented in sub-section 5.4.1 with ten different orderings of set K. H2 corresponds to the execution of the second heuristic with $\alpha = 0.1$. The empirical tests identify the first heuristic as the superior one in both computational time and in the solution values. Table 5-6 displays the solution values obtained by the two heuristics.

a x b	*/K/*	**H1** *t (sec)*	**H2** *t (sec)*	**H1** *ratio*	**H2** *ratio*
5 x 5	5	0.0	0.3	100%	656%
	10	0.1	0.6	100%	733%
	15	0.1	0.9	100%	749%
6 x 6	5	0.1	0.7	100%	838%
	10	0.2	1.4	100%	860%
	15	0.2	1.9	100%	819%
7 x 7	5	0.1			
	10	0.2			
	15	0.4			
8 x 8	5	0.2			
	10	0.4			
	15	0.5			
9 x 9	5	0.3			
	10	0.6			
	15	1.0			

Table 5-5: Computational time for heuristics

Table 5-6 presents the average gap between the heuristics H1 and H2 to the lower bound, computed using CG0%. We can observe that H1 present good results, particularly if compared to H2.

a x b	*/K/*	**H1** gap	**H2** gap
5 x 5	5	2.09%	22.17%
	10	5.10%	23.13%
	15	6.34%	21.54%
6 x 6	5	1.63%	10.13%
	10	2.46%	11.25%
	15	3.06%	11.16%

Table 5-6: Heuristic gap from CG0%

a x b	*|K|*	H1 gap from CG2%	H1 gap from CG2%+
5 x 5	5	2.00%	2.04%
	10	5.06%	5.05%
	15	6.32%	6.32%
6 x 6	5	1.44%	1.45%
	10	2.26%	2.24%
	15	2.87%	2.86%
7 x 7	5	2.23%	2.22%
	10	5.18%	5.14%
	15	5.15%	5.19%

Table 5-7: Heuristic 1 gap from CG2% and CG2%+

5.6 Conclusion

From the computational results above, we can observe that using CG2% to obtain a lower bound is limited to small graphs. In particular, the computational effort of the column generation algorithm seems to grow exponentially with the number of communication pairs. Therefore, little hope exists for applying the algorithm directly to the Alberta SuperNet network design to identify lower bounds and for consequently evaluating the optimality gap. On the other hand, the heuristic H1 seems promising, and we intend to apply it to the Alberta SuperNet problem.

5.7 References

Balakrishnan, A., T.L. Magnanti, P. Mirchandani. 1997. Network Design. Dell'Amico, M., F. Maffioli, S. Martello, Eds. *Annotated Bibliographies in Combinatorial Optimization*. John Wiley & Sons, Chichester.

Garey, M.R., D.S. Johnson. 1979. *Computers and intractability: A guide to the theory of NP-completeness*. W.H. Freeman, San Francisco.

Magnanti, T.L., R.T. Wong. 1984. Network Design and Transportation Planning: Models And Algorithms. *Telecommunication Systems* **18**(1) 1-55.

Ortega, F., L. Wolsey. 2003. A Branch-And-Cut Algorithm for the Single Commodity Uncapacitated Fixed Charge Network Flow Problem. *Networks* **41**(3) 143-158.

Raghavan, S., T.L. Magnanti. 1997. Network Connectivity. Dell'Amico, M., F. Maffioli, S. Martello, Eds. *Annotated Bibliographies in Combinatorial Optimization*. John Wiley & Sons, Chichester.

Takahashi, H., A. Matsuyama. 1980. An Approximate Solution for the Steiner Problem in Graphs. *Mathematica Japonica* **24**(6) 573-577.

Tcha, D.-W., M.-G. Yoon. 1995. Conduit and Cable Installation for a Centralized Network with Logical Star-Star Topology. *IEEE Transactions on Communications* **43**(2-4) 958-967.

Yoon, M.G., Y.H. Baek, D.W. Tcha. 1998. Design Of A Distributed Fiber Transport Network With Hubbing Topology. *European Journal of Operational Research* **104**(3) 510-520.

6 Network Loading and Technology Selection Problem (NLTSP)

6.1 Introduction

Chapter 5 presented the *Network Design with Resource Regeneration Problem* (NDwRRP), a network design problem where optical fibre trenches and equipment shelters must be installed to provide communication paths for origin-destination pairs, while minimizing trench costs, shelter costs, and assuring shelters along communication paths to be within a distance threshold Λ. A solution for a given NDwRRP would assign a *path* and a *repeater pattern* for each communication pair, with paths identifying arcs in the network where trenches are created to host optical fibres cables, and repeater patterns identifying nodes where shelters are built to host communication equipment. This communication equipment, which can be classified as *multiplexers*[4] or *switchers*[5], is required to reinforce the signal traveling along a path. Once the paths and repeater patterns are chosen, it is necessary to define technology by which the signals travel along their paths, in order to minimize costs of fibre and equipment installation. We call the problems of loading the trenches with an appropriate quantity of fibres and of selecting the technology of travel between shelters the *Network Loading and Technology Selection Problem* (NLTSP).

As presented in Chapters 1 and 2, we are designing a wide-area Internet network that is capable of achieving long distances while simultaneously supporting real-time communication for video conference and video broadcasting. As none of the technologies available for Internet can achieve long distances without introducing delays that invalidate real-time communication, a mixture of different telecommunication technologies is required. In this thesis, we consider three different types of optical signal transport technologies: *Gigabit Ethernet* (GE), *Synchronous Optical Network* (SONET) and *Dense Wavelength Division Multiplex* (DWDM). Among these, only GE is Internet compatible, therefore, in order to have an Internet network in place, users must receive and transmit signals in GE technology.

[4] A multiplexer bundle signals together. Therefore, before attaining their destination, the signals must pass through a de-multiplexer to be separated. The signals are reinforced whenever they pass through a multiplexer.

[5] A switcher converts a signal in one technology to another technology, while reinforcing the signal.

SONET and DWDM are well established technologies for telecommunication purposes, and they provide more capacity per fibre and add less delay to the signal than the GE technology.

We now provide details about each technology. As for capacity, GE has a limited transmission capacity per fibre of 2.5 Gbps (Gigabits per second), compared to 10 Gbps/fibre for SONET and 40 Gbps/fibre for DWDM. Considering costs, DWDM equipment is the most expensive technology (in the order of $70,000 to $100,000), followed by SONET (with prices ranging from $60,000 to $70,000), and GE (with equipment prices ranging from $10,000 to $20,000). We have no specific information about delays for these alternative technologies, but according to the engineers at Bell West Inc., no signal in the Alberta SuperNet should traverse more than four GE multiplexers or switchers. Moreover, these engineers affirmed that SONET and DWDM technologies add insignificant delay to the signal; hence, delay could be ignored for these two technologies. However, instead of limiting ourselves to this statement, the model presented here considers each equipment to have a particular delay associated with it and assumes that all signals must respect a threshold value of Δ_{max} units of delay.

Our model assumes the usage of single-mode optical fibres that will suit all three technologies mentioned above. Fibre cable prices (per metre) depend on the number of fibre strands inside the cable, usually following a stepwise structure as presented in Table 6-1. Hence, if the transmission load requires a cable with a minimum of 20 strands on a 1,000 m road segment, one would use a cable of type 2 which would cost $2,850.

cable type	1	2	3	4	5	6	7
# of strands/cable	12	24	36	48	72	96	144
$/metre	2.25	2.85	3.58	5.35	6.50	7.90	10.63

Table 6-1: Fibre cable prices per strands

Consequently, we have a non-linear price function for fibre used in the final solution. As the quantity of fibre required by a segment depends strongly on the technology used, the solution algorithm must be able to consider trade-offs between capacity and signal delay as well as the corresponding equipment costs incurred by such trade-offs.

This chapter is organized as follows. In the next section we present an alternative representation of the NDwRRP solution, one that is more appropriate to the NLTSP. Section

6.3 presents a mathematical formulation of NLTSP. Section 6.4 describes our proposed tabu search heuristic for the NLTSP and Section 6.5 presents our computational experiments used to evaluate the algorithm proposed. Finally, we present our conclusions and remarks about our approach to the NLTSP in Section 6.6.

6.2 Simplifying the NDwRRP Solution

Before describing the solution to the NLTSP, we first eliminate all unnecessary information from the NDwRRP solution. Given an undirected graph $G=(N,E)$ and a set of communicating nodes K defined as $K=\{(o(k),d(k)):o(k),d(k)\in N, k=1,\ldots,|K|\}$, where $o(k)$ is the origin node and $d(k)$ is the destination node, a solution to NDwRRP is composed of paths p_k and repeater patterns r_k for each communication pair in K, for which the overall network design costs (for equipment shelters and fibre trenches) is minimized. In other words, for each communication pair $(o(k),d(k))\in K$, a path $p_k=\{i_0=o(k),i_1,\ldots,i_{|p|+1}=d(k)\}$ with $|p|+1$ nodes in N, such that $(i_{l-1},i)\in E$ for all $l=1,\ldots,|p|+1$, and a repeater pattern $r_k\subseteq R\cap p\setminus\{o(k),d(k)\}$ are provided as part of a solution to the NDwRR problem. R is the set of all nodes where an equipment shelter is available.

For the NLTSP, the NDwRRP solution can be simplified to a smaller directed graph, because nodes that are within paths do not need to be explicitly accounted for. In order to illustrate this concept, we present a NDwRRP solution sub-graph in Figure 6-1 for communication pairs $K=\{(o_1,d_1),(o_2,d_2),(o_3,d_3)\}$ and path/repeater patterns:

$$p_1=\{o_1,\ldots,s_1,\ldots,s_2,\ldots,d_1\} \qquad r_1=\{s_1,s_2\}$$
$$p_2=\{o_2,\ldots,s_1,\ldots,d_2\} \qquad r_2=\{s_1\}$$
$$p_3=\{o_3,\ldots,o_2,\ldots,s_1,\ldots,s_2,\ldots,d_3\} \qquad r_3=\{o_2,s_1,s_2\}$$

After obtaining a solution from the NDwRRP, one can add shelters to some origin and destination nodes in K so that signals can depart/arrive at those nodes in either SONET or DWDM. For the example given in Figure 6-1, we set nodes o_1 and d_2 as having a shelter in each one of them. Consequently, the new set of nodes containing shelters is $R' = \{o_1, d_2, s_1, s_2, o_2\}$ for the given example.

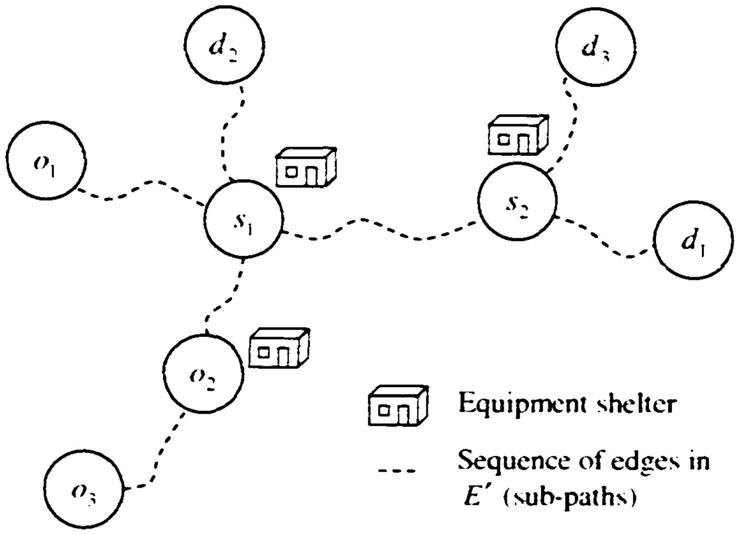

Figure 6-1: Sub-graph from NDwRRP solution

The NLTSP can be defined on a simplified graph $G' = (N', A')$ which can be obtained from p_k and R', where node set N' is composed of origin and destination nodes for all communication pairs, and all nodes with a shelter available, i.e., $N' = \{o(k), d(k) : \forall k \in K\} \cup R'$ and A' is formed by sub-paths in each p_k. Figure 6-2 presents the equivalent graph simplification for the graph presented in Figure 6-1. The following path p'_k and repeater patterns r'_k are now available for the NLTSP, defined only with elements from G':

$$p'_1 = \{o_1, s_1, s_2, d_1\} \qquad r'_1 = \{s_1, s_2\}$$

$$p'_2 = \{o_2, s_1, d_2\} \qquad r'_2 = \{s_1\}$$

$$p'_3 = \{o_3, o_2, s_1, s_2, d_3\} \qquad r'_3 = \{o_2, s_1, s_2\}$$

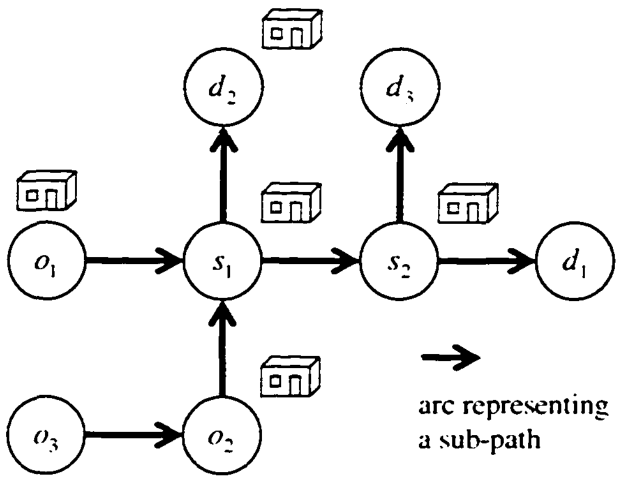

Figure 6-2: Graph G'

Each arc $(i,j) \in A'$ represents a sub-path in the NDwRRP solution and contains one parameter relative to the original path in G, the total length of fibre l_{ij} that must be placed into the ground, which is simply the summation of the length of each edge $(i,j) \in E$ that is inside the sub-path.

6.3 Problem Formulation

Set $T = \{1, 2, 3\}$ represents all technologies available for the NLTSP, 1 corresponding to GE, 2 corresponding to SONET, and 3 corresponding to DWDM. For each $t \in T$, a communication capacity per fibre σ^t is provided; let $\sigma^1 = 2.5$, $\sigma^2 = 10$, and $\sigma^3 = 40$.

Set $\tau = \{(t,t') : \forall t,t' \in T\}$ represent all possible communication equipment that can be installed at a shelter. When $t = t'$, a multiplexer is represented. When $t \neq t'$, a switcher is represented. Equipment $(t,t') \in \tau$ installed at node $i \in R'$ can handle all signals entering node i in technology t and all signals leaving i in technology t', adding a delay $\delta^{tt'}$ to each of the signals traversing it, whenever necessary (later we describe an exception for GE multiplexers at origin and destination nodes). The cost to install equipment $(t,t')\in \tau$ at node i is given by $\rho^{tt'}$.

A communication flow demand ϕ_k (measured in Gbps) is defined for each pair of communicating nodes $k\in K$. Moreover, a maximum signal delay of Δ_{max} is required for each communication pair $k\in K$.

Binary variables X_{ij}^{kt} represent technology selection for the communication k along arc $(i,j)\in A'$. If $X_{ij}^{kt}=1$, signals from communication k traverse arc (i,j) using technology t. Since signal bifurcation is not allowed, $\sum_{t\in T} X_{ij}^{kt}=1$ for all $k\in K$ and $(i,j)\in A'$.

Binary variables $Y_i^{ktt'}$ represent the choice between multiplexers and switchers for communication k at node $i\in R'$. If $Y_i^{ktt'}=1$, the signal from communication k enters node i in signal technology t, and leaves node i in technology t'. Since signal bifurcation is not allowed, $\sum_{(t,t')\in \tau} Y_i^{ktt'}=1$ for all $k\in K$ and $i\in N'$.

In order to illustrate all possible choices of X_{ij}^{kt} and $Y_i^{ktt'}$ for a given communication pair k, we present in Figure 6-3 the sub-graph of choices for the communication pair $k=1$ in the example presented in Figure 6-2. Then, one set of admissible values for X_{ij}^{kt} and $Y_i^{ktt'}$ is presented in Figure 6-4, where $Y_{o_1}^{112}=Y_{s_1}^{123}=Y_{s_2}^{131}=Y_{d_1}^{111}=1$ and $X_{o_1 s_1}^{12}=X_{s_1 s_2}^{13}=X_{s_2 d_1}^{11}=1$.

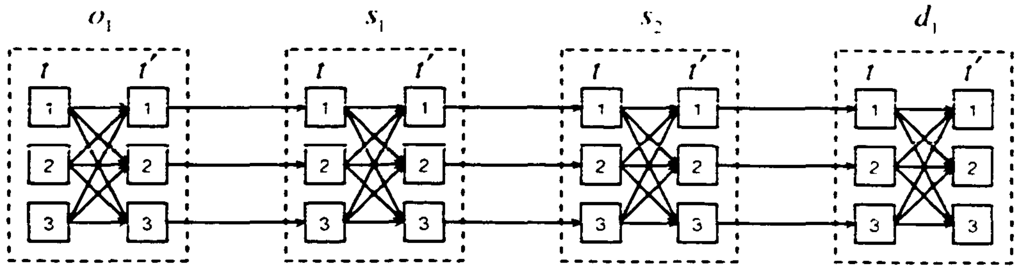

Figure 6-3: Possible choices of binary variables for communication k = 1.

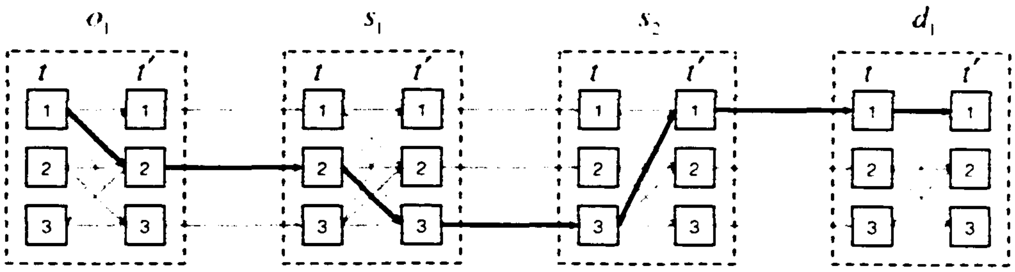

Figure 6-4: One possible solution.

In this particular example, the origin node o_1 has a shelter and consequently the signal may be converted in o_1 to SONET or DWDM to obtain the best solution possible. Therefore, we know that $\sum_{t' \in T} Y_{o_1}^{11t'} = 1$, i.e., the signal enters node o_1 in GE but can leave node o_1 in any of the available technologies. If a shelter was not available at node o_1, $Y_{o_1}^{k11} = 1$ would be used instead to force the signal to leave node o_1 in GE.

The destination node in this example does not have a shelter associated with it, i.e., $d_1 \notin R'$, and therefore $Y_{d_1}^{k11} = 1$. If a shelter was available at node d_1, the constraint $\sum_{t \in T} Y_{d_1}^{1t1} = 1$ would be used instead in order to indicate that any signal could arrive at node d_1 as long as it was converted to GE at node d_1, if necessary.

Binary variables $Z_i^{tt'}$ correspond to the installation of equipment responsible for signal regeneration of type $(t,t') \in \tau$ at node i. If $t = t'$, a multiplexer technology t is installed at node i. If $t \neq t'$, a switcher t to t' is installed at node i. Consequently, if a binary variable $Y_i^{ktt'} = 1$, then $Z_i^{tt'}$ must be equal to 1 to have a feasible solution to the NLTSP. Multiplexers and switchers are capable of handling multiple signals entering and leaving a node, but often require the addition of a card in the equipment that adds an insignificant cost compared to the total equipment cost.

Variables Φ_{ij}^t correspond to the total amount of communication traversing an arc $(i,j) \in A'$ in technology t. It is obtained through summation of all communication flows ϕ_k traversing an arc (i,j). Integer variables θ_{ij}^t correspond to the number of fibre strands required for technology t along arc (i,j), and are greater or equal to Φ_{ij}^t divided by σ^t, the communication capacity per fibre for technology t.

Binary variables Ψ_{ij}^u are equal to 1 if the cable of type $u \in U = \{1,2,3,4,5,6,7\}$ is installed in arc (i,j) (see cable type information in Table 6-1). We use constants γ_u to represent the cost per metre of using a cable of type u and constants ψ_u to represent the number of total fibre strands inside a cable of type u.

The following is a mathematical model that represents the Network Loading and Technology Selection problem (NLTSP):

(IP1) Min

$$\sum_{u \in U} \sum_{(i,j) \in A'} l_{ij} \gamma_u \Psi_{ij}^u + \sum_{\substack{i \in N' \\ (t,t') \in \tau}} \rho^{tt'} Z_i^{tt'} \tag{6.1}$$

S.t.

$$\sum_{t' \in T} Y_{o(k)}^{k1t'} = 1 \qquad \forall k \in K, o(k) \in R' \tag{6.2}$$

$$Y_{o(k)}^{k11} = 1 \qquad \forall k \in K, o(k) \notin R' \tag{6.3}$$

$$\sum_{t \in T} Y_{d(k)}^{kt1} = 1 \qquad \forall k \in K, d(k) \in R' \tag{6.4}$$

$$Y_{d(k)}^{k11} = 1 \qquad \forall k \in K, d(k) \notin R' \tag{6.5}$$

$$\sum_{t\in T} Y_i^{ktt'} = X_{ij}^{kt'} \qquad \forall k\in K, t'\in T, (i,j)\in p'_k, i\in r'_k \quad (6.6)$$

$$X_{ij}^{kt} = \sum_{t'\in T} Y_j^{ktt'} \qquad \forall k\in K, t\in T, (i,j)\in p'_k, j\in r'_k \quad (6.7)$$

$$Z_i^{tt'} \ge Y_i^{ktt'} \qquad \forall k\in K, (t,t')\in \tau, i\in r'_k \quad (6.8)$$

$$Z_i^{tt'} \ge Y_i^{ktt'} \qquad \forall k\in K, (t,t')\in \tau\setminus\{(1,1)\}, i\in\{o(k), d(k)\}, i\in R' \quad (6.9)$$

$$\sum_{i\in r'_k}\sum_{(t,t')\in\tau}\delta^{tt'}Y_i^{ktt'} + \sum_{(t,t')\in\tau\setminus\{(1,1)\}}\delta^{tt'}\left(Y_{o(k)}^{ktt'} + Y_{d(k)}^{ktt'}\right) \le \Delta_{max} \qquad \forall k\in K \quad (6.10)$$

$$\Phi_{ij}^t = \sum_{k\in K}\phi_k X_{ij}^{kt} \qquad \forall (i,j)\in A', t\in T \quad (6.11)$$

$$\Omega_{ij}^t \ge \frac{\Phi_{ij}^t}{\sigma^t} \qquad \forall (i,j)\in A', t\in T \quad (6.12)$$

$$\sum_{t\in T}\Omega_{ij}^t \le \sum_{u\in U}\psi^u \Psi_{ij}^u \qquad \forall (i,j)\in A' \quad (6.13)$$

The first term in the objective function (6.1) calculates the fibre costs, whereas the second term calculates the equipment costs.

Constraints (6.2) guarantee that a signal $k\in K$ originating at a sheltered node $o(k)\in R'$ leaves the node in either GE, or SONET or DWDM, whereas constraints (6.3) guarantee that non-sheltered origin nodes $o(k)\notin R'$ send their signal in GE.

Constraints (6.4) guarantee that a signal $k\in K$ achieving a sheltered destination node $d(k)\in R'$ enters it in either GE, or SONET or DWDM, whereas constraints (6.5) guarantee that non-sheltered destination nodes $d(k)\notin R'$ can only receive a GE signal.

Constraints (6.6) guarantee that every signal $k\in K$ leaving a node i in technology t' is appropriately carried along arc $(i,j)\in p'_k$ in the same technology t'.

Constraints (6.7) guarantee that every signal $k\in K$ traversing arc $(i,j)\in p'_k$ will have either a multiplexer or a switcher assigned to it at node j.

Constraints (6.8) and (6.9) tie variables $Z_i^{tt'}$ to variables $Y_i^{ktt'}$. The first set of constraints consider only the nodes at which a multiplexer or a switcher is required, i.e., those nodes in r_k' for a given signal $k \in K$. The second set of constraints consider the origin and destination nodes for a given signal $k \in K$, at which a switcher may be installed if the signal leaving or entering the node is different than GE; (t,t') is defined for all τ excluding $(1,1)$.

Constraints (6.10) guarantee that each communication k respects the maximum delay threshold of Δ_{max}. Each time a signal passes through a multiplexer or a switcher, an amount of delay $\delta^{tt'}$ is added to the signal. Consequently, for every $Y_i^{ktt'} = 1$ in a communication path k, where $i \in p_k$ and $(t,t') \in \tau$, a delay $\delta^{tt'}$ must be added, except in the cases of the origin and destination nodes. For these nodes, variables $Y_{o(k)}^{k11}$ and $Y_{d(k)}^{k11}$ should not be accounted as having a GE multiplexer in place, simply because the signal does not need any regeneration at these extremities of the path.

Constraints (6.11) attribute variable Φ_{ij}^t with the total communication flow in technology t along arc (i,j), whereas constraints (6.12) attribute variable Ω_{ij}^t with the total number of fibres along arc (i,j) that are necessary to carry the signals in technology t.

Finally, constraints (6.13) tie binary variables Ψ_{ij}^n in order to guarantee that appropriately sized cables will be installed along arc (i,j) to accommodate the required number of fibre optical strands.

6.4 Tabu Search Algorithm for the NLTSP

Tabu Search (TS) is a meta-heuristic first described entirely in Glover (1986), although some of its key elements were presented earlier in Glover (1977), most notably the short term memory (to prevent cycling) and the long term frequency memory (to explore most promising solutions). A meta-heuristic is composed of a general strategy that guides and controls local heuristics and is specifically tailored to the problem at hand. TS has been successfully applied to many problems including machine scheduling, transport network design, multi-commodity location/allocation, hub bandwidth packing, path assignment, vehicle routing, traveling salesman, fixed charge optimization, and many others. Glover and Laguna (1997) not only present a complete list of applications, but also describe TS in detail, and add research improvements to the original proposition of Glover (1986).

A TS algorithm explores the *solution space* by, at each iteration, moving from one solution to the best solution in its neighbourhood. A *move* is an operation that transforms a given solution into another one, generally requiring minimum computational effort for updating the objective function and checking for constraint violation. The *solution neighbourhood* is the set of all possible solutions attainable from the present solution through the application of the *move* operation. The *search space* is the space of all possible solutions that can be visited during the exploration of the solution space.

One key characteristic of TS is that non-improving moves are permitted. Hence we avoid the trap of local optima, a weakness that troubles other local search heuristics. TS uses memory implementations to guide and control the moves; the moves are divided into *Recency-based*, *Frequency-based*, *Quality-based* and *Influence-based* categories, as described by Glover and Laguna (1997). In this chapter, we focus on a simpler tabu search algorithm, one that has only the first two types of memory implementations. Memory can also be classified as *explicit*, if the entire solution is stored, or *attributive*, if only a local characteristic of the solution is stored.

Recency-based memory is a short-term memory that prevents cycling by forbidding the algorithm from going back to recently visited solutions; it is usually implemented through usage of attributive memory. Moves are considered *tabu* for a certain number of iterations (the *tabu tenure*), which can be of fixed length, θ, or randomly picked within a given range, $[\theta_{min}, \theta_{max}]$. According to Gendreau (2003), many articles have discussed the preference for the latter as it reduces the chances for cycling. A *tabu list* is maintained in order to identify those movements considered tabu; moves are eliminated once their tabu tenure has expired. Tabu classifications do not need to be symmetric, i.e., the tabu structure can be designed to give added and dropped elements different tenures.

To avoid missing good solutions, a tabu movement can be overruled through the use of *aspiration criteria* strategies. Among these, one common implementation is the *improved-best* aspiration criterion, in which a tabu movement leading to a solution that is better than the present best solution has it tabu status overruled.

Frequency-based memory, or *long-term memory*, is used to complement the information provided by the tabu list and, in particular, to *intensify* or to *diversify* the search. During intensification, the choice rules are modified to encourage move combinations and solution features that historically have been found to be superior. Intensification can be used to return to attractive regions to search them more thoroughly, especially if one is evaluating only a sub-set of the solution neighbourhood, in a limited but faster movement strategy. During diversification, the goal is to explore solutions that have not been visited before, either by generating solutions with elements not visited earlier by the algorithm, or by the use of penalty functions over solution characteristics that have occurred frequently in all solutions.

Finally, the most common stopping criteria used in TS algorithms are: stop after reaching a maximum limit of number of iterations (or computational time limit); stop after performing a pre-determined number of successive iterations without improvement; and stop upon reaching a solution with an objective function value that satisfies a predetermined threshold value. A TS implementation may use a combination of the above criteria in order to minimize computational effort and to obtain a reasonable solution.

We now detail the TS elements used in our implementation.

6.4.1 Solution representation

A solution to the NLTSP can be fully described by assigning values to the X_{ij}^{kt} variables, and by calculating all other variables after these variables. If we define variables x_{ij}^{k} for each sub-path $(i,j)\in A'$ in a communication flow $k\in K$, x_{ij}^{k} as holding a value 1, 2, or 3 according to the technology assigned to the sub-path on communication k, a *single* move can be defined as the change of one of these x_{ij}^{k} variables from its present value to another value in T, and a *trunk* move can be defined as the change of all x_{ij}^{k} variables sharing a common link $(i,j)\in A'$ from their present value to another value in T.

Let us represent a solution to the NLTS problem through the vector $\mathbf{x} = \{x_{ij}^k : \forall k \in K, (i, j) \in p_k\}$, and use the set $\mathbf{X}$ to represent the search space. Each $\mathbf{x} \in \mathbf{X}$ has an associated neighbourhood $\mathbf{N}(\mathbf{x}) \subset \mathbf{X}$, for which each solution $\mathbf{x}' \in \mathbf{N}(\mathbf{x})$ can be reached from a solution $\mathbf{x}$ by a move operation. A single move that affects one variable x_{ij}^k is named *dropping* (k, i, j, x_{ij}^k) and *adding* $(k, i, j, x_{ij}'^k \in T \setminus \{x_{ij}^k\})$, or alternatively is represented by (k, i, j, t, t'). A trunk move affects all communications pairs $k \in K$ where $(i, j) \in p_k$, and for that we define *dropping* (i, j, t) and *adding* $(i, j, t' \in T \setminus \{t\})$, or alternatively represent it by $(i, j, t, t' \in T \setminus \{t\})$.

6.4.2 Initial solution

In order to verify that a feasible solution exists for a NLTSP instance, one can first find the communication k that requires the highest number of multiplexers and switchers, i.e., $k = \max_{k \in K} |r_k'|$. Given such a path, and knowing that switchers do add more delay than multiplexers, we search a technology $t \in T$ that respects $\delta^{1t} + (|r_k'| - 2)\delta^{tt} + \delta^{t1} \leq \Delta_{max}$. If there is such technology t, a feasible solution can be obtained by setting $x_{ij}^k = t$ for all $k \in K$ and $(i, j) \in p_k$. If not, the NLTSP instance is considered to be infeasible.

Three heuristics were developed to provide an initial solution to the NLTSP. The first heuristic initially sets all x_{ij}^k to 1 (forming a pure GE network). Then, it calculates communication delay for each communication flow, in order to identify those communication flows that violate the Δ_{max} threshold. Once a communication flow k is identified as violating the delay constraint, all links along the path that contain shelters in both end nodes are promoted to SONET, i.e., $x_{ij}^k = 2$ for all $(i, j) \in p_k : i, j \in R'$. Those promoted paths have their communication delay recalculated and, if they again violate the Δ_{max} threshold, the entire path is again promoted to DWDM. If, after this last promotion, the communication delay still violates the Δ_{max} threshold, we know that the problem is infeasible. Since each communication path is independently upgraded in technology to consider only delay constraints, we name this heuristic as Delay Bound Heuristic, or DBH.

Although the first heuristic produces solutions quickly, its results contain many links that transport signals in different technologies, and it does not take advantage of bundling those signals to reduce the number of multiplexers and switchers along the communication paths. The second heuristic compensates for this deficiency by identifying the highest x_{ij}^k value for a given arc $(i,j)\in A'$, and then promoting all signals in parallel along this arc to the highest technology in use along this arc. Since this heuristic considers shared equipment savings, we name it Delay Bound Equipment Saver Heuristic, or DBESH.

Finally, the third heuristic simply sets all links sheltered at both ends to have their signal transported in SONET, and the other links to transport signals in GE. Although it does not guarantee feasible solutions for any equipment delay setting, for the test sets evaluated in the computational results, all solutions were feasible and provided a good starting point for the tabu search algorithm. This last heuristic is simply named SONET Heuristic, or simply SH.

6.4.3 Neighbourhood definition and candidate list strategy

As described in Section 6.4.1 two move operations are considered for this TS implementation: the *single* move, in which only one variable x_{ij}^k changes its value at an iteration; and the *trunk* move, in which all variables x_{ij}^k for a given arc $(i,j)\in A'$ have their value changed to the same value at an iteration. For single moves, the usage of variables to store each communication's delay as well as the flows along arcs and nodes permits a fast recalculation of the objective function and verification of the solution feasibility. For trunk moves, however, recalculating the objective function and checking for feasibility is comparatively more demanding, because a full recalculation is performed instead of a local update.

The size of a neighbourhood for a single move is bounded by $m(|T|-1)$, where $m=\sum_{k\in K}|p_k'|-1$. The exact size of $\mathbf{N}(\mathbf{x})$ can be found by taking into consideration shelters at the origin and destination of the communication paths, however, the upper bound above is sufficient to estimate the size of the solution neighbourhood. The neighbourhood for a trunk move is given by the number of links that have shelters at both ends, and is bounded by $|A'|$.

6.4.4 Recency-based memory and aspiration criteria

In our TS implementation, four arrays are used to store all necessary information needed to handle tabu status and tabu tenure, for both single and trunk moves, and for both adding and dropping actions. For single moves, two three-dimension tables, $\Gamma^{\text{single-add}}$ and $\Gamma^{\text{single-drop}}$, are stored the iteration in which the tabu status for adding and dropping would expire. Similarly, two two-dimension tables, $\Gamma^{\text{trunk-add}}$ and $\Gamma^{\text{trunk-drop}}$, are used for trunk moves.

Initially, all cells would contain a value of zero, meaning that no move is tabu. Later on, if the cell $\Gamma^{\text{single-add}}_{k,(i,j),t'}$ contained a value greater than the current iteration number, we would know that the single move which assigns technology t' to communication flow k along arc (i,j) is tabu, and we would not move in this direction unless the aspiration criterion was satisfied.

As an illustration of tabu tenure setting, if, at the 5[th] iteration, a single move (k,i,j,t,t') was performed and accompanied by a single added tenure of 4 and a single dropped tenure of 3, the value 9 would be stored at $\Gamma^{\text{single-add}}_{k,(i,j),t'}$ and the value 8 would be stored at $\Gamma^{\text{single-drop}}_{k,(i,j),t}$. Similarly, the trunk move (i,j,t,t') would affect cells $\Gamma^{\text{trunk-add}}_{(i,j),t'}$ and $\Gamma^{\text{trunk-drop}}_{(i,j),t}$.

As stated above, a move considered tabu would not be performed unless it satisfied the aspiration criterion. Our implementation uses the *improved-best* aspiration criterion, in which a tabu move leading to a solution better than the best solution obtained so far has its tabu status overruled.

6.4.5 Intensification strategy

Three versions of the TS algorithm were created. The first one, named *SingleTS*, used only the single moves; the second one, named *TrunkTS*, used solely trunk moves; finally, the third one combined both single and trunk moves, and was named simply *NLTS-TS*. During preliminary tests, we observed that improvements from *TrunkTS* largely surpassed those obtained by *SingleTS*. However, we observed that solutions from *TrunkTS* could still benefit from applying *SingleTS* later on, in a fashion similar to the intensification strategy. Results in Section 6.5 reinforce this statement, and perhaps future work can be concentrated on developing more elaborate intensification and diversification strategies.

6.4.6 Neighbourhood evaluation

In our implementation, we maintain a set of auxiliary variables representing the local influence of each x^k_{it} variable on the objective function and on the signal delay associated with communication pair $k \in K$. These auxiliary variables are π^k, $o^{tt'}_i$, and w^t_{ij}, and stand for the present solution delay for communication flow $k \in K$, the communication flow passing equipment $(t,t') \in \tau$ at node i, and the communication flow in technology $t \in T$ traveling along arc $(i,j) \in A'$, respectively. When evaluating a single move (k,i,j,t_{old},t_{new}), a set of temporary variables related to nodes i and j, and link (i,j) is created, for both t_{old} and t_{new}, and the increment in the objective function, cost, is calculated based on the differences in cost associated with implementing the old set of values compared to the new set of variables. Such local calculation is less demanding than recalculating the cost of the new solution, as well as verifying all signal delays.

When a trunk move is evaluated, however, the set of changes in the objective function is more difficult to calculate, as it impacts all communication flows using the link $(i,j) \in A'$. Although the same auxiliary variables could be used to calculate the change in costs and delay times, we decided to recalculate the objective function and verify all communication delays. Hence, evaluating trunk moves is more demanding than evaluating the single moves in the present implementation of our TS algorithm.

6.4.7 Stopping rule

We implemented a combination of two stopping criteria for our algorithm, the maximum number of total iterations and the maximum number of iterations without solution improvement.

6.5 Computational Results

All computational tests were carried out on a HP Pavilion zv5120 laptop with one Pentium Celeron 2800 MHz processor, 512 Mbytes of RAM and Windows XP Home Edition operating system. The algorithm was coded in C++ and compiled with a *g++ 3.3.1* compiler, running in CygWin emulation software.

The test networks used in this computational test have a grid structure, with a rows and b columns, and randomly (uniform) generated integer values for arc length. Each node inside the grid is connected by edges to its four adjacent nodes (left, up, right and down, whenever possible), with length and cost values selected from [60, 70], which would be the average distance between shelters in a solution for the NDwRR problem. Factor *Ksize* determines the number of communication signals in K, where an origin node $o(k)$ and each destination node $d(k)$ are defined randomly. In order to increase equipment and fibre sharing, a tree network was obtained by iteratively creating each path p_k from $o(k)$ to $d(k)$, using the shortest path between those nodes, and setting the lengths of the arcs in p_k to zero so that the next path would use the same links. Communication flows were randomly generated; the flow on an arc is 1 Gbps with a probability of 0.9 and 64 Gbps with a probability of 0.1. This bias was introduced with the objective of achieving possible savings through usage of DWDM over the network.

Other parameters for the test sets are: maximum signal delay, Δ_{max}; tabu tenure boundaries θ_{min}^{add}, θ_{max}^{add}, θ_{min}^{drop}, θ_{max}^{drop}; maximum number of steps, *maxNbrSteps*; maximum number of steps without improvement, *maxNbrStepsWoImpr*; and the type of tabu tenure used, *tenureType*. This last parameter decides if a fixed or a randomly generated tenure will be used for the short term memory; if *tenureType* = 0, the adding tenure is equal to θ_{max}^{add}, and the dropping tenure is equal to θ_{max}^{drop}; if *tenureType* = 1, the adding tenure is randomly chosen from $[\theta_{min}^{add}, \theta_{max}^{add}]$, and the dropping tenure is randomly chosen from $[\theta_{min}^{drop}, \theta_{max}^{drop}]$.

Table 6-2 presents the equipment cost and delay times for different $(t, t') \in \tau$ used in the computational analysis.

	$t' = 1$	$t' = 2$	$t' = 3$
$t = 1$	(10000, 1)	(20000, 1)	(40000, 1)
$t = 2$	(20000, 1)	(15000, 0)	(25000, 0)
$t = 3$	(40000, 1)	(25000, 0)	(35000, 0)

Table 6-2: Equipment cost and delay

A small test set was initially evaluated to check for impact of tabu tenure values. Using $a = 10$, $b = 10$, $|K| = 10$, $\Delta_{max} = 4$, $maxNbrSteps = 500$, and $maxNbrStepsWoImpr = 50$, a set of 50 different instances were evaluated for the values of θ_{min}^{add}, θ_{max}^{add}, θ_{min}^{drop}, θ_{max}^{drop}, and the *tenureType* presented in Table 6-3. The column ratio for each TS implementation is calculated using the best heuristic solution cost as numerator and the TS result as denominator. Consequently, the higher the ratio value, the better is the TS implementation.

			SingleTS		TrunkTS		NLTS-TS	
Tenure	***θ_add***	***θ_drop***	*ratio*	*t (sec)*	*ratio*	*t (sec)*	*ratio*	*t (sec)*
Fixed	10	10	101.5%	0.46	102.1%	0.45	102.3%	0.86
		50	101.7%	0.59	102.3%	0.49	102.4%	0.89
	50	10	101.7%	0.57	102.3%	0.48	102.4%	0.88
		50	101.7%	0.58	102.3%	0.48	102.4%	0.88
		50	101.7%	0.59	102.3%	0.48	102.4%	0.89
		100	101.6%	0.52	102.3%	0.48	102.4%	0.88
	100	50	101.6%	0.52	102.3%	0.47	102.4%	0.88
		100	101.6%	0.52	102.3%	0.48	102.4%	0.88
Random	[5, 10]	[5, 10]	101.5%	0.48	102.2%	0.49	102.4%	0.91
		[5, 50]	101.6%	0.55	102.3%	0.48	102.4%	0.89
	[5, 50]	[5, 10]	101.6%	0.53	102.2%	0.48	102.4%	0.89
		[5, 50]	101.7%	0.55	102.3%	0.48	102.4%	0.88
	[25, 50]	[25, 50]	101.7%	0.58	102.3%	0.49	102.4%	0.90
		[25, 100]	101.6%	0.53	102.3%	0.48	102.4%	0.88
	[25, 100]	[25, 50]	101.6%	0.52	102.3%	0.48	102.4%	0.89
		[25, 100]	101.6%	0.52	102.3%	0.48	102.4%	0.89

Table 6-3: Tabu tenure impact

From Table 6-3, we observe that *NLTS-TS* obtained the best ratio values, followed closely by *TrunkTS*. One may argue that the computational time spent to improve from *TrunkTS* to *NLTS-TS* may not justify such improvement; however, given the strategic nature of telecommunication network design problems and the high costs involved in such enterprises, even a minor improvement may represent savings of thousands or even hundreds of thousands of dollars in installation cost.

The experimentation with tenure values is inconclusive, and we will use random tenure values in the range given by [25, 50].

Table 6-4 presents the average computational time required to run the three heuristics and obtain the initial solution. The ratio, in this particular case, corresponds to the average solution cost value obtained by using the heuristic in the numerator and the cost value of the best heuristic solution in the denominator. Consequently, the lower the ratio, the better is the heuristic solution. We can see from this table that both *DBESH* and *SH* heuristics were very close to the best initial solution, with SH having the advantage that its computational time was less demanding than for DBESH. Nevertheless, we must recall that DBESH can handle different values of equipment delay, whereas SH is limited to providing a SONET network.

DBH		DBESH		SH	
ratio	*t (sec)*	*ratio*	*t (sec)*	*ratio*	*t (sec)*
103.7%	0.006	100.7%	0.007	100.6%	0.001

Table 6-4: Heuristic computational time and ratios

After fixing the tabu tenures to be randomly generated between $[25, 50]$ for both adding and dropping tenures, we tested the NLTS-TS algorithm for different values of a, b, and $|K|$. The results are summarized in Table 6-5. The ratio is calculated by using the best heuristic solution value in the numerator and the NLTS-TS algorithm solution value in the denominator. Accordingly, the higher the ratio, the greater the improvement obtained by the NLTS-TS algorithm.

	a* x *b 10 x 10		***a* x *b*** 12 x 12		***a* x *b*** 14 x 14	
$\|K\|$	**ratio**	**t (sec)**	**ratio**	**t (sec)**	**ratio**	**t (sec)**
10	102.4%	0.69	102.4%	0.98	102.0%	1.68
20	103.5%	1.34	103.1%	1.99	102.7%	3.29
30	104.7%	2.19	104.2%	3.07	103.8%	4.94
40	106.2%	2.98	105.6%	4.42	105.4%	6.65
50	107.5%	4.19	107.1%	5.41	106.7%	9.03

Table 6-5: Extended computational experiment

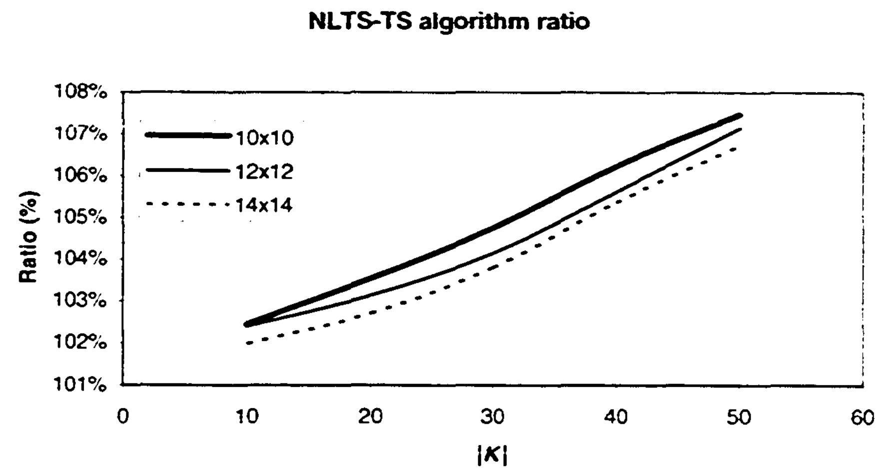

Figure 6-5: NLTS-TS algorithm ratio

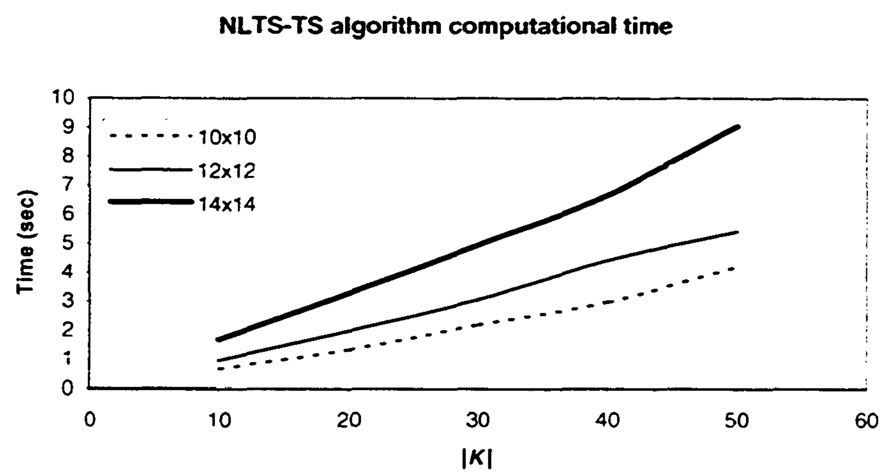

Figure 6-6: NLTS-TS algorithm computational time

From Figure 6-5, we can observe that the NLTS-TS algorithm actually improves its performance as the number of communication flows increase in size. Also, as the size of the network increases from 10 x 10 to 14 x 14, the NLTS-TS algorithm's performance decreases. As one can readily observe, if $|K|$ increases but a and b do not, the number of shared links increase, and savings from signal bundling also rises. In the same sense, if a and b increase but $|K|$ does not, the number of shared links decreases.

From Figure 6-6, we observe that the algorithm's computational effort seems to grow linearly in K, which is encouraging for solving larger problems—perhaps even for the 422 communities that are specified for the Alberta SuperNet project.

6.6 Conclusions and Comments

In this chapter, we presented and solved the *Network Loading and Technology Selection Problem* (NLTSP), a problem that uses the solution from the *Network Design with Resource Regeneration Problem* (NDwRRP) to make a decision about which technology the signals should use to travel along the network, in order to minimize the costs of fibre and the costs of equipment that is needed for signal regeneration.

We presented three starting heuristics, two of them could produce a quick feasible solution to the problem at hand, and a tabu search algorithm that improved the solution obtained through those two initial heuristics.

Based on computational experiments, we concluded that the best approach was to first solve the problem with a sub-set of moves called the *trunk* moves, and then to perform a search intensification after the first part of the tabu search was concluded (either by reaching the maximum number of steps or after reaching a certain number of steps without solution improvement). During the intensification, a set of moves, called *single* moves, was used and these gave the algorithm a chance to obtain smaller changes that could improve the overall solution. The results are promising and, in particular, they suggest that we might be able to solve larger problems, such as the Alberta SuperNet Problem.

Although the mathematical formulation for the NLTSP was presented in this chapter, further work is needed in order to incorporate this formulation into AMPL/CPLEX, to calculate lower bounds to the NLTSP, and to validate the solutions obtained by our tabu search algorithm.

6.7 Reference

Gendreau, M. 2003. An Introduction to Tabu Search. Glover, F., G.A. Kochenberger, Eds. *Handbook of Metaheuristics - International Series in Operations Research and Management Science*. Kluwer Academic Publishers, Boston.

Glover, F. 1977. Heuristics for Integer Programming Using Surrogate Constraints. *Decision Sciences* **8** 156-166.

Glover, F. 1986. Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers And Operations Research* **13** 533-549.

Glover, F., G.A. Kochenberger. 2003. *Handbook of Metaheuristics - International Series in Operations Research and Management Science*. Kluwer Academic Publishers, Boston.

Glover, F., M. Laguna. 1997. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA.

7 Conclusions and Future Research

In this thesis, we studied the wide area telecommunication network design problem. Wide area networks are interesting, because they involve aspects that are often disregarded in smaller network design problems, notably the presence of mandatory and non-mandatory nodes, the usage of different technologies/protocols to balance equipment and fibre costs, constraints on delay created by this usage of different technologies/protocols, and the fixed cost of housing fibres and equipment in the network. In addition to these aspects, we identified an aspect so far ignored by network design researchers, the constraints on distance between signal regeneration equipment. We expect our approach to be a major contribution to the field.

Since the mathematical model presented in Chapter 3 can only be solved for small scale examples, and an algorithm containing all elements from this formulation was impractical, we divided the wide area network design problem into a set of sub-problems: the Constrained Shortest Path with Resource Regeneration Problem (CSPwRRP), the Network Design with Resource Regeneration Problem (NDwRRP), and the Network Loading and Technology Selection Problem (NLTSP). The CSPwRRP finds a minimum cost path where repeaters are installed to respect the maximum distance between repeater constraints. The NDwRRP designs a network that minimizes the costs of placing optical fibre trenches and equipment shelters for multiple signals flowing through the network, using the CSPwRRP as an internal algorithm. Finally, the NLTSP decided in which technology the signals would travel between repeaters, in order to minimize costs of equipment choice and fibre cables, while respecting the signal delay constraints.

In Chapter 4, we presented three successful algorithms to solve the CSPwRRP. The best algorithm was capable of solving problems of up to 12,000 nodes to optimality within a second, using a label-correcting algorithm with an embedded merge sort structure. The CSPwRRP can be seen as an extension of the Constrained Shortest Path Problem, which has received considerable attention in the literature, as detailed in Section 4.1. Another application of the CSPwRRP algorithm involves vehicle routing problems, where refuelling stations are scarce and the vehicle requires multiple refuels on its route. Such a problem appears in the context of alternative fuel vehicles such as propane or electricity.

In Chapter 5, one algorithm to obtain lower bounds on the NDwRRP based on column generation was presented, as well as two heuristics to obtain a solution to the problem. Although our implementation of the column generation algorithm for the NDwRRP was limited to small scale problems, the heuristics proposed were able to solve medium sized problems. All these implementations use the CSPwRRP algorithm as a sub-routine, and consequently, any improvement to the CSPwRRP algorithm will improve the results presented there. The NDwRRP can also be applied to transportation network design, involving either vehicles with limited autonomy, or loads that must receive special services at stations after a certain distance of travel.

The last problem, the NLTSP, was presented in Chapter 6, with a description of a basic tabu search algorithm capable of solving problems with 50 communication flows and 14 by 14 nodes in less than 10 seconds. Due to its computational time behaviour, we believe that problems comparable to the Alberta SuperNet could be solved in a reasonable amount of time. Future research for the NLTSP involves obtaining a lower bound on the problem to estimate the gap between the heuristic and the solutions provided by the tabu search algorithm. As we implemented only basic tabu search mechanisms to solve the problem, there is still a possibility of improving either the quality of the solutions obtained or the time spent to find a reasonable solution.

For future research, a heuristic capable of solving the network design problem as described in Chapter 3 is desirable. Since the best heuristic for the NDwRRP in Chapter 5 produced different solutions for different orderings of communication flows, one could apply the NLTS-TS algorithm described in Chapter 6 to obtain a total cost for each given solution, storing the best of them as a final solution to the network design problem, as described in Section 3.1. Given a large set of possible different orderings, one could use tabu search to explore the solution space and obtain the best solution for the original problem.

One specific aspect of the Alberta SuperNet Project that could become an interesting extension to the models presented here is the possibility of connecting some of the communities through wireless Internet to the network. On the one hand, the wireless links are cheaper to implement, as they do not incur trench costs. On the other hand, they are of limited capacity and introduce a higher delay, which could impact the network design. Nevertheless, one can only evaluate the benefits after implementing the links into the solution. In a network, those wireless links would be located at the periphery of the network. A simple extension to the tabu search algorithm proposed in the paragraph above would merge the demand of a wireless community to a wired community, creating a wireless link between the two communities. The NDwRRP would not consider the wireless communities, as they would not impact the shelter and trench costs. The NLTSP algorithm, however, would need to consider the wireless technology, allocating the proper delay associated with this technology.

Network reliability is another important extension, as we focused mainly on a network design without redundancy. During the literature research, we were particularly interested in designing a partially protected network, where only part of the communication flow would be given alternative paths. Our inspiration came from the fact that the Alberta SuperNet Project was developed with no reliability considerations, to minimize construction costs. We believe that a network design that could "cushion" part of the communication flow while increasing the construction costs by a certain fraction might be more desirable in most applications than a network with no protection. This would be an interesting research direction for the future.