Application of ILP-based Heuristic and Dantzig-Wolfe Decomposition to Solve the Multi-period Survivable Network Augmentation Problem

by

Yali Wang

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Engineering Management

Department of Mechanical Engineering
University of Alberta

Abstract

Multi-period planning is a cost efficient method for designing backbone networks and has been widely used for many years. To ensure the quality of the network service, network survivability has also become a critical requirement in network planning and design. The purpose of this thesis is to take multi-period incremental demands, network survivability and economies of scale into account, and to focus on the optimization of network topology design, working demand routing, and spare capacity allocation. To fulfill this objective, an integer linear programming (ILP) model for a multi-period survivable network augmentation (MPSNA) problem is developed, and the shared backup path protection (SBPP) mechanism is used. However, the MPSNA problem is very time-consuming to solve even for a very small network. To overcome this difficulty, a four-stage ILP-based heuristic method is developed to solve the MPSNA problem. In addition, the effectiveness of the implementation of the Dantzig-Wolfe decomposition for solving the MPSNA is also investigated.

Preface

This thesis work is completed by Yali Wang with the supervision of Dr. John Doucette. The main body of this thesis is produced based on an accepted conference paper and a paper which is ready to submit. See below for details.

Chapter 5 and 6 of this thesis is mainly based on an accepted conference paper: Y. Wang, J. Doucette, "ILP-based Heuristic Method for the Multi-period Survivable Network Augmentation Problem," *Reliable Network Design and Modeling* (*RNDM 2016*), accepted: 17-June, 2016.

Chapter 7 is mainly based on a paper ready to submit: Y. Wang, J. Doucette, "Dantzig-Wolfe Decomposition Algorithm for the Multi-period Survivable Network Augmentation Problem,".

Acknowledgements

I gratefully thank my supervisor Dr. Doucette for his guidance and encouragement during my M.Sc. study. The valuable academic knowledge and advices he provided will benefit me in my whole life. It is my great honor to conduct research under his guidance.

I would like to thank my committee members, Dr. Don Raboud, Dr. Ming Zuo, and Dr. Zhigang (Will) Tian, who took time to attend my defense and made this oral defense happen.

I am also grateful to all group members in my lab for their help and support. I especially thank Brody Toddy for his help and advices at the beginning of my research.

Finally, I would like to thank my family members, my husband, my daughter, my parents, and my parents-in-law, and would like to dedicate this work to them. Without their love, care, support and encouragement, I could not make this success.

Table of Content

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Background

Telecommunication networks play a very important role in modern society. They also have experienced significant traffic increases. For example, global Internet traffic in 2008 was more than twice as much as it was in 2006. The availability of television programs on network websites and different kinds of online services are leading to a dramatic increase in bandwidth demand [1].

With the explosion of telecommunication technologies in areas like personal life, business activity, banking systems and government work, survivable networks are deeply depended upon by society [2]. Once a network failure occurs, the loss caused by the failure can be immeasurable. Table 1 shows varying degrees of impacts of network service outage. As one can see, if the outage time is longer than 10 seconds, the failure will affect a business, which can lead to a huge loss. However, network failure does indeed happen frequently [3]. Hence, how to ensure the survivability of telecommunication networks has become a popular research topic. Meanwhile, many failure protection and restoration techniques have also been developed. As a reference [4], the author gives a detailed introduction to these survivability principles, such as *automatic protection switching* (APS), *shared-backup path protection* (SBPP), *span restoration*, *path restoration*, *p-cycles*, etc.

Table 1: Service outage impacts [5].

Time of outage (second)	0.01-0.1	0.1-1	1-10	10-100	100-5000	5000-1000
Service impact severity	No impact.	5% voice disconnect.	Minor delays, some voice calls dropped, video degradation.	Business impacts. All voice call lost	Business impacts. Application timeout	Severe business impacts.

Based on the geographic scale, telecommunication networks can be divided into *local area network* (LAN), *metropolitan area network* (MAN) and *wide area network* (WAN) [6]. In general terms, telecommunication networks break down to *backbone networks* [7] and *access networks* [8]. An access network usually directly connects to an end user or customers. A backbone network interconnects diverse access networks, collects all the information and provides paths for exchanging information between different access networks so that it is usually made up of high-speed and high-capacity links. The relationship between a backbone network and access networks is visualized in Figure 1.

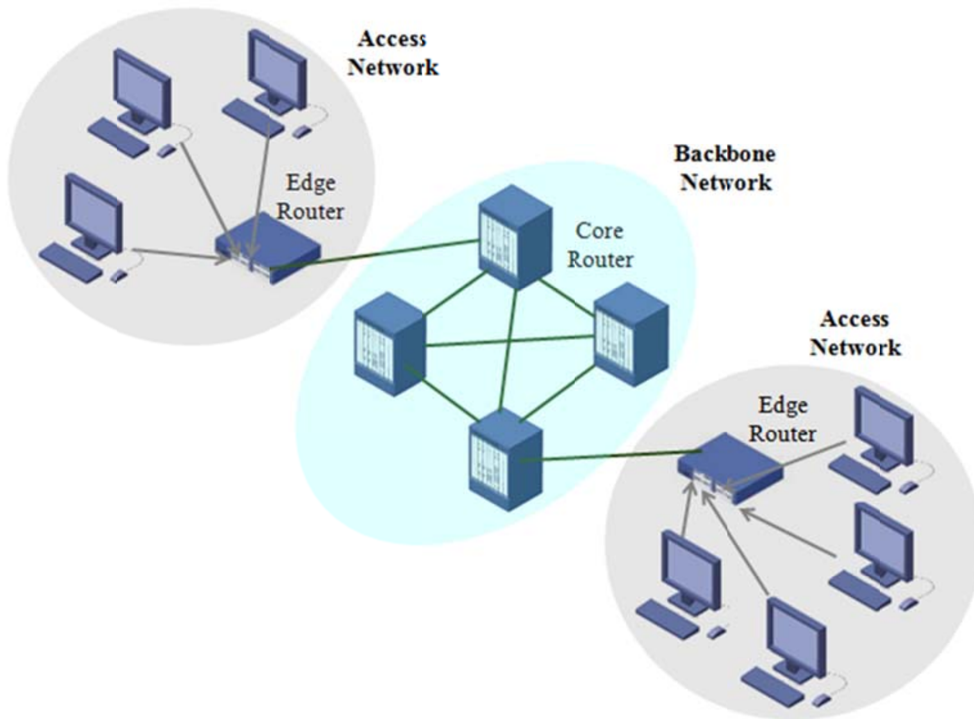


Figure 1: The relationship between a backbone and access networks [9].

Wavelength division multiplexing (WDM) [10] is widely applied in today's backbone networks. The WDM technique enables one single fiber to carry many different wavelengths. In other words, this technique can expand the capacity of networks without laying more fibers [11].

The backbone network is the core of a telecommunication network and other networks that grow from it, and it usually takes significant investment and time to build out. How to build a robust and survivable backbone network with the least money becomes extremely important. On the one hand, the rapid development of technology such as the WDM technique makes it possible to use less money to carry more capacity. On the other hand, with the dramatic growth in bandwidth demand, how to plan a network to adapt to this dynamic and increasing demand is still a challenge.

So far, there are many papers that study the survivable network design problem. Some articles such as [12] and [13] focus on the network topology design issue. Some articles such as [14]-[16] concentrate on the issue of working and spare capacity placement with given network topology. However, these studies all focus on the one period network planning that cannot satisfy the dynamic and incremental demands.

In order to plan networks that adapt to the growth in demand and find a cost efficient way to build a robust backbone network, *multi-period network design* [17], which refers to network design problems that span over a time horizon, is recommended . We also study the topic in this study.

1.2 Thesis Outline

The thesis is organized as follows:

In Chapter 1, we give a brief introduction of our research topic, followed by the thesis outline.

In Chapter 2, we give an introduction to optimization. First, we introduce integer linear programming (ILP). Then, we introduce some classical optimization techniques such as branch-and-bound, column generation, Dantzig-Wolfe decomposition and unimodularity. In the end, meta-heuristic techniques and some custom heuristics are introduced.

In Chapter 3, we introduce network survivability mechanisms and network design paradigms. In addition, we give a review of previous works about optimization of SBPP problems, network topological design and optimization, and multi-period network design and optimization.

In Chapter 4, we describe the motivation and research goals. The proposed methodology is also presented in the chapter.

In Chapter 5, we develop an ILP model for the *multi-period survivable network augmentation* (MPSNA) problem, discuss the complexity of the MPSNA problem, and provide experimental networks that will be used in the chapter and remaining chapters. In addition, we also discuss the experimental study method and present the benchmark solutions of MPSNA problems. Our benchmark solutions show that the MPSNA problem is very time-consuming to solve even for a small network. For example, it takes 24 hours to only get a solution with a 32% optimality gap for even a small 12-node test network.

In Chapter 6, to reduce the solution runtime of the MPSNA problem, we apply a *four-stage ILP-based heuristic* approach based on specific structural characteristics of the MPSNA model to solve this problem. The details of this approach and experimental study method are presented, followed by results and discussion. Our results show that the four-stage ILP-based heuristic approach is very effective for solving the MPSNA problem with a reasonable optimality gap.

In Chapter 7, we investigate the effectiveness of implementing of the Dantzig-Wolfe decomposition to improve the four-stage ILP-based heuristic approach. We find that the ILP model of the MPSNA problem is suitable for Dantzig-Wolfe decomposition and reformulate the ILP model of the MPSNA problem with the implementation of the Dantzig-Wolfe decomposition algorithm. Our experimental results demonstrate that Dantzig-Wolfe decomposition is not an effective algorithm for solving a problem whose sub-problems are unimodular. A detailed results analysis is discussed in the chapter.

In Chapter 8, we summarize this thesis and list contributions of this study, followed by a brief description of possible future research opportunities.

Chapter 2: Introduction to Optimization

2.1 Integer Linear Programming

A *linear programming* (LP) problem is an optimization problem in which we attempt to maximize (or minimize) a linear function of the continuous decision variables and the values of the continuous decision variable must satisfy a set of constraints [18]. The algebraic form of a LP is as follows:

Objective function: Maximize: $$z = \sum_i c_i x_i \qquad (2.1)$$

Constraints: Subject to: $$\sum_i a_{ij} x_i \le b_j \qquad (j = 1, 2, \ldots\ldots n) \qquad (2.2)$$

Bounds: $$x_i \ge 0 \qquad (i = 1, 2, \ldots\ldots m) \qquad (2.3)$$

c_i, a_{ij}, b_j are the model parameters and x_i is a decision variable. If one or more variables are restricted to integer values in the above model, we can call this model an *integer linear programming* (ILP) problem [19].

It is common to see that many ILP problems have a huge number of variables and constraints. Those ILP problems occurring in practice can be extremely large and very hard to solve. In order to effectively solve the large-scale ILP problems, a variety of tactics can be used at the solution phase. For example, adding valid constraints and bounds to the original problem can reduce the ILP problem's search space, or relaxing certain integer variables can also help speed up solutions, or a user-defined MIP gap and runtime limits can be set when we use some solvers such as CPLEX and Gurobi to solve ILP problems [4]. Besides, there are many advanced techniques developed to help solve large-scale ILP problems. In the following sections, we give a description of these techniques.

2.2 Classical Optimization Techniques

2.2.1 Branch-and-bound

Branch-and-bound [20] is a basic and widely used technique for solving ILP problems. The first step of the branch-and-bound approach is to solve a LP relaxation of an ILP problem. LP relaxation means all integer variables in an ILP problem should be relaxed to continues variables. After solving the LP relaxation of an ILP problem, if the results are integer solutions, we can stop as the solutions are optimal. However, if the results are fractional solutions, one real-valued integer variable needs to be selected for branching. Then two new sub-problems are generated, each with more restrictive bounds for the branching variables. We continue to solve each sub-problem and branch the fractional solutions until all branches are formed. *Cutting plane algorithm* [21] is another widely used technique to tighten the LP relaxation. It can be integrated with the branch-and-bound approach to generate a new algorithm, which is called *branch-and-cut*

[22]. Nowadays, some commercial ILP solvers such as CPLEX and Gurobi usually use a branch-and-cut algorithm to solve ILP problems, as in [18], [23] and [24].

2.2.2 Lagrangian Relaxation

Lagrangian relaxation [25] is a mathematical programming relaxation method that is being increasingly used in large-scale ILP applications. The general idea of Lagrangian relaxation is to relax the problem by removing the "hard" (difficult) constraints and putting them into the objective function with appropriate *Lagrangian multipliers* (weights) by which we are penalized for not satisfying the hard constraints. The relaxed problem of maximizing or minimizing the objective function of the Lagrangian multipliers is called the *Lagrangian dual* problem. Properly solving the Lagrangian dual problem and selecting an appropriate set of values for Lagrangian multipliers can result in a tight bound. In practice, the relaxed problem can be often solved more easily than the original problem as the "hard" constraints have been moved to an objective function. After solving the relaxed problem, a lower or upper bound for the original problem can be acquired. Adding this bound back to the original problem can reduce the solution space, thereby speeding up solutions [4][25].

2.2.3 Column Generation

Column generation [26] is an efficient algorithm for solving large-scale LP problems that contain huge numbers of variables (columns). For those problems, the value of most of the variables is zero in the optimal solutions; only a subset of variables needs to be considered. As a result, it is not necessary to tabulate all columns when we use column generation to solve LP problems. With the application of the column generation algorithm, the original LP problem is divided into a master problem and a sub-problem. The master problem is solved with a small set of variables. Then, the sub-problem is solved to decide which other variables are needed to access to the master problem. Column generation repeats this process until its solution is optimal [18][27]. In order to effectively solve large-scale ILP problems with a huge number of variables, the column generation algorithm can be integrated with the branch-and-bound approach. The combination of these two algorithms generates a new algorithm called *branch-and-price.* The main idea of branch-and-price is to apply column generation at each branch-and-bound tree [27]. One of the most successful applications of column generation or branch-and-price is to solve the LP or ILP *cutting-stock problem* [27].

2.2.4 Dantzig-Wolfe Decomposition

Dantzig-Wolfe decomposition [29] is another efficient technique for solving large-scale LP problems whose coefficient matrices have a *block angular structure.* The description of a block angular structure is visually expressed in Figure 2. As shown in Figure 2, the *A* matrix represents the coupling constraints and each *B* represents the independent sub-matrices. In a LP model, if we can find one or more independent constraints linked by coupling constraints, then this model can be effectively solved by Dantzig-Wolfe decomposition algorithm. This approach translates an original problem to an equivalent master problem that has fewer constraints but much more columns than the original problem. The equivalent master problem can be solved by the column generation algorithm [19][27]. The Dantzig-Wolfe decomposition algorithm will be discussed in detail in this section, as it is one of the algorithms used in this study.

$$\begin{array}{cccc} A_1 & A_2 & \cdots & A_n \\ B_1 & & & \\ & B_2 & & \\ & & \cdots & B_n \end{array}$$

Figure 2: An example of the block angular structure.

The development of the Dantzig-Wolfe decomposition algorithm depends on the theorems on *convex combination* and the column generation algorithm, respectively [27]. Consider a LP problem whose constraint matrix has a block angular structure:

Original problem:

Minimize $z = cx$ (2.4)

Subject to $Ax = d_1$ (the number of m coupling constraints) (2.5)

$Bx = d_2$ (the number of n independent constraints) (2.6)

$$x \geq 0 \tag{2.7}$$

c, d_1, d_2 are all vectors and A and B are coefficient matrices. We call this problem the original problem.

Assume that the convex polyhedron is bounded.

$$S = \{x \mid Bx = d_2, x \geq 0\} \tag{2.8}$$

According to the theorem of convex combination, any elements x in S can be written as

$$x = \sum_j \lambda_j x^j \tag{2.9}$$

where $$\sum_j \lambda_j = 1,\quad \lambda_j \geq 0 \tag{2.10}$$

Substituting constraint (2.9) into (2.4) and (2.5) and combining constraint (2.10) are seen to comprise a linear program in λ_j. We call this reformulated problem the master problem, which is equivalent to the original problem.

Master problem:

Minimize $$\sum_j (cx^j) \cdot \lambda_j \tag{2.11}$$

Subject to $$\sum_j (Ax^j) \cdot \lambda_j = d_1 \tag{2.12}$$

$$\sum_{j} \lambda_j = 1 \tag{2.13}$$

$$\lambda_j \geq 0 \tag{2.14}$$

x^j are the extreme points of the region S, and λ_j becomes new variable in the master problem [18][19][27].

The master problem is equivalent to the original problem. However, it only has m+1 constraints, which are much less than the number of constraints $m+n$ in the original problem. If n is large, there will be a sizable saving. On the other hand, the master problem will have many columns as the polyhedron S has a huge number of extreme points if n is large. Rather than tabulating all these columns, we can generate columns to enter the basis as needed by solving a column generation sub-problem:

Sub-problem:

Minimize: $$(c - \pi A)x - \rho \tag{2.15}$$

Subject to: $$Bx = d_2 \tag{2.16}$$

$$x \geq 0 \tag{2.17}$$

π and ρ are the *shadow prices* of constraints (2.12) and (2.13), respectively [26]. Here, we perform an operation called *pricing out* [26] to calculate the objective function.

For an ILP problem, if it has a block angular structure and the sub-problems are bounded, we can also reformulate it using the Dantzig-Wolfe decomposition algorithm. In this case, the bounded polyhedron in the LP problem is changed to a finite integer set in the ILP problem. The way of reformulating a pure integer problem using Dantzig-Wolfe decomposition is presented by Vanderbeck [30] and is briefly described below.

If we restrict the variable x in the original problem to an integer and assume that the integer polyhedron is bounded.

$$S = \{x \mid Bx = d_2, x \geq 0 and\ int\ eger\} \tag{2.18}$$

Then any elements x in S can be written as

$$x = \sum_j \lambda_j x^j \tag{2.19}$$

where $$\sum_j \lambda_j = 1,\quad \lambda_j \in \{0,1\} \tag{2.20}$$

Accordingly, formulations of master and sub-problem of the ILP problem take the following forms.

Master problem of the ILP problem:

Minimize $$\sum_j (cx^j) \cdot \lambda_j \tag{2.21}$$

Subject to: $$\sum_j (Ax^j) \cdot \lambda_j = d_1 \tag{2.22}$$

$$\sum_j \lambda_j = 1 \quad (2.23)$$

$$\lambda_j \in \{0,1\} \quad (2.24)$$

Sub-problem of the ILP problem:

Minimize $(c-\pi A)x-\rho$ (2.25)

Subject to: $Bx = d_2$ (2.26)

$x \geq 0$ and integer (2.27)

If we compare the master and sub-problem of the ILP problem with the ones of the LP problem, we can see that the formulations are almost the same, except that λ_j becomes a binary variable rather than a continuous variable and x is restricted to an integer.

It is obvious that the reformulation of the ILP problem gives rise to an integer master problem. But we can solve the LP relaxation of the ILP master problem by using column generation, which provides a tighter bound for the original ILP problem since the LP relaxation of ILP master problem is actually equivalent to the Lagrangian dual problem [30][31]. In addition, the Dantzig-Wolfe decomposition algorithm can also be integrated with branch-and-bound techniques to solve ILP problems [30][32]. Cutting planes can also be added to strengthen the relaxation of each node of branch and bound tree, which is called *branch-price-and-cut*. Even though there are many successful applications of branch-and-price and branch-price-and-cut in industry [33], there are still some challenges when we implement these techniques in practice due to the difficulties of branching [32]. However, we can at least get a tighter bound for the original ILP problem if only the LP relaxation of the ILP master problem is solved by column generation. In other words, the column generation is implemented only in the root node rather than each node of the branch-and-bound tree [32].

In this work, we take the advantage of the block angular structure of the multi-period survivable network augmentation (MPSNA) problem to reformulate the ILP instance of this problem, followed by solving the LP relaxation of the ILP master problem by column generation to get a tight bound. Then, we will add this tight bound back to the original ILP problem and resolve the MPSNA problem.

2.2.5 Unimodularity

Generally, it is hard to solve an ILP problem because of the integer variable. However, there are some exceptions. For example, if one integer problem has the property of *unimodularity* [34], it can be solved as a linear programming (LP) problem and the results can still be kept in integer form.

A matrix is unimodular if every square sub-matrix has determinant 0, 1 or -1 [34]. For an ILP problem, if its coefficient matrix has this property, then this ILP problem is unimodular. The detailed sufficient conditions for a matrix to be unimodular are given in [35].

Unimodularity is a useful property for solving ILP problems because the ILP problems with this property can be solved as LP problems. In other words, the results of variables are still integers even though the integrality of variables is relaxed. Therefore, if an ILP problem is unimodular (e.g., the minimum-cost network flow problem is unimodular naturally) or if it can be modified to a related unimodular problem, then the ILP problem can be solved easily taking advantage of the unimodularity.

2.3 Meta-heuristic Techniques

The optimization algorithms discussed in the previous section such as Lagrangian relaxation, column generation and Dantzig-Wolfe decomposition can be classified as deterministic optimization algorithms. Correspondingly, the *meta-heuristic* algorithms are classified as stochastic optimization algorithms. A meta-heuristic algorithm is a basic framework and a high-level strategy for optimization that seeks to improve on candidate solutions until a near optimal solution is found. Meta-heuristic algorithms have been widely used in recent years since they are applicable to a variety of different problems by only expressing problems in their frameworks with no other dependence on particular problem specifics [3][4]. Genetic algorithm, tabu search and simulated annealing are three well-developed and most commonly used meta-heuristic algorithms [4], [36]-[38].

Genetic algorithm is inspired by ideas of natural evolution. The basic idea of genetic algorithm is that they start from a population of randomly generated individuals and then improve the fitness of all individuals in the population through producing more generations. The problems that can be presented in binary are often solved effectively by genetic algorithms [4][37]. Tabu search is another widely used meta-heuristic algorithm. It starts with an initial feasible solution and then continues to search for better solutions in the neighborhood. One of the features of tabu search is that it avoids searching the solutions that have been previously visited through use of a tabu list, which contains a number of rules to disallow certain moves [38]. Simulated annealing is inspired by the concept of annealing, which is a technique of heating and slowly cooling the metal to strengthen its crystalline structure in a particular fashion [3], and it is designed to search for the global minimum among the local minima. The typical feature of simulated annealing is that it not only accepts the improvements but also accepts deteriorations with some probability to escape local minima, as seen in [3], [4] and [39]. In practice, meta-heuristic algorithms were recently applied to a multi-period network design problem [40].

Meta-heuristic algorithms can also be combined with other algorithms to solve some complicated problems. For example, a new heuristic method has been developed that combines the genetic algorithm and the shortest path algorithm to solve a long-term network design problem [41].

Meta-heuristics have been the promising and popular algorithms in recent years as they are not problem-specific. However, they still have some drawbacks. For example, the main disadvantage of meta-heuristics is that they cannot guarantee an optimal solution or even a feasible solution and they cannot know how far from optimal solutions their results are [3][4].

2.4 Other Heuristic Techniques

2.4.1 Custom Heuristics

Besides the meta-heuristic algorithms discussed above, there are some other custom heuristic algorithms. In practice, well-developed custom heuristics can provide a good solution for problems. However, most custom heuristics are designed to fit specific problems rather than a variety of different problems [42]. Custom Heuristic algorithms are also developed to solve network design problems. For example, an ultra-fast heuristic algorithm is developed based on statistical analysis of experimental data for finding working paths to solve a shared backup path protection problem. The results showed that their proposed custom heuristic algorithm had better overall performance than its ILP [43].

2.4.2 ILP-based Heuristics

The last technique discussed in this section is the ILP-based heuristic method. As the name implies, the ILP-based heuristic method is developed based on the problem's ILP model. The key idea of the ILP-based heuristic method is to reduce the complexity of the original ILP model and then to resolve the simplified ILP model [3]. ILP-based heuristic method is easy to use since only little effort is needed for code development. In addition, the ILP-based heuristic method can often outperform pure heuristics for the same problem [4]. Researchers also have discussed different classed of such heuristics [4].

ILP-based heuristic methods are also widely used to solve network design problems. For example, some researchers usually generate limited numbers of eligible route candidates instead of all distinct routs to reduce the complexity of problems when they use the *arc-path* approach [3][4] to formulate their network design problems. This widely used tactic is one class of ILP-based heuristics. A three-stage ILP-based heuristic solution method is developed to solve a topological transport network design problem [13]. This approach aims to form a smaller topology space when compared to the full network by generating working span first and then augmenting backup spans. In this study, we will slightly modify this three-stage ILP-based heuristic solution method and use the modified heuristic approach, which is called the four-stage ILP-based heuristic approach, to solve the multi-period survivable network augmentation (MPSNA) problem.

Chapter 3: Network Survivability and Network Design and Optimization

3.1 Network Survivability

3.1.1 Introduction

As we briefly discussed in Chapter 1, today's society highly depends on a survivable network, and network survivability has been one of the critical requirements of network planning and design. As a result, many failure protection and restoration techniques have been developed in recent years. General speaking, there are two kinds of survivability mechanisms. They are protection scheme and restoration scheme, respectively. A protection schemes should assign backup paths in advance of failure and be dedicated to protecting against a specific failure such as 1+1 APS, 1:1 APS and shared backup path protection (SBPP) mechanisms. On the other hand, a restoration scheme in when backup paths do not need to be pre-defined but can be configured when a failure arises, such as span restoration and path restoration [3]. The main difference between failure protection schemes and restoration schemes is that the restoration paths used to protect the failure span are nominally re-routed or cross-connected in real time, rather than pre-defined and possibly pre-connected. The main advantage of a restoration scheme is that it is generally more flexible than a protection scheme, and it is easier to adapt to unexpected changes in the network. However, a restoration scheme is often slower than a protection scheme.

Though these differences were once well understand and defined, the differences of these two types of survivability schemes are now more "fuzzy" than their original meanings, with substantial overlap between the two. For instance, many restoration mechanisms can be modified into pre-planned protection mechanisms depending on their specific implementations [3].

In this chapter, we will briefly discuss the following widely used survivability mechanisms: 1+1 APS, 1:1 APS, SBPP, span restoration and path restoration. All of these survivability mechanisms work under a *mesh network* [44].

3.1.2 Automatic Protection Switching (APS)

The automatic protection switching (APS) system [4] can be considered as the simplest class of network survivability schemes in the event of network failure. 1+1 APS [4] is the simplest and most basic one among different kind of APS system [3][45][46]. In 1+1 APS, the same traffic signals are dual-fed into two disjoint paths simultaneously. If a failure occurs on one of two disjoint paths, the receiver will choose traffic signals from the other path. Otherwise, the receiver will utilize signals from any one of two paths and give up another signal. 1+1 APS is an end-to-end path protection scheme and the protection path is pre-defined before the failure arises. The basic operation of 1+1 APS is illustrated in Figure 3, where path 1 and path 2 are two disjoint paths and carry the same traffic signals. Because 1+1 APS is a very simple survivability mechanism, it is widely used when simple protection protocols are needed, especially when capacity efficiency is not the highest concern [3]. However, the redundancy of spare capacity of 1+1 APS is 100%, which is the highest among the redundancy of APS, SBPP, span restoration and path restoration mechanisms.

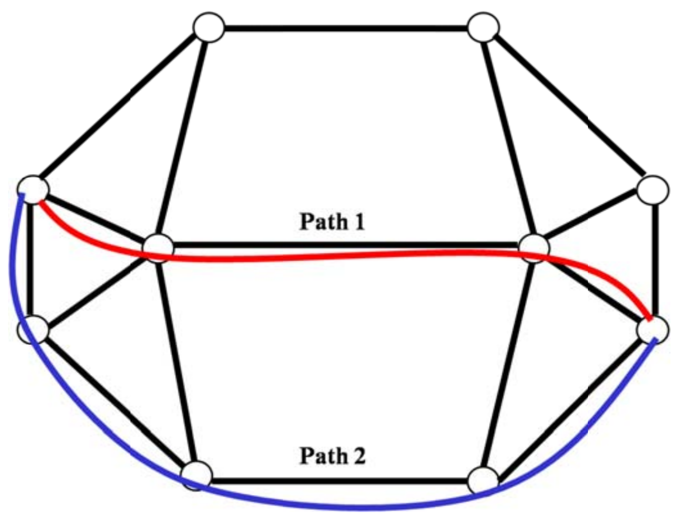


Figure 3: An illustration of the 1+1 automatic protection switching.

1:1 APS [4] is another protection mechanism in APS systems. In 1:1 APS, for every working path, a dedicated protection path must exist and must be pre-defined before a failure, which is same as 1+1 APS. The difference between 1:1 APS and 1+1 APS is that the protection path in 1:1 APS does not need to carry real protection traffic signals until the working path fails. But an unused standby protection path is kept in fully operating condition for use to replace the failed path. 1:1 APS can take advantage of this unused path in maintenance and trouble testing. For example, when a path is tested, signals from this path can be switched to another standby path. However, the protection speed of 1:1 APS is slower than 1+1 APS because the transmit signal is not always bridged for the protection path [4].

3.1.3 Shared Backup Path Protection (SBPP)

The shared backup path protection (SBPP) [47] is similar to 1+1 APS. It is also an end-to-end path protection mechanism and each working path has a fully disjoint backup path that is pre-determined at path-provisioning time. If there is no failure on the network, the traffic signals are carried on the working path only; the shared backup (protection) path is only activated when there is a failure in one of those working paths. Since the same backup path is activated regardless of where the single failure occurs on the working path, SBPP is also called *failure-independent path protection* (FIPP) [3].

The operation of SBPP is illustrated in Figure 4. A-B (blue line) and C-D (blue line) are two fully disjoint working paths. A-B and C-D in red lines are two backup (protection) paths for A-B and C-D working paths, respectively. In SBPP, the path E-D (dash line) can be used to protect both A-B and C-D working paths since we only consider the single-span failure on the network.

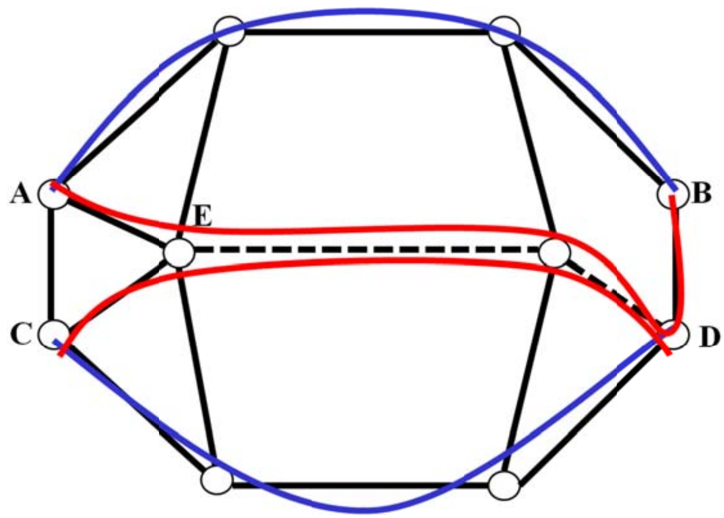


Figure 4: An illustration of the shared backup path protection.

When compared to 1+1 APS, SBPP is much more efficient since the spare capacity on backup path can be shared between fully disjoint working paths. Figure 5 shows the spare capacity of 1+1 APS versus the spare capacity of SBPP. We assume that 1 capacity demand is carried on both working paths A-B (blue line) and C-D (blue line). In 1+1 APS, a total of 7 spare links are used to protect these two primary paths. However, only 5 spare links are needed if the network is protected by SBPP mechanism.

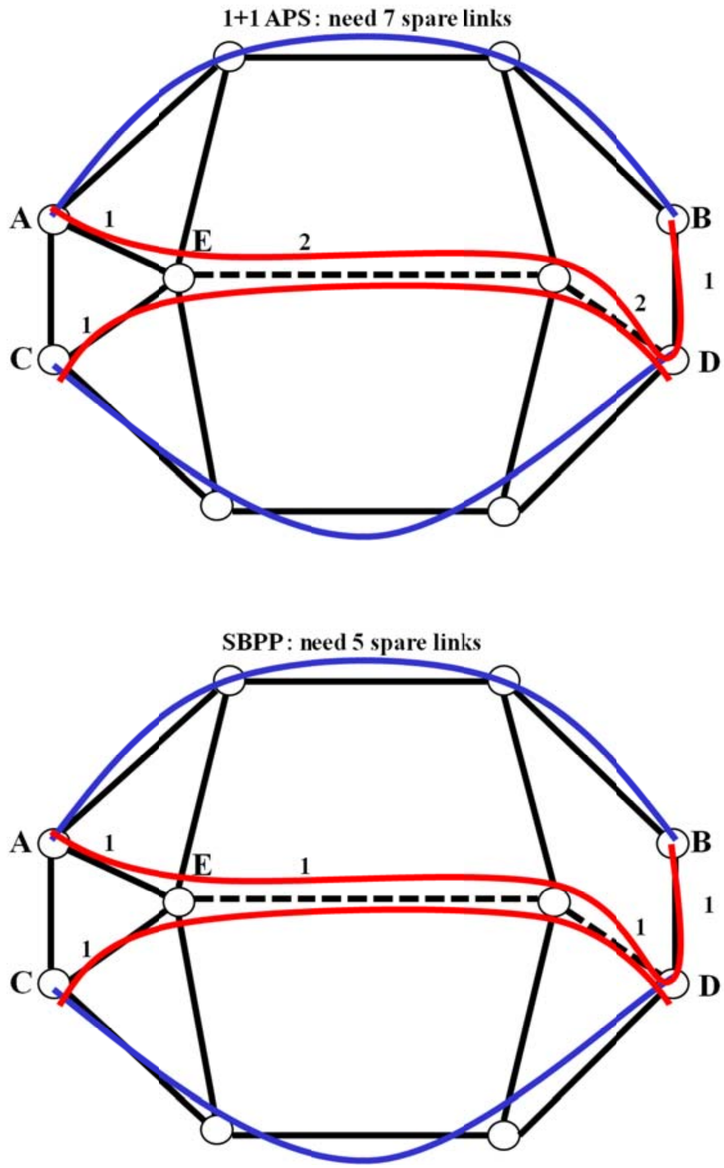


Figure 5: The spare capacity comparison between the 1+1 APS and the SBPP.

Because SBPP has a lot of advantages over other survivability mechanisms, it has become a very popular and widely used network protection scheme in recent years, particularly in IP networks [3]. Here we list several advantages of the SBPP mechanism. First, SBPP is much more efficient than 1+1 and 1:1 APS mechanisms because of the shared backup capacity. It is even slightly more efficient than span restoration since protection is performed between O-D pair nodes instead of between end nodes of the failed span [48]. Moreover, because of the pre-planned response to the failure, the protection speed of SBPP is faster, and the operability is simpler than those restoration schemes such as span restoration and path restoration. Overall, SBPP is an end-to-end protection mechanism that attempts to balance capacity efficiency and operation complexity. Due to the advantages of SBPP, this survivability mechanism is adopted in this study for the design of multi-period survivable networks.

3.1.4 Span Restoration

Span restoration [49] is one kind of restoration scheme, where restoration paths re-route locally between the end-nodes of the failed span. Figure 6 illustrates the operation of span restoration. As one can see, for each working channel on the failed span E-F, the restoration path is established between the end-nodes of the failed span and is formed through any number of distinct routes (e.g., routes E-A-B-F and E-C-D-F). The advantage of span restoration is that the quality of the optical path can be more easily controlled under span restoration when compared to the path-oriented scheme. The drawback of this scheme is that the span restoration network needs more spare capacity than path-oriented protection or restoration network (except for APS) due to its localized response to failure [4].

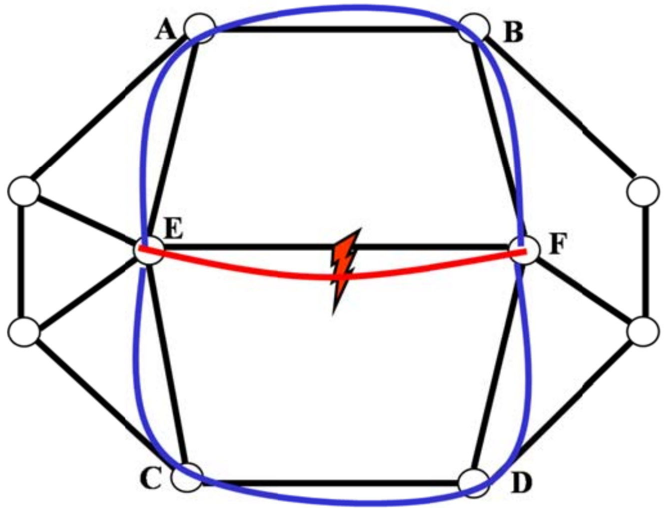


Figure 6: An illustration of the span restoration.

3.1.5 Path Restoration

Path restoration [50] is also a restoration scheme. The failure response mode of path restoration is end-to-end response rather than localized response. The operation mode of path restoration is similar to SBPP, but with some differences. For example, path restoration doesn't need to pre-determine restoration routes for working paths. In addition, path restoration allows *stub-release*, where the survivable portions of all affected light paths are rapidly released prior to restoration and can be re-used by restoration paths of other demands when a failure is detected. Among all of the above survivable mechanisms discussed in Section 3.1, path restoration is the most efficient one due to the feature of stub-release. However, stub-release also complicates the operation of path restoration.

3.2 Network Design Paradigms

Network survivability has been one of the critical requirements in network planning and design since it can maintain the quality of network service under failure scenarios. There are two classical network design problems that are under the consideration of survivability schemes. They are the *spare capacity allocation* (SCA) problem and the *joint capacity allocation* (JCA) problem. In some papers, they are also called *spare capacity placement* (SCP) and *joint capacity placement* (JCP). The SCA problem aims to find a least costly way to assign sufficient spare capacity to spans for full restorability on a network where the working demands have already been routed [4]. There are many studies related to this topic. For example, in [51], the authors develop a spare capacity assignment model with hop limit; and in [52], two heuristic strategies are developed to solve the SCA problem. In contrast to SCA, the JCA aims to minimize the total cost of working and spare capacity. JCA design usually outperforms SCA since it can reduce total capacity through intelligent working flow balance [4]. On the other hand, the JCA problem is much more complex than the SCA problem, especially for a SBPP network. A detailed literature review of the optimization of the SBPP problem is presented in the following section, as it is the mechanism studied in this study.

Many network design problems are studied based on a fixed network topology space, and the cost of the network depends only on the capacity assigned to spans. However, with the dramatic growth in demand, new capacities, new spans or even new nodes may need to be added to an existing network topology. It is well known that adding new spans is expensive and topology change (e.g., deletion of spans) is disruptive to existing services. Hence, network topological design is another concern in the network design problem, especially for long-term (multi-period) network design [4][53]. Network topological design determines where to place the components (nodes) and fibers (spans). Moreover, the problem of topological design with the consideration of network survivability can globally minimize the cost of span installation and capacity placement.

Compared to one period network design, multi-period network design is more flexible and responsive to the incremental demand since the spans can be augmented and capacities can also be expended with the growth in demand over a time horizon In addition, many studies (e.g., [54] and [55]) have shown that multi-period planning can provide cost efficient solutions for long-term operated networks such as backbone networks when compared to single-period network design. Hence, multi-period planning approach is recommended for finding a cost efficient way to build a robust backbone network. Meanwhile, various multi-period planning approaches have also been developed. For example, *all-periods approach*, *incremental approach*, *end-of-life* (EoL) *approach* and *begin-of-life* (BoL) *approach* are four basic and widely used approaches among different kinds of multi-period planning approaches. According to the planning conditions and aims, network designers can decide which approach to use for a specific multi-period network design problem [56].

3.3 Optimization of the SBPP Problem

As we have discussed in Section 3.1.2.2, SBPP is a simple but efficient protection survivability mechanism and is widely implemented in survivable network design problems. Many studies have proven that the SBPP is an NP-hard optimization problem [43], [57]-[59], and so there have been a number of algorithms that have been developed in order to quickly solve this problem [43], [57]-[63].

In [59], a two-phase heuristic approach is developed to solve the SBPP problem. In the first phase, a backtracking-based heuristic method, called CAFES, is used to find a feasible solution (i.e., an eligible pair of working paths and backup paths) for every lightpath request; and in the second phase, a general optimization procedure, called OPT, is applied to the feasible solution found in the first phase to minimize the total cost of working and backup paths. The results show that their two-phase heuristic algorithm leads to better backup sharing than another effective algorithm called full information routing (FIR), which is illustrated in [62]. Work in [60] applies the column generation algorithm to solve a LP relaxation of the SBPP problem. The results show that the medium-sized networks (up to 37 nodes and 57 links) can be solved to optimality within 5 hours. However, for ILP SBPP problems, it takes a longer time to solve. The authors in [57] present a new multi-flow SBPP-type protection mechanism and an accompanying multi-flow optimization ILP model. The difference between the conventional SBPP and the multi-flow SBPP scheme is that the traffic between a single demand pair of a multi-flow SBPP can be routed on multiple working paths rather than a single working path and each working path can be protected by multiple backup paths. The authors also point out that the multi-flow SBPP ILP model is much easier to solve to optimality and there is an average 1.7% of capacity cost reduction compare to the conventional SBPP model. In [58], the authors compare different heuristic algorithms (e.g., naive two-step algorithm, simulated annealing and adaptive large neighborhood search) for solving the SBPP problem and they also present a novel lower-bound method to improve the performance of the heuristics. In above papers, the SBPP problem is usually formulated using an arc-path approach. In [63], an arc-flow approach is used to formulate the SBPP model and a variable-aggregation method is developed to reduce the complexity of the arc-flow formulations.

3.4 Network Topological Design and Optimization

Topological design has been studied for quite a long time. A lot of work has made contributions to this topic, such as [64]-[66], addressing issues of topological design of ring networks, local area networks, and wide area (backbone) networks, respectively. Different types of network topological design problems such as the node location problem, the joint node location and link connectivity problem are introduced in [53]. In [4], besides the topological design model, some widely used algorithms for the network topological design such as the branch exchange algorithm, the cut saturation algorithm and the MENTOR algorithm are also discussed. In the early stage, some research regarding this topic only focused on the network topological design, and some research considered the link capacity design with the topology design simultaneously; however, very little research took the network survivability into consideration. In studies [3] and [13], the authors developed an ILP model for a survivable network topological design problem which is specifically called *the mesh topology, routing and sparing* (MTRS) problem based on the *fixed charge plus routing* (FCR) problem. In their work, the MTRS problem is formulated using the arc-flow approach with the consideration of span restoration mechanism and is proven an NP-hard problem. For instance, for a 20n40s1 full-mesh network, 24 hours of runtime can only provide a solution with an 81.8% optimality gap. As a result, an ILP-based heuristic approach is developed to solve the MTRS problem.

3.5 Multi-period Network Design and Optimization

Many models and algorithms for multi-period design have been developed [53]. The multi-period model for network design is possibly first discussed in [55]. This paper presents a heuristic technique that attempts to satisfy the growth in flow requirement for a dynamic model of the long-haul communication network. When compared to the early work on the design of static multi-flow commodity networks, this dynamic network model makes a great improvement in cost saving. However, this dynamic model is non-restorable. In [67], the authors consider the uncertainties of the demand forecast in a multi-period planning network to optimize the allocation of capacities. They also analyze the trade-off between the method of installing capacity earlier and the method of installing capacity later. However, this study only considers the communication between two nodes rather than a mesh network, and the network restoration is not considered.

In [40] and [68]-[72], the authors focus on using different heuristic methods to solve various cases of multi-period network design problems. In [40], the authors develop a new generic auto-calibrating local search algorithm to solve a multi-period network expansion problem that aims to decide which arc's capacity should be expanded. In this paper, the genetic algorithm, the hybrid algorithm, genetic/tabu search and genetic/simulated annealing are implemented to solve this problem. The computational experiments show that the performance of the new generic auto-calibrating local search algorithm is much better than other algorithms. In [68], the authors formulate an optimization model for the capacitated network design with multiple time periods problem with the consideration of traffic uncertainty, and they propose a hybrid primal heuristic approach to solve this problem. Through testing on a set of 30 instances, their results showed that the hybrid primal heuristic approach outperformed CPLEX. In [69] and [70], in order to solve a multi-period capacity expansion problem for a local telecommunication network with a tree topology, different heuristics are proposed. First, the authors propose a local search heuristic to solve this problem based on different cost and demand structures; and then, they use a genetic algorithm to further improve the solution. In [71], the authors prove that the local access telecommunication network expansion problem (LATNEP) is weakly NP-hard, and they propose a pseudo-polynomial dynamic programming algorithm to solve this problem. Their results indicate that this dynamic programming algorithm is very efficient for solving the LATNEP problem and can solve a real-life problem instance to optimality within minutes. In [72], the fiber optic network expansion problem (NEP) is shown to be strongly NP-complete and three heuristic algorithms are proposed in order to effectively solve the NEP. Through comparing the performance of three heuristic algorithms, the simple greedy algorithm is verified to be an effective method to solve the NEP. However, these papers all study tree-like topology networks and don't take the network survivability into account.

In addition, there are some other articles such as [56] and [73]-[75] concentrate on studies of different multi-period planning approaches' cost impacts on network design. For instance, [56] gives an overview on different multi-period planning approaches and investigates the impacts of input parameters on multi-period network design; [73] aims to develop a new design orientation based on cost optimization and limited budget; [74] proposes a controlled shortest path routing approach for multi-period network design; and [75] compares three multi-period planning approaches based on the considerations of multi-period aspects of uncertainty, reduction of component cost over time and so on. The three approaches are the end-of life-planning approach, incremental planning with forecast approach and incremental planning without forecast approach. The drawback of the above references is that they all neglect the network failure protection or restoration.

Since network survivability is one of the critical requirements in network planning and design, some researchers have begun to consider this factor when they designed multi-period networks in recent years. For example, in [54] the authors develop a multi-period planning model with the consideration of 1+1 APS mechanism for WDM networks, and they also compare the performance of a sequential single-period approach and an integrated multi-period approach for the long-term network design. The paper concludes that the integrated multi-period approach results in an average 4.4% cost saving in comparison with the sequential single-period approach through testing on a variety of problem instances. In [76], the authors analyze the cost impact of different multi-period planning techniques on the requirements of WDM systems and optical transponders. A medium-sized national network that contains 16 nodes and 25 links is considered as an experimental network; however, in order to reduce the complexity of the problem, the traffic of this network is taken as a mix between half unprotected and half 1+1 optical path protected demands. In this paper, the experimental results show that varied design parameters such as different traffic distribution and volume evolution projections can strongly affect the results. In [54] and [76], due to the complexity of the multi-period network design problem, only the simplest survivability mechanism, the 1+1 APS mechanism, is considered. However, as we discussed before, the efficiency of the

1+1 APS mechanism is pretty low and the redundancy is up to 100%. In [77], the authors presents an ILP formulation and give an optimal solution for a demand-wise shared protection (DSP) survivable network augmentation problem. But, only an 8-node network is tested because of the complexity of the problem. The authors in [78] investigates the suitability of two different protection schemes (1+1 APS scheme and SBPP scheme) for multi-period network design using two different multi-period approaches (incremental planning approach and end-of-life planning approach). In this paper, a 14-node-21-edge network is considered as a experimental network and five protection strategies (1+1 APS incrementally planned, 1+1 APS with EoL planning, SBPP incrementally planned with fixed backup path routing, etc) are compared in terms of overall network costs and demand volume. Their results show that the SBPP scheme outperforms 1+1 APS in terms of both flexibility and cost efficiency. However, the main objective of this paper is to analyze two protection schemes' suitability for multi-period networks rather than to focus on the solution quality and solution runtime, and only a 14-node-21-edge network is tested.

Chapter 4: Research Goals and Methodology

4.1 Motivation and Goals

To find a cost efficient way to design robust and survivable networks, the network survivability should be taken into account and multi-period planning is recommended, especially for the design of backbone networks. So far most studies related to multi-period network design mainly focus on developing different heuristic algorithms to solve this NP-hard problem and analyzing the cost impact of different multi-period planning techniques on network design, but neglect the network survivability.

Only few papers consider the network survivability for multi-period network design. However, because of the complexity of the problem, these papers usually adopt the most basic and simplest protection scheme (1+1 APS) while the redundancy of spare capacity of 1+1 APS is 100% and is the highest compared with other survivability mechanisms. More recently, a demand-wise shared protection (DSP) mechanism is chosen for a multi-period network augmentation design. But, only an 8-node network is solved.

SBPP is another popular and promising form of network protection mechanism and it is much more efficient than the 1+1 APS mechanism as the spare capacity on the backup path can be shared between fully disjoint working paths. As we have discussed in Chapter 3, there has been a significant amount of work done with SBPP in recent years.

To our knowledge, however, multi-period network design with the consideration of an SBPP mechanism has not been well studied so far.

The purpose of this study is to take multi-period incremental demands, network survivability and economies of scale into account, and focus on the optimization of network topology design, working demand routing, and spare capacity allocation. To fulfill this objective, we develop an integer linear programming (ILP) model for the multi-period survivable network augmentation (MPSNA) problem, using the shared backup path protection (SBPP) mechanism. It has been shown that the topological design problem and the working and spare capacity allocation problem in an SBPP network are both NP-hard problems, [58] and [79]. The MPSNA model not only combines these two problems, but also incorporates multi-period planning. This makes the MPSNA model very time-consuming to solve even for a very small network. Therefore, a primary goal of the work herein is to reduce the computational time by applying a problem-specific ILP-based heuristic technique and the Dantzig-Wolfe decomposition algorithm.

In summary, the goals of this work can be briefly described as follows:

- Develop an ILP model for the MPSNA problem using the SBPP mechanism.
- Develop a problem-specific ILP-based heuristic method to solve the MPSNA problem to achieve the purpose of reducing the computational time.

- Investigate the effectiveness of the Dantzig-Wolfe decomposition algorithm for solving the MPSNA problem.

4.2 Proposed Methodology

The intent of this study is to develop an ILP model and the associated effective solution procedure for solving the MPSNA problem. More specifically, we adopted the following methodologies in this work:

- Formulation methods: the arc-flow approach and all-periods of multi-period planning approach are adopted to formulate the ILP model for the MPSNA problem.

- Solving method: All ILP formulations are implemented in AMPL (a mathematical programming language), and are solved using Gurobi Optimizer 6.0.3.

- Algorithms: A four-stage ILP-based heuristic approach is applied to solve the MPSNA problem, followed by an implementation of the Dantzig-Wolfe decomposition algorithm that aims to get a lower bound for the above problem.

Chapter 5: Multi-period Survivable Network Augmentation (MPSNA) Problem

5.1 Problem Description

5.1.1 Introduction

It is defined in [53] that "multi-period design refers to network design problems that span over a time horizon in terms of weeks to months, and sometimes even to several years." For instance, we assume the entire planning period is nine years which can be divided into three three-year periods, with new (incremental) demand for each time period, to plan the network to adapt to the incremental demands, one can add either new capacity or new spans in each period [53]. In other words, multi-period design allows each span to increase in capacity and allows new spans to come in and meet the growth in demand over time. As we have discussed before, multi-period planning can provide a cost efficient solution for backbone networks. Hence, it is necessary to use multi-period planning to design backbone networks.

5.1.2 Adopted Approaches

Arc-path and arc-flow are two basic approaches widely used to formulate network design problems. Which approach network design takes usually depends on specific conditions. For example, if the network topology is fixed and pre-defined, an arc-path approach is preferred as we can control the number of the eligible working and restoration routes. As we have discussed in Section 2.1.4, generating a limited number of eligible routes can reduce the complexity of problems. However, if the network topology itself is a decision variable, the preliminary processing required for an arc-path formulation becomes untenable since the number of possible combinations of edges selected (topology space) is huge and it is hard to generate eligible routes for all these combinations [3][4]. This work adopts the arc-flow approach to formulate the multi-period survivable network augmentation (MPSNA) problem since the network topology itself is a decision variable. In this case, the arc-flow approach can avoid generating working and restoration routes ahead.

As we discussed in Section 2.3, there are many multi-period planning approaches such as all-periods approach, incremental approach, end-of-life (EoL) approach and begin-of-life (BoL) approach. Among these techniques, we adopt the all-periods approach, to minimize the total network costs over all periods of time at once. Even though an all-periods approach to formulating the ILP model for the MPSNA problem increases its computational complexity because the topology expansion is included in this problem and it necessitates an arc-flow ILP design, this approach leads to an optimal overall solution which is useful when comparing other approaches. In addition, a greenfield situation is assumed for the MPSNA problem in this work.

5.1.3 Economic Considerations

Since we take the long-term incremental demand into account, economic factors should be considered. However, they are very complex in reality, as is the MPSNA problem. In this study, we consider the following three important economic factors.

- **Time value of money.** The multi-period network design problem spans several periods. The basic idea of multi-period design is to install most of the capacity and spans in the first period according to that period's demand, and then augment them in subsequent periods with the growth in demand. In other words, one does not install capacity and spans all at once, but augments them continuously. In our problem, we need to minimize total network costs over all periods of time at once. However, today's dollar is worth more than next year's dollar due to its earn capacity. Hence, our model applies a discount rate to convert future value to present value.

- **Capacity cost reduction.** Besides the economic factor of time value of money, with the rapid innovation of technology (such as WDM), capacity cost will decrease with time. As a result, an effective discount rate which combines the capacity cost reduction rate with the discount rate of time value of money is applied in our model.

- **Modularity and economies of scale.** WDM technology is now widely used in today's backbone networks, and the capacity is expected to be highly modular in WDM networks since the adoption of modular capacities can aid both users and system designers [4]. Studies [80] have also shown that the modular capacity and the economy of scale have great influences on network topological design, and it also recommend that the WDM network design should move to the truly modular formulation. Economies of scale are another consideration that comes into play when designing a network over multiple time periods, but the effects can be more convoluted than those of time value of money [79]. While there may be an incentive to place larger amount of capacity early due to decreasing per-unit costs as the amount of capacity placed goes up, the granular nature of capacity in various technology options (and perhaps also the unavailability of some technologies or sizes until later time periods) will provide an impulse in the opposite direction. Thus, this work also considers modularity and economies of scale. In [4], one finds a detailed explanation of both terms. Here, we give a simple example, suppose two different sizes of modular capacity systems are available in a backbone network. They are 12 units system and 48 units system, respectively. If the requirement of working capacity on a given span is 22 units, then one 48 units system or two 12 units systems may be required since only the modular capacity can be placed on the given span. To represent economies of scale, we use the notation *"N times Y times"*. For example, a 4-times-2-times cost-scaling factor means that four times of capacity results in a doubling of cost. In this case, the cost of 48 units system is two times that of 12 units system (not four times).

5.2 Basic ILP Model

Suppose we have a given network *G* with node set *N* and span set *S*. Any two nodes constitute each O-D pair. All spans in network *G* are candidates for installation and expansion with incremental demands over *T* time periods. To provide a reliable network against fiber cuts, SBPP mechanism is considered. In addition, this model implements the arc-flow approach and the all-periods approach.

5.2.1 Notations

The ILP model of the MPSNA problem uses following notations:

Sets:

- N is the set of nodes in network *G*, and is indexed by *n*.
- S is the set of candidate edges (spans) in network *G*, and is indexed by *s*. If the network is a full mesh network, it usually includes *N*(*N*-1) spans. However, the number of spans in network *G* is *2N* since we assume all experimental networks have the following character where the number of spans is twice as many as the number of nodes.

- T is the set of time periods in the planning horizon, and is indexed by *t*. The period *t*-1 represents the previous period of period *t*. The period $t = 0$ represents the initial network infrastructure which is before the first period.

- D is the set of all O-D pairs in the network, and is indexed by *r*. Each O-D pair *r* has an origin node and a destination node.

- $S_n \subseteq S$ is the set of all spans connected to node *n*, and is also indexed by *s*.

- $O_r \subseteq N$ is the set of origin node of O-D pair *r*, and is also indexed by *n*.

- $D_r \subseteq N$ is the set of destination node of O-D pair *r*, and is also indexed by *n*.

- M is the set of different module capacities, and is indexed by *m*.

Input Parameters:

- M_1 is an extremely large number. In our model, $M_1 = \sum_{r \in D} \sum_{t \in T} d_t^r$, which means M_1 is equal to the sum of traffics of each demand pair *r* over *t* time periods.

- $c_t^{m,s}$ is the cost of a module of the m^{th} size on span *s* at time period *t*. In our model, there are three different sizes of modules; hence, m = 1, 2, 3. Modular sizes 1, 2 and 3 contain 12 units, 24 units and 48 units of capacity, respectively. In our work, the cost of modular capacity includes the associated cost of lighting the fiber channels, channel interfaces, generation and modulation equipment, and an average distance-amortized cost for modular capacity regenerators [4].

- z^m is the number of capacity units of the module of the m^{th} size.

- f_t^s is the cost of establishing fibers on span *s* at time period *t*. In our work, f_t^s includes the cost of rights-of-way acquisition, ducting, cable and fibers, nodal equipment (WDM multiplexors, de-multiplexors, etc.), and the installation cost of this equipment [4].

- d_t^r is the number of units of traffic demand for O-D pair *r* at time period *t*, and the demand unit increases over time.

Decision Variables

- $\lambda_t^s \in \{0,1\}$ is a topological variable. It is 1 if a fiber is installed on span *s* at time period *t*; otherwise, it is 0.

- $n_t^{m,s} \geq 0$ is the number of modules of m^{th} size placed on span *s* at time period *t*, and this variable is restricted to integers.

- F_t^s is the fixed cost of span *s* at time period *t*. If there is a fiber which has been installed on span *s* at time period *t*-1 already, there will be no fixed cost for span *s* at time period *t*; otherwise, if there is no fiber installed on span *s* at time period *t*-1 and a fiber is needed to be installed on span *s* at time period *t*, there will be a charge for the fiber installation on span *s* at time period *t*.

- C_t^s is the total cost of module capacity on span *s* at time period *t*. The expression of C_t^s is similar to F_t^s. If a cost for the placement of a module has been charged on span *s* at time period *t*-1 already, there will be no cost for this module on span *s* at time period *t*; and only if new modules are added on span *s* at time period *t*, there will be a cost.

- $w_t^s \geq 0$ is the number of units of working capacity assigned on span *s* at time period *t*, and this variable is restricted to integers.

- $p_t^s \geq 0$ is the number of units of protection capacity assigned on span *s* at time period *t*, and this variable is restricted to integers.

- a_t^s is the total capacity (i.e., the sum of working capacity and protection capacity) of span *s* at time period *t*. Since w_t^s and p_t^s are all integer variables and a_t^s is the sum of w_t^s and p_t^s, a_t^s is also an integer variable. Since M_1 is defined as an extremely large number, $a_t^s \leq M_1$.

- $w_{r,t}^s \in \{0,1\}$ is a binary variable. It is 1 if the working flow is routed on span *s* at time period *t* for O-D pair *r*; otherwise, it is 0.

- $p_{r,t}^{s} \in \{0,1\}$ is a binary variable. It is 1 if the protection flow on span *s* at time period *t* is routed for O-D pair *r*; otherwise, it is 0.

- $x_{r,t}^{s,sf} \in \{0,1\}$ is an auxiliary variable, and act as a proxy to the product of $w_{r,t}^{sf}$ and $p_{r,t}^{s}$ (that is, $x_{r,t}^{s,sf} = w_{r,t}^{sf} \cdot p_{r,t}^{s}$). It represents the required protection flow routed on span *s* if span *sf* (failed span) fails for O-D pair *r* at time period *t*. If $x_{r,t}^{s,sf} = 1$, then the protection flow is routed on span *s* to restore the working flows passing over the failed span *sf* for O-D pair *r* at time period *t*. If $x_{r,t}^{s,sf} = 0$, then there are three situations. The first is that if $w_{r,t}^{sf}$=0 and $p_{r,t}^{s}$=0, that means neither working flow nor protection flow is routed on span *sf* and *s* at time period *t*, respectively. The second is that if $w_{r,t}^{sf}$=0 and $p_{r,t}^{s}$=1, the protection flow routed on span *s* is not used to restore the working flows routed on the failed span *sf* but another failed span. The third is that if $w_{r,t}^{sf}$=1 and $p_{r,t}^{s}$=0, there is no protection flow routed on span *s* to restore the working flows routed on the failed span *sf* for O-D pair *r* at time period *t*.

- $w_{r,t}^{n} \in \{0,1\}$ is a binary variable. It is 1 if the working flow is routed over node *n* at time period *t* for O-D pair *r*; otherwise, it is 0.

- $p_{r,t}^{n} \in \{0,1\}$ is a binary variable. It is 1 if the protection flow is routed over node *n* at time period *t* for O-D pair *r*; otherwise, it is 0.

As we mentioned in Section of Input Parameters, the period $t = 0$ represents the initial network infrastructure (before the first period). In our model, there are some input parameters related to period $t = 0$, such as λ_0^s and $c_0^{m,s}$. Since this model is formulated based on a greenfield situation, the value of these variables are all zero.

5.2.1 ILP Model Formulation

The ILP model of the MPSNA problem can be stated as follows.

$$\text{Minimize} \quad \sum_{s \in S} \sum_{t \in T} (F_t^s + C_t^s) \tag{5.1}$$

Subject to:

$$F_t^s = (\lambda_t^s - \lambda_{t-1}^s) \cdot f_t^s \qquad \forall s \in S \quad \forall t \in T \tag{5.2}$$

$$C_t^s = \sum_{m \in M} (c_t^{m,s} \cdot n_t^{m,s}) - \sum_{m \in M} (c_{t-1}^{m,s} \cdot n_{t-1}^{m,s}) \qquad \forall s \in S \quad \forall t \in T \tag{5.3}$$

$$a_t^s \le \lambda_t^s \cdot M_1 \qquad \forall s \in S \quad \forall t \in T \tag{5.4}$$

$$a_t^s \le \sum_{m \in M} (n_t^{m,s} \cdot z^m) \qquad \forall s \in S \quad \forall t \in T \tag{5.5}$$

$$a_t^s \ge a_{t-1}^s \qquad \forall s \in S \quad \forall t \in T \tag{5.6}$$

$$n_t^{m,s} \ge n_{t-1}^{m,s} \qquad \forall s \in S \quad \forall t \in T \quad \forall m \in M \tag{5.7}$$

$$\lambda_t^s \ge \lambda_{t-1}^s \qquad \forall s \in S \quad \forall t \in T \tag{5.8}$$

$$a_t^s = w_t^s + p_t^s \qquad \forall s \in S \quad \forall t \in T \tag{5.9}$$

$$w_t^s = \sum_{r \in D} (w_{r,t}^s \cdot d_t^r) \qquad \forall s \in S \quad \forall t \in T \tag{5.10}$$

$$p_t^s \geq \sum_{r \in D} (x_{r,t}^{s,sf} \cdot d_t^r) \qquad \forall s \in S \quad \forall sf \in S \quad \forall s \neq \forall sf \quad \forall t \in T \tag{5.11}$$

$$x_{r,t}^{s,sf} \geq w_{r,t}^{sf} + p_{r,t}^s - 1 \qquad \forall r \in D \quad \forall s \in S \quad \forall sf \in S \quad \forall s \neq \forall sf \quad \forall t \in T \tag{5.12}$$

$$\sum_{j \in S_n} w_{r,t}^j = 1 \qquad \forall r \in D \quad \forall n \in O_r \quad \forall t \in T \tag{5.13}$$

$$\sum_{j \in S_n} w_{r,t}^j = 1 \qquad \forall r \in D \quad \forall n \in D_r \quad \forall t \in T \tag{5.14}$$

$$\sum_{j \in S_n} w_{r,t}^j = 2 w_{r,t}^n \qquad \forall r \in D \quad \forall n \in N \mid n \notin \{ O_r, D_r \} \quad \forall t \in T \tag{5.15}$$

$$\sum_{j \in S_n} p_{r,t}^j = 1 \qquad \forall r \in D \quad \forall n \in O_r \quad \forall t \in T \tag{5.16}$$

$$\sum_{j \in S_n} p_{r,t}^j = 1 \qquad \forall r \in D \quad \forall n \in D_r \quad \forall t \in T \tag{5.17}$$

$$\sum_{j \in S_n} p_{r,t}^j = 2 \cdot p_{r,t}^n \qquad \forall r \in D \quad \forall n \in N \mid n \notin \{ O_r, D_r \} \quad \forall t \in T \tag{5.18}$$

$$w_{r,t}^s + p_{r,t}^s \leq 1 \qquad \forall r \in D \quad \forall s \in S \quad \forall t \in T \tag{5.19}$$

$$w_{r,t}^n + p_{r,t}^n \leq 1 \qquad \forall r \in D \quad \forall n \in N \mid n \notin \{ O_r, D_r \} \quad \forall t \in T \tag{5.20}$$

Constraints in equation (5.1) seek to minimize total span installation and capacity costs over all time periods. Note that time value of money effects do not explicitly appear in the equations; rather, they are simply incorporated into the costs as they are calculated. Furthermore, economies of scale are applied in subsequent constraints where capacity costs are calculated. Constraints in equations (5.2) through (5.8) describe the multi-period network topology design and capacity modularity. Constraints in equations (5.9) through (5.20) describe the working flow routing and the spare capacity allocation based on SBPP mechanism. The detailed explanation of each equation is presented below.

Constraints in equations (5.2) and (5.3) calculate the span installation cost and the capacity cost of span s at period t, respectively. In equation (5.4), the span decision variables are forced to be 1 as long as capacity is assigned on span s at period t. Constraints in equation (5.5) assign a sufficient number of capacity modules for each size on span s. Constraints in equations (5.6) and (5.7) ensure there is no downgrade for each span's real capacity and the number of each module on span s at period t, respectively. In other words, once the capacity or a module is assigned on span s at time period t-1, it cannot be removed on span s at time period t. Constraints in equation (5.8) ensure that if one span is selected at time period t-1, it must be also selected at time period t.

Constraints in equation (5.9) indicate the total capacity on span s at time period t. The total capacity is the sum of working capacity and protection capacity. Equation (5.10) ensures that enough working capacity is assigned to span s at time period t to accommodate all working flows passing over it. Equation (5.11) ensures that enough protection capacity is assigned to span *s* at time period *t* to accommodate the largest concurrent protection flows for any span failure. Constraints in equation (5.12) calculate the concurrent spare flow on span *s* at time period *t* for restoring the working flow on failed span *sf* for O-D pair *r*. The function of equation (5.12) is to transfer the non-liner equation ($x_{r,t}^{s,sf} = w_{r,t}^{sf} \cdot p_{r,t}^{s}$) to a linear one. In [3], the author presents a classic arc-path formulation of the SBPP problem. A similar equation and a detailed explanation of this equation can be found in this reference. Constraint in equations (5.13) and (5.14) ensure that working demands for original nodes and destination nodes are routed. Constraints in equation (5.15) ensure working nodes have two spans connecting to them. Constraints in equations (5.16) and (5.17) ensure that protection demands for original nodes and destination nodes are routed. Constraints in equation (5.18) ensure that rest nodes also have two spans connecting to them. Constraints in equations (5.19) and (5.20) force working routes and protection routes to be fully disjoint.

5.3 A Discussion on MPSNA Complexity

When we look at the detailed structure of the MPSNA problem, its complexity becomes very clear. If we only consider one time period ($T = 1$), the MPSNA problem reduces to a combined topology and capacity design problem [41][54]. It includes two problems, a topological design problem and a working and sparing capacity allocation problem in a SBPP network. Many studies have shown that the topological design and SBPP problem are both NP-hard problems. For example, in [81], a mesh network topological optimization and routing problem is proven NP-hard and a heuristic algorithm which is called MENTOR is developed to solve this topological design problem. The authors in [57], [58], [61]-[63] and [82] all show that the SBPP problem is a NP-hard optimization problem, and different kinds of heuristic or meta-heuristic algorithms emerged in these papers to effectively solve it. However, the MPSNA problem not only combines these two problems, but also involves multi-period planning, which makes it very time-consuming to solve even for a small network.

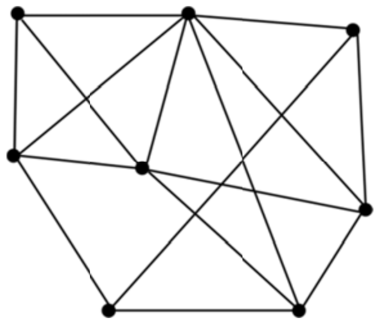
The complexity of the MPSNA problem can be reflected in the integrality of its variables. In this problem, all variables are restricted to integer variables since it is not realistic to allow capacity variables or the variable of number of units of modules to take on real (non-integer) value. In addition, most of integer variables are binary variables such as span-selection (topological) variable λ_t^s, flow variables $w_{r,t}^s$ and so on. It is generally known that a binary variable can only take value 1 or 0. Take the topological variable λ_t^s as an example, if it is 1, then a fiber is installed on span s at time period t; otherwise, it is 0. If we allow this binary variable to take on real value, there will be no meaning for the network span selection. As a result, the integrality of binary variables cannot be relaxed. However, there is no doubt that these binary variables increase the complexity of the MPSNA problem.

In addition, the complexity of the MPSNA problem can also be expressed in terms of numbers of variables and constraints contained in this model [3]. In a small network with 8 nodes and 16 spans, if we set the long-term planning horizon is three time periods, there will be 12,360 integer variables (11,976 binary variables) and 12,844 constraints involved in the MPSNA model. With the increase in the number of candidate spans, the number of variables and constraints grows exponentially. If we take a 30n60s1 network as an example; there will be 2,409,588 integer variables (2,408,148 binary variables) and 2,402,124 constraints in the MPSNA problem. The number of variables and constraints in network 30n60s1 is almost 200 times that in network 8n16s1. As networks become larger and larger, the Gurobi Optimizer cannot solve the MPSNA problem directly. Thus, one recommends decomposition or heuristic approaches to solve it effectively. The detailed algorithm implementation is discussed in Chapter 6.

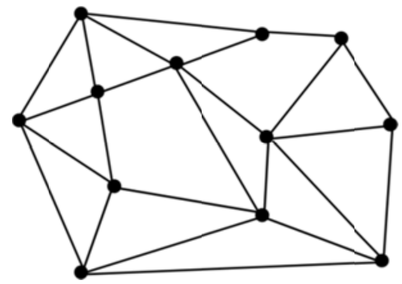
5.4 Experimental Networks

The ILP model of the MPSNA problem presented in this chapter and the heuristic algorithm described in the following chapters are all validated through simulations conducted on six networks. These networks are created as test network topologies by our group members. They are not real transport networks, while they show the characteristic of real transport networks [3].

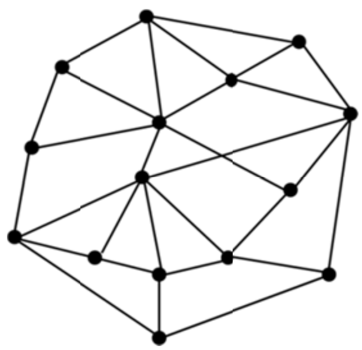
The networks are named as 8n16s1, 12n24s1, 15n30s1, 20n40s1, 25n50s1 and 30n60s1, respectively. The name of networks indicates the number of nodes and spans in their graph. For example, network 8n16s1 contains 8 nodes and 16 spans. The average nodal degree of these networks is 4 [2][3][57], and the number of spans is two times the number of nodes. Figure 7 shows the topologies of these six experimental networks.



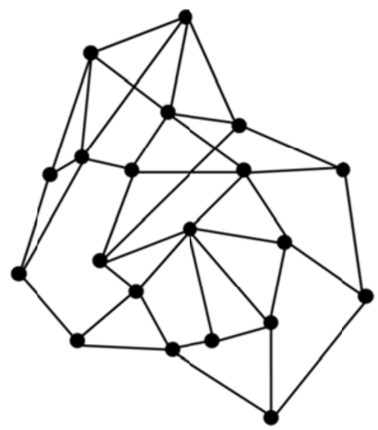
(a): 8n16s1 network.



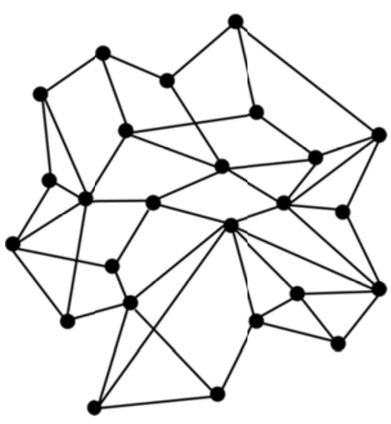
(b): 12n24s1 network.



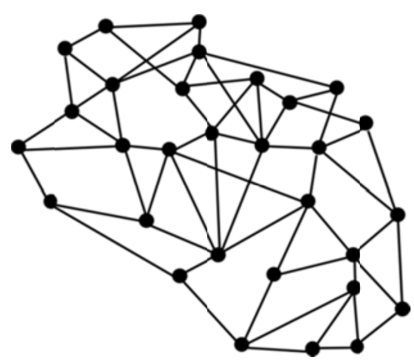
(c): 15n30s1 network.



(d): 20n40s1 network.



(e): 25n50s1 network.



(f): 30n60s1 network.

Figure 7: Network topologies of 8n16s1, 12n24s1, 15n30s1, 20n40s1, 25n50s1 and 30n60s1.

5.5 Experimental Study Method and Considerations

5.5.1 Experimental Study Method

The ILP model of the MPSNA problem was implemented in AMPL (a mathematical programming language), and solved by Gurobi Optimizer 6.0.3 on a Mac Pro server with a 3.5 GHZ processor and a 32 GB RAM. Since an arc-flow approach is used to formulate the ILP model in this work, pre-processing for eligible working and protection routes was not required. However, the data files, which contain some information about network topologies and other input parameters, were prepared on a Lenovo laptop with a 2.4GHZ dual core processor and a 6 GB RAM. Due to the complexity of the MPSNA problem, the experimental cases were solved with time-limited Gurobi runs with various optimality gaps ranging from 1% to 52%. The optimality gap of 1% means the solution is guaranteed to be within 1% of optimal. If we set the optimality gap to be 1%, the solver will stop as soon as it finds a feasible integer solution proved to be within 1% of optimal. To accurately record the solution runtime of each experimental test, no other task is running when the Gurobi is running (that is, the computer is devoted to the Gurobi task full time [3]), and the Gurobi runs only one test each time.

5.5.2 Experimental Study Considerations

This work makes a number of assumptions and considerations due to the characteristics of the MPSNA problem.

- The greenfield situation is assumed. To be specific, there is no span or capacity pre-existing at the beginning of the design. But, it is not hard to formulate the ILP model with some existing spans or capacities. For example, we can just easily add parameters $\lambda_0^s = 1$, which represents that the span *s* is pre-existing to the data file. For the term of the existing capacities, we can add parameters a_0^s which represents the value of capacity pre-installed on span *s* before the first time period to the data file, and since we take the modular capacity into account, we also need to add a constraint $a_0^s \le \sum_{m \in M} (n_0^{m,s} \cdot z^m)$ to the model to make sure there are enough number of different sizes of modules to accommodate the initial capacities.

- Three time periods ($|T| = 3$) are used as the planning horizon in this work. One period can represent one or several years, depending on the requirements of the specific problem. Network designers will consider a number of factors when they select a specific length of time: the network planning budget, the data of demand forecast, etc. If a company has enough available funds and accurate demand data for a short time period, network designers can select one year or even several months as one time period to design a network. Otherwise, they may select a much longer time, even several years. However, one thing we need to know is that the MPSNA problem becomes more complicated when the value of *T* becomes bigger and bigger.

- Half of a full mesh of demands is used in each test case network (wherein half of all node pairs exchange traffic, randomized), where each demand is a uniformly random integer between 1 and 10. Demands increase randomly on each span such that the average increase is a doubling of demand in subsequent periods, as per [77].

- As we discussed in Section 5.1.3, the economic consideration, time value of money, should be taken into account due to the characteristics of the multi-period network planning. As is typical in many business applications, a discount rate of 10% is used to discount the future value to present value, while this value may vary from one industry to another [77]. Formula (5.21) shows how to calculate the present value of a future value.

$$PV = \frac{FV}{(1+i)^t} \tag{5.21}$$

  where PV is the present value, FV is the future value, i denotes the discount rate and t describes the number of periods.

- As we discussed in Section 5.1.3, with the innovation of the technology, the cost of capacity decreases over time in reality. For capacity cost, an effective discount rate of 25% is applied in this work, where the effective discount rate is the combination of the discount rate of time value of money and the capacity cost reduction rate.

Same as the discount rate, the value of the effective discount rate may vary between different organizations and industries [77]. The formula (4.22) calculates the capacity cost at time period t+1.

$$c_{t+1} = \frac{c_t}{(1+\sigma)^t} \tag{5.22}$$

where c_t represents the capacity cost at time period t, c_{t+1} represents the capacity cost at time period t+1, σ is the effective discount rate and t is the number of periods.

- In term of modularity, three module sizes are used in this work. They are 12, 24 and 48 units, respectively. In term of economies of scale, the cost-scaling rule of 4-times-2-times is employed. We assume the cost of 12 units is 12 at the first time period, and costs for other modules at the first time period are generated according to the 4-times-2-times rule. Table 2 shows the cost of all modules.

Table 2: The cost of three sized modules at the first time period [80].

Cost Scheme	**Module size 1 (12 units)**	**Module size 2 (24 units)**	**Module size 3 (48 units)**
4-times-2times	12	17	24

5.6 Benchmark Solutions

The MPSNA model is tested on six experimental networks. They are 8n16s1, 12n24s1, 15n30s1, 20n40s1, 25n50s1 and 30n60s1, respectively. As we mentioned before, the ILP model of the MPSNA problem was implemented in AMPL, and was solved by Gurobi Optimizer 6.0.3 on a Mac Pro server with a 3.5 GHZ processor and a 32 GB RAM.

Table 3 summarizes the benchmark solutions. These solutions are produced through solving the ILP model of the MPSNA problem directly. In this table, the "Network" column represents experimental networks tested in this work. The "Cost" column represents the total cost of installing fibers and capacity for all time periods. The "Runtime" and "Optimality gap" columns represent the Gurobi runtimes and optimality gaps of the MPSNA solutions, respectively. As shown in Table 3, it takes 0.62 hours to solve an 8-node network to a 1% optimality gap. However, the complexity of the problem scales in an exponential-like fashion with the size of the network; the optimality gap increases up to 52% after 48 hours of solution time on the 20-node network. For the 25-node and 30-node test case networks, even feasible solutions were not achievable after 3 weeks of solution time.

The benchmark solutions of experimental networks demonstrate the complexity of the MPSNA problem. As we discussed in Section 5.3, to solve the MPSNA problem efficiently, decomposition or heuristic approaches appear best. In this study, a four-stage

ILP-based heuristic approach and the Dantzig-Wolfe decomposition algorithm are applied to solve the MPSNA problem. The detailed algorithm implementation is discussed in following two chapters.

Table 3: Benchmark solutions of experimental networks.

Benchmark			
Network	**Cost**	**Runtime**	**Optimality gap**
8n16s1	1,936,223	0.62 hours	1.0%
12n24s1	1,069,105	24 hours	32.0%
15n30s1	3,452,152	36 hours	45.8%
20n40s1	3,479,002	48 hours	51.7%
25n50s1	N/A	3 weeks	N/A
30n60s1	N/A	>3 weeks	N/A

To have a better understanding about the MPSNA problem, we take an 8-node experimental network as an example to show the augmentation of capacity and spans with the growth in demand over time. Figure 8 illustrates the topology of an 8-node network, all spans of which are considered as candidate spans for augmentation. Figure 9 through Figure 11 illustrate network topologies and the number of modules installed on span *s* at each time period (M1, M2 and M3 in these figures represent the module size 1, 2 and 3, respectively.).

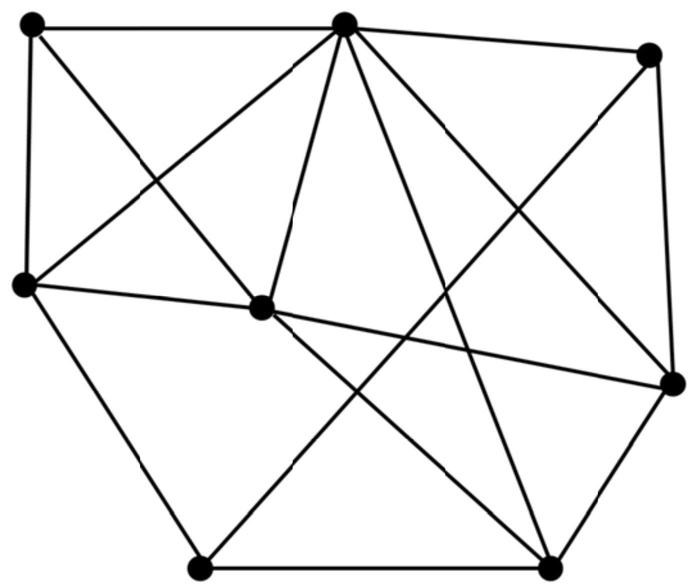


Figure 8: The topology of the 8n16s1 network.

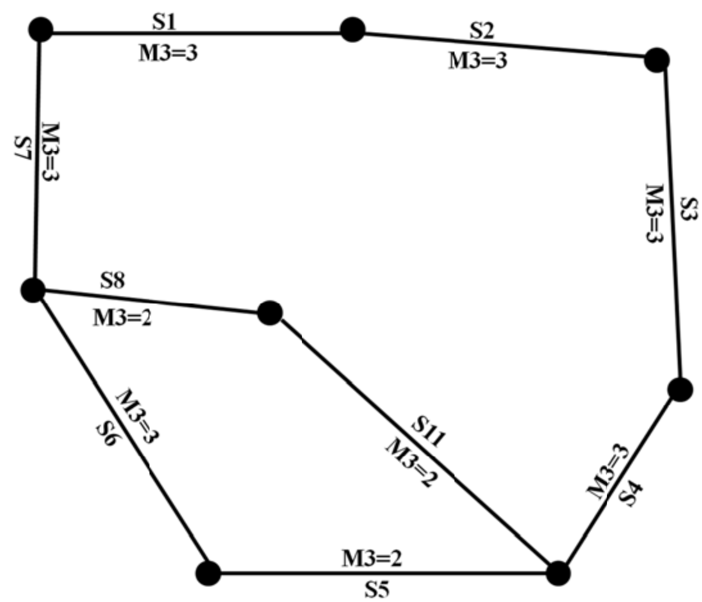


Figure 9: The 8-node network topology with modular capacity at the first period.

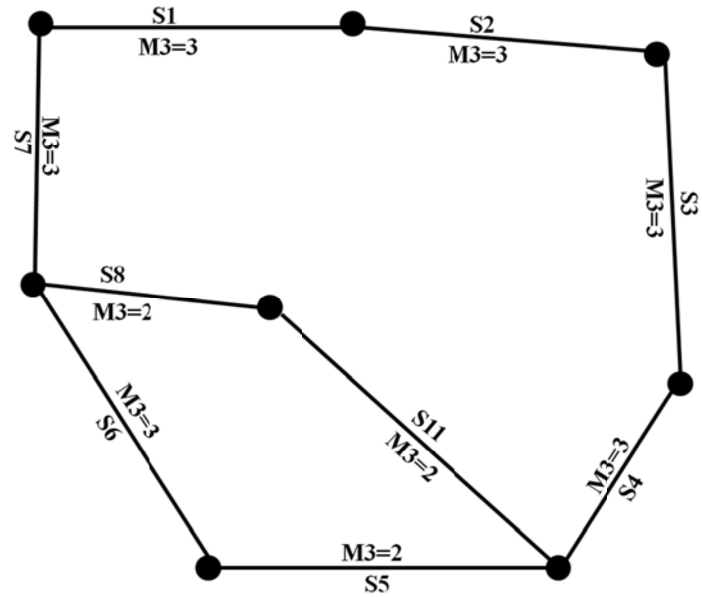


Figure 10: The 8-node network topology with module capacity at the second period.

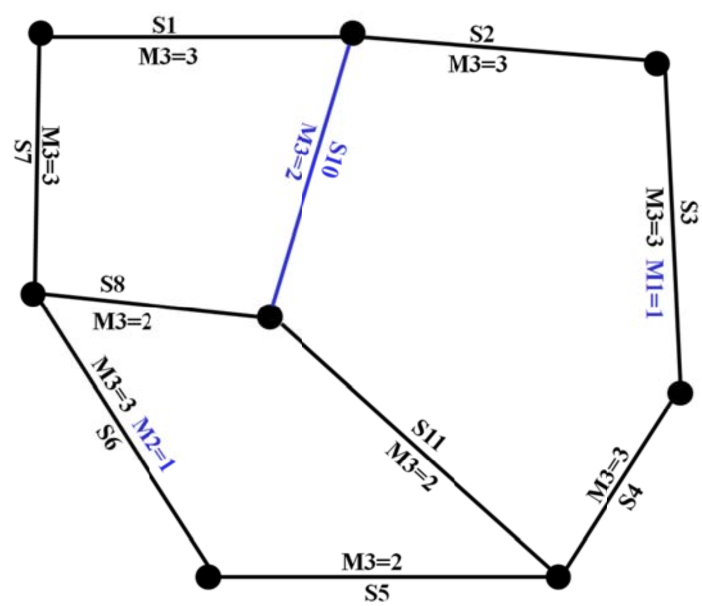


Figure 11: The 8-node network topology with modular capacity at the third period.

As shown in Figure 9 through Figure 11, nine spans are selected at the first period for routing the working & spare capacity and only module size 3 is assigned on each span because of the effect of the economies of scale. We also find that there is no new capacity and spans added to the network at the second time period. The reason is that there is enough capacity at the first period to accommodate the demand of the second time period. As the demand grows continuously at the third period, the new capacity and a new span are added. For example, one unit of module size 1 is added to span 3, one unit of module size 2 is added to span 6 and a new span, span 10, with two units of module size 3 joins the network.

Another observation we can find from Figure 8 through Figure 11 is that the near-optimal (1% optimality gap) 8-node network topology at each time period tend to be quite sparser than the topology of the experimental 8n16s1 network. The number of spans selected at the third period should be the most one among the three time periods; however, only 11 spans are selected at the third time period, which is a 31.25% span reduction compared to 16 spans of 8n16s1 network. In other words, there are 5 candidate spans not useful during the whole process of the multi-period network design; however, they definitely increase the complexity of the MPSNA problem since the number of variables and constraints of the MPSNA problem grows exponentially with the increase in the number of candidate spans. In addition, as discussed in [3], for the topological design problem, the solver has to spend a considerable amount of time looking for a bi-connected graph before it can perform the working and spare routing and capacity placement. There are at least 614 million possible span-selection combinations even for a 15-node-30-span network ($c_{30}^{15} + c_{30}^{16} + ... + c_{30}^{30}$), but only a few of them can form bi-connected graphs that the solver need to spend large amount of time looking for. For the 8n16s1 network, if we remove the 5 useless candidate spans and resolve this problem, we expect that there is a considerable runtime saving due to two benefits obtained from the reduction of the number of candidate spans. The first benefit is a sizeable saving of the number of variables and constraints. The second is a sizeable saving of the number of span-selected combinations and therefore of time looking for bi-connected graphs. As a result, if we can find a high-merit smaller topology space when compared to the full network topology space and solve a restricted version of the MPSNA problem using only a subset of eligible spans rather than all candidate spans, the solution runtime may decrease greatly.

In addition, we compared the total cost of single-period and multi-period network planning on the 8-node network. We solved the single-period problem using only the demand in the final time period, but the assumption is that the costs are all incurred at the present, so time value of money had no bearing on this solution. More specifically, single-period planning refers to a network design in which all spans and capacity are installed all at once rather than augmenting topology and capacity over time.

Table 4 summarizes the cost comparison of single-period planning and multi-period planning of one 8-node network. As we can see, multi-period planning is nearly 4% less costly than single-period planning. This validates our assertion that multi-period planning is more cost efficient than single-period planning.

Table 4: The cost comparison of single-period planning and multi-period planning.

Network	**Single-period planning**	**Multi-period planning**
	Total cost (the value of the objective function)	Total cost (the value of the objective function)
8n16s1	2,012,466	1,936,223

Chapter 6: Four-stage ILP-based Heuristic Approach

6.1 Approach Description

As we discussed in Section 5.6, the number of variables and constraints of the MPSNA problem grows exponentially with the number of candidate spans, which greatly increase the complexity of the MPSNA problem. So does the number of span-selection combinations; thus, the solver has to spend much more time looking for bi-connected graphs among the larger number of span-selection combinations. As a consequence, finding a high-merit and small topology with only subset of candidate spans for each experimental network may be an effective way to decrease the complexity of the MPSNA problem.

In [3] and [13], a three-stage heuristic solution method is developed to solve the mesh topology, routing, and sparing (MTRS) problem, and the experimental results show that this ILP-based heuristic is very effective for solving the MTRS problem. The basic idea of this approach is to form a smaller topology compared to the full network by generating working span first and then augmenting backup spans. As we discussed before, we expect that we can find an effective and high-merit small topology for each experimental network to efficiently solve the MPSNA problem. The three-stage heuristic solution method is exactly the approach which we are looking for. As a result, we will follow the basic idea of the three-stage heuristic solution method and modify this approach slightly for the MPSNA problem. The modified three-stage heuristic solution method is called the four-stage ILP-based heuristic approach in this study, and has four steps.

- Step 1: Identify good spans for working light path routing alone (i.e., no survivability), based on the demands in the final time period in isolation. In this step, we solve a fixed charge plus routing (FCR) problem which aims to determine the network topology and minimize the span selection cost and optimal working capacity routing cost simultaneously. The spans identified in this step is expected to be of high merit for consideration in the MPSNA design since they are play an important role in serving the working demand flows [3] [13].

- Step 2: Augment good spans for spare capacity based on the demands in the final time period in isolation. In this step, we solve a *reserve network fixed charge and sparing* (RN-FCS) problem which aims to minimize the total cost of spare capacity and installation of additional spans to augment the topology from step 1 to become a bi-connected graph and to ensure full restorability of the working flows from step 1 [3] [13].

- Step 3: Refine the spans inherited from step 1 and step 2 based on the demands in the final time period in isolation. In this step, we solve an *one-period topological SBPP* (OPT-SBPP) problem which aims to refine the spans inherited from step 1 and step 2 under the interaction of the working and spare capacity. In other words, some spans inherited from step 1 and step 2 may be eliminated in this step since we consider the placement of working and spare capacity simultaneously.

- Step 4: Resolve a restricted version of the MPSNA problem using only a subset of eligible spans from step 3, rather than all spans from the experimental networks. The subset of eligible spans from 3 forms a smaller bi-connected graph when compared to the topology of the complete MPSNA problem. We assume that this smaller bi-connected graph is a high-quality sub-graph for consideration of the complete MPSNA design since the spans selected for routing working capacity and spare capacity are all expected to be of high merit for the complete MPSNA problem by virtue of their key role in step 1 and step 2. As we discussed before, since the growth of number of candidate spans dramatically increases the complexity of MPSNA problem, reducing the number of candidate spans will result in a runtime saving to a great extent.

The differences between the four-stage ILP-based heuristic approach modified in this work and the three-stage heuristic solution method developed in [3] and [13] are that the span selection in step 1 and step 2 in this work is based on the demand in the final time period; and before we resolve a restricted version of the MPSNA problem using only a subset of eligible spans inherited from step 1 and step 2, we add one more step called step 3 to refine the network topology in order to get a more high-merit topology.

With the growth of demands over time, we expect that new spans are installed and capacities are expanded gradually as time progresses, and so each subsequent time period will make use of a set of spans that is at least as large as that from the previous time period. In other words, if we consider only one time period (i.e., T=1), it is obvious that the number of spans selected based on the demand in the final time period is larger or at least equal to the one of previous time period. Since the core of our approach is to form a smaller topology space (bi-connected sub-graph) when compared to the full network, it is reasonable to generate this high-merit small topology space by using the demand of the final time period. The ILP formulation of each step is presented in the following section.

6.2 Four-stage ILP-based Heuristic Approach

6.2.1 Step 1: Identify Good Spans for Working Capacity

In step 1, we solve a topology design with working capacity allocation problem without the consideration of survivability, and this kind of problem is usually called fixed charge plus routing (FCR) problem. The purpose of step 1 is to identify high-merit spans for routing working capacity. Since we only consider the allocation of working capacity with no concern for network restoration, the network topology of step 1 is expected to be tree-like. In step 1, all spans in network G are candidate spans for selection. The information which is generated from this step and will be forwarded to step 2 is the solutions of variables of span selection (λ^s), working flow routed on span *s* for demand *r* (w_r^s) and working flow routed over node *n* for demand *r* (w_r^n).

As authors discussed in [3] and [13], the FCR problem involved in step 1 is also a difficult problem. In order to speed up step 1, some tactics can be applied in this step. Firstly, since the solutions of variables of working capacity (w^s) and the number of modules ($n^{m,s}$) are useless for the following steps, the integrality of w^s and $n^{m,s}$ can be relaxed. In addition, it is also acceptable to solve the FCR problem with a reasonable optimality gap since the purpose of the step 1 is to identify high-merit edges rather than to get optimal solutions.

The ILP model for step 1 uses the same notations defined in Chapter 5. The difference between the FCR model in this step and the complete MPSNA model is that all variables and constraints related to restoration are removed since the FCR problem doesn't consider the network survivability. In addition, because we only consider one time period in this step, the time-related subscript, *t*, is eliminated.

The formulation is expressed as follows:

$$\text{Minimize} \qquad \sum_{s \in S} (F^s + C^s) \tag{6.1}$$

Subject to:

$$F^s = \lambda^s \cdot f^s \qquad \forall s \in S \qquad (6.2)$$

$$C^s = \sum_{m \in M} (c^{m,s} \cdot n^{m,s}) \qquad \forall s \in S \qquad (6.3)$$

$$w^s \leq \lambda^s \cdot M_1 \qquad \forall s \in S \qquad (6.4)$$

$$w^s \leq \sum_{m \in M} (n^{m,s} \cdot z^m) \qquad \forall s \in S \qquad (6.5)$$

$$w^s = \sum_{r \in D} w_r^s \cdot d^r \qquad \forall s \in S \qquad (6.6)$$

$$\sum_{j \in S_n} w_r^j = 1 \qquad \forall r \in D \quad \forall n \in O_r \qquad (6.7)$$

$$\sum_{j \in S_n} w_r^j = 1 \qquad \forall r \in D \quad \forall n \in D_r \qquad (6.8)$$

$$\sum_{j \in S_n} w_r^j = 2w_r^n \qquad \forall r \in D \quad \forall n \in N \mid n \notin \{O_r, D_r\} \qquad (6.9)$$

The objective function (6.1) seeks to minimize the sum of the span installation and capacity costs. Constraints in equation (6.2) calculate the span installation cost. If λ^s=1, there will be a span installation cost for span *s*. If λ^s=0, the span installation cost for span *s* will be zero. Constraints in equation (6.3) calculate the capacity cost. Constraints in equation (6.4) force the topological variable, λ^s, to be 1 as long as there is any working capacity assigned to span *s*. Equation (6.5) assigns a sufficient number of modules of each size on span *s* to accommodate the working capacity. Constraints in equation (6.6) through (6.9) describe the strategy of routing working capacity. They are similar to equations (5.10), (5.13), (5.14) and (5.15) formulated in Chapter 5. But the key difference is that we only consider one time period in this step. Equation (6.6) calculates the working capacity assigned to span *s*. Constraints in equations (6.7) and (6.8) ensure that there is only one span selected to carry the working flow for the origin node of O-D pair *r* and the destination node of O-D pair *r*, respectively. Equation (6.9) ensures that there are two spans connecting to the working nodes (excludes the origin node and destination node of O-D pair *r*).

6.2.2 Step 2: Augment Good Spans for Spare Capacity

Step 2 aims to augment the network by adding spans as needed for routing backup paths, thereby creating a bi-connected (i.e., restorable) topology space. In step 2, we will solve a topology design with sparing capacity allocation problem which is also called reserve network fixed charge and sparing (RN-FCS) problem [3]. Only spans inherited from step 1 are considered for failure scenarios in this step, and the working paths are as routed in step 1. All candidate spans in the network are considered as restoration spans. In addition, we also need to know the span installation cost of the spans inherited from step 1 is zero in step 2.

The ILP model for step 2 also uses the same notation defined in Chapter 5. The time-related subscript, *t*, is eliminated here as well because only a single time period is considered. w_r^s and w_r^n become parameters in this step (with values obtained from step 1). In addition, one new set is used here:

$S^w \subseteq S$ is the set of spans in network *G* for which λ^s=1 in the solution of step 1. Spans in set S^w are the working spans identified in step 1 and only these spans are considered for failure scenarios in step 2.

The formulation is expressed as follows:

Minimize $$\sum_{s\in S|s\notin S^w} F^s + \sum_{s\in S} C^s \quad (6.10)$$

Subject to:

$$F^s = \lambda^s \cdot f^s \qquad \forall s \in S \mid s \notin S^w \quad (6.11)$$

$$C^s = \sum_{m\in M} (c^{m,s} \cdot n^{m,s}) \qquad \forall s \in S \quad (6.12)$$

$$p^s \le \lambda^s \cdot M_1 \qquad \forall s \in S \mid s \notin S^w \quad (6.13)$$

$$p^s \le \sum_{m\in M} (n^{m,s} \cdot z^m) \qquad \forall s \in S \quad (6.14)$$

$$p^s \ge \sum_{r\in D} (x_r^{s,sf} \cdot d^r) \qquad \forall s \in S \quad \forall sf \in S^w \quad \forall s \ne \forall sf \quad (6.15)$$

$$x_r^{s,sf} \ge w_r^{sf} + p_r^s - 1 \qquad \forall r \in D \quad \forall s \in S \quad \forall sf \in S^w \quad \forall s \ne \forall sf \quad (6.16)$$

$$\sum_{j\in S_n} p_r^j = 1 \qquad \forall r \in D \quad \forall n \in O_r \quad (6.17)$$

$$\sum_{j \in S_n} p_r^j = 1 \qquad \forall r \in D \quad \forall n \in D_r \qquad (6.18)$$

$$\sum_{j \in S_n} p_r^j = 2 p_r^n \qquad \forall r \in D \quad \forall n \in N \mid n \notin \{O_r, D_r\} \qquad (6.19)$$

$$w_r^s + p_r^s \le 1 \qquad \forall r \in D \quad \forall s \in S \qquad (6.20)$$

$$w_r^n + p_r^n \le 1 \qquad \forall r \in D \quad \forall n \in N \mid n \notin \{O_r, D_r\} \qquad (6.21)$$

The objective function in (6.10) minimizes the sum of span installation cost (the cost of installing new spans) and capacity costs. Constraints in equation (6.11) to (6.14) have the similar function as constraints in equations (6.2) to (6.5). The only difference is that spare capacity is considered here, rather than working capacity. Constraints in equations (6.15) and (6.16) have the similar function as constraints in equations (5.11) and (5.12), but only a single time period is considered here and only spans generated from step 1 are iterated over for the failure scenarios. For example, equation (6.15) ensures that enough protection capacity is assigned to span s to accommodate the largest concurrent protection flows for span s in set S^w. Equation (6.16) calculates the concurrent spare flows on span s for restoring the working flow on failed span *sf*. Constraints in equation (6.17) to (6.21) have the similar function as constraints (5.16) to (5.20), but only a single time period is used.

6.2.3 Step 3: Refine the Spans Inherited from Step 1 and 2

In step 3, we refine the network topology generated from steps 1 and 2 via a one-period topological SBPP problem. In step 1 and step 2, we only consider the assignment of working and spare capacity, respectively. However, in step 3, we take them into account simultaneously. As a result, some spans may be eliminated and a more high-merit network topology can be produced. In addition, time-related subscript, *t*, is eliminated here as well, as we produce a single period design appropriate for demands from the final time period only.

The formulation is expressed as follows:

Minimize $$\sum_{s\in S}(F^s + C^s) \quad (6.22)$$

Subject to:

$$F^s = \lambda^s \cdot f^s \qquad \forall s \in S \quad (6.23)$$

$$C^s = \sum_{m\in M}(c^{m,s} \cdot n^{m,s}) \qquad \forall s \in S \quad (6.24)$$

$$a^s \le \lambda^s \cdot M_1 \qquad \forall s \in S \quad (6.25)$$

$$a^s \le \sum_{m\in M}(n^{m,s} \cdot z^m) \qquad \forall s \in S \quad (6.26)$$

$$a^s = w^s + p^s \qquad \forall s \in S \quad (6.27)$$

$$w^s = \sum_{r\in D} w_r^s \cdot d^r \qquad \forall s \in S \quad (6.28)$$

$$p^s \ge \sum_{r\in D}(x_r^{s,sf} \cdot d^r) \qquad \forall s \in S \quad \forall sf \in S \quad \forall s \ne \forall sf \quad (6.29)$$

$$x_r^{s,sf} \geq w_r^{sf} + p_r^s - 1 \qquad \forall r \in D \quad \forall s \in S \quad \forall sf \in S \quad \forall s \neq \forall sf \qquad (6.30)$$

$$\sum_{j \in S_n} w_r^j = 1 \qquad \forall r \in D \quad \forall n \in O_r \qquad (6.31)$$

$$\sum_{j \in S_n} w_r^j = 1 \qquad \forall r \in D \quad \forall n \in D_r \qquad (6.32)$$

$$\sum_{j \in S_n} w_r^j = 2w_r^n \qquad \forall r \in D \quad \forall n \in N \mid n \notin \{O_r, D_r\} \qquad (6.33)$$

$$\sum_{j \in S_n} p_r^j = 1 \qquad \forall r \in D \quad \forall n \in O_r \qquad (6.34)$$

$$\sum_{j \in S_n} p_r^j = 1 \qquad \forall r \in D \quad \forall n \in D_r \qquad (6.35)$$

$$\sum_{j \in S_n} p_r^j = 2p_r^n \qquad \forall r \in D \quad \forall n \in N \mid n \notin \{O_r, D_r\} \qquad (6.36)$$

$$w_r^s + p_r^s \leq 1 \qquad \forall r \in D \quad \forall s \in S \qquad (6.37)$$

$$w_r^n + p_r^n \leq 1 \qquad \forall r \in D \quad \forall n \in N \mid n \notin \{O_r, D_r\} \qquad (6.38)$$

The model in step 3 is nearly identical to the full MPSNA model described by (6.1) through (6.20), but there are two key differences. The first difference is that time-related subscript, t, is eliminated, since only a single time period (the final one) is considered. The second difference is that the constraints in equation (5.6) through (5.8) have no counterparts here, since those constraints deal with interactions between subsequent time period (as noted, we only consider a single time period here).

6.2.4 Step 4: Resolve the MPSNA Problem with a Smaller Candidate Topology

In the final step, we solve the full MPSNA model described by (5.1) through (5.20) in its entirety. However, the difference is that the set of eligible spans, *S*, consists of only those spans for which λ^s=1 after step 3 is solved (i.e., this is the set of high-merit spans). To be proper, we could define this set of high-merit spans with some new notation, say *S′*, and express the model for step 4 using equations (5.1) through (5.20) except we replace all instances of *S* with *S′*. Since the number of candidate spans (*S′*) here are less than the number of spans in original network *G*, the solution runtime of a restricted version of the MPSNA problem is expected to be reduced dramatically.

The ILP model in step 4 is totally same as the MPSNA model formulated in chapter 5, except that we use a smaller topology space as a candidate topology rather than the original network *G*.

In conclusion, the four-stage ILP-based heuristic approach can be described briefly as follows.

- Step 1: Identify good spans for working light path routing alone based on the demands in the final time period in isolation.

- Step2: Augment good spans for spare capacity based on the demands in the final time period in isolation.

- Step 3: Refine the spans inherited from step 1 and step 2 based on the demands in the final time period in isolation.

- Step 4: Resolve a restricted version of the MPSNA problem using only the subset of eligible spans inherited from step 3, rather than all spans.

6.3 Experimental Study Method

In order to verify the efficiency of the four-stage ILP-based heuristic approach, we validate this approach through simulations which are conducted on the same six networks presented in Chapter 5.

All ILP formulations were all implemented in AMPL, and were solved by Gurobi Optimizer 6.0.3 on a Mac Pro server with a 3.5 GHZ processor and a 32 GB RAM. Since an arc-flow approach is used to formulate the ILP model in this work, pre-processing for eligible working and protection routes was not required. However, some results in step 1 would be forwarded to step 2 as the input parameters, and data files of step 3 and step 4 also need to be processed because of the reduction of the candidate edges. As a result, pre-processing of data was completed on a Lenovo laptop with a 2.4GHZ dual core processor and a 6 GB RAM. All of the various steps in the heuristic are run to completion with various optimality gaps (as specified in the results and discussion). In order to accurately record the solution runtime of each experimental test, there is no other task running when the Gurobi is running (i.e., the computer is devoted to the Gurobi task full time [3]), and the Gurobi runs only one test each time.

6.4 Results and Discussion

The four-stage ILP-based heuristic solution characteristics are illustrated in Table 5, which summarizes the experimental networks, number of spans selected or eliminated in each step and the associated solution runtimes and optimality gaps.

Due to the complexity of the ILP problem in each step, we used optimality gap settings ranging from 1% to 25% in step 1, 1% in step 2, 1% to 15% in step 3, and 1% in step 4 (a setting of 1% means the solution will terminate when it is guaranteed to be within 1% of optimal). Larger optimality gaps are acceptable in steps 1 through 3 because those steps are merely selecting for high-merit spans, and so a strictly optimal solution isn't required in any of those individual steps (and experience has shown us that optimality of the overall solution is not largely affected). The smallest network, 8-node network, can be solved in seconds (or less), while the 30-node network took approximately 3.3 hours of solution time. Steps 1 and 3 appear to be the most computationally challenging for test case networks. The reason for that is the fixed charge plus routing (FCR) problem in step 1 and the one-period topological SBPP (OPT-SBPP) problem in step 3 are both NP-hard problem which are known to be difficult to solve. However, these two problems have been widely studied for many years. Prior work can be applied to them to improve the quality of solution and runtime as needed. In contrast to step 1 and step 3, the step 2 and step 4 can be solved very quickly. For instance, in most test cases, these two steps can be solved in few seconds to a 1% optimality gap. The reason of step 2 being easily solved is that only number of $|S^w|$ spans is considered for failure scenarios in this step instead of number of $|S|$ spans. There are two reasons for explaining why the step 4 can be easily solved. The first reason is that we resolve the MPSNA problem in step 4 by using only a subset of high-merit spans rather than all candidate spans in the original networks. As we can see from the solution of step 4, the number of high-merit candidate spans is much less than the number of candidate spans in the original networks. For example, in test case of 8n16s1 network, the number of candidate spans in step 4 is 9 which is much less than 16 (the number of candidate spans in the original network). The other reason is that the average nodal degree of the candidate topologies in step 4 is around d=2.2 which is much smaller than the nodal degree of the original networks (d=4). It is obvious that the problem with a small nodal degree can be easily solved.

For all test case networks, the topology (i.e., subset of spans) selected in step 1 was augmented (by 2 to 12 spans) in step 2. That topology was subsequently reduced (by 2 to 6 spans) in four of the six networks in step 3, while the topology remained the same in two of the networks. Finally, the topologies in all six test case networks remained unchanged in step 4 relative to the topologies of high-merit spans selected by step 3, though the constituent working and backup routing and associated working and spare capacities were quite different.

Table 5: ILP-based heuristic approach solutions of all six experimental networks.

Network	**Step 1**			**Step 2**		
	No. of spans	**Runtime (Seconds)**	**Optimality gap**	**No. of new spans**	**Runtime (Seconds)**	**Optimality gap**
8n16s1	7	0.19	1%	2	0.02	1%
12n24s1	11	6.36	1%	4	0.17	1%
15n30s1	14	200	1%	3	0.42	1%
20n40s1	19	833	15%	8	3.63	1%
25n50s1	24	590	20%	8	48.74	1%
30n60s1	30	1136	25%	12	4201	1%
Network	**Step 3**			**Step 4**		
	No. of eliminated spans	**Runtime (Seconds)**	**Optimality gap**	**No. of spans**	**Runtime (Seconds)**	**Optimality**
8n16s1	0	0.03	1%	9	0.15	1%
12n24s1	2	0.27	1%	13	0.46	1%
15n30s1	0	0.49	1%	17	1.95	1%
20n40s1	5	1915	1%	22	5.77	1%
25n50s1	4	2068	12%	28	421	1%
30n60s1	6	5148	15%	36	1266	1%

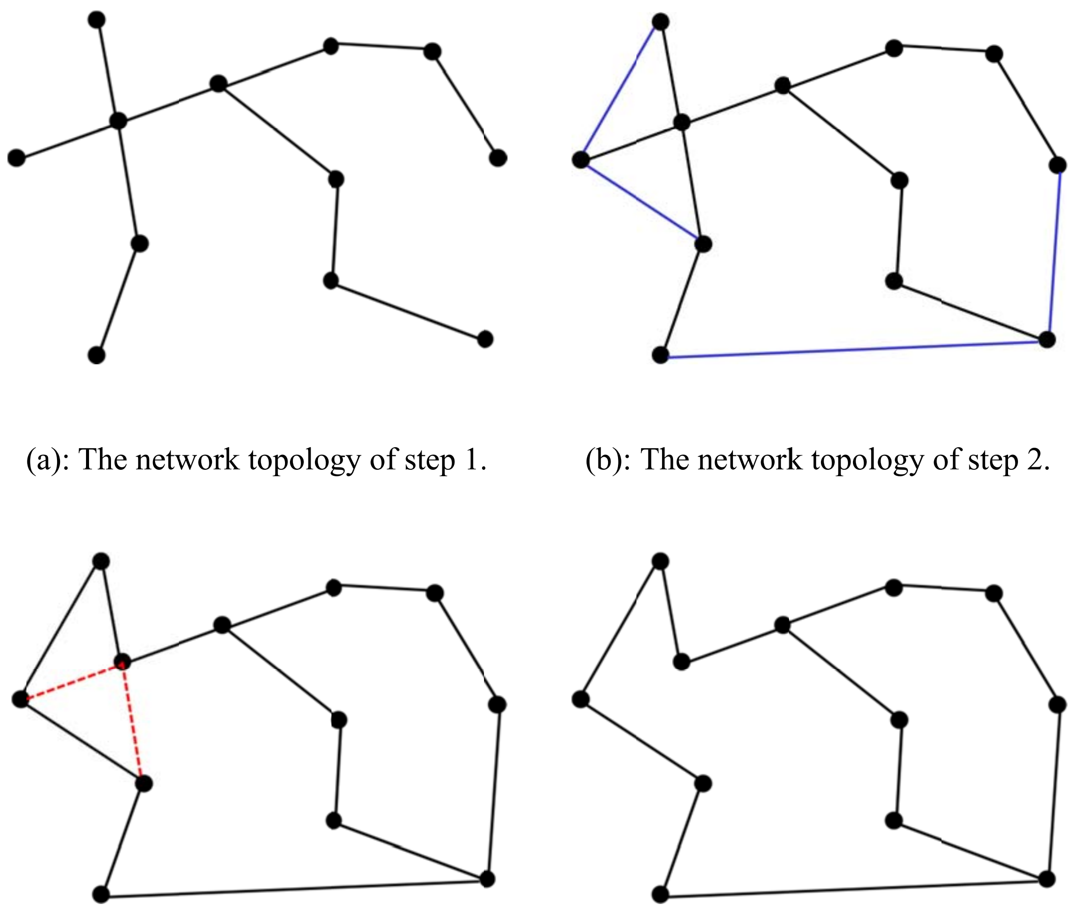
Table 6 summarizes the total costs (i.e., objective function values) and solution times for all of the full MPSNA solutions (benchmark) and heuristic solutions. Meanwhile, the percentage improvement of cost saving is calculated. In the smallest test case, where the benchmark solution can be solved to completion in less than 40 minutes with a 1% optimality gap, the heuristic solution was achievable in just four seconds with a 6.65%

loss in optimality. Results for the 12-node network were similar (i.e., substantial reduction in solution time with a modest loss in optimality). For those small network test cases, the users can choose the approach according to their needs. If they focus on best solutions, they can choose to solve the MPSNA problem directly; if they pay more attention to the solution runtime and can accept a good solution but not best one, they can use the four-stage ILP-based heuristic approach. For middle and large networks, the four-stage ILP-based heuristic has a good performance in terms of both solution runtimes and solution quality. In the 15-node and 20-node networks, the heuristic solutions showed solution quality improvements of over 11% in only a small fraction of the time required for the full MPSNA solutions. For the two largest test cases networks, where the full benchmark solutions were not obtainable at all, the heuristic solution times were 51 minutes and 3.3 hours, respectively. Through above discussions, we can summary that the four-stage ILP-based heuristic approach can provide very high-quality solutions in less time than benchmark solutions and the heuristic approach is very effective to solve the MPSNA problems.

Table 6: % improvement of ILP-based heuristic solutions.

Network	Benchmark		ILP-based heuristic		
	Cost	**Runtime**	**Cost**	**Runtime**	**Improvement (%)**
8n16s1	1,936,223	0.62hrs	2,064,919	0.07min	-6.65
12n24s1	1,069,105	24hrs	1,151,002	0.12min	-7.66
15n30s1	3,452,152	36hrs	3,051,168	3.4min	11.62
20n40s1	3,479,002	48hrs	3,073,526	46min	11.65
25n50s1	N/A	3weeks	3,826,661	51min	---
30n60s1	N/A	>3weeks	5,070,193	196min	---

To have a better understanding about the four-stage ILP-based heuristic solutions, we take 12-node experimental network as an example to show its network topology in each step. The evolution of the 12-node network through the four steps of the heuristic are illustrated in Figure 12 through Figure 15 shows the resultant topologies arising in each step. As we can see, the network topology of step 1 is a tree consisting of 11 spans, as we expected since we only consider the working routing here. In step 2, four new spans (shown in blue) are added to create a bi-connected topology and permit backup routing. As we expected these new spans play a role in either bearing the spare capacity or closing a graph (i.e., creating a bi-connected graph), or both two aspects. In step 3, two of the spans from the previous step (shown in red) are eliminated, since the purpose of this step is to create a much more high-merit network topology when compared to the topology inherited from step 2. As described above, the topology from step 3 remains unchanged after solution of step 4.



(a): The network topology of step 1. (b): The network topology of step 2.

(c): The network topology of step 3. (d): The network topology for step 4.

Figure 12: 12-node network topologies produced in step 1, 2 and 3, and the candidate network topology for step 4.

Figure 13 through Figure 15 illustrate the network evolution of network capacity in the 12-node network in each time period of step of the heuristic. Obviously, the capacity will increase quite substantially from one time period to the next, but interestingly the full topology it utilized from the first time period, thereby requiring no topology augmentation over time. The reason for that is the candidate spans in step 4 are very high-merit, it is reasonable to see candidate spans are all used in the first time period and there is no spans' augmenting in the following time periods.

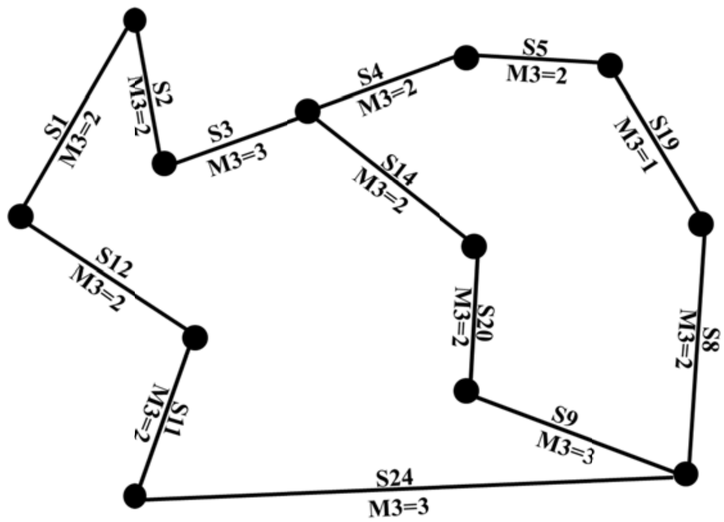


Figure 13: The 12-node network topology with modular capacity at the first period in step 4 of the ILP-based heuristic.

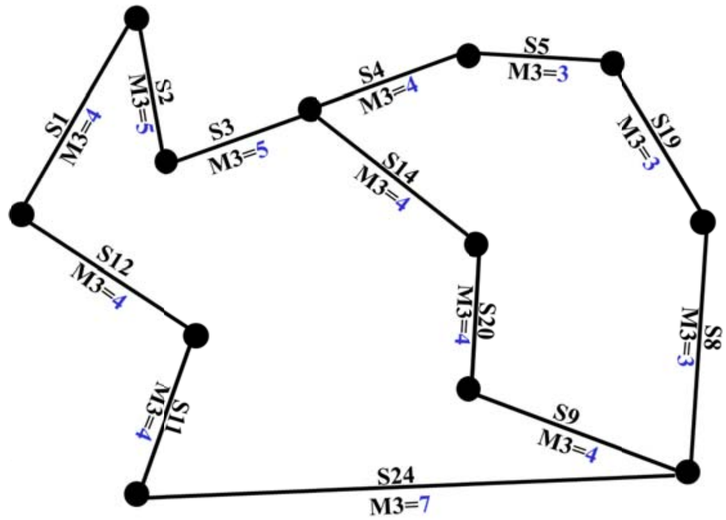


Figure 14: The 12-node network topology with modular capacity at the second period in step 4 of the ILP-based heuristic.

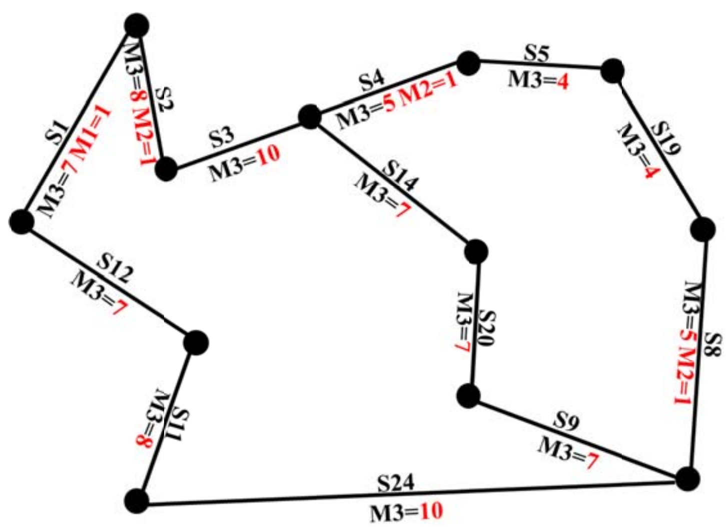


Figure 15: The 12-node network topology with modular capacity at the third period in step 4 of the ILP-based heuristic.

Chapter 7: Dantzig-Wolfe Decomposition Algorithm

7.1 Motivation of the Implementation of the Dantzig-Wolfe Decomposition

Let's recall step 4 of the four-stage ILP-based heuristic approach. In step 4, we solve a restricted version of the MPSNA problem using only a subset of the eligible spans (high-merit spans generated from the previous steps) rather than full spans. The ILP model in step 4 is totally the same as the MPSNA model formulated in Chapter 5. The only difference is that we use a smaller topology space in step 4 as a candidate topology rather than the original network topology. As we expected, due to its reduction in the number of candidate spans, the restricted version of the MPSNA problem can be solved very quickly. For example, in test cases of networks 8n16s1 through 20n40s1, the restricted version of the MPSNA problems can all be solved in 6 seconds within 1% optimality gap. However, solution runtimes of the restricted version of the MPSNA problems are increased as networks become larger. For instance, in test case of 25n50s1 network, the solution runtime in step 4 goes up to 420.78 seconds. Even though 420.78 seconds is not long when compared to solution runtimes in previous steps (e.g., the solution runtime of the FCR problem in step 1 is 590 seconds and the optimality gap is 20%), it is still better to find a method to decrease the solution runtime of the restricted version of the MPSNA problem.

In this chapter, we will investigate the effectiveness of the implementation of the Dantzig-Wolfe decomposition for solving the restricted version of the MPSNA problem since we suspect the MPSNA problem has a block angular structure. We also want to remind the readers that the purpose of Chapter 7 is to decrease the solution runtime of step 4; so we will implement the Dantzig-Wolfe decomposition algorithm to solve the MPSNA problem using only subset of eligible spans (spans inherited from step 3) rather than full spans. But if we want to use Dantzig-Wolfe decomposition to solve the full MPSNA problem, we just need to change the data file (that is, use full candidate spans instead of subset of eligible spans).

7.2 Implement the Dantzig-Wolfe Decomposition to Solve ILP Problems

If a liner programming (LP) problem has a block angular structure, this problem can be solved easily by Dantzig-Wolfe decomposition. Likewise, the Dantzig-Wolfe decomposition algorithm can also be used to solve the ILP problems with a block angular structure.

Because we have detailed the Dantzig-Wolfe decomposition algorithm in Chapter 2, here we briefly summarize how to implement the Dantzig-Wolfe decomposition algorithm to solve LP and ILP problems. To solve LP problems:

- First, we need to reformulate the original LP problem based on the convex combinations theorem [27].

- Second, we need to obtain an LP master problem totally equivalent to the original LP problem.

- Then, we can use a column generation algorithm to solve the LP master problem since it may contain a large number of columns. Finally, an optimal solution can be obtained when the master problem cannot be further improved by any new columns from the sub-problems.

For LP problems, we can get the same solutions as the original LP problems after the implementation of the Dantzig-Wolfe decomposition algorithm. However, for ILP problems, the algorithm can only get us a tight lower bound for a minimum problem or a tight upper bound for a maximum problem.

The procedure of implementing the Dantzig-Wolfe decomposition for solving ILP problems is similar to LP problems'. However, there still are some differences. The procedure and differences are described as follows:

- First, reformulate the original ILP problem. Since variables in the original ILP problem are restricted to integer, the new variables in the master problem are restricted to binary (either 0 or 1).

- Second, obtain an ILP master problem which is totally equivalent to the original ILP problem after the reformulation.

- Then, relax the integrality of the ILP master problem and solve the LP relaxation of the ILP master problem (LP master problem) using column generation algorithm.

- In addition, we can get a tight lower bound for a minimum ILP problem or a tight upper bound for a maximum ILP problem after solving the LP master problem since it is well known that the LP master is the dual formulation of the Lagrangian dual [30][31].

- Finally, add this tight bound to the original ILP problem and resolve it.

Since the MPSNA problem is a minimum problem, a tight lower bound can be obtained after solving the LP master of the MPSNA problem. We expect that the solution runtime of the MPSNA problem is decreased after adding this lower bound back to it. Figure 16 illustrates the detailed procedure of how to apply the Dantzig-Wolfe decomposition to solve the MPSNA problem.

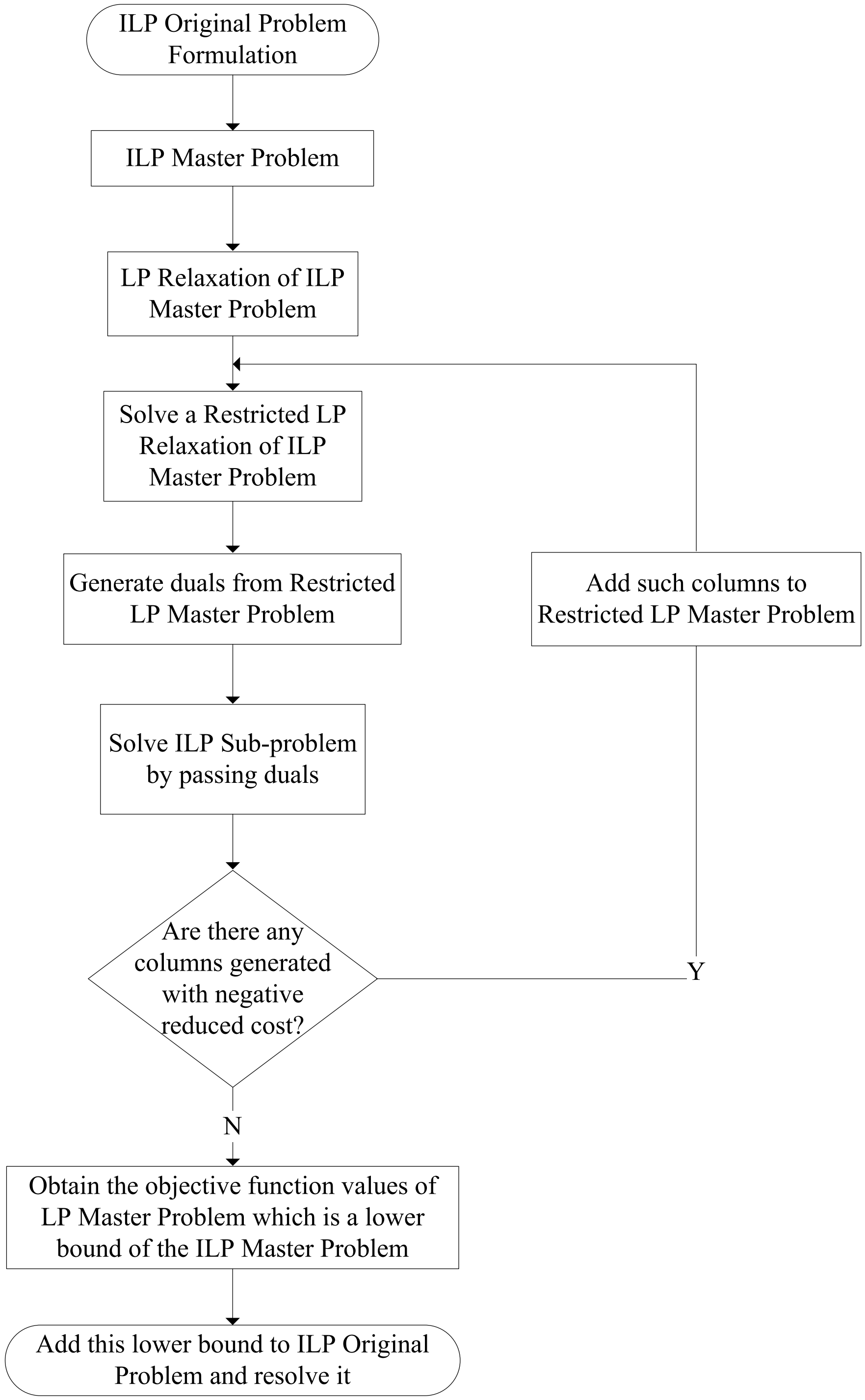


Figure 16: The flow chart of applying the Dantzig-Wolfe decomposition algorithm to solve the ILP MPSNA problem.

7.2 Block Angular Structure of the MPSNA Model

Through merging similar items and simplifying constraints of the MPSNA model formulated in Chapter 5, we find that the MPSNA model has a block angular structure. The simplified MPSNA model is expressed as follows.

Simplified ILP model of the MPSNA problem:

Minimize $$\sum_{s \in S} \sum_{t \in T} ((\lambda_t^s - \lambda_{t-1}^s) \cdot f_t^s + \sum_{m \in M} (c_t^{m,s} \cdot n_t^{m,s}) - \sum_{m \in M} (c_{t-1}^{m,s} \cdot n_{t-1}^{m,s})) \quad (7.1)$$

Subject to

$$a_t^s - \lambda_t^s \cdot M_1 \le 0 \qquad \forall s \in S \quad \forall t \in T \quad (7.2)$$

$$a_t^s - \sum_{m \in M} (n_t^{m,s} \cdot z^m) \le 0 \qquad \forall s \in S \quad \forall t \in T \quad (7.3)$$

$$-a_t^s + \sum_{r \in D} d_t^r (w_{r,t}^s + x_{r,t}^{s,sf}) \le 0 \qquad \forall s \in S \quad \forall sf \in S \quad \forall s \ne \forall sf \quad \forall t \in T \quad (7.4)$$

$$w_{r,t}^{sf} + p_{r,t}^s - x_{r,t}^{s,sf} \le 1 \qquad \forall r \in D \quad \forall s \in S \quad \forall sf \in S \quad \forall s \ne \forall sf \quad \forall t \in T \quad (7.5)$$

$$\lambda_t^s - \lambda_{t-1}^s \ge 0 \qquad \forall s \in S \quad \forall t \in T \quad (7.6)$$

$$n_t^{m,s} - n_{t-1}^{m,s} \ge 0 \qquad \forall s \in S \quad \forall t \in T \quad (7.7)$$

$$a_t^s - a_{t-1}^s \ge 0 \qquad \forall s \in S \quad \forall t \in T \quad (7.8)$$

$$\sum_{j \in S_n} w_{r,t}^{j} = 1 \qquad \forall r \in D \quad \forall n \in O_r \quad \forall t \in T \qquad (7.9)$$

$$\sum_{j \in S_n} w_{r,t}^{j} = 1 \qquad \forall r \in D \quad \forall n \in D_r \quad \forall t \in T \qquad (7.10)$$

$$\sum_{j \in S_n} p_{r,t}^{j} = 1 \qquad \forall r \in D \quad \forall n \in O_r \quad \forall t \in T \qquad (7.11)$$

$$\sum_{j \in S_n} p_{r,t}^{j} = 1 \qquad \forall r \in D \quad \forall n \in D_r \quad \forall t \in T \qquad (7.12)$$

$$w_{r,t}^{s} + p_{r,t}^{s} \le 1 \qquad \forall r \in D \quad \forall s \in S \quad \forall t \in T \qquad (7.13)$$

$$\sum_{j \in S_n} w_{r,t}^{j} + \sum_{j \in S_n} p_{r,t}^{j} \le 2 \qquad \forall r \in D \quad \forall n \in N \mid n \notin \{O_r, D_r\} \quad \forall t \in T \qquad (7.14)$$

The detailed information about how to simplify the MPSNA model and how to organize the constraints to present a block angular structure is below.

Recall the following constraints formulated in Chapter 5.

$$\sum_{s \in S} \sum_{t \in T} (F_t^s + C_t^s) \qquad (5.1)$$

$$F_t^s = (\lambda_t^s - \lambda_{t-1}^s) \cdot f_t^s \qquad \forall s \in S \quad \forall t \in T \qquad (5.7)$$

$$C_t^s = \sum_{m \in M} (c_t^{m,s} \cdot n_t^{m,s}) - \sum_{m \in M} (c_{t-1}^{m,s} \cdot n_{t-1}^{m,s}) \qquad \forall s \in S \quad \forall t \in T \qquad (5.8)$$

If we insert constraints (5.7) and (5.8) to constraint (5.1), the objective function (7.1) can be obtained. Likewise, inserting constraints (5.10) and (5.11) to constraint (5.9) can produce constraint (7.4).

In addition, through reorganizing the constraints (5.15), (5.18), and constraint (5.20), the constraint (7.14) can be produced. The detailed procedure is described as follows.

The constraint (5.15) ($\sum_{j \in S_n} w_{r,t}^j = 2w_{r,t}^n \quad \forall r \in D \;\; \forall n \in N \mid n \notin \{O_r, D_r\} \;\; \forall t \in T$) can be expressed in another form ($w_{r,t}^n = \frac{1}{2} \sum_{j \in S_n} w_{r,t}^j \;\; \forall r \in D \;\; \forall n \in N \mid n \notin \{O_r, D_r\} \;\; \forall t \in T$). Likewise, the constraint (5.18) can be expressed as $p_{r,t}^j = \frac{1}{2} \sum_{j \in S_n} p_{r,t}^j \quad \forall r \in D$ $\forall n \in N \mid n \notin \{O_r, D_r\} \quad \forall t \in T$. If we insert these two new constraints to constraint (4.20) ($w_{r,t}^n + p_{r,t}^n \le 1 \quad \forall r \in D \;\; \forall n \in N \mid n \notin \{O_r, D_r\} \;\; \forall t \in T$), then the constraint (7.14) is produced.

To make the MPSNA formulation clear, we put all variables in the left side of the constraints and put all constant numbers on the right side of the constraints. These reorganizations generate the simplified MPSNA model expressed in constraints (7.1) through (7.14).

If we take a closer look at the structure of the simplified MPSNA model, we find that the simplified MPSNA model has a block angular structure. In this model, constraints (7.2) through (7.5) can be considered as coupling constraints or hard constraints because they contain most of variables of the model and these variables have non-zero coefficients. The remaining constraints (constraints (7.6) through (7.14)) are considered as easy constraints and can be grouped into four independent blocks (sub-problems) since these constraints satisfy the following property: if a variable has non-zero coefficient in one block, it won't have non-zero coefficient in other blocks. For example, variable λ_t^s has non-zero coefficient in constraint (7.6), but it doesn't have non-zero coefficient in constraints (7.7) through (7.14). In addition, variables $w_{r,t}^s$ and $p_{r,t}^s$ have non-zero coefficients in constraints (7.9) through (7.14), but they don't have non-zero coefficients in constraints (7.6) through (7.8). The block angular structure of the MPSNA problem is visualized in Figure 17.

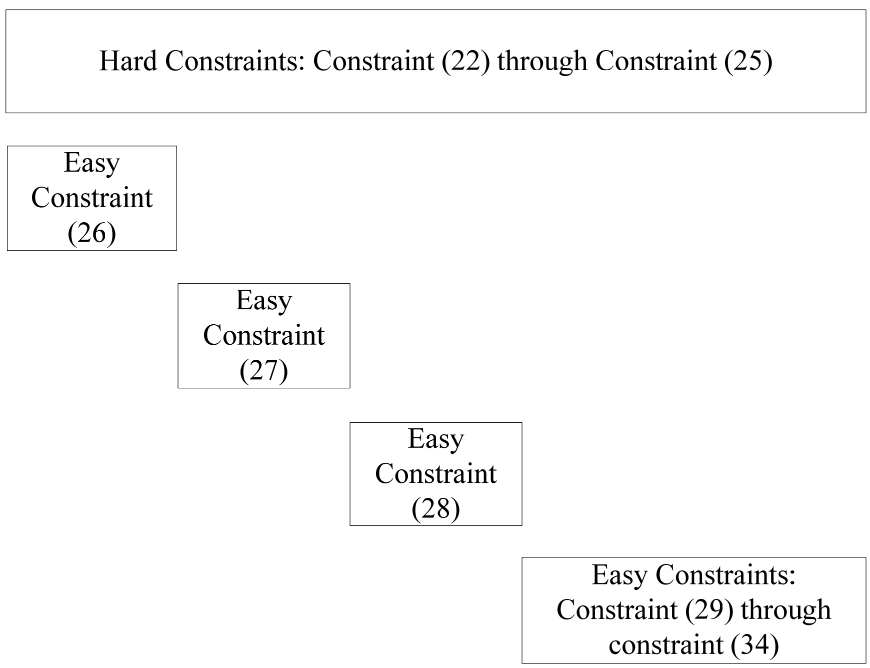


Figure 17: The block angular structure of the MPSNA problem.

Since the MPSNA model has the block angular structure, the LP relaxation of the ILP MPSNA master problem should be solved quickly by Dantzig-Wolfe decomposition and a tight lower bound can be obtained. In the next section, we will describe how to reformulate the MPSNA model by implementing Dantzig-Wolfe decomposition.

7.3 Reformulation of the MPSNA Model Based on the Dantzig-Wolfe Decomposition Algorithm

Through observing the constraints and bounds of variables in these four blocks, we find that the four convex polyhedrons are all bounded. Based on the ILP version of the convex combination theorem [83], the variables (λ_t^s , $n_t^{m,s}$, a_t^s , $w_{r,t}^s$ and $p_{r,t}^s$) in the MPSNA model can be represented by the combination of extreme points of finite integer sets, as follows:

$$\lambda_t^s = \sum_{k \in K} \theta_k \lambda_{t,k}^s, \ \sum_{k \in K} \theta_k = 1 \ \text{ and } \ \theta_k \in \{0,1\} \tag{7.15}$$

$$n_t^{m,s} = \sum_{e \in E} \alpha_e n_{t,e}^{m,s}, \ \sum_{e \in E} \alpha_e = 1 \ \text{ and } \ \alpha_e \in \{0,1\} \tag{7.16}$$

$$a_t^s = \sum_{g \in G} \varphi_g a_{t,g}^s, \ \sum_{g \in G} \varphi_g = 1 \ \text{ and } \ \varphi_g \in \{0,1\} \tag{7.17}$$

$$w_{r,t}^s = \sum_h \sigma_h w_{t,h}^{r,s}, \ p_{r,t}^s = \sum_h \sigma_h p_{t,h}^{r,s}, \ \sum_h \sigma_h = 1 \ \text{ and } \ \sigma_h \in \{0,1\} \tag{7.18}$$

If we replace λ_t^s , $n_t^{m,s}$, a_t^s , $w_{r,t}^s$ and $p_{r,t}^s$ with these new expressions, the ILP master model will be produced. Here, we need to mention that $\lambda_{t,k}^s$, $n_{t,e}^{m,s}$, $a_{t,g}^s$, $w_{t,h}^{r,s}$ and $p_{t,h}^{r,s}$ are extreme points of finite integer sets and become parameters in the ILP master problem, and θ_k , α_e , φ_g and σ_h become new variables in the ILP master model. The new notations and formulation of the ILP master problem are described in Section 7.3.1 and 7.3.2, respectively.

7.3.1 Notations

New Sets:

- K is the set of numbers. These numbers represent orders of extreme points of a finite integer set bounded by constraint (7.6) and its variable bound, the number interval is 1 and the biggest number in this set represents the total number of extreme points.

- E is the set of numbers. These numbers represent orders of extreme points of a finite integer set bounded by constraint (7.7) and its variable bound, the number interval is 1 and the biggest number in this set represents the total number of extreme points.

- G is the set of numbers. These numbers represent orders of extreme points of a finite integer set bounded by constraint (7.8) and its variable bound, the number interval is 1 and the biggest number in this set represents the total number of extreme points.

- H is the set of numbers. These numbers represent orders of extreme points of a finite integer set bounded by constraints (7.9) through (7.14) and bounds of their variables, the number interval is 1 and the biggest number in this set represents the total number of extreme points.

- K_1 is a subset of set K (i.e., $K_1 \subseteq K$).

- E_1 is a subset of set E (i.e., $E_1 \subseteq E$).

- G_1 is a subset of set G (i.e., $G_1 \subseteq G$).

- H_1 is a subset of set H (i.e., $H_1 \subseteq H$).

New Input Parameters:

- β_t^s is the shadow price of constraint (7.20).

- δ_t^s is the shadow price of constraint (7.21).

- $\varepsilon_t^{s,sf}$ is the shadow price of constraint (7.22).

- $\iota_{r,t}^{s,sf}$ is the shadow price of constraint (7.23).

- μ is the shadow price of constraint (7.24).

- υ is the shadow price of the constraint (7.25).

- π is the shadow price of constraint (7.26).

- ρ is the shadow price of constraint (7.27).

- $\lambda_{t,k}^{s}$ is the extreme point of a finite integer set bounded by constraint (7.6) and its variable bound.

- $n_{t,e}^{m,s}$ is the extreme point of a finite integer set bounded by constraint (7.7) and its variable bound.

- $a_{t,g}^{s}$ is the extreme point of a finite integer set bounded by constraint (7.8) and its variable bound.

- $w_{t,h}^{r,s}$ and $p_{t,h}^{r,s}$ are extreme points of a finite integer set bounded by constraints (7.9) through (7.14) and their variables' bounds.

New Decision Variables:

- $\theta_k \in \{0,1\}$ is the weight of the parameter $\lambda_{t,k}^{s}$.

- $\alpha_e \in \{0,1\}$ is the weight of the parameter $n_{t,e}^{m,s}$.

- $\varphi_g \in \{0,1\}$ is the weight of the parameter $a_{t,g}^{s}$.

- $\sigma_h \in \{0,1\}$ is the weight of parameters $w_{t,h}^{r,s}$ and $p_{t,h}^{r,s}$.

7.3.2 ILP Master Problem

The ILP master problem is totally equivalent to the ILP original problem. The formulation of the ILP master problem is expressed as follows:

Minimize

$$\sum_{k\in K}[\sum_{\substack{s\in S\\ t\in T}}(\lambda_{t,k}^{s}-\lambda_{t-1,k}^{s})\cdot f_{t}^{s}]\cdot\theta_{k}+\sum_{e\in E}[\sum_{\substack{s\in S\\ t\in T\\ m\in M}}(c_{t}^{m,s}\cdot n_{t,e}^{m,s})-\sum_{\substack{s\in S\\ t\in T\\ m\in M}}(c_{t-1}^{m,s}\cdot n_{t-1,e}^{m,s})]\cdot\alpha_{e} \tag{7.19}$$

Subject to:

$$\sum_{g\in G}(a_{t,g}^{s})\cdot\varphi_{g}-\sum_{k\in K}(\lambda_{t,k}^{s}\cdot M_{1})\cdot\theta_{k}\leq 0 \qquad \forall s\in S \quad \forall t\in T \tag{7.20}$$

$$\sum_{g\in G}(a_{t,g}^{s})\cdot\varphi_{g}-\sum_{e\in E}(\sum_{m\in M}n_{t,e}^{m,s}\cdot z^{m})\cdot\alpha_{e}\leq 0 \qquad \forall s\in S \quad \forall t\in T \tag{7.21}$$

$$-\sum_{g\in G}(a_{t,g}^{s})\cdot\varphi_{g}+\sum_{h\in H}(\sum_{r\in D}d_{t}^{r}\cdot w_{t,h}^{r,s})\cdot\sigma_{h}+\sum_{r\in D}d_{t}^{r}\cdot x_{t}^{r,s,sf}\leq 0$$

$$\forall s\in S \quad \forall sf\in S \quad \forall s\neq\forall sf \quad \forall t\in T \tag{7.22}$$

$$\sum_{h\in H}(w_{t,h}^{r,s}+p_{t,h}^{r,s})\cdot\sigma_{h}-x_{t}^{r,s,sf}\leq 1 \qquad \forall r\in D \quad \forall s\in S \quad \forall sf\in S \quad \forall s\neq\forall sf \quad \forall t\in T \tag{7.23}$$

$$\sum_{k\in K}\theta_{k}=1 \tag{7.24}$$

$$\sum_{e\in E}\alpha_{e}=1 \tag{7.25}$$

$$\sum_{g\in G}\varphi_{g}=1 \tag{7.26}$$

$$\sum_{h\in H}\sigma_{h}=1 \tag{7.27}$$

So far, we have obtained the ILP master problem. The next step is to use column generation to solve this ILP master problem. First, we need to relax the integrality of the

ILP master problem. Second, we start to solve the LP master problem with some small set of variables (the restricted LP master problem). Then, we solve the sub-problems to decide which other variables to access to master problem as needed. Finally, we repeat this process until we obtain the optimal solution (the lower bound for the MPSNA problem).

7.3.3 Restricted LP Master Problem and Sub-problems

7.3.3.1 Restricted LP Master Problem

The formulation of the restricted LP master problem is expressed as follows:

Minimize

$$\sum_{k \in K_1} [\sum_{\substack{s \in S \\ t \in T}} (\lambda_{t,k}^{s} - \lambda_{t-1,k}^{s}) \cdot f_t^{s}] \cdot \theta_k + \sum_{e \in E_1} [\sum_{\substack{s \in S \\ t \in T \\ m \in M}} (c_t^{m,s} \cdot n_{t,e}^{m,s}) - \sum_{\substack{s \in S \\ t \in T \\ m \in M}} (c_{t-1}^{m,s} \cdot n_{t-1,e}^{m,s})] \cdot \alpha_e \tag{7.28}$$

Subject to:

$$\sum_{g \in G_1} (a_{t,g}^{s}) \cdot \varphi_g - \sum_{k \in K_1} (\lambda_{t,k}^{s} \cdot M_1) \cdot \theta_k \leq 0 \qquad \forall s \in S \quad \forall t \in T \tag{7.29}$$

$$\sum_{g \in G_1} (a_{t,g}^{s}) \cdot \varphi_g - \sum_{e \in E_1} (\sum_{m \in M} n_{t,e}^{m,s} \cdot z^{m}) \cdot \alpha_e \leq 0 \qquad \forall s \in S \quad \forall t \in T \tag{7.30}$$

$$-\sum_{g \in G_1} (a_{t,g}^{s}) \cdot \varphi_g + \sum_{h \in H_1} (\sum_{r \in D} d_t^{r} \cdot w_{t,h}^{r,s}) \cdot \sigma_h + \sum_{r \in D} d_t^{r} \cdot x_t^{r,s,sf} \leq 0$$

$$\forall s \in S \quad \forall sf \in S \quad \forall s \neq \forall sf \quad \forall t \in T \tag{7.31}$$

$$\sum_{h \in H_1} (w_{t,h}^{r,s} + p_{t,h}^{r,s}) \cdot \sigma_h - x_t^{r,s,sf} \leq 1 \quad \forall r \in D \quad \forall s \in S \quad \forall sf \in S \quad \forall s \neq \forall sf \quad \forall t \in T \tag{7.32}$$

$$\sum_{k \in K_1} \theta_k = 1 \tag{7.33}$$

$$\sum_{e \in E_1} \alpha_e = 1 \tag{7.34}$$

$$\sum_{g \in G_1} \varphi_g = 1 \tag{7.35}$$

$$\sum_{h \in H_1} \sigma_h = 1 \tag{7.36}$$

$$\theta_k, \alpha_e, \varphi_g, \sigma_h \geq 0 \tag{7.37}$$

7.3.3.2 ILP Sub-Problems:

Because the MPSNA problem has four blocks, four sub-problems can be formulated. In addition, the variables in these sub-problems all retain integrality. We expected the sub-problems can be solved quickly since these constraints only contain one or two types of variables and look very simple. The objective functions of sub-problems are obtained through pricing out [31] and the constraints are copied from the simplified MPSNA model. The formulations are expressed as follows:

Sub-problem 1:

$$\text{Minimize} \quad \sum_{\substack{s \in S \\ t \in T}} (\lambda_t^s - \lambda_{t-1}^s) \cdot f_t^s + \sum_{\substack{s \in S \\ t \in T}} \lambda_t^s \cdot M_1 \cdot \beta_t^s - \mu \tag{7.38}$$

Subject to: $\lambda_t^s - \lambda_{t-1}^s \geq 0 \qquad \forall s \in S \quad \forall t \in T$ (7.39)

$0 \leq \lambda_t^s \leq 1$ and integer $\qquad \forall s \in S \quad \forall t \in T$ (7.40)

Sub-problem 2:

Minimize $$\sum_{\substack{s\in S \\ t\in T \\ m\in M}} (\mathrm{c}_t^{m,s} \cdot \mathrm{n}_t^{m,s}) - \sum_{\substack{s\in S \\ t\in T \\ m\in M}} (\mathrm{c}_{t-1}^{m,s} \cdot \mathrm{n}_{t-1}^{m,s}) + \sum_{\substack{s\in S \\ t\in T \\ m\in M}} n_t^{m,s} \cdot z^m \cdot \delta_t^s - \upsilon \qquad (7.41)$$

Subject to: $n_t^{m,s} - n_{t-1}^{m,s} \geq 0 \qquad \forall s \in S \quad \forall t \in T$ (7.42)

$0 \leq n_t^{m,s} \leq M_1$ and integer $\qquad \forall s \in S \quad \forall t \in T$ (7.43)

Sub-problem 3:

Minimize $$-\sum_{\substack{s\in S \\ t\in T}} a_t^s \cdot \beta_t^s - \sum_{\substack{s\in S \\ t\in T}} a_t^s \cdot \delta_t^s + \sum_{\substack{s\in S \\ sf\in S \\ s\neq sf \\ t\in T}} a_t^s \cdot \varepsilon_t^{s,sf} - \pi \qquad (7.44)$$

Subject to: $a_t^s - a_{t-1}^s \geq 0 \qquad \forall s \in S \quad \forall t \in T$ (7.45)

$0 \leq a_t^s \leq M_1$ and integer $\qquad \forall s \in S \quad \forall t \in T$ (7.46)

Sub-problem 4:

Minimize $$-\sum_{\substack{r\in D \\ s\in S \\ sf\in S \\ s\neq sf \\ t\in T}} d_t^r \cdot w_t^{r,s} \cdot \varepsilon_t^{s,sf} - \sum_{\substack{r\in D \\ s\in S \\ sf\in S \\ s\neq sf \\ t\in T}} (\mathrm{w}_t^{r,s} + \mathrm{p}_t^{r,s}) \cdot \iota_t^{r,s,sf} - \rho \qquad (7.47)$$

Subject to:

$$\sum_{j \in S_n} w_{r,t}^{j} = 1 \qquad \forall r \in D \quad \forall n \in O_r \quad \forall t \in T \qquad (7.48)$$

$$\sum_{j \in S_n} w_{r,t}^{j} = 1 \qquad \forall r \in D \quad \forall n \in D_r \quad \forall t \in T \qquad (7.49)$$

$$\sum_{j \in S_n} p_{r,t}^{j} = 1 \qquad \forall r \in D \quad \forall n \in O_r \quad \forall t \in T \qquad (7.50)$$

$$\sum_{j \in S_n} p_{r,t}^{j} = 1 \qquad \forall r \in D \quad \forall n \in D_r \quad \forall t \in T \qquad (7.51)$$

$$w_{r,t}^{s} + p_{r,t}^{s} \le 1 \qquad \forall r \in D \quad \forall s \in S \quad \forall t \in T \qquad (7.52)$$

$$\sum_{j \in S_n} w_{r,t}^{j} + \sum_{j \in S_n} p_{r,t}^{j} \le 2 \qquad \forall r \in D \quad \forall n \in N \mid n \notin \{O_r, D_r\} \quad \forall t \in T \qquad (7.53)$$

$$0 \le w_{r,t}^{s}, p_{r,t}^{s} \le 1 \text{ and integer} \qquad \forall r \in D \quad \forall s \in S \quad \forall t \in T \qquad (7.54)$$

7.4 Experimental Study Method

To verify the efficiency of the Dantzig-Wolfe decomposition, we validate this approach through simulations conducted on six restricted networks. In other words, we use a subset of the eligible spans inherited from step 3 rather than full spans from original experimental networks. The formulations of restricted LP master MPSNA problem, sub-problems and the Dantzig-Wolfe decomposition, were implemented in AMPL and solved with Gurobi Optimizer 6.0.3 on a Mac Pro server with a 3.5 GHZ processor and a 32 GB RAM. After obtaining the tight lower bound, we added this bound back to the ILP

MPSNA problem and resolved it to see whether the solution runtime of the restricted version of the ILP MPSNA problem is decreased or not; and this new ILP problem was also solved with 1% optimality gap.

7.5 Results and Discussion

The Dantzig-Wolfe decomposition solutions of all six restricted networks appear in Table 7. More specifically, this table describes experimental networks, the solutions of step 4 of ILP-based heuristic which are considered as benchmarks in this step, and solutions of Dantzig-Wolfe decomposition. Here the Dantzig-Wofle decomposition solutions include two parts. One part is the solutions of getting lower bounds. The other is the solutions of resolving the ILP MPSNA problems with lower bounds. Each includes the information of values of objective functions, solution runtimes and optimality gaps. Because we solve a LP relaxation of ILP master problem for getting a lower bound, the solution optimality gap is optimal. However, for solving the ILP MPSNA problem with a lower bound, the optimality gap is set to 1% since the problem is an integer problem.

As shown in Table 7, the total solution runtimes of Dantzig-Wolfe decomposition is longer than solution runtimes of step 4 of the ILP-based heuristic. However, this result is not what we expected. If we look at Dantzig-Wolfe decomposition solutions, we find that it usually takes several seconds to get lower bounds, as we expected since the problem with block angular structure should be quickly solved by Dantzig-Wolfe decomposition.

In test cases of networks 8n9s through 20n22s, the solution runtimes of resolving ILP problem with lower bound are slightly shorter than the solution runtimes of step 4 of ILP-based heuristic (they are 0.1 seconds vs 0.15 seconds, 0.37 seconds vs 0.46 seconds, 1.53 seconds vs 1.95 seconds and 5.65 seconds vs 5.77 seconds, respectively). The comparison demonstrates that the lower bound play an effective role in reducing the search space of feasible solutions as we expected. However, in test cases of 25n28s and 30n36s networks, the solution runtimes of resolving the ILP problem with lower bound are increased by hundreds of seconds rather than decreased. In addition, in most test cases, the objective function values (total cost) of ILP problems with lower bounds are also worse than benchmarks. Overall, the Dantzig-Wolfe decomposition solutions do not meet our expectations and they prompt us to examine the quality of the lower bound (is it tight or loose?).

Table 7: The comparison of step 4 solutions of ILP-based heuristic and Dantzig-Wolfe decomposition solutions of all six restricted version of networks.

Network	**Step 4 of ILP-based heuristic**			**Dantzig-Wolfe decomposition**				
				Lower bound		**Add lower bound back to ILP problem and resolve it**		
	Cost	**Runtime (sec)**	**Optimality gap**	**Cost**	**Runtime (sec)**	**Cost**	**Runtime (sec)**	**Optimality gap**
8n9s	2,064,919	0.15	1%	246,269	0.06	2,064,919	0.10	1%
12n13s	1,151,002	0.46	1%	104,389	0.55	1,151,924	0.37	1%
15n17s	3,051,168	1.95	1%	247,848	1.36	3,050,337	1.53	1%
20n22s	3,073,526	5.77	1%	256,804	3.43	3,076,116	5.65	1%
25n28s	3,826,661	421	1%	300,046	10.76	3,831,890	576	1%
30n36s	5,070,193	1266	1%	406,557	17.58	5,073,403	1464	1%

Through examining the formulation of the MPSNA model, we suspect that the ILP models of sub-problems are unimodular since the variable coefficients in sub-problems are all 1, -1 or 0. As we have discussed in Chapter 2, the basic idea of the unimodularity is that if one ILP problem's coefficients have unimodular matrix, this ILP problem can be solved as a LP problem (relaxing the integrality of all variables) and all variables will still take integer values. In this case, if the MPSNA sub-problems have this property, the MPSNA problem will include the following relationship: $V_{(LP)} = V_{(DW)} < V_{(ILP)}$ (setting the objective function value of the LP MPSNA problem equal to the objective function value of the LP relaxation of the ILP master problem since we can get the same solutions even though we relax the integrality of all sub-problems' variables) rather than $V_{(LP)} < V_{(DW)} < V_{(ILP)}$ as we expected at the beginning.

Table 8: The solutions of the LP relaxation of ILP master problems and the LP relaxation of ILP original problems.

Network	**LP relaxation of the ILP master problem solved by Column Generation**		**LP relaxation of the ILP original problem**	
	Cost	**Optimality gap**	**Cost**	**Optimality gap**
8n9s	246,269.4422	Optimal	246,269.4422	Optimal
12n13s	104,388.6313	Optimal	104,388.6313	Optimal
15n17s	247,847.6279	Optimal	247,847.6279	Optimal
20n22s	256,804.2963	Optimal	256,804.2963	Optimal
25n28s	300,046.3288	Optimal	300,046.3288	Optimal
30n36s	406,557.1989	Optimal	406,557.1989	Optimal

The solutions of the LP relaxation of ILP master problems and the LP relaxation of ILP original problems are summarized in Table 8. After solving the LP relaxation of MPSNA problems, we found that the solutions of the LP relaxation of the MPSNA problems were totally equal to the solutions of LP relaxation of the ILP master MPSNA problems, which verified our suspicions. Now, we can explain why the solution runtimes of the MPSNA problems with lower bounds didn't decrease greatly; and in some test cases, it even increased. The reason for that is the ILP sub-problems are unimodular and the tight lower bounds we expected are actually equal to the solutions of the liner relaxation of the MPSNA problem. In other words, the lower bounds are loose lower bounds rather than tight lower bounds.

In summary, if the sub-problems of an ILP problem which has a block angular structure can be confirmed to be unimodular, we can reach the following conclusion: the bound of the ILP problem obtained by Dantzig-Wolfe decomposition is not a tight bound.

Chapter 8: Conclusion and Discussion

8.1 Summary of Thesis

The main objective of this thesis is to provide a method to effectively solve a multi-period survivable augmentation (MPSNA) problem. In this problem, the shared backup path protection (SBPP) mechanism is considered. To achieve the objective, we develop the ILP model for the MPSNA problem. However, we found that this problem is very time-consuming to solve even for an 8-node small network. To overcome this difficulty, we applied the four-stage ILP based-heuristic approach to solve the MPSNA problem and we also investigated the effectiveness of implementing Dantzig-Wolfe decomposition to solve this problem.

A brief background of backbone network and thesis outline is described in Chapter 1. Chapter 2 discusses integer linear programming optimization techniques and relevant literature reviews. Chapter 3 describes introduction of network survivability, network design paradigms and literatures of SBPP problem, topological network design problem and multi-period network design problem. In addition, Chapter 4 presents research goals and adopted methodologies.

In Chapter 5, we developed an ILP model for MPSNA problem. The MPSNA problem took the long-term incremental demands, network survivability and economies of scale into account, and focuses on network topology design, working demand routing, and spare capacity allocation. In addition, the SBPP mechanism was considered in this problem. The ILP model of MPSNA problem was implemented in AMPL and solved by Gurobi optimizer 6.0.3 with various optimality gaps. The experimental results showed that the MPSNA problem is very difficult to solve even for a small network test case.

In Chapter 6, we focused on decreasing the solution runtime of the MPSNA problem. A four-stage ILP-based heuristic approach was applied to this problem. We also implemented the ILP model of each step in AMPL and solved it with Gurobi Optimizer 6.0.3. Our experimental results showed that the ILP-based heuristic approach could provide better solutions in shorter runtimes in most test cases when compared to benchmark solutions. For example, in test case of a 15-node network, the solution runtime decreased by 99.8% and the total cost by 11.62%. In summary, the four-stage ILP-based heuristic approach was very effective to solve the MPSNA problem and could provide high-quality solutions in fast runtimes.

In Chapter 7, to reduce the solution runtimes further, we applied the Dantzig-Wolfe decomposition algorithm to solve the restricted version of the MPSNA problem. However, the experimental results showed no significant reduction in solution runtime. On the contrary, the implementation of Dantzig-Wolfe decomposition increased the solution runtime slightly. After the investigation, we found that Dantzig-Wolfe decomposition algorithm is not efficient to solve the ILP problems whose coefficient matrices are unimodular since this kind of problem has the following property: the LP relaxation of the ILP original problems is totally equal to the LP relaxation of the ILP master problem. As a result, we could not get a tight bound for the ILP problems.

8.2 Main contributions

There are three main contributions of this thesis. These contributions are summarized as follows:

1. Chapter 5: Develop an ILP model for the multi-period network augmentation (MPSNA) problem with the consideration of the SBPP mechanism.

2. Chapter 6: Develop a four-stage ILP-based heuristic approach based on the three-stage heuristic method proposed in [3] and [13] for effectively solving the MPSNA problem.

3. Chapter 7: Simplify the ILP model of the MPSNA problem, find the model's block angular structure, apply the Dantzig-Wolfe decomposition to solve the MPSNA problem and investigate the effectiveness of implementation of Dantzig-Wolfe decomposition.

8.3 Other contributions

Besides the contributions listed in Section 8.2, one conference paper has been accepted and one manuscript has been finished ready to submit.

1. Y. Wang, J. Doucette, "ILP-based Heuristic Method for the Multi-period Survivable Network Augmentation Problem," *Reliable Network Design and Modeling* (*RNDM 2016*), accepted: 17-June, 2016.

2. Y. Wang, J. Doucette, "Dantzig-Wolfe Decomposition Algorithm for the Multi-period Survivable Network Augmentation Problem," ready to submit.

8.4 Future Work

Several research topics are recommended for future study. They are listed as follows:

First, the demand uncertainty can be taken into account when the traffic demand is forecasted. As we all know the accuracy of the demand forecasting affects the network design directly; however, the traditional demand forecasting methods (e.g. according to past demand history or using some econometric forecasting models to forecast future demand) sometimes cannot satisfy present telecommunication environment which is dynamic and uncertain . As a result, it is necessary to design the multi-period network with the consideration of the demand uncertainty; however, the complexity of the problem will be definitely increased.

Second, with the rapid technology innovation, when to introduce new technologies to network is also a problem which network designers care about during the multi-period network design. Like other economic factors (e.g., modularity and economies of scale), the factor of technology innovation can also be formulated into the ILP model.

Third, the MPSNA model can be formulated with other popular network survivability mechanisms such as span restoration and *p*-cycle. In addition, the four-stage ILP-based heuristic approach developed in this thesis can also be applied to solve MPSNA problems with other survivability mechanisms because of its wide suitability and ease of use.

Reference

[1] TIA, "TIA's 2009 ICT Market Review and Forecast," 2009. [Online]. Available: http://www.tiaonline.org/images/mrf/Chapter2_2009MRF_preview.pdf. [Accessed: 16-Aug-2015].

[2] B. Todd, "The Use of Demand-wise Shared Protection in Creating Topology Optimized High Availability Networks," University of Alberta, Canada, 2009.

[3] J. Doucette, "Advances on Design and Analysis of Mesh-restorable Networks," University of Alberta, Canada, 2005.

[4] W. D. Grover, *Mesh-based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*. Upper Saddle River, NJ: Prentice Hall PTR, 2004.

[5] J. Doucette, "Basic Network Survivability Concepts," Lecture 02, ENG M 680 B1, Network Optimization and Reliability, University of Alberta, Winter 2006.

[6] "Computer Network," *Wikipedia, the free encyclopedia*. 04-Aug-2015.

[7] "Backbone network," *Wikipedia, the free encyclopedia*. 07-Jun-2015.

[8] "Access network," *Wikipedia, the free encyclopedia*. 24-Mar-2016.

[9] PCMag, "Acess Network." [Online]. Available: http://www.pcmag.com/encyclopedia/term/37391/access-network. [Accessed: 05-Jan-2016].

[10] "Wavelength-division multiplexing," *Wikipedia, the free encyclopedia*. 04-Apr-2016.

[11] M. Bahleda and K. Blunar, "The Gain of Performance of Optical WDM Networks," *Journal of Computer Networks and Communications*, vol. 2008, p. e289690, Mar. 2008.

[12] T. G. Robertazzi, *Planning Telecommunication Networks*, 1st edition Wiley-IEEE Press, 1998.

[13] W. D. Grover and J. Doucette, “Topological Design of Survivable Mesh-Based Transport Networks,” *Annals of Operations Research*, vol. 106, no. 1–4, pp. 79–125, Sep. 2001.

[14] B. Van Caenegem, W. Van Parys, F. De Turck, and P. M. Demeester, “Dimensioning of Survivable WDM Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, pp. 1146–1157, Sep. 1998.

[15] Y. Miyao and H. Saito, “Optimal Design and Evaluation of Survivable WDM Transport Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, pp. 1190–1198, Sep. 1998.

[16] W. D. Grover and D. Y. Li, “The Forcer Concept and Express Route Planning in Mesh-survivable Networks,” *Journal of Network and Systems Management*, vol. 7, no. 2, pp. 199–223, 1999.

[17] S.-G. Chang and B. Gavish, “Telecommunications network topological design and capacity expansion: Formulations and algorithms,” *Telecommunication Systems*, vol. 1, no. 1, pp. 99–131, Dec. 1993.

[18] W. L. Winston, *Operations Research: Applications and Algorithms*, 4th edition. Belmont, CA: Cengage Learning, 2003.

[19] D.-S. Chen, R. G. Batson, and Y. Dang, *Applied Integer Programming: modeling and solution*. Hoboken, New Jersey: John Wiley & Sons, 2010.

[20] E. L. Lawler and D. E. Wood, “Branch-and-Bound Methods: A Survey,” *Operations Research*, vol. 14, no. 4, pp. 699–719, Aug. 1966.

[21] J. E. Kelley, “The cutting-plane method for solving convex programs,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 4, pp. 703–712, 1960.

[22] M. Padberg and G. Rinaldi, “A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems,” *SIAM Review*, vol. 33, no. 1, pp. 60–100, Mar. 1991.

[23] IBM, “IBM ILOG AMPL Version 12.1 User’s Guide,” 2009.

[24] M. Noor-E-Alam, “Advanced Integer Linear Programming Techniques for Large Scale Grid-Based Location Problems,” University of Alberta, Canada, 2013.

[25] M. L. Fisher, “An Applications Oriented Guide to Lagrangian Relaxation,” *Interfaces*, vol. 15, no. 2, pp. 10–21, Mar. 1985.

[26] J. Desrosiers and M. E. Lübbecke, “A Primer in Column Generation,” in *Column Generation*, Springer US, 2005, pp. 1–32.

[27] L. S. Lasdon, *Optimization Theory for Large Systems*. Mineola, New York: Dover Publications, 2011.

[28] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, “Branch-and-Price: Column Generation for Solving Huge Integer Programs,” *Operation Research*, vol. 46, pp. 316–329, 1996.

[29] G. B. Dantzig and P. Wolfe, “Decomposition Principle for Linear Programs,” *Operation Research*, vol. 8, no. 1, pp. 101–111, Jan. 1960.

[30] F. Vanderbeck, “On Dantzig-Wolfe Decomposition in Integer Programming and ways to Perform Branching in a Branch-and-Price Algorithm,” *Operation Research*, vol. 48, no. 1, pp. 111–128, Feb. 2000.

[31] G. Desaulniers, J. Desrosiers, and M. M. Solomon, Eds., *Column generation*. New York: Springer, 2005.

[32] J. Desrosiers, M. E. Lübbecke, J. J. Cochran, L. A. Cox, P. Keskinocak, J. P. Kharoufeh, and J. C. Smith, “Branch-Price-and-Cut Algorithms,” in *Wiley Encyclopedia of Operations Research and Management Science*, John Wiley & Sons, Inc., 2010.

[33] M. Lübbecke and J. Desrosiers, “Selected Topics in Column Generation,” *Operation Research*, vol. 53, pp. 1007–1023, 2004.

[34] A. J. Hoffman, “Total unimodularity and combinatorial theorems,” *Linear Algebra and its Applications*, vol. 13, no. 1, pp. 103–108, Jan. 1976.

[35] “Unimodular matrix,” *Wikipedia, the free encyclopedia*. 18-Apr-2016.

[36] C. Blum and A. Roli, “Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison,” *ACM Computing Surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, Sep. 2003.

[37] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, Massachusetts: MIT Press, 1998.

[38] F. Glover, “Tabu Search—Part I,” *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, Aug. 1989.

[39] E. Aarts, J. Korst, and W. Michiels, “Simulated annealing,” in *Search methodologies*, Springer, 2005, pp. 187–210.

[40] B.-L. Garcia, P. Mahey, and L. J. LeBlanc, “Iterative improvement methods for a multiperiod network design problem,” *European Journal of Operational Research*, vol. 110, no. 1, pp. 150–165, Oct. 1998.

[41] M. Pickavet and P. Demeester, “Multi-Period Planning of Survivable WDM Networks,” *European Transactions on Telecommunications*, vol. 11, no. 1, pp. 7–16, Jan. 2000.

[42] K. Ross, “Meta-heuristics.” [Online]. Available: https://classes.soe.ucsc.edu/ism206/Fall10/Lecture12.pdf. [Accessed: 28-Sep-2015].

[43] D. Xu, C. Qiao, and Y. Xiong, “An Ultra-fast Shared Path Protection Scheme - Distributed Partial Information Management, Part II,” in *10th IEEE International Conference on Network Protocols, 2002. Proceedings*, 2002, pp. 344–353.

[44] K. Zhu and B. Mukherjee, “Traffic grooming in an optical WDM mesh network,” *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, pp. 122–133, Jan. 2002.

[45] S. Gorshe, “A Tutorial on SONET/SDH Automatic Protection Switching (APS),” 2005.

[46] R. Bhandari, *Survivable Networks - Algorithms for Diverse Routing*, 1st Edition, Springer US, 1999.

[47] C. Wang, G. Shen, B. Chen, and L. Peng, “Protection Path-based Hitless Spectrum Defragmentation in Eastic Optical Networks: Shared Backup Path Protection,” presented at the Optical Fiber Communication Conference, 2015.

[48] J. Doucette, M. Clouqueur, and G. Wayne D., “On the Availability and Capacity Requirement of Shared Backup Path Protected Mesh Networks,” *Optical Networks Magazine*, pp. 29–44, 2003.

[49] M. Herzberg, S. J. Bye, and A. Utano, “The hop-limit approach for spare-capacity assignment in survivable networks,” *IEEE/ACM Transactions on Networking*, vol. 3, no. 6, pp. 775–784, Dec. 1995.

[50] J. L. Kennington and M. W. Lewis, “The Path Restoration Version of the Spare Capacity Allocation Problem with Modularity Restrictions: Models, Algorithms, and an Empirical Analysis,” *INFORMS Journal on Computing*, vol. 13, no. 3, pp. 181–190, 2001.

[51] M. Herzberg and S. J. Bye, “An Optimal Spare-capacity Assignment Model for Survivable Networks with Hop limits,” in *IEEE Global Telecommunications Conference, 1994. GLOBECOM ’94 Communications: The Global Bridge*, 1994, vol. 3, pp. 1601–1606.

[52] B. D. Venables, W. D. Grover, and M. H. MacGregor, “Two Strategies for Spare Capacity Placement in Mesh Restorable Networks,” in *Technical Program, Conference Record, IEEE International Conference on Communications, 1993. ICC ’93 Geneva*, 1993, vol. 1, pp. 267–271.

[53] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Amsterdam; Boston: Elsevier/Morgan Kaufmann, 2004.

[54] M. Pickavet and P. Demeester, "Long-Term Planning of WDM Networks: A Comparison between Single-Period and Multi-Period Techniques," *Photonic Network Communication*, vol. 1, no. 4, pp. 331–346, Dec. 1999.

[55] B. Yaged, "Minimum Cost Routing for Dynamic Network Models," *Networks*, vol. 3, no. 3, pp. 193–224, Jan. 1973.

[56] D. A. Schupke and C. Kronberger, "Multiperiod Planning for Optical Networks," in *Telecommunications Network Strategy and Planning Symposium (NETWORKS), 2010 14th International*, 2010, pp. 1–6.

[57] B. Todd and J. Doucette, "Fast Efficient Design of Shared Backup Path Protected Networks Using a Multi-Flow Optimization Model," *IEEE Transactions on Reliability*, vol. 60, no. 4, pp. 788–800, Dec. 2011.

[58] J. T. Haahr, T. Stidsen, and M. Zachariasen, "Heuristic Methods for Shared Backup Path Protection Planning," in *2012 4th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2012, pp. 712–718,

[59] C. Ou, J. Zhang, H. Zang, L. H. Sahasrabuddhe, and B. Mukherjee, "New and Improved Approaches for Shared-path Protection in WDM Mesh Networks," *Journal of Lightwave Technology*, vol. 22, no. 5, pp. 1223–1232, May 2004.

[60] T. Stidsen, B. Petersen, S. Spoorendonk, M. Zachariasen, and K. B. Rasmussen, "Optimal Routing with Failure-independent Path Protection," *Networks*, vol. 55, no. 2, pp. 125–137, Mar. 2010.

[61] H. N. Nguyen, D. Habibi, Q. V. Phung, and K. Lo, "A Shared Backup Path Protection Scheme for Optical Mesh Networks," in *2005 Asia-Pacific Conference on Communications*, 2005, pp. 309–313.

[62] G. Li, D. Wang, C. Kalmanek, and R. Doverspike, "Efficient Distributed Path Selection for Shared Restoration Connections," in *IEEE INFOCOM 2002.*

Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings, 2002, vol. 1, pp. 140–149 vol.1.

[63] M. Tornatore, G. Maier, and A. Pattavina, "Variable Aggregation in the ILP Design of WDM Networks with Dedicated Protection," *Journal of Communications and Networks*, vol. 9, no. 4, pp. 419–427, Dec. 2007.

[64] K. Altinkemer, "Topological Design of Ring Networks," *Computers & Operations Research*, vol. 21, no. 4, pp. 421–431, Apr. 1994.

[65] R. Elbaum and M. Sidi, "Topological Design of Local-area Networks Using Genetic Algorithms," *IEEE/ACM Transactions on Networking (TON)*, vol. 4, no. 5, pp. 766–778, 1996.

[66] K. Altinkemer and Z. Yu, "Topological Design of Wide Area Communication Networks," *Annals of Operations Research*, vol. 36, no. 1, pp. 365–381, 1992.

[67] K. Guan and C. Guan, "Long-term Capacity Planning of WDM Transport Networks Under Demand Uncertainties and Economy of Scale," in *2006 11th International Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks*, 2006, pp. 100–106.

[68] F. D'Andreagiovanni, J. Krolikowski, and J. Pulaj, "A fast hybrid primal heuristic for multiband robust capacitated network design with multiple time periods," *Applied Soft Computing*, vol. 26, pp. 497–507, Jan. 2015.

[69] R. Kouassi, M. Gendreau, J.-Y. Potvin, and P. Soriano, "Heuristics for multi-period capacity expansion in local telecommunications networks," *Journal of Heuristics*, vol. 15, no. 4, pp. 381–402, Aug. 2009.

[70] M. Gendreau, J.-Y. Potvin, A. Smires, and P. Soriano, "Multi-period capacity expansion for a local access telecommunications network," *European Journal of Operational Research*, vol. 172, no. 3, pp. 1051–1066, Aug. 2006.

[71] O. E. Flippo, A. W. Kolen, A. M. Koster, and R. L. Van de Leensel, "A dynamic programming algorithm for the local access telecommunication network expansion

problem," *European Journal of Operational Research*, vol. 127, no. 1, pp. 189–202, 2000.

[72] H. F. Lee and D. R. Dooly, "Heuristic algorithms for the fiber optic network expansion problem," *Telecommunication Systems*, vol. 7, no. 4, pp. 355–378, Aug. 1997.

[73] C. Meusburger, D. A. Schupke, and J. Eberspacher, "Multiperiod Planning for Optical Networks - Approaches Based on Cost Optimization and Limited Budget," in *IEEE International Conference on Communications, 2008. ICC '08*, 2008, pp. 5390–5395.

[74] C. Kronberger, D. A. Schupke, and J. Eberspacher, "Multiperiod Planning with Controlled Shortest Path Routing," in *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, 2010, pp. 1–6.

[75] S. Straub, A. Kirstadter, and D. A. Schupke, "Multi-period Planning of WDM-Networks: Comparison of Incremental and EoL Approaches," in *2006 2nd IEEE/IFIP International Conference in Central Asia on Internet*, 2006, pp. 1–7.

[76] N. Geary, A. Antonopoulos, E. Drakopoulos, and J. O'Reilly, "Analysis of Optimisation Issues in Multi-period DWDM Network Planning," in *IEEE INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, 2001, vol. 1, pp. 152–158 vol.1.

[77] B. Todd and J. Doucette, "DSP survivable network capacity allocation and topology design using multi-period network augmentation," in *Design of Reliable Communication Networks (DRCN), 2015 11th International Conference on the*, 2015, pp. 41–48.

[78] R. N. de A. Vidal, C. Kronberger, and D. A. Schupke, "Multiperiod planning of optical protection schemes," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*, 2011, pp. 641–645.

[79] M. Pioro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Elsevier, 2004.

[80] J. Doucette and W. D. Grover, “Influence of modularity and economy-of-scale effects on design of mesh-restorable DWDM networks,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1912–1923, Oct. 2000.

[81] A. Kershenbaum, P. Kermani, and G. A. Grover, “MENTOR: An Algorithm for Mesh Network Topological Optimization and Routing,” *IEEE Transactions on Communications*, vol. 39, no. 4, pp. 503–513, Apr. 1991.

[82] D. Xu, C. Qiao, and Y. Xiong, “An Ultra-fast Shared Path Protection Scheme - Distributed Partial Information Management, Part II,” in *10th IEEE International Conference on Network Protocols, 2002. Proceedings*, 2002, pp. 344–353.

[83] F. Vanderbeck and M. W. P. Savelsbergh, “A generic view of Dantzig–Wolfe decomposition in mixed integer programming,” Operations Research Letters, vol. 34, no. 3, pp. 296–306, May 2006.