2004

A tabu search heuristic for routing in WDM networks.

Shaohong Wang
University of Windsor

Recommended Citation

Wang, Shaohong, "A tabu search heuristic for routing in WDM networks." (2004). *Electronic Theses and Dissertations*. 845.
https://scholar.uwindsor.ca/etd/845

A Tabu Search Heuristic for Routing in WDM Networks

By

Shaohong Wang

A Thesis

Submitted to the Faculty of Graduate Studies and Research through the School of Computer Science in Partial Fulfillment of the Requirements for the Degree of Master of Science at the University of Windsor

Windsor, Ontario, Canada

January 2004

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisisitons et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-92532-3
Our file Notre référence
ISBN: 0-612-92532-3

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Abstract

Optical networks and Wavelength-Division Multiplexing (WDM) have been widely studied and utilized in recent years. By exploiting the huge bandwidth of optical networks, WDM appears to be one of the most promising technologies to meet the dramatically increased demand for bandwidth. Since optical resources in optical networks are very expensive, development of dynamic lightpath allocation strategies, which utilize network resource efficiently, is an important area of research. We assume that there is no optical wavelength conversion device in the network, and the wavelength-continuity constraint must be satisfied.

Exact optimization techniques are typically too time-consuming to be useful for practical-sized networks. In this thesis we present a tabu search based heuristic approach which is used to establish an optimal lightpath dynamically in response to a new communication request in a WDM network. As far as we know, this is the first investigation using tabu search techniques for dynamical lightpath allocation in WDM networks. We have tested our approach with networks having different sizes. And then we have compared our results with those obtained using the MILP approach. In the vast majority of cases, tabu search was able to quickly generate a solution that was optimal or near-optimal, indicating that tabu search is a promising approach for the dynamic lightpath allocation problem in WDM networks.

Acknowledgments

A special thank and appreciation must be offered to my supervisor, Dr. Subir Bandyopadhyay, for his invaluable help and support in the whole master degree pursuiting process. Your kindness, patience, and readiness to help have impressed me greatly. You are greatly appreciated and will be remembered forever. And my co-supervisor, Dr. Arunita Jaekel, has also assisted me greatly, especially in the thesis project selection, idea creation, and smoothness of the thesis. You are also greatly appreciated. A special thank must also be offered to the committee members, Dr. Kao, Dr. Aggarwal and Dr. Lu for your time contribution in reading my thesis and the great suggestions. And thanks must also be offered to Mr. Sood, your assistance makes the thesis more smoothly and completely. You are appreciated too. Meanwhile, thanks must also be offered to all of my teachers and friends who have helped me so much. Without all of the above assistance, this thesis could not have happened. All of you will be greatly appreciated.

A special thank should also be offered to my parents and my daughter, for your love, support and encourage. Courage is not simply one of the virtues at the testing point. Also as you know, there is no elevator to success, but I would like to climb on my feet all the way along at full speed as you hoped.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

With the rapid development of communication networks in recent years, the demand for network bandwidth has been increasing dramatically. Optical networks are widely utilized in long-haul networks. Wavelength-Division Multiplexing (WDM) appears to be one of the most promising technologies to satisfy the exploding need for increased bandwidth resulting from e-commerce, cluster computing and other needs for large scale communication of data. In an optical network, the transmissions from many different end-users can be combined and transmitted on the same fiber [Muk97]. By transmitting data using a number of separate optical carriers on each fiber, with each optical carrier having a distinct wavelength, WDM technology can exploit the huge bandwidth of optical networks.

A typical WDM network is shown in Figure 1.1. In this diagram each node, denoted by a square, is an end-node which can generate data and can receive data. These are the computers in a network that can communicate with each other. Each node denoted by a circle is a router node which is an optical device that has a number of incoming and outgoing fibers, each carrying optical signals. These routers can take an optical signal at some frequency say ω_1, using any incoming fiber and send it on any outgoing fiber using some optical frequency ω_2. In some scenarios the routers are equipped with wavelength conversion devices so that ω_1 and ω_2 may be distinct. In real-life networks, all-optical wavelength converters are rarely used. In such cases $\omega_1 = \omega_2$. In our investigation we do not allow wavelength conversion.

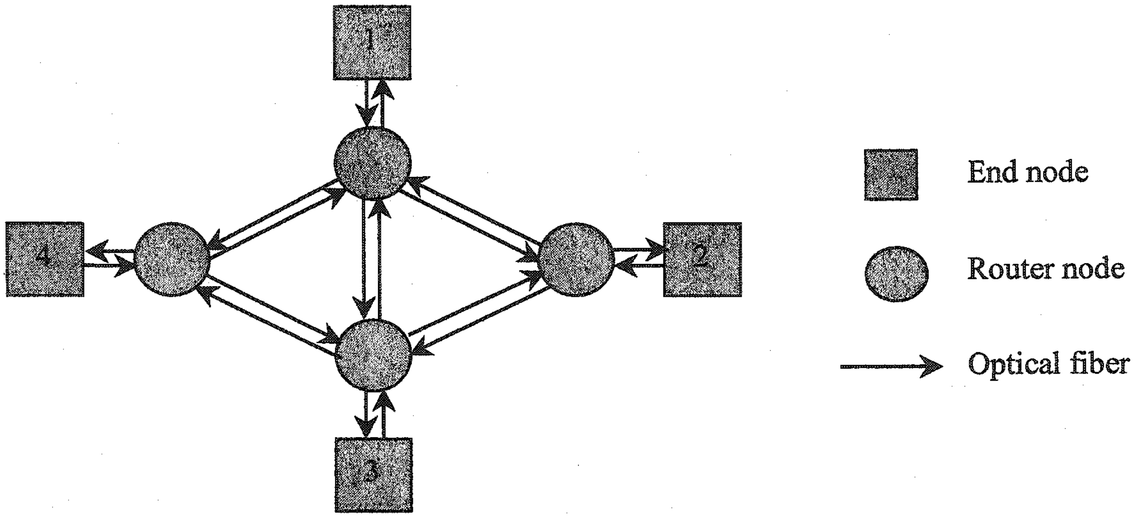


Figure 1.1 An Optical Network

The low impedance region of an optical fiber is 1200 -1600nm. This region may be divided into a number of intervals so that each optical carrier using any given fiber operates within a given band. Depending on the technology used and the characteristics of the fibers in the network, there can be up to 200 different carrier wavelengths on a single fiber [Muk97]. Each possible carrier wavelength on a given fiber is called a *channel*. If a channel is being used for some communication that channel cannot be used for any other communication. The channels that are not used for any communication are called *free*.

In order to communicate from a source s to a destination d, it must be possible to find a route $s \rightarrow x_1 \rightarrow x_2 \rightarrow \ldots x_k \rightarrow d$ such that

a) there exists a fiber connecting node s to node x_1, a fiber connecting node x_1 to node x_2, … a fiber connecting node x_k to node d,

b) there exists a free channel on the fiber connecting node s to node x_1, free channel on the fiber connecting node x_1 to node x_2, … free channel on the fiber connecting node x_k to node d.

A lightpath [Muk00] from a source s to a destination d is an all-optical communication path between nodes s and d in the network. If a lightpath spans more than one fiber link, the intermediate nodes in the fiber path are optical routers that direct the lightpath in the optical domain from some incoming fiber to some outgoing fiber. In an all-optical WDM network, all communication uses lightpaths. Once we are given a physical topology of an optical network, and the traffic requirements between each end-node in the network, the design problem is to find a set of lightpaths to carry the specified traffic using a minimum amount of network resources.

Two scenarios of setting up lightpaths have been considered by researchers. In the static scenario, the traffic requirements are specified in advance and the problem is to find a set of lightpaths to handle the traffic, using maximum demands on the system [Mar93] [Chl93] [CGLM+00]. In such a case, the traffic requirement is met beforehand and lightpaths are set up on a quasi permanent basis. In the dynamic scenario, each lightpath from a source node ***s*** to a destination node ***d*** is set up in response to a request for communication from ***s*** to ***d***. Later on, when the communication is over, the lightpath is taken down [Ger96a][Ger96b]. In such a case, the problem is to set up the lightpath, on demand using a minimum amount of network resources. In other words, the number of channels needed for the lightpath should be as low as possible. An important feature of dynamic lightpath allocation is the fact that not all requests for communication may be successful. For example, it is quite possible that a source node ***s*** needs to communicate with some destination node ***d***. However, the existing lightpaths are such that it is not possible to set up any lightpath between ***s*** and ***d***. In such cases the request is said to be *blocked*.

Tabu search (TS) is a widely utilized meta-heuristic procedure which is based on concepts that unite the fields of Artificial Intelligence and Operations Research [Glo90] [Ach00]. By using TS, many problems such as vehicle routing problems [BO99][XK96], quadratic assignment problem [CC98] and timetabling problems [Sch96] can find solutions superior to the best previously obtained by alternative methods [GT93]. To our knowledge, the tabu search has not been used to solve the problem of dynamic lightpath allocation. In this thesis our goal is to investigate whether tabu search may be profitably used for this problem.

1.1 Motivation

The objective of this investigation is to study whether tabu search may be employed to find a "good" route and wavelength for each optical signal in a WDM network. In our approach we wish to use as little network resources as possible. Our objective is to find a wavelength and a route such that the number of fibers used by each lightpath is as small as possible. We wish to determine if the solutions obtained using tabu search is comparable in quality to those obtained from exact optimization techniques.

1.2 Problem Outline

In this investigation, the network is an all-optical WDM network, and we already have a number of existing lightpaths which were set up in response to previous requests for communication in the network. When a new connection request from a given source to a given destination node arrives, by using a tabu search, we attempt to establish a lightpath from the source to the destination. The lightpath should be as short as possible. As stated earlier, it is assumed that the wavelength-continuity constraint must be satisfied.

1.3 Thesis Organization

In Chapter 2 we will introduce all-optical WDM network and tabu search which are two of the most important parts of the background information. Three dynamic lightpath allocation approaches: breadth-first search, MILP (Mixed Integer Linear Programming), and tabu search will be introduced in Chapter 3. And in Chapter 4, we will state how to implement the tabu search approach in detail and will compare the results of the tabu search approach with those of the MILP approach. And finally, Chapter 5 will include the conclusions and future work.

Chapter 2 Background Review

2.1 All-Optical Network

There are three generations of networks based on the physical technology having been utilized. Before the employment of the optical fiber technology, the networks were mainly based on the technology of copper-link or microwave radio. These are the first generation computer networks. Examples of such networks are Ethernet, IEEE 802.4 token bus, IEEE 802.5 token ring, and Cambridge ring. In second-generation networks, the optical fibers are employed to replace copper-links in traditional networks. Compared to the copper links, optical fibers have higher data rates, lower error rates, and reduced electromagnetic emissions from the cabling. Such networks employ electronic front ends at the network nodes, and electro-optic conversion is needed at each node. The third generation networks are all-optical networks. In these networks, the end-to-end connections are established in the optical domain. Electronics at a node are only needed to handle the data intended for that node. Data that is being routed through a node remain in the optical domain and routing is accomplished using all-optical routers. The information transmitted may remain optical domain and there is no need in electronic-optical conversion [Muk92]. These networks can eliminate the electro-optic bottleneck since from the source to the destination of the network optical signals remain in the optical domain. Therefore, in an all-optical network the information transfer rates are significantly beyond the rates that are possible in an electronic network [MA98].

2.2 Major Components of Optical Network

There are three major components in the optical network: Multiplexer, Demultiplexer, and Optical Router.

- **Multiplexer**: A device which combines the optical signals at the different wavelengths arriving on the different incoming fibers into one fiber.
- **Demultiplexer**: The device splits the optical signals at different carrier wavelengths on one fiber into signals on a number of fibers where each fiber carries only one signal at distinct wavelength [RS98].
- **Optical Router**: A wavelength router is an optical switch that is capable of routing a signal based on its input port and its wavelength [MA98].

2.3 Wavelength Division Multiplexing (WDM) Networks

Wavelength Division Multiplexing technique (WDM) [Muk97] is an approach which can be utilized to exploit the huge bandwidth of optical networks. In WDM networks, the bandwidth of the fiber is divided into many non-overlapping wavelength bands called WDM channels. Each channel can be used on a single fiber in parallel at the peak electronic speed, e.g., a few gigabits per second (GBps). Therefore, the aggregated network capacity of WDM network can reach the capacity of each channel multiplied by the number of channels. Nowadays, the number of channels on each fiber can reach up to 200, while each channel supports a data rate at 2.5 – 10 (GBps) [AQ00][Gua03][Hou03]. Two popular architectures have evolved as candidates for WDM networks: broadcast-and-select networks and wavelength routed networks.

2.4 Broadcast-and-Select WDM Networks

Broadcast-and-select networks are suitable for local area networks (LAN) with a small number of users (Figure 2.1). The transmitters on the end nodes broadcast the signals on distinct wavelengths on the network, The star coupler combines these signals and distributes the aggregate signals to the receivers. The receivers can tuned to receive the desired signal.

Suppose there are N nodes in the network, the N nodes are connected through an N*N passive broadcast star. Thus, the signal transmitted by any node is received by all nodes. Routing, management, and control of the connections in the broadcast-and-select networks are relatively simple. However, there is a lot of wasted power, since each transmission is received by all other nodes, even if they are not the intended recipient [MA98].

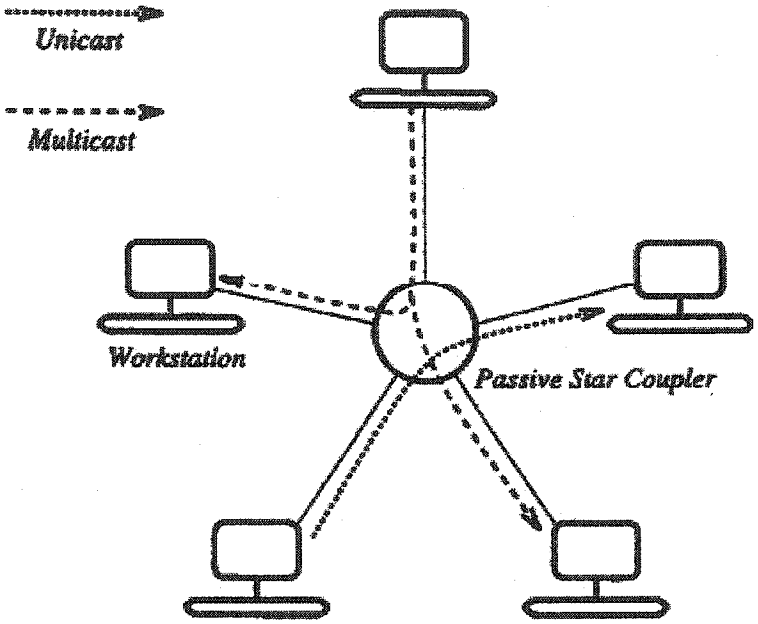
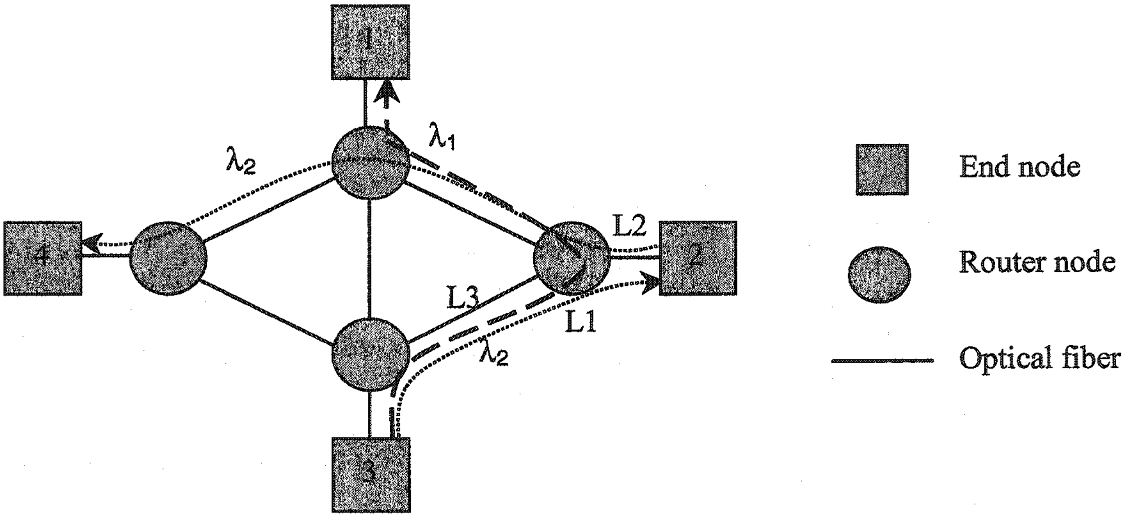


Figure 2.1 A Passive-Star-Based Local Optical WDM Network

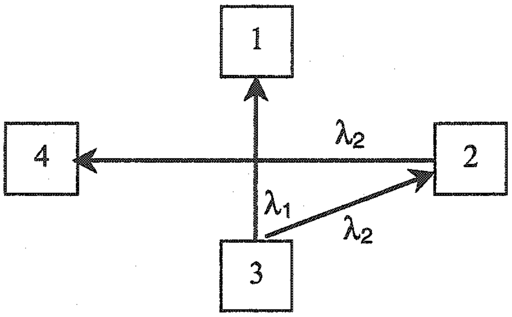
Because of the power budget limitations and the lack of wavelength reuse, the broadcast-and-select architecture is not suitable for wide area networks (WAN). Another architecture that is widely used for WANs is the wavelength routed WDM network [MA98].

2.5 Wavelength Routed WDM Networks

Wavelength routed networks (Refer to Figure 2.2) are composed of optical routing nodes interconnected by optical links. Each link is assumed to be bidirectional and actually consists of a pair of unidirectional optical fibers. An optical routing node consists of an end node and an optical router which has a number of input and output ports. An optical router can route each wavelength on an incoming port to any outgoing port. However, the same wavelength on the different incoming ports cannot be routed simultaneously onto a single outgoing port [RS95][Hou03].



a) Wavelength Routed Network (Physical Topology)



b) Wavelength Routed Network (Logical Topology)

Figure 2.2 Physical Topology and Logical Topology

2.5.1 Physical Topology

The physical topology of a wavelength routed network consists of optical routers, end-nodes, and the fibers connecting them [Muk97]. End-nodes are the nodes which are attached to the optical fibers that can generate, send, and receive data. Figure 2.2a is an example of physical topology of wavelength routed network.

2.5.2 Lightpath

In an optical network, a lightpath is a point-to-point communication path that connects a transmitter at a source node to a receiver at a destination node [SRM02]. There are no optical-to-electronic conversions in the lightpath, and thus there is no need to worry about the electronic delay at intermediate nodes along the lightpath [Sah00]. For example, in Figure 2.2a, there are three lightpaths shown as, 3 →2, 2 →4 and 3 →1. The lightpath between node 3 and node 1 is a connection on wavelength λ_1 which passes through three intermediate optical routers. Similarly the lightpath between node 3 and node 2 is a connection on wavelength λ_2 which passes through two intermediate optical routers.

2.5.2.1 Wavelength Continuity

It is known that the wavelength router is an optical switch that is capable of routing a signal from an incoming port to an outgoing port. In some wavelength routed networks, wavelength converters are utilized. Wavelength converters have the capability of changing the wavelength of the signal it routes. In networks employing wavelength converters, a lightpath can use different wavelengths on all of fibers along its route. If there is no wavelength-conversion in the network, a lightpath is required to have the same wavelength throughout its entire path. This is referred to as the wavelength-continuity constraint of the lightpath [Muk00]. In the network of Figure 2.2a, wavelength-continuity is assumed, so the lightpath from node 3 to node 1 is required to use the same wavelength (λ_1) on all fibers from the source to the destination.

2.5.2.2 Wavelength Distinct

In a wavelength-routed optical network, if two or more lightpaths traverse the same fiber link, they must be on different wavelength channels. This ensures that the two signals do not interfere with one another [Muk00]. For example, in Figure 2.2a, the two lightpaths L_1 and L_3 originating at node 3, both traverse the same fiber (3→2). Therefore, they cannot be assigned to the same wavelength. In our example, they are assigned λ_2 and λ_1 respectively.

2.5.3 Logical Topology

The logical topology of an optical network [Muk97] can be regarded as a directed graph, in which the vertices correspond of the network to the end nodes and the edges represent lightpaths. It can be visualized as an optical layer on which high speed packet switched networks can be built [CGK1][Ach00]. Figure 2.2(b) is the logical topology for the networks of Figure 2.2(a).

2.6 Single-Hop and Multi-Hop Networks

If in an optical network all nodes can communicate with one another in just one hop (i.e. by traversing one lightpath), it is called a single-hop network. In a single-hop network, a packet transmitted from its source reaches its destination directly without going through any end nodes. Single-hop networks are all-optical networks [Muk92a] which are designed to avoid the electronics bottleneck (the performance limitation imposed by the maximum speed of electronics) employed in switches and end-nodes. That is, the information is conveyed in optical domain (without any electro-optical conversions) through the network until it reaches the destination which results very fast communication.

Obviously, in order to ensure single-hop communication, a network must have a lightpath between all source-destination pairs. This means, N*(N-1) lightpaths are needed in an N-node network. Normally, this is impractical. In the real-life networks, if two end-nodes s and d are not connected by a lightpath, we need to determine a sequence of nodes x_1, x_2, ..., x_k so that there is a lightpath from source node s to node x_1, a lightpath from node x_1 to x_2, ..., node x_k to destination node d. The communication from s to d can be accomplished in a number of hops. These networks are often called multi-hop networks [Muk92b] [Gua03].

In Figure 2.2, communications from node 3 to node 1, from node 3 to node 2, and from node 2 to node 4 are examples of single-hop communication, using lightpaths L3, L1 and L2 respectively. However, transmission from node 3 to node 4 requires multiple hops, over lightpaths L1 and L2 to establish this communication.

2.7 Routing and Wavelength Assignment

In a wavelength-routed WDM network, the end users communicate with one another via lightpaths. A lightpath is utilized to support a connection in a wavelength-routed WDM network, and it may span multiple fiber links [ZJM00].

Given a set of connections, the problem of setting up lightpaths by routing over the physical topology and assigning a wavelength to each connection is called the *routing and wavelength assignment* (RWA) problem. Typically, connection requests may be of two types, static and dynamic, depending on whether all the lightpath requests are known initially and fixed over time or not.

With static traffic, the entire set of connections is known in advance, and the problem is then to set up lightpaths for these connections. The objective is to minimize network resources such as the number of distinct carrier wavelengths required to accommodate the traffic. Alternatively, for a given fixed number of wavelengths, one may attempt to set up as many of these connections as possible. The RWA problem for static traffic is known as the *Static Lightpath Establishment* (SLE) problem. This is suitable for large transport networks or wide-area networks (WANs), in which the traffic demand is relatively fixed over time.

In the dynamic traffic case, the lightpath requests are assumed to arrive at sequentially. The lightpath is set up when the request arrives, and the resources used by the lightpath are released after the connection is terminated. The objective, in the dynamic traffic case, is to set up lightpaths and assign wavelengths in a manner which minimizes the amount of resource required, or which maximizes the number of connections that are established in the network at any time. This problem is referred to as the *Dynamic Lightpath Establishment* (DLE) problem. This is suitable for the data networks or local-area networks (LAN), in which the traffic demand frequently changes over time [ZJM00] [MBLN93].

In the wavelength-routed optical networks, the routing problem consists of two components. The first is to determine a path (over the physical topology) along which the connection can be established. The second component is to assign a wavelength to the selected path. In networks without wavelength converting devices, the same wavelength must be used on all links along the selected path. The wavelength assignment must be such that no two lightpaths sharing a link are assigned to the same wavelength [MA98].

2.8 Breadth-First Search and Depth-First Search

Breadth-First and Depth-First Searches are two types of the most important blind search techniques. They are called *blind* searches because they make no use of information concerning where the goal might be located in the search space. They just exhaustively search the space, checking each object for a possible goal [DAA95].

These two techniques can be described by how the *search tree* (representing all the possible paths from the starting state) will be traversed. In the depth-first search, when a state is examined, all of its children and their descendants are examined before any of its siblings. The depth-first search gets very quickly into a deep search space but it may not find the best solution [Lug02] [DAA95]. The order in which nodes are visited during depth-first search in Figure 2.3 [DAA95] is A, B, D, E, C, F, and G.

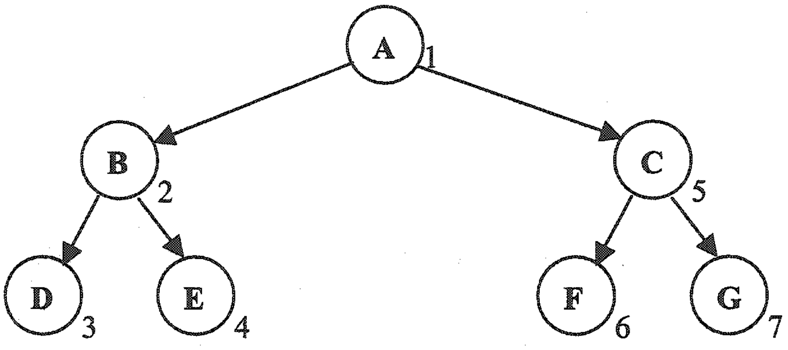


Figure 2.3 Visit Order of Depth-first search

The following are the algorithm of the depth-first search [DAA95]:

1. Set *N* to be a list of initial nodes.
2. If *N* is empty, then exit and give a signal of failure.
3. Set *n* to be the first node in *N*, and remove *n* from *N*.
4. If *n* is a goal node, then exit and signal success.
5. Otherwise, add the children of *n* to the front of *N* and return to step 2.

The breadth-first search explores the space by using level-by-level fashion. Only when there are no more states to be explored at a given level does the algorithm move on to the next level [Lug02]. The order in which the nodes are visited during breadth-first search in Figure 2.4 [DAA95] is A, B, C, D, E, F, and G.

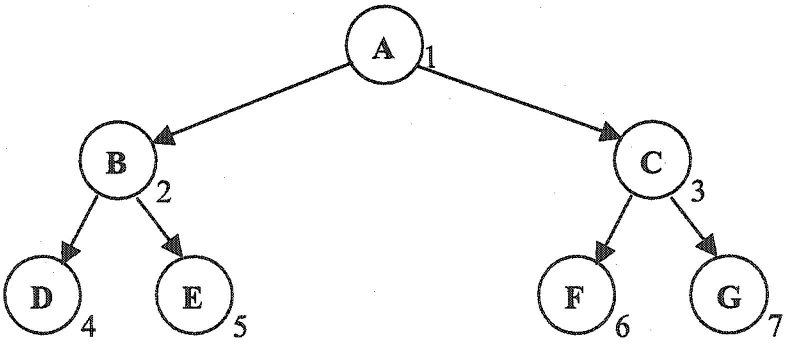


Figure 2.4 Visit Order of Breadth-first search

The following are the algorithm of the breadth-first search [DAA95]:

1. Set *N* to be a list of initial nodes.
2. If *N* is empty, then exit and give a signal of failure.
3. Set *n* to be the first node in *N*, and remove *n* from *N*.
4. If *n* is a goal node, then exit and signal success.
5. Otherwise, add the children of n to the end of *N* and return to step 2.

The breadth-first search can find the best solution if one does exist, provided that there are a finite number of branches in the tree [Ric83] [DAA95]. Because the breadth-first search considers every node at each level of the diagram before going deeper into the space, all states are first reached along the shortest path from the start state. Therefore, the breadth-first search is guaranteed to find the shortest path from the start state to the goal [Lug02]

2.9 Heuristic Search

We assume that we do not use any information about the structure of the search space to guide our search in the blind search techniques, instead, we perform an exhaustive exploration in the search space [DAA95]. But in some cases, a problem, such as medical diagnosis, may not have an exact solution because of inherent ambiguities in the problem statement or available data [Lug02]. In the other cases, a problem may have an exact solution, but the computational cost of finding it may be prohibitive, such as chess

[Lug02]. In these situations, it is not practical to use the blind search techniques to find the intended solutions.

In order to solve these hard problems efficiently, it is often necessary to compromise the requirements of mobility and systematicity and to construct a control structure that is no longer guaranteed to find the best solution but that will almost always find a very good solution [Ric83]. A *heuristic* was defined as such technique that will help in the discovery of solutions to problems even though there is no guarantee that it will never lead in the wrong direction [Ric83]. In state space search, heuristics are formalized as rules for choosing those branches in a state space that are most likely to lead to an acceptable problem solution [Lug02].

In heuristic technique, we assume a metric on objects on the search space that allows us to estimate the distance from a node to a goal [DAA95]. A *heuristic function* is a function that maps from problem state descriptions to measures of desirability [Ric83]. In other words, a heuristic function is such a metric. A well-designed heuristic function can play an important part in guiding a search process efficiently toward a solution [Ric83].

2.10 Tabu Search

Many of the newly-developed optimization techniques come from the existing natural and physical phenomena. For example, genetic algorithms resemble the biological phenomenon of evolutionary reproduction, while simulated annealing is derived from the physical process of metallurgy. Tabu search is a technique which is based on selected concepts of artificial intelligence [GT93]. Although tabu search was proposed in the late 1960's and in the early 1970's, its 'modern form', which is currently widely utilized was introduced by F. Glover in 1986[Glo86].

Tabu search (TS) is a general heuristic procedure for guiding search to obtain good solutions in complex solution spaces [GT93]. As a heuristic procedure, TS can solve many problems in the real world. For example, a TS approach of production scheduling problem in hot strip mill was introduced in [LCG98], high school timetabling problems using the tabu search techniques was discussed in [Sch96], and a new TS method for solving power utilities problem was presented in [GQ96].

It shows that, for some hard problems, such as the vehicle routing problems, [XK96] and [BO99], the quadratic assignment problem, [CC98], and the timetabling problems, [Sch96], TS can find solutions superior to the best previously obtained by alternative methods [GT93].

2. 10.1 What is Tabu Search?

Tabu search can be viewed as a repetitive process that explores a set of problem solutions, denoted by X, by making moves from one solution s to another solution s' in its neighborhood B(s) iteratively. The objective of the search is to efficiently yield an optimal or near-optimal solution by the evaluation of some objective function f(*s*) to be minimized [GT93]. By using a "forbidding" strategy, this approach can overcome the problem of local optimality and obtain global optimization. This approach might also be called "weak inhibition" search, because the forbidden moves (the moves which are marked as "tabu") are generally only a small part of the moves in the search space, and they will be available soon after they lose the tabu status [Glo86].

Generally speaking, there are two kinds of TS, one simpler and one more advanced, *short-term memory* and *longer-term memory* [GL97]. Both short-term memory and longer-term memory are designated to modify the neighborhood structure of the current state. But in the short-term memory, the modified neighborhood is a subset of the original one, however, in the longer-term memory, the modified neighborhood can probably be expanded because of the newly added solutions in the neighborhood [Ach00].

2.10.2 Short-Term Memory

"The core of TS is embedded in its short-term memory process" [Glo90]. That means that the most important features of TS are embedded in its short-term memory process, any of the characteristics that reappear in the longer-term memory will be amplified in some degree but will not be changed greatly.

Figure 2.5 is a simplified diagram depicting how tabu search works using the short-term memory [Glo90]. The short-term memory forms an aggressive exploration which can be employed to find the best move in a set of available moves that satisfy certain constraints. These constraints including the tabu restrictions are designed to prevent reversal or repetition of some moves by identifying some attributes of these moves that are forbidden (tabu). The step of choosing the best admissible candidate, the most important step in short-term memory, is illustrated in Figure 2.6 [Glo90].

"The primary goal of the tabu restrictions is to permit the method to go beyond the points of local optimality while still making high-quality moves at each step." [Glo90]. If there are no such restrictions, the method may take a move which is far away from a local optimum, and then by taking the best move available at that point, it can also reach the local optimum again [Glov90]. But sometimes, the tabu conditions based on selected attributes of moves and solution may be so strong that they may also be prevented from visiting the unvisited solution, especially for those that may be very promising. Therefore, in some situations, it is necessary to overrule the tabu status of moves. This is controlled by *aspiration criteria* [GT93]. In general, by cooperating with the application of aspiration criteria, the tabu restrictions are employed to avoid such cycling behavior and try to lead the search for a new trajectory if it revisits some earlier solutions again and again [Glo90].

Choosing the best admissible candidate (refer to Figure 2.6) is the most important step in the short-term memory. The moves that are in the candidate list are evaluated one by one. When a move is in "tabu" status, the move will be forbidden for some time. In other words, that move cannot be applied for a number of iterations called the *tabu tenure* for the move. But how to evaluate a move is an important issue. In some problems, the evaluation for the move can be based on how much it can change the value of the objective function. But in other problems, to evaluate a move is not so easy because not all the current variables have the assigned values. The evaluation may be based on an approximate solution or just used for local measures of the attractiveness of the move [Glo90].

Because the moves which are marked 'tabu' are relatively much fewer than those which are not marked as 'tabu', for situations where it is not very expensive to evaluate a move, it would be beneficial to check if a given move is better than others first before checking the its tabu status. If the move is not tabu, it is accepted as admissible immediately. Otherwise, it may also qualify as admissible if it satisfies the aspiration criteria [Glo90].

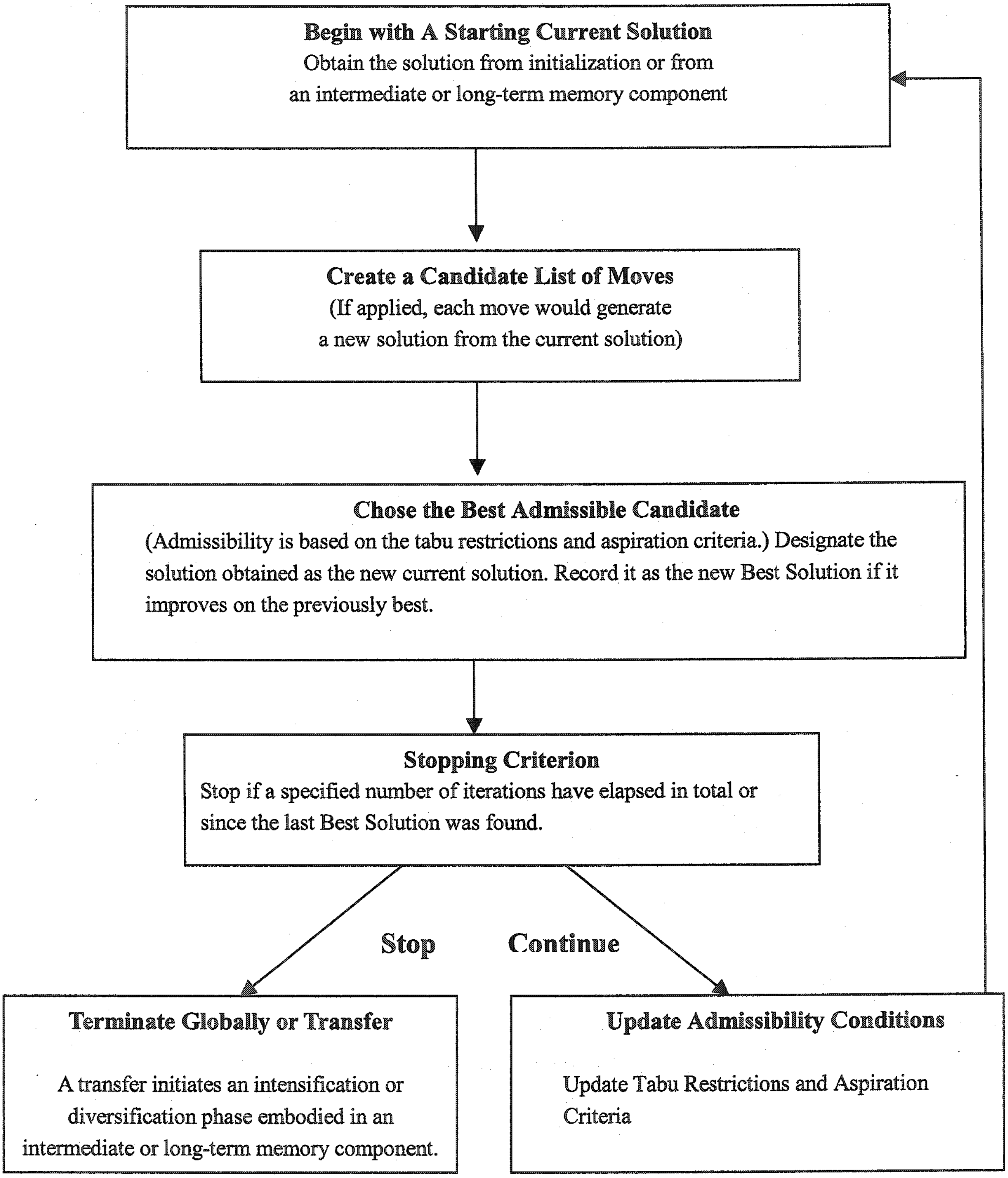


Figure 2.5 Tabu Search Short-Term Memory Components

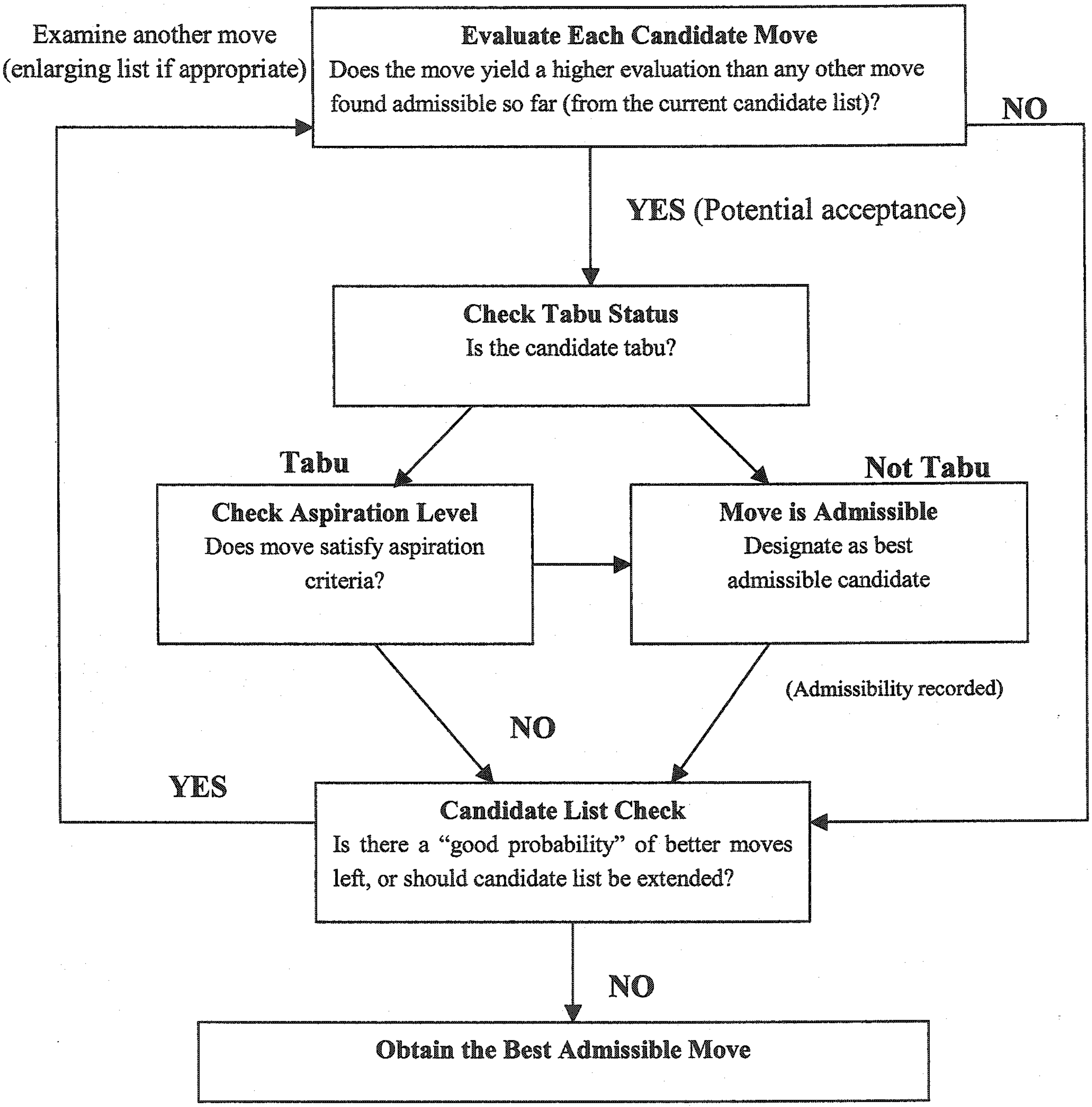


Figure 2.6 Selecting the Best Admissible Candidate

In a candidate list strategy, examining the next move may be needed. In some cases, none of the available moves may qualify as admissible (which means acceptable) because of the limitation of the tabu restrictions and aspiration criteria. A "least inadmissible" move is selected as admissible when this situation occurs [Glo90].

2.10.3 Longer-Term Memory

In many cases, just by using the short-term memory, one can obtain a high quality solution. But for many problems, especially for the hard problems, in order to obtain good solutions, longer-term memory has to be employed, such as in [CFG98][SAMD98].

TS memory can be used in a learning process: after visiting some solutions, it will identify whether the good solutions which have been visited so far have some common characteristics. This generates an intensification scheme for the search [GT93]. *Intensification strategies* are based on modifying choice rules to encourage move combinations and solution features historically found good [GL97]. Then, in the same stage of the search, the neighborhood is restricted to select the solutions with the good characteristics [GT93].

In order to get the best results, intensification only is not enough. One should allow the most effective search over the neighborhood. This is *diversification strategy*, the complementary notion of intensification strategy. Which means it can induce the search to other search areas in which the solutions with other characteristics can be obtained [GT93].

As the basis of strategies for intensifying and diversifying the search, in the longer-term, by alternatively using intensification and diversification, TS iteratively intensifies the search in a promising area and diversifies the search to other areas [GT93].

In fact, the short-term memory has already included the fundamental element of intensification and diversification strategies, since tabu list of short-term memory acts as an intensification role by temporarily forbidding using some attributes [Glo90]. Actually, the longer-term memory process can be composed by creating and testing the short-term memory first, and then adding the refinement [Glo90]. And there are some papers discuss intensification and diversification strategies such as [LMC99][FILG01].

2.10.4 The Extensions of Tabu Search

Reactive tabu search is an extension of basic tabu search. The examination for the repetition of a configuration is added to the original tabu search scheme. By responding to the appearance of cycles, the size of the search list will be known automatically in this scheme. If a number of solutions repeat too many times, the search is diversified by making a variety of random moves according to the moving average of the cycle length

[BT94]. There are some successful applications of reactive tabu search, such as, [BR99], [OW02], [BT95a] and [BT95b].

In [BR99], the authors develop a reactive tabu search with a path relinking algorithm for the Steiner problem in graphs. In this approach, the neighborhood is defined by insertions and deletions of the Steiner nodes. By comparing with the previous algorithm that is the algorithm using reactive tabu search can obtain better performance.

Based on a modified greedy search component, the optimization process of the reactive tabu search algorithm is utilized in [BT95b]. By marking the most recent move as tabu status, the process successfully prevents possible cycling, and the prohibition period is adapted automatically. The diversification strategy is used in the algorithm to prevent the search process from being restricted to a limited area of the search space.

In [NW00], the authors develop a reactive tabu search algorithm to solve the pickup and delivery problem with time windows. Three different move neighborhoods in that algorithm are utilized. In order to communicate with different areas of the solution space and adjust search direction, a hierarchical search methodology is used to dynamically alternate between neighborhoods. The authors claim that "this is the first fully implemented method to be effectively applied to a set of practical-sized multiple vehicle instances of the pickup and delivery problem with time windows".

In [KL98], by using two variables in the unconstrained discrete optimization model, the authors try to study the effectiveness of reactive tabu search with a static tabu list. One of the most important characteristics of this problem is that it can provide graphical descriptions of the reactive tabu search.

Except for reactive tabu search, there are some other extensions of TS such as MOTS (Multi-objective Optimization Tabu Search) that was introduced in [Han97], ECTS (Enhanced Continuous Tabu Search) that was presented in [CS00], Robust Tabu Search that was introduced in [Tai91], and so forth. But they are not as widely used as reactive tabu search.

2.11 Introduction of CPLEX

CPLEX is an optimization software tool for large-scale, mission-critical applications. It can offer high-performance, flexible optimizers for solving linear, mixed-integer and quadratic programming. In addition to providing the robust algorithms for demanding problems, CPLEX can solve many problems with millions of constraints and variables, and as a mathematical programming software, it keeps very good performance. It is a reliable, flexible and robust support for the development of industry applications [IL01].

The name "CPLEX" originates from two words, with the letter "C" representing C programming language, and the word "PLEX" representing the 'simplex' method for linear programming. Developed in the C programming language, CPLEX is the first commercial linear programming optimizer. CPLEX Linear Optimizer and CPLEX Callable Library are the first CPLEX products provided by CPLEX Optimization, Inc., a company founded in 1988 with the mission of providing the highest-performance optimizers for linear programming [IL01].

CPLEX is a very efficient tool to deal with linear optimization problems that are generally referred to as Linear Programming (LP) problems, of the form:

$$\begin{array}{ll} \text{Minimize (or Maximize)} & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\ \text{Subject to} & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \sim b_1 \\ & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \sim b_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \sim b_m \end{array}$$

with these bounds

$$l_i \leq x_i \leq u_i$$

$\cdot$

$\cdot$

$l_n \leq x_n \leq u_n$

where ~ can be ≤,≥ or =, and the upper bounds u_i and lower bounds l_i maybe positive infinity, negative infinity, or any real number.

The data the user provide as input for this LP are:

- Objective function coefficients $c_1,\ldots, c_n$
- Constraint coefficients $a_{11},\ldots a_{mn}$
- Right_hand sides $b_1,\ldots,b_m$
- Upper and lower bounds $u_1,\ldots,u_n$ and $l_1,\ldots,l_n$

The optimal solutions that CPLEX computes and returns are the values of the variables $x_1,\ldots x_n$.

Chapter 3 Techniques for Dynamic Lightpath Allocation in WDM Networks

In this chapter we will discuss three possible ways we may attempt to allocate a lightpath when we get a request for communication from a source node *s* to a destination node ***d***. The first approach is based on a standard breadth-first search of the network. The second approach uses a Mixed Integer Linear Program (MILP) to specify the problem. The third approach is based on the Tabu Search. We start with a model of the WDM network and follow up with the three approaches using this model.

3.1 The Network Model

This is a model of the all-optical WDM networks. In the model we will use the wavelength continuity constraint so that the entire lightpath is allotted a single channel. The network consisting of N end nodes, R optical routers and E edges may have any mesh topology. Each edge connects two nodes (which can be end nodes or router nodes). Each edge in the network is a bidirectional link and consists of two undirectional fibers. Each fiber has the capability of carrying the same set of carrier wavelength $\lambda_1, \lambda_2, ..., \lambda_k$, simultaneously and we will use channel j to denote the wavelength λ_j, for any j, $1 \leq j \leq k$. Figure 3.1 is an example of a WDM all-optical network. In this figure, each circle with a digit in it represents a node (which can be either an end node or a router) and we assume that k = 2 so that there are two channels having carrier wavelengths λ_1 and λ_2 on each fiber in this network.

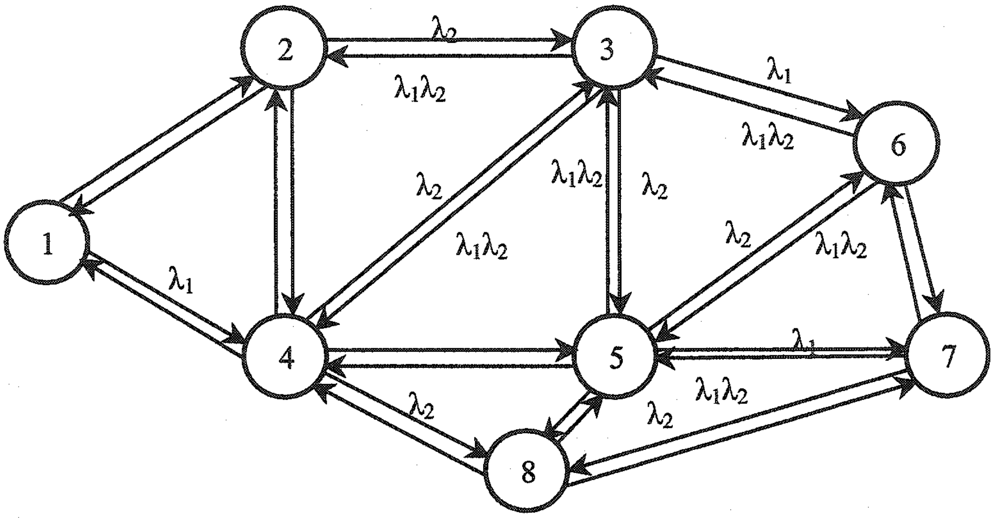


Figure 3.1 An Example of a Network with Some Existing Lightpaths

3.2 The Objective

Our goal is to establish an optimal lightpath when there is a request for communication from some source node s to some destination node d. In response to such a request, we wish to find a path $s \rightarrow x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_r \rightarrow d$ and a channel p (i.e., a carrier wavelength λ_p) for some p, $1 \le p \le k$ such that there is a fiber from s to x_1, a fiber from x_1 to x_2, … a fiber from x_r to d such that channel p is not allotted to any other existing communication that uses any of the fibers in the path from s to d.

This is the general case where the path $s \rightarrow x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_r \rightarrow d$ involves (r + 1) fibers and r router nodes, $r \geq 0$.

The lightpath should be optimal, in the sense of using as little network resources as possible. This means that we are interested in finding a lightpath that uses the smallest value of r. For example, in Figure 3.2(a), the network already has some existing lightpaths. The wavelengths shown beside each fiber are the wavelengths which have been allocated to existing lightpaths traversing that fiber.

Let there be a new request for a connection from node 2 to node 6. The problem is to find an optimal lightpath in response to this request. Since the wavelength continuity constraints must be satisfied, the wavelengths on each fiber traversed by the lightpath must be the same. In general, in response to such a request for communication from *s* to ***d,*** there are many lightpaths that we may use. For example, for communication from node 2 to node 6, the following lightpaths may be used:

i) 2 → 4 → 5 → 7 → 6, with wavelength λ_2.

ii) 2 → 1 → 4 → 5 → 7 → 6, with wavelength λ_2.

iii) 2 → 4 → 8 → 7 → 6, with wavelength λ_1.

iv) 2 → 4 → 5 → 6, with wavelength λ_1.

v) 2 → 4 → 8 → 5 → 6, with wavelength λ_1,

vi) 2 → 3 → 5 → 6, with wavelength λ_1.

A network may have more than one optimum lightpath. In this example, the lightpaths using the path 2 → 4 → 5 → 6 (with the wavelength λ_1) and the path 2 → 3→ 5 → 6 (with the wavelength λ_1) are the optimum lightpaths since they have the shortest path length.

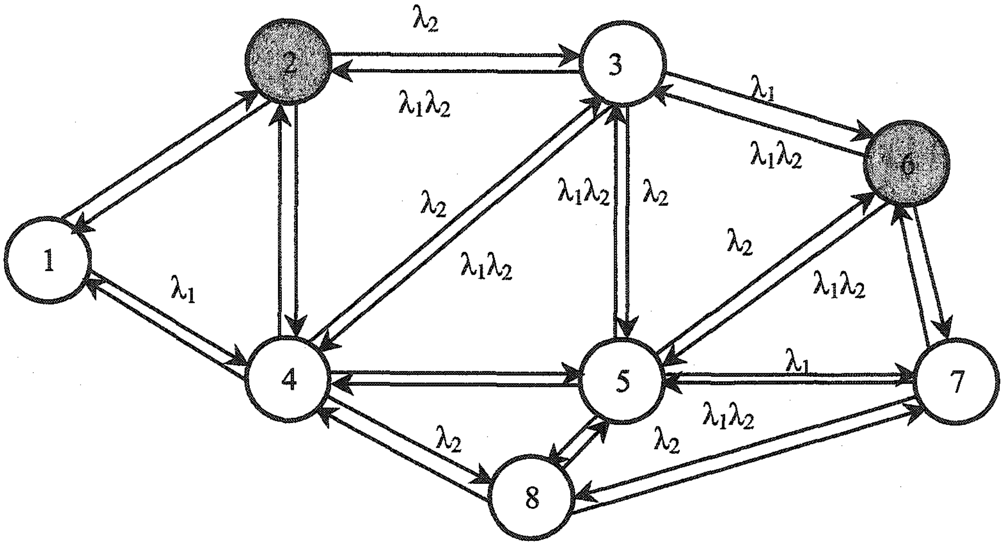


Figure 3.2(a) Servicing a Request for Communication from Node 2 to Node 6

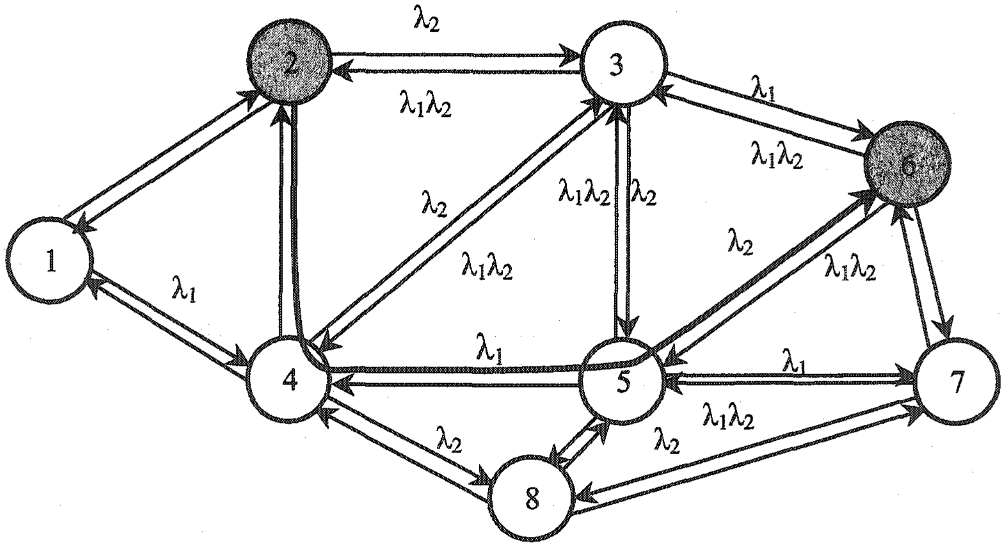


Figure 3.2(b) Servicing a Request for Communication from Node 2 to Node 6

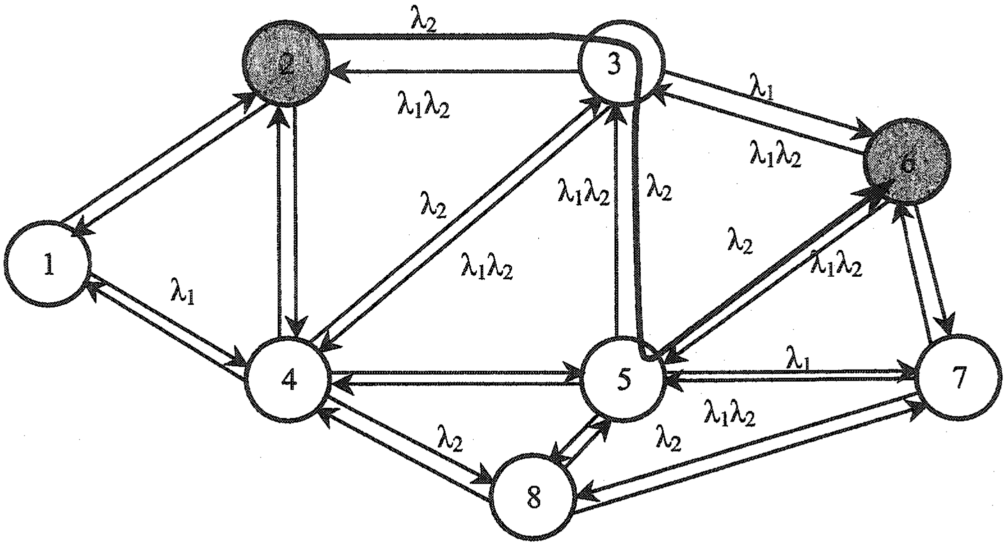


Figure 3.2(c) Servicing a Request for Communication from Node 2 to Node 6

3.3 Breadth-First Search Approach

We can use the breadth-first search to solve the problem of allocating lightpaths in dynamic WDM networks. To take account of the wavelength continuity constraint, we have to modify the basic breadth-first search algorithm. Our breadth-first algorithm, which can be used to find a shortest path for a given source-destination pair (***s***, ***d***), under the wavelength continuity constraint, is given below.

In this algorithm, we will use $L_{i,j}$ to denote the set of wavelengths already used on the edge $i \rightarrow j$. For example, in Fig 3.2, $L_{5,3}$ is $\{\lambda_1, \lambda_2\}$. We will also use ***LP*** to denote an ordered list of pairs (***P***, Λ) where ***P*** denotes a path $s \rightarrow x_1 \rightarrow x_2 \rightarrow \ldots x_p$ from the source node s to some node x_p in the network and Λ denotes a non-empty set of wavelengths. This means that we may use the path $P = s \rightarrow x_1 \rightarrow x_2 \rightarrow \ldots x_p$ to establish a lightpath from source node s to some node x_p using any of the wavelengths in Λ. We will use lastnode(P) to denote the last node x_p in such a path. Unlike normal breadth-first searches, there may be multiple occurrences of a node x_p in the paths appearing in different elements of ***LP***. However the first time a node x_p appears in any path-wavelengths pair (P, Λ) in ***LP***, the length of the path from s to x_p is the shortest path from s to x_p. The steps of the algorithm are given below:

Step 1) Set ***LP*** to be a list consisting of one element - the pair (s, $\{\lambda_1, \lambda_2, \ldots \lambda_k\}$) where s is the source node and $\lambda_1, \lambda_2, \ldots \lambda_k$ are the wavelengths of the k channels available on any fiber in the network.

Step 2) If ***LP*** is empty, then exit and signal failure.

Step 3) Let $n = (P, \Lambda)$ be the first element in the ordered list ***LP***, where P is the path $s \rightarrow \ldots \rightarrow x$, Remove n from ***LP***.

Step 4) If x is the destination node, then exit and signal success.

Step 5) Otherwise, repeat steps 6 - 7 for all nodes y such that $x \rightarrow y$ is a directed edge in the network and y does not appear in path P.

Step 6) If $\Lambda - L_{x,y}$ is not empty, check if there is any other element (P', Λ') in the ordered list ***LP***, such that,

a. Lastnode(P') = y

b. $\Lambda' \supseteq \Lambda - L_{x,y}$

Step 7) If $\Lambda - L_{x,y}$ is not empty and an element (P', Λ') satisfying the conditions of step 6 cannot be found, create a new path $P^{new} = s \rightarrow \ldots \rightarrow x \rightarrow y$ and insert the pair $(P^{new}, \Lambda - L_{x,y})$ so that the ordered list ***LP*** is sorted in ascending order of the number of elements in the path ***P*** of each element of ***LP***.

For example, we want to establish an optimal lightpath from the source node 2 to destination node 6 in Figure 3.1. $\Lambda = \{\lambda_1, \lambda_2\}$.

The steps are given below:

1) Start with the source node 2. ***LP*** $= [(2, \{\lambda_1, \lambda_2\})]$
2) $n = (2, \{\lambda_1, \lambda_2\})$. ***LP*** $= []$
3) 2 is not the destination node.
4) $2 \rightarrow 1$ is an edge. The fiber is not carrying any lightpath giving us the pair $2 \rightarrow 1$, $\{\lambda_1, \lambda_2\}$.

   $2 \rightarrow 3$ is an edge. The wavelength λ_2 has been used in the fiber giving us the pair $2 \rightarrow 3$, $\{\lambda_1\}$.

   $2 \rightarrow 4$ is an edge. The fiber is not carrying any lightpath giving us the pair $2 \rightarrow 4$, $\{\lambda_1, \lambda_2\}$.
5) ***LP*** $= [(2 \rightarrow 1, \{\lambda_1, \lambda_2\}), (2 \rightarrow 3, \{\lambda_{1,}\}), (2 \rightarrow 4, \{\lambda_1, \lambda_2\})]$
6) $n = (2 \rightarrow 1, \{\lambda_1, \lambda_2\})$, ***LP*** $= [(2 \rightarrow 3, \{\lambda_1\}), (2 \rightarrow 4, \{\lambda_1, \lambda_2\})]$
7) 1 is not the destination node.

8) $1 \rightarrow 4$ is an edge. The wavelength λ_2 has been used, we have $2 \rightarrow 1 \rightarrow 4$, $\{ \lambda_2 \}$, but there is a element $(2 \rightarrow 4, \{ \lambda_1, \lambda_2 \})$, ***P'*** = 4, $\Lambda' = \{ \lambda_1, \lambda_2 \}$, $\Lambda - L_{1,4} = \{ \lambda_1 \}$, $\Lambda' \supseteq \Lambda - L_{1,4}$, so we will not add $(2 \rightarrow 1 \rightarrow 4, \{ \lambda_2 \})$ to ***LP***.

9) ***LP*** = $[(2 \rightarrow 3, \{ \lambda_1 \}), (2 \rightarrow 4, \{ \lambda_1, \lambda_2 \})]$

10) n = $(2 \rightarrow 3, \{ \lambda_1 \})$, ***LP*** = $[(2 \rightarrow 4, \{ \lambda_1, \lambda_2 \})]$

11) 3 is not the destination node.

12) $3 \rightarrow 4$ is an edge. But the both wavelength λ_1 and λ_2 have been used in the fiber.

$3 \rightarrow 5$ is an edge. The wavelength λ_2 has been used giving us the pair $2 \rightarrow 3 \rightarrow 5$, $\{\lambda_1\}$.

$3 \rightarrow 6$ is an edge. But the both wavelength λ_1 and λ_2 have been used up to node 6.

13) ***LP*** = $[(2 \rightarrow 4, \{ \lambda_1, \lambda_2 \}), (2 \rightarrow 3 \rightarrow 5, \{ \lambda_1 \})]$

14) n = $(2 \rightarrow 4, \{ \lambda_1, \lambda_2 \})$. ***LP*** = $[(2 \rightarrow 3 \rightarrow 5, \{ \lambda_1 \})]$

15) 4 is not the destination node.

16) $4 \rightarrow 1$ is an edge. The fiber is not carrying any lightpath giving us the pair $2 \rightarrow 4 \rightarrow 1$, $\{ \lambda_1, \lambda_2 \}$.

$4 \rightarrow 3$ is an edge. The wavelength λ_2 has been used giving us the pair $2 \rightarrow 4 \rightarrow 3$, $\{ \lambda_1 \}$.

$4 \rightarrow 5$ is an edge. The fiber is not carrying any lightpath giving us the pair $2 \rightarrow 4 \rightarrow 5$, $\{\lambda_1, \lambda_2\}$.

$4 \rightarrow 8$ is an edge. The wavelength λ_2 has been used, giving us the pair $2 \rightarrow 4 \rightarrow 8$, $\{ \lambda_1 \}$.

17) ***LP*** = [(2 → 3 → 5, { λ_1 }), (2 → 4 → 1, { λ_1, λ_2 }) , (2 → 4 → 3, { λ_1 }), (2 → 4 → 5, { λ_1, λ_2 }), (2 → 4 → 8,{ λ_1 })]

18) n = (2 → 3 →5, { λ_1 }), ***LP*** = [(2 → 4 → 1, { λ_1, λ_2 }), (2 → 4 → 3,{ λ_1 }), (2 → 4 → 5, { λ_1, λ_2 }), (2 → 4 → 8,{ λ_1 })]

19) 5 is not the destination node.

5 → 6 is an edge. The wavelength λ_2 has been used, giving us the pair ***2 → 3 → 5 → 6, { λ_1 }***.

The remaining steps are omitted since we have reached the destination node 6.

The value of *LP* in different steps are given below:

$[(2, \{ \lambda_1, \lambda_2 \})]$

⇨

$[(2 \to 1, \{ \lambda_1, \lambda_2 \}),$
$(2 \to 3, \{ \lambda_1 \}),$
$(2 \to 4, \{ \lambda_1, \lambda_2 \})]$

⇨

$[(2 \to 3, \{\lambda_1\}),$
$(2 \to 4, \{\lambda_1, \lambda_2\})]$

⇨

$[(2 \to 4, \{\lambda_1, \lambda_2\}),$
$(2 \to 3 \to 5, \{\lambda_1\})]$

⇨

$[(2 \to 3 \to 5, \{\lambda_1\}),$
$(2 \to 4 \to 1, \{ \lambda_1, \lambda_2 \}),$
$(2 \to 4 \to 3, \{ \lambda_1 \}),$
$(2 \to 4 \to 5, \{ \lambda_1, \lambda_2 \}),$
$(2 \to 4 \to 8, \{ \lambda_1 \})]$

⇨

$[(2 \to 4 \to 1, \{ \lambda_1, \lambda_2 \}),$
$(2 \to 4 \to 3, \{ \lambda_1 \}),$
$(2 \to 4 \to 5, \{ \lambda_1, \lambda_2 \}),$
$(2 \to 4 \to 8, \{ \lambda_1 \}),$
$(2 \to 3 \to 5 \to 4, \{\lambda_1\}),$
$(2 \to 3 \to 5 \to 6, \{\lambda_1\})]$

⇨

$[(2 \to 4 \to 3, \{ \lambda_1 \}),$
$(2 \to 4 \to 5, \{ \lambda_1, \lambda_2 \}),$
$(2 \to 4 \to 8, \{ \lambda_1 \}),$
$(2 \to 3 \to 5 \to 4, \{\lambda_1\}),$
$(2 \to 3 \to 5 \to 6, \{\lambda_1\})]$

⇨

$[(2 \to 4 \to 5, \{ \lambda_1, \lambda_2 \}),$
$(2 \to 4 \to 8, \{ \lambda_1 \}),$
$(2 \to 3 \to 5 \to 4, \{\lambda_1\}),$
$(2 \to 3 \to 5 \to 6, \{\lambda_1\})]$

⇨

$[(2 \to 4 \to 8, \{ \lambda_1 \}),$
$(2 \to 3 \to 5 \to 4, \{\lambda_1\}),$
$(2 \to 3 \to 5 \to 6, \{\lambda_1\}),$
$(2 \to 4 \to 5 \to 7, \{\lambda_2\}),$
$(2 \to 4 \to 5 \to 8, \{\lambda_1, \lambda_2\})]$

⇨

$[(2 \to 3 \to 5 \to 4, \{\lambda_1\}),$
$(2 \to 3 \to 5 \to 6, \{\lambda_1\}),$
$(2 \to 4 \to 5 \to 7, \{\lambda_2\}),$
$(2 \to 4 \to 5 \to 8, \{\lambda_1, \lambda_2\}),$
$(2 \to 4 \to 8 \to 5, \{ \lambda_1 \})]$
$(2 \to 4 \to 8 \to 7, \{ \lambda_1 \})$

⇨

$(2 \to 3 \to 5 \to 6, \{\lambda_1\})$
$(2 \to 4 \to 5 \to 7, \{\lambda_2\}),$
$(2 \to 4 \to 5 \to 8, \{\lambda_1, \lambda_2\}),$
$(2 \to 4 \to 8 \to 5, \{ \lambda_1 \})$
$(2 \to 4 \to 8 \to 7, \{ \lambda_1 \})$

⇨

$(2 \to 4 \to 5 \to 7, \{\lambda_2\}),$
$(2 \to 4 \to 5 \to 8, \{\lambda_1, \lambda_2\}),$
$(2 \to 4 \to 8 \to 5, \{ \lambda_1 \})$
$(2 \to 4 \to 8 \to 7, \{ \lambda_1 \})$
$(2 \to 3 \to 5 \to 4 \to 8, \{\lambda_1\})$

Stop here

$n = (2 \to 3 \to 5 \to 6, \{\lambda_1\})$

The optimal lightpath

Table 3.1 The value change of ordered list *LP*

3.4 MILP Formulation

The physical topology of a optical network **G[N,E]** with |**N**|= ***n*** nodes and |**E**| = ***m*** edges is given. Each edge (***i, j***) ∈ ***E***, can accommodate a set ***K*** for channels, with |***K***| = ***k***. A set ***P*** of existing lightpaths in the network, with |***P*** | = ***p***. For the ligthpaths, which are already established in the network, we are given the following information

A = the edge-path incidence matrix (a_{ij}^{P}) with:

$$a_{ij}^{P} = \begin{cases} 1, & \text{the edge } i \rightarrow j \text{ has been used by the } p^{th} \text{ established lightpath.} \\ 0, & \text{otherwise.} \end{cases}$$

Kp = the channel assigned to the pth established lightpath.

We have defined two types of binary (0-1 integer) variables in the formulations. These are the path allocation variables (x_{ij}) and the channel allocation variables (ω_k). The binary variables are defined below.

$$x_{ij} = \begin{cases} 1, & \text{the edge } i \rightarrow j \text{ have been used in a lightpath.} \\ 0, & \text{otherwise.} \end{cases}$$

There is one path allocation variable associated with each edge in the network. Therefore, there are ***m*** such variables in our formulation

$$\omega_k = \begin{cases} 1, & \text{channel k is assigned to the new lightpath,} \\ 0, & \text{otherwise.} \end{cases}$$

There is one channel allocation variable associated with each available wavelength. So, there are ***k*** such variables in our formulation.

Objective function is:

$$\text{Minimize} \sum_{(i,j)\in E} x_{ij}$$

Subject to

$$\sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1 & \text{i is the source node (s)} \\ -1 & \text{i is the destination node (d)} \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

$$\sum_k \omega_k = 1 \qquad (2)$$

$$x_{ij} + \omega_{K_p} <= 1, \forall ij \ni a_{ij}^{p} = 1, \forall p \qquad (3)$$

The three situations for the constrain need to be considered (refer Equation 1):

Case 1 (i = s): In this case, constraint (1) can be written as

$$\sum_j x_{sj} - \sum_j x_{js} = 1$$

The above equation states that there is one outgoing edge (s , j) ∈ E, such that $x_{sj} = 1$ and this is the first edge in the physical route (from *s* to ***d***) for a lightpath. None of the incoming edges to node i = *s* is on the physical route associated with the lightpath.

Case 2 (i = ***d***): In this case, constraint (1) can be written as

$$\sum_j x_{dj} - \sum_j x_{jd} = -1$$

The above equation implies that there is one incoming edge (j, ***d***) ∈ E, such that $x_{jd} = 1$ and this is the last edge in the physical route (from ***s*** to ***d***) for a lightpath. None of the outgoing edges to node i = ***d*** is on the physical route associated with the lightpath.

Case 3 (i ≠ ***s***, i ≠ ***d***): In this case, constraint (1) can be written as

$$\sum_j x_{ij} - \sum_j x_{ji} = 0$$

This equation holds for all other nodes in the network. In this case, if *i* is an intermediate node in a lightpath from ***s*** to ***d***, there is exactly one incoming edge to node *i* and one outgoing edge from node *i* which are on the physical route for the lightpath. Otherwise, none of the incoming or outgoing edges of node *i* are on the physical route of the lightpath.

Constraint (2) enforces the wavelength continuity constraint for the lightpath. Since the channel assignment variables are binary variables, $\sum_k \omega_k = 1$ means that just one wavelength can be chosen for each lightpath. This ensures that a lightpath is assigned exactly one channel.

Constraint (3) ensures that if the new lightpath shares an edge (i,j) with the pth established lightpath, then it is never assigned the same channel K_p as the pth lightpath.

3.5 Tabu Search Approach

Tabu search is one of the most popular meta-heuristic methods widely utilized in many areas. In this investigation, we are trying to develop a tabu search approach to find the optimal lightpath in a WDM network. For a given source-destination pair, we try to find an initial lightpath from the source *s* to the destination ***d*** by using a depth-first search. As mentioned in section 2.7, there are two types of blind search: the breadth-first search and the depth-first search. Using the breadth-first search we can always find the shortest lightpath, if at least one lightpath from *s* to ***d*** exists. But by using the depth-first search, when we find the first path from the source *s* to the destination ***d***, the path is not necessarily the shortest one. Our approach is to start with some initial lightpath obtained using the depth-first search, and then optimize the lightpath by using the tabu search.
Our algorithm using tabu search approach is given below.

Given the physical network, our heuristic finds a near-optimal lightpath from the source node *s* to the destination node ***d***. In the heuristic, at all points, we have a subgraph S of the graph G representing the network where the subgraph always includes nodes *s* and ***d***. As the heuristic proceeds, two types of situations may arise. In the first situation (the preferred situation), the subgraph consists of the nodes and edges defining a path *P* from

s to ***d*** (Fig 3.3 a). The path *P* is such that there is at least one wavelength λ_r, $1 \le r \le k$, such that wavelength λ_r has not been allotted to any existing communication that uses any of the edges used in the path *P*.

In the second situation (Fig 3.3 b), the subgraph consists of two parts:

- The nodes and edges constituting a tree T_1 with root ***s***. The number of leaves in T_1 is restricted by L_k. For example, if L_k is 5, the number of leaves in T_1 can be at most 5.
- The nodes and edges constituting a path P_2 from some node ***q*** to destination node ***d***

There is no node that appears in both parts.

Let L_{set} be the set of leaf nodes of the tree T_1. There is at least one lightpath that can be used from the source node ***s*** to each node in L_{set} at any time. The path P_2 is such that there is at least one wavelength λ_r, $1 \le r \le k$ such that wavelength λ_r has not been allotted to any existing communication that uses any of the edges used in the path P_2. To use the heuristic, we need a heuristic function to evaluate the fitness of the current subgraph S. The heuristic adds an edge to the subgraph S or deletes one or more edges from the subgraph. When we add or delete an edge, a tabu tenure is associated with the edge added or deleted so that, until the tabu tenure is exceeded, that operation will not be undone. The tabu tenure must also be specified for the heuristic to work.

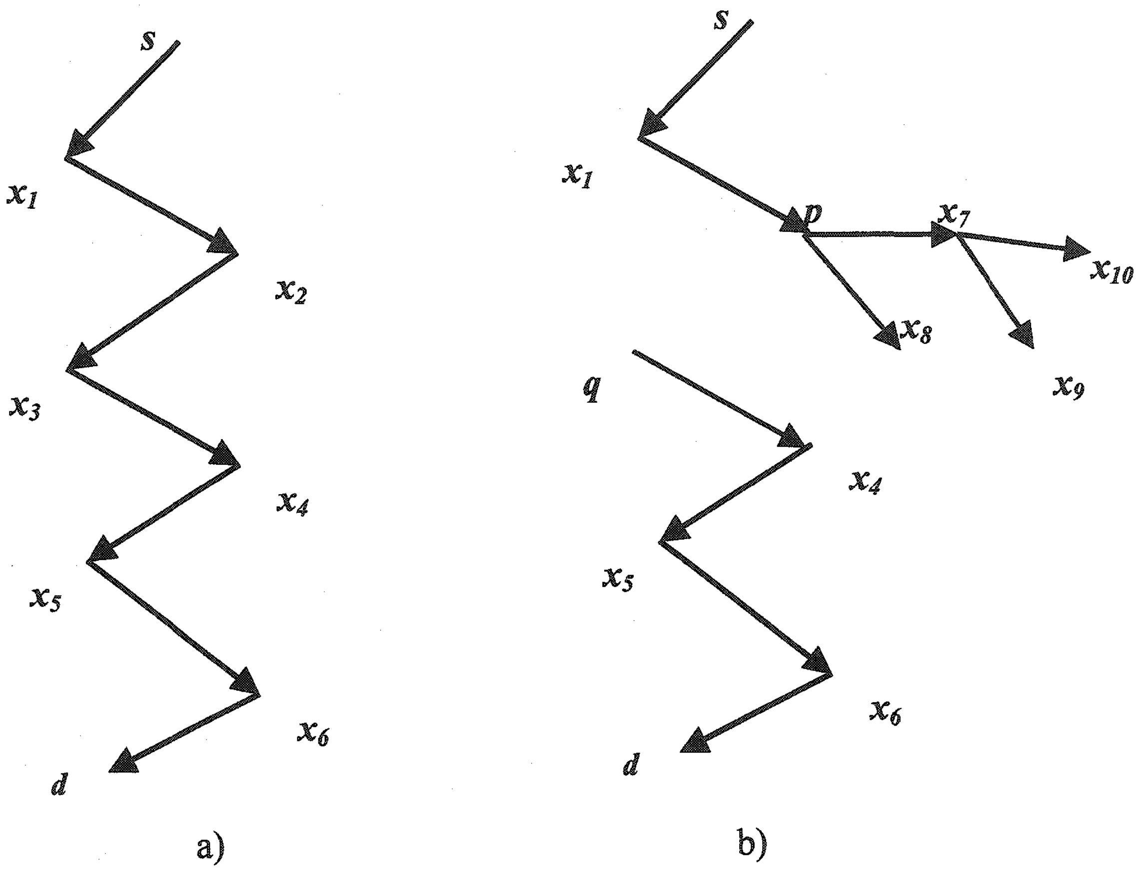


Figure 3.3 The Two Situations

We now describe the moves in our heuristic. The current subgraph S may be either in situation 1 or in situation 2. After the operation, the new subgraph S_{new} will also be either in situation 1 or in situation 2. There are three possible moves for our heuristic. Our heuristic function is such that the heuristic value for a subgraph S in situation 1 is always less than that for a subgraph S in situation 2. The idea is that we wish to remain, if possible, in situation 1 because that represents a single feasible path from the source *s* to the destination ***d*** that we may use to define a lightpath.

The steps of our heuristic algorithm are as follows:

Step 1: Find, if possible, using depth-first search, an initial lightpath using some wavelength λ_r, $1 \leq r \leq k$ and a path $s(=x_0) \rightarrow x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_r \rightarrow d(=x_{r+1})$ for the given network.

Step 2: If the initial lightpath does not exist, report failure and stop.

Otherwise, the subgraph S consists of the graph with the nodes s, x_1, x_2, ..., x_r, d and the edges $s \rightarrow x_1$, $x_1 \rightarrow x_2$, ..., $x_r \rightarrow d$.

Step 3: The best lightpath found so far = the initial lightpath found in step 1.

Step 4: while (more subgraphs should be explored) repeat step 5 - 6.

Step 5: Apply the best move, using the restrictions of tabu search, to the current subgraph S creating a new subgraph S_{new} which becomes the current subgraph for the next iteration.

Step 6: If the new subgraph S_{new} is a graph in situation 1 and is better than the best lightpath found so far, use S_{new} and one of the wavelengths to go from s to d in S_{new} to redefine the best lightpath found so far.

An Example:

For example, in the network shown in Fig 3.4 (a), we want to find optimal lightpath from node 2 to node 6.

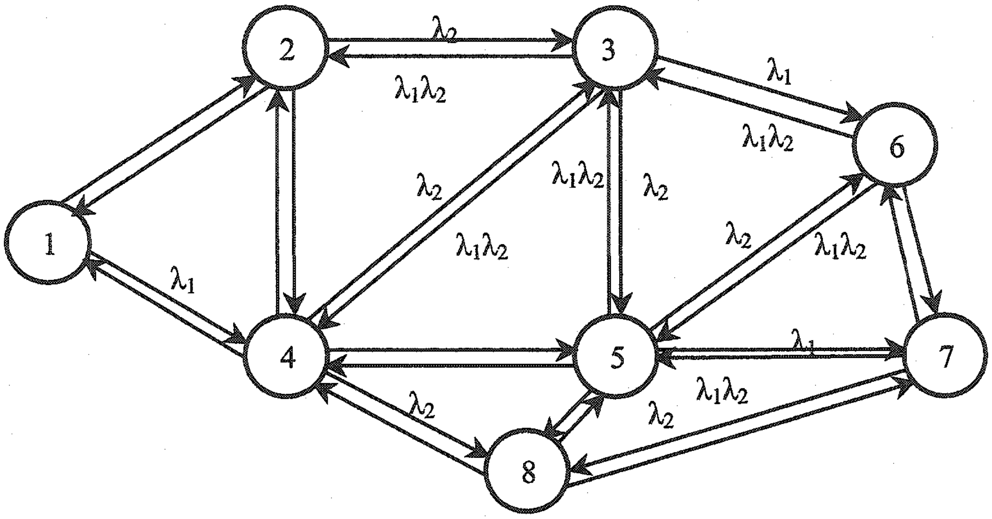


Figure 3.4 (a) An Example of Tabu Search Approach

Suppose by using depth-first search, we found an initial lightpath, using path 2→4→8→7 →6 and wavelength λ_1 as shown in Figure 3.4 (b). The current best lightpath is the initial lightpath. Since it is a single lightpath, we are in situation 1 now. S consists of the initial lightpath. The initial lightpath is also the best lightpath found so far.

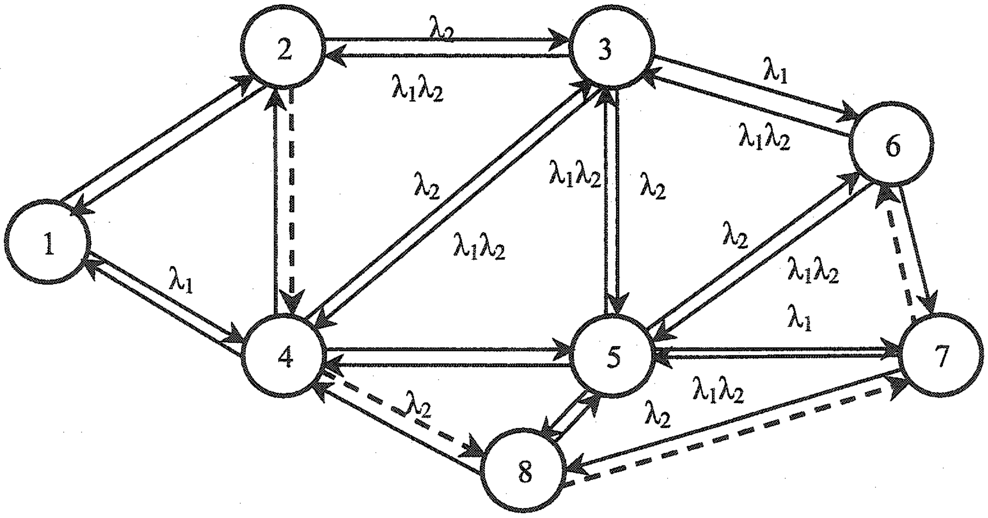


Figure 3.4 (b) An Example of Tabu Search Approach

In order to get a better solution, we want to move from situation 1 to situation 2. Therefore, we need to delete one of the edges on the initial lightpath. Suppose, after we calculate the heuristic value, we found that the edge 4→8 is the best edge to delete. Therefore, we delete it as shown in Figure 3.4 (c). Now we got a new S which consists of two parts: 2→4 and 8→7→6.

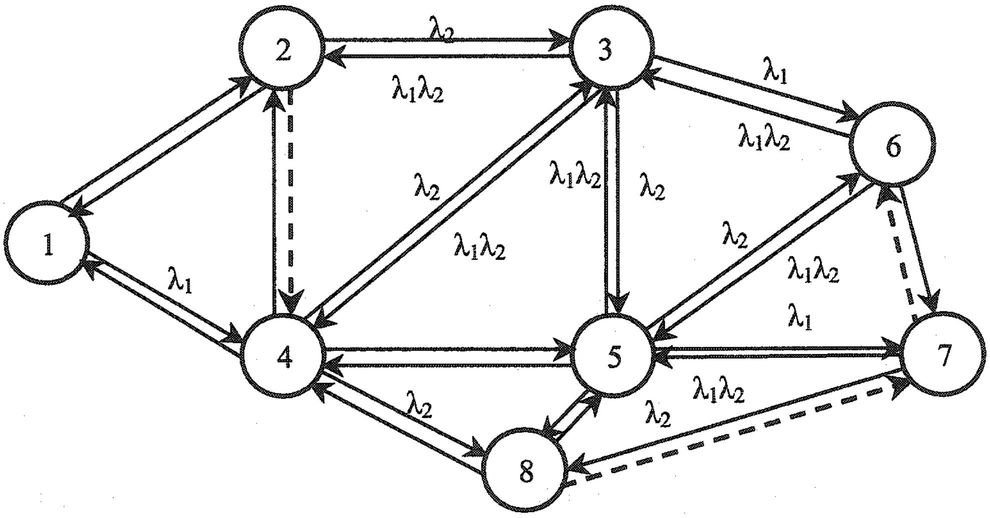


Figure 3.4 (c) An Example of Tabu Search Approach

We have many choices to do next. For example we can add edge 4→1,4→3 or 4→5. But the edge 4→8 have just been deleted, it is in the tabu list, so it cannot be added in recent iterations. Suppose after calculation, the edge 4→5 has the best heuristic value. We apply this move and get the new *S* which is shown in Figure 3.4 (d).

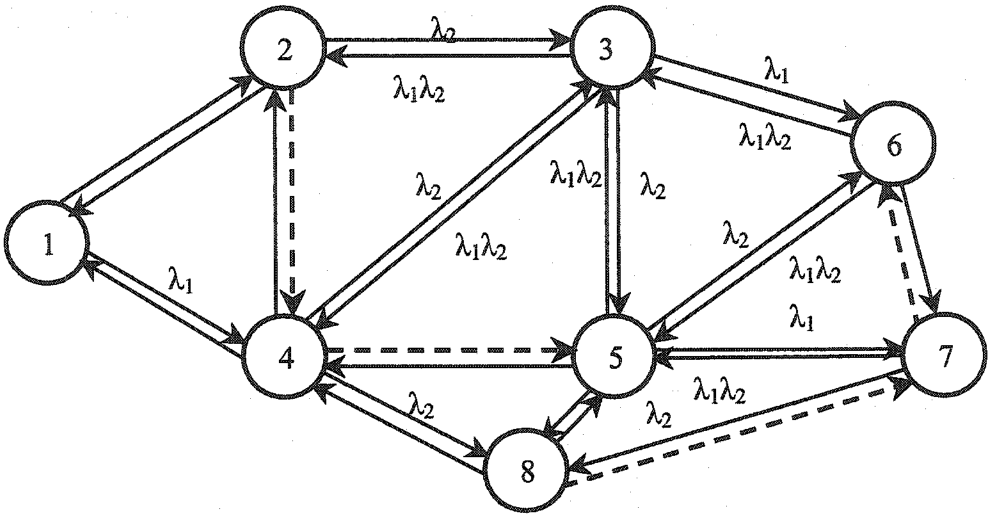


Figure 3.4 (d) An Example of Tabu Search Approach

Suppose on the next stage, the best move is to add edge 5→6 (Figure 3.4 (e)). Since node 6 is the destination node, we get back to situation 1, and get another lightpath 2 →4→5→6, wavelength λ_1. Since this lightpath is shorter than the current best lightpath, we assign this lightpath as the current best lightpath Fig 3.4 (f). We can repeat this process, until we find the best solution or the solution which we think is good enough.

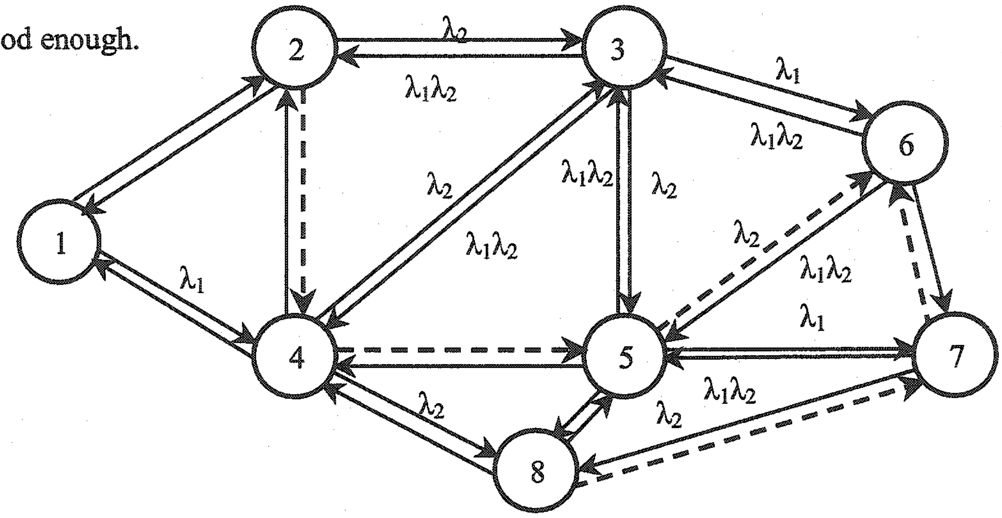


Figure 3.4 (e) An Example of Tabu Search Approach

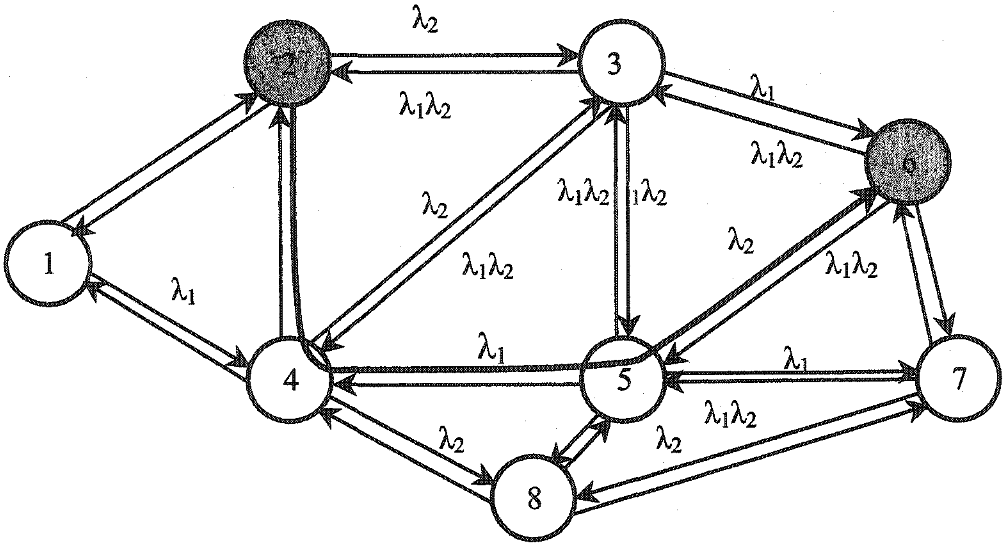


Figure 3.4(f) An Example of Tabu Search Approach

We will next discuss how to apply a move to the existing subgraph S to create a new subgraph S_{new}. We have explained that the subgraph S can be either in situation 1 or in situation 2. In each of these situations, a move is either to add a new edge (and possibly a new node) from the graph **G** – **S** to subgraph S or to delete an existing edge (and, if necessary, a node) of the subgraph S. Thus we have three possible moves as described below.

Type 1 move (delete an edge $x \rightarrow y$ when subgraph S is in situation 1)

This move creates a subgraph S_{new} which is in situation 2 (Fig 3.5).

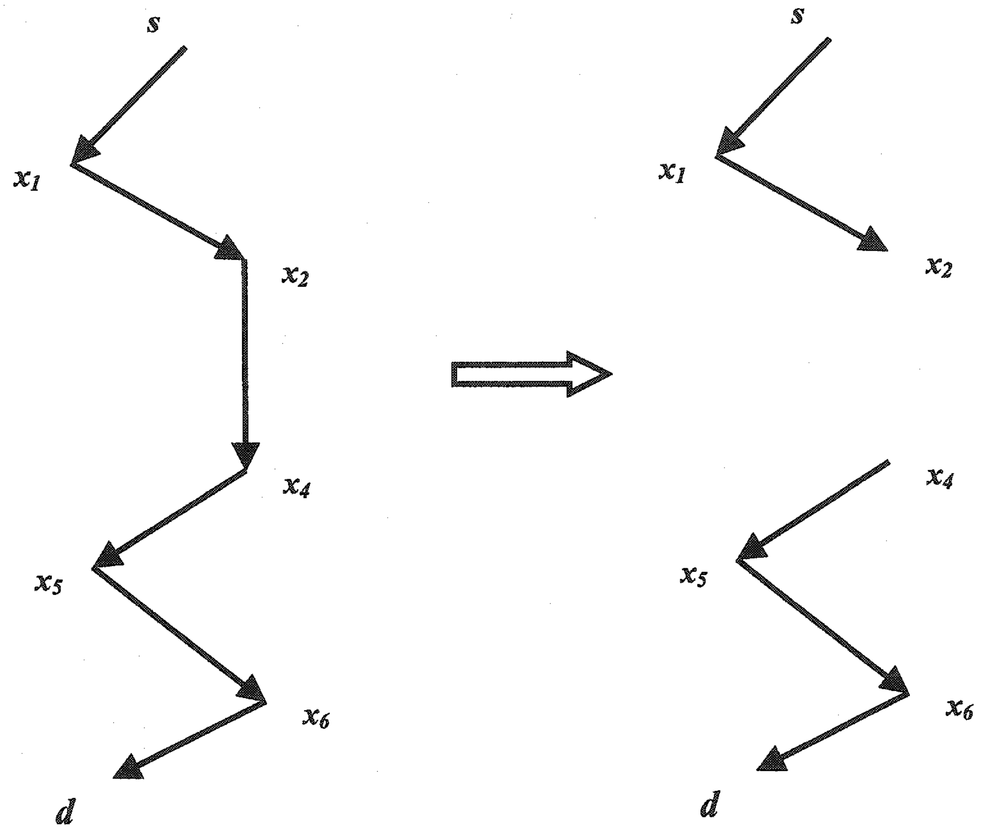


Figure 3.5 Type 1 Move

Type 2 move (add an edge $x \rightarrow y$ when subgraph S in situation 1)

The move works in two phases. In the first phase, we create a subgraph which is neither in situation 1 nor in situation 2. In the second phase we start again from S, delete some of the edges and nodes from S then add the edge $x \rightarrow y$ to get a subgraph S_{new} in situation 1.

The node y must be one of the existing nodes in the path from x to *d*. In the first phase, after the addition of the edge $x \rightarrow y$, there are two paths from node x to node y, one using the new edge x → y that we just added, the other using the path from x to y that originally existed in subgraph S. Since the length of the path from x to y in the subgraph S is greater than 1, if we use the new edge x → y, to form a new path from the source node *s* to the destination node ***d***, we get a path that is shorter than the path from ***s*** to ***d*** in the subgraph S. This is potentially a highly desirable move because the path through the new edge is shorter than the original path in subgraph S. Our intent is to remove, in phase 2, the edges and nodes in the original path in subgraph S so that we get a subgraph S_{new} in situation 1.

However, phase 1 is valid only if there is at least one wavelength λ_r, $1 \leq r \leq k$, such that wavelength λ_r has not been allotted to any existing communication that uses any of the edges used in the path from *s* to ***d*** using the new edge x → y. In other words there must be at least one wavelength λ_r that

i) appears in the set of wavelengths that we may use to go from source node *s* to node x,

ii) appears in the set of wavelengths that we may use to go from node y to destination node ***d***,

iii) does not appear in $L_{x,y}$, the set of wavelengths that are already in use for existing communication that uses the edge x → y.

If these conditions are satisfied, we move into phase 2 where we form the new subgraph S_{new} (which is also in situation 1) from S by deleting the existing edges in the path from x to y and then adding the edge x → y (Fig 3.6).

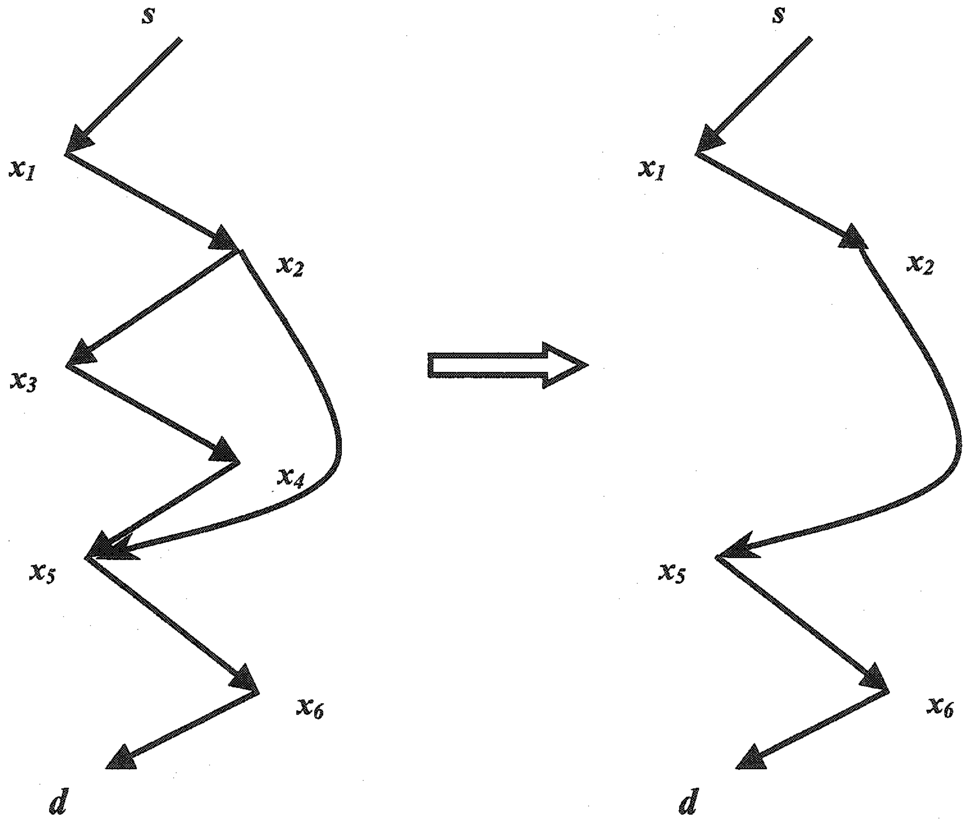


Figure 3.6 Type 2 Move

<u>Type 3 move (add an edge x → y when subgraph S is in situation 2)</u>

The move works in one or in two phases, depending on circumstances. In the first phase, we create a subgraph which is either in situation 2 or a subgraph which is neither in situation 1 nor in situation 2. If the new subgraph is in situation 2 and the number of leaves of T_1 is not more than L_{set} (in Figure 3.7, the value of L_{set} is assumed to be 2), the second phase is not needed. Otherwise, we enter phase 2, delete some of the edges and nodes from the subgraph we created in phase 1 to get a subgraph S_{new} either in situation 1 or stay in situation 2.

This move is possible only if x is a node in the tree T_1. There are three cases:

a) The node y is one of the existing nodes in the path P_2 (refer to Figure3.7 a). In this case, after adding the edge x → y to T_1, the new subgraph is neither in situation 1 nor in situation 2. In phase 2, we delete some nodes and edges to get a new subgraph S_{new} in situation 1.

b) The node y is not one of the existing nodes in the path P_2, the current number of leaves in T_1 is either

- Less than L_{set} or
- Equals L_{set} but the added edge x → y will not increase the number of leaves of the tree (refer to Figure 3.7 b).

In this case the move just works in one phase since we are just extending the tree T_1 by adding a new edge x → y . The new subgraph S_{new} is in situation 2.

c) The node y is not one of the existing nodes in the path P_2, the current number of leaves in T_1 equals to L_{set} and the number of leaves in T_1 will be increased by adding the edge x → y (Figure3.7 c).

In the first phase, we will expand the T_1 by adding the new edge x → y and node (edge $x_8 \rightarrow x_9$ and node x_9 in Figure3.7 c) to it, and delete some edges and edges ($x_8 \rightarrow x_{10}$ and node x_{10} in Figure3.7 c) in the phase 2 to get a new subgraph S_{new} in situation 2.

However, the move is valid only if the wavelength constraint is valid. To check this, our procedure depends on whether the new subgraph is in situation 2 or not.

There are two possibilities to consider:

Possibility 1: Node y is one of the existing nodes in the path P_2, the new subgraph is neither in situation 1 nor in situation 2 (Figure 3.7 a). In this case there must be at least one wavelength λ_r that

a) appears in the set of wavelengths that we may use to go from source node *s* to node x,

b) appears in the set of wavelengths that we may use to go from node y to destination node ***d***,

c) does not appear in $L_{x,y}$, the set of wavelengths that are already in use for existing communication that uses the edge x → y.

Possibility 2: Otherwise, the situation is as shown in Fig 3.7 b, c. In this case, there must be at least one wavelength λ_r that

a) appears in the set of wavelengths that we may use to go from source node *s* to node x,

b) does not appear in $L_{x,y}$, the set of wavelengths that are already in use for existing communication that uses the edge x → y.

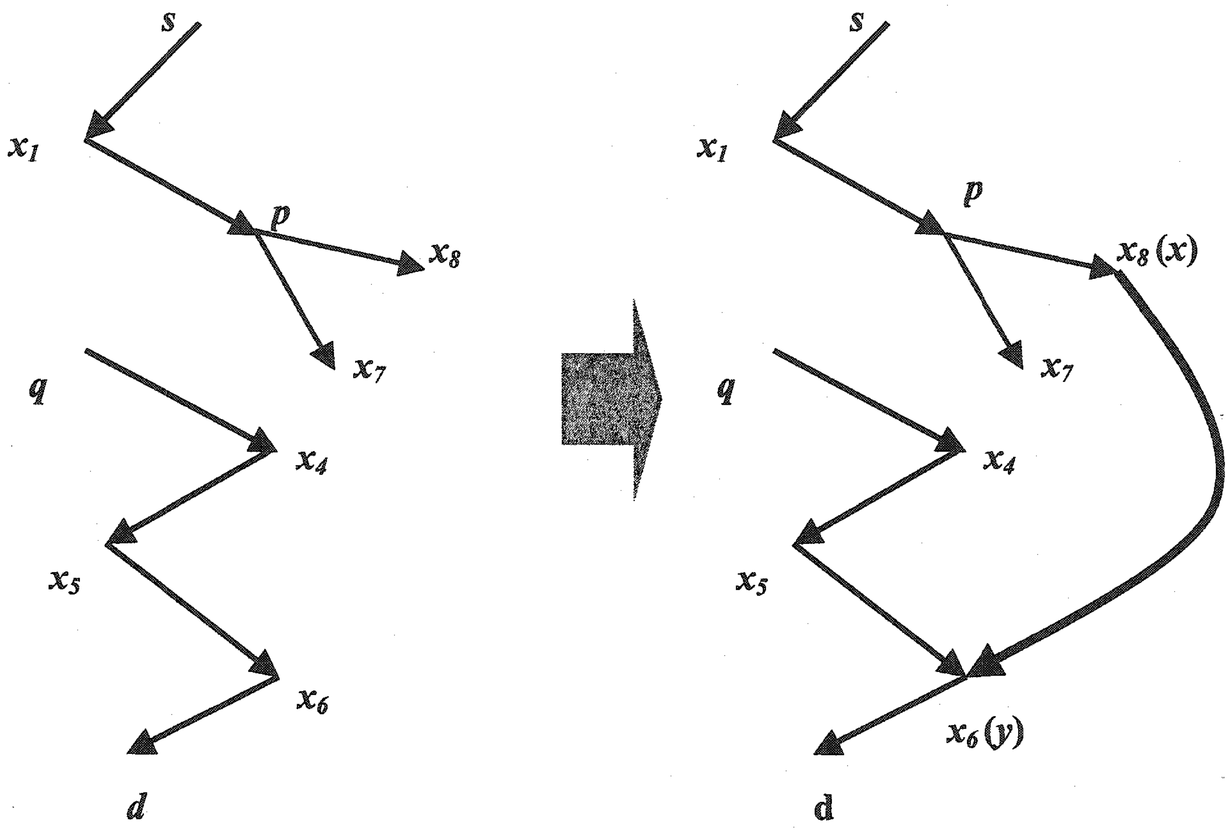


Fig 3.7(a) Type 3 Move (node y is in P2)

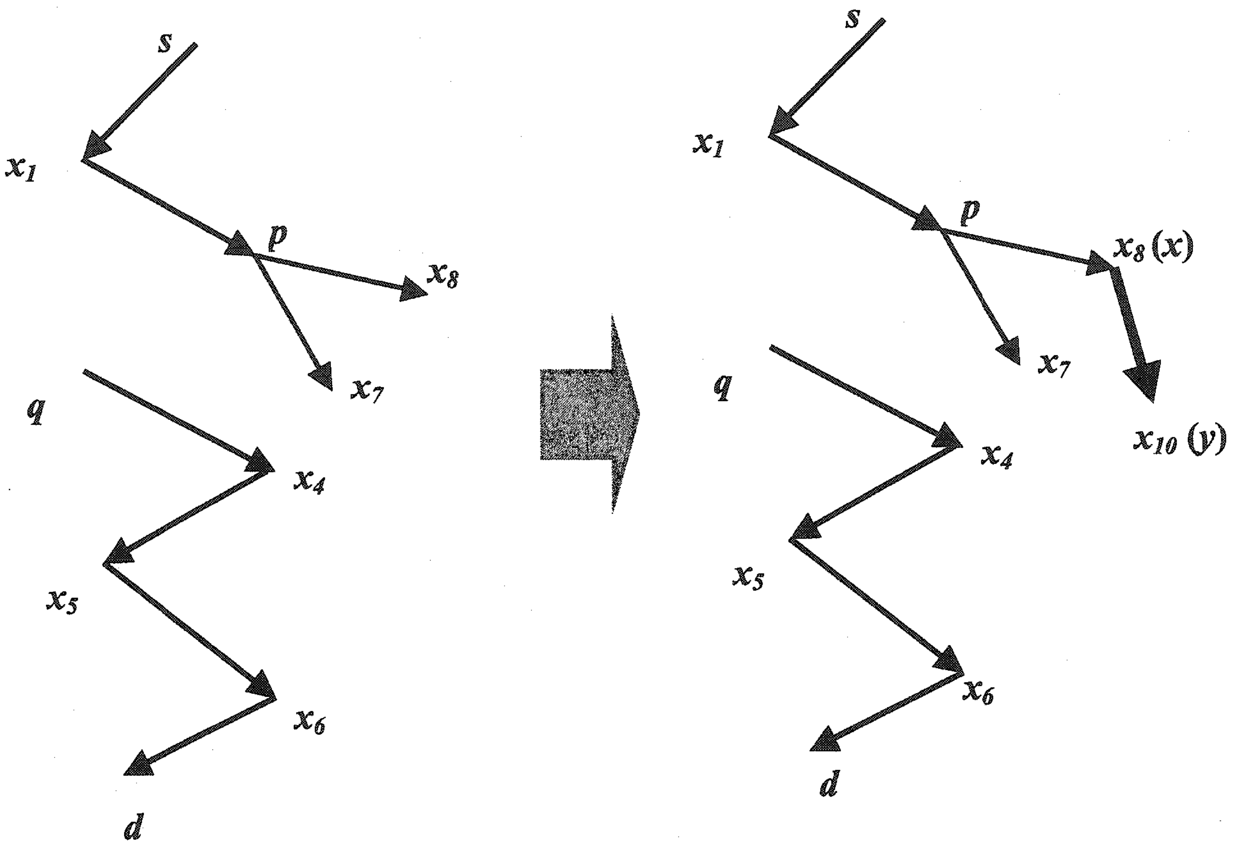


Fig 3.7(b) Type 3 Move (node y is not in P2)

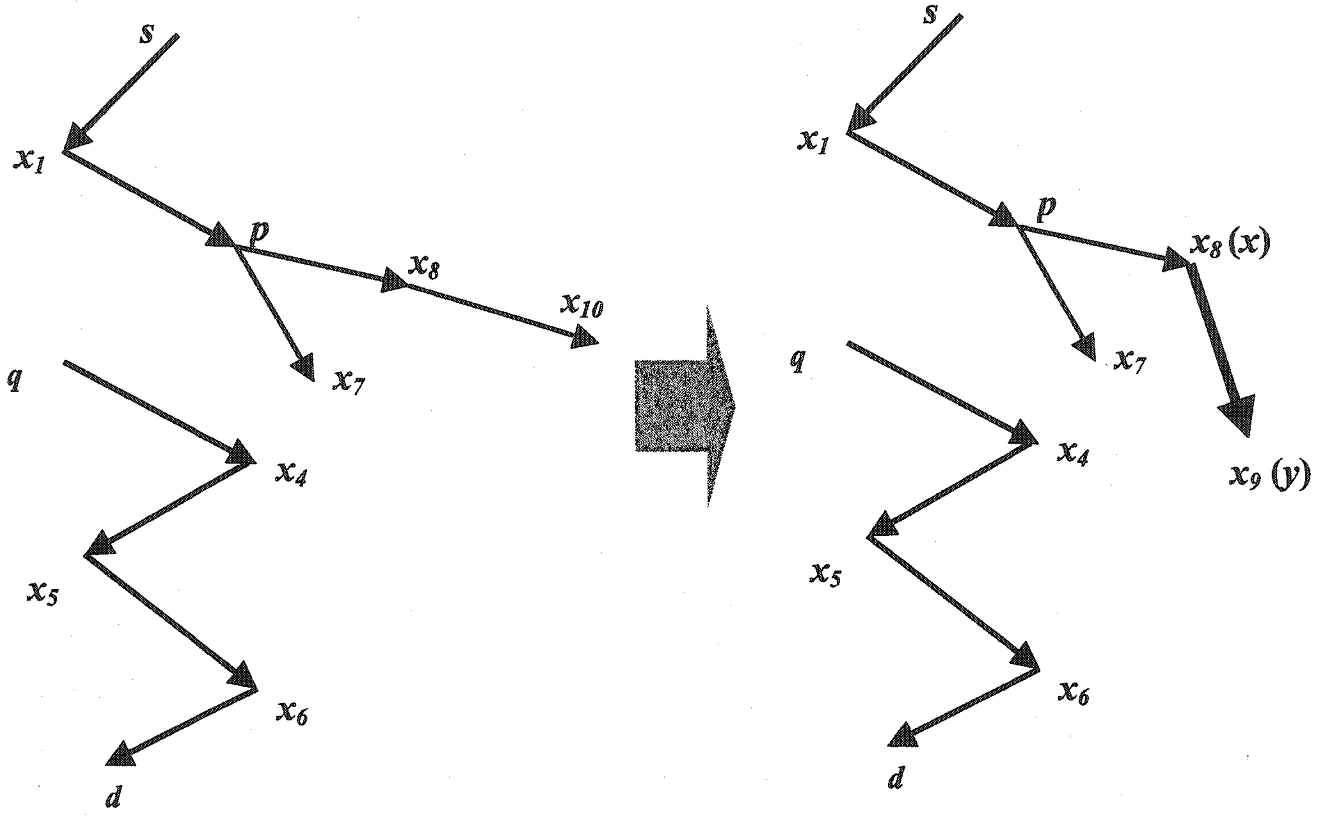


Fig 3.7(c) Type 3 Move (node y is not in P2)

Chapter 4 Implementation and Experiments

In this chapter, we will describe in detail how we implemented the tabu search heuristic approach. The goal of this heuristic is to establish a near-optimal lightpath for a specified source-destination pair in a given WDM network. We have already described the algorithm used in this approach in Chapter 3. Here we will introduce the implementation of the algorithm.

The process of tabu search approach is shown as in Figure 4.1. In order to find the optimal or near-optimal lightpath by using tabu search, first of all, we need to know the information such as the network topology and the existing lightpaths used for communication. We used a data file to keep the information and we read this data file first before proceeding with the tabu search. Next we try to find an initial path from the source node *s* to the destination node *d* by using depth-first search. If we cannot find a lightpath by using depth-first search, it means that it is impossible to establish a lightpath for this source-destination pair in the network in the current situation.

The most important part of this approach to optimize this initial lightpath using tabu search. We have described the three types of moves used in our approach. Suppose we have already got an initial lightpath. The current subgraph S is consisted of the initial lightpath (situation 1), so we try to apply the type 2 move first. If a valid type 2 move exists, the new subgraph will still be in situation 1 after we apply the type 2 move to S. If we are currently in situation 1 and there is no valid type 2 move, then we can move from the situation 1 to situation 2 by applying a type 1 move. After we apply a type 1 move, we are in situation 2, and then we can apply a type 3 move. After some iterations, it is possible for subgraph to revert back to situation 1. By repeating this process (shown in Figure 4.2), we can get the optimal or near-optimal lightpath for a given source and destination pair.

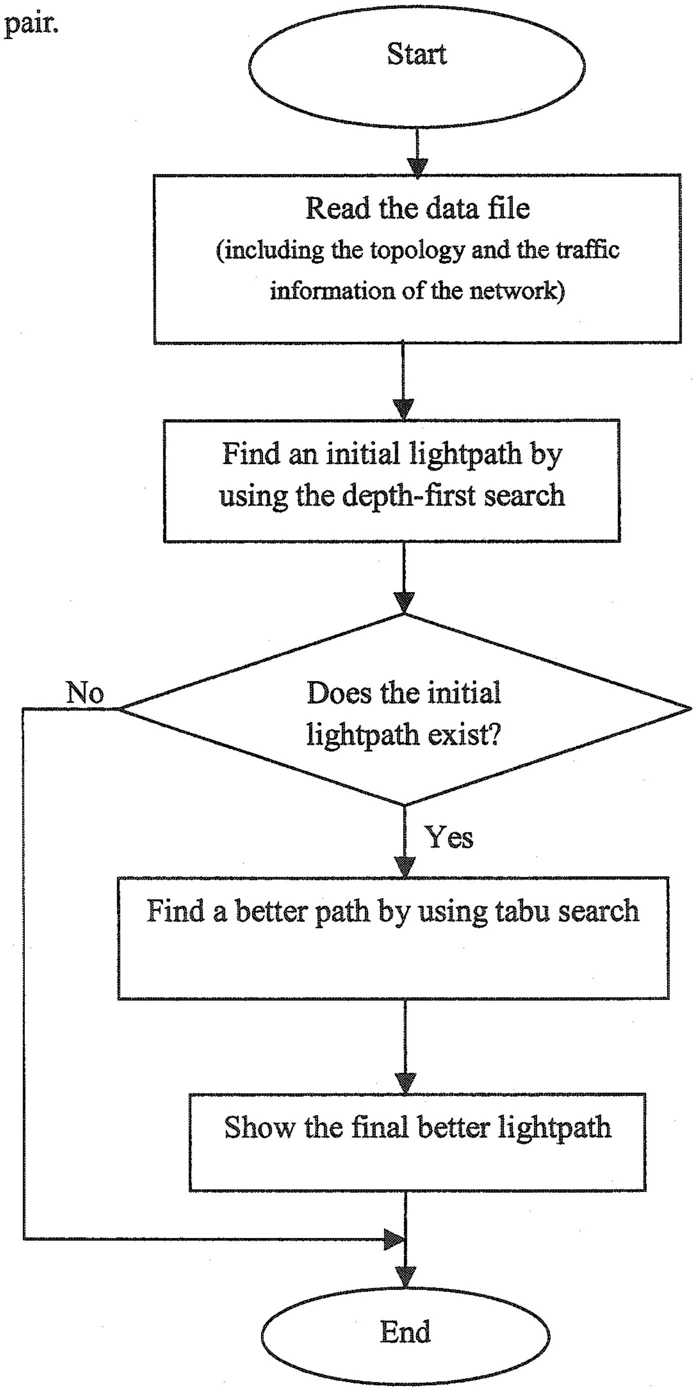


Figure 4.1 The Tabu Search Approach

In the remainder of this chapter, we will discuss in detail the data file, the heuristic functions, how to find the initial path, and some important parameters utilized in the tabu search approach.

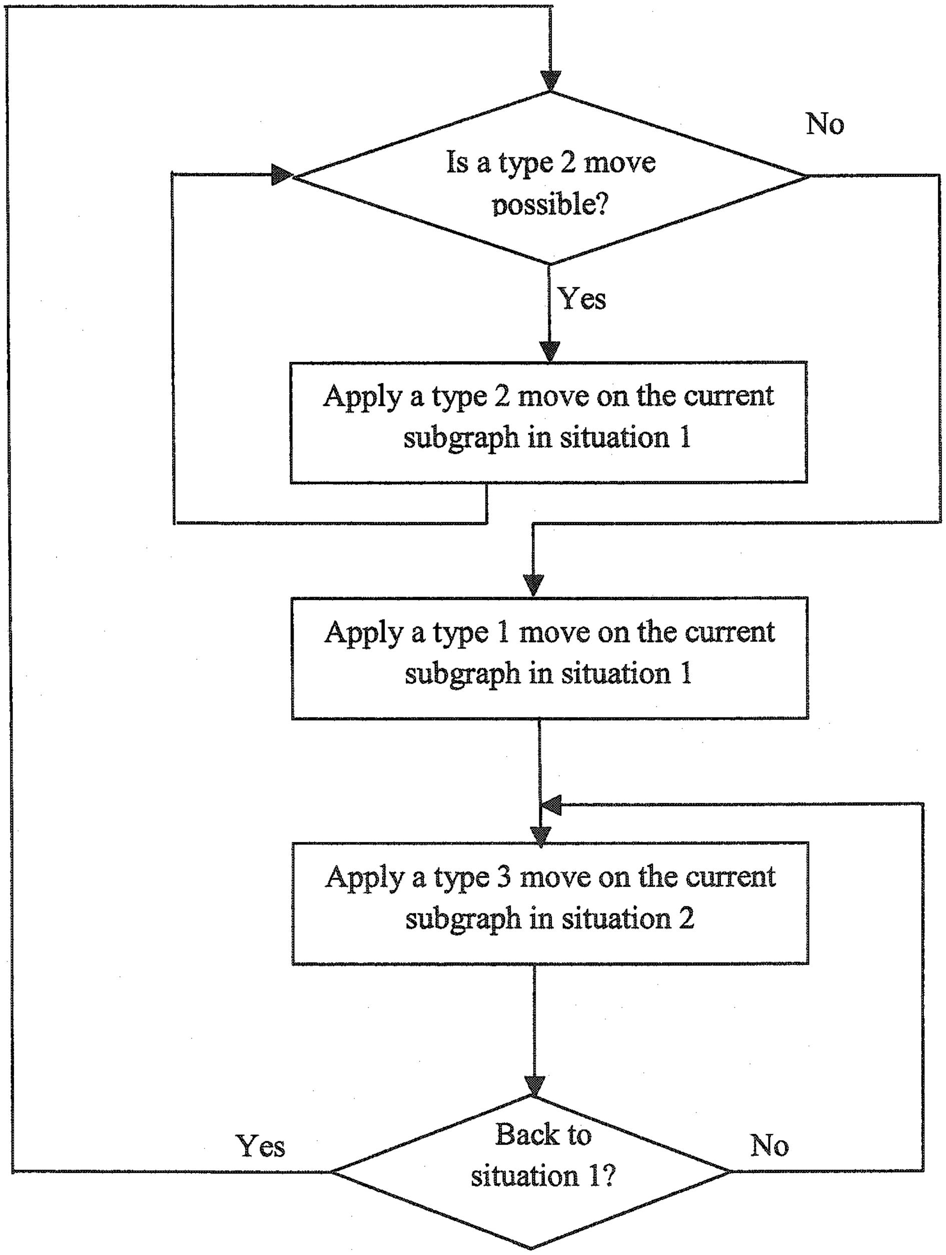


Figure 4.2 The Application of the Three Type of Moves

4.1 Tabu List

We put the moves which are in their tabu status in a tabu list. That means a move in the tabu list cannot be applied in current iteration. Tabu tenure gives the number of iterations for which the move cannot be applied. We maintain the tabu list for each iteration and, if the tabu tenure of some move expires, we delete the move from tabu list, so we can apply the move again. We check the tabu list before we apply a move, if it is in the tabu list and it does not satisfy the aspiration criteria, we cannot apply it.

In our approach, the elements in tabu list have four pieces of information: from, to, type and end-iteration.

- From: the start node of an edge.
- To: the end node of the edge.
- Type: either "add" or "delete".
- End-iteration: the current iteration + tabu tenure.

From and to are the starting and the end node of the edge, respectively. Type means which type of move is forbidden: add the edge or delete the edge. In our approach, there is a counter which counts the iterations. The current iteration is the value of this counter $\mathbf{C}_{\mathbf{counter}}$. The end-iteration means when tabu tenure of this move is expired (when the move can be applied).

For example, we just deleted an edge $x_5 \rightarrow x_6$. We must move this edge to the tabu list. Suppose the tabu tenure we used is 3, and the value of the counter is 258. So the start node is x_5, end node is x_6, type is "add" which means we cannot add the edge for the duration of the tabu tenure. The end-iteration is 258 + 3 = 261 (which means we can not add this edge until $\mathbf{C}_{\mathbf{counter}}$ equals 261).

4.2 Candidate List

In each iteration, we store the moves we can use in subgraph S in a candidate list. The most important items in the candidate list are: the start node, the end node, the length from the source node *s*, the wavelength and the heuristic value of this node.

We will compare the heuristic value of each move in the candidate list, and choose the move which has the best heuristic value as the current best move. For example, in Figure 4.6(b), we add the move add $p \rightarrow x_7$, add $x_8 \rightarrow x_{10}$ and add $x_8 \rightarrow x_6$ into candidate list. And we find that add $x_8 \rightarrow x_6$ has the best heuristic value, we set it as the current best move. If the current best move is either not in tabu status or although in tabu status but it is satisfied aspiration criteria, we will remove the move from the candidate list and put it into tabu list. And then by applying this move, we can get a new subgraph $\mathbf{S}_{\mathbf{new}}$.

4.3 Aspiration criteria

Normally, we cannot apply the moves which have been marked as tabu. But in some case, we want to apply the moves even though they are in tabu status. Such as, sometimes, by applying the moves in tabu status we can get the better solution than ever. The aspiration criteria which we utilized in this approach is: if applying a move can get a solution better than the current best solution, we will use this move even though it is in tabu status.

For example, in Figure 4.6 (b), if the move add $x_8 \rightarrow x_6$ is marked as tabu, since we apply this move can get a better solution (the length is 5) than the current solution (the length of current best lightpath is 7). We can use the aspiration criteria and apply this move.

4.4 The Data File

In order to solve the problem, first of all, we need to know the necessary information about the given network. We created a data file which stores the network information. The data file contains the following information:

- Network topology information -- the number of the nodes, the number of edges and the number of channels in each fiber.
- The network traffic information -- the wavelengths which are already in use, for each fiber in the current network.
- The source node and destination node for the new request for communication.

We will read the data in this data file before we solve the problem.

4.5 Find the Initial Path

We used depth-first search to find the initial lightpath. In this algorithm, we will use $L_{i,j}$ to denote the set of wavelengths already used on the edge $i \rightarrow j$. We will also use ***LP*** to denote an ordered list of pairs (***P***, Λ) where ***P*** denotes a path $s \rightarrow x_1 \rightarrow x_2 \rightarrow \dots x_p$ from the source node *s* to some node x_p in the network and Λ denotes a non-empty set of wavelengths. This means that we may use the path $P = s \rightarrow x_1 \rightarrow x_2 \rightarrow \dots x_p$ to establish a lightpath from source node *s* to some node x_p using any of the wavelengths in Λ. We will use lastnode(***P***) to denote the last node x_p in such a path. The steps of the algorithm are given below:

1. Set ***LP*** to be a list consisting of one element - the pair (*s*, $\{\lambda_1, \lambda_2, \dots \lambda_k\}$) where *s* is the source node and $\lambda_1, \lambda_2, \dots \lambda_k$ are the wavelengths of the *k* channels available on any fiber in the network.
2. If ***LP*** is empty, then exit and signal failure.
3. Let *n* = (***P***, Λ) be the first element in the ordered list ***LP***, where ***P*** is the path $s \rightarrow \dots \rightarrow x$, Remove *n* from ***LP***.
4. If *x* is the destination node, then exit and signal success.
5. If *x* is a leaf node, Remove *n* from ***LP*** Go to step 3.
6. Otherwise, if $x \rightarrow y$ is a directed edge in the network and y does not appear in path ***P***.
7. If $\Lambda - L_{x,y}$ is empty, Remove *n* from ***LP*** Go to step 3.

8. If $\Lambda - L_{x,y}$ is not empty, check if there is any other element (P', Λ') in the ordered list ***LP***, such that,
    a. Lastnode(P') = y
    b. $\Lambda' \supseteq \Lambda - L_{x,y}$
9. If $\Lambda - L_{x,y}$ is not empty and an element (P', Λ') satisfying the conditions of step 6 cannot be found, create a new path $P^{new} = s \rightarrow \ldots \rightarrow x \rightarrow y$ and insert the pair (P^{new}, $\Lambda - L_{x,y}$) as the first element of ***LP***.
10. Go to step 3.

4.6 Choose the Current Best Move

The most important part for this approach is to optimize the current best lightpath by tabu search. In our approach, we can apply the three types of moves as shown in Figure 4.2. But how can we choose the best move? The main process is shown as Figure 4.3

In each iteration of our approach, by using the heuristic function, we can calculate the heuristic value of each move, and the move which has the best heuristic value is the current best move. If the current best move is not in tabu list, we can apply this move directly. But if it is marked as having tabu status, we still can apply it if it satisfies the aspiration criteria. Otherwise, we will choose another current best move.

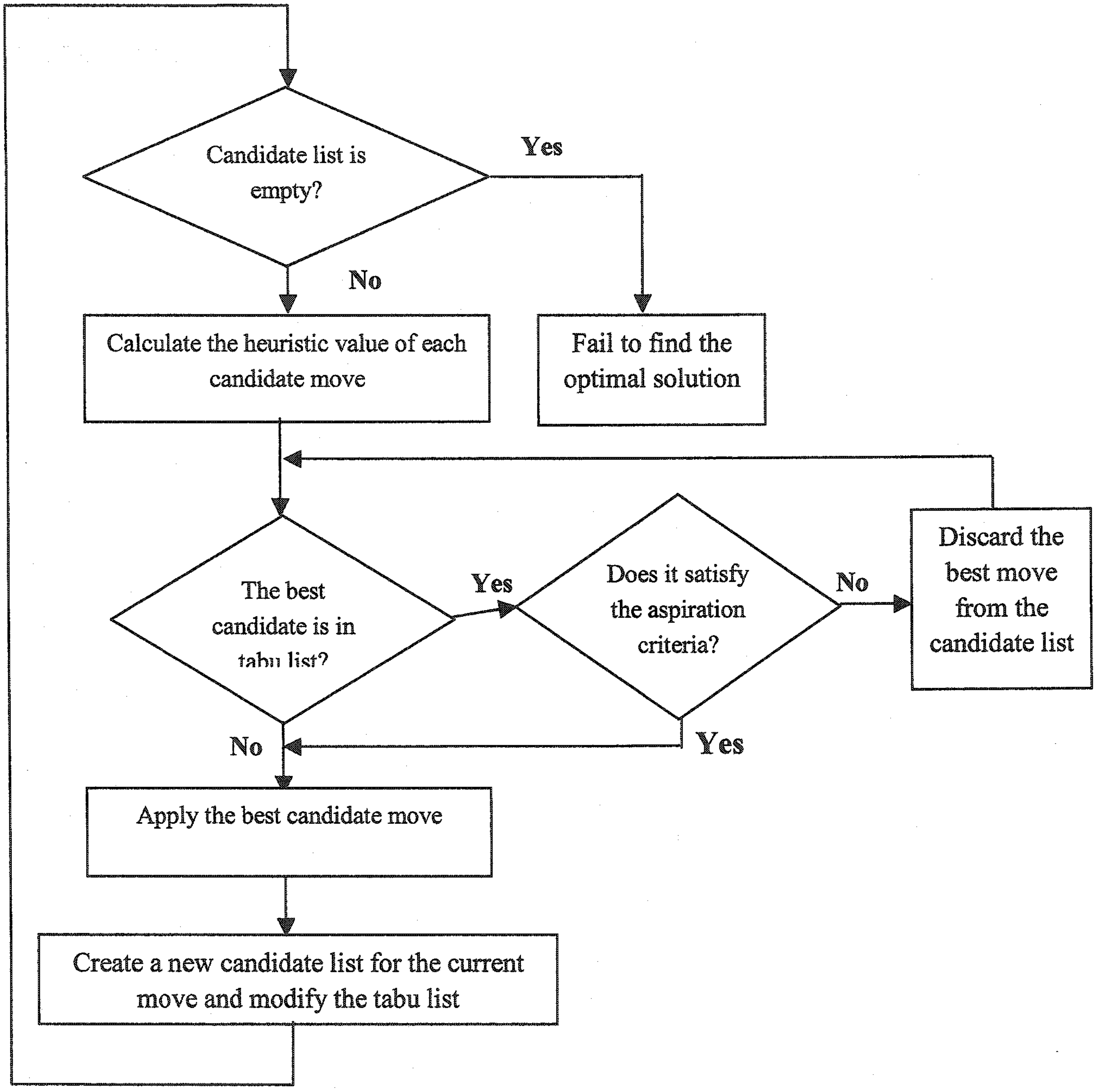


Figure 4.3 How to Choose the Best Move

4.7 The Heuristic Function

Heuristic functions are very important in the heuristic approaches. They are used to evaluate the candidate moves. We have described the three types of moves in our approach in Chapter 3. And we have three heuristic functions in this approach. In our approach, the move with the smallest heuristic value is the current best move.

4.7.1 Heuristic Function for Type 1 Move

In order to move from situation 1 to situation 2 (Type 1 move), we need to select the most promising edge in the lightpath, to be the deleted edge. We need a heuristic function (***H1***), which is used to check the heuristic values of each node of the initial lightpath. The heuristic function we have used here is:

$$\boldsymbol{H1} = V1 * M1 + V2 * M2 + V3 * M3$$

M1, M2 and M3 are three constants. And there are three parameters in this heuristic function:

- V1 = The length from the source node to the current node in the current lightpath
- V2 = The number of wavelengths lost up to current node
- V3 = The total number of nodes in the network – the out degree of the current node

There is an example of the use of heuristic function of type 1 move. Suppose there are four wavelength λ_1, λ_2, λ_3 and λ_4, and 20 nodes in the network shown as in Figure 4.4. Suppose the solid line path is a lightpath from source node *s* to destination node ***d***. The dotted arrow in Figure 4.4 (a) are the edges which start from the nodes in the lightpath. They are in the network but not in the current subgraph. In order to move from situation 1 to situation 2, we need to choose an edge to delete. But which is the best edge to delete? We need to use the heuristic function ***H1*** to find it.

For example, we assumed M1 = M2 = M3 = 1 here, for the node *s*, V1 = 0, V2 = 0, V3 =20-1, so the heursitic value of *s* is 19, for node x_1, V1 = 1, V2 = 1, V3 =20-2, value of x_1 is 20, for node x_2, V1 = 2, V2 = 1, V3 =20-5, value of x_2 is 18. Similarly, the value of x_4 =24, x_5 = 24. So we pick edge $x_2 \rightarrow x_4$ as the deleted edge since it has the best heuristic value.

The main idea for designing the heuristic functions is: in the current subgraph *s*,

i) if the current node has shorter distance to the source node, there is more possibility to find the optimal lightpath;

ii) if there is less wavelength lost up to this node, there is more possibility to found the optimal lightpath;

iii) if the current node has more "children", there is more possibility to found the optimal lightpath.

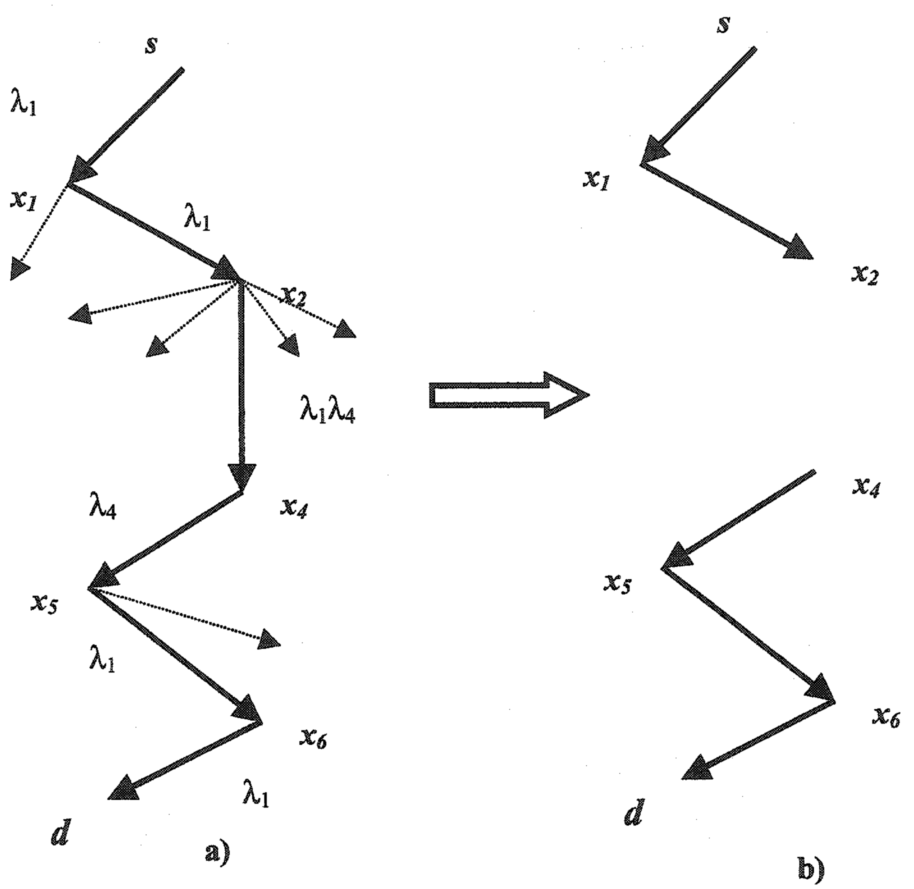


Fig 4.4 An Example of Heuristic Function of Type 1 Move

4.7.2 Heuristic Function for Type 2 Move

The second heuristic function (*H2*) is used for the type 2 move (add an edge x → y in situation 1) shown as in Figure 4.5.

H2 = V4 * M4

M4 is a constant.

V4 = Length from *s* to the node x in the current subgraph (in situation 1) - length from *s* to the node y in the current subgraph (in situation 1)

Since the length from the source node to the node x in shorter than the length from the source node to the node y, the heuristic value will be a negative value.

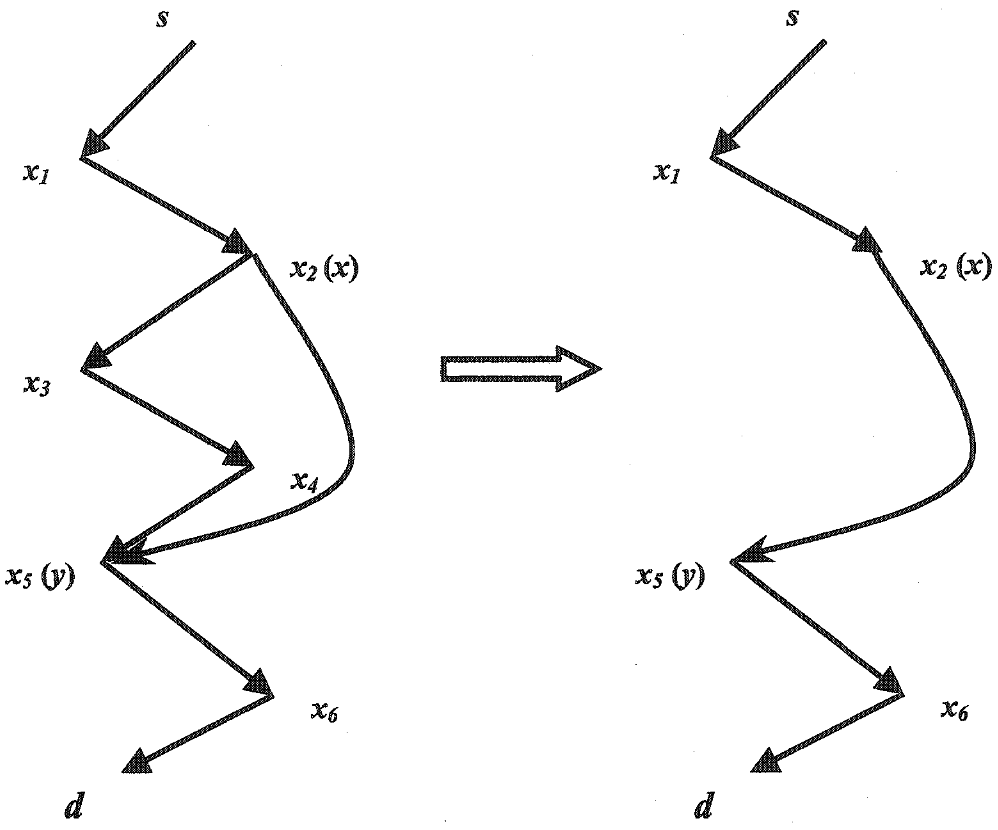


Figure 4.5 Heuristic Function for Type 2 Move

The main idea for design this heuristic functions is: we will pick an edge x → y, in which span the most number of edges, as the best move in the current lightpath.

4.7.3 Heuristic Function for Type 3 Move

The heuristic function which is used to evaluate the candidate moves for type 3 move (add an edge x → y in situation 2) is ***H3***. We defined ***H3*** as:

$H3$= V5* M5 + V6 * M6 + V7 * M7.

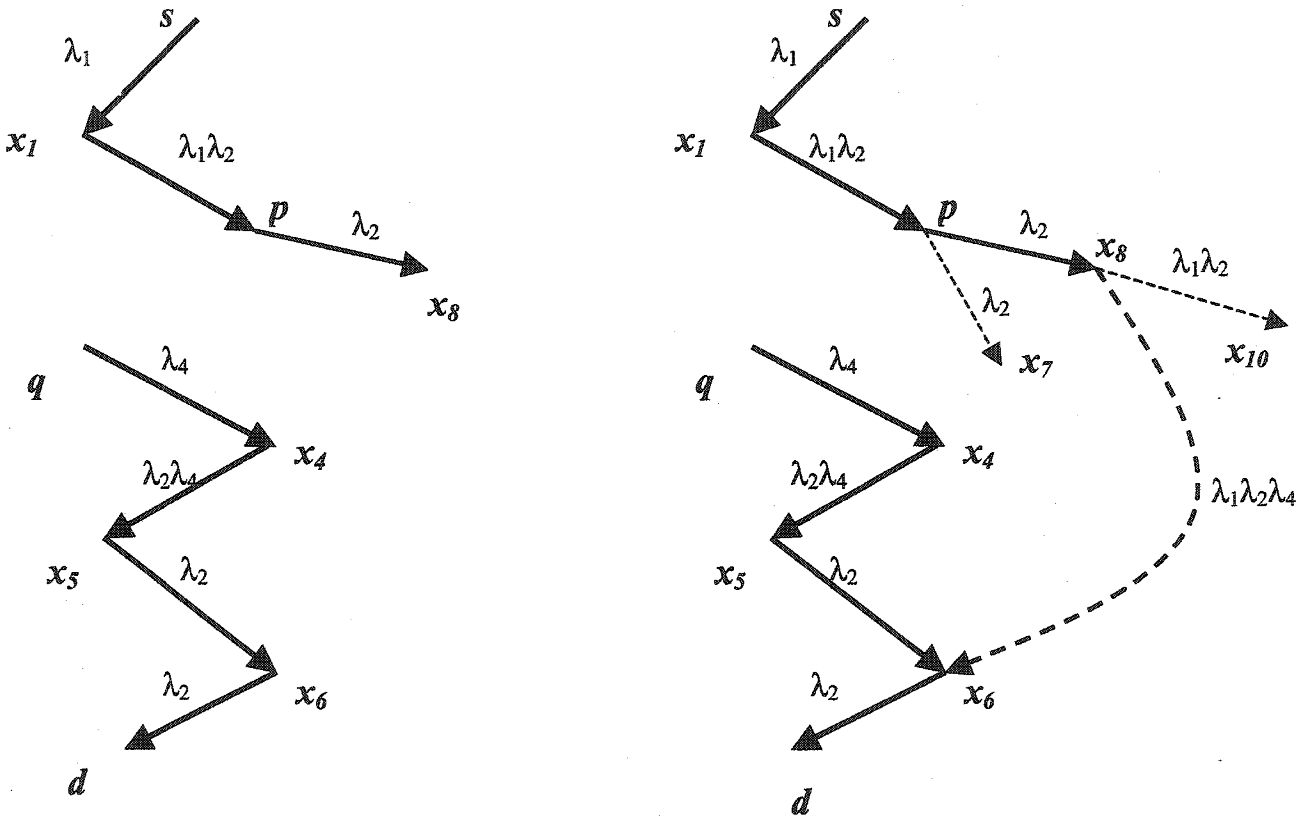
M5, M6 and M7 are three constants. But the value of M7 is much larger than M5 and M6. There are three parameters in this heuristic function:

- V5 = The length from the source node to the current node in the current lightpath
- V6 = The number of wavelengths lost up to current node
- $$V7 = \begin{cases} \mathbf{L_{new}} - \mathbf{L_{old}}, & \text{node y is one of nodes in P2 and } \mathbf{L_{new}} < \mathbf{L_{old}} \\ 0, & \text{otherwise} \end{cases}$$

Here, $\mathbf{L_{old}}$ is the length of current best lightpath, and $\mathbf{L_{new}}$ = length of current lightpath from source node s to the current node x + the length of node y to destination node d + 1.

Since $\mathbf{L_{new}} < \mathbf{L_{old}}$, V7 will be a negative number if it is not zero.

The main idea for designing this heuristic functions is: in the current subgraph s, if the current node x has a shorter distance to the source node, there is more possibility to find the optimal lightpath; if there is less wavelength lost until this node, there is more possibility to find the optimal lightpath. If the node y is in the path $q \rightarrow d$, it will return to situation 1 and have a shorter path than the current best lightpath (which is our most desirable situation), and in this case, the heuristic function always gives much better heuristic value than staying in situation 2 (since the value of V7 is much larger than V5 and V6).



(a)

(b)

Figure 4.6 Heuristic Function for Type 3 Move

For example, Figure 4.6 shows a given network which has 4 wavelengths: λ_1, λ_2, λ_3 and λ_4, and the current subgraph S is shown in Figure 4.6 (a). Suppose we have three candidate moves to add edge $p \rightarrow x_7$, $x_8 \rightarrow x_{10}$ and $x_8 \rightarrow x_6$ as shown in Figure 4.6 (b). We assume that the length of current best lightpath is 7. Node x_6 is in P_2 and the length of path $s \rightarrow x_1 \rightarrow p \rightarrow x_8 \rightarrow x_6 \rightarrow d$ is 5, which is shorter than the current best lightpath. Because the value of M7 is much larger than M5 and M6, and $\mathbf{L_{new}} - \mathbf{L_{old}}$ is –2, the heuristic value of adding the edge $x_8 \rightarrow x_6$ will be much smaller than adding the other two edges. So we add this edge and go back to situation 1 and get a new current best lightpath $s \rightarrow x_1 \rightarrow p \rightarrow x_8 \rightarrow x_6 \rightarrow d$, using wavelength is λ_3.

4.8 The Experiments

We have tested our algorithm on a number of different networks. The networks range in size from a small 5-node network to more practical sized networks of 26 and 31 nodes.

In each case we use our MILP formulation and CPLEX to find the optimal solution. The test data are shown in appendix A (Tables 4.2 – 4.6). For each network, we first select the lightpath to be established. This is done by randomly selecting a source *s* and a destination *d* for the lightpath. Then we invoke the tabu search algorithm to find, if possible, a path and an available wavelength for the lightpath. Once this is completed a new lightpath is selected and the process continues as before.

4.9 Result Analysis

Table 4.1 summarizes the results of our experiments for the different networks.

# of Nodes	Percentage of optimal solution	Percentage within 2-hops of optimal solution
5	100	100
10	90	100
18	86	90.5
26	79	91.6
31	75	91.6

Table 4.1 Experimental Results

From Table 4.1, we can see that our algorithm works extremely well for smaller networks. It always gives the optimal solution for the 5-node network. For the 10-node network, the solution is usually optimal and always within 2-hops of the best solution. As the network size increases, it is more difficult to find an optimal solution. Even for the larger networks, the algorithm performs quite well, giving us a solution close to the optimal over 90% of the time. So, based on our preliminary results, it seems that tabu search is a promising candidate for dynamic lightpath allocation.

Chapter 5 Conclusions and Future Work

The objective of this thesis is to find whether the tabu search can be used to set up, in a WDM network, a lightpath which uses as little optical resources as possible. In this investigation, the network is an all-optical WDM network, and there are already some existing lightpaths in the network which were set up in response to previous requests for communication. We have developed a tabu search heuristic approach which is used to establish a near-optimal lightpath dynamically for a new communication request satisfying the wavelength continuity constraint in a WDM network. As far as we know, it is the first application of the tabu search method for WDM networks. We have tested our approach with networks of varying sizes from 5 nodes to 31 nodes. By comparing our results with the results obtained using the MILP approach, we find that the tabu search is a promising technology for the dynamic lightpath allocation problem in WDM networks. For all the networks we have tested, our technique finding the optimal solution at least 75% of the time. It is especially suited for smaller networks.

Due to lack of time we could not study the time required for the tabu search approach and compare it to the time required for the MILP approach. In our approach, we focused on short-term memory to solve the problem. For more complex real-life networks, if we apply the component long term memory, it should be more effective. The effect of tabu tenure and development of more efficient heuristic are also expected to improve the search. These areas should be included in any future study.

Bibliography

[ABJ02] Y. P. Aneja, S. Bandyopadhyay and A. Jaekel, "An efficient protection scheme for WDM networks using dynamic lightpath allocation", HPC-Asia, December 2002.

[Ach00] K. Achan, "Tabu Search in Multi-Hop Optical Network Design", Master's Thesis, Windsor, University of Windsor, CA. 2000.

[Agr97] G. Agrawal, "Fiber-Optic Communication Systems", John Wiley & Sons, Inc. 1997.

[AS00] A. Armaral and R. Scrich, "Tabu Search for Minimizing Total Tardiness in a Job Shop", International Journal of Production Economics, Volume 63, Issue: 2, pp. 131-140, January 15, 2000.

[BB98] J. Blue and K. Bennett, "Hybrid extreme point tabu search", European Journal of Operational Research, Volume 106, Number 2-3, 1998.

[BM80] E. Balas and M. Martin, "Pivot and Complement - A Heuristic for 0-1 Programming", Management Science, 26 (1), pp 86 - 96, 1980.

[BO99]G. Barbarosoglu & D. Ozgur "A Tabu Search Algorithm for the Vehicle Routing Problem", Computers & Operations Research, pp. 255-270, Volume: 26, Issue: 3, March 1999.

[BR03] C. Blum, A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison", ACM Computing Surveys (CSUR), Volume 35, Issue 3, Pages: 268 – 308, September 2003.

[BR99] M. Bastos and C. Ribeiro, "Reactive Tabu Search with Path Relinking for the Steiner Problem in Graphs", In Proceedings of the Third Metaheuristics International Conference, pp. 31-36, 1999.

[BT94] R. Battiti and G. Tecchiolli, "The Reactive Tabu Search", ORSA Journal on Computing, 6(2): 126-140, 1994.

[BT95a] R. Battiti and G. Tecchiolli, "The Continuous Reactive Tabu Search: Blending Combinatorial Optimization and Stochastic Search for Global Optimization", Annals of Operations Research, in press, 1995.

[BT95b] R. Battiti and G. Tecchiolli, "Training Neural Nets with Reactive Tabu Search", IEEE Trans. on Neural Networks, pp. 1185-1200, 6(5), 1995.

[CC98] W. Chiang and C. Chi, "Intelligent Local Search Strategies for Solving Facility Layout Problems with the Quadratic Assignment Problem Formulation", Journal of Operational Research, Volume: 106, Issue: 2-3, pp. 457-488, April 16, 1998.

[CFG98] E. Costamagna, A.Fanni and G. Giacinto, "A Tabu Search Algorithm for the Optimization of Telecommunication Networks", European Journal of Operational Research, Volume: 106, pp. 357-372, 1998.

[CGK92] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath Communications: An Approach to High-Bandwidth Optical WAN's", IEEE Transactions on Communications, Volume: 40, no. 7, pp. 1171-1182, July 1992.

[CGLM+00] J. S. Choi, N. Golmie, G. Lapeyrere, F. Mouveaux, and D. Su, "A Functional Classification of Routing and Wavelength Assignment Schemes in DWDM Networks: Static Case", Published at Proceedings of the 7th International Conference on Optical Communications and Networks, IPNET, pp. 1109-1115. Paris, France, January 2000.

[Chl93] I. Chlamtac et al., "Lightnets: topologies for high-speed optical networks", IEEE/OSA Journal of Lightwave Technology, vol. 11. pp. 951-961, May/June, 1993.

[CS00] R. Chelouah and P. Siarry, "Tabu Search Applied to Global Optimization", European Journal of Operational Research, pp. 256-270, Volume: 123, Issue: 2, June 1, 2000.

[DAA95] T. Dean, J. Allen and Y. Aloimonos, "Artificial Intelligence: Theory and Practice", Addison-Wesley Publishing Company, 1995.

[FILG01] J. Ferland, S. Ichoua, A. Lavoie and E. Gagne, " Scheduling Using Tabu Search Methods with Intensification and Diversification", Computers and Operations Research, pp. 1075-1092, Volume: 28, Issue: 11, September 2001.

[Ger96a] O. Gerstel et al., "Dynamic channel assignment for WDM optical Networks with little or no wavelength conversion", Proc. 34th Annual Allerton Conf. pp. 32-34. Monticello, IL, October 1996.

[Ger96b] O. Gerstel et al., "Dynamic wavelength allocation in WDM ring networks", IBM Research Report RC 20462, May 1996.

[GHG94] M. Gendreau, A. Hertz and L. Gilbert, "A Tabu Search Heuristic for the Vehicle Routing problem", Management Science, Volume: 40, Issue: 10, pp. 1276-1290, 1994.

[GK97] O. Gerstel and S. Kutten, "Dynamic Wavelength Allocation in All-Optical Ring Networks," Proc., IEEE ICC'97, Montreal, Quebec, Canada, vol. 1, pp. 432-436, June 1997.

[GL97] F. Glover and M. Laguna, "Tabu Search", Kluwer Acadumic Publishers, 1997.

[Glo86] F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence", Computers and Operations Research 13, No. 5, pp533-549, 1986.

[Glo90] F. Glover, "Tabu Search: A tutorial", Interface, vol.20, pp.74-94, August 1990.

[Glo93] F. Glover, "A User's Guide to Tabu Search", Annals of Operations Research, pp. 3-28, 1993.

[Glo98] F. Glover, "Tabu Search - Wellsprings and Challenges", European Journal of Operational Research, Volume: 106, Issue: 2-3, pp. 221-225, April 16, 1998.

[GM03] A. Gencata and B. Mukherjee, "Virtual-Topology Adaptation of WDM Mesh Networks Under Dynamic Traffic", IEEE ACM Transactions on Networking, vol. 11, No. 2, pp. 236-247, April 2003.

[GQ96] D. Gan and Z. Qu, "Large-Scale var Optimization and Planning by Tabu Search", Electric Power Systems Research, Volume: 39, Issue: 3, pp. 195-204, December 1996.

[GT93] F.Glover & E. Taillard "A User's Guide to Tabu Search", Annals of Operations Research, pp. 3-28, 1993.

[Han97] M. Hansen, "Tabu Search for Multiobjective Optimization: MOTS", 13th International Conference on Multiple Criteria Decision Making, University of Cape Town, January 6-10, 1997.

[HJB99] P. Hansen, B. Jaumard and D. Guetse, "Best Second Order Bounds for Two-Terminal Network Reliability with Dependent Edge Failures", Discrete Applied Mathematics, Volume: 96-97, pp. 375-393, October 15, 1999.

[HK02] J. Hurink and S. Knust, "A Tabu Search Algorithm for Scheduling Single Robot in Job-Shop Environment", Discrete Applied Mathematics, Volume: 119, Issue: 1-2, pp. 181-203, June 15, 2002.

[IL01] ILOG, "ILOG CPLEX 7.5, Getting Started", November 2001.

[KL98] R. K. Kincaid and K. E. Laba, "Reactive Tabu Search and Sensor Selection in Active Structural Acoustic Control Problems, Journal of Heuristics", Volume: 4, Issue: 3, pp. 199-220, September 1998.

[LCG98] L. Lopaz, M. Carter and M. Gendreau, "The Hot Strip Mill Production Scheduling Problem: A Tabu Search Approach", European Journal of Operational Research, Volume: 106, Issue: 2-3, pp. 317-335, April 16, 1998.

[LG98] A. Løkketangen and F. Glover, "Solving Zero-One Mixed Integer Programming Problems Using Tabu Search", European Journal of Operational Research, Volume: 106, Issue: 2-3, pp. 624-658, April 16, 1998.

[LMC99] M. Laguna, R. Marti and V. Compos, "Intensification and Diversification with Elite Tabu Search Solutions for the Linear Ordering Problem", Computers and Operations Research, pp. 1217-1230, Volume: 26, Issue: 12, October 1999.

[LOQ97] G. Laporte, J. Potvin and F. Quilleret, "A Tabu Search Heuristic using Genetic Diversification for the Clustered Traveling Salesman Problem", Journal of Heuristice, Volume: 2, Issue: 3, pp. 187-200, 1997.

[Lug02] G. Luger, "Artificial Intelligence: Structures and Strategies for Complex Problem Solving" Fourth Edition, Pearson Education Limited, 2002.

[MA98] A. Mokhtar, M. Azizoğlu, "Adaptice Wavelength Routing in All-Optical Networks", IEEE ACM Transactions on Networking, vol. 6, pp. 197-206, April 1998.

[Mar93] M. A. Marson et al. "Topologies for Wavelength Routing All-Optical Networks", IEEE/ACM Tran. Networking, Vol. 1, pp. 534-546. October 1993.

[MBLN93] M. Marsan, A. Bianco, E. Leonardi, F. Neri, "Topologies for wavelength-routing all-optical networks", IEEE ACM Transactions on Networking, vol. 1, pp. 534-546. October 1993.

[MBRM96] B. Mukherjee, D. Banerjee, S. Ramamurthy, A. Mukherjee, "Some princles for designing a wide-area WDM otpical networks", IEEE ACM Transactions on Networking, vol. 4, pp. 684-696, October 1996.

[MG01] G. Morley and W. Grover, "Tabu Search Optimization of Optical Ring Transport Networks", Proceedings IEEE Globecom, pp. 25-29, November 2001.

[MPPM02] G. Maier, A. Pattavina, S. Patre, M. Martinelli, "Optical Network Survivability: Protection Techniques in the WDM Layer", Photonic Network Communication, 4:3/4, 251-269, 2002.

[Muk00] B. Mukherjee, "WDM Optical Networks: Progress and Challenges", (Invited Paper), IEEE Journal on Selected Areas in Communications (Special Issue on "PROTOCOLS AND ARCHITECTURES FOR NEXT GENERATION OPTICAL WDM NETWORKS"), vol. 18, no. 10, pp. 1810-1824, October 2000.

[Muk97] B. Mukherjee, "Optical Communication Networks", McGraw-Hill, July 1997.

[Mukh92-1] B. Mukherijee, "WDM-based local lightwave Networks, partI: Single-Hop Systems", IEEE Network, vol. 6, Issue. 3, pp. 12-27, May 1992.

[Mukh92-2] B. Mukherijee, "WDM-based local lightwave Networks, partII: Multi-Hop Systems", IEEE Network, vol. 6, Issue. 4, pp. 20-32, May 1992.

[NI98] K. Nonobe and T. Ibaraki, "A Tabu Search Approach to the Constraint Satisfaction Problem as a General Problem Solver", European Journal of Operational Research, 1998.

[NW00] W. Nanry and B. Wesley, "Solving the Pickup and Delivery Problem with Time Windows Using Reactive Tabu Search", Transportation Research Part B: Methodological, Volume: 34, Issue: 2, pp. 107-121, February 2000.

[OB03] A. Ozdaglar, D. Bertsekas, "Routing and wavelength assignment in optical networks", IEEE ACM Transactions on Networking, vol.11, no. 2, pp. 259-272, April 2003.

[OW02] I. Osman and N. Wassan, "A Reactive Tabu Search Meta-Heuristic for the Vehicle Routing Problem with Back-Hauls", Journal of Scheduling, pp. 263 – 285, Volume: 5, Issue: 4, July/August 2002.

[PAR00] S. Ponnambalam, P. Aravindan and S. Rajesh, "A Tabu Search Algorithm for Job Shop Scheduling", International Journal of Advanced Manufacturing Technology, Volume: 16, Issue: 10, pp. 765 – 771, August 21, 2000.

[Pir96] M. Pirlot, "General local search methods", European Journal of Operational Research, Volume: 92, Issue: 3, pp. 493-511, August 9, 1996.

[Reg98] C. Rego, "Relaxed Tours and Path Ejection for the Traveling Salesman Problem", European Journal of Operational Research, Vol. 106, Issue: 2-3, pp. 552-538, April 1998.

[Ric83] E. Rich, "Artificial Intelligence", McGraw-Hill Book Company, 1983.

[Rog01] A. Rogers, "Understanding optical fiber communications", Artech House, 2001.

[RS95] R. Ramaswami and K. Sivarajan, "Routing and Wavelength Assignemnt in All-Optical Networks", IEEE ACM Transactions on Networking, vol. 3, no. 5, pp. 489-500, 1995.

[RS98] R. Ramaswami and K. N. Sivarajan, "Optical Networks: A Practical Perspective", Morgan Kaufmann Publishers, 1998.

[Sab00] R. Sabella, "Key Elements for WDM Transport Networks", Photonic Network Communication, vol. 2, Issue: 1, pp. 7-14, March 2000.

[Sah00] D. Saha, "Lightpath Versus Semi-LightPath: Some Studies on Optimal Routing in WDM Optical Networks", Photonic Network Communications, 2:2, 155-161, 2000.

[Sal02] S. Salhi, "Defining Tabu List Size and Aspiration Criterion Within Tabu Search Methods", Computers and Operations Reaearch, Volume: 29, Issue: 1, pp. 67-86, January 2002.

[SAMD98] M. Sun, J. Aronson, P. Mckeown and D. Drinka, "A Tabu Search Heuristic Procedure for the Fixed Charge Transportation Problem", European Journal of Operational Research, Volume: 106, Issue: 2-3, pp. 441-456, April 16, 1998.

[Sch96] A. Schaerf, "Tabu Search Techniques for Large High-School Timetabling Problems", In Proceedings of the Fourteenth National Conference on Artificial Intelligence, pp. 363 - 368, Portland, Oregon, August 1996.

[SGLO97] Y. Sharaiha, M. Gendreau, G. Laporte and I. Osman, "A Tabu Search Algorithm for the Capacitated Shortest Spanning Tree Problem", Networks, Volume: 29, Issue: 3, pp. 161 - 171, May 1997.

[SLW99] C. Shyur, T. Lu and U. Wen, "Applying Tabu Search to Spare Capacity Planning for Network Restoration", Computers & Operations Research, Volume: 26, Issue: 12, pp. 1175-1194, October 1999.

[Tah82] H. A. Taha, "Operations Research, An introduction third edition", Macmillan Publishers, 1982.

[Tai91] E. Taillard, "Robust Tabu Search for the Quadratic Assignment Problem", Parallel Computing, No. 17, pp. 443-455, 1991.

[TB97] E. Taillard and P. Badeau, "A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows", Transportation Science 31, pp. 170-186, 1997.

[TJ98] S. Tsubakitani and E. James, "Optimizing Tabu List Size for the Traveling Salesman Problem", Computers & Operations Research, Volume: 25, pp. 91-97, Issue: 2, February 1998.

[TS02] P. Tomsu and C. Schmutzer, "Next Generation Optical Networks", Prentice-Hall, Inc. 2002.

[WC97] F. Wen and C. Chang, "Transmission Network Optimal Planning Using the Tabu Search Method", Electric Power System Research, Volume: 42, Issue: 2, pp. 153-163, August 1997.

[WWWJ02] J. M. Ware, I. Wilson, J. A. Ware, C. Jones, "A tabu search approach to automated map generalisation", Proceedings of the tenth ACM international symposium on Advances in geographic information systems, November 2002.

[XK96] J. Xu & J. Kelly "A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem", Transportation Science 30, pp. 379-393, 1996.

[YDA03] S. Yan, J. Deogun and M. Ali, "Routing in Sparse Splitting Optical Networks with Multicast Traffic", Computer Networks, Volume: 41, Issue: 1, pp. 89-113, January 15, 2003.

[YDM00] S. Yao, S. Dixit, and B. Mukherjee, "Advances in Photonic Packet Switching: An Overview", IEEE Communications Magazine, vol. 38, no. 2, pp. 84-94, February 2000.

[YJ01] S.Yuan and J. Jue, "A heuristic routing algorithm for shared protection in connection – oriented network", in SPIE proceeding OptiComm 2001, vol. 4599, pp. 142 – 152, Denver, August 2001.

[Zang03] H.Zang, C. Ou, B. Mukherjee, "Path-protection routing and wavelength assignment in (RWA) in WDM mesh networks under duct-layer constraints", Networking, IEEE/ACM Transactions on, vol. 11 Issue: 2, April 2003.

[ZJM00] H. Zang, J, P. Jue and B. Mukherjee, "A review of routing and wavelength assignment approaches for savelength-routed optical WDM networks", SPIE/Baltzer Optical Networks Magazine (ONM), vol. 1, no. 1, pp. 47-60, January 2000.

[ZJSR+01] H. Zang, J. P. Jue, L. Sahasrabuddhe, S. Ramamurthy, and B. Mukherjee, "Dynamic Lightpath Establishment in Wavelength-Routed WDM Networks", IEEE Communications Magazine, vol. 39, no. 9, pp. 100-108, September 2001.

[ZM01] H. Zang and B. Mukherjee, "Connection Management for Survivable Wavelength-Routed WDM Mesh Networks", SPIE Optical Networks Magazine, Special Issue on Protection and Survivability in Optical Networks, vol. 2, no. 4, pp. 17-28, July 2001.

Appendix A: Testing Data

Source	Destination	Tabu Search	W(T)	CPLEX	W(C)	Shortest
0	4	0--2--1--3--4	w1	0--2--1--3--4	w1	Yes
1	3	1--3	w1	1--3	w1	Yes
2	4	2--1--4	w0	2--1--4	w0	Yes
1	4	1--4	w0	1--4	w0	Yes
4	1	4--1	w1	4--1	w1	Yes
3	0	3--2--0	w0	3--2--0	w1	Yes
0	1	0--2--1	w1	0--2--1	w1	Yes
0	2	0--2	w1	0--2	w1	Yes
0	3	0--2--1--3	w1	0--2--1--3	w1	Yes
1	2	1--2	w0	1--2	w1	Yes
1	0	1--0	w0	1--0	w1	Yes
4	0	4--1--0	w1	4--1--0	w1	Yes

Table 4.2 The results of a 5-node network

Source	Destination	Tabu Search	W (T)	CPLEX	W (C)	Shortest
2	7	2--3--7	w0	2--3--7	w3	Yes
9	1	9--8--1	w0	9--0--1	w0	Yes
3	5	3--4--5	w0	3--4--5	w0	Yes
7	4	7--6--5--4	w0	7--3--4	w0	No
5	8	5--1--8	w0	5--1--8	w0	Yes
2	5	2--1--5	w0	2--1--5	w0	Yes
6	2	6--5--1--2	w0	6--5--1--2	w0	Yes
5	9	5--1--8--9	w0	5--1--0--9	w0	Yes
3	8	3--8	w0	3--8	w0	Yes
6	9	6--7--8--9	w0	6--7--8--9	w0	Yes
8	2	8--3--2	w0	8--1--2	w0	Yes
0	5	0--1--5	w0	0--1--5	w0	Yes
7	0	7--8--9--0	w0	7--8--1--0	w0	Yes
4	9	4--3--7--8--9	w0	4--3--8--9	w0	No
0	2	0--1--2	w0	0--1--2	w0	Yes
1	6	1--5--6	w0	1--5--6	w0	Yes
3	4	3--4	w0	3--4	w0	Yes
9	3	9--8--3	w0	9--8--3	w0	Yes
1	3	1--8--3	w0	1--8--3	w0	Yes

Table 4.3 The results of a 10-node network

Source	Destination	Tabu Search	W(T)	CPLEX	W(C)	Shortest
12	7	12-11-4-7	w0	12-11-4-7	w2	Yes
3	15	3-4-7-15	w0	3-4-7-15	w2	Yes
0	10	0-1-2-14-15-16-8-9-10	w1	0-1-2-14-15-7-4-11-10	w2	Yes
6	7	6-5-17-16-15-7	w0	6-5-17-16-8-7	w2	Yes
2	12	2-14-15-16-8-9-10-11-12	w1	2-14-15-7-4-11-12	w2	No
10	17	10-9-8-16-17	w1	10-9-8-16-17	w2	Yes
2	6	2-14-15-16-17-5-6	w1	2-14-15-16-17-5-6	w2	Yes
16	0	16-15-14-2-1-0	w0	16-15-14-2-1-0	w2	Yes
13	16	13-3-4-7-15-16	w2	13-3-4-7-15-16	w2	Yes
11	17	11-10-9-8-16-17	w2	11-10-9-8-16-17	w2	Yes
7	12	7-4-11-12	w0	7-4-11-12	w2	Yes
17	9	17-16-15-7-4-11-10-9	w0	17-16-8-9	w1	No
9	3	9-8-7-4-3	w0	9-8-7-4-3	w2	Yes
8	1	8-16-15-14-2-1	w0	8-16-15-14-2-1	w2	Yes
5	11	5-17-16-15-7-4-11	w0	5-17-16-8-7-4-11	w2	Yes
15	4	15-7-4	w0	15-7-4	w2	Yes
1	5	1-2-14-15-16-17-5	w1	1-2-14-15-16-17-5	w2	Yes
2	3	2-14-15-16-8-9-10-11-12-13-3	w1	2-14-15-7-4-3	w2	No
4	5	4-7-15-16-17-5	w2	4-11-18-17-5	w1	Yes
13	14	13-3-4-7-15-14	w2	13-3-4-7-15-14	w2	Yes
4	6	4-7-15-16-17-5-6	w2	4-7-15-16-17-5-6	w2	Yes

Table 4.4 The results of a 18-node network

Source	Destination	Tabu Search	W(T)	CPLEX	W(C)	Shortest
15	11	15-14-13-12-11	w0	15-14-13-12-11	w1	Yes
12	20	12-13-14-21-20	w0	12-11-0-19-20	w1	Yes
18	4	18-25-7-8-3-4	w0	18-11-0-1-5-4	w1	Yes
22	9	22-23-24-25-18-17-9	w0	22-23-24-16-17-9	w1	No
8	16	8-9-17-16	w1	8-9-17-16	w1	Yes
19	10	19-0-25-24-16-17-9-8	w0	19-0-11-10	w1	No
13	21	13-14-21	w0	13-14-21	w1	Yes
9	14	9-17-18-25-21-14	w0	9-17-16-15-14	w1	No
24	15	24-16-15	w0	24-16-15	w1	Yes
19	24	19-20-21-14	w0	19-20-21-14	w1	Yes
6	19	6-5-1-0-19	w1	6-7-25-0-19	w1	Yes
1	15	1-0-25-21-14-15	w0	1-0-25-21-14-15	w1	Yes
4	23	4-5-6-7-25-24-23	w0	4-5-1-0-25-24-23	w1	Yes
17	3	17-9-8-3	w0	17-9-8-3	w1	Yes
0	20	0-19-20	w0	0-19-20	w1	Yes
3	14	3-2-1-0-25-21-14	w0	3-8-7-25-21-14	w1	No
16	7	16-24-25-7	w0	16-24-25-7	w1	Yes
25	8	25-7-8	w0	25-7-8	w1	Yes
23	7	23-24-25-7	w0	23-24-25-7	w1	Yes
13	3	13-12-11-10-9-8-3	w1	13-12-11-0-1-2-3	w1	Yes
15	6	15-14-13-12-11-10-9-8-3-4-5-6	w1	15-16-24-25-7-6	w1	No
12	23	12-11-0-15-24-23	w0	12-11-0-15-24-23	w1	Yes
2	13	2-1-0-11-12-13	w0	2-1-0-11-12-13	w1	Yes
5	15	5-6-7-25-24-16-15	w0	5-1-0-25-21-14-15	w1	Yes

Table 4.5 The results of a 26-node network

Source	Destination	Tabu Search	W (T)	CPLEX	W (C)	Shortest
11	21	11-12-13-14-21	w2	11-12-13-14-21	w3	Yes
15	11	15-14-13-12-11	w0	15-14-13-12-11	w3	Yes
12	20	12-11-0-19-20	w0	12-11-0-19-20	w3	Yes
22	9	22-21-25-18-19-0-11-10-9	w0	22-21-25-18-17-9	w4	No
13	21	13-14-21	w2	13-14-21	w3	Yes
10	20	10-9-17-18-19-20	w1	10-11-0-19-20	w1	No
6	19	6-5-1-0-19	w0	6-5-1-0-19	w3	Yes
1	15	1-8-7-25-21-16-15	w0	1-8-7-25-21-14-15	w3	Yes
4	23	4-3-8-7-25-21-22-23	w0	4-5-1-0-25-24-23	w1	Yes
17	3	17-9-10-11-0-1-8-3	w0	17-18-25-7-8-3	w3	No
3	14	3-8-7-25-24-16-15-14	w0	3-8-7-25-21-14	w3	No
7	0	7-8-1-0	w0	7-8-1-0	w3	Yes
18	4	18-19-0-1-8-3-4	w0	18-25-7-8-3-4	w3	No
25	8	25-7-8	w0	25-7-8	w3	Yes
13	3	13-12-11-0-1-8-3	w0	13-12-11-0-1-2-3	w3	Yes
15	6	15-14-13-12-11-0-1-5-6	w0	15-16-24-25-7-6	w3	No
12	22	12-13-14-21-22	w2	12-13-14-21-22	w3	Yes
9	14	9-10-11-12-13-14	w2	9-17-18-25-21-14	w3	Yes
23	7	23-22-21-25-7	w0	23-22-21-25-7	w4	Yes
13	0	13-12-11-0	w0	13-12-11-0	w3	Yes
0	20	0-19-20	w0	0-19-20	w3	Yes
16	7	16-24-25-7	w0	16-24-25-7	w3	Yes
12	23	12-13-14-21-22-23	w2	12-11-0-15-24-23	w1	Yes
8	16	8-9-17-16	w1	8-9-17-16	w1	Yes

Table 4.6 The results of a 31-node network

Vita Auctoris

Shaohong Wang obtained her B.S. in Computer Science at Beijing University in 1994. She is currently a candidate for the Master's Degree in Computer Science at the University of Windsor.