

## University of Windsor Scholarship at UWindsor

---

Electronic Theses and Dissertations

---

2000

# Tabu search in multi-hop optical network design.

Kannan. Achan  
*University of Windsor*

Follow this and additional works at: <http://scholar.uwindsor.ca/etd>

---

### Recommended Citation

Achan, Kannan., "Tabu search in multi-hop optical network design." (2000). *Electronic Theses and Dissertations*. Paper 3715.

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email ([scholarship@uwindsor.ca](mailto:scholarship@uwindsor.ca)) or by telephone at 519-253-3000ext. 3208.

## **INFORMATION TO USERS**

**This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.**

**The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.**

**In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.**

**Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.**

**Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.**

**Bell & Howell Information and Learning  
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA  
800-521-0600**

**UMI<sup>®</sup>**



**Tabu Search in  
Multi-hop Optical Network Design**

**by**

**Kannan Achan**

**A Thesis**

**Submitted to the College 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**

**2000**



**National Library  
of Canada**

**Acquisitions and  
Bibliographic Services**

**395 Wellington Street  
Ottawa ON K1A 0N4  
Canada**

**Bibliothèque nationale  
du Canada**

**Acquisitions et  
services bibliographiques**

**395, rue Wellington  
Ottawa ON K1A 0N4  
Canada**

*Your file Votre référence*

*Our file Notre référence*

**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.**

**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 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.**

**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 autrement reproduits sans son autorisation.**

**0-612-52497-3**

**Canada**

**Kannan Achan 2000**

**© All Rights Reserved**

## **Abstract**

The logical topology design of a multi-hop optical network involves choosing the logical edges and simultaneously identifying a routing strategy which minimizes congestion in the network. Even small instances of this combined problem is extremely intractable due to the presence of a large number of 0-1 variables. This problem suffers from the inherent limitations imposed by classical algorithms like the branch and bound for finding integer solutions. Several heuristic algorithms have been proposed to circumvent this setback but the performance of these heuristic algorithms vary with the type of traffic and other network related constraints. In this thesis we present a tabu search based approach to solve the logical topology design problem. We compared our results with branch and bound and HLDA, a heuristic topology design algorithm.

***to Govinda...***



## **Acknowledgments**

I take this opportunity to thank my two advisors, Dr. Subir Bandyopadhyay and Dr. Yash Aneja for their continuous encouragement and intensive support throughout my graduate studies. This work would not have been achieved without their support and advice. I am grateful to Dr. Indra Tjandra, Dr. Richard Caron and Dr. Arunita Jaekel for their appreciated suggestions. I am thankful to my friend Anwar Haque for generating topologies using HLDA. Finally, I want to express my gratitude to my family for their support and love over these years.

# Contents

<b>Abstract</b>	iii
<b>Acknowledgments</b>	v
<b>List of Tables</b>	viii
<b>List of Diagrams</b>	ix
<b>Chapter 1 Introduction</b>	1
1.1 Thesis Organization	2
<b>Chapter 2 Review of Literature</b>	4
2.1 All-Optical networks	5
2.1.1 Components of optical network	6
2.1.2 Routing	6
2.1.3 Multiplexing	7
2.2 Wavelength Division Multiplexing (WDM)	8
2.3 Single-hop networks	10
2.4 Multi-hop networks	11
2.5 Wavelength routed optical networks (WRON)	13
2.6 Regular and irregular topologies	15
2.7 Logical topology design (WRON)	15
2.8 Tabu Search - Introduction	19
2.9 Memory and Tabu search	21
2.9.1 Short-term memory	21
2.9.2 Tabu list and Tabu tenure	23
2.10 Candidate list strategy	24
2.11 Aspiration criteria	25

2.12 Long-term memory	27
2.12.1 Intensification and Diversification	27
<b>Chapter 3 Problem Specification and Approach</b>	<b>29</b>
3.1 The Problem	29
3.2 Aggregate formulation	34
3.3 Our approach	39
3.3.1 Candidate list	40
3.3.2 Guidance mechanism	40
3.3.3 Move classification	41
3.3.4 Move evaluation	43
3.3.5 Move selection	43
3.3.6 Tabu conditions	44
3.3.7 Aspiration criteria	45
<b>Chapter 4 Implementation</b>	<b>47</b>
4.1 LU decomposition	47
<b>Chapter 5 Experiments</b>	<b>49</b>
5.1 Methodology	49
5.2 Test data	50
5.3 Experiments	53
<b>Chapter 6 Conclusion and Future Work</b>	<b>58</b>
<b>Selected Bibliography</b>	<b>60</b>
<b>Appendix 1</b>	<b>62</b>
<b>Appendix 2</b>	<b>68</b>
<b>Appendix 3</b>	<b>70</b>
<b>Vita Auctoris</b>	<b>72</b>

## **List of Diagrams**

<b>Wavelength Division Multiplexing</b>	<b>9</b>
<b>Single hop network (fixed transmitters and receivers)</b>	<b>11</b>
<b>4 node Multi-hop network</b>	<b>12</b>
<b>Wavelength Routed Optical Network (WRON)</b>	<b>14</b>
<b>Best move selection - Short-term memory</b>	<b>24</b>
<b>Tabu Search using short term memory</b>	<b>25</b>
<b>Flow Conservation</b>	<b>30</b>
<b>Average delay</b>	<b>32</b>
<b>Move classification</b>	<b>41</b>
<b>Logical topologies using TM1</b>	<b>54</b>
<b>LU decomposition</b>	<b>65</b>
<b>Upper Hessenberg matrix</b>	<b>66</b>
<b>Normalization</b>	<b>67</b>

## **List of Tables**

<b>Components of Optical network</b>	<b>6</b>
<b>Aggregate vs Disaggregare formulation</b>	<b>34</b>
<b>Congestion obtained for TM1 with <math>\Delta_l = 1</math></b>	<b>53</b>
<b>Congestion obtained for TM1</b>	<b>54</b>
<b>Congestion obtained for TM-A (7 nodes)</b>	<b>55</b>
<b>Congestion obtained for TM-A (8 nodes)</b>	<b>56</b>
<b>Comparative study using 7 six node networks</b>	<b>57</b>

## **Chapter 1**

# **Introduction**

Rapid advances in the field of optical network technology has made the realization of high speed wide area networks a practical possibility. All optical wavelength division multiplexed networks using wavelength routing are considered to be potential candidates for the next generation of wide area optical networks [RS2]. It is expected that high volume back bones in the near future will be based on this technology.

This thesis studies the problem of designing a logical topology and the associated routing algorithm for a wavelength routed optical network, given the underlying physical topology and a traffic matrix. Congestion, the maximum load offered to any logical link, is a good metric for determining the efficiency of a network[RS1]. The smaller the value of congestion, the better will be the throughput of the network. The problem of just routing the commodities by minimizing congestion is a linear programming problem and can be efficiently solved. But simultaneously fix-

ing the logical edges in the network adds an enormous amount of complexity to this problem as it involves solving a mixed integer program. The most common approach to solve this kind of problems is to use the standard branch and bound technique. But the resulting MILP is extremely intractable even for small networks due to the size of the resulting problem. Several researchers have proposed heuristic algorithms to solve this problem.

Tabu search is a meta heuristic that uses memory based strategies to guide a search to obtain good solutions. In this thesis we investigate the use of a tabu search heuristic proposed in [LG1] to solve the logical topology design problem. To address the issue of the efficiency of this approach, we compare the performance of this heuristic with results obtained by using a branch and bound routine.

## **1.1 Thesis Organization**

In chapter 2 we present a literature review of all the necessary background material. This includes a brief introduction to optical network technology and tabu search. In chapter 3 we discuss the problem specification and the approach employed to solve the problem. Here we discuss in detail the tabu search heuristic and the associated procedures.

Chapter 4 throws some insight to the implementation details. We discuss the problems with round-off errors and explain the use of LU decomposition. In chapter 5

we discuss the experimental results along with our interpretations of the results obtained. Finally in chapter 6 we present the conclusion and future work.



## **Chapter 2**

# **Review of Literature**

Telecommunication has taken a new direction due to the increased demand for voice and data networks. The need for bandwidth is increasing everyday. Rapid advances in optical fiber technology has made the realization of universal, high speed, wide area networks a practical possibility. An optical fiber provides countless benefits including low crosstalk, immunity to electromagnetic interference, short delay, high security and compactness. "Optical fiber is a remarkable communication medium, compared to other media such as copper or free space" [RS1].

An optical network transmits information encoded as lightwave signals in a very fast and reliable manner. In several networks, optical fiber is primarily used as a transmission medium while switching and routing are handled by electronics. These networks are grouped as first generation optical networks [Gre1]. These networks are widely employed in the telecommunications industry. SONET is a popular first generation optical network that is being widely used in the

telecommunication industry.

In these networks the electronic circuits at a particular node not only operate on the data intended for it but also on the data which are destined for other nodes. The load on the electronic circuits at a node can be considerably reduced if the latter could be done in the optical domain itself. This is the key feature behind the second generation optical networks.

## **2.1 All-Optical Networks**

These refer to the class of networks where the information path between the source node and the destination node remains entirely optical [Gre1]. Such a network offers protocol transparency apart from its high speed of operation [Muk1].

All-optical network technology is far superior to its previous generation networks by allowing a tremendous increase in capacity compared to the speed limitation of electronics. The important mechanism behind an all-optical network is the light path. Light paths are communication channels between two nodes and it is possible that a single light path might span several links [CGK1].

### 2.1.1 Components of Optical Network:

**Table 1: Components of Optical network**

Device	Function
Optical Regenerators	Regenerators convert optical signals to electrical signals, amplifies, reconverts to optical domain and then retransmits it.
Optical Amplifiers	An optical amplifier restores the signal level in a transmission and retransmits it. It amplifies without converting the signal to the electrical domain
Couplers	Couplers are used to split and to combine signals on a optical network
Multiplexer	A wavelength division multiplexer combines individual optical signals coming from various ports into a single outgoing port
De-multiplexer	A wavelength division demultiplexer splits signals at different wavelengths on the input port and routes them to different output ports.
Router	A router determines the next node to which a packet should be forwarded in order to reach its destination

### 2.1.2 Routing

Routing is the mechanism by which a packet travels from a source node to a destination node [Muk1]. There are two types of routing strategies that are usually

employed while building routers: static and dynamic routing. Static routing uses a predefined path for all communications. It is relatively simple and works faster because of the passive nature of its components. The main disadvantage of this routing scheme is that it is not scalable and a large number of wavelengths are needed to ensure that no two lightpaths sharing a fiber have the same wavelength. Dynamic routing determines the path taken by a lightpath at run time. This approach requires fewer wavelengths compared to static routing but involves more set-up time and more expensive optical components.

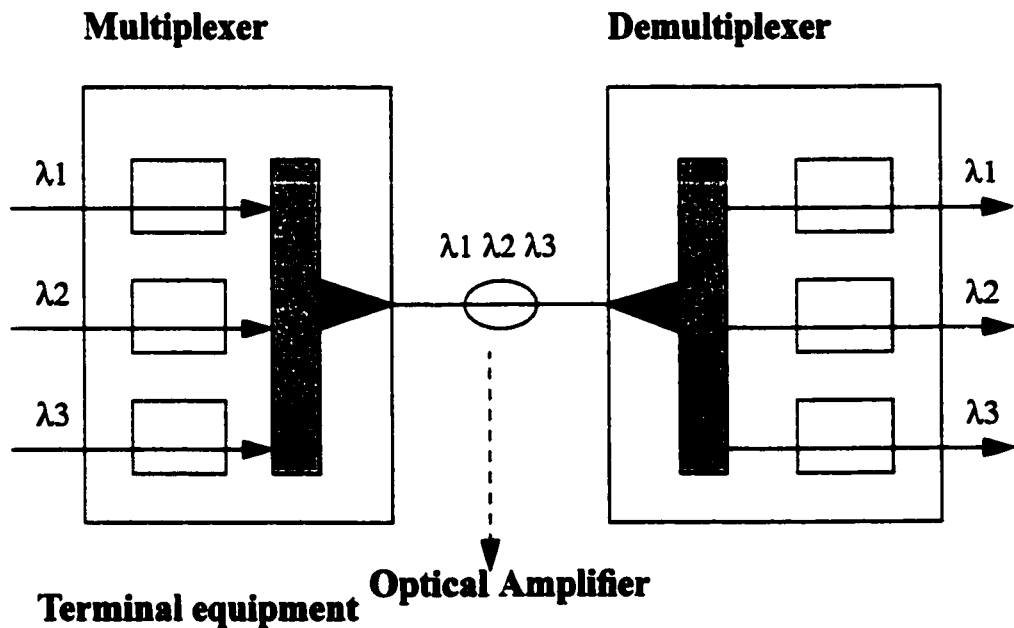
### **2.1.3 Multiplexing**

The bandwidth of optical fibers is in the order of tera bits per second which is much higher than the electronic data transfer rate. In order to fully utilize the bandwidth provided by the optical fiber one has to introduce concurrency among multiple user transmissions into the network architecture. There are two approaches commonly employed to achieve this concurrency. They are time division multiplexing (TDM) and Wave Length Division Multiplexing (WDM). In TDM, the emphasis is on increasing the bit rate and in WDM, the objective is to pack several wavelengths on a single fiber. Essentially, in WDM several lower speed streams at different wavelengths are multiplexed into a higher speed stream [Ger1].

## **2.2 Wavelength Division Multiplexing**

Wavelength Division Multiplexing (WDM) technology offers the capability of building very large wide area networks consisting of thousands of nodes with per node throughputs of the order of a giga bit per second [RS3]. In WDM optical networks the huge bandwidth available on a fiber is partitioned into several channels where each channel operates at a different wavelength. In other words, it allows the transmission of multiple data streams independently on a single fiber. "It is anticipated that the next generation of the internet will employ WDM based optical backbones" [Muk1].

A WDM system uses a multiplexer at the source to multiplex several wavelengths on the fiber and demultiplexes at the receiving end with the help of a demultiplexer. In fig 2.1, the working of a WDM based network is illustrated by using 3 terminal equipments operating at different wavelengths  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$ . The multiplexers at the source combines light signals coming from the 3 ports and transports it across the fiber. At the receiving end, the demultiplexers split the bundled signal into its constituent wavelengths and routes the signals to the destination nodes.



**Figure 2.1: Wavelength Division Multiplexing**

### Physical Topology

The physical topology of a network is the physical set of router/end nodes and the fiber optic links connecting them. The light paths are from one end node to another. This topology may be a broadcast star or a bus topology and in some cases a wavelength routing network, which consists of point-to-point links interconnecting several local area networks.

### Light Path

A lightpath can be viewed as a pipe between two end nodes with a wavelength

assigned to it. A lightpath provides single hop connectivity between end nodes.

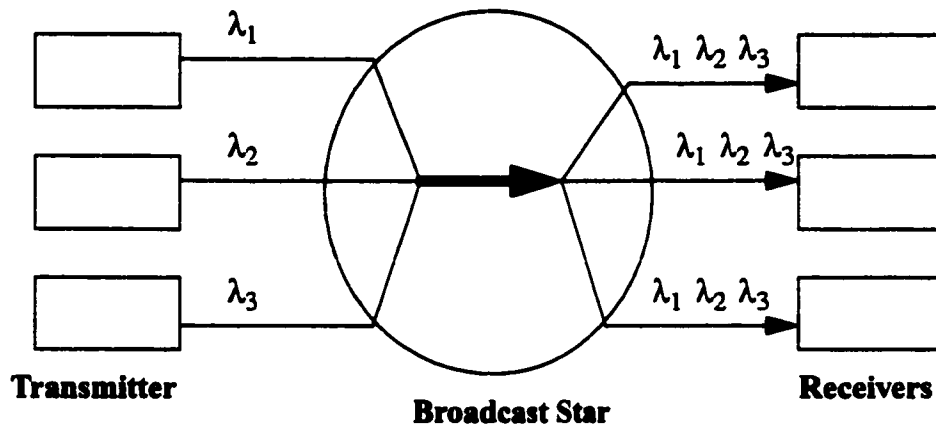
Two light paths that share a link must necessarily use different wavelengths.

### **Logical Topology**

The connectivity pattern that one imposes on the physical topology to provide dedicated connections between select pairs of nodes is referred to as the logical topology. The set of all lightpaths that have been set up on a network constitutes the logical topology of the network. Logical topology can be viewed as a directed graph with end nodes corresponding to vertices and the edges representing light paths. It can be visualized as an optical layer on which high speed packet switched networks can be built [CGK1]. Logical topology is also known as virtual topology as it represents just the communication channels between the source and destination eliminating other intermediary components. Unlike physical topology, logical topology can be easily reconfigured to adapt to changing traffic.

## **2.3 Single-Hop networks**

In a single-hop network (fig 2.1) a signal transmitted from a source reaches its destination directly without going through other end nodes. These networks are all-optical by their nature as signals continue to remain in the optical domain throughout their transmission [Muk1]. This results in very fast communication.



**Figure 2.1 Single hop network (fixed transmitters and receivers)**

However, these networks require a lot of coordination between the source and the destination nodes and are not scalable.

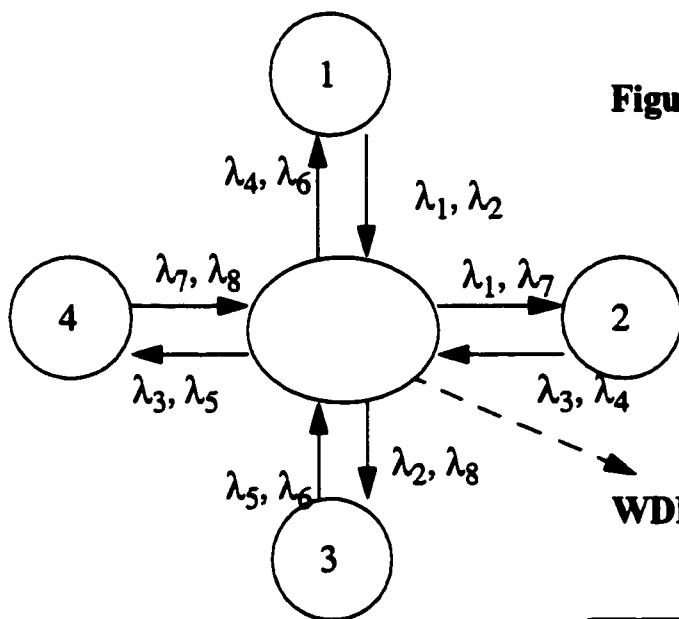
## 2.4 Multi-Hop Networks

In a multi-hop network, information from a source is routed through several intermediate nodes before reaching its destination. This architecture was originally introduced by Acampora in [Aca1]. Multi-hop networks are used in situations where a direct link between a source and a destination node is, in general, not available. Signals that require a multi-hop path are converted to electrical domain at each intermediate node and then converted back to the optical domain for further propagation. Multi-hop networks are not suitable for handling high throughput, real time and delay sensitive traffic [Muk1]. In figures 2.2(a) and 2.3(b) the physi-



cal topology and the corresponding logical topology of a multihop network with 4 nodes is shown.

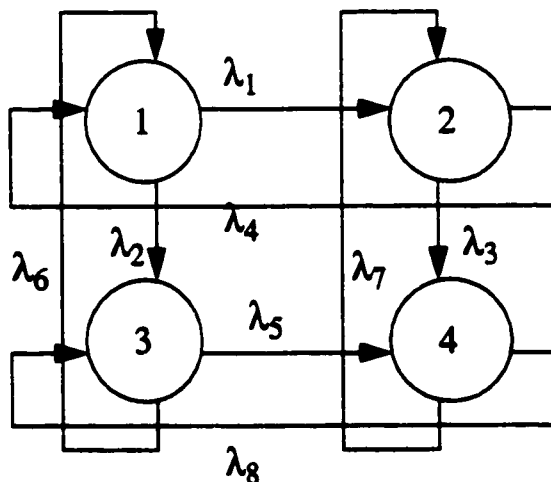
**4 node Multi-hop network**



**Figure 2.2(a) Physical topology**

[Mukl, pp. 172]

**WDM Passive Star Coupler**



**Figure 2.2(b) Logical topology**

[Mukl, pp. 172]

It can be noted that certain communications need more than one hop before they can reach their destination.

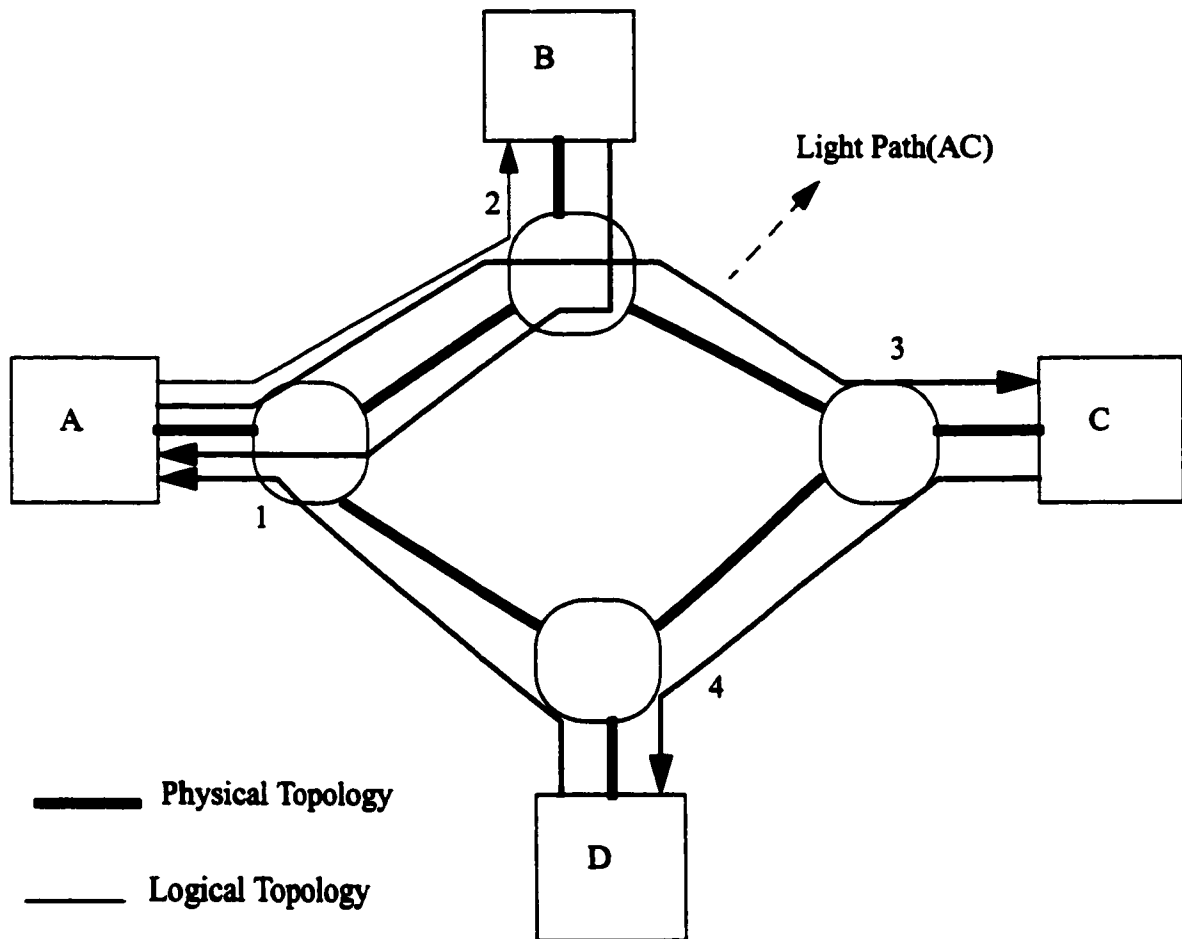
In certain networks, information is selectively transmitted to other nodes based on information received from the source node. Wide area networks operate on this kind of a setup where typically several local area networks are interconnected by point to point links [Raj1]. This type of network constitutes wavelength routing networks and have the benefits of both single-hop and multi-hop networks.

## **2.5 Wavelength Routed Optical Networks (WRON)**

“All optical wavelength division multiplexed networks using wavelength routing are considered to be potential candidates for the next generation of wide area backbone networks” [RS2]

Their main advantage comes from the fact that they avoid the “splitting-loss problems” [RS3] associated with the broadcast topologies and also allow spatial reuse of wavelengths. A wavelength routed optical network can be operated both as a circuit switched and as a packet switched network. In the circuit switched mode, there is no opto-electric conversion at intermediary nodes [Raj1]. The logical topology operates as a packet switched network and exploits the relative strength of both optics and electronics. In these kind of networks, signals usually take multi-hop paths, but continue to remain in the optical domain until packets have to be forwarded between two light paths [CGK1]. These networks are essentially a combination of principles from single-hop and multi-hop networks and they exploit the characteristics of both. Lightpaths provide single hop communication

while communication across lightpaths is achieved by multi-hop paths. Optical circuit switching at intermediary nodes over a light path is done by using a Wavelength Routing Switch (WRS) which routes packets to the next destination without converting the signals to the electrical domain [Gre1].



**Figure 2.3: Wavelength Routed Optical Network (WRON)**

The logical out-degree of an end node is the number of lightpaths that originate

from that node and similarly the logical in-degree is the number of lightpaths that terminate at that node [RS1]

## **2.6 Regular and Irregular Topologies**

The taxonomy of network topologies from a design perspective can be done by classifying the underlying graph. This results in two important sets of topologies namely regular and irregular topologies [Muk1]. A network designed using a regular topology has its components adhere to some pre-defined pattern such as a graph or a mathematical function. Some common examples of regular topologies include ShuffleNet[Muk1] and De Bruijn graph[Muk1]. A network following an irregular topology has no specific structure or connectivity pattern as to how nodes are connected in the network.

## **2.7 Logical Topology Design (WRON)**

“The independence between the logical connectivity and the physical topology, or fiber layout, in multihop lightwave networks provide the freedom to dynamically reconfigure the connectivity diagram” [LA1]. This independence allows the design of large scalable topologies.

In a network with  $N$  nodes it would be ideal if we could set up lightpaths between all the  $N(N-1)$  nodes. This is usually not possible while designing large networks because of two important reasons. Firstly the number of wavelengths available is

limited. It has been shown in [RS2] that on a number of randomly chosen network with 128 nodes with physical degree 4, an average of only 640 paths can be setup using 32 wavelengths. This is much which is much less than the 127 paths required per node. Secondly each node can be the source and sink of only a limited number of lightpaths (say  $\Delta_1$ ). This depends on the amount of information a node can handle and also usually on the transmitters and receivers present at that node [RS2].

When two nodes cannot be connected by a direct path, multiple light paths can be used to allow propagation. When packets take multihop paths they have to be converted to electronic form at intermediate nodes and again reconverted to optical form while re-transmission. The number of signals (say  $\Delta_2$ ) that a node can electronically switch again depends on the hardware, which further imposes constraints on the logical topology. Thus the degree of logical topology can be at most  $\Delta_l = \min(\Delta_1, \Delta_2)$  [RS2]. We notice that even if enough wavelengths are available to setup  $N(N-1)$  light paths, we will not be able to setup the light paths if  $\Delta_l < N - 1$ .

The maximum load offered to a logical link is called congestion. While designing the logical topology it is desirable to minimize both the network congestion and the average packet delay [LA1]. It can be observed that electronic switching

speeds are directly proportional to congestion. The average packet delay experienced at a node is due to queueing delays and the propagation delay. Queueing delays are usually ignored while designing wide area networks as queueing is not an issue in a WAN environment [Raj1]. Hence the problem is to route packets in a fashion which will keep the maximum total flow on any network as low as possible. This mathematical problem is known as the maximum concurrent flow problem [AM01, BG1].

Several mixed integer programming formulations have been proposed in the past with minimal variations [LA1, CGK1, BG1]. The objective of all the formulations that deal with wide area networks is to minimize a form of congestion. All the proposed formulations suffer the inherent limitations imposed by classical algorithms like branch and bound for finding integer solutions. Several heuristic algorithms have been proposed to circumvent this setback.

## **Tabu Search**

### **2.8 Introduction**

The practical significance of several difficult optimization problems has led to the development of several powerful optimization techniques. Many of these newly developed methodologies have their roots derived from existing natural and physical phenomenon. For instance, simulated annealing is based on metallurgy, while genetic algorithms resemble the biology of evolutionary reproduction. Tabu Search is based on concepts that unite the fields of Artificial Intelligence and Operations Research. "Tabu Search is a general heuristic procedure for guiding search to obtain good solutions in complex spaces [Glv1]".

Tabu Search is rapidly getting popular and is being extensively used to solve hard combinatorial problems that frequently occur in practical settings. Various diverse fields have witnessed the successful use of tabu search in recent years [Glv2]. These include several variations of the scheduling problem, character recognition

using neural networks, path assignment in telecommunications among others.

Tabu Search can be viewed as a powerful extension of the standard hill-climbing search defined relative to a neighborhood structure and a move evaluation function. It is based on procedures that cross boundaries of local optimality and explore new regions in quest for the local optimum. The basic ideas behind Tabu Search began to originate in the 1970s and its basic form was recently proposed by Glover. The limitations of several classical methods has further fueled the use of tabu search and its variants in several diverse fields. Tabu Search along with genetic algorithms and simulated annealing has been widely recognized by the Committee on Next Decade of Optimization Research (CONDOR) as an extremely promising technique to solve difficult combinatorial optimization problems[LG1][Glv1].

A meta-heuristic refers to a master strategy that works on top of other heuristic procedures in an attempt to guide the lower level heuristic procedures beyond local optimality. Tabu Search, in essence, is a meta-heuristic that uses memory based strategies to impose certain restrictions on the search process and thereby allowing the exploration of otherwise difficult regions. Adaptive memory and responsive exploration are the key features that make Tabu Search intelligent [LG1]. Tabu Search contrasts with other memory-less models which rely on random sampling of neighborhood of promising solutions and population based approaches. "The emphasis on responsive exploration in Tabu Search, whether in a deterministic or



probabilistic implementation, derives from the supposition that a bad strategic choice can yield more information than a good random choice” [LG1]. It is also superior to other techniques like the A\* algorithm [RK1] and branch and bound technique that offer a rigid form of memory.

## **2.9 Memory and Tabu Search**

Memory in Tabu Search can be both explicit and attributive. Explicit memory records complete solutions that appear promising. This information can be used to explore the neighborhood of these elite solutions at a later stage. This form of memory is memory intensive and requires very clever data structures. Attributive memory, on the other hand, is primarily aimed at guiding the search. This type of memory structure records information about the attributes that change while moving from one solution to another. This can be used to avoid cycles and to guide the search to unexplored regions in the search space.

### **2.9.1 Short-term memory**

“The core of tabu search is embedded in it short-term memory process” [Glv2].

Both, long-term and short-term memory are intended to modify the structure of the neighborhood of the current solution. Short-term memory modifies the neighborhood to a subset of it while long-term memory can add new solutions not originally found in the given neighborhood. This dynamically changing neighborhood

structure is one of the key features behind tabu search.

Short-term memory facilitates aggressive exploration of a given neighborhood resulting in the selection of the best available move. We mentioned earlier that tabu search is an extension of the traditional hill climbing algorithm. A general form of hill climbing algorithm is stated below.

```
Choose an initial solution S  
Set move:= true  
Do  
    Generate a candidate subset  $V^*$  of solutions found  
    in the neighborhood  $N(S)$  of S.  
    Find  $S' \in V^*$  such that  $f(S') \geq f(x)$ ,  $\forall x$  in  $V^*$   
    If  $f(S) > f(S')$   
        S:= S'  
    else  
        move:= false  
until (move = true)
```

This algorithm is often bound to get stuck at a local optimum. In order to overcome this, our guidance structure should be able to accept moves that are not favorable (i.e.  $f(s') < f(s)$ ) and simultaneously avoid cycles.

Tabu Search alleviates the problem of local optimality by employing a tabu list which forces the search away from moves selected in the recent iterations. This notion of using memory to forbid moves has a sharp impact on the solution neighborhood that is being generated. The solution neighborhood actually depends on the time stream and hence on the iteration number [Glv1].

### **2.9.2 Tabu list and Tabu tenure**

A tabu list can be perceived as a FIFO (First-In First-Out) list based on certain attributes of the  $k$  preceding moves (The parameter  $k$  is usually associated with the number of iterations completed). The list of such attributes are recorded in the same sequence in which their corresponding solutions are generated. A move which should be forbidden (or made tabu) is added to the tabu list and it continues to be in the list for a specified number of iterations. The period for which a move remains tabu is usually referred as the tabu tenure of that move. Depending on the nature of the problem, tabu tenure may remain constant throughout the search or may dynamically vary as the search progresses.

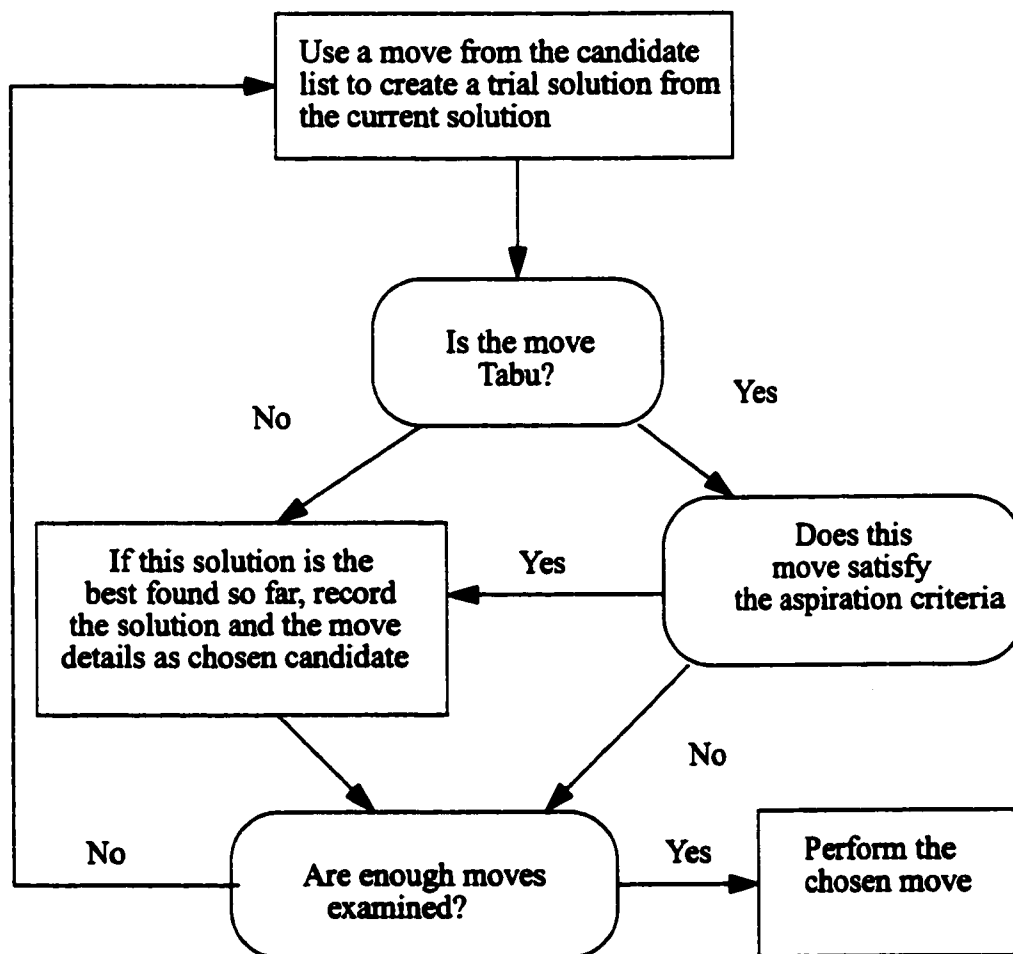
Intuition tells us that the length of tabu list should grow with the size of the problem under consideration. There is no general rule which tells us the size of tabu lists for a given problem. The size is determined empirically. One should watch for

the occurrence of cycles with low values of tabu size and also note the maximum size that can be reached without compromising on the quality of solution obtained. An ideal tabu size should lie between these bounds [LG1].

## **2.10 Candidate list strategy**

The idea of examining the complete neighborhood of a solution is desirable as it will yield high quality solutions. But in a practical setting, this may turn out to be extremely time consuming in terms of CPU time and in several cases may not be feasible at all due to inherent limitations on the amount of computing power available. In such situations we are forced to analyze a subset of moves present in the solution neighborhood.

Several strategies have been proposed in literature[LG1] which provide broad guidelines for selecting a elite subset of promising moves from the huge set of available moves. This elite set of moves is referred to as candidate list.

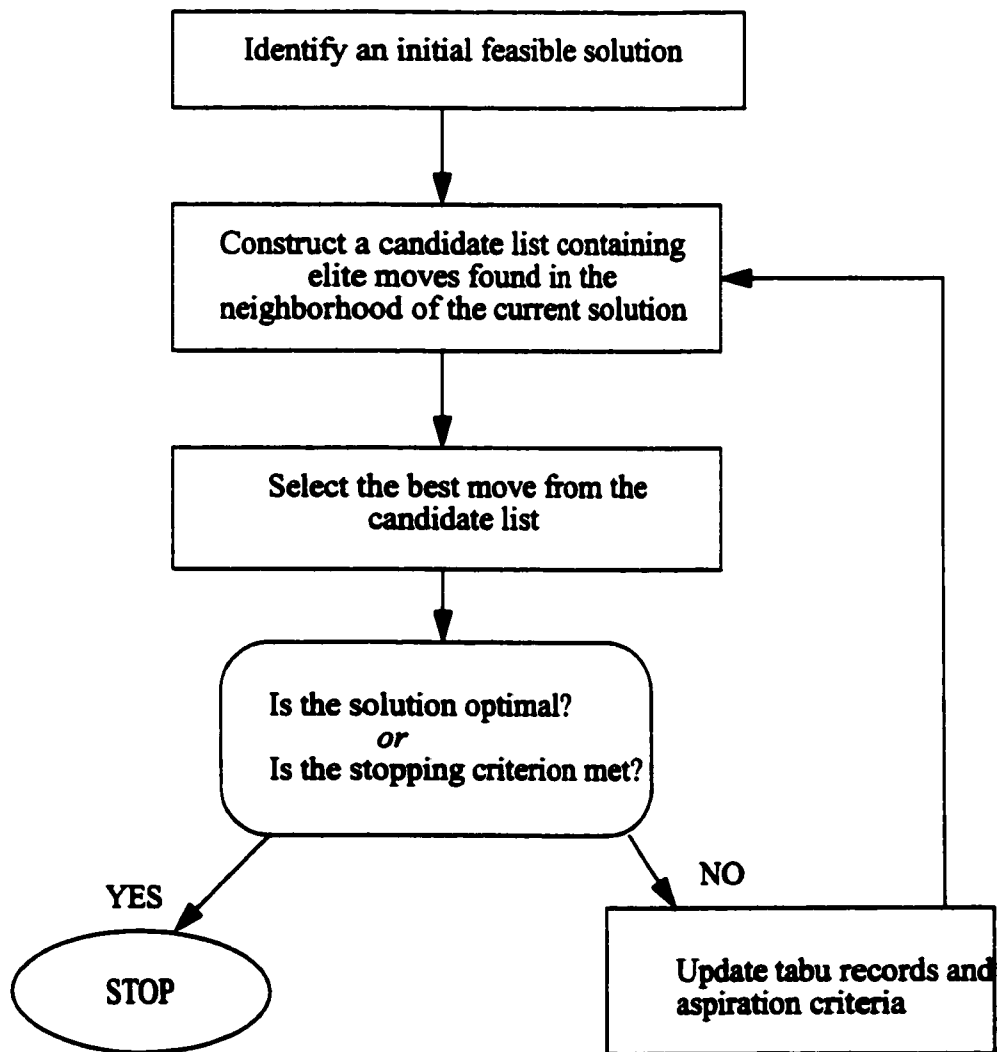


**Best move selection - Short-term memory**

## 2.11 Aspiration Criteria

A closer look at the tabu restrictions will reveal the fact that it might sometimes become too restrictive by forbidding moves leading to unvisited solutions that are attractive. Thus, there may be situations when we want our guidance procedure to

override the tabu status of a move and accept it. This is performed by employing aspiration level conditions. The aspiration level component allows the guidance procedure to override tabu status if a particular move is sufficiently attractive. The aspiration criteria further ensures that the search does not get into cycles [Glv2].



**Tabu Search using short term memory**

In several commercial applications of tabu search, the use of short-term memory has itself found superior solutions[Glv2]. Still, long term memory is useful in obtaining good solutions for hard problems. Long term memory is primarily used to intensify or diversify the search.

## **2.12 Long-term memory**

The scope of memory used to implement tabu conditions operate on a very short term horizon and is primarily intended to prevent cycling. Memory in tabu search can also be used to learn more about the properties of good solutions as they are visited during the search. This notion of learning forms the motivation for introducing intensification and diversification schemes for the search. In some literature this approach is referred as Adaptive Memory Programming (AMP) [LG1]. We mentioned earlier that the purpose of both long-term and short-term memory is to create modified neighborhoods of the current solution. The modified neighborhood in case of short-term memory is always a subset of the neighborhood of the current solution. Long-term memory can include moves that are not found in the immediate neighborhood of the current solution. This is an important difference between long-term and short-term memory.

### **2.12.1 Intensification and Diversification**

The purpose of intensification strategies is to reinforce move combinations and

solution features historically found good [Glv2]. The idea is to favor moves that have properties similar to the ones that were identified good by a ranking mechanism. Intensification alone is not sufficient to yield best solutions as it tries to intensify search in a particular region. To offset this the complementary notion of diversification is introduced which directs the search over unexplored regions in the search space. Diversification is particularly useful in escaping local optimality. A finer look at the short-term memory strategies will reveal the fact that they embody the fundamentals of intensification and diversification.

Several other techniques like strategic oscillation and target analysis are of late finding place in literature and are used in association with long-term memory to constructively drive the search towards global optimum.

Experiments performed recently have demonstrated the ability of tabu search to produce superior solutions that match or even surpass the best known solutions of certain hard problems[Glv2]. Tabu search is a relatively new area of research and many more advances can be expected in the near future.



## Chapter 3

# Problem Specification and Approach

### 3.1 The Problem

We are interested in designing the logical topology of an optical network, given a traffic matrix and the underlying physical topology. We assume that the networks considered here use Wavelength Division Multiplexing (WDM) and wavelength routing. These type of networks, as noted earlier are referred to as Wavelength routed Optical Networks(WRON).

In [RS2] the authors have proposed a mixed integer linear problem (MILP) formulation for the logical topology design problem. Given  $(\lambda^{ij})$  the traffic matrix and the physical topology of the underlying network, the MILP produces the logical topology and the routing strategy that is to be used on it by minimizing the maximum traffic flowing on any logical edge. The MILP gets computationally intractable for even moderately sized networks due to the presence of a large number of integer (0-1) variables. Classical methods like branch and bound and other cutting plane

approaches fail to solve the problem in a reasonable amount of time and require enormous amount of computing power.

In this thesis we investigate the use of tabu search meta-heuristic to solve the MILP, hence producing the logical topology and a corresponding routing algorithm. We first state the MILP to be solved[RS2] and then describe our approach using tabu search. This formulation involves a multi-commodity flow problem, with each pair of nodes representing a commodity flowing over the network.

### **Parameters**

- $T = (\lambda^{ij})$  represents the traffic matrix. i.e.,  $\lambda^{sd}$  represents the amount of traffic/unit time originating from source node  $s$  and having destination node  $d$ .
- $d_{ij}$  the shortest path distance between nodes  $i$  and  $j$ .
- $d_{\max}$  represents the maximum distance between any two source destination pairs.
- $\Delta_l$  denotes the logical degree of the network.

### **Variables**

- $b_{ij}$  are binary variables such that  $b_{ij} = 1$  if there is a logical edge (light path) between node  $i$  and  $j$ , and 0 otherwise;
- $\lambda_{ij}^{sd}$  represents the amount of traffic, flowing on logical link  $ij$ , that originated from source node  $s$  and having destination node  $d$ ;

- $\lambda_{\max}$  is the maximum load on a logical link. This is the congestion on the network which we are trying to minimize;

Minimize  $\lambda_{\max}$  subject to:

Flow conservation at each node:

$$\sum_j \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} \lambda^{sd} & \text{if}(s = i) \\ -\lambda^{sd} & \text{if}(d = i) \\ 0, & \text{otherwise} \end{cases} \quad \text{for all } s, d \text{ and } i.$$

These constraints ensure flow conservation of each commodity at each node. They are further illustrated in figure 3.1 below.

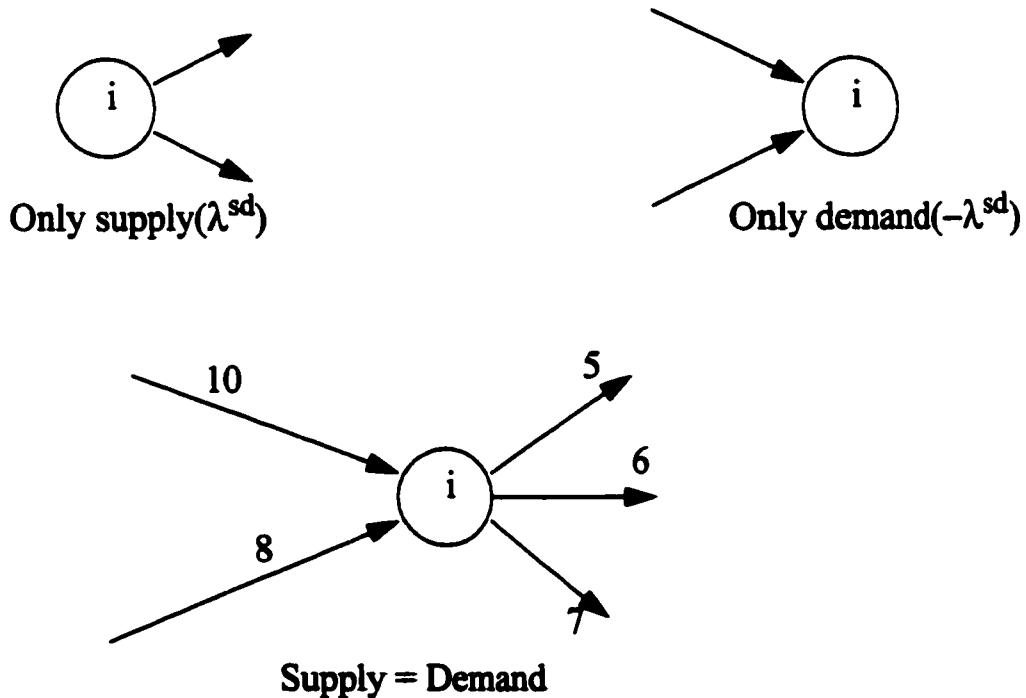


Figure 3.1: Flow Conservation

*Total flow on a logical link:*

$$\sum_{sd} \lambda_{ij}^{sd} \leq \lambda_{max} \text{ for all } i, j$$

$\lambda_{max}$ , the congestion represents the maximum load on any logical link. The above constraints ensure that the total flow on any logical link is never greater than  $\lambda_{max}$

The sum on the left denotes the total traffic flowing on logical link  $ij$ .

$$\lambda_{ij}^{sd} \leq b_{ij} \lambda^{sd} \text{ for all } i, j, s, d$$

These constraints ensure that the flow of a commodity on a logical link is never greater than the total supply of that commodity. Moreover, if  $b_{ij}=0$ , then  $\lambda_{ij}^{sd}$  is forced to be zero for all  $sd$  pairs. The inequality becomes a equality if there is a logical link between  $i$  and  $j$  and the entire traffic flowing from source  $s$  to destination  $d$  flows on it.

*Average delay constraint:*

$$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \lambda^{sd} \alpha_d^{max} \text{ for every } s, d$$

These constraints impose bounds on the delay experienced by every transmission.

$\alpha_{d,max}$  is the maximum delay allowed. Higher values of  $\alpha$  will tend to introduce multi-hop transmission and hence lower congestion.

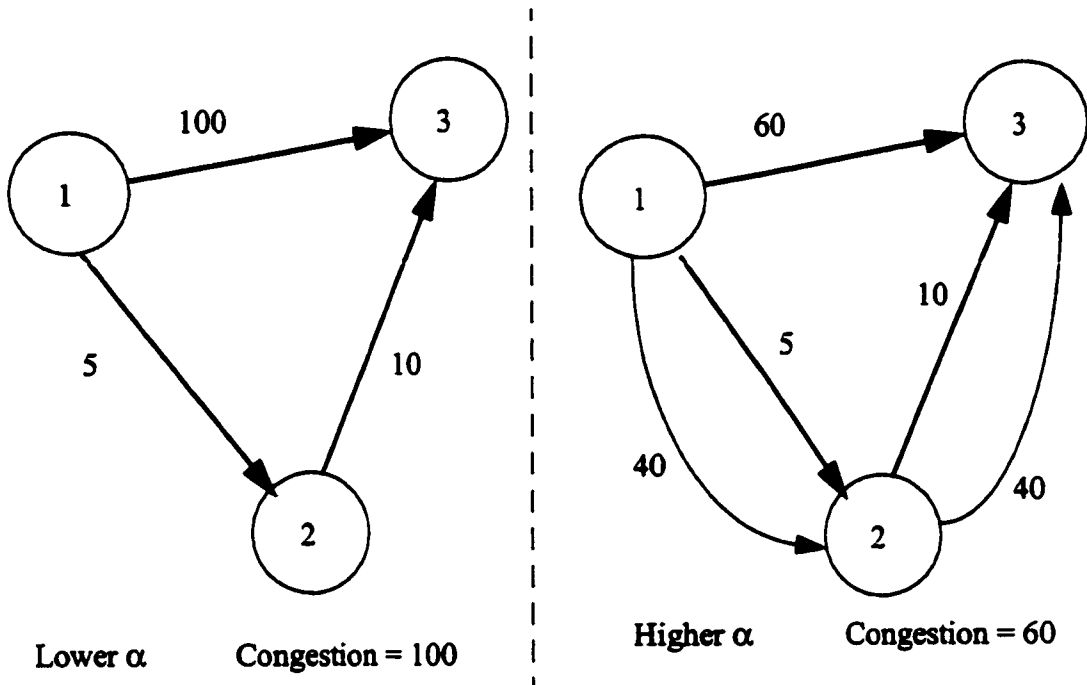


Figure 4.2 Average delay

*Degree constraints:*

$$\sum_i b_{ij} = \Delta_l \text{ for all } j$$

$$\sum_j b_{ij} = \Delta_l \text{ for all } i$$

The degree constraints ensure that the in-degree and out-degree bounds are not violated. These constraints are dependent on the type of switching equipment like routers and the number of transmitters and receivers present at every node.

It naturally follows that  $\lambda_{ij}^{sd}, \lambda_{ij}, \lambda_{max} \geq 0$  for all  $i, j, s, d$  and

$$b_{ij} \in \{0, 1\} \text{ for all } i, j$$

The number of variables and constraints in the above MILP increases exponentially as the number of nodes in the network increases. For a 14 node network there are more than 35000 constraints and twice as many variables(including slack and surplus). There are 182 integer 0-1 variables which further complicate the problem. This makes the problem tremendously difficult even for the most modern computer.

### **3.2 Aggregate formulation**

In the formulation we discussed earlier every source destination pair in the demand matrix constituted a commodity. In [BG1] the authors have introduced a different approach for formulating this problem, commonly referred to as aggregate formulation. In the aggregate formulation, every source in the network is assumed to constitute a commodity. In other words, every row in the traffic matrix will represent a commodity resulting in a significant reduction in the size of the problem in terms of both the number of variables and constraints. For a 14 node network we only have 5475 variables and 2968 constraints which is a significant reduction in size compared to the disaggregate formulation. Still, the disaggregate formulation discussed earlier is superior to aggregate formulation as it provides improved bounds and hence produces high quality solutions.

The size of the problem, in terms of the number of variables and constraints, when formulated using aggregate and disaggregate formulation for a  $n$  node network is given below:

**Table 2: Aggregate vs. Disaggregate formulation**

	Variables	Constraints
Disaggregate	$2n^4 - 4n^3 + 5n^2 - 3n + 1$	$n^4 - n^3 + 2n^2$
Aggregate	$2n^3 - n + 1$	$n^3 + n^2 + 2n$

We will now discuss the logical topology design problem using aggregate formulation. We will be solving this MILP using tabu search.

**Minimize**  $\lambda_{max}$  **subject to:**

*Flow conservation at each node:*

$$\sum_j \lambda_{ij}^s - \sum_j \lambda_{ji}^s = \begin{cases} \lambda^s & \text{if}(s = i) \\ -\lambda^s & \text{if}(s \neq i) \end{cases} \quad \text{for all } s \text{ and } i.$$

*Total flow on a logical link:*

$$\sum_s \lambda_{ij}^s \leq \lambda_{max} \quad \text{for all } i, j$$

$$\lambda_{ij}^s \leq b_{ij} \lambda^s \quad \text{for all } i, j, s$$

*Average delay constraint:*

$$\sum_{i,j} \lambda_{ij}^s d_{ij} \leq \lambda^s \alpha d^{max} \quad \text{for every } s$$

*Degree constraints:*

$$\sum_i b_{ij} = \Delta_l \quad \text{for all } j$$

$$\sum_j b_{ij} = \Delta_l \quad \text{for all } i$$

$$\lambda_{ij}^s, \lambda_{ij}, \lambda_{max} \geq 0 \quad \text{for all } i, j, s \text{ and } b_{ij} \in \{0, 1\} \quad \text{for all } i, j$$



### Example

We consider the following traffic matrix:

$$T = \begin{bmatrix} 0 & 3 & 6 \\ 1 & 0 & 9 \\ 4 & 8 & 0 \end{bmatrix}$$

As noted earlier in the disaggregate formulation every non zero entry in the traffic matrix corresponds to a commodity flowing on the network. For example,  $\lambda^{01}$  represents a commodity and for the traffic matrix stated above, 3 units of commodity  $\lambda^{01}$  flows from node 0 to node 1. Hence, for transmissions coming out of node 0 ( $\lambda^{01}$  and  $\lambda^{02}$ ) we have 6 equations pertaining to flow conservation:

$$\lambda_{01}^{01} + \lambda_{02}^{01} - (\lambda_{10}^{01} + \lambda_{20}^{01}) = 3$$

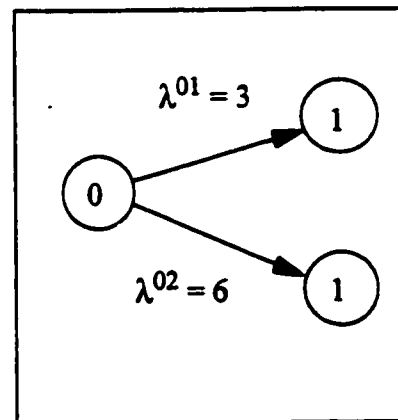
$$\lambda_{10}^{01} + \lambda_{12}^{01} - (\lambda_{01}^{01} + \lambda_{21}^{01}) = -3$$

$$\lambda_{20}^{01} + \lambda_{21}^{01} - (\lambda_{02}^{01} + \lambda_{12}^{01}) = 0$$

$$\lambda_{01}^{02} + \lambda_{02}^{02} - (\lambda_{10}^{02} + \lambda_{20}^{02}) = 6$$

$$\lambda_{10}^{02} + \lambda_{12}^{02} - (\lambda_{01}^{02} + \lambda_{21}^{02}) = -6$$

$$\lambda_{20}^{02} + \lambda_{21}^{02} - (\lambda_{02}^{02} + \lambda_{12}^{02}) = 0$$



But, in the aggregate formulation every source in the traffic matrix correspond to a

commodity. In the traffic matrix above,  $\lambda^0 = 9$  and represents the commodity produced at node 0. For transmissions coming out of node 0, we have only 3 equations pertaining to flow conservation:

$$\lambda_{01}^0 + \lambda_{02}^0 - (\lambda_{10}^0 + \lambda_{20}^0) = 9$$

$$\lambda_{10}^0 + \lambda_{12}^0 - (\lambda_{01}^0 + \lambda_{20}^0) = -3$$

$$\lambda_{20}^0 + \lambda_{21}^0 - (\lambda_{02}^0 + \lambda_{12}^0) = -6$$

The aggregate formulation is clearly more tractable compared to the disaggregate version. This is a significant step towards obtaining a solution compared to the disaggregate formulation. However, solving this problem using branch and bound is extremely difficult due to the overhead involved in terms of time and computing resources.

The logical topology design problem can be divided in to two sub-problems:

- fixing the logical edges in the network
- design of a routing strategy that is to be adopted on the network.

In [RS1] the authors have proposed some heuristic algorithms to solve the logical topology design problem and then plug-in the values to the linear program to find the routing algorithm. The use of heuristic algorithms does not yield consistent results and the quality of results vary with the type of the network considered. We

have discussed about a heuristic topology design algorithm in appendix-3.

### **3.3 Our Approach**

Tabu search in the past has been successfully experimented with problems involving several integer variables. In [GL], the authors have proposed a novel approach to solve mixed integer linear programs using tabu search. This work is the motivation for this thesis. We apply a modified version of this strategy to solve the aggregate version of the MILP stated earlier. We use simplex method with LU decomposition to solve the relaxed integer linear program as it offers good control over handling the round-off errors that occur very frequently while performing arithmetic with single and double precision numbers.

The tabu search heuristic proposed in [LG1] has its framework embedded directly into the bounded simplex method. Essentially it appears as a natural extension of the primal simplex algorithm. It is a well known result that an optimal solution for a 0-1 mixed integer linear program, if one exists, can be found at an extreme point of the LP feasible set[LG1]. The heuristic proposed in [LG1] exploits this important result and it uses the bounded simplex algorithm to move to adjacent extreme points in its quest to find an integer solution.

A broad overview of the algorithm is given below:

- Step 1: We solve the LP relaxation of the original MILP to obtain an optimal extreme point solution. This will be the initial solution to the TS heuristic. Let  $z^*$  be the associated objective value.
- Step 2: We examine all the feasible pivot moves from the current solution. It may be desirable to examine a candidate subset of the available moves. If a move produces a MILP solution  $x$ , with an objective of  $z > z^*$ , record  $z^*$  and the corresponding solution  $x$ .
- Step 3: We rank all the candidate moves based on a TS move evaluation function which penalizes moves based on their tabu status. We select the move with the highest evaluation.
- Step 4: We perform the pivot move using the selected move. We also update the memory and guidance structures and move to step 2.

### **3.3.1 Candidate List**

We examine the entire set of available feasible extreme points. It is highly desirable to analyze a candidate list especially when dealing with large networks as the number of extreme points available can become too high.

### **3.3.2 Guidance mechanism**

A proper guidance mechanism is the key to the TS algorithm. We use two mea-

measures 1) Objective function change, 2) Integer infeasibility measure, to guide us towards a MILP solution. The integer infeasibility measure (IFM) represents the degree to which a solution fails to meet the integer restrictions. For a given extreme point solution  $x(h)$ , let  $x_j(h)$  denote the value assigned to  $x_j$  and let  $z(h)$  denote the objective value. Let  $I$  denote the set of integer variables. IFM is defined as follows:

$$u(h) = \sum (u_j(h) : (j \in I))$$

$$\text{where } u_j(h) = |x_j(h) - \text{near}(x_j(h))|^p$$

$\text{near}(x)$  denotes the integer value nearest to  $x$ . For our study we fixed the value of  $p$  at 1. We note that  $u(h) = 0$  implies that  $x(h)$  is a perfect integer solution. Now, to define a move evaluation scheme for guiding the search, we consider the following:

$$\bullet \Delta z(h) = z(h) - z(0);$$

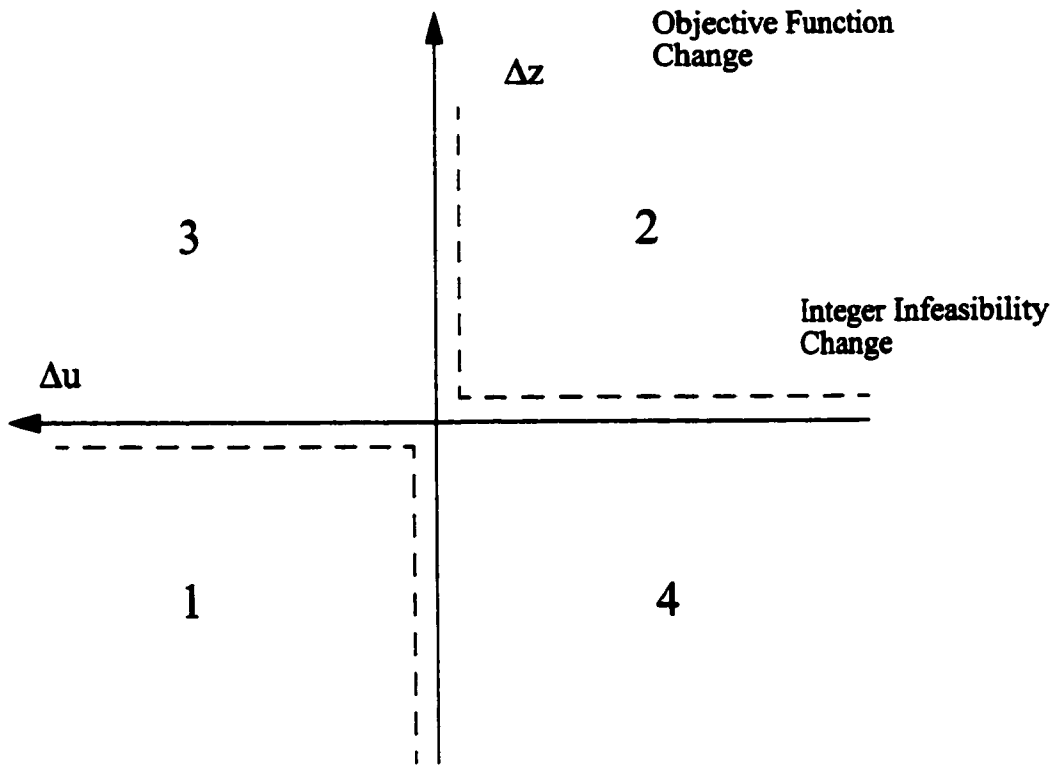
$$\bullet \Delta u(h) = u(0) - u(h);$$

$\Delta z(h) > 0$  indicates a direction of objective increase and  $\Delta u(h) > 0$  indicates a direction of integer feasibility.

### **3.3.3 Move classification**

Based on the measures  $\Delta z(h)$  and  $\Delta u(h)$  moves can be classified in to 4 categories. In Fig. 3.1 each of the quadrant represent a particular move type. Let M1, M2, M3

and M4 represent the set of move type 1, 2, 3 and 4 respectively.



**Fig 3.1 Move classification**

[GL pp 628]

Move type 1 takes the search in a direction which produces better integer solution and decreasing objective value.

$$M1 = \{h \in \text{Moves} / \Delta z(h) < 0; \Delta u(h) > 0 \text{ and } x(h) \text{ is not tabu}\}$$

Similarly, move type 2 points in a direction which increases profit and integer infeasibility. Move type 3, represents non-decreasing profit and nonincreasing inte-

ger infeasibility and move type 4 indicates decreasing profit and nondecreasing integer infeasibility.

### **3.3.4 Move evaluation**

We use ratio test choice rules described below to evaluate the moves. When several moves of a certain type is available we choose the best move by the following choice rules.

- Move type 1:  $m1 = \operatorname{argmax} \{ \Delta z(h) / \Delta u(h) : h \in M1 \}$
- Move type 2:  $m2 = \operatorname{argmax} \{ \Delta u(h) / \Delta z(h) : h \in M2 \}$
- Move type 3:  $m3 = \operatorname{argmax} \{ \Delta z(h) * \Delta u(h) : h \in M3 \}$
- Move type 4:  $m4 = \operatorname{argmin} \{ \Delta z(h) / \Delta u(h) : h \in M4 \}$

### **3.3.5 Move selection**

Move type 3 is most desirable, provided both  $\Delta z$  and  $\Delta u$  are positive. In fact we prefer to choose move type 3 if at least one of  $\Delta z$  or  $\Delta u$  is positive. We accept moves with  $\Delta z$  and  $\Delta u$  equal to zero with a very low probability. In the absence of move type 3 with at least one of  $\Delta z$  or  $\Delta u$  with positive values, we focus on move type 1 and 2. If only one of move type 1 or 2 exists, we choose that move. When both move types 1 and 2 exist we choose move type 1 with a high probability(0.9)

as this bias has the ability to move the search towards an integer feasible solution. Otherwise we choose one of  $m_1$  or  $m_2$  by a normalization procedure described in appendix-2.

Move type 4 is least desirable and is chosen only in the absence of other move types. It is suggested in literature[LG1] that move type 4 can be used to introduce a form of diversification in the search which has the ability to explore new regions of feasibility. We allow move type 3 with  $\Delta z=0$  and  $\Delta U_h=0$  to precede other moves with a very low probability as this has the ability to prevent a bad move.

### **3.3.6 Tabu conditions**

We use a one dimensional array,  $\text{TabuStart}[j]$ , to record the tabu status of a move.

We initialize the entire array with -1. During a pivot move if  $x_i$  is the leaving variable then we record,  $\text{TabuStart}[x_i] = \text{current\_iteration}$ . Then  $x_i$  remains tabu (or restricted) for a certain number of iterations which is determined by tabu tenure.

The function  $\text{isTabu}$  shown below illustrates the implementation of tabu condition.



```

Function isTabu(enter)
{
    if (TabuStart[enter] == -1)
        return false
    else if (iteration - TabuStart[enter] < TabuTenure)
        return true
    else
        return false
}

```

We use a dynamic tabu tenure which varies once every 10 iterations. We randomly choose an integer in the range  $[\text{sqrt}[N], 2 * \text{sqrt}[N]]$  and make that the tabu tenure. [LG1].

### **3.3.7 Aspiration criteria**

The restrictive nature of the tabu conditions can block certain promising moves. To escape such a situation we use an aspiration criteria, which allows a promising tabu move to be executed.

We create an array,  $\text{AspireObj}[A]$  where every cell corresponds to an objective value. This involves association of objective values to various integer levels vary-

ing from 0 to  $A_{Max}$ , where  $A_{Max}$  is the size of the *AspireObj* array. We then store the integer infeasibility level corresponding to an objective value in its cell. This way we can select a tabu move if it shows an improvement in IFM.

Let initial objective value be  $z'$ . We make it correspond to a integer level of  $0.01 * z'$ ; This gives us a conversion factor  $C = (0.01 * L_{Max}) / z'$ ; A solution with objective value  $z(k)$  will correspond to a integer level given by:

$$\min(\text{ceiling}(C * z(k)), L_{max})$$

For a six node network we fixed  $L_{Max}$  at 8000. We increased  $L_{Max}$  as the number of integer variables increased.

## **Chapter 4**

# **Implementation**

The implementation was done using the Java language. Java was chosen as it is platform independent and object oriented. The implementation has a total of 12 classes and runs to around 5000 lines of code. The program solves the relaxed MILP using LU decomposition and then applies tabu search principles to find an integer solution. We stop as soon as we find an integer solution, though it may be desirable to further investigate the search space for better solutions.

### **4.1 LU decomposition**

Solving large problems involving floating point numbers using computers always produces round-off errors. In some cases these errors can accumulate and result in a totally wrong solution. Our experience with the bounded simplex method revealed that round-off errors can produce erroneous results in this problem.

As an alternative, we tried using revised simplex algorithm which reduces the number of calculations significantly by restricting pivot computations to the work-

ing inverse matrix. We also re-inverted the basis matrix very frequently (even though it was a huge overhead) to stabilize the inverse as it might have accumulated some round of errors during the course of the algorithm. In fact this was our favorite choice and it worked fairly well with small networks (up to 5nodes). Frequent re-inversions failed to help the revised simplex implementation for larger networks. This forced us to look for a new and stable simplex implementation.

LU decomposition turned out to be ideal as it offers good control over round-off errors. The method decomposes the inverse into an upper triangular matrix and a sequence of pivot and permutation matrices. The need for efficient data structures was the key to an effective implementation of this method. LU decomposition is further explained in appendix-1. We also frequently recreated the LU decomposition from the current basis to get rid of any round-off errors present in the matrices.

We use LU decomposition to find the initial basic feasible solution for the relaxed problem. Since we do not have an initial basis to work with we solve the LP using the phase 1/phase 2 method by introducing an artificial basis. We also use LU decomposition to move to adjacent feasible solutions during the tabu search process. Once we identify the objective value for the relaxed problem we make it the starting point for the tabu search algorithm. From this extreme point solution (the current solution) we analyze all the adjacent feasible extreme points and classify them based on the move classification rules described in chapter 3.

One important observation here is that we do not perform the pivot operation for

every feasible pivot move while analyzing the extreme points. We compute the updated column for the entering variable under consideration, and using this we can find the new value of the basic variables without performing the pivot. Moreover we are interested in determining the change in integer variable only. Hence we save lot of time by focussing on integer variables that are present in the basis.

## **Chapter 5**

# **Experiments**

In this chapter we discuss the various experiments we performed to test the implementation. We used the results obtained from branch and bound algorithm to bench mark our results. LINDO [Lin], an optimization package, was used to perform experiments involving the branch and bound algorithm. We also compared our results with the results obtained by using HLDA, a topology design algorithm described in [RS1]. We have provided details of HLDA in appendix-3.

### **5.1 Methodology**

Due to the fact that solving LP for large problems becomes time consuming, we focussed our study on smaller networks. Most of the experiments were performed on networks with six nodes because of our inability to solve large problems using the branch and bound algorithm.

We evaluated our approach based on the following objectives:

- To test our algorithm with different values of  $\alpha$  and  $\Delta_b$ , given a traffic matrix.

- 
- To compare the congestion obtained from our approach with that of branch and bound algorithm using Lindo.
  - To compare the congestion obtained from our approach with HLDA.

We did not do several tests for topologies with logical degree, 1. This is not a very interesting case as we always get a ring topology. Since random numbers are extensively used during the tabu search process, we may not get the same solution after every run. For every test case, we executed our algorithm thrice and then selected the best result.

## 5.2 Test data

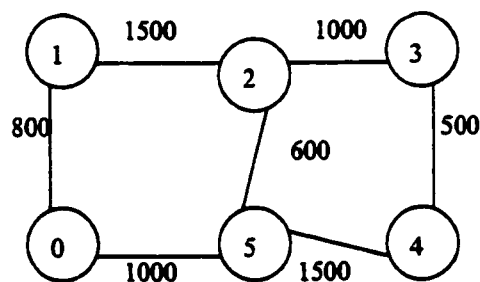
We used a total of eight traffic matrices for the six node network and one each for the seven and eight node networks. We designed topologies with different logical degrees and simultaneously varied  $\alpha$  to study the impact of propagation delay.

HLDA places multiple logical edges between the same pair of nodes if the traffic between them is very high, provided degree constraints are not violated. The MILP formulation we used does not allow multiple logical links between the same pair of nodes. Except in the case of TM1, traffic was chosen in such a way that HLDA doesn't create multiple edges between the same pair of nodes in the logical topology. This allowed us to compare the performance of HLDA with our approach. Traffic matrix entries was randomly generated using an uniform distribution in (0, 1). We ignored several traffic matrices in order to generate a topology that doesn't

carry multiple logical links between the same pair of nodes when HLDA is applied.

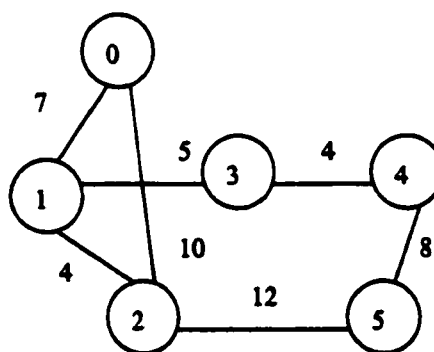
**Traffic matrix TM1 (taken from [RS1])**

$$\text{TM1} = \begin{bmatrix}
 0 & 0.537 & 0.524 & 0.710 & 0.803 & 0.974 \\
 0.391 & 0 & 0.203 & 0.234 & 0.141 & 0.831 \\
 0.060 & 0.453 & 0 & 0.645 & 0.204 & 0.106 \\
 0.508 & 0.660 & 0.494 & 0 & 0.426 & 0.682 \\
 0.480 & 0.174 & 0.522 & 0.879 & 0 & 0.241 \\
 0.950 & 0.406 & 0.175 & 0.656 & 0.193 & 0
 \end{bmatrix}$$



Physical topology

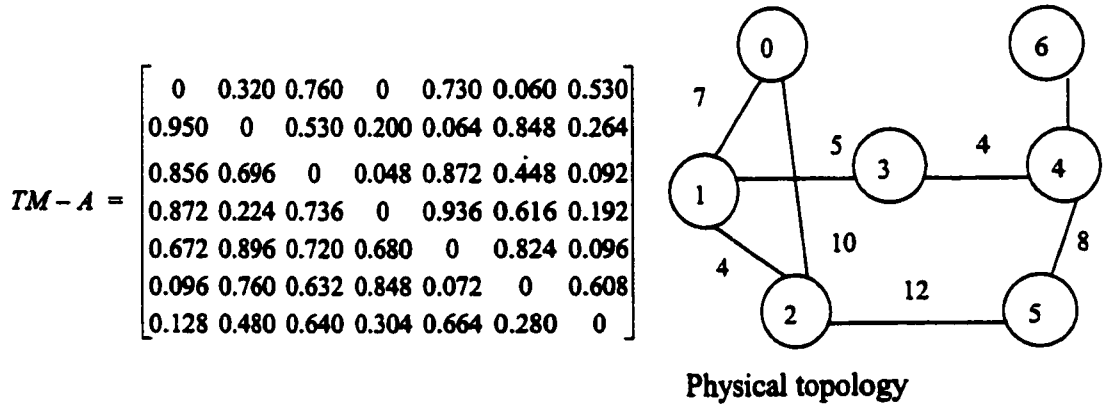
**Traffic matrices TM2 - TM8:** The physical topology for all the seven traffic matrices(TM2 to TM8) is shown below. It is a subset of the 14 node NSFNET which is as a primary internet backbone for North America.



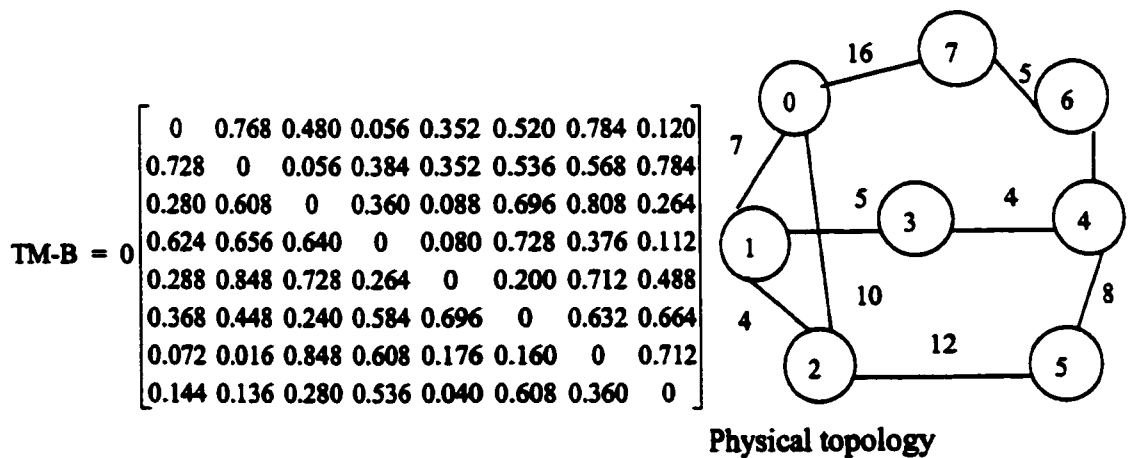
Physical topology



**Traffic matrix TM-A:** This is a seven node network. The physical topology is a subset of the NFSNET.



**Traffic matrix TM-B:** This is a 8 node network and the physical topology is a subset of the NSFNET.



## 5.3 Experiments

We performed 3 set of experiments for every topology designed as follows:

1. We ran the branch and bound procedure to solve the MILP
2. Used our tabu search based approach to solve the MILP
3. Identify the logical edges (0-1 variables) using HLDA and then solved the resulting LP to determine the congestion.

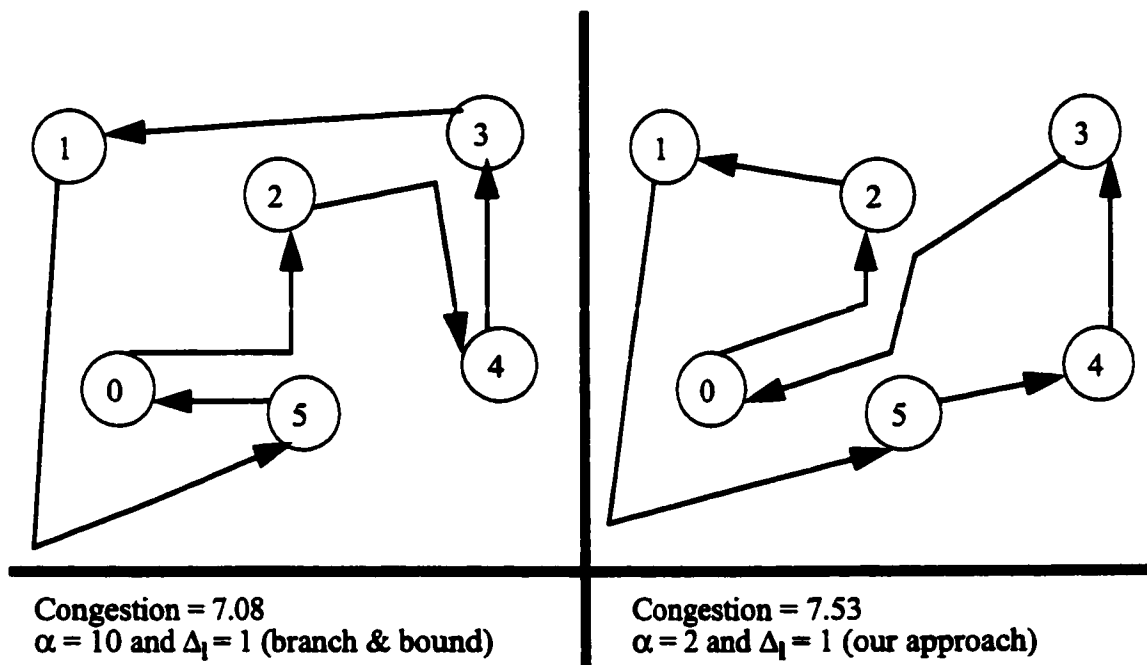
### Experiment 1

In this experiment we designed topologies with logical degree 1, 2 and 3 using TM1.

**Table 3: Congestion obtained for  
TM1 with  $\Delta_l = 1$ .**

	Branch & Bound	Our approach	HLDA
$\alpha = 1$	infeasible	infeasible	infeasible
$\alpha = 2$	7.34	7.53	infeasible
$\alpha = 3$	7.08	7.08	infeasible

As  $\alpha$  increases we noticed a decrease in congestion. This is because of the fact that higher values of  $\alpha$  distributes the traffic by allowing multihop communication.



**Logical topologies using TM1**

**Table 4: Congestion obtained for TM1**

Degree ( $\Delta_1$ )	$\alpha$	Branch & Bound	Our approach	HLDA
2	1	infeasible	infeasible	infeasible
2	2	2.04	2.24	2.19
2	3	2.04	2.04	2.19
3	1	1.21	1.27	*
3	2	1.18	1.27	*
3	3	1.18	1.27	*

(\*) HLDA places multiple logical links between the same pair of nodes

## Experiment 2

In this experiment we designed topologies using TM-A.

**Table 5: Congestion obtained for TM-A (7 nodes)**

Degree ( $\Delta_i$ )	$\alpha$	Branch & Bound	Our approach	HLDA
1	1	infeasible	infeasible	infeasible
1	2	10.576	12.40	infeasible
1	3	1.610	1.803	infeasible
2	1	2.96	3.41	infeasible
2	2	2.96	3.73 <sup>#</sup>	3.68
2	3	2.96	3.01 <sup>#</sup>	3.68
3	1	1.62	2.402	1.84
3	2	1.57	2.192	1.84
3	3	1.57	1.780	1.84

(#) The solution was obtained after several attempts (around 6) as our approach failed to escape cycles. We accepted move type 4 with a very high probability to diversify the search to new areas in the search space.

### Experiment 3

In this experiment we design topologies using TM-B.

**Table 6: Congestion obtained for TM-A (8 nodes)**

Degree ( $\Delta_1$ )	$\alpha$	Branch & Bound*	Our approach	HLDA
1	1	infeasible	infeasible	infeasible
1	2	11.592	12.4	infeasible
1	3	11.390	13.15@	infeasible
2	1	3.10	4.12	infeasible
2	2	3.10	3.50	3.56
2	3	3.10	3.202	3.56
3	1	1.707	1.75	2.05
3	2	1.707	2.1	2.05
3	3	1.707	1.75	2.05

(\*) Best feasible solution obtained after  $10^7$  iterations was chosen.

Cycling was a major problem while working on this 8 node network. Several runs were required to obtain a feasible solution. The need for a longer term memory component was clearly evident as the network size grows. The diversification strategy discussed in the previous experiment did not help in escaping local optimality.

Using a fixed tabu tenure helped in one particular case(@), but in general proved to be too restrictive.

#### Experiment 4

In this experiment we used seven traffic matrices TM1 - TM7, and compared the performance of HLDA with our approach.

**Table 7: Comparative study using 7 six node networks**

Degree ( $\Delta_l$ )	$\alpha$	Our Approach / Branch & Bound		HLDA / Branch & Bound	
		Mean	Standard deviation	Mean	Standard deviation
2	1	1.25	0.18	1.11	0.05
2	2	1.31	0.13	1.101	0.052
3	1	1.286	0.105	1.031	0.024
3	2	1.179	0.091	1.028	0.025

It is interesting to note that HLDA performs much better when the delay constraints are not too restrictive. Our approach did not yield very good solutions mainly because we terminated the search as soon as we come across the first integer solution.

## **Conclusions and Future Work**

In the course of this thesis we presented a tabu search based approach to solve the logical topology design problem. We compared our approach with branch and bound and HLDA. Our approach yields solutions which are comparable with HLDA even though not superior to it in many cases. Our approach gives a solution for any feasible MILP whereas HLDA fails for networks with very tight delay constraints even if a feasible topology exists.

We focussed mainly on providing a basic framework for applying tabu search to the logical topology design problem. We did not address issues like long term memory and more over we stopped our search as soon as we found an integer solution. These factors have definitely affected the quality of our solution.

Another interesting observation would be the behavior of HLDA under relaxed delay constraints. It performs very well under such situations and it will be a good idea to start our approach with an initial solution obtained from HLDA and then improve on it.

Tabu search is an interesting area of research which is relatively new and is rapidly growing. As we worked on this topic we were able to identify several potential areas of research which can be done by extending this work.

In our approach we did not use any candidate list strategy and hence we searched entire neighborhoods in every iteration. Implementing the search using a proper candidate list strategy will definitely speed up the search significantly. It is also desirable to enlarge our candidate list (the entire neighborhood in our case) when move type 4 is only available. We also suggest the use of other tabu search components like long-term memory, strategic oscillation etc. It may be useful to study the behavior of probabilistic tabu search for this problem.

Our formulation did not allow the creation of multiple logical edges between nodes in the logical topology. This can be incorporated in future studies as it has the potential to bring down congestion significantly.

An approach to distribute the problem along several computers can definitely speed up the problem. With the availability of new technologies like RPC and CORBA which aid distribution effectively, a thought in this direction can lead to an excellent research topic.



## **Selected Bibliography**

[Aca1] A. S. Acampora, "A multichannel multihop local lightwave network," Proc IEEE Globecom'87, Nov. 1987.

[AMO1] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, "Network Flows", Prentice Hall, Englewood Cliffs, NJ 1993

[BG1] D. Bienstock and O. Gunluk, "Computational experience with a difficult mixed-integer multicommodity flow problem", Mathematical Programming, vol. 68, pp. 213-237, 1995.

[CGK1] I. Chlamtac, A. Ganz, and G. Karmi, "Lightnets: Topologies for high speed optical networks," IEEE/OSA Journal of Lightwave Technology, vol. 11, pp. 951-961, May/June 1993.

[Ger1] Ori Gerstel, "On the future of wavelength routing networks", IBM T.J. Watson research center, NY, USA

[Glv1] F. Glover, "A user's guide to tabu search ", Annals of Operations Research, pp. 3 -28, 1993

[Glv2] F. Glover, "Tabu search: A tutorial ", Interfaces, vol. 20, pp. 74-94, Aug 1990

[GL1] F. Glover and M. Laguna, "Tabu Search", Kluwer Academic Publishers, 1997

- 
- [Gre1] P. E. Green, *Fiber Optic Networks*, Prentice Hall, Englewood Cliffs, NJ 1992
- [Lin] LINDO user's manual, LINDO Systems Inc., IL, USA, 1996
- [LA1] J. -F. P. Labourdette and A. S. Acampora, "Logically rearrangeable multi-hop lightwave networks," *IEEE Transactions on Communications*, vol. 39, pp. 1223-1230, Aug. 1991.
- [LG1] A. Lokketangen and F. Glover, "Solving zero-one mixed integer programming problems using tabu search", *European Journal of Operational Research*, pp. 624-658, 1998
- [Muk1] B. Mukherjee, "Optical Communication Networks", McGraw Hill Text, 1997
- [Mur1] K. Murty, "Linear and combinatorial programming", John Wiley and sons, 1976
- [Mur2] K. Murty, "Linear programming", John Wiley and sons, 1983
- [Raj1] Rajesh M. K, "Algorithms for routing, wavelength assignment and topology design in optical networks", Ph.D. Thesis, Department of Electrical and Communication Engineering, Indian Institute of Science, Bangalore, India Nov. 1998
- [RK1] Rich. E and Knight. K, "Artificial Intelligence", McGraw Hill College Div. 2nd edition, June 1991
- [RS1] R. Ramaswami and K. N. Sivarajan, "Optical Networks: A practical perspective", Morgan Kaufmann Publishers, 1998

---

[RS2] Rajiv Ramaswami, Kumar N. Sivarajan, "Design of logical topologies for wavelength routed optical networks", IEEE Journal on Selected Areas in Communications", vol. 14, No.5, June 1996

[RS3] Rajiv Ramaswami, Kumar N. Sivarajan,"Routing and wavelength assignment in all optical networks", IEEE Transactions on networking, vol. 3, no.5, Oct 1995

---

---

## Appendix 1

# LU Decomposition

The basic idea behind LU decomposition is to perform row operations in a specific manner, which convert a given basis  $B$  to an upper triangular matrix and a sequence of lower triangular and pivot matrices. The method starts with the first column, swaps the first row, if necessary with the row having the largest coefficient in absolute value in this column. Then row operations are performed to reduce all the coefficients from the 2nd row onwards to zero. This is achieved by pivoting with respect to the first row coefficient. We do this process recursively for all the remaining columns. One should note that the row swapping is done relative to the diagonal element.

Consider the LP in standard form:

$$\text{minimize } Z = CX$$

$$\text{subject to: } AX = B \text{ where } X \geq 0$$

---

Let  $B$  be the current basis under examination. This method generates an upper triangular matrix  $U$ , a sequence of lower triangular matrices  $\{\Gamma_i\}$  and a sequence of permutation matrices  $\{Q_i\}$  such that

$$\Gamma_k Q_k \dots \Gamma_2 Q_2 \Gamma_1 Q_1 B = U$$

This decomposition is called LU decomposition and the product,  $\Gamma_k Q_k \dots \Gamma_2 Q_2 \Gamma_1 Q_1$  is usually denoted by  $L^{-1}$ , so that  $B = LU$ . In the absence of permutation matrices,  $L$  is a lower triangular matrix and hence the name LU decomposition [Mur1].

### Obtaining LU decomposition

Set  $B^0 = B$

for  $r = 1$  to  $m-1$  repeat

Let  $B^{r-1} = (b_{ij}^{r-1})$ ; Select  $s$  such that  $|b_{s,r}^{r-1}| = \max \{|b_{i,r}^{r-1}| / i = r \text{ to } m\}$

Multiply  $B^{r-1}$  on the left by a identity matrix with rows  $r$  and  $s$  switched. Note this permutation matrix  $Q_r$

Add suitable multiples of row  $r$  to row  $i$  so that the entry at  $(i, r)$  is transformed to zero for  $i > r$ . Again, this is equivalent to multiplying the present matrix on the left by a matrix  $\Gamma_r$  of the form shown in fig A1.1. where  $g_{ir} = -a_{ir}/a_{rr}$  for  $i = r+1$  to  $m$ . These matrices differ from the unit only in the  $r^{\text{th}}$  column.

$$\Gamma_r = \begin{array}{|c|c|c|c|} \hline 1 & & & \\ \hline & 1 & & \\ \hline & g_{r+1,r} & 1 & \\ \hline & & & \\ \hline & g_{m,r} & & 1 \\ \hline \end{array}$$

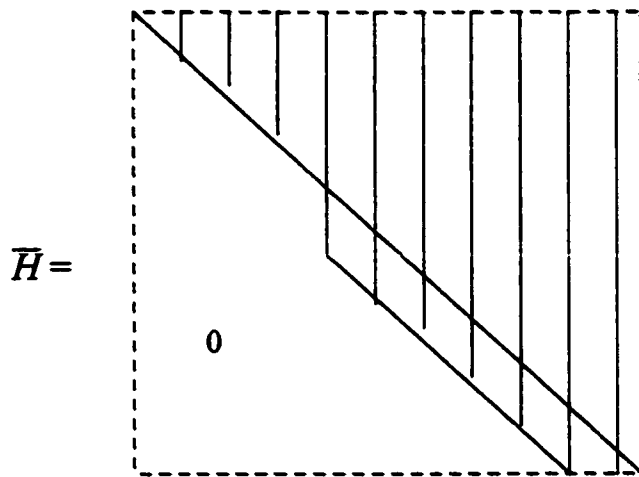
**Fig. A1.1: LU decomposition [Mur2]**

### Updating LU decomposition

Let  $B = (\bar{A}_1, \bar{A}_2, \dots, \bar{A}_m)$ . Let us suppose that  $A_s$  and  $\bar{A}_r$  are the entering and leaving columns. While updating the LU decomposition, the entering column is always treated as the right most column in the new basis. All columns that are to the right of the leaving column are moved one column to their left and the entering column takes position on the right corner. The new basis  $\bar{B} = (\bar{A}_1, \dots, \bar{A}_{r-1}, \bar{A}_{r+1}, A_m, A_s)$ .

Define  $f = \Gamma_k Q_k \dots \Gamma_2 Q_2 \Gamma_1 Q_1 A_s$ ;

Let  $\bar{H} = \Gamma_k Q_k \dots \Gamma_2 Q_2 \Gamma_1 Q_1 \bar{B} = (U_1, U_2, \dots, U_{r-1}, U_{r+1}, \dots, U_m, f)$ . The structure of matrix  $\bar{H}$  is shown in fig. A1.2. A matrix of this form is known as an upper Hessenberg matrix.  $\bar{H}$  can be easily transformed in to an upper triangular matrix (U) by performing row operations described in the previous section on all rows  $k, k > r-1$ .



**Fig. A1.2: Upper Hessenberg matrix [Mur2]**

### **Computations using LU decomposition**

Computing basic variables  $x_b$  involves solving the equation  $Bx_b = b$ ; We first compute  $t = \Gamma_k Q_k \dots \Gamma_2 Q_2 \Gamma_1 Q_1 b$  and then solve  $Ux_b = t$ , by back substitution. A similar approach can be used to compute the updated column  $\bar{A}_s$  which is obtained by solving  $B\bar{A}_s = A_s$ . Let  $\pi_b$  be the reduced cost of the current basis.

Computing  $\pi_b$  involves solving the equation  $\pi_b B = C_b$ , where  $C_b$  represents the cost vector of basis. Compute vector  $h$  from  $hU = C_b$  by back substitution, Then  $\pi_b$  is given by

$$\pi_b = h \Gamma_k Q_k \dots \Gamma_2 Q_2 \Gamma_1 Q_1$$

---

---

## Appendix 2

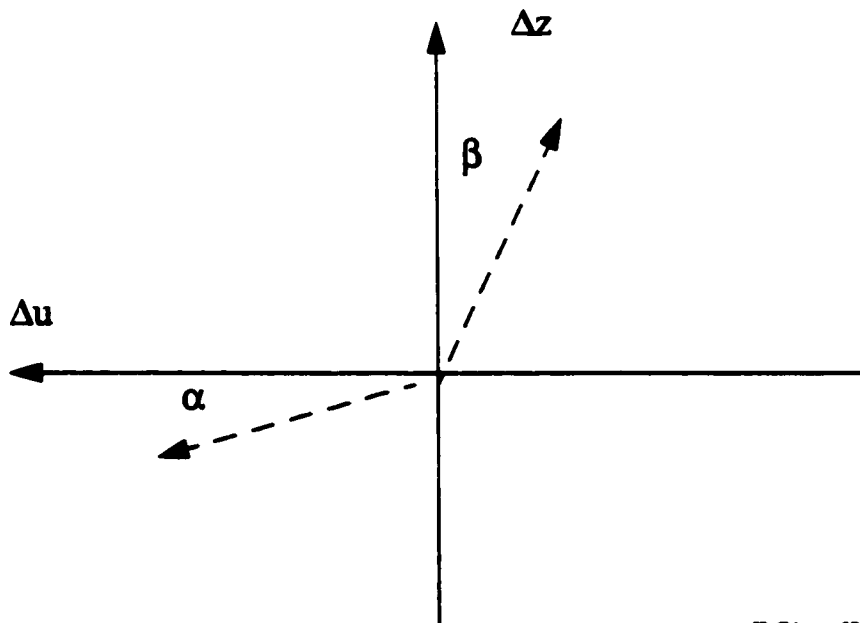
### Normalization

Define  $F(w,q)$  an aggregate integer infeasibility measure as,  $F = \sum_{h \in H} |\Delta u(h)|^q$ .

Define  $R = \left( \sum_{h \in H} |\Delta Z(h)| \right) / (F(w,q))$ ,

$$R1(h) = (\Delta Z(h) / \Delta u(h)) / R$$

$$R1(h) = (\Delta Z(h) / \Delta u(h)) \cdot R$$



**Fig A2.1: Normalization**

[LG1 pp 628]



---

In order to resolve between two moves  $h_1$  and  $h_2$  of type 1 and 2 respectively, we

find  $h^* = \text{Argmax}\{R_1(h_1), R_2(h_2)\}$  [LG1].

In essence we are trying to minimize the angles  $\alpha$  and  $\beta$  illustrated in figure A2.1

---

---

## **Appendix 3**

### **HLDA**

HLDA, stands for heuristic logical topology design algorithm [RS2]. Given a traffic matrix, this heuristic places logical edges between nodes in order of descending traffic. The heuristic is traffic sensitive and ignores delay constraints while constructing the logical topology. A small description of the heuristic is presented below:

---

**Step 1:** Copy the given traffic matrix  $T=(t_{ij})$  to  $Q=(q_{ij})$

**Step 2:** Select the source destination pair with largest traffic.

$$q_{i_{max}j_{max}} = \max (q_{ij})$$

if( $q_{i_{max}j_{max}} = 0$ ) go to step 4

**Step 3:** Subject to wavelength availability and degree constraints we try to place a logical edge between  $i_{max}$  and  $j_{max}$  along the shortest path.

if logical edge is created

Find  $q_{n1,n2} = \max q_{ij}$ , such that  $(n1,n2) \neq (i_{max},i_{max})$

Set  $q_{i_{max}j_{max}} = q_{i_{max}j_{max}} - q_{n1,n2}$

else

$$q_{i_{max}j_{max}} = 0$$

go to step 2

**Step 4:** If total logical edges is less than  $N\Delta_1$  then place as many edges randomly without violating the degree and wavelength availability constraints.

---

## **Vita Auctoris**

**Kannan Achan was born on Sept. 22, 1973 in Tambaram - India. He obtained his BSc and Msc degrees in Mathematics from the University of Madras - India in 1993 and 1995 respectively. He is currently a candidate for the Master's degree in Computer Science at the University of Windsor.**