VIRTUAL TOPOLOGY RECONFIGURATION IN WAVELENGTH-ROUTED OPTICAL NETWORKS

A DISSERTATION IN
Software Architecture
and
Computer Networking

Presented to the Faculty of the University
of Missouri-Kansas City in partial fulfillment
of the requirements for the degree

DOCTOR OF PHILOSOPHY

by
PASSAKON PRATHOMBUTR

B.S., Chiang Mai University, Thailand, 1990
M.S., Chulalongkorn University, Thailand, 1993
M.S., Oklahoma State University, 1998

Kansas City, Missouri
2003

VIRTUAL TOPOLOGY RECONFIGURATION IN
WAVELENGTH-ROUTED OPTICAL NETWORKS

Passakon Prathombutr, Candidate for the Doctor of Philosophy Degree
University of Missouri - Kansas City, 2003

ABSTRACT

Reconfiguration in a Wavelength-Routed Optical Network is a process of rearranging a virtual topology to meet traffic demands that change over a period of time. This dissertation studies a series of reconfigurations corresponding to a series of changes in traffic demand matrices. A change in a virtual topology is costly in terms of traffic disruption. However, without response to this change, the virtual topology would lose its optimality and might not serve the new traffic demand. Therefore, the reconfiguration problem is a trade-off between a performance objective and a cost objective. This research describes the reconfiguration problem from two perspectives. First, the reconfiguration problem is a multi-objective optimization such that a single-objective optimization method could not be applied. Second, the reconfiguration problem consists of a series of reconfigurations with corresponding traffic demands, thus reconfigurations that consider only the current traffic demand cannot guarantee the optimal average outcome. Therefore, sequential decision-making is required to optimize the average outcome from a series of reconfigurations. Since the reconfiguration objectives are conflicting there exists a Pareto front or a set of non-dominated solutions in all objectives. A Multi-Objective Evolutionary Algorithm (MOEA) is required to search the Pareto front and a decision-making process will pick one solution in the Pareto front accordingly. The major contribution of this research is a complete reconfiguration model applicable to any kind of traffic. It is a stochastic model consisting of two tasks: a reconfiguration process and a policy. For each reconfiguration in a series, the reconfiguration process finds a Pareto front and the policy picks a solution from the Pareto front to perform a reconfiguration operation. Our research presents the problem formulation mathematically and the design of the model is based on realistic SONET/SDH traffic streams. We use a MOEA called Strength Pareto Evolutionary Algorithm (SPEA) in the reconfiguration process and use a Markov Decision Process in the policy. A case study based on simulation experiments is conducted to illustrate the application and efficiency of the model. It shows that our model generates a higher average outcome than that of reconfigurations considering only the current traffic demand.

This abstract of 348 words is approved as to form and content.

Eun Kyo Park, Ph.D.
Professor (Co-advisor)
School of Computing and Engineering

Jerry Stach, Ph.D.
Associate Professor (Co-advisor)
School of Computing and Engineering

The undersigned, appointed by the Dean of the School of Graduate Studies, have examined a dissertation titled “Virtual Topology Reconfiguration in Wavelength-Routed Optical Networks,” presented by Passakon Prathombutr, candidate for the Doctor of Philosophy degree, and hereby certify that in their opinion it is worthy of acceptance.

E K Park, Ph.D. Department of Computer Science	Date
Jerry Stach, Ph.D. Department of Computer Science	Date
Yugi Lee, Ph.D. Department of Computer Science	Date
Xiaojun Shen, Ph.D. Department of Computer Science	Date
Nihat Oguz, Ph.D. Department of Computer Science	Date

TABLE OF CONTENTS

ILLUSTRATIONS

TABLES

ACKNOWLEDGEMENTS

I would like to thank my co-advisors, Dr. EK Park and Dr. Jerry Stach for their guidance and support. Beyond the obvious support in the area of my interdisciplinary, they provided the knowledge in scientific method, the professional ways of paper producing and publishing and the real world research.

Gratitude goes to all member of my committee for their help through ongoing discussions and preparation through the research process.

I express sincere gratitude to my wife and my parent for the long support on the fulfillment of my entire education experience at UMKC and my daughter who inspires me to speed up the degree. I would like to thank Thai Student Association members and Alumni especially Ms. Pattareeya Ouppariputtikul and Mr. Nattapong Assalee for hospitality since the first day I arrive in the Kansas City and for a unforgettable friendship.

To My Wife and My Daughter

CHAPTER 1

INTRODUCTION

1.1 General Overview

Wavelength Division Multiplexing (WDM) is a promising approach for extremely high speed communication over an optical fiber. The WDM allows multiple wavelengths carrying multiple traffic sessions to be transmitted over a single fiber. An optical network over which traffic sessions are routed based on their wavelengths is called a wavelength-routed optical network. Such a network consists of optical nodes connected by fibers that form a physical topology. The optical node multiplexes multiple wavelengths into a single fiber transmission and an optical cross-connect (OXC) to switch optical channels called lightpaths. The optical node is an access node if it contains an Optical Add/Drop Multiplexer (OADM) to add/drop traffic. In some particular type of wavelength-routed optical network, it allows the lightpath to operate on multiple wavelengths along the paths. Such a network provides flexibility for a wavelength assignment but it requires costly wavelength converters at the intermediate nodes.

There are two levels of topology in the optical network; a physical topology level and a virtual topology (or logical topology) level. The physical topology is a set of optical fibers linked to each other and the virtual topology is a set of lightpaths that carry optical signals unidirectionally from source to destination nodes according to a traffic demand matrix. Although a lightpath is a virtual view, it requires a routing and wavelength assignment along the physical fiber links that it spans. The design of virtual topology over an existing physical topology for a given traffic demand matrix is a major task. When the traffic demand matrix is changed, the topology may not perform well as it did previously. Obviously, the optimal rearranging of a virtual topology is required to reduce the traffic disruption (also the rearranging operation cost) and increase the performance. The process of rearranging the virtual topology to meet the new traffic requirement is called a reconfiguration process[20]. Unlike the virtual topology design, the reconfiguration process is a trade-off between the performance objective in virtual topology design and the cost objective in terms of the number of changes in the virtual topology. Since the reconfiguration is not only a one-time operation, it could be activated whenever the traffic demand is changed. The consequent problem is how and when to perform the reconfiguration process. There must be a policy to control the reconfiguration process to gain the optimal outcome in a long term. The reconfiguration process and its policy are challenging problems especially in large optical networks e.g., a WDM backbone network.

1.2 Our Work

There are two contexts in the studies of a reconfiguration in optical network: the reconfiguration in broadcast WDM network and the reconfiguration in wavelength-routed optical network. Our work is the design of reconfiguration model for mesh wavelength-routed networks. The model includes two tasks: a reconfiguration process and a reconfiguration policy. A wavelength-routed optical network is typically a transport network or a backbone network, and a WAN rather than a LAN. Thus we approach our model with realistic SONET/SDH traffic demands that change regularly on a daily or weekly basis. The SONET/SDH based traffic is comprised of multiple streams (e.g., OC-3, OC-12 and OC-48), the grooming at the edge is required to route the traffic streams.

We have stated that the reconfiguration problem consists of a reconfiguration process and a reconfiguration policy. The reconfiguration process is the process to redesign a virtual topology under a given physical topology, previous virtual topology and a new traffic demand matrix such that the new virtual topology is not too different from the previous one while it still serves the new demand efficiently. The reconfiguration policy tells us how and when to perform the reconfiguration process to gain the maximum average outcome in a series of reconfigurations. We propose a complete reconfiguration model consisting of a reconfiguration process and a reconfiguration policy.

1.2.1 Reconfiguration Process Approach

Since a reconfiguration is a problem of two competitive objectives, we propose a reconfiguration process that optimizes both objectives by the concept of Pareto optimality. Our reconfiguration process generates a set of non-dominated solutions known as a Pareto front using a Multi-objective Evolutionary Algorithm (MOEA). Then we pick the most preferred solution from this set according to a reconfiguration policy.

1.2.2 Reconfiguration Policy Approach

We propose a Markov Decision Process (MDP) to make a decision for our reconfiguration policy. The policy picks a solution from the Pareto front that returns the highest average outcome for a long series of reconfigurations. We define the states, actions, state transitions, reward/cost function and epoch for the reconfiguration policy model.

CHAPTER 2

BACKGROUND

2.1 Early Reconfiguration Process Approaches

Early research in a reconfiguration process has been studied in two different contexts: a broadcast optical network and a wavelength-routed optical network. They are slightly different in objectives and constraints by nature of the network types. A broadcast network is mostly applied for LAN or MAN network in a passive star topology while a wavelength-routed network is applied for a transport backbone network in a mesh topology. Therefore a reconfiguration in a broadcast network is performed more frequently in terms of packet-by-packet basis and requires faster turning receivers (or transmitters) than those of a wavelength-routed network. There are plenty of heuristic approaches in a reconfiguration problem of broadcast optical networks.

Labourdette et al. [15] considers a reconfiguration as a transition diagram that disrupts the traffic minimally through a sequence of branch exchange operations. The problem of finding the shortest sequence is equivalent to the problem of finding a decomposition of auxiliary graph algorithms. The shortest sequence provides the minimum duration of reconfiguration phase. At each step, two links are disrupted (exchanged) on the ring topology. A similar algorithm called the Dynamic Single-Step Optimization (DSSO) is introduced by Narula-Tam and Modiano [16]. The DSSO has been proposed for load balancing that tracks rapid changes in a traffic pattern using branch exchange sequence. Ernest et al. [7] focuses on the cost-benefit analysis to reduce a reconfiguration cost. Their Merge Split Reconfiguration (MSR) algorithm reduces the number of lightpaths that need to be reconfigured while keeping the network congestion as low as possible. Baldine and Rouskas [3] and Alfouzan and Jayasumana [2] consider a reconfiguration problem as a trade-off between the number of receiver retunings and the degree of load balance. Baldine and Rouskas presented a new algorithm that attempts to construct the new wavelength assignment in a way that simultaneously achieve both objectives. Alfouzan and Jayasumana developed the Most and Least Loaded Channel Balance (MLLCB) algorithm such that the demand on most loaded channel is reduced by exchanging one node with the least loaded channel.

In a wavelength-routed optical network reconfiguration, the existing research attempts to maximize the performance and to minimize the number of changes in a virtual topology. The performance in the wavelength-routed optical network can be measured by various metrics including the average propagation delay of a lightpath, the average hop-distance of traffic, the success traffic throughput, the maximum load offered to any lightpath (congestion) and the utilization of traffic over the lightpaths.

Banerjee and Mukherjee [5] formulated the reconfiguration problem using linear programming. Their performance objective is to minimize the average packet hop distance in the network. In the first step, they search for the optimal value of the performance objective under a new demand and a new virtual topology. In the second step, they minimize the number of changes in the virtual topology using the optimal performance objective in the first step as a constraint. Ramamurthy and Ramakrishnan [18] extend the objective in [5] to minimize the average number of packet hops in the network, minimize the total number of lightpaths, minimize the hops as well as the number of physical links or the sum of these objectives. Since the reconfiguration problem is proved as an NP-hard problem [23], the linear programming approaches do not scale well for a large network.

Sreenath et al. [20] proposed a two-phase heuristic for a reconfiguration problem such that the performance objective is to minimize the average weighted hop count. The heuristic is designed to maintain the near-optimality of the virtual topology, to provide a compromise between the trade-off objectives and to quickly find the lightpaths to be reconfigured. Although the algorithm scales well for a large network, the solution relies on the setup parameter i.e., the bound on the number of changes.

Zheng et al. [24] focus on the virtual private network such that the traffic demand is in the term of wavelengths required and the performance objective is to minimize the average propagation delay of the lightpaths. They proposed a Balanced Alternate Routing Algorithm (BARA) based on a genetic algorithm to solve the problem. They use the weighted combination of trade-off objectives when applies to a (single objective) genetic algorithm.

Takagi et al. [23] focuses on the sequence of reconfiguration process in order to minimize the disruption or maximize the network availability. They propose four heuristic algorithms including Longest Lightpath First (LPF), Shortest Lightpath First (SPF), Minimal Disrupted lightPath First (MDPF) and Tree Search (TS) algorithms. The LPF and SPF result in a low performance. The TS and MDPF provide good performance but both have computational complexity.

The other heuristic approach is introduced by Gencata and Mukherjee [10]. They consider an adaptation mechanism for reconfiguration. The proposed algorithm redesigns the virtual topology according to an expected traffic pattern. It detects the imbalances of the network by high and low watermark parameters on lightpath loads and reacting promptly (by adding or deleting lightpath one at a time) to balance the loads.

2.2 Early Reconfiguration Policy Approaches

The reconfiguration process has an overhead cost that interrupts traffic and rearranges the lightpaths. Therefore frequent reconfiguration is costly. However, infrequent reconfiguration will downgrade the performance. The policy guides us when is the best time to perform a reconfiguration process and what level of reconfiguration to be performed. There are few studies in the reconfiguration policy. Some of them are described below.

Geary et al. [9] detects the best time to perform reconfiguration when the overall average network utilization goes beyond the specified threshold or when a link runs out of capacity. This method is known as a threshold approach which is difficult to define the threshold value, and may not result in the optimal outcome. Baldine and Rouskas[4] propose the reconfiguration policy on the broadcast optical network using the Markov Decision Process (MDP) to obtain optimal outcome. They define the reward and cost functions to calculate the optimal outcome. This is an approximate model, the outcome depends on how close of the state transition probabilities and the reward/cost functions are to the real network. Usually the model parameters are obtained from the simulation or real network. However, they have shown that the MDP model outperforms the threshold approach for a long term.

CHAPTER 3

RECONFIGURATION PROBLEM

3.1 Problem Definition of The Reconfiguration Process

We assume that our reconfiguration problem is a centralized optimization problem which acquires a global view or status of the network. The reconfiguration is activated or triggered by the changes in a traffic demand, neither the failure of equipment nor the changes in a physical topology.

The reconfiguration process is a multi-objective problem that consider not only the network performance but also the number of changes in the virtual topology. The problem formulation is different from the ordinary virtual topology design in that it requires another objective (to minimize the changes in virtual topology) besides the performance objective. Therefore, it requires both the new traffic demand and the previous logical topology as inputs. We formulate the reconfiguration process problem as a Linear Programming (LP) using the principle of multicommodity flow of the set of lightpaths mapped to the physical layer and the set of traffic on the designed virtual topology. In order to deal with two objectives in the LP, we have to set one objective as a constraint while LP optimizes the other and then takes a turn. Unlike the LP, our approach optimizes both objectives concurrently by the concept of the Pareto Optimal. Our approach creates a set of non-dominated solutions in a single run unlike the LP that provides one solution at a time.

We begin the formulation in the virtual topology design (including the grooming) and then we show the dealing of two objectives in the LP for the reconfiguration problem.

3.1.1 Virtual Topology Design Problem Formulation

Given a physical topology and traffic demand matrices, we classify a virtual topology design problem into three subproblems: a lightpath routing subproblem, a traffic routing subproblem and a wavelength assignment subproblem. The lightpath routing subproblem is the design of lightpaths routed on the physical topology. The traffic routing subproblem is a routing of low-speed traffic on a virtual topology under some limitations (e.g.,the available number of transceivers). This subproblem is known as a traffic grooming subproblem since multiple low traffic streams are groomed into a single lightpath. The wavelength assignment subproblem is an assignment of limited number of wavelengths per fiber under a Distinct Color Assignment (DCA) constraint and a wavelength continuity constraint. The DCA constraint states that lightpaths on the same fiber link must be assigned with the distinct colors. The wavelength continuity constraint states that a lightpath must occupy the same wavelength along the links that it spans. The wavelength continuity constraint could be relaxed by deploying the wavelength conversion that allows a lightpath to switch to any wavelengths at the links that it spans.

We assume that all nodes are capable of grooming low-speed traffic to the available capacity of a lightpath for as many traffics as needed and a transceiver is freely tuned to any wavelengths. We do not allow the de-multiplexing of OC-x lower than its capacity when routing through the network. However two or more OC-x streams of the same source and destination may pick a different route. (The detail of traffic grooming over WDM network is presented in [17].)

Although our problem formulation considers the SONET streams, it could be applied to any type of traffic streams or a fraction of lightpath capacity.

Given Parameters:

- N: Number of Optical nodes.
- W: Number of Wavelengths that can be multiplexed on a single fiber i.e., the DWDM capacity.
- T_i: Number of transmitters at node i; $T_i \geq 1 \, \forall i$
- R_i: Number of receivers at node i; $R_i \geq 1 \, \forall i$
- K: Number of shortest paths or alternative routes.
- P: Physical topology matrix.

$$P = [P_{mn}; m, n = 1, 2, \ldots, N]_{N \times N},$$

  where P_{mn} is the number of fibers between node m and node n. Note that: $P_{mn} = P_{nm}$.

- Λ: $N \times N$ traffic demand matrix

$$\Lambda^x = [\Lambda^x_{sd}; s, d = 1, 2, \ldots, N]_{N \times N},$$

  where Λ^x_{sd} is the demand of low speed streams, OC-x, between node s and node d; $x \in \{1, 3, 12\}$.

- L_{max} The load of maximally-loaded lightpath in the network.

Variables:

- $\sigma^k_{ij,mn,w}$: An indicator representing the existence of lightpath where

$$\sigma^k_{ij,mn,w} = \begin{cases} 1, & \text{if there exists a lightpath from } i \text{ to } j \\ & \text{being routed through fiber link } mn \text{ on} \\ & \text{the } k^{th} \text{ path in } K \text{ and using wavelength} \\ & w \text{ in } W. \\ 0, & \text{otherwise.} \end{cases}$$

- V_{ij}: The number of lightpaths from node i to node j in the virtual topology.

- $\lambda^x_{sd,ij}$: The number of OC-x streams from node s to node d being routed on the lightpath ij.

- C: The capacity of a lightpath e.g., $C = 48$ for OC-48.

- S^x_{sd}: The number of OC-x streams requested from node s to node d that are successfully routed. The traffic is blocked if $S^x_{sd} < \Lambda^x_{sd}$.

Constraints:

- Traffic (Multicommodity-flow equations for lightpath routing):

$$\sum_m \sum_k \sum_w \sigma^k_{ij,mv,w} = \sum_n \sum_k \sum_w \sigma^k_{ij,vn,w} \text{if } v \neq i, j \forall ij \tag{3.1}$$

$$\sum_n \sum_k \sum_w \sigma^k_{ij,in,w} = V_{ij} \quad \forall ij \tag{3.2}$$

$$\sum_{m}\sum_{k}\sum_{w}\sigma^{k}_{ij,mj,w} = V_{ij} \quad \forall ij \tag{3.3}$$

Equation (3.1) allows a lightpath to have any wavelength in each link along the path. It implies that wavelength converters are available at all nodes. Otherwise, if we need to reserve the wavelength continuity rule, Equations (3.1), (3.2) and (3.3) will become Equations (3.4), (3.5) and (3.6) respectively.

$$\sum_{m}\sum_{k}\sigma^{k}_{ij,mv,w} = \sum_{n}\sum_{k}\sigma^{k}_{ij,vn,w} \quad \text{if } v \neq i,j \quad \forall ij,w \tag{3.4}$$

$$\sum_{n}\sum_{k}\sigma^{k}_{ij,in,w} = V_{ij} \quad \forall ij,w \tag{3.5}$$

$$\sum_{m}\sum_{k}\sigma^{k}_{ij,mj,w} = V_{ij} \quad \forall ij,w \tag{3.6}$$

- Wavelength Constraints:

$$\sum_{ij}\sum_{k}\sigma^{k}_{ij,mn,w} \leq P_{mn} \quad \forall mn,w \tag{3.7}$$

Equation (3.7) ensures that distinct channels (lightpaths) on the same fiber link cannot be assigned the same wavelength. Note that lightpaths ij using different fibers between link mn are known to be on different paths in K.

- Resource Constraints:

$$\sum_{j} V_{ij} \leq T_i \quad \forall i \tag{3.8}$$

$$\sum_{i} V_{ij} \leq R_j \quad \forall j \tag{3.9}$$

Equation (3.8) ensures that the number of lightpaths originate from node s is not greater than the number of transmitters at that node. Likewise Equation (3.9) ensures that the number of lightpaths terminated at node d is not greater than the number of receivers at that node.

- Traffic (Multicommodity-flow equations for traffic routing):

$$\sum_i \lambda^x_{sd,iv} = \sum_j \lambda^x_{sd,vj} \quad \text{if } v \neq s, d \quad \forall sd, x \tag{3.10}$$

$$\sum_j \lambda^x_{sd,sj} = S^x_{sd} \quad \forall sd, x \tag{3.11}$$

$$\sum_i \lambda^x_{sd,id} = S^x_{sd} \quad \forall sd, x \tag{3.12}$$

$$S^x_{sd} \leq \Lambda^x_{sd} \quad \forall sd, x \tag{3.13}$$

- Capacity Constraint

$$\sum_x \sum_{sd} (x \times \lambda^x_{sd,ij}) \leq V_{ij} \times C \quad \forall ij \tag{3.14}$$

Objectives: We list the possible performance objectives in the Optical network design area as shown below:

- Minimize the Average Propagation Delay of the lightpath (APD):
  The APD relies on the length of media which represent the cost and performance of the network. In the high speed networks, the propagation delay is a dominant delay while the queuing delay is neglected. The lengthy fiber causes not only a considerable delay but also the impairments of noise accumulation, fiber chromatic dispersion, polarization mode dispersion and fiber nonlineari-

ties. Moreover, minimizing the APD could reduce the "cleaning up" cost at intermediate nodes that work as 3R optical regenerators (Re-amplification, Re-timing and Re-shaping).

$$\text{Min}\frac{1}{\sum_{i,j} V_{ij}} \sum_{i,j} \sum_{m,n} (d_{mn} \sum_{k} \sum_{w} \sigma^{k}_{ij,mn,w}) \tag{3.15}$$

Where d_{mn} is the propagation delay weight factor on the fiber link from node m to n. The value of d_{mn} depends on the length of the fiber.

Note that if $d_{mn} = 1 \; \forall \; m, n$ the APD becomes the average hop distance (or hop count) of the lightpaths.

- Maximize the traffic throughput:

$$\text{Max} \sum_{sd,x} (x \times S^{x}_{sd}) \tag{3.16}$$

- Minimize the Maximally-loaded lightpath:

  The L_{max} represents the maximal congestion on a link. The minimizing of L_{max} will distribute the load to the entire links.

$$\text{Min } L_{max} \tag{3.17}$$

$$\text{Where } L_{max} = \text{Max } \sum_{sd,x} (x \times \lambda^{x}_{sd,ij}) \quad ; \forall \; ij$$

- Minimize the Average Hop-distance of Traffic (AHT):

$$\text{Min}\frac{1}{\sum_{sd,x}\Lambda_{sd}^{x}}\sum_{ij}\sum_{sd,x}(x\times\lambda_{sd,ij}^{x}) \tag{3.18}$$

Note that the AHT is defined for the non-blocking traffic network since this is SONET traffic that cannot be discarded. We consider the AHT because it reflects the cost of grooming traffic. Low SONET streams are groomed at the edge in the electrical domain before they are converted to a light form and carried over a lightpath. The higher value in the AHT, the more cost and delay in the network according to O-E-O conversion at the intermediate nodes. Hence, the AHT must be minimized. The lower bound of AHT is one hop (i.e., no intermediate grooming).

- Maximize the Virtual Topology Utilization:

  The virtual topology utilization is the use of lightpaths.

$$\text{Max}\frac{1}{\sum_{ij}V_{ij}}\sum_{ij}\sum_{sd,x}(x\times\lambda_{sd,ij}^{x}) \tag{3.19}$$

- Minimize the Network Resources:

  Network resources include the number of lightpaths or the number of transmitters and receivers or the number of wavelengths. Since the optical switch specification is usually fixed, these resources are set as constraints.

3.1.2 Dealing with Two Objectives in Reconfiguration Process

The Reconfiguration problem formulation is similar to the Optical Network design formulation but in the reconfiguration part, we do not only maximize network performance but also minimize the number of changes made in the reconfiguration

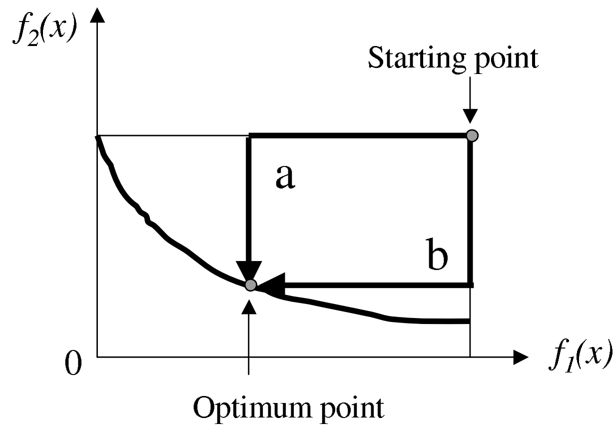


Figure 1: The Two Steps of Optimization.

process. Since changes in virtual topology are costly in terms of traffic disruption and overhead to make the changes e.g., re-tuning of wavelength. However, these objectives are in conflict e.g., the lower AHT, the more changes required and vice versa. To deal with two objectives, we must fix one objective while minimize another one and then take turns to yield the best solution. Next we show an example of the reconfiguration formulation with two objectives: minimize AHT and minimize numbers of changes in the virtual topology. The process requires two steps: fix one objective and then minimize another objective. The process can be run either way (**a** or **b** in Figure 1) or run iteratively until the optimal solution is met. This process generates only one non-dominated solution on the Pareto front for each run. Moreover, the solution depends on the initial point and we cannot predict the result position on the Pareto front. Therefore it is difficult to form the Pareto-front. Note that we describe the Pareto front and the non-dominated solutions in Chapter 4.

3.1.2.1 Fix AHT, Minimize Number of Changes

First we find AHT' of the new traffic demand regardless of the number of changes in the virtual topology. Next we replace the objective function in ILP with the Minimize of the number of changes in lightpaths.

$$\text{Min} \sum_{i,j} \sum_{m,n} \sum_{k} \sum_{w} |\sigma'^{k}_{ij,mn,w} - \sigma^{k}_{ij,mn,w}| \tag{3.20}$$

where $\sigma'^{k}_{ij,mn,w}$ is an indicator presenting the existing of lightpath from i to j being routed through fiber link mn on the k^{th} path using wavelength w of the previous solutions. The equation (3.20) is linear since the $\sigma'^{k}_{ij,mn,w}$ are binary. We can rewrite the absolute term in (3.20) as following.

$$|\sigma'^{k}_{ij,mn,w} - \sigma^{k}_{ij,mn,w}| = \begin{cases} 1 - \sigma^{k}_{ij,mn,w}, & \text{if } \sigma'^{k}_{ij,mn,w} = 1 \\ \sigma^{k}_{ij,mn,w}, & \text{if } \sigma'^{k}_{ij,mn,w} = 0 \end{cases}$$

Next we add new constraint to the ILP.

$$\frac{1}{\sum_{sd,x} \Lambda^{x}_{sd}} \sum_{ij} \sum_{sd,x} (x \times \lambda^{x}_{sd,ij}) \leq AHT' \tag{3.21}$$

3.1.2.2 Fix Number of Change, Minimize AHT

Given NoC be the number of changes in lightpath. First we replace the objective function in ILP with the Minimize of AHT.

$$\text{Min} \frac{1}{\sum_{sd,x} (x \times \Lambda^{x}_{sd})} \sum_{ij} \sum_{sd,x} (x \times \lambda^{x}_{sd,ij}) \tag{3.22}$$

Next we add new constraint to the ILP.

$$\sum_{i,j} \sum_{m,n} \sum_{k} \sum_{w} |\sigma'^{k}_{ij,mn,w} - \sigma^{k}_{ij,mn,w}| \leq NoC \tag{3.23}$$

3.2 Problem Definition in Reconfiguration Policy

We formulate the reconfiguration policy as a set of decisions in a Markov Decision Process model or a Dynamic Programming model. The MDP model consists of five elements:

1. A set of decision epochs which is a period of time that triggers the action.
2. A set of states which indicates the status of the network e.g., a performance parameter and a current traffic demand.
3. A set of actions.
4. A set of state and actions dependent on immediate rewards and costs. The reward is the benefit gained from doing the particular action while the cost is incurred from that action.
5. A set of state transition probabilities which relies on the action and the arrival traffic that changes the state.

This model is a discrete time model such that actions, rewards, costs and the state transition probabilities depend only on the current state (Markov property). Let $R_i(H)$ be the reward function of H, the performance variable in round i^{th} and $C_i(\eta)$ be the cost function of η, the number of changes in lightpaths in round i^{th} of reconfiguration. For each state transition with a performed action, we want to maximize the expected outcome O in every reconfiguration rounds where

$$O = \lim_{y \to \infty} \frac{1}{y} \, E\{\sum_{i=1}^{y} (R_i(H) - C_i(\eta))\} \tag{3.24}$$

The policy or set of decisions tell us what action we should select in each state to maximize the expected outcome O.

3.3 Hypothesis

If the reconfiguration problem is a multi-objective problem in which the objectives are conflicting to each other, there exists a Pareto front corresponding to the objectives and if we define the status of network as a state, a set of state transitions, a set of reward/cost functions, and a set of actions in a Markov Decision Process, we can find the optimal policy by assigning the action in each state. In the long term, the expected outcome of the MDP policy is higher than that of the Immediate Highest Outcome (IHO) policy which takes the action that generates the highest outcome at each state transition.

Next Chapter we describe the detail of Pareto front in the multi-objective optimization and explain why we need the Multi-Objective Evolutionary Algorithm in the reconfiguration problem.

CHAPTER 4
RECONFIGURATION PROCESS APPROACH

4.1 Multi-Objective Optimization

A Multi-objective Optimization (MO) or multicriteria/multiperformance/vector optimization problem solves several competing objectives simultaneously. In the MO, there is a set of optimal solutions that non-dominate each other within the set but dominate other solutions outside of the set when considering all objectives. The set of optimal solutions is known as a Pareto Optimal set. Most MO algorithms use the concept of domination to search for the Pareto Optimal set. The definition of domination is defined in Definition 4.1.1.

Definition 4.1.1 *(Pareto Optimal Definition) Given "$\lhd$" be the operator such that $f_k(\mathbf{x}) \lhd f_k(\mathbf{y})$ if a solution $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ is a better solution than a solution $\mathbf{y} = (y_1, y2, \ldots, y_m)$ for the k^{th} objective and m parameters (decision variables). The "better" means "less than" in case of minimization or means "greater than" in case of "maximization".*

If there are n objectives, a solution $\mathbf{x}$ is said to dominate a solution $\mathbf{y}$ if

$$\forall i \in \{1, 2, \ldots, n\} : \; f_i(\mathbf{x}) \ntriangleright f_i(\mathbf{y}) \quad \wedge$$

$$\exists i \in \{1, 2, \ldots, n\} : \; f_i(\mathbf{x}) \lhd f_i(\mathbf{y})$$

In words, a solution $\mathbf{x}$ is said to dominate a solution $\mathbf{y}$ (or $\mathbf{x}$ is non-dominated by $\mathbf{y}$) if 1 and 2 are true: 1) The solution $\mathbf{x}$ is no worse than $\mathbf{y}$ for all objectives and 2) The solution $\mathbf{x}$ is strictly better than $\mathbf{y}$ in at least one objective.

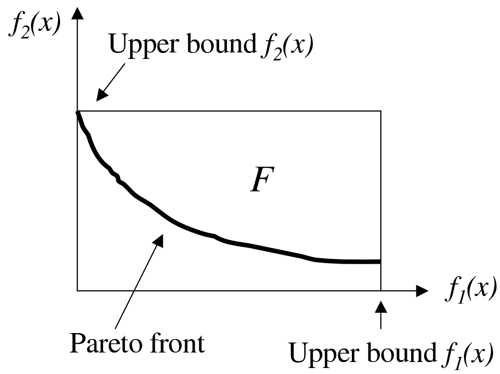


Figure 2: The Pareto Front and Feasible Solution Area, F, of Two Objective Functions.

The plot of non-dominated solutions over the objective axes will form the Pareto front. For instance, the Pareto front for the "minimization" on both objective functions, $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ is shown in Figure 2 where the F area denotes the feasible solutions area. Note that the Pareto front can be either convex or non-convex.

Usually there are more than one solution in the Pareto optimal, called non-dominated solutions or non-inferior solutions and there are many methods to search this set of solutions or Pareto optimal set. One of the traditional approaches is the aggregation methods. It combines the objectives into a scalar function and applies the single objective optimization methods like a simulated annealing, a stochastic local search or a tabu search on it. Examples of combining objectives can be found in weighting method [13], constraint method [13], goal programming [22] and min-max method [14]. In the weighting method, one may define the utility function that combines multiple objective functions together as shown in equation (4.1).

$$U = \sum_{i=1}^{n} W_i \frac{f_i(\overline{x})}{f_i^*} \tag{4.1}$$

where W_i is a weighting factor for each objective function and f_i^* is the scaling parameters for the i^{th} objective function.

Another form of utility function which eliminates the scaling problem with upper bound and lower bound is shown in equation (4.2).

$$U = \sum_{i=1}^{n} W_i \frac{f_i(\overline{x}) - f_i^0}{f_i^{max} - f_i^0} \tag{4.2}$$

where f_i^0 and f_i^{max} are the lower bound and upper bound for the i^{th} objective function respectively.

Although the aggregation methods are simple and applicable to the single objective methods, they cannot generate all members of the Pareto optimum set with non-convex Pareto front. Also the weighting factor is quite subjective rather than straightforward and difficult to define.

Other methods for multi-objective optimization are the Evolutionary Algorithms called Multi-Objective Evolutionary Algorithm (MOEA) that simulates the process of natural evolution using a class of stochastic optimization methods. These methods are able to capture a Pareto-optimal set in a single run. Moreover, they are less susceptible to the shape or continuity of the Pareto fronts (i.e., it can search on a problem with non-convex Pareto front.) In the reconfiguration problem, the sequence of changes effects the disruption and network availability as presented in [23]. In our case, we consider the number of changes, not the sequence. The MOEA will search the possible sequence of changes in the virtual topology that generate the best performance. Next we show that a different sequence of changes effect the performance.

Theorem 4.1.1 *A sequence of changes in virtual topology affects network performance.*

Proof We prove the theorem based on the AHT. Given the network with the physical topology and the previous virtual topology as shown in Figure 3. Suppose LP_1 and LP_2 are the lightpaths from source to destination using different wavelengths as shown in the previous virtual topology in Figure 3. The lightpath LP_1 and LP_2 have one unit of traffic with m hops and n hops between source and destination respectively where $m > n$. In the final virtual topology, it is required to move both lightpaths to the middle physical path which is one hop away between source and destination and is available for both lightpaths.

If the LP_1 is moved first (to the middle path), the $NoC = 1$ and $AHT = \frac{(n+1)}{\Lambda}$ where NoC denotes the numbers of changes and Λ denotes the total traffic. Next the LP_2 is moved, the $NoC = 2$ and $AHT = \frac{(1+1)}{\Lambda}$.

If the LP_2 is moved first (to the middle path), the $NoC' = 1$ and $AHT' = \frac{(m+1)}{\Lambda}$. Next the LP_2 is moved, the $NoC' = 2$ and $AHT' = \frac{(1+1)}{\Lambda}$.

Although both sequences end up with the same AHT, but at the first move, the former sequence has better performance that the latter i.e.,$AHT' < AHT$ ∎

Theorem 4.1.1 shows that the sequence of changes effects the performance. Therefore, the changes may increase or decrease the performance. We define the sequence of changes into two types.

Definition 4.1.2 *A sequence of necessary changes is a sequence of changes in virtual topology that improves the performance. Otherwise it is a sequence of unnecessary changes.*

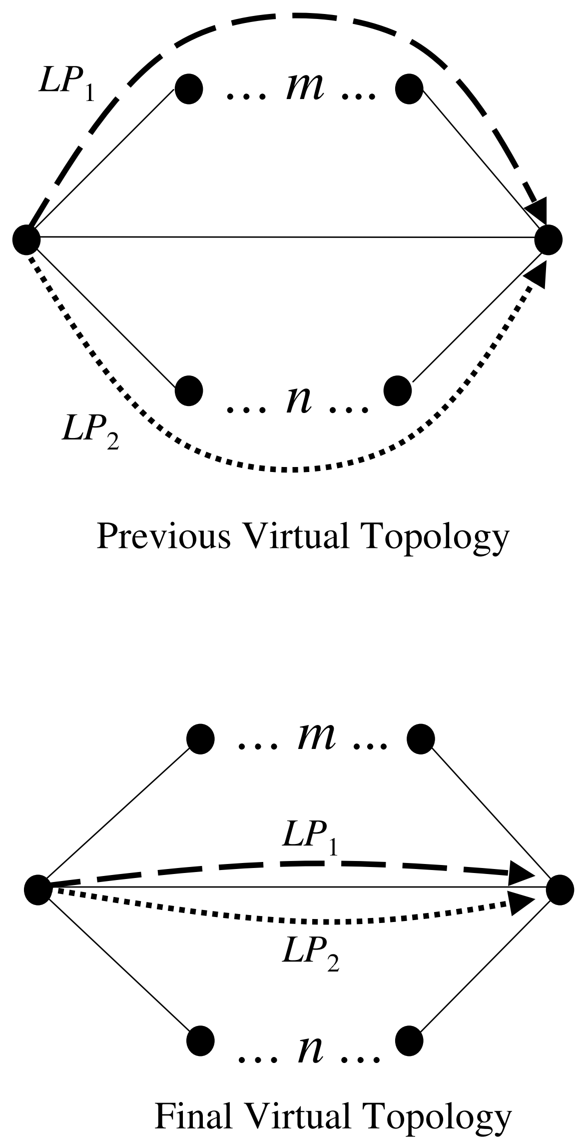


Figure 3: The Previous and Final Virtual Topology.

Theorem 4.1.2 *If the AHT of network is not yet optimized, there exists a Pareto front between "min AHT" objective and "max number of necessary changes" objective.*

Proof Given a non-optimized optical network after the traffic has been changed. There are three possible types of changes in a lightpath corresponding to the difference between two traffic demand matrices (previous vs new traffic demand matrices). These three types are the addition of lightpath(s), the deletion of lightpath(s) and the re-routing/coloring of lightpath(s). Let a sequence of necessary changes contains i additions, j deletions and k re-routings/colorings. Therefore, $NoC = i + j + k$. According to the constraints, the addition of lightpath requires the availability of transceiver and color. Thus, i depends on j and k. At particular $j + k$ changes, the NoC depends on i. Since every addition of lightpath will create one hop away for a new traffic demand and the AHT is decreased, the plot of the AHT and the NoC forms the Pareto front. $\blacksquare$

The reconfiguration problem happens when the traffic demand is changed and causes the performance to be non-optimized. Theorem 4.1.2 shows that there exists Pareto front for the reconfiguration problem. Note that we simulate the network and traffic demand to show the Pareto front in our experiment. After we created the Pareto front, the policy will play its role to pick the best solution in the Pareto front. The MOEA that creates the Pareto front takes advantage over an aggregation or an LP method in term of "choices" of the policy has.

4.2 MOEA Overview

Different stochastic search techniques were introduced to solve many real-world scientific and engineering multi-objective problems. In Germany, the technique

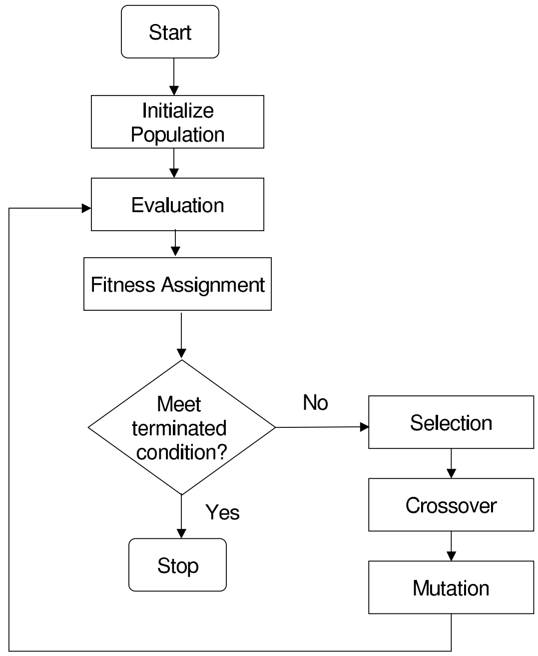


Figure 4: The MOEA-Operation Diagram.

called evolution strategies (ES) was proposed, while in the United State the genetic algorithms (GA) and the concept of evolutionary programming (EP) were introduced. These techniques which transfer evolutionary principles or the Darwinian concept of "Survival of the Fittest" into the search space of programming languages, are summarized today under the names evolutionary algorithms (EA) or evolutionary computation (EC). (The term EA is an algorithm used in the EC). The MOEA is the EA that deals with multiple objectives based on the Pareto Optimal definition. Most MOEAs are derived from the single-objective evolutionary algorithms like Genetic Algorithm. Therefore the operations of MOEA consist of a population initialization, an evaluation, a fitness assignment, a reproduction (selection), a crossover and a mutation. Figure 4 shows the MOEA-operation diagram.

The population initialization generates a set of chromosomes or individuals randomly. The chromosome is an encoded solution to the problem which is usually presented in binary or string format. Each chromosome consists of genes which take on certain values (alleles). A size of population depends on the user. If the size is too big, it will waste the time to evaluate the chromosomes. If it is too small, the optimal solution may not be found. An evaluation operation measures how well (good fitness) the chromosome to be survived in the next generation. An evaluation function or a fitness function is based on objective functions of the problem. In the MOEA, the fitness function is a combination of objective functions based on the strategies corresponding to each particular MOEA technique. A reproduction or a selection operation allows the good solutions with a high chance to be duplicated and the bad solutions to be eliminated while maintaining the same population size for the next generation. The common selection schemes are a tournament selection, a proportionate selection and a ranking selection. The tournament operation copies solutions into two sets and then matches up each pair randomly. The winner (better fitness of the pair) is placed in the mating pool so the size of population is the same while a good fitness solution has a chance to win both tournaments and has two copies in the new population. In the proportionate selection, solutions are assigned the copies proportional to their fitness values. The proportionate selection causes a scaling problem or genetic drift problem such that a population tends to converge to a single "super" solution. This problem can be avoided by the ranking selection operation. In this operation, the chromosomes are sorted by their fitness from the worst (rank 1) to the best (rank N, where N $\leq$ Population size). Each member has its rank used in place of a fitness value. The proportionate selection is then applied with this rank value. The next operation is the most important operation called the crossover operator. Like in biological systems, the crossover process yields recombination of alleles by exchanging segments between pairs of chromosomes. Two chromosomes are picked randomly to exchange the segments. If the segment is assigned in a single position of the chromosome, it is called single-point crossover. Similarly, the n-point crossover, choose n random crossover points. Another scheme called a uniform crossover exchanges bit-by-bit of chromosomes rather than segments. The mutation operation flips the bit in the chromosome to keep diversity in the population. The term crossover rate and the mutation rate are a probability to perform a crossover operation and a mutation operation respectively. For example, a typical crossover rate is 0.6, and a mutation rate is 0.001 with the population size of 100. However, there is no specific rule to define these rates.

There are several approaches in the MOEA including the population-based non-Pareto approach like Vector Evaluated Genetic Algorithm (VEGA) [19], and the Pareto-based approaches like the Multi-Objective Genetic Algorithm (MOGA) [8] , the Non-dominated Sorting Genetic Algorithm (NSGA) [21], Strength Pareto Evolutionary Algorithm (SPEA) [25] and Niched Pareto Genetic Algorithm (NPGA) [11]. They intend to find widely spread non-dominated solutions using their unique fitness assignment schemes for multiple objectives.

The VEGA is simply modified from EA by randomly dividing the mating pool into an equal size of n parts (n objectives) to deal with multiple objectives. Each part is assigned a fitness based on a different objective functions, i.e., part one population are assigned by fitness of the first objective only, part two are assigned by a fitness of the second objective and so on. The mutation and crossover are performed as usual. Therefore VEGA is easy to implement and good for problems that satisfy the solutions near the individual best solution of each objective. The solutions are not necessarily globally non-dominated.

The MOGA has its own fitness assignment scheme to deal with multiple objectives. The rank of a certain chromosome corresponds to the number of individuals in the current population by which it is dominated. The chromosomes in the same rank are assigned by the average fitness among themselves. In order to maintain diversity among non-dominated solutions, the MOGA uses niche count with sharing function to distribute the population over non-dominated solution (the less-crowded region will have a better scaled fitness). The niche count and sharing function technique have difficulty in assigning the parameters and it may happen that a solution of lower rank has a better scaled fitness (if there exist many crowded solutions with a better rank).

In NSGA, the population is ranked using Pareto ranking i.e., non-dominated chromosomes are classified into one category with a dummy fitness value that is proportional to the population size. The fitness assignment process of NSGA allows non-dominated front being emphasized systematically toward the Pareto-optimal region front-wise. The distance used in the sharing function is calculated with decision variables (phenotype, not the genotype but can do either) so that it allows phenotypically diverse solutions. However, the sharing function requires fixing parameters (there exist dynamic parameter approach) which affect the performance of the algorithm. Besides the sharing function itself is complex by the size of members in each front.

The NPGA uses a tournament selection scheme (different from single-objective optimization) based on Pareto dominance. A higher number of chromosomes is involved in the competition (it defines the subpopulation with t_{dom} size for comparison). If the competitors tie, the result of the tournament is decided through fitness sharing in the objective domain. There is no need for specifying any particular fitness value to each solution. The selection operator prefers non-dominated solutions in a stochastic manner. Since the domain-check is performed only within subpopulations of solutions, the complexity does not depend on the number of objectives, but the t_{dom} (the subpopulation size). Therefore, NPGA is efficient for a problem with high number of objectives. However, it requires two fixed parameters, a sharing function parameter and t_{dom} which affect the performance.

The SPEA is a Pareto-based approach like MOGA, NSGA and NPGA which implement the Pareto-based fitness assignment strategy to determine the reproduction probability of each individual. It maintains the set of non-dominated solutions in the separated population. Hence, there are two populations, the dominated population P_t of size N and the external non-dominated population $\bar{P}_t$ with the limited size of $\bar{N}$, where t denotes the t^{th} generation. The SPEA maintains the external population of the Pareto optimal solutions to reserve the elites in every generation. It assigns a scalar fitness called strength to the external population and assigns fitness to (internal) population based on the domination and strength values. The size of non-dominant solutions in Pareto front,$\bar{N}$, is controlled by a clustering algorithm such that less crowded elites are kept for the next generation. The selection or reproduction is a binary tournament selection procedure with better fitness values. A crossover operator and a mutation operator are applied to the mating pool as usual. Unlike former Pareto-based approaches that control the distribution of non-dominated solution by a sharing function, the SPEA clustering ensures that a better spread is achieved among the obtained non-dominated solutions. This clustering requires no external parameter excepting $\bar{N}$, the size of external population. If $\bar{N}$ is too small, the effect of elitism will be lost.

4.3 MOEA Approach

In this dissertation we study the reconfiguration process of a large mesh wavelength-routed network. We have illustrated that the reconfiguration process using the linear programming is not applicable. Hence, the Multi-Objective Evolutionary Algorithm dealing with the Multi-Objectives optimization which can generate all candidate solutions in a single run is the right algorithm for the reconfiguration problem.

We design the MOEA for the reconfiguration process using the SPEA which outperforms other MOEAs as stated in [25]. We summarize the SPEA fitness assignment sub-operations in Figure 5. The first block is to find non-dominated solutions within P_t. Then the non-dominated solutions are copied into the external non-dominated population $\bar{P}_t$. Thereafter, some of the copied solutions may dominate the existing solutions in $\bar{P}_t$. The dominated solutions found in $\bar{P}_t$ must be deleted. This is to ensure that non-dominated solutions are kept in $\bar{P}_t$ and carried through the next generation (elitist property). In the next step, it maintains the size of $\bar{P}_t$ i.e., the number of solutions in $\bar{P}_t$ must be less than or equal to $\bar{N}$. Otherwise, the clustering algorithm is performed to reduce the size of $\bar{P}_t$ to $\bar{N}$. The clustering algorithm is based on the Euclidean distance. At the beginning, each solution itself is a cluster. Thereafter, two clusters with the minimum cluster-distance are merged into a bigger cluster. The merging is repeated until the number of clusters is reduced to $\bar{N}$. Next the number of solutions in each cluster must be reduced to one. The

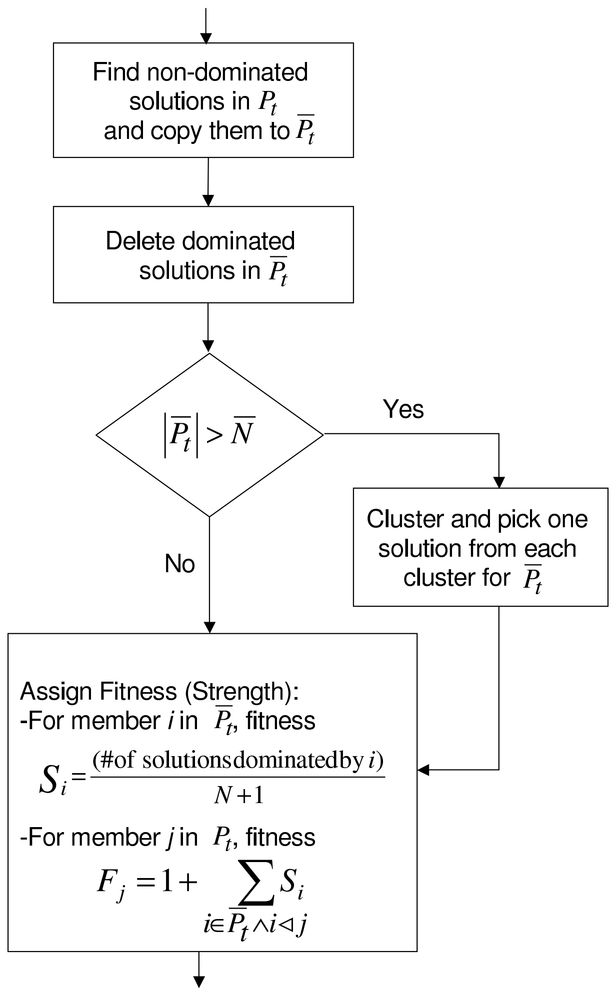


Figure 5: The SPEA Fitness Assignment Diagram.

algorithm keeps the solution which has the minimum average distance from other solutions in the cluster and deletes the others in that cluster. After the size of $\bar{P}_t$ is reduced, the fitness called Strength is assigned to each solution in $\bar{P}_t$ and P_t by the equations shown in the final block of Figure 5. Note that we consider a two-objective minimization problem, so a smaller fitness value represents a better solution. More detail of the SPEA algorithm can be found in [25] and [6].

The reconfiguration objectives are incorporated in the SPEA fitness assignment process to generate the Pareto front. For the reconfiguration process, the solution is a virtual topology with the objectives that minimizes the AHT in Equation (3.18) and minimizes the number of changes of lightpaths in Equation (3.20). We optimize these objective by the concept of Pareto Optimal i.e., our solutions are the non-dominated solutions. A solution x is said to dominate a solution y if 1 and 2 are true:

1. The solution x has equal or less AHT than that of y and has equal or lower number of changes in lightpaths than that of y.

2. There exists one objective that the solution x is better (not just equal) than that of y. Where the term "better" means less AHT or lower number of changes in lightpaths.

4.3.1 Chromosome Encoding

The reconfiguration process is based on the design of virtual topology. Therefore the chromosome or individual is the encoded virtual topology which is a solution. We encode the solution by the string of $N \times (N-1)$ elements, where N is the total number of nodes in the optical network. The first element presents the lightpaths

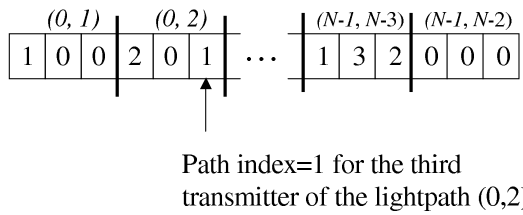


Figure 6: An Example of Chromosome Encoding.

from node 0 to node 1, the second element for node 0 to node 2, the third element for node 0 to node 3 and so forth. Each element contains T_i unit of path indexes from node i to node j where $i \neq j$ and T_i is the number of transmitters at node i. Each path index presents the physical route of a lightpath. If the path index $k^{th} = 0$, there is no lightpath on that transmitter. Otherwise, the lightpath is using the k^{th} path. The path index between node i and j is pre-calculated based on the K-shortest paths over a physical topology or by the random alternative paths. The set of shortest paths or alternative paths is calculated in advance regarding to the given physical topology. Figure 6 shows the example of chromosome encoding where number of transmitters equal 3. There is one lightpath from node 0 to 1 using the first transmitter with path index=1.

4.3.2 Traffic Routing and Wavelength Assignment

After randomly generating the virtual topology solutions in a population, it is possible that the number of required lightpaths is greater than the number of available transmitters. We have to delete some lightpaths to satisfy this constraint. We take the heuristic process by eliminating a lightpath which occupy the lowest traffic first, and then repeat the process until the constraint is satisfied. The traffic in this process is the sum of SONET OC-streams in traffic demand matrices required between source and destination nodes of the lightpath.

Next steps are the Traffic Routing and the Wavelength Assignment. The chromosome is not encoded with a traffic route and an assigned wavelength. Otherwise, a size of the chromosome is considerably large which makes a huge search space. We route the traffic and assign wavelength based on our heuristics. The traffic is routed over a virtual topology using a shortest path algorithm. The routing starts from the highest streams first e.g., route OC-12 demands first, followed by OC-3 demands and OC-1 demands. Bifurcate routing is allowed only in the same OC stream level (i.e., an OC-12 stream cannot be broken into four OC-3s and routed separately but two OC-12 streams of the same source and destination may use the different routes.) We route traffic streams as many as possible over a single hop of lightpath first. The remaining traffic after that is routed over multiple hops of lightpaths. If all SONET streams are routed over a single-hop, the AHT will equal to 1 which is the lower bound.

We number the entire wavelengths (colors) and keep them in a stack one for each fiber link. If the wavelength continuity is considered, we assign the lowest available number (comparing to every stack that is on the lightpath span) to the lightpath that has the maximum hop-count (physical hop) first and so on. Otherwise, without wavelength continuity constraint, we assign the lowest color number found in each link's stack to the lightpath. We set a penalty function to the chromosome if any of the lightpaths in the chromosome cannot assign a color or the traffic is blocked. The penalty function will downgrade the fitness value of chromosome and cause it to be eliminated in the next generation. The flowchart of fitness assignment is presented in Appendix A. The result of the SPEA is the set of non-dominated solutions (non-blocking virtual topology) or the Pareto front that optimizes the objectives and restricts to the constraints. Next step is a reconfiguration policy that picks one of the solutions in the Pareto front.

4.3.3 Changes in Virtual Topology

There are four kinds of changes in lightpaths according to the problem formulation:

1. Changing route,
2. Changing wavelength along the path,
3. Adding new lightpath and
4. Deleting lightpath.

If the "delete" operation is counted as a change, the reconfiguration process will try not to delete the lightpaths since it must minimize the number of changes. Therefore, the reconfiguration process will let the virtual topology keep expanding in each round if it still complies with the constraints. This results in a low utilization of the lightpaths. We get rid of this problem by not counting the "delete" operation.

CHAPTER 5

RECONFIGURATION POLICY APPROACH

5.1 MDP Overview

A Markov Decision Process (MDP) is sometime known as a sequential stochastic optimization, a discrete-time stochastic control and a stochastic dynamic programming. It is a study of a sequential optimization of discrete time stochastic systems. We refer to the word "Markov" since the MDP is based on a Markov Chain or a discrete time, discrete state stochastic process that satisfies the Markov property. The Markov property states that the conditional probability of any future state given an arbitrary sequence of past states and the present state depends only on the present state. The MDP is a Markov Chain excepting a transition matrix which depends on a set of actions taken at each transition step. A different action generates a different outcome. The goal is to find a decision or an action to be taken in each state, called a policy, so as to maximize the expected outcome. The outcome consists of two parts, the reward that gains from the action and the cost that incurs from the action. Notice that the outcome could be negative value if the cost is greater than the reward and it needs not be in the monetary unit. In many situation decision with the highest immediate outcome may not be good in view of future events. In our case, we state in the hypothesis that MPD policy generates the optimal expected outcome in the reconfiguration problem.

The MDP is defined through the following five elements:

1. A set of decision epochs.
2. A set of states

3. A set of actions.

4. A set of state and action dependent immediate rewards and costs.

5. A set of state and action dependent transition probabilities.

We study the discrete-time process which the epoch (or time) between transitions is a constant. Without the action, suppose the MDP contains N states, o_{ij} and p_{ij} is the outcome and transition probability when it makes a transition from state i to state j, respectively. Let $v_i(n)$ be the expected total outcome earnings in the next n transitions if the system is currently in state i. The $v_i(n)$ in the recursive form is:

$$v_i(n) = \sum_{j=1}^{N} p_{ij}[o_{ij} + v_j(n-1)] \quad i = 1, 2, \ldots, N \quad n = 1, 2, 3, \ldots \tag{5.1}$$

Equation (5.1) may be rewritten as:

$$v_i(n) = \sum_{j=1}^{N} p_{ij} o_{ij} + \sum_{j=1}^{N} p_{ij} v_j(n-1) \tag{5.2}$$

Let a quantity q_i be the outcome to be expected in the next transition out of state i or the expected immediate outcome of state i.

$$q_i = \sum_{j=1}^{N} p_{ij} o_{ij} \tag{5.3}$$

Equation (5.2) becomes the following form.

$$v_i(n) = q_i + \sum_{j=1}^{N} p_{ij} v_j(n-1) \tag{5.4}$$

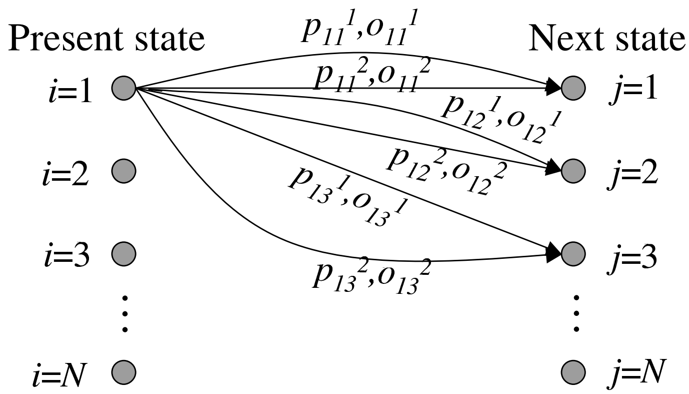


Figure 7: A State Diagram and Alternative Actions.

Now we consider a set of actions. Each outcome and transition probability has its specific values according to the action. Let o_{ij}^k and p_{ij}^k be the outcome and transition probability when moving from state i to j using the action k. Figure 7 shows the diagram of states and alternative actions.

The quantity q_i^k becomes the expected outcome from a single transition from state i using action k.

$$q_i^k = \sum_{j=1}^{N} p_{ij}^k o_{ij}^k \tag{5.5}$$

The $v_i(n)$ becomes the total expected outcome in the next n transition starting from state i and making a move (selecting action) by following the optimal policy. The optimal policy is action for each state that maximize total expected outcome. We define the policy in a decision vector $\mathbf{d}$.

$$v_i(n+1) = \max_k \sum_{j=1}^{N} p_{ij}^k [o_{ij}^k + v_j(n)] \quad n = 0, 1, 2, \dots \tag{5.6}$$

Equation (5.6) is rewritten with term q_i^k below.

$$v_i(n+1) = \max_k [q_i^k + \sum_{j=1}^{N} p_{ij}^k v_j(n)] \tag{5.7}$$

Equation (5.7) is called the value-iteration equation. We can use this recursive equation with the initial $v_i(0)$ to find the optimal policy. This method is known as a value-iteration method. It may take a long process before termination. The other method called a policy-iteration method by Howard [12] can find the optimal policy in a smaller number of iteration than the previous one. The policy-iteration consists of two parts; the value-determination operation and the policy-improvement routine.

- The value-determination operation uses p_{ij} and q_i for a given policy to solve Equation (5.8) by setting $v_N = 0$

$$g + v_i = q_i + \sum_{j=1}^{N} p_{ij} v_{ij} \quad i = 1, 2, \ldots, N \tag{5.8}$$

  where g is the gain (expected outcome per transition) of the system, $g = \sum_{i=1}^{N} \pi_i q_i$, where π_i is the probability of being in State i.

- The policy-improvement routine finds the alternative action k' that maximizes $q_i^k + \sum_{j=1}^{N} p_{ij}^k v_i$ then set $d_i = k'$ in each State i

These two operations take turns and produce the gain g. The iteration cycle could begin in either part. However it is convenient to start in the policy-improvement routine by setting all $v_i = 0$. Then let the policy-improvement select the policy. The iteration will be terminated when g does not improve i.e., we found the optimal policy in the decision vector $\mathbf{d}$.

5.2 MDP Approach

We model a policy or a set of decision by the Markov Decision Process model. The MPD elements for the reconfiguration policy are the following.

1. A set of decision epochs:

   Our MDP is a discrete-time Markov process. We assume that the time between transitions is a constant. For example, the SONET/SDH demand matrices are modified weekly so the decision is made in every week.

2. A set of states:

   A state reflects the shape and position of Pareto front in the next transition. Next we define the term used in our state.

   Definition 5.2.1 *(Virtual Topology Utilization) Given N be the number of optical node, T be the maximum number of transmitters per node and C be a capacity of lightpath. The virtual topology utilization Ψ is defined by the fraction between the volume of traffic routed through the network and the upper bound of virtual topology capacity of the network.*

   $$\Psi = \frac{\sum_{sd,x}(x \times S^x_{sd})}{N \times T \times C} \tag{5.9}$$

   Since we consider the non-blocking network, Equation (5.9) becomes:

   $$\Psi = \frac{\sum_{sd,x}(x \times \Lambda^x_{sd})}{N \times T \times C} \tag{5.10}$$

   The terms N, T and C are constant or rarely upgraded unless the total network capacity is full. Therefore Ψ relies on the volume of traffic demand. The Ψ

reflects the Pareto front curve because the reconfiguration process in a high demand volume (or high virtual utilization environment) requires more number of changes than in a low demand volume. Since the Pareto front is the function of NoC and AHT, we define state as a tuple (AHT, Ψ) for our model. Note that using both Ψ and AHT increase the state space thus in the experiment we retain the volume of traffic in each reconfiguration round so that the state becomes the term AHT only.

3. A set of actions:

   An action states how to perform the reconfiguration process or how to pick the solution on the Pareto front. Performing different solutions (different positions) on the Pareto front will transfer to different states or different Pareto front curves since the solution with low AHT trends to generate the Pareto front lower than those of high AHT.

   We define the set of actions as the different positions of the Pareto front's curve as an example as shown in Figure 8. For each required position (action), we select the solution that has a pseudo-weight factor closest to that position. The pseudo-weight factor in Equation (5.11) is calculated for each solution on the Pareto front's curve. The f_i^{max} and f_i^{min} are the maximum value and the minimum value of the objective function i respectively.

$$w_i = \frac{\left(\frac{f_i^{max} - f_i(x)}{f_i^{max} - f_i^{min}}\right)}{\sum_j^{Obj}\left(\frac{f_j^{max} - f_j(x)}{f_j^{max} - f_j^{min}}\right)} \tag{5.11}$$

   where Obj is the number of objective functions ($Obj = 2$ in our case) and x is the solution in the Pareto front. Figure 8 shows an example of five actions

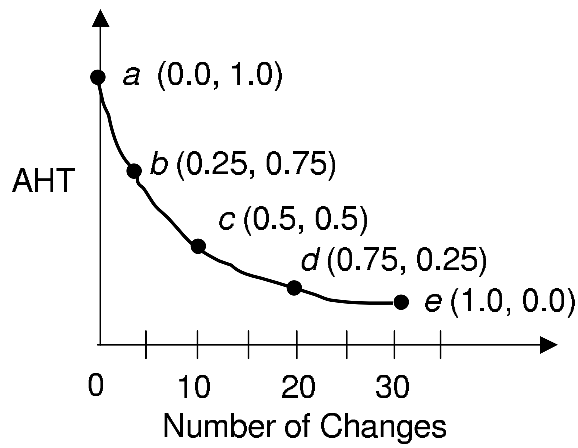


Figure 8: An Example of Pareto Front with Five Actions $[a..e]$.

$[a..e]$ with their expected pseudo-weight (w_i, w_j) where i belong to the number of changes objective and j belongs to the AHT objective. The action a is the solution at the position (0.0, 1.0). That solution does zero change and 100% or highest AHT available on the curve. The action d is the solution at the position (0.75, 0.25) which performs 20 changes in the reconfiguration process. Note that there may not exist the solution at the marked position defined in the action set but we take the solution which has the nearest pseudo-weight to the marked position.

4. A set of state and action dependent immediate rewards or costs:

   We define the outcome o_{ij}^k be the outcome when moving from state i to j using action k as shown below.

$$o_{ij}^k = r_{ij}^k - c_{ij}^k \tag{5.12}$$

where r_{ij}^k and c_{ij}^k is the immediate gaining reward and incurring cost respectively when moving from state i to j using action k. We set the reward function and cost function as linear functions shown below.

Reward function:

$$r_{ij}^k(H_{ij}^k) = \beta H_{ij}^k + c \tag{5.13}$$

where H_{ij}^k is the Average Hop-Distance of Traffic when moving from state i to j using action k, β is a weight assigned to the reward and c is a constant.

Cost function:

$$c_{ij}^k(\eta_{ij}^k) = \alpha\eta_{ij}^k + \gamma \tag{5.14}$$

where η_{ij}^k denotes the average number of changes required in the reconfiguration process from state i to j using action k, α is the weight assigned to the cost and γ is a one-time charge when start the reconfiguration operations.

Note that reward and cost functions can be any functions that reflect performance and cost such as delay, throughput, packet loss, load balance, management cost, resource cost and etc.

5. A set of state transition probabilities:

   A state transition probability P_{ij}^k is the probability when transferring from state i to j under action k. Notice that each action has its own transition probability.

   We find the optimal outcome as stated in Equation (3.24) by using Howard's Policy-Iteration method [12].

CHAPTER 6

EXPERIMENTATION DESIGN AND RESULTS

6.1 Design of Experimentation

We setup the experiment on the 6-node network and the 14-node NSFNET T1 backbone network to illustrate the Pareto front of the reconfiguration process. The 6-node network is shown in Figure 9 and the NSFNET is a mesh topology shown in Figure 10. Each link in both network is a pair of an optical fiber, one for each direction. We assume each node is working as both an access node and a routing node. Therefore the node is capable of grooming at the edge and equipped with a wavelength converter (i.e., no wavelength continuity constraint). The lightpath capacity is OC-48 for the 6-node network and OC-192 for the 14-node NSFNET network. The capacity of the DWDM is eight wavelength multiplexing, $W = 8$ thus a single fiber can carry the traffic stream up to OC-1536 (or 1536 $\times$ OC-1; OC-1 = 51.84Mbps) in the NSFNET network. The number of transceivers per node, $T_x = R_x = 6$ so there are at most six lightpaths initiated or terminated at a node. These transceivers are tunable to any wavelengths on the fiber links. There are three types of traffic streams, OC-1, OC-3 and OC-12. In the grooming capability, we assume that each node has unlimited multiplexing/demultiplexing and time-slot interchange capabilities. The lower streams can be groomed as long as the groomed traffic does not exceed the lightpath capacity. We generate the initial OC-1, OC-3 and OC-12 traffic streams by a uniform distributed random number between [0,8], [0,4] and [0,1] respectively. Table B1-B3 in Appendix B show the traffic demand matrices for the NSFNET network. The total traffic streams are OC-2976 (or 2976

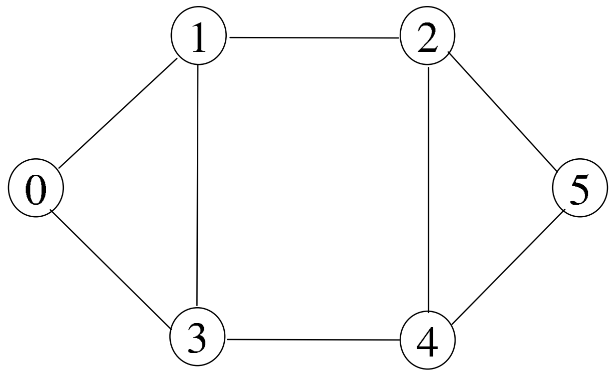


Figure 9: The 6-node Network.

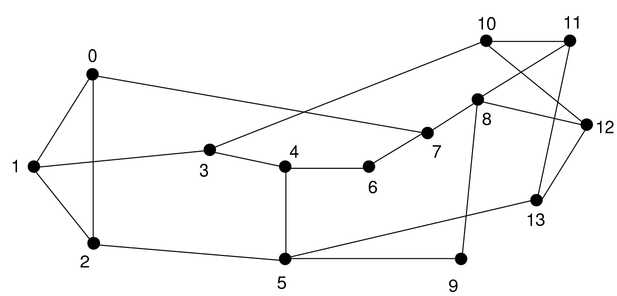


Figure 10: The 14-node NSFNET Network.

OC-1s).

We initialize a virtual topology for the first round of the reconfiguration by assigning a blank topology as a previous virtual topology and performing the reconfiguration process as usual then picking up one solution in the Pareto front. The transition of traffic in each round could be in any patterns but we simulate the change of traffic by swapping the data randomly within each traffic matrix to preserve the Ψ values. We expect to randomly swap all pairs of data or $\frac{N(N-1)}{2}$ pairs (i.e., we swap 91 pairs of data in NSFNET). The results are the new traffic demand matrices used in the next round of a reconfiguration process. We show the second round of traffic matrices in Table B4-B6 in Appendix B. The total traffic streams remain as OC-2976. We prepare thirty sets of traffic demand matrices for 29 rounds of reconfiguration processes used in the policy experiment. The SPEA parameters are set as follows: the probability of crossover = 0.6, the probability of mutation = 0.01, the dominated population size = 50 and the external non-dominated population size = 50. We run the experiments to show the Pareto front with the experimental variable including the number of generations, the number of shortest K and Ψ.

In the reconfiguration policy, we simplify the problem and reduce the state space by considering only the traffic with the same Ψ. Therefore we can ignore Ψ in the state tuple (AHT, Ψ). Now the state is defined by the AHT only. Since the AHT is a continuous value (real number), we define a discrete state based on a range of the AHT instead and use the median of the range to represent that state. The more the intervals of the AHT, the higher the accuracy of the model. However it increases the state space. The reward function, cost functions and the transition probabilities are defined. We compare the efficiency of our policy with the Immediate

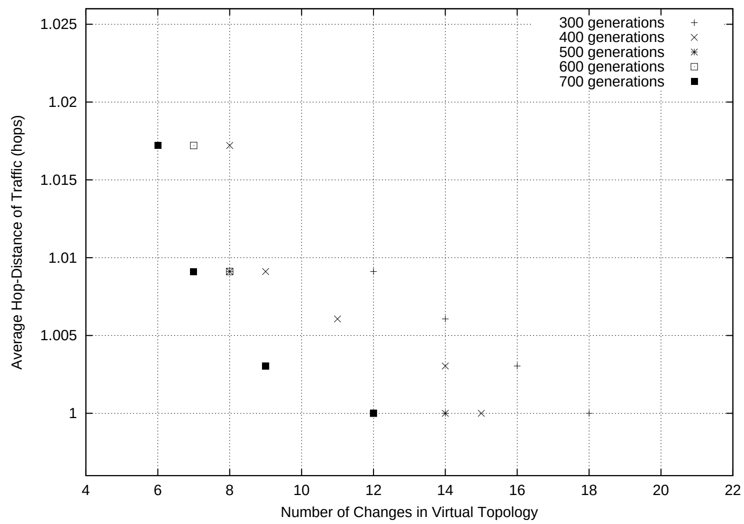


Figure 11: The Pareto Front of the Reconfiguration with Different Number of Generations at K=3 for 6-node Network.

Highest Outcome (IHO) policy over thirty sets of traffic demand matrices (29 rounds of reconfigurations).

6.2 Experimental Results

First we compare the results of reconfiguration process performed on the first round of traffic change. We plot the Pareto fronts varied by the number of generations of the SPEA for the 6-node network and the NSFNET network as illustrated in Figure 11 and Figure 12 respectively. The horizontal axis is the number of changes in the virtual topology and the vertical axis is the AHT in hops.

The plots show that the more the number of generations, the better the results (the curve is approaching the origin point). However, the result was not improved much when the number of generations are over 600 generations in the 6-node network and 1000 generations in the NSFNET network. There are more non-dominated solutions in the NSFNET network than those of the 6-node network since the NSFNET

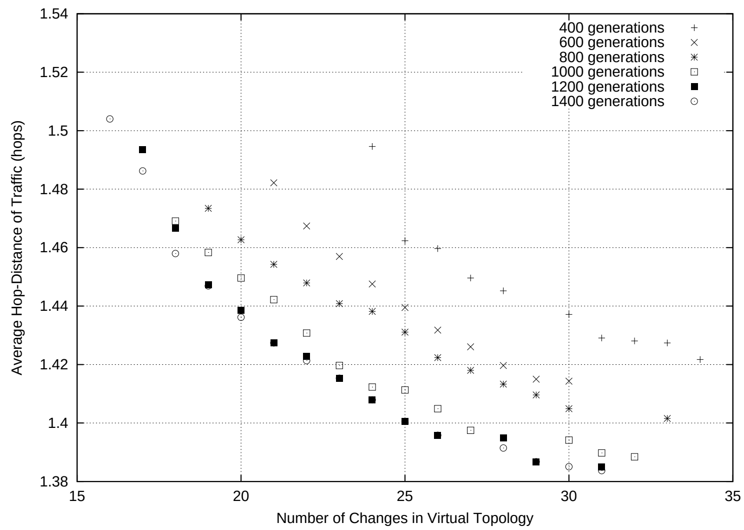


Figure 12: The Pareto Front of the Reconfiguration with Different Number of Generations at K=2 for NSFNET Network.

network is larger and thus more choices of changes. Table 1 shows the detail of reconfiguration performed on 6-node network. The initial virtual topology consists of 28 lightpaths with AHT=1.0232 hops. After the reconfiguration process, the new virtual topology contains 32 lightpaths with result in AHT=1.009 hops. The process requires seven changes (including the add operations).

Next we plot the Pareto front varying by the values of K. The result in $K = 2$ is better than those of $K = 1$ since it provides more choices of paths. However, we found that $K = 2$ generated better results than those of $K = 3$ and $K = 4$. We will use $K = 2$ in the policy experiments. Since the search space grows along the value of K. Therefore the optimal results of $K = 3$ and $K = 4$ are not yet found at 1000 generations.

Next we compare the Pareto front by the values of Ψ to show that the states in a Markov process relies on the Ψ value. In Figure 14, the curve of $\Psi = 0.355$ has a

Table 1: Virtual Topology Reconfiguration Detail on 6-node Network.

Initial Virtual Topology			New Virtual Topology			Reconfigure
Src, Dst	Path	Color	Src, Dst	Path	Color	
0, 1	0 1	4	0, 1	0 1	4	-
0, 2	0 1 2	2 2	0, 2	0 1 2	2 2	-
0, 3	0 1 3	3 2	0, 3	0 1 3	3 2	-
0, 4	0 1 2 4	0 0 0	0, 4	0 1 2 4	0 0 0	-
0, 5	0 1 2 5	1 1 0	0, 5	0 1 2 5	1 1 0	-
1, 0	1 0	2	1, 0	1 0	2	-
1, 2	1 2	5	1, 2	1 2	5	-
1, 3	1 0 3	1 0	1, 3	1 0 3	1 0	-
1, 4	1 2 4	3 2	1, 4	1 2 4	3 2	-
1, 5	1 3 4 5	0 0 0	1, 5	1 3 4 5	0 0 0	-
-	-	-	2, 0	2 1 0	4 3	add
-	-	-	2, 0	2 1 0	5 4	add
2, 1	2 1	3	2, 1	2 1	3	-
2, 3	2 4 3	3 2	2, 3	2 4 3	3 2	-
2, 4	2 5 4	1 1	2, 4	2 5 4	1 1	-
2, 5	2 4 5	4 1	2, 5	2 4 5	4 1	-
3, 0	3 0	1	3, 0	3 0	1	-
3, 1	3 1	2	3, 1	3 1	2	-
3, 2	3 1 2	1 4	3, 2	3 1 2	1 6	color
3, 4	3 4	2	3, 4	3 4	2	-
3, 5	3 4 5	1 2	3, 5	3 4 5	1 2	-
4, 0	4 3 0	3 0	4, 0	4 3 0	3 0	-
4, 1	4 2 1	1 2	4, 1	4 3 1	4 3	path&color
-	-	-	4, 2	4 3 1 2	0 0 4	add
4, 3	4 2 1 3	0 0 1	4, 3	4 2 1 3	0 0 1	-
4, 5	4 2 5	2 2	4, 5	4 2 5	2 2	-
5, 0	5 2 1 0	0 1 0	5, 0	5 2 1 0	0 1 0	-
5, 1	5 4 3 1	0 0 0	5, 1	5 4 2 1	0 1 2	path&color
-	-	-	5, 2	5 2	2	add
5, 2	5 4 2	2 3	5, 2	5 4 2	2 3	-
5, 3	5 2 4 3	1 1 1	5, 3	5 2 4 3	1 1 1	-
5, 4	5 4	3	5, 4	5 4	3	-

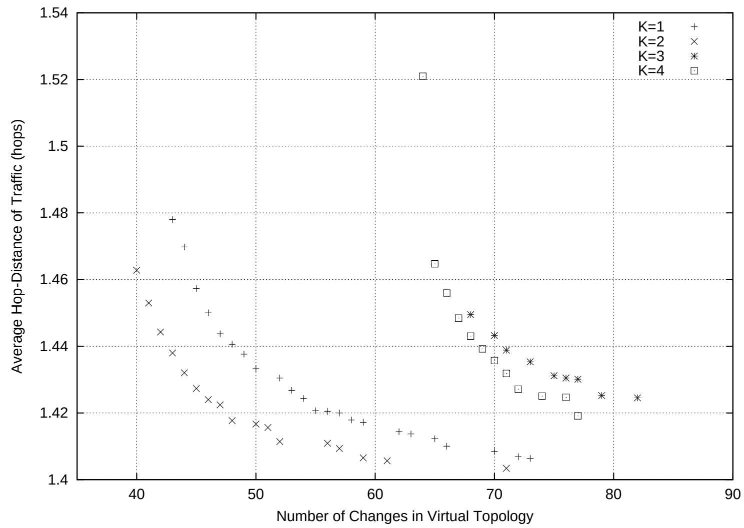


Figure 13: The Pareto Front of the Reconfiguration with Different K Values at 1000 Generations.

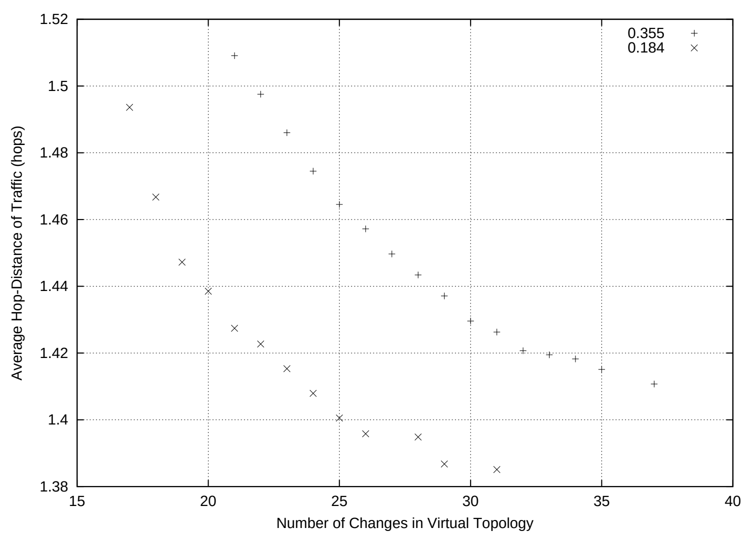


Figure 14: The Pareto Front of the Reconfiguration with $\Psi = 0.355$ and $\Psi = 0.184$.

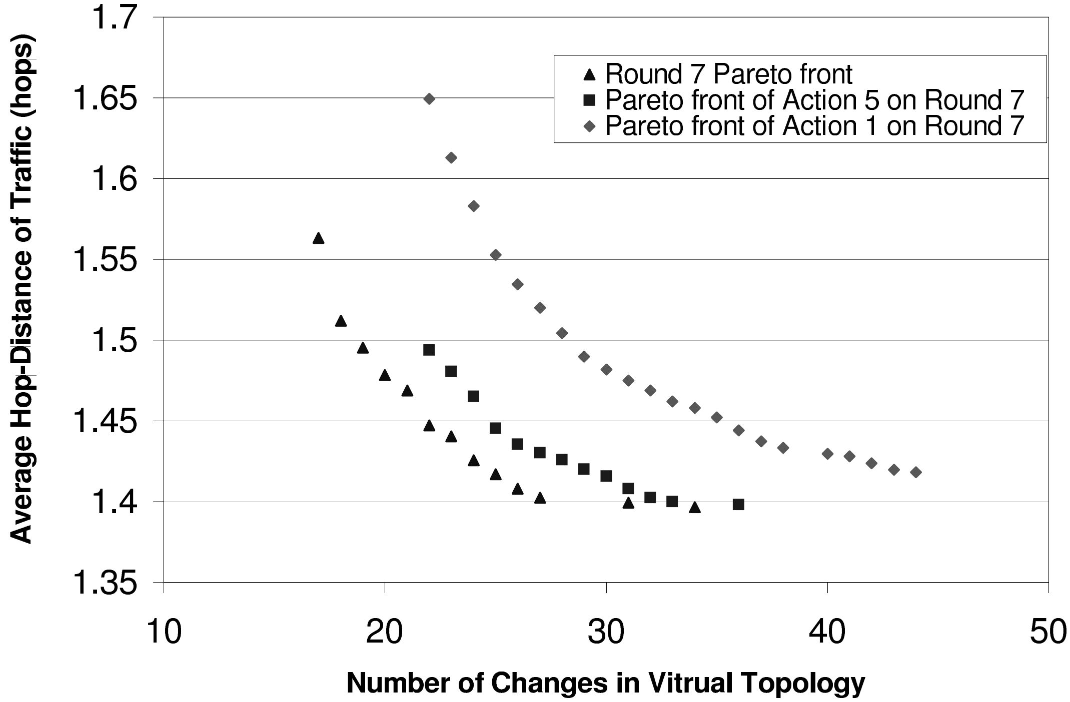


Figure 15: The Pareto Fronts of Action 1 and 5 Performed on Round 7's Pareto Front.

higher range of the AHT than those of $\Psi = 0.184$ because the solutions of the former curve do not have much room to grow and serve their high demands.

Next we show that an action with low AHT generates a Pareto front curve closer to the origin than those of an action with high AHT. We plot Pareto front of data in round 7 and perform reconfiguration process using Action 1 and Action 5 as shown in Figure 15. It shows that the Pareto front of Action 5 is closer to the origin than those of Action 1. With this property, we can estimate the state transition probabilities and their rewards and costs to find the optimal outcome.

Next we calculate the optimal policy. As we stated previously, we simplify the problem by fixing the Ψ at 0.184 in the traffic demand matrices (we swap the demand pairs to alter the data). Therefore, the states of a Markov process are stated by the

AHT only. We setup the set of states into six states with five actions. The states and their range are

1. AHT=1.375 with range [1.35-1.40],
2. AHT=1.425 with range [1.41-1.45],
3. AHT=1.475 with range [1.46-1.50],
4. AHT=1.525 with range [1.51-1.55],
5. AHT=1.575 with range [1.56-1.60] and
6. AHT=1.625 with range [1.61-1.65].

The actions are:

1. Do a reconfigure process by the solution at position [0.0, 1.0],
2. Do a reconfigure process by the solution at position [0.25, 0.75],
3. Do a reconfigure process by the solution at position [0.5, 0.5],
4. Do a reconfigure process by the solution at position [0.75, 0.25] and
5. Do a reconfigure process by the solution at position [1.0, 0.0] of the Pareto front.

State diagram in Figure 16 shows only the state transitions originated from state 1 to other states. A p_{ij}^k denotes the state transition probability from state i to state j using action k and an o_{ij}^k denotes the outcome when transferring from state i to state j using action k.

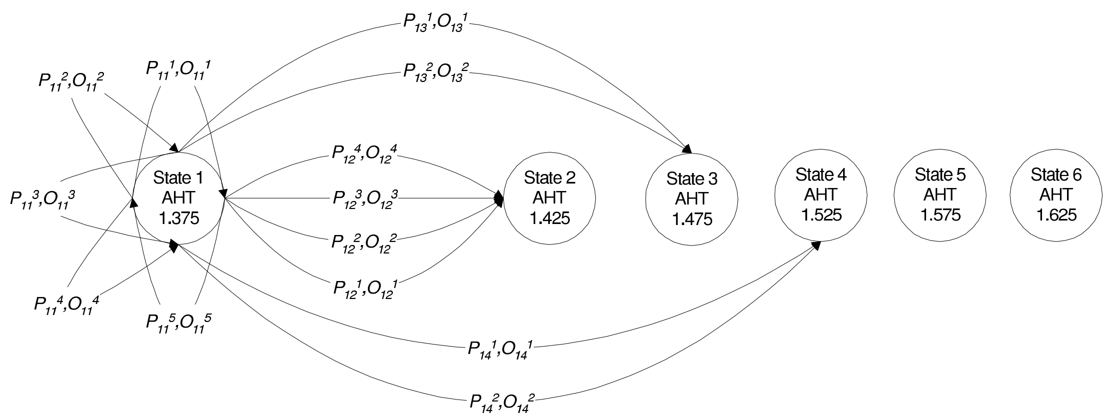


Figure 16: State Diagram of Outgoing Transitions from State 1.

Given R^k and C^k be the reward matrix and cost matrix for the action k with element r_{ij}^k obtained by the reward function and c_{ij}^k obtained by the cost function. The outcome for the transition from state i to j using action k is

$$o_{ij}^k = r_{ij}^k - c_{ij}^k$$

Since we want to show the performance of the policy, we balance the reward and cost weighted in the experiment. We do simulation to find out the margin of the H and η. Note that AHT is H and NoC(Number of Change) is η in our case. We found that the H is in the range [1.35-1.65] and η is in the range [20-50]. We create the linear function of outcome such that $H = 1.65$ when $\eta = 20$ and $H = 1.35$ when $\eta = 50$. Therefore, $\beta = \frac{(50-20)}{(1.35-1.65)} = -100$, $c = 185$, $\alpha = 1$ and $\gamma = 0$ with regard to parameters in Equation (5.13) and (5.14). The outcome function becomes $o_{ij}^k = -100H_{ij}^k + 185 - \eta_{ij}^k$. For example, the $o_{12}^2 = -100(1.425) + 185 - 24 = 18.5$ where $H_{i2}^2 = 1.425$ and $\eta_{12}^2 = 24$ when transferring from state 1 to 2 using action 2.

We apply the iterative cycle of Howard [12] to find the optimal decision.

Table 2: Initial Data for Reconfiguration Policy.

i	k	p_{ij}^k						o_{ij}^k						q_i^k
		j=1	2	3	4	5	6	j=1	2	3	4	5	6	
1	1	0.020	0.080	0.300	0.600	0.000	0.000	27.50	22.50	17.50	12.50	7.50	2.50	15.10
	2	0.050	0.150	0.600	0.200	0.000	0.000	23.50	18.50	13.50	8.50	3.50	-1.50	13.75
	3	0.800	0.200	0.000	0.000	0.000	0.000	19.50	14.50	9.50	4.50	-0.50	-5.50	18.50
	4	0.900	0.100	0.000	0.000	0.000	0.000	15.50	10.50	5.50	0.50	-4.50	-9.50	15.00
	5	1.000	0.000	0.000	0.000	0.000	0.000	11.50	6.50	1.50	-3.50	-8.50	-13.50	11.50
2	1	0.050	0.150	0.650	0.080	0.060	0.010	27.50	22.50	17.50	12.50	7.50	2.50	17.60
	2	0.010	0.680	0.310	0.000	0.000	0.000	22.50	17.50	12.50	7.50	2.50	-2.50	16.00
	3	0.200	0.800	0.000	0.000	0.000	0.000	17.50	12.50	7.50	2.50	-2.50	-7.50	13.50
	4	0.700	0.300	0.000	0.000	0.000	0.000	12.50	7.50	2.50	-2.50	-7.50	-12.50	11.00
	5	0.900	0.100	0.000	0.000	0.000	0.000	7.50	2.50	-2.50	-7.50	-12.50	-17.50	7.00
3	1	0.000	0.100	0.700	0.100	0.080	0.020	23.50	18.50	13.50	8.50	3.50	-1.50	12.40
	2	0.050	0.900	0.050	0.000	0.000	0.000	17.50	12.50	7.50	2.50	-2.50	-7.50	12.50
	3	0.350	0.550	0.080	0.020	0.000	0.000	12.50	7.50	2.50	-2.50	-7.50	-12.50	8.65
	4	0.600	0.400	0.000	0.000	0.000	0.000	6.50	1.50	-3.50	-8.50	-13.50	-18.50	4.50
	5	0.800	0.200	0.000	0.000	0.000	0.000	1.50	-3.50	-8.50	-13.50	-18.50	-23.50	0.50
4	1	0.000	0.050	0.100	0.500	0.200	0.150	22.50	17.50	12.50	7.50	2.50	-2.50	6.00
	2	0.050	0.100	0.700	0.150	0.000	0.000	15.50	10.50	5.50	0.50	-4.50	-9.50	5.75
	3	0.300	0.600	0.100	0.000	0.000	0.000	9.50	4.50	-0.50	-5.50	-10.50	-15.50	5.50
	4	0.600	0.350	0.050	0.000	0.000	0.000	6.50	1.50	-3.50	-8.50	-13.50	-18.50	4.25
	5	0.900	0.080	0.020	0.000	0.000	0.000	3.50	-1.50	-6.50	-11.50	-16.50	-21.50	2.90
5	1	0.000	0.000	0.000	0.020	0.180	0.800	22.50	17.50	12.50	7.50	2.50	-2.50	-1.40
	2	0.000	0.000	0.100	0.700	0.200	0.000	14.50	9.50	4.50	-0.50	-5.50	-10.50	-1.00
	3	0.050	0.100	0.700	0.150	0.000	0.000	7.50	2.50	-2.50	-7.50	-12.50	-17.50	-2.25
	4	0.300	0.600	0.100	0.000	0.000	0.000	-0.50	-5.50	-10.50	-15.50	-20.50	-25.50	-4.50
	5	0.600	0.300	0.100	0.000	0.000	0.000	-7.50	-12.50	-17.50	-22.50	-27.50	-32.50	-10.00
6	1	0.000	0.000	0.000	0.000	0.100	0.900	17.50	12.50	7.50	2.50	-2.50	-7.50	-7.00
	2	0.000	0.000	0.000	0.100	0.500	0.400	9.50	4.50	-0.50	-5.50	-10.50	-15.50	-12.00
	3	0.000	0.050	0.150	0.700	0.080	0.020	2.50	-2.50	-7.50	-12.50	-17.50	-22.50	-11.85
	4	0.300	0.600	0.100	0.000	0.000	0.000	-5.50	-10.50	-15.50	-20.50	-25.50	-30.50	-9.50
	5	0.600	0.300	0.100	0.000	0.000	0.000	-12.50	-17.50	-22.50	-27.50	-32.50	-37.50	-15.00

Howard's method contains two operations, the value-determination operation and the policy-improvement operation to solve as stated in previous Chapter. The complete set of state transition probabilities, outcomes and q_i^k are presented in Table 2. These data could be obtained either from the actual network or simulations.

Table 2 shows that performing reconfiguration process with the lower action number causes a higher chance to transfer up to a higher state while performing with the higher action number, it causes a higher chance to transfer down to a lower state. The lower states have more outcome than the upper state. Therefore, the MDP tries to keep the policy in lower states to gain the optimum outcome.

We apply iterative cycle of Howard [12], beginning with the policy-improvement routine by setting $v_i = 0; i = 1, \ldots, 6$ since we have six states. Then we select the initial policy that maximizes expected immediate outcome i.e., select action k in each state i that has the maximum q_i^k. From Table 2, the initial policy in the decision vector **d** are

$$\mathbf{d} = \begin{bmatrix} 3 \\ 1 \\ 2 \\ 1 \\ 2 \\ 1 \end{bmatrix}$$

The state transition probabilities matrix **P** and expected immediate rewards corresponding to this policy are

$$\mathbf{P} = \begin{bmatrix} 0.800 & 0.200 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.050 & 0.150 & 0.650 & 0.080 & 0.060 & 0.010 \\ 0.050 & 0.900 & 0.050 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.050 & 0.100 & 0.500 & 0.200 & 0.150 \\ 0.000 & 0.000 & 0.100 & 0.700 & 0.200 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.100 & 0.900 \end{bmatrix} \qquad \mathbf{q} = \begin{bmatrix} 18.50 \\ 17.60 \\ 12.50 \\ 6.00 \\ -1.00 \\ -7.00 \end{bmatrix}$$

Now the iteration begins at the Value-Determination Operation.

$$g + v_1 = 18.5 + 0.8v_1 + 0.2v_2 + 0.0v_3 + 0.0v_4 + 0.0v_5 + 0.0v_6 \tag{6.1}$$

$$g + v_2 = 17.6 + 0.5v_1 + 0.15v_2 + 0.65v_3 + 0.08v_4 + 0.06v_5 + 0.01v_6 \tag{6.2}$$

$$g + v_3 = 12.5 + 0.5v_1 + 0.9v_2 + 0.05v_3 + 0.0v_4 + 0.0v_5 + 0.0v_6 \tag{6.3}$$

$$g + v_4 = 6.0 + 0.0v_1 + 0.05v_2 + 0.1v_3 + 0.5v_4 + 0.2v_5 + 0.15v_6 \tag{6.4}$$

$$g + v_5 = -1.0 + 0.0v_1 + 0.0v_2 + 0.1v_3 + 0.7v_4 + 0.2v_5 + 0.0v_6 \tag{6.5}$$

$$g + v_6 = -7.0 + 0.0v_1 + 0.0v_2 + 0.0v_3 + 0.0v_4 + 0.1v_5 + 0.9v_6 \tag{6.6}$$

Solving Equation (6.1),(6.2),(6.3),(6.4),(6.5) and (6.6) by setting $v_6 = 0$, we obtain

$$g = 6.894 \quad v_1 = 313.6670 \quad v_2 = 255.640 \quad v_3 = 264.594$$

$$v_4 = 132.272 \quad v_5 = 138.944 \quad v_6 = 0$$

Note that we can select to set any $v_i = 0$. It will come out the same optimal policy.

Next we return the iteration to the Policy-Improvement Operations. We found the new policy that maximize the reward by the calculation in Table 3.

Now the decision vector **d** has been changed corresponding the maximum test

Table 3: The First Iteration for the Policy-Improvement.

State i	Action k	Test Quantity $q_i^k + \sum_{j=1}^{N} p_{ij}^k v_i$
1	1	200.57
	2	252.99
	3	320.56
	4	322.86
	5	**325.17**
2	1	262.53
	2	275.00
	3	280.75
	4	307.26
	5	**314.86**
3	1	247.52
	2	271.49
	3	282.85
	4	294.96
	5	**302.56**
4	1	139.17
	2	252.05
	3	279.44
	4	295.15
	5	**310.94**
5	1	26.26
	2	145.84
	3	244.05
	4	269.44
	5	**281.35**
6	1	6.89
	2	70.70
	3	144.33
	4	264.44
	5	**276.35**

quantity of each state in Table 3.

$$\mathbf{d} = \begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \\ 5 \\ 5 \end{bmatrix}$$

From the new vector $\mathbf{d}$, the matrix $\mathbf{P}$ and the vector $\mathbf{q}$ become

$$\mathbf{P} = \begin{bmatrix} 1.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.900 & 0.100 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.800 & 0.200 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.900 & 0.080 & 0.020 & 0.000 & 0.000 & 0.000 \\ 0.600 & 0.300 & 0.100 & 0.000 & 0.000 & 0.000 \\ 0.600 & 0.300 & 0.100 & 0.000 & 0.000 & 0.000 \end{bmatrix} \quad \mathbf{q} = \begin{bmatrix} 11.50 \\ 7.00 \\ 0.50 \\ 2.90 \\ -10.00 \\ -15.00 \end{bmatrix}$$

Next the iteration continues at the Value-Determination Operations.

$$g + v_1 = 11.5 + 1.0v_1 + 0.0v_2 + 0.0v_3 + 0.0v_4 + 0.0v_5 + 0.0v_6 \tag{6.7}$$

$$g + v_2 = 7.0 + 0.9v_1 + 0.1v_2 + 0.0v_3 + 0.0v_4 + 0.0v_5 + 0.0v_6 \tag{6.8}$$

$$g + v_3 = 0.5 + 0.8v_1 + 0.2v_2 + 0.0v_3 + 0.0v_4 + 0.0v_5 + 0.0v_6 \tag{6.9}$$

$$g + v_4 = 2.9 + 0.9v_1 + 0.08v_2 + 0.02v_3 + 0.0v_4 + 0.0v_5 + 0.0v_6 \tag{6.10}$$

$$g + v_5 = -10.0 + 0.6v_1 + 0.3v_2 + 0.1v_3 + 0.0v_4 + 0.0v_5 + 0.0v_6 \tag{6.11}$$

$$g + v_6 = -15.0 + 0.6v_1 + 0.3v_2 + 0.1v_3 + 0.0v_4 + 0.0v_5 + 0.0v_6 \tag{6.12}$$

Again we solve Equation (6.7),(6.8),(6.9),(6.10),(6.11) and (6.12) by setting $v_6 = 0$, we obtain

$$g = 11.5 \quad v_1 = 29.20 \quad v_2 = 24.20 \quad v_3 = 17.20$$

$$v_4 = 19.96 \quad v_5 = 5.00 \quad v_6 = 0$$

We found that the gain g is improved. We then return to the Policy-Improvement operations. The iteration continues until the gain g is not improved. Eventually we found the maximum gain $g = 16.833$ with the decision vector $\mathbf{d}$ below

$$\mathbf{d} = \begin{bmatrix} 3 \\ 4 \\ 2 \\ 5 \\ 4 \\ 4 \end{bmatrix}$$

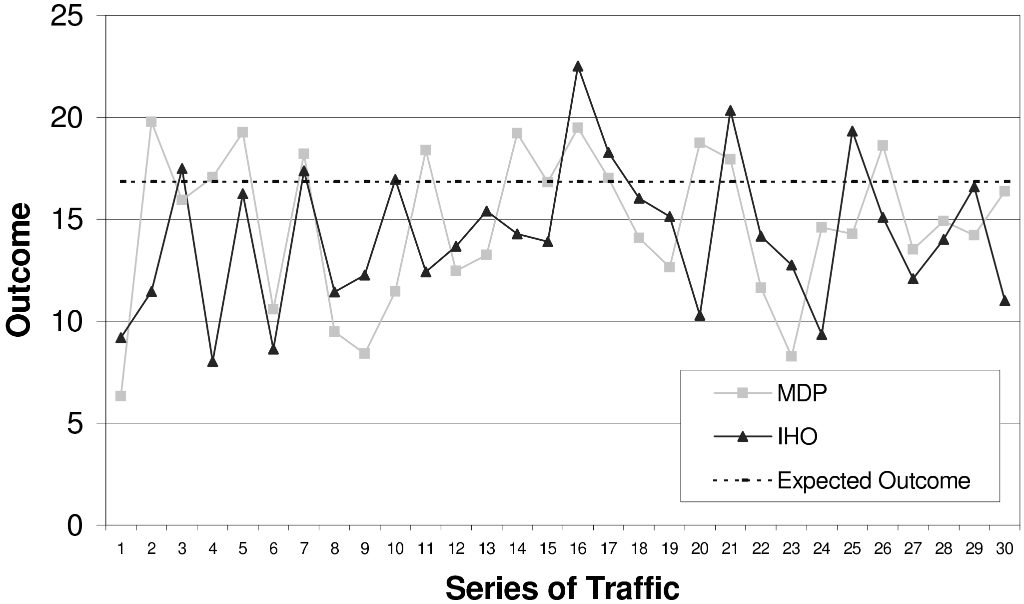


Figure 17: The Outcome of Reconfiguration from Round 1 to Round 29.

Thus the expected optimal outcome is 16.833 under this policy and MDP model. The policy in the vector **d** denotes that if the current state is 1 then perform action 3, if the current state is 2 then perform action 4, if the state is 3 then perform action 2, if the state is 4 then perform action 5 and if the state is 5 or 6 then perform action 4.

We compare our MDP policy with the Immediate Highest Outcome (IHO) policy. The IHO selects the solution in the Pareto front that produces the immediate highest outcome in the current state to perform the reconfiguration process. We ran both policy on the same set of traffic series for 29 rounds and compare the outcomes in each round as shown in Figure 17. Although the IHO selected the highest outcome in every round, it does not generate an overall outcome better than those of the MDP. Different selections lead to different states which have a different Pareto front and outcome. We show the Pareto front of the first round in Figure 19. The IHO selects

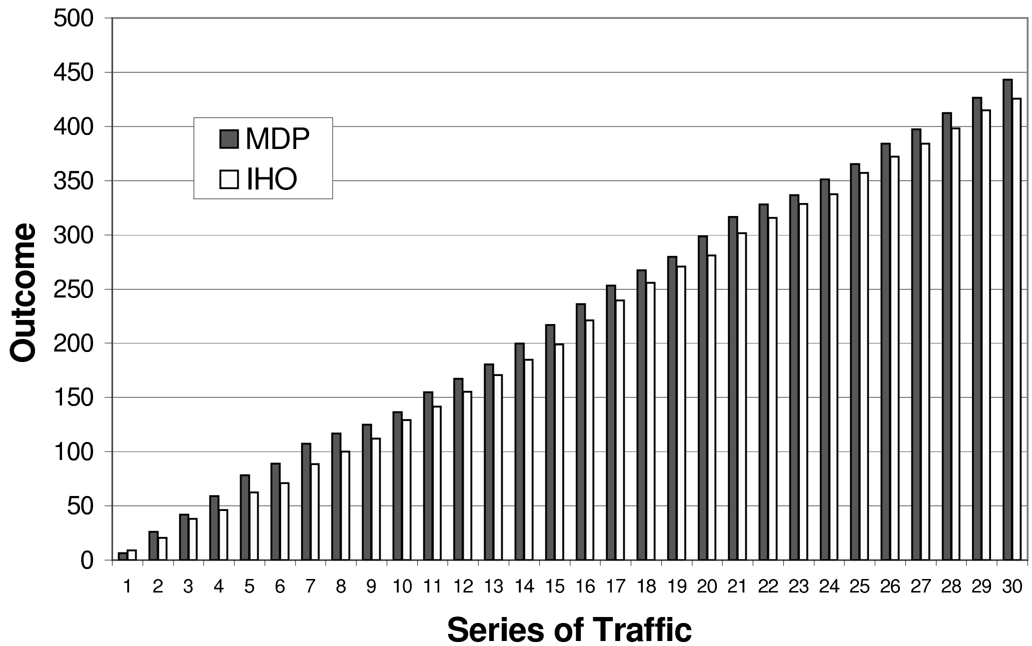


Figure 18: The Accumulative Outcome of Reconfiguration from Round 1 to Round 29.

AHT=1.438 and NoC=32 which generate outcome=9.183, while the MDP selects action 4 where AHT=1.387 and NoC=40 which generate an outcome=6.324. The IHO transfers to state 2 and the MDP transfers to state 1. The Pareto front of MDP in the second round generates a better outcome than those of IHO. In the long term, the MDP produces a greater outcome than those of IHO. We plot the accumulative outcome in Figure 18. After 29 rounds, IHO produces outcome=425.787 and MDP produces outcome=442.947.

All of the experiments performed in this dissertation were carried out using 800 MHz Intel based processor. The worst case experiment took less than 30 minutes which is considered acceptable for the reconfiguration process where the traffic demand matrices are changed in a weekly basis. The computational complexity of

First Round

AHT	NoC	Outcome
1.37466	45	2.534
1.37601	44	3.399
1.37702	43	4.298
1.38105	42	4.895
1.38508	41	5.492
1.38676	40	6.324
1.39147	39	6.853
1.39651	38	7.349
1.40188	37	7.812
1.40927	36	8.073
1.41599	35	8.401
1.42305	34	8.695
1.43044	33	8.956
1.43817	32	9.183
1.45060	31	8.94
1.45833	30	9.167
1.46875	29	9.125
1.48589	28	8.411
1.49530	27	8.47
1.51647	26	7.353

MDP

MDP Second Round

AHT	NoC	Outcome
1.38844	32	14.156
1.38978	31	15.022
1.39415	30	15.585
1.39751	27	18.249
1.40491	26	18.509
1.41129	25	18.871
1.41230	24	19.77
1.41969	23	20.031
1.43078	22	19.922
1.44220	21	19.78
1.44960	20	20.04
1.45833	19	20.167
1.46909	18	20.091

IHO

IHO Second Round

AHT	NoC	Outcome
1.40121	36	8.879
1.40457	34	10.543
1.41028	33	10.972
1.41599	32	11.401
1.42540	31	11.46
1.43548	30	11.452
1.44960	29	11.04
1.45867	28	11.133
1.46606	27	11.394
1.48152	26	10.848
1.50000	25	10
1.54805	24	6.195
1.58333	23	3.667

Figure 19: The Pareto Front with Outcome for the First Round and the Second Round of Reconfiguration using the IHO Policy and the MDP Policy.

the SPEA is $O(\mathrm{P}^2)$ where P is the population size. The routing and wavelength assignment computational complexity is $O(N^2)$ where N is the number of nodes in the network. Thus the overall complexity needed in each generation of the SPEA is $O((\mathrm{P}N)^2)$.

CHAPTER 7

RECONFIGURATION APPROACH FOR DYNAMIC TRAFFIC

7.1 Overview

In this chapter we are interested in a wavelength-routed network under dynamic traffic demand such that the traffic pattern is not known in advance. Therefore the MDP approach could not be applied to the reconfiguration policy. The reconfiguration must act immediately if the network performance falls below the acceptable point. Therefore the performance is monitored regularly and the reconfiguration process is initiated as needed. It appears that the virtual topology is self-adaptive or self-reconfiguration, since it takes action before it reaches the critical point.

The self-reconfiguration has been proposed by Gencata and Mukherjee [10]. Their approach monitors the load of each link to make a decision whether or not to perform a reconfiguration. At each observation period, a lightpath may be added if the load is above the high watermark or deleted if the load is below the low watermark. Otherwise the virtual topology needs not change. Self-tuning is attractive but it brings a set of difficulties. The optimal solution is influenced by the parameters (e.g., the level of high/low watermark and the length of observation period). Also the oscillation problem could happen if lightpath is alternately added and deleted in each of decision period. Besides, drastic changes could happen in a short period of time such as a coming of burst traffic or multiple equipment failures. In this case, the self-reconfiguration of one change at a time could not handle that traffic or outage.

7.2 A Heuristic Approach

We propose the heuristic models using the advantage of MOEA to allow mul-

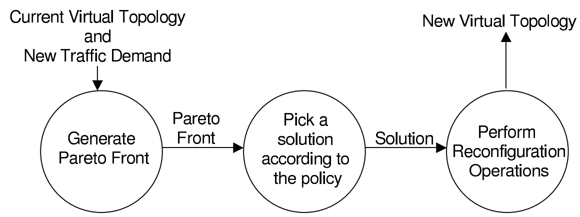


Figure 20: Data Flow Diagram of the Reconfiguration Model.

tiple changes in the reconfiguration process in each observation period. The models have a common Data-Flow Diagram (DFD) and Control-Flow Diagram (CFD) as presented in Figure 20 and 21 respectively. The first process in the DFD generates a Pareto front between AHT and NoC based on a given traffic demand, a Physical topology and a specification of optical node/link (known as constraints).

Next process is to pick a solution from the Pareto front according to the given policy. If the pattern of traffic is known in advance, the MDP policy could be applied. Otherwise some policies must be applied.

Next process is to perform the reconfiguration operations (e.g., add, delete, reroute, or switch the wavelength) according to the selected solution above. The result is the new virtual topology for a new traffic demand.

In the CFD, the Perception of Network process monitors the network status and the traffic demand then collects the data to the Reflective Control Process. If there is a significant change in a traffic demand, the process will trigger the Reflective Control Process. Then the Reflective Control Process picks the right policy and starts the Reconfiguration Process (shown in DFD). The Reflective Control Process may collect the data and analyze the pattern of traffic to make some adjustment on the policy.

Notice that the policy could be an adaptive (dynamic) one regarding to the

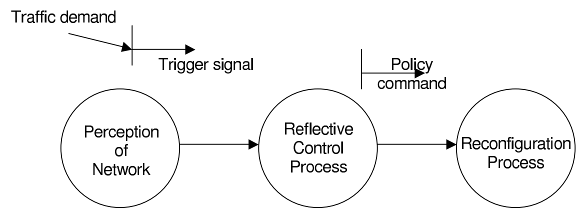


Figure 21: Control Flow Diagram of the Reconfiguration Model.

particular traffic pattern. The advantage of adaptive policy makes the model applicable to various types of traffic.

7.3 A Heuristic Algorithm

The reconfiguration problem is a trade off between the performance and the cost objectives. We still consider the performance in term of the Average Hop-distance of Traffic (AHT) and the cost in term of a number of changes in the lightpaths. We keep considering in the transport backbone network but in a short term and we consider the SONET streams that changed within one day. Since it is a backbone network, the traffic must be non-blocking. The traffic monitoring will be done hourly to trigger the reconfiguration processes.

We propose a heuristic algorithm that keeps monitoring the AHT of the network. Whenever the AHT value is above the upper-limit or when any traffic streams are blocked, the reconfiguration will be activated. In our case, the upper-limit is the maximum level of AHT that we allow in the network. If the traffic is blocked, it implies that the current virtual topology is no longer serving the demand. Hence reconfiguration is necessary. The reconfiguration starts with the searching for solutions and forming the Pareto Front. We propose two heuristic policies to pick a solution; the Pick-Min policy which picks a solution with the minimum NoC and the

Pick-Max policy which picks a solution at $NoC \leq n$ where n is the maximum number of changes allowed. If there is no feasible solution (i.e., traffic is blocked), some constraints must be relaxed likes extending the OXC ports, increasing the DWDM capability or installing an extra fiber underground. We present the algorithm with a Pick-Max policy picking a solution with $NoC \leq n$ in Figure 22.

In order to deal with a dynamic traffic, we introduce the upper-limit of AHT and NoC. The upper-limit is a breakpoint such that the cost of starting reconfiguration if AHT is below this point is worthless. In the other words, if the virtual topology is able to serve the variation of dynamic traffic, the reconfiguration is unneeded.

Not only the AHT that we concern, we also consider the utilization of the lightpath. A lightpath that serves less than $LW\%$ of its capacity (for the single-hop traffic) will be deleted. Where $LW\%$ is a low-watermark of the virtual topology in percentage. For the high-watermark, we set it at 100% to allow the maximum usage of lightpath.

7.4 Performance Study

7.4.1 Simulation Environment

We conducted the experiment on the 14-node NSFNET network with a lightpath capacity of OC-192, $W = 8$, and number of transmitters = 6 for each node. The transmitters are tunable to any wavelengths on the fiber links. There are three types of traffic streams, OC-1, OC-3 and OC-12 and the grooming capabilities are available at all nodes. We consider the dynamic traffic during one day period by imitating the actual traffic pattern of Abilene Network [1]. We found that traffic is slightly changed hourly. Therefore we alter data in the OC-1 traffic matrix only. According to the

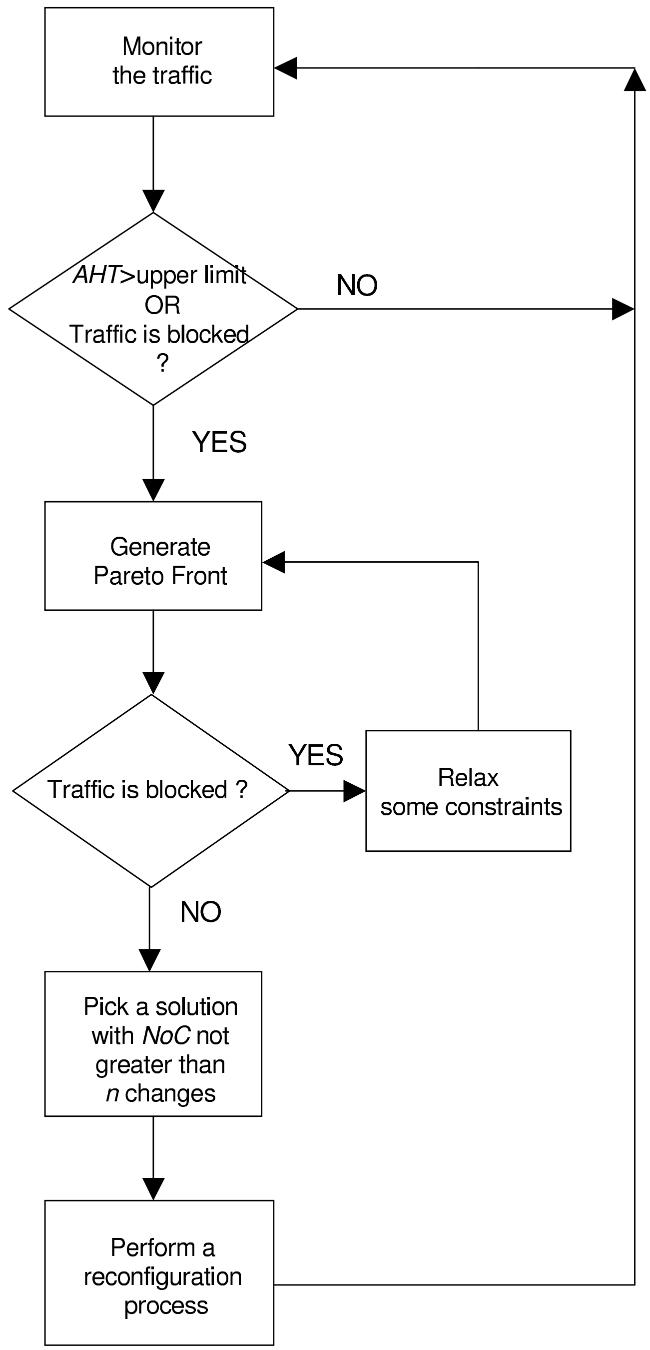


Figure 22: Heuristic Algorithm with the Pick-Max with $NoC \leq n$.

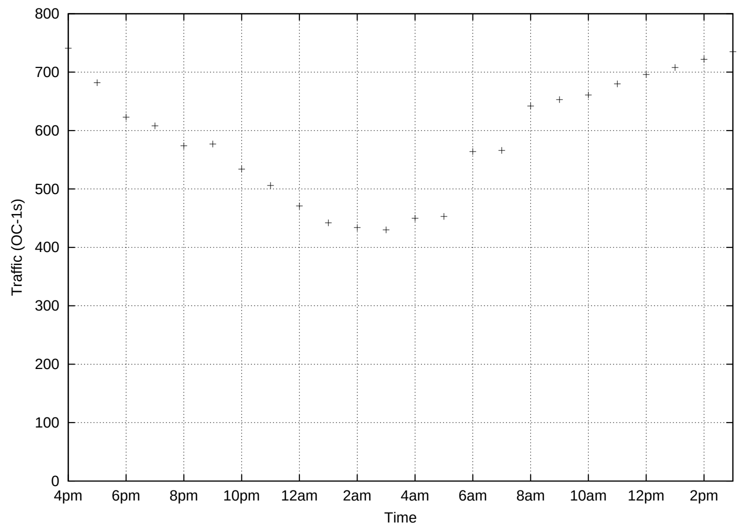


Figure 23: One-day Traffic for the Simulation.

Abilene's traffic pattern, we generate a one-day OC-1 traffic as shown in Figure 23. We decrease OC-1 traffic by randomly adding a value in the set of {-2, -1, 0, 1} to traffic in each source-destination pair and adding a value in the set of {-1, 0, 1, 2} for the increasing traffic.

The simulation was conducted on different heuristic policies that pick a solution from the Pareto front (i.e., pick a solution with the minimum NoC or pick the one with $NoC \leq n$ where n is the maximum number of changes allowed.)

7.4.2 Experimental Results

We compare the different policies that pick a solution. They are: the Pick-Min policy that picks a solution with the minimum NoC and the AHT below the upper-limit, the Pick-Max policy with $n = 10$ at most, the Pick-Max policy with $n = 15$ at most, and the Pick-Max policy with $n = 20$ at most. The chosen solution must have the AHT less than the upper limit, otherwise a solution is unacceptable

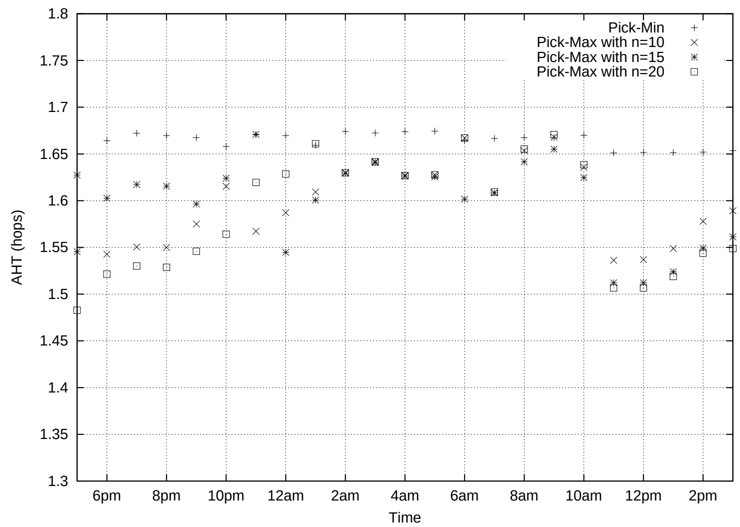


Figure 24: The AHT Results of Pick-Min, Pick-Max with n=10, Pick-Max with n=15 and Pick-Max with n=20 Policies.

and the constraints must be attuned. First experiment, we compare the performance objective (the AHT) at different policies as presented in Figure 24. We set the upper-limit of AHT at 1.675 hops. (This AHT value is obtained from the AHT after the 6^{th} state in the MDP.)

Figure 24 shows that the higher the number of changes, the better performance (lower AHT). The average AHT of each of the Pick-Max with n=20, the Pick-Max with n=15, the Pick-Max with n=10, and Pick-Min is 1.586 hops, 1.597 hops, 1.599 hops, and 1.663 hops respectively. We can see that the Pick-Max policy generates better performance than that of the Pick-Min policy. There are not much different among the Pick-Max policies except the first reconfiguration process (at 5pm). After the first reconfiguration process, the virtual topology serves the next traffic demand very well whereas a solution with $NoC < 15$ dominated the other solutions with higher NoC.

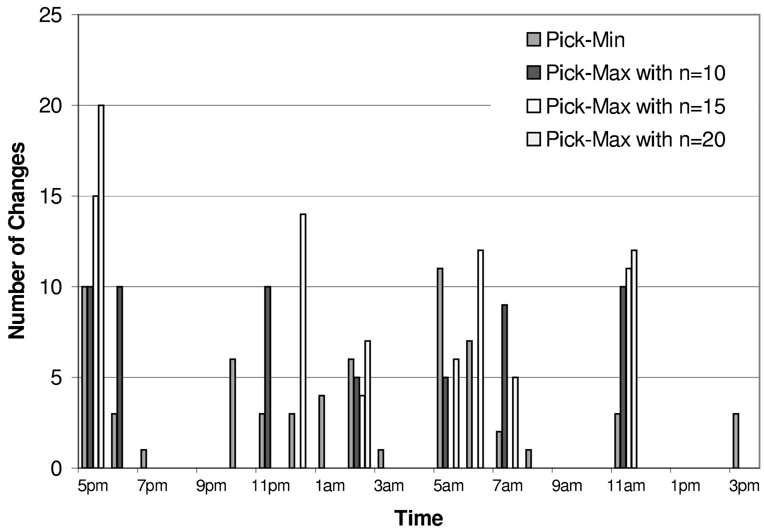


Figure 25: The Number of Changes for the Pick-Min, the Pick-Max with n=10, the Pick-Max with n=15 and the Pick-Max with n=20 Policies.

Next we compare the values of NoC at different policies as presented in Figure 25. Although the Pick-Min chooses a solution with the minimum changes in the lightpaths in each period but it takes totally 64 changes of 15 reconfiguration processes in a day. Note that the reconfiguration process takes place only at the beginning of the period which causes the traffic interruption. The Pick-Max with n=10 takes 59 changes in 7 reconfiguration processes, the Pick-Max with n=15 takes 56 changes in 5 reconfiguration processes while the Pick-Max with n=20 takes 50 changes in 5 reconfiguration processes. We see that the Pick-Min is costly since it interrupts the traffic most frequently. In this environment, the Pick-Max with n=20 is the best choice since the first reconfiguration with 20 changes produces a virtual topology that could serve traffic for a longer period.

Next we consider the utilization issue. We plot the number of lightpaths required in each hour and the overall utilization in each hour as shown in Figure 26

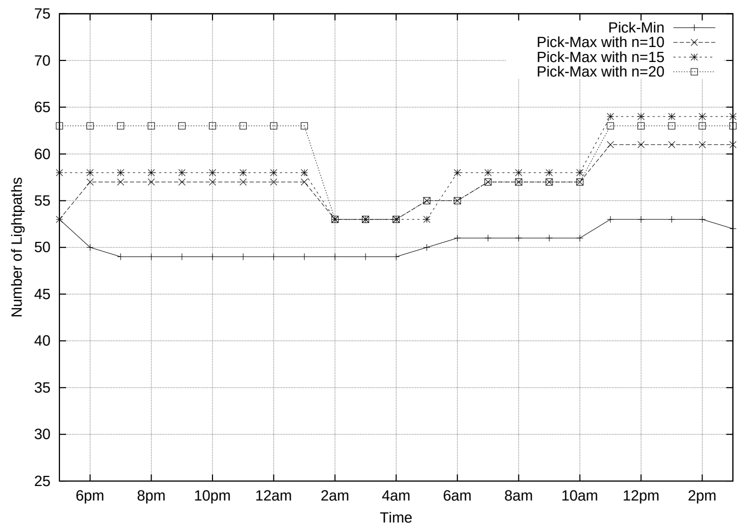


Figure 26: The Number of Lightpaths for Pick-Min, Pick-Max with n=10, Pick-Max with n=15 and Pick-Max with n=20 Policies.

and Figure 27 respectively. The overall utilization is the fraction between total traffic and the network capacity. The network capacity is the number of lightpaths multiplied by a lightpath capacity. The results show that the Pick-Min consumes less number of lightpaths than those of Pick-Max policies. Consequently, the Pick-Min results in a superior overall utilization. The overall utilization of the Pick-Max with n=20, the Pick-Max with n=15, the Pick-Max with n=10 and the Pick-Min policies are 24.56%, 25.16%, 25.77% and 29.05% respectively. It is a trade-off between the AHT performance and the utilization. The Pick-Min performs reconfiguration processes frequently thus the virtual topology is always in a good shape that fits the new demand (the unused lightpaths are deleted). So the Pick-Min policy brings in a higher utilization.

Notice that the utilization seems pretty low because the lightpaths come in a big trunk (OC-192) while the base unit of traffic is in OC-1.

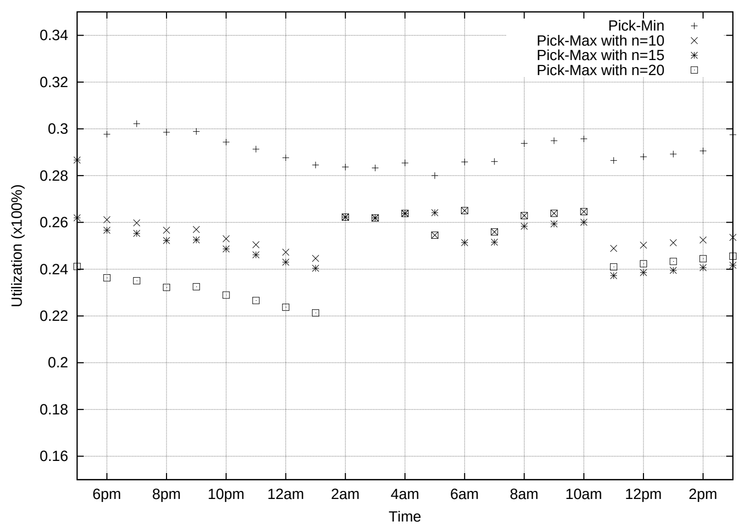


Figure 27: The Utilization for Pick-min, Pick-max with n=10 and Pick-max with n=15 Policies.

CHAPTER 8
CONCLUSIONS

8.1 Conclusions

In this dissertation, we proposed a complete model of reconfiguration in wavelength-routed optical network. Wavelength-routed optical networks are usually in a mesh topology and serve as the backbone for wide area networks. Therefore we conducted research on a mesh topology with realistic SONET/SDH traffic streams, mostly deployed in a wide area network backbone. The grooming at the access node is our key to route multiple low speed streams onto a huge lightpath capacity. We implemented grooming as part of the reconfiguration. We found that the reconfiguration problem is a multi-objective optimization since the reconfiguration objectives are the network performance optimization and the cost minimization simultaneously. These objectives are conflicting thus there are multiple solutions to satisfy the objectives. We presented the set of objectives and selected the AHT as a performance objective and considered the number of changes in lightpaths as a cost objective. Since the AHT reflects the number of O-E-O conversions at intermediate nodes. The lower the AHT, the higher the network performance. We optimize the objectives by the Pareto optimal concept. The ILP and previous literature approaches are not able to generate the set of satisfied solutions or the Pareto front since they consider one objective at a time. We propose the Multi-objective Evolutionary Algorithm to create the set of Pareto optimal solutions and the policy to pick one of the solutions for each round of reconfiguration (when traffic demands are changed). The MOEA is one of stochastic searches that work out from the random solutions. A solution is evaluated and promoted systematically if it fits well with the objectives. The fitness function, the evolutionary operations e.g., selection, crossover and mutation are determined.

Our model of reconfiguration contains two tasks: a reconfiguration process and a reconfiguration policy. We present the technique by applying the SPEA, one of the MOEAs, in the reconfiguration process. The SPEA outperforms other MOEAs and generates a perfect Pareto front that distributes solutions along the curve using a clustering method. We need the perfect curve since our policy Actions rely on it. The reconfiguration policy picks one of the solutions in the Pareto front that generates the maximum expected outcome. We present the Markov Decision Process and its elements that apply to our problem. The case study based on simulation of the 14-node NSFNET network which is considered as a large network, illustrates the Pareto front in the reconfiguration process. It shows that a reconfiguration problem is a multiple conflict objectives and a Pareto front corresponding to the problem is formed. We defined the MDP for a reconfiguration policy in the case study, compared the efficiency of MDP with the IHO policy and found that we can find the optimal policy which the expected outcome greater than that of the IHO policy. Therefore the hypothesis is accepted. The result of the MDP policy is superior to that of the IHO policy because the MDP optimizes the reconfigurations in the entire series not just a particular one like the IHO. Since the MDP model is based on estimation or probability, the accuracy of the model depends on how close the model is to the actual network.

We also extend the study of the reconfiguration on the dynamic traffic using a similar model. Since dynamic traffic is unpredictable, the pattern of traffic cannot be defined by a stochastic model. We introduce the Pick-Min and Pick-Max policies to pick the solution in the Pareto front. The experiments of a one-day traffic show the effective of the proposed heuristics. The model can make multiple changes or zero change according to the new traffic, current virtual topology and the policy. In the other words, the virtual topology is adaptable according to the traffic. In the model, we leave the cost or weight cost between the AHT and NoC open. Hence, it is applicable to any kind of traffic and cost function.

8.2 Future Work

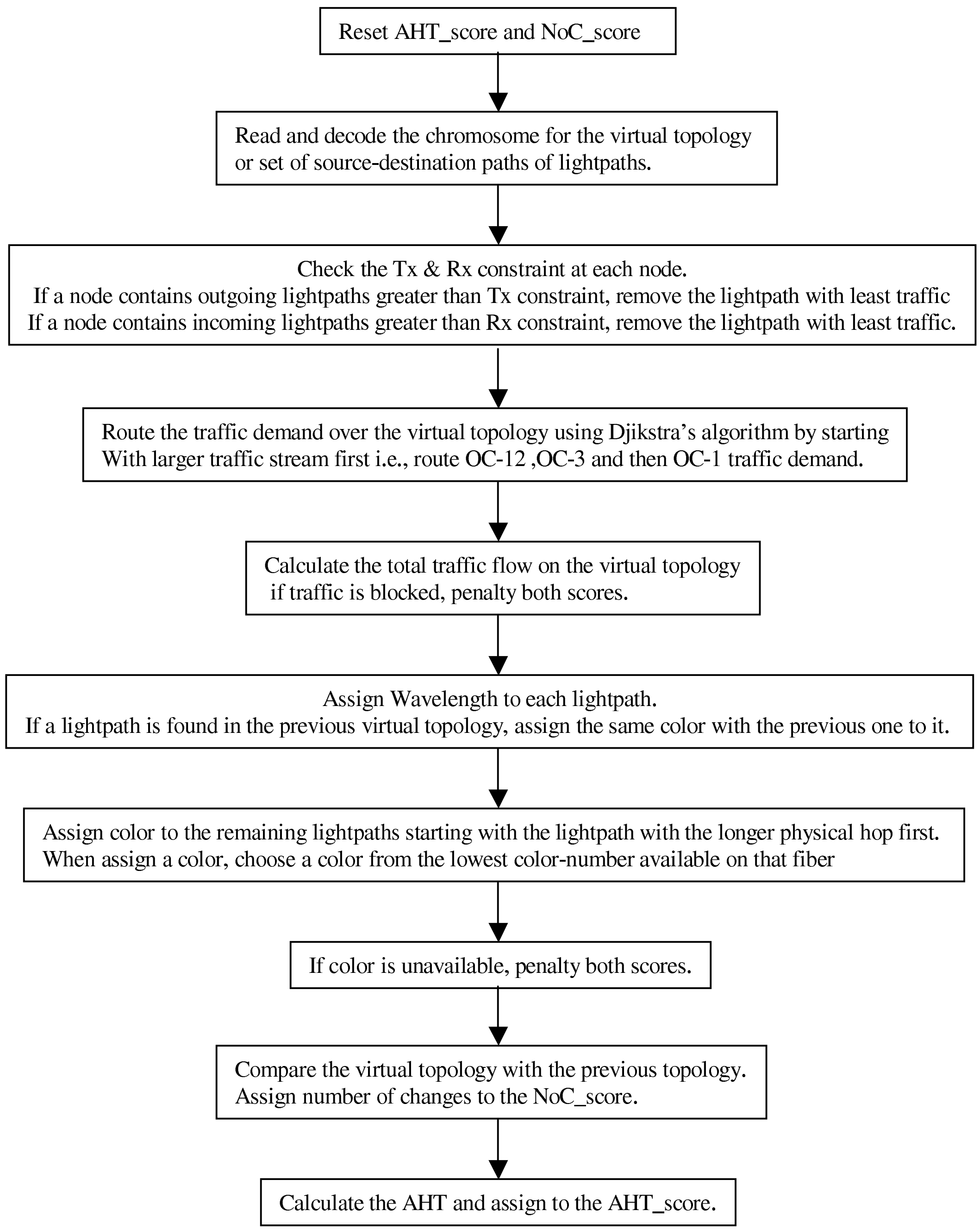
Our model is not restricted to SONET/SDH streams. It could be applied to other traffic like the fractional lightpath or packet-rate traffic. Other interesting efforts are in the definition of the cost and reward functions in the policy. They need not be linear and may be acquired from existing network parameters. Besides the accuracy of the model could be improved by refining the states of the network. The range of AHT in a state should be narrowed down and the Ψ should be included if the volume of traffic is varied in each round.

The dynamic traffic reconfiguration model is affected by several parameters like a Low-watermark and a monitoring period. It requires a deep study on these parameters to satisfy the objectives. Another research effort is the policy that could predict the dynamic traffic pattern since the recent record of traffic may predict the incoming traffic. In addition, multiple policies (adaptable policy) may be applied to the problem according to the current environment.

APPENDIX A

FITNESS (SCORE) ASSIGNMENT CHART

This chart is the part of the SPEA fitness assignment. It reads the chromosome to be decoded as a virtual topology. Traffic routing and color assignment are processed and evaluated in term of the scores. Each chromosome has two scores corresponding to the objectives.


Reset AHT_score and NoC_score
Read and decode the chromosome for the virtual topology or set of source-destination paths of lightpaths.
Check the Tx & Rx constraint at each node.
If a node contains outgoing lightpaths greater than Tx constraint, remove the lightpath with least traffic
If a node contains incoming lightpaths greater than Rx constraint, remove the lightpath with least traffic.
Route the traffic demand over the virtual topology using Djikstra's algorithm by starting With larger traffic stream first i.e., route OC-12 ,OC-3 and then OC-1 traffic demand.
Calculate the total traffic flow on the virtual topology if traffic is blocked, penalty both scores.
Assign Wavelength to each lightpath.
If a lightpath is found in the previous virtual topology, assign the same color with the previous one to it.
Assign color to the remaining lightpaths starting with the lightpath with the longer physical hop first.
When assign a color, choose a color from the lowest color-number available on that fiber
If color is unavailable, penalty both scores.
Compare the virtual topology with the previous topology.
Assign number of changes to the NoC_score.
Calculate the AHT and assign to the AHT_score.

APPENDIX B

INPUT TRAFFIC DEMAND MATRICES FOR NSFNET

Table B1: The First Round of OC-1 Traffic Demand Matrix for NSFNET.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	5	1	0	5	5	7	7	8	8	6	1	0	1
1	3	0	2	1	1	6	0	4	0	0	2	6	8	2
2	6	4	0	7	8	7	3	1	3	5	2	8	6	5
3	3	8	2	0	7	3	5	7	6	7	2	7	7	8
4	4	8	8	5	0	0	4	7	6	8	2	5	4	2
5	8	0	4	3	3	0	0	1	4	0	1	5	2	1
6	3	2	2	5	7	2	0	7	8	1	4	8	3	3
7	2	2	4	0	3	7	1	0	0	4	1	3	0	5
8	6	5	4	4	0	5	1	8	0	0	6	2	4	8
9	4	7	8	4	7	6	5	7	0	0	6	2	5	8
10	0	0	5	3	3	4	8	0	0	3	0	3	8	0
11	1	6	7	4	5	3	5	1	2	4	5	0	6	5
12	7	6	7	0	7	5	8	7	5	8	5	3	0	0
13	2	8	8	4	5	6	2	3	4	3	3	0	4	0

Table B2: The First Round of OC-3 Traffic Demand Matrix for NSFNET.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	4	1	1	2	1	3	1	1	3	3	4	0	2
1	3	0	0	2	4	2	3	0	3	3	4	4	3	1
2	2	2	0	4	1	2	3	4	2	4	3	4	2	0
3	0	4	4	0	1	3	4	1	0	2	2	2	4	3
4	1	1	1	3	0	4	0	3	4	3	3	0	0	4
5	3	4	2	1	0	0	4	1	2	0	0	3	4	2
6	2	0	3	0	4	0	0	3	1	1	2	0	1	1
7	2	2	0	3	2	4	4	0	2	3	3	2	4	0
8	4	3	2	1	3	1	3	0	0	3	0	1	1	3
9	2	1	1	0	3	0	3	0	4	0	0	1	1	1
10	2	4	4	1	1	1	1	0	3	4	0	0	4	0
11	3	2	2	0	1	1	3	4	2	0	4	0	1	4
12	0	0	1	4	3	4	3	2	1	0	3	3	0	4
13	0	4	3	0	2	1	0	0	2	0	1	3	4	0

Table B3: The First Round of OC-12 Traffic Demand Matrix for NSFNET.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	1	0	0	1	1	1	1	0	1	1	1	1	1
1	1	0	1	0	1	1	1	0	1	1	1	1	0	0
2	0	0	0	1	0	0	1	1	0	1	1	1	1	1
3	1	1	0	0	0	1	0	0	0	1	0	0	0	0
4	1	0	1	0	0	1	1	1	1	0	1	0	1	1
5	0	1	1	0	0	0	0	1	0	0	1	0	0	0
6	1	0	0	1	1	0	0	0	0	1	1	1	0	0
7	1	0	0	1	1	0	0	0	1	1	0	1	0	1
8	0	0	0	1	0	0	1	1	0	1	0	1	1	0
9	0	1	0	1	0	1	1	1	0	0	1	0	1	0
10	1	1	0	1	0	1	0	0	1	1	0	1	0	0
11	1	1	0	1	1	0	1	0	0	0	1	0	1	0
12	0	1	0	0	0	0	1	1	0	1	1	1	0	1
13	0	1	1	0	0	1	0	0	0	0	0	1	0	0

Table B4: The Second Round of OC-1 Traffic Demand Matrix for NSFNET.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	5	1	4	5	8	0	0	8	8	5	1	0	0
1	0	0	5	0	2	6	0	4	0	0	2	2	8	2
2	8	8	0	7	5	7	3	1	3	5	0	3	6	5
3	5	8	5	0	7	2	1	7	6	7	2	6	7	1
4	8	8	1	2	0	4	4	5	6	8	2	5	4	4
5	8	6	4	3	1	0	4	1	4	4	1	5	2	7
6	3	2	7	6	7	2	0	3	0	1	5	8	3	3
7	4	2	7	0	3	5	7	0	3	6	4	7	0	2
8	7	6	7	6	7	5	0	8	0	8	2	6	4	5
9	4	3	3	7	3	6	8	8	8	0	8	6	3	3
11	1	6	7	2	5	1	5	1	2	4	5	0	6	4
12	7	4	7	0	0	3	4	3	1	7	0	1	0	5
13	2	8	8	5	8	8	0	3	4	0	3	4	5	0

Table B5: The Second Round of OC-3 Traffic Demand Matrix for NSFNET.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	0	2	0	1	0	3	1	1	3	2	3	3	3
1	3	0	4	2	0	1	0	2	3	1	4	4	4	3
2	2	2	0	2	4	2	1	3	2	4	3	4	0	0
3	2	3	4	0	3	3	1	1	0	1	3	2	1	2
4	1	3	3	3	0	4	1	1	2	4	2	0	3	4
5	4	4	3	4	0	0	4	1	3	1	4	3	2	4
6	2	3	0	1	4	3	0	0	0	1	0	1	1	2
7	4	2	0	1	4	4	2	0	2	3	3	3	4	2
8	4	0	2	1	1	1	2	1	0	3	3	1	1	3
9	0	1	3	4	3	3	3	4	4	0	0	1	0	1
10	2	0	2	4	4	0	1	0	1	3	0	1	2	0
11	0	0	0	1	2	1	4	0	4	0	4	0	1	4
12	0	0	4	4	0	4	0	2	1	3	3	3	0	0
13	0	2	2	4	2	1	0	4	0	3	0	3	1	0

Table B6: The Second Round of OC-12 Traffic Demand Matrix for NSFNET.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	0	1	0	1	1	1	0	0	1	1	0	0	1
1	1	0	1	1	1	1	1	1	0	1	1	1	1	0
2	1	1	0	1	0	1	1	1	0	1	1	1	0	1
3	1	1	0	0	1	0	0	0	0	0	0	0	0	0
4	1	1	0	0	0	1	1	1	0	1	0	1	1	1
5	1	1	0	0	1	0	0	0	0	0	0	0	0	0
6	1	0	1	1	1	1	0	0	0	0	1	0	0	0
7	1	0	0	1	0	0	0	0	1	1	1	1	0	1
8	1	1	0	1	0	0	1	1	0	0	1	1	1	1
9	0	0	1	0	0	0	0	1	0	0	1	0	1	0
10	1	0	0	1	1	0	0	1	1	1	0	1	0	0
11	1	0	0	1	1	0	1	0	1	0	1	0	1	0
12	0	0	0	0	0	1	1	0	0	1	1	1	0	1
13	0	1	1	0	0	1	1	0	1	0	1	0	0	0

BIBLIOGRAPHY

1. Abilene Networking for Leading-edge Research and Education, Observatory Summary Data View, *http://abilene.internet2.edu/observatory/data-views.html*, August 28, 2003.

2. I. Alfouzan and A. Jayasumana, Dynamic Reconfiguration of Wavelength-routed WDM Networks, *Proc. of 26th Annual IEEE Conference on Local Computer Networks*, pp. 477–485, 2001.

3. I. Baldine and G. N. Rouskas, Dynamic Load Balancing in Broadcast WDM Networks with Tuning Latencies, *Proc. of 17th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '98*, Vol. 1, pp. 78–85, 1998.

4. I. Baldine and G. N. Rouskas, Dynamic Reconfiguration Policies for WDM Networks, *Proc. of IEEE INFOCOM'99*, March 1999.

5. D. Banerjee and B. Murkherjee, Wavelength-routed Optical Networks: Linear Formulation, Resource Budget Tradeoffs, and a Reconfiguration Study, *IEEE/ACM Transactions on Networking*, Vol. 8, No. 5, pp. 598–607, October 2000.

6. K. Deb, *Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach*, Vol. 3, John Wiley & Sons Inc., New York, November 1999.

7. P.H.H. Ernest, G. Mohan and V. Bharadwaj, An Efficient Algorithm for Virtual Topology Reconfiguration in WDM Optical Ring Networks, *Proc. of 10th International Conference on Computer Communication and Networks*, pp. 55–60, October 2001.

8. C. M. Fonseca and P. J. Fleming, Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generaliation, *Proc. of Fifth International Conference on Genetic Algorithms*, pp. 416–423, 1993.

9. N. Geary, N. Parnis, A. Antonopoulos, E. Drakopoulos and J. O'Reilly, The Benefits of Reconfiguration in Optical Networks, *Proc. of 10th International Telecommunication Network Strategy and Planning Symposium, Networks 2002*, June 2002.

10. A.E. Gencata and B. Mukherjee, Virtual-topology Adaptation for WDM Mesh Networks under Dynamic Traffic, *Proc. of 21st Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM'02*, Vol. 1, pp. 48 –56, 2002.

11. J. Horn, N. Nafpliotis and D. E. Goldberg, A Niched Pareto Genetic Algorithm for Multiobjective Optimization, *Proc. of 1st IEEE Conference on Evolutionary Computation*, Vol. 1, pp. 82–87, June 1994.

12. R. A. Howard, *Dynamic Programming and Markov Processs*, M.I.T. Press, Cambridge, 1960.

13. J.L.Cohon, *Multiobjective Programming and Planning*, Academic Press, New York, 1978.

14. J. Koski. *Multicriterion Optimization in Structural Design, New Directions in Optimum Structural Design*, pp. 483–503, Wiley, New York, 1984.

15. J.F.P. Labourdette, G.W. Hart and A.S. Acampora, Branch-exchange Sequences for Reconfiguration of Lightwave Networks, *IEEE Transactions on Communications*, Vol. 42, pp. 2822–2832, October 1994.

16. Narula-Tam and E. Modiano, Dynamic Load Balancing for WDM-based Packet Networks, *Proc. of IEEE INFOCOM'00*, pp. 1010–1019, October 2000.

17. P. Prathombutr, E.K. Park and J. Stach, An Algorithm for Traffic Grooming in WDM Optical Mesh Networks with Multiple Objectives, *Proc. of IEEE 12th International Conference on Computer Communications and Networks, ICCCN*, pp. 405–411, 2003.

18. B. Ramamurthy and A. Ramakrishnan, Virtual Topology Reconfiguration of Wavelength-Routed Optical WDM Networks, *Proc. of Global Telecommunications Conference*, Vol. 2, pp. 1269–1275, 2000.

19. J. D. Schaffer, Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, in Genetic Algorithms and Their Application, *Proc. of the 1st International Conference on Genetic Algorithms*, pp. 93–100, 1985.

20. N. Sreenath, B. H. Gurucharan, G. Mohan and C. Siva Ram Murthy, A Two-stage Approach for Virtual Topology Reconfiguration using Path-add Heuristics, *Optical Networks Magazine*, Vol. 2, No. 3, pp. 58–71, May/June 2001.

21. N. Srinavas and K. Deb, Multiobjective Optimization using Nondominated Sort-

ing in Genetic Algorithms, *Evolutionary Computation*, Vol. 2, No. 3, pp. 221–248, 1994.

22. R. E. Steuer, *Multiple Criteria Optimization: Theory, Comutation and Applicaion*, Wiley, New York, 1986.

23. H. Takagi, Y. Zhang and X. H. Jia, Reconfiguration Heuristics for Logical Topologies in Wide-area WDM Networks, *Proc. of IEEE Global Telecommunications Conference, GLOBECOM'02*, Vol 3, pp. 2701–2705, 2002.

24. J. Zheng, B. Zhou and H. T. Mouftah, Design and Reconfiguration of Virtual Private Networks (VPNs) over All-Optical WDM Networks, *Proc. of 11th International Conference on Computer Communications and Networks*, pp. 599–602, October 2002.

25. E. Zitzler and L. Thiele, Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp. 257–271, November 1999.

VITA

Passakon Prathombutr was born on August 8, 1968 in Nakhon Sawan, Thailand. He attended Chiang Mai University and received the Bachelor degree in Physics (HONS) in 1990. He pursued the higher degree and gradated a Masters of Science in Computer Science from Chulalongkorn University, Thailand in 1993. While he became the researcher at the National Electronic and Computer Technology Center (NECTEC) Ministry of Science, Thailand, he received the government scholarship to study the Ph.D. in data communication field in the United States. He attended Oklahoma State University and graduated in 1998 with a Masters of Science in Computer Science. He continued his studies by pursing an interdisciplinary Ph.D. in Software Architecture and Computer Networking.