Abstract

LEA, DJUANA PIGFORD. Soft Computing Approaches to Routing and Wavelength Assignment in Wavelength-Routed Optical Networks. (Under the direction of Dr. Shu-Cherng Fang.)

The routing and wavelength assignment (RWA) problem is essential for achieving efficient performance in wavelength-routed optical networks. For a network without wavelength conversion capabilities, the RWA problem consists of selecting an appropriate path and wavelength for each connection request while ensuring that paths that share common links are not assigned the same wavelength. The purpose of this research is to develop efficient adaptive methods for routing and wavelength assignment in wavelength-routed optical networks with dynamic traffic. The proposed methods utilize soft computing techniques including genetic algorithms, fuzzy control theory, simulated annealing, and tabu search. All four algorithms consider the current availability of network resources before making a routing decision. Simulations for each algorithm show that each method outperforms fixed and alternate routing strategies. The fuzzy-controlled algorithm achieved the lowest blocking rates and the shortest running times in most cases.

SOFT COMPUTING APPROACHES TO ROUTING AND WAVELENGTH ASSIGNMENT IN WAVELENGTH-ROUTED OPTICAL NETWORKS

by

DJUANA PIGFORD LEA

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

OPERATIONS RESEARCH

Raleigh

2004

APPROVED BY:

Dr. Salah E. Elmaghraby Advisory Committee	Dr. Elmor L. Peterson Advisory Committee
Dr. Shu-Cherng Fang Chair of Advisory Committee	Dr. Henry L. W. Nuttle Advisory Committee

Dedication

To my husband, children, and parents

Thank you for your love, support, and sacrifice

Biography

Djuana P. Lea was born in Hempstead, NY. In 1993, she received a BS in mathematics from Spelman College and a Bachelor of Electrical Engineering from Georgia Institute of Technology. While completing her undergraduate work at Spelman and Georgia Tech, Djuana was a NASA Women in Science and Engineering Scholar. She continued her academic studies at Clark Atlanta University where she completed a MS in mathematics in 1995. In the fall of 1996, she began graduate studies in the Operations Research program at North Carolina State University with funding from the AT&T Labs Fellowship Program. Djuana received a Master of Operations Research in 1999 and continued in the PhD program. Upon completion of the PhD, Djuana plans to move to Ohio with her husband and two daughters where she will work as an operations research analyst.

Acknowledgements

I would like to thank my Lord and Savior, Jesus Christ, for giving me the strength, courage, and ability to complete graduate school. I also thank Him for placing many people in my life that have supported and encouraged me through the tough times. I will begin by thanking my advisor and committee chairman, Dr. Shu-Cherng Fang. Thank you for your unwavering support and encouragement. Most of all, thank you for believing in me when I did not believe in myself. I also thank my committee members, Dr. Henry Nuttle, Dr. Salah Elmaghraby, and Dr. Elmor Peterson. I greatly appreciate all of your time and patience. I would also like to thank AT&T Labs for their financial support.

Finally, I thank my husband, Marcus, daughters, Maya and Camille, parents, Thomas and Velma Pigford, and sisters, Terri and Vicki. Words cannot express the gratitude I feel for each of you.

Contents

List of Figures

List of Tables

.

Chapter 1

Introduction

Corning introduced the first low-loss optical fiber over 25 years ago. Since that time, optical fiber has played an important role in information transmission [22]. Optical fiber is so attractive as a transmission medium due to its large transmission capacity, low loss, and low signal distortion [35]. A single-mode fiber has a potential bandwidth of approximately 50 terabits per second (1 Tbps = 10^{12} bps), which is nearly four orders of magnitude higher than electronic data rates of a few gigabits per second (1 Gbps = 10^9 bps) [43]. High-speed applications such as medical imaging, high-definition TV (HDTV), video conferencing, and the interconnection of supercomputers at universities and research centers, will continue to motivate research in the area of lightwave technology [35].

1.1 Optical Networks

Three generations of optical networks can be defined [22, 42]. In the first generation of networks, all links are copper coax cable and conversions between waveforms and bits take place at every node along the path between the source and destination. In the second generation of optical networks, all links are upgraded to optical fiber. This change improves performance by increasing bit rate and decreasing bit error rates but an electro-optical (e/o) conversion is still required at each node along a path. The propagation delay induced by e/o conversions in the network motivated the next generation of networks. In the third generation of optical networks, new technology is employed to exploit the large capacity of optical fiber and e/o conversions are required only at the ends of a path. Third generation networks are also called *all-optical* networks because the entire path between source and destination is passive and optical [22].

The capacity of optical fiber can be increased further through the use of multiplexing techniques. Multiplexers combine multiple signals on its input ports onto a single output port providing multiple transmissions in parallel. Simultaneous transmissions may be performed according to time slots, wave shape, or wavelength (frequency). In a *time division multiplexing* (TDM) system, nodes take turns transmitting on a single channel. A node is allowed to transmit for a specific amount of time before it must give up the channel. Each node is usually given an equal share of time but unequal sharing is also permitted. Nodes in TDM systems must be perfectly synchronized to the same time clock so that transmissions do not collide or run over the allotted time. The TDM bottleneck refers to the fact that TDM demands that each node be capable of handling the aggregate bit rate of all channels in the system. As a result, the TDM bit rate may be much higher than the processing speed attainable by electronic devices [23]. Multiplexing according to wave shape is known as *code division multiplexing* (CDM). CDM assigns a code to each transmission and also requires the source and destination nodes to synchronize to the same time base [29]. *Wavelength division multiplexing* (WDM) operates by dividing the low-loss regions of the optical transmission spectrum into many non-overlapping wavelengths. Each wavelength supports one communication channel that can be operated at a desired bit rate allowing multiple channels to coexist on a single fiber. It does not require nodes to synchronize to the same clock. Another advantage of WDM is that no equipment needs to run any faster than the bit rate of a single WDM channel, which can be chosen arbitrarily [43]. As a result, WDM is the current favorite among multiplexing techniques for optical networks.

1.2 WDM Networks

In a WDM network, the number of wavelengths that each fiber can carry simultaneously is limited by the physical characteristics of the fiber and the optical technology used to combine the wavelengths onto the fiber and separate them off [48]. Currently, the number

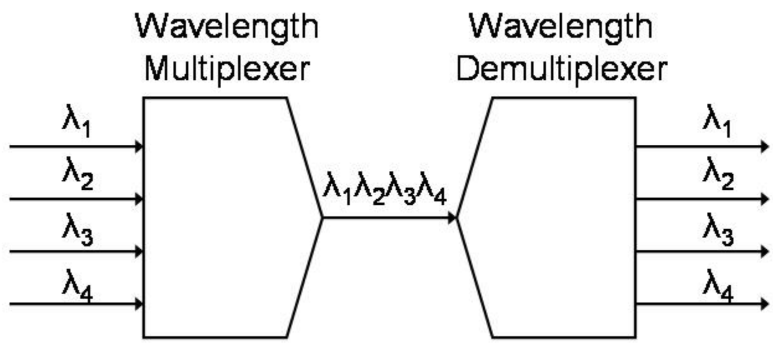

Figure 1-1: Wavelength multiplexer and demultiplexer

of wavelengths that can be carried simultaneously on a single fiber is of the order of 100 and growing [51].

WDM is achieved by using optical transmitters and receivers at end nodes. Lasers and light-emitting diodes (LEDs) can both serve as transmitters in optical networks, however, lasers are the most widely used. Receivers are simply optical filters that separate out desired wavelengths. Tunable lasers and filters allow the wavelength of light emitted or received to be adjusted as desired.

WDM network architectures can be classified into two categories: broadcast and select networks and wavelength-routed networks [46]. In *broadcast and select* networks, the N nodes of the network are connected to an $N \times N$ coupler. The transmissions from each node are broadcast over the network, allowing its reception by all nodes in the network. However, receivers may be tuned to select desired wavelengths for reception. One drawback of the broadcast and select architecture is that it requires a large number of wavelengths because typically each node transmits on a different wavelength [46]. This type of network is suitable for local area networks (LANs) and metropolitan area networks.

Wavelength routing is considered a more sophisticated and practical architecture. The nodes in *wavelength-routed* WDM networks are capable of routing wavelengths individually. This allows for reuse of wavelengths on routes that do not share common links. Therefore,

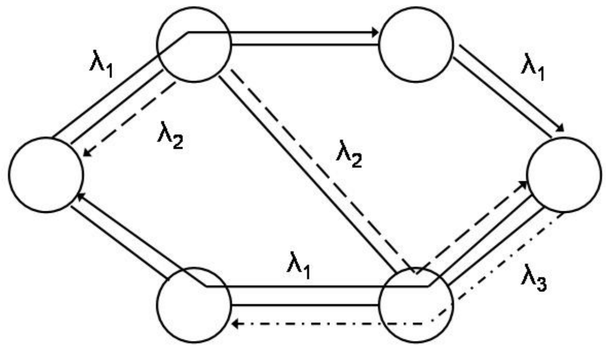

Figure 1-2: WDM network with six lightpaths

this architecture is better suited for use in wide areas. A wavelength router is an optical switch that is capable of routing a signal coming in on one wavelength at an input port to any other output port. A wavelength router may also be capable of changing the wavelength of the signal. This type of optical switch is sometimes called a *wavelength converter*.

End-users in a wavelength-routed all-optical network communicate with each other via an all-optical channel (also called a *lightpath* [13] or connection). A lightpath may span several physical links but essentially provides a circuit-switched connection between the source and destination nodes [43]. In the absence of wavelength converters, a lightpath must operate on the same wavelength on all physical links that define the lightpath. This is known as the *wavelength continuity constraint*. Therefore, two lightpaths that share a common physical link must be assigned different wavelengths in the absence of wavelength converters.

1.3 The Routing and Wavelength Assignment Problem

Given a set of lightpaths that need to be established, the routing problem in wavelength-routed all-optical networks consists of two subproblems that may be solved concurrently or sequentially. The first subproblem is to determine the physical links that will define each lightpath. The second subproblem is to assign a wavelength to each lightpath subject to the wavelength continuity constraint (if the network does not have wavelength conversion capabilities). This is referred to as the *routing and wavelength assignment* (RWA) problem.

The RWA problem where all connection requests are known in advance is called *static lightpath establishment (*SLE*)*. In this case, a lightpath remains in place once it is established. The dynamic RWA problem considers the case where connection requests arrive randomly. Once a lightpath is established, it remains in place for some amount of time and is then dismantled. This problem is referred to as *dynamic lightpath establishment* (DLE). When traffic demands are known in advance, as in SLE, a routing and wavelength assignment policy can be determined that minimizes the amount of network resources that are consumed. However, DLE does not have this advantage. For DLE, it is possible that insufficient network resources will prevent lightpath establishment for a given connection request. In this case, the request is blocked. Thus the objective for DLE is to minimize the blocking probability of connection requests. In addition to routing and wavelength assignment, DLE requires that lightpaths be taken down when no longer in use.

The RWA problem can be formulated as an integer program using various objective functions. For the static problem, one possible objective is to minimize the number of wavelengths required to establish all of the given connection requests. The following integer linear program (ILP) formulation studied in [47] seeks to maximize the number of connections that are routed subject to a constraint on the number of wavelengths. The network is represented by an undirected graph G where each node in the graph corresponds to a node in the network and each edge corresponds to a bidirectional link. The following notation is used.

Given

- J is the number of source-destination (s, d) pairs in the network.
- L is the number of physical links.
- W is the number of wavelengths available on each link. It is assumed to be the same for each link.
- ρ_j is the offered load between (s, d) pair j for $j = 1, \cdots, J$.

- P is the total number of available paths that can be used to route connections.
- $A = (a_{pj})$ is the $P \times J$ path-(s,d) pair incidence matrix where

$$a_{pj} = \begin{cases} 1, & \text{if path } p \text{ is between } (s,d) \text{ pair } j, \\ 0, & \text{otherwise.} \end{cases}$$

- $B = (b_{pl})$ is the $P \times L$ path-link incidence matrix where

$$b_{pl} = \begin{cases} 1, & \text{if link } l \text{ is on path } p, \\ 0, & \text{otherwise.} \end{cases}$$

Define variables

- m_j as the number of connections carried between (s,d) pair j for $j = 1, \cdots, J$.
- $C = (c_{pw})$ as the $P \times W$ path-wavelength assignment matrix where

$$c_{pw} = \begin{cases} 1, & \text{if wavelength } w \text{ is assigned to path } p, \\ 0, & \text{otherwise.} \end{cases}$$

The ILP is written as

$$\textit{Maximize} \qquad \sum_{j=1}^{J} m_j \tag{1.1}$$

subject to

$$\sum_{p=1}^{P} b_{pl} c_{pw} \leq 1, \qquad \text{for } w = 1, \cdots, W, \quad l = 1, \cdots, L \tag{1.2}$$

$$m_j \leq \sum_{w=1}^{W} \sum_{p=1}^{P} a_{pj} c_{pw}, \qquad \text{for } j = 1, \cdots, J \tag{1.3}$$

$$m_j \leq \rho_j, \qquad \text{for } j = 1, \cdots, J \tag{1.4}$$

$$m_j \geq 0, \quad m_s\text{: integer}, \qquad \text{for } j = 1, \cdots, J \tag{1.5}$$

$$c_{pw} = 0 \text{ or } 1, \qquad \text{for } p = 1, \cdots, P, \quad w = 1, \cdots, W \tag{1.6}$$

Constraint (1.2) ensures that the same wavelength is used at most once on each link. Constraint (1.3) is a bound on the amount of carried traffic for (s, d) pair j. Constraint (1.4) ensures that the amount of traffic carried by the network is no more than the offered load.

The total number of variables for this formulation is $P \times W + J$. Given that the number of paths P can increase exponentially with the number of network nodes and links, the RWA problem is usually addressed by using heuristic techniques. Since SLE is a special case of the more general dynamic problem, heuristic approaches are also used for DLE.

1.4 Scope and Objectives of Research

Various strategies have been proposed in current literature that address heuristic approaches to routing and wavelength assignment in wavelength-routed optical networks. However, there are relatively few studies that investigate the performance of soft computing approaches to the dynamic RWA problem. A search of the IEEE Explore database shows two recently published applications [3, 26] of fuzzy control to the RWA problem for packet-switched optical networks. Genetic algorithms have been used to solve the wavelength assignment subproblem [27] and the static RWA problem [45, 54, 55]. Simulated annealing is used for wavelength assignment planning in [50]. Two studies [30, 32] are also found in the IEEE database that examine using tabu search to solve variations of the static lightpath establishment problem.

The primary objective of our research is to evaluate the performance of soft computing approaches to the dynamic RWA problem for circuit-switched optical networks. According to Zadeh [62], soft computing is a "collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness, and low solution cost." The main constituents of soft computing are fuzzy logic, neuro-computing, and probabilistic reasoning. In our research, we will study the performance of solution techniques based on fuzzy control theory, genetic algorithms, simulated annealing, and tabu search. Since adaptive routing techniques show improved network performance over static routing approaches [6, 24], all four proposed methods are adaptive RWA algorithms for DLE where connections are routed using global information about the current state of the network.

1.5 Organization of the Dissertation

Chapter 2 provides a survey of previous studies of the RWA problem in WDM networks. Our attention is focused on the dynamic RWA problem for circuit-switched WDM networks without wavelength conversion capabilities. Chapter 3 presents the methodology used to evaluate each of the proposed approaches. Chapters 4 through 7 discuss approaches to DLE based on genetic algorithms, fuzzy control, simulated annealing, and tabu search, respectively. Results of simulation experiments using each technique are also presented at the end of the chapter for each approach. A comparison of all four methods is presented in Chapter 8 and Chapter 9 contains a summary of our research and recommendations for future study.

Chapter 2

Literature Review

2.1 Routing Techniques

Many of the routing techniques used for optical networks are borrowed from routing methods used for traditional circuit-switched telephone networks. In general, a routing algorithm can be classified as static or adaptive [17]. A *static* routing algorithm assigns a route independent of the current state of the network. Fixed and alternate routing are examples of static routing methods. Both fixed routing and alternate routing are considered *constrained* routing algorithms because all possible paths are not included in the selection process. An *unconstrained* routing algorithm considers all possible paths between the source and destination in the routing decision. *Adaptive* routing algorithms use information about the current network state to make routing decisions. Several routing methods are summarized below.

Fixed Routing

In fixed routing, a single fixed route is predetermined for each (s, d) pair. If a common wavelength is not available on all links in the route when a connection request arrives, the call is blocked. Fixed routing is very simple to implement. However, it may result in high blocking probabilities. A common example of fixed routing is *fixed shortest path* (fixed-SP) routing where each (s, d) pair is assigned a shortest route that is calculated beforehand using a shortest path algorithm such as Dijkstra's algorithm or the Bellman-Ford algorithm.

Alternate Routing

Alternate routing allows the consideration of multiple routes for each (s, d) pair. Each source node maintains an ordered list of fixed routes to each destination node. For example, the list of alternate routes for an (s,d) pair may include the k shortest paths for that pair where k is a fixed number for all (s, d) pairs. When a connection request arrives, the network attempts to place the call on each route in sequence. If no route is available from the list of alternatives or a common wavelength cannot be found, the call is blocked. In most cases, the routing tables are ordered by the number of hops to the destination. Therefore, the shortest path to the destination is usually the first route in the routing table which is known as the *primary route*. Alternate routing is also relatively simple to implement and provides a reduction in blocking probability compared to fixed routing [24].

Adaptive Routing

In adaptive routing, the route for a connection request is chosen dynamically based on the state of the network at the time of the request. The network state is defined by the routes of all connections currently in progress and wavelengths in use on those routes. An example of adaptive routing is *least-congested path* (LCP) routing. In LCP routing, the least congested path is chosen from a set of predetermined routes. The congestion of a link is determined by the number of wavelengths available on the link. The congestion of a path is defined as the congestion of the most congested link in the path. Adaptive routing methods improve network performance compared to fixed-SP and alternate routing approaches [24].

2.2 Wavelength Assignment Policies

The following wavelength assignment policies (WAPs) have been proposed in previous studies. They can be combined with one of the routing techniques described in the previous section to provide a solution to the RWA problem.

Random

This policy chooses a wavelength randomly from the set of all wavelengths that are available on the specified route.

First-Fit

The *first-fit* policy starts by numbering all wavelengths. When a connection request arrives and a route is determined, the first available wavelength with the smallest number is assigned to the route.

Least-Used/SPREAD

In the *least-used* policy, the wavelength that is the least used in the network is selected. This scheme attempts to evenly distribute the connections among all wavelengths. If the least used wavelength is not free on the required route, the second least used wavelength is selected, and so on.

Most-Used/PACK

The *most-used* policy is the opposite of least-used. It selects the most used wavelength in the network. If this wavelength is not free on the required route, the second most used wavelength is selected, and so on. This scheme attempts to pack connections into fewer wavelengths.

Wavelength Reservation

In *wavelength-reservation*, a wavelength on a particular link is reserved for traffic between a specific (s, d) pair. Reserving wavelengths ensures that a particular traffic stream will not be blocked but may decrease the global network performance because wavelengths cannot be shared.

Protecting Threshold

The *protecting threshold* policy assigns a wavelength to a single-hop connection only if the number of idle wavelengths on the link is at or above a given threshold. The objective of this method is to protect multi-hop traffic from competing with single-hop traffic and thereby balance the blocking probabilities of both types of traffic. Note that this method must be combined with another wavelength assignment scheme to determine which wavelength is assigned.

2.3 The Routing and Wavelength Assignment Problem

The RWA problem can be considered under two different traffic conditions. Static lightpath establishment (SLE) addresses the case where the set of connections is known in advance. Dynamic lightpath establishment (DLE) considers the case where connection requests arrive randomly. In either case, the routing component and the wavelength assignment component of the problem are often solved sequentially rather than simultaneously in order to make the problem more tractable.

The wavelength assignment subproblem is studied in [13]. It is shown to be equivalent to the graph coloring problem and is therefore NP-complete. Hence, heuristic methods are used for larger networks. The authors propose a greedy heuristic for the wavelength assignment subproblem that iteratively allocates a given wavelength to as many lightpaths as possible before moving to the next wavelength. SLE and DLE scenarios are simulated on a sample network. The authors conclude that a small increase in the number of available wavelengths can result in a substantial reduction in lightpath blocking probability.

ILP formulations of the RWA problem are presented in [40, 47]. In addition to the formulation presented in the previous chapter, Ramaswami and Sivarajan [47] present a second formulation of the RWA problem as a multicommodity flow problem. The objective is to maximize the number of connections that are successfully routed subject to capacity constraints and traffic demands. Using this formulation, the authors derive an upper bound on the number of connections that can be carried. This upper bound is equivalent to a lower bound on the blocking probability. They also show that these bounds are achievable when the number of wavelengths is large.

The use of the bounds derived in [47] is illustrated by simulating a heuristic RWA algorithm based on shortest paths. The set of shortest paths between an (s,d) pair is ordered in some manner. The set of wavelengths is also ordered. A new connection is routed on the first path that has an available wavelength. If more than one wavelength is available on that path then the first one is chosen. If a path can not be found , the connection is blocked. A comparison of results for the heuristic approach to the theoretical bounds show that the performance of their simple heuristic is "very close to that of an optimal algorithm."

The ILP formulation in [47] requires the enumeration of path candidates. As a result, the size of the ILP increases with the number of path candidates and may increase exponentially with the number of nodes or links. A different ILP formulation of a similar problem is presented in [40] that maintains a fixed size. However, this alternate formulation is still computationally complex.

A multi-stage approach is presented in [6]. The large RWA problem is solved in four stages that are solved sequentially with the output of one stage used as input for the next stage. The first stage uses a linear programming formulation of the RWA problem without the wavelength continuity constraint. It is assumed that at most one lightpath may exist from a source to a destination. The objective is to minimize the flow on each link. This objective corresponds to minimizing the number lightpaths passing through a particular link thereby minimizing the number of wavelengths required to establish the given set of connection requests.

The number of equations and variables in this formulation grows rapidly with size of the network. Therefore the authors propose reducing the problem size by restricting path candidates to a set of shortest paths between a given (s, d) pair. After determining the alternate shortest paths, the problem size is further reduced by relaxing the integrality constraints. The resulting linear program (LP) is solved and a randomized rounding approach is employed to obtain an integer solution to the routing subproblem. A sequential graph coloring technique [43] is then used to assign wavelengths to each lightpath in accordance with the wavelength continuity constraint.

Other heuristic methods also employ shortest path algorithms for the routing subproblem. Baroni and Bayvel [8] assign each (s, d) pair a shortest path. Each (s, d) pair is then considered individually and an alternative shortest path replaces the original path if the traffic load is lighter on the alternate path. The algorithm continues until no more substitutions are possible. A greedy algorithm is then used for wavelength assignment with wavelengths allocated to the longest paths first. Alanyali and Ayanoglu [2] use a similar approach. In their study, it is assumed that each link in the network has an associated cost and the cost of establishing a lightpath is the sum of the costs for each link in the lightpath. The RWA algorithm seeks to minimize the total network cost. Initially, each connection is assigned an arbitrary route and wavelength and the associated cost is calculated. Each connection is then considered individually and an alternate route and wavelength assignment replaces the original if it reduces the total cost of the network.

Birman and Kershenbaum [10] perform a case study in which lightpaths are dynamically established and torn down in response to a random pattern of arriving connection requests and connection holding times. The authors propose fixed routing and alternate fixed routing methods to dynamically route lightpaths. Wavelength reservation and protecting threshold are two techniques used for wavelength assignment.

Harai, Murata, and Miyahara [24] analyze the performance of alternate routing methods and the effects of different wavelength assignment policies. They propose a *limited alternate routing* method where longer primary routes are given more alternate routes. Similar to the protecting threshold technique, the objective of this method is to improve the performance of connections with more hops at expense of traffic with fewer hops. The performance of limited alternate routing in combination with random, first-fit, and most-used wavelength assignment policies is simulated. First-fit wavelength assignment outperformed the random method. However, the most-used wavelength assignment policy yielded results similar to the first-fit method.

In [41], Moktar and Azizoğlu propose a method of performing DLE jointly and adaptively. In this method, a link-wavelength status matrix tracks the availability of wavelengths on each link in the network. This matrix is searched for an available route from source to destination nodes. Least-used, most-used, and random wavelength assignment techniques are compared. The most-used scheme showed the best performance on their experimental network.

Li and Somani propose two strategies for DLE in [33]. The first dynamic routing algorithm called fixed-paths least-congestion (FPLC) uses alternate routing and least-used wavelength assignment. The second algorithm also searches a set of alternate routes but instead of searching all of the links on each route for idle wavelengths, only the first k links on each route is searched. The value of k is determined by the network topology and performance requirements. Simulations show that the first algorithm performs better on networks with an irregular topology.

Genetic algorithms are used for SLE in [45] and [55]. The authors of [45] study the RWA problem with the objective of establishing the maximum number of connections with the least number of wavelength converters. A network with L links and W wavelengths per link is thought of as a network with W single wavelength links between nodes resulting in a total of $L \times W$ links in the network. Each link and connection request is numbered for identification. Potential solutions are encoded as an array with $L \times W$ elements where each array element has a value of -1 if the corresponding link is free. Otherwise, elements with the same number represent the route for the corresponding request with the same number. Crossover and mutation operations are performed to extend the range of the population. The authors found that increasing the number of wavelength converters from 0 to 1 yielded more improvement in the number of established connections than an increase from 1 to 2 or larger.

Sinclair [52, 53, 54, 55] develops a tool for optical network modeling and design based on a genetic algorithm/heuristic hybrid approach. Static connection requests are considered with the objective of minimizing the number of wavelengths required to route all connection requests. A cost model that attempts to approximate the costs to purchase, install, and maintain network equipment is used to evaluate solutions. Unlike basic genetic algorithms, potential solutions are not encoded in this study. Instead, the author uses an object-oriented representation of networks where the network objects themselves undergo four genetic operations including path mutation, crossover and two rerouting operators. Experimental results show that the hybrid method was able to find solutions with lower costs when compared with other SLE algorithms.

Chapter 3

Methodology

In our research, we evaluate the performance of four adaptive RWA algorithms for DLE. This chapter provides an overview of each solution method. In addition, descriptions of the test networks and simulation conditions are presented.

3.1 Proposed Solution Methods

Genetic Algorithm

Genetic algorithms (GAs) are search algorithms based on Darwin's theory of survival of the fittest [11]. The idea was first developed by Holland and his colleagues in 1965, although he published a book on the subject ten years later in 1975 [25]. In contrast to ordinary local search techniques, GAs perform a local search on a *population* of current solutions rather than just one. In addition, the best solutions of the current population are mutated and recombined to create *children* that retain the best characteristics of their *parents*.

We wish to study the dynamic RWA problem where the number of potential paths for lightpath establishment is unconstrained. This number can grow exponentially with the number of nodes and links in the network. Therefore, we chose a GA-based approach to provide an efficient way of searching the potentially large number of possible solutions.

Fuzzy Control

Fuzzy set theory was first documented by Lotfi A. Zadeh in 1965 [58]. Fuzzy logic, which is based on fuzzy set theory, provides a way of dealing with uncertainty when modeling real situations. According to Zadeh, fuzzy logic is related to the formal principles of approximate reasoning [61]. He also states:

> In more specific terms, what is central about fuzzy logic is that, unlike classical logical systems, it aims at modeling the imprecise modes of reasoning that play an essential role in the remarkable human ability to make rational decisions in an environment of uncertainty and imprecision. This ability depends, in turn, on our ability to infer an approximate answer to a question based on a store of knowledge that is inexact, incomplete, or not totally reliable [61].

Fuzzy control (FC) theory applies the concepts of fuzzy logic to automatically control a system. It has been used in many commercial products to control various operations in washing machines, automatic cameras, and automatic transmissions in cars. For our research, we use fuzzy control to deal with a dynamic problem where the availability of network resources at any given time is uncertain.

Simulated Annealing

Simulated annealing (SA) is a heuristic search technique introduced independently by Kirkpatrick et al. [31] in 1982 and Cĕrny [12] in 1985. The benefit of using SA, as opposed to a simple local search technique like hill-climbing, is that SA may allow the acceptance of a neighboring solution even if it is worse than the current solution. This prevents the algorithm from being trapped in a local optima. As a result, a greater area of the solution space is covered.

Tabu Search

Tabu search (TS) provides another way of escaping local optima. Unlike SA, TS considers several neighbors of a current solution and chooses one to accept. The algorithm does not simply accept the best neighbor but takes into account how the neighbor was reached. If a neighboring solution was obtained using moves that were performed recently, the neighbor would be labelled *taboo* and could not be accepted as the new current solution. Thus, an implementation of TS contains some form of memory to record information about recent moves. The present form of TS was first presented by Glover in 1986 [18]. There are many aspects to TS that will not be discussed in this study. An exhaustive treatment of TS and all of its elements can be found in [19].

3.2 Experimental Networks

NSFnet

The NSFnet backbone network shown in Figure 3-1 is a wide-area network (WAN) developed under the direction of the National Science Foundation (NSF). It consists of 14 nodes and 21 links. The NSFnet served as the main government network linking universities and research facilities. Although it was dismantled in 1995, the network topology is still used in many research studies as an experimental network [47, 51].

4×4 Mesh-Torus Network

In contrast to NSFnet, the 4×4 mesh-torus network has a regular topology with 16 nodes and 32 links. It is also used in studies to test the performance of routing algorithms on a network with regular topology.

3.3 Description of Simulations

Simulations are performed using each approach for the NSFnet and the 4×4 mesh-torus network We consider each network with 4 and 8 wavelengths per link. In addition, we study the effect of varying the wavelength assignment policy within each algorithm. The random, first-fit, least-used and most-used wavelength assignment strategies described in Chapter 2 are evaluated. Each method is compared to fixed-SP routing and alternate routing using the corresponding wavelength assignment policy. For the alternate routing method, each

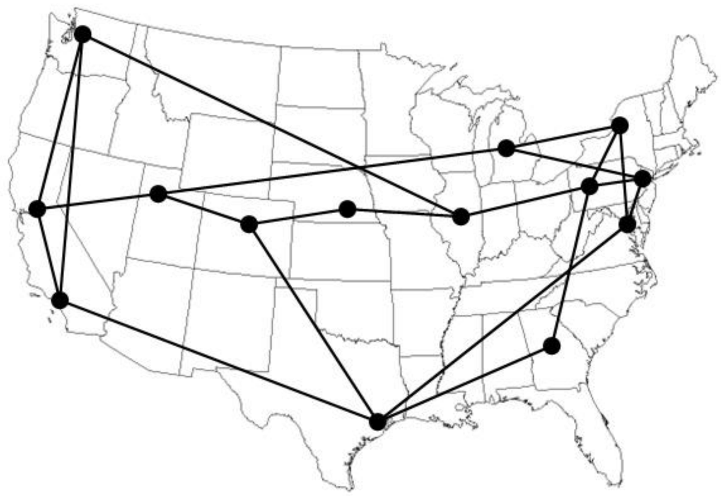

Figure 3-1: NSFnet backbone network

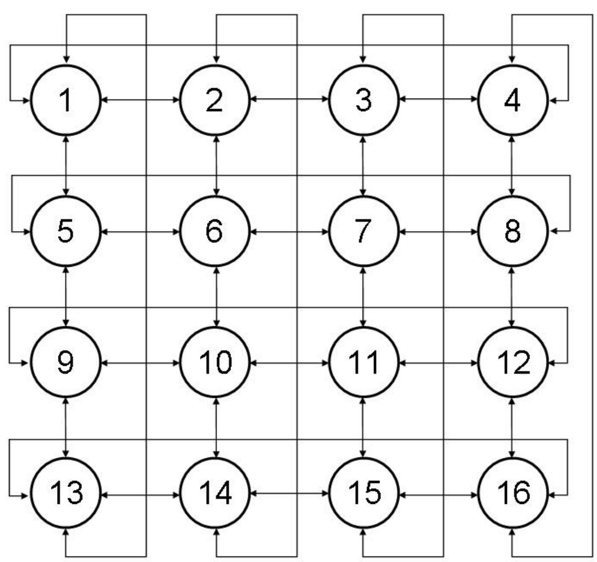

Figure 3-2: 4 × 4 Mesh-torus network

Table 3.1: Summary of simulations

Simulation	**Network**	**# Wavelengths**	**WA Policy**
1	NSFnet	4	Random (R)
2	NSFnet	4	First-Fit (FF)
3	NSFnet	4	Least-Used (LU)
4	NSFnet	4	Most-Used (MU)
5	NSFnet	8	R
6	NSFnet	8	FF
7	NSFnet	8	LU
8	NSFnet	8	MU
9	4×4 Mesh-Torus	4	R
10	4×4 Mesh-Torus	4	FF
11	4×4 Mesh-Torus	4	LU
12	4×4 Mesh-Torus	4	MU
13	4×4 Mesh-Torus	8	R
14	4×4 Mesh-Torus	8	FF
15	4×4 Mesh-Torus	8	LU
16	4×4 Mesh-Torus	8	MU

(s, d) pair is assigned 3 possible routes in the routing table where the first route is a shortest path. Thus, for each proposed method presented in Chapters 4 through 7, a total of 16 simulations are performed. A summary of the simulation conditions are shown in Table 3.1. Simulations are coded in Matlab and run on a computer with an Intel Celeron 2.4GHz processor.

A dynamic traffic model in which connection requests arrive at each node according to a Poisson process with network wide arrival rate β is used. The connection holding time is exponentially distributed with mean $1/\mu$. Thus the load per source-destination (s, d) pair is $\rho = \beta/N(N-1)\mu$. A node may be involved in multiple sessions and parallel sessions may occur between an (s,d) pair. For each case, 10 repetitions of 2000 connection requests are simulated. The average blocking rate (ABR) and average running time per connection (ART) are used as performance measures when comparing the algorithms and wavelength assignment policies. The average blocking rate is computed by dividing the number of blocked requests by the total number of requests. The average running times are measured in seconds. A plot of the ABR from each simulation described in Table 3.1 is provided in the appropriate chapter. These plots show the average blocking rates obtained from our proposed routing method in comparison to a fixed and alternate routing strategy. We also provide tables that compare the ABR and ART for the four wavelength assignment policies tested. In order to further illustrate the difference in the ABR and ART for our proposed approaches and the fixed or alternate routing methods, each table includes the percent increase/decrease between the fixed or alternate method and the proposed algorithm. For example, if our proposed algorithm has an ABR of .0432 and the fixed method has an ABR of .2829, the using the fixed algorithm results in a $(.2829/.0432) * 100 = 655\%$ increase in the average running time. A similar calculation is performed for the average running time per connection.

There are two type of requests used in each algorithm: connection requests and termination requests. Connection requests arrive at individual nodes and contain the source node s, destination node d, and holding time h for the connection. Termination requests are setup by each node once a lightpath has been established. These requests include the links that compose the lightpath, the assigned wavelength, and the time at which the lightpath is to be disconnected. The network maintains a $L \times W$ link-wavelength status matrix S where

$$S_{lw} = \begin{cases} 1, & \text{if wavelength } w \text{ is in use on link } l, \\ 0, & \text{otherwise.} \end{cases}$$

This matrix is used by each algorithm to check the availability of wavelengths on network links. The network also maintains a table T that holds the information for termination requests.

Chapter 4

A Genetic Algorithm Approach

4.1 Genetic Algorithms

Genetic algorithms (GAs) are search algorithms based on Darwin's theory of survival of the fittest [11]. As a result, many terms from the field of biology are found in the literature concerning genetic algorithms. A GA starts with a set of potential solutions (*population*). Each solution (*individual*) is *encoded* as a string (*chromosome*) and is evaluated using some measure of *fitness*. The actual solution that can be evaluated is called the *phenotype* and the encoded solution is referred to as the *genotype*. At each iteration, the GA generates new solutions (*children*) by performing genetic operations on select members (*parents*) of the solution set. The new solutions are also evaluated for fitness and the most fit solutions are selected to form a new population for the next iteration.

Mutation and *crossover* are the two types of *genetic operators*. These operations are performed on the genotype representation of the solution. The crossover operator creates children by combining parts of two distinct parents. The mutation operator makes changes in a single parent to create new children. Crossover and mutation occur with probabilities p_c and p_m, respectively during each iteration of the GA. Crossover is usually used as the principle operator while mutation functions as a background operator [16]. Introductions to some commonly used genetic operators, chromosome encoding methods, and selection techniques are given in [16]. The basic procedure for a GA is shown in Algorithm 4.1.

Algorithm 4.1 Basic Genetic Algorithm

Initialize:
    Iteration counter: t=0.
    Initial population: P(0).
Evaluate fitness of P(0).
While (termination criterion not fulfilled)
    Perform crossover operation on selected individuals: C(t)=c(P(t)).
    Perform mutation operation on selected individual: M(t)=m(P(t)).
    Evaluate fitness of C(t) and M(t).
    Select new population: P(t+1)=s(P(t) ∪ C(t) ∪ M(t)).
    Set t=t+1.
End

4.2 GA Design for the RWA Problem

Genetic algorithms were created by Holland [25] as a generic method using binary encoding and binary genetic operators. For more complex problems, solution encoding and genetic operators are usually adapted to the specific problem. In this section, we discuss the encoding and genetic operators used in the proposed GA-based adaptive RWA algorithm. These operators are similar to those described in [16] and [44] for packet-switched ATM networks. However, modifications are made for adaptation to wavelength-routed optical networks.

Encoding

Potential solutions to the RWA problem consist of a route r defined on the physical network topology and one wavelength w assigned to every link on the route. For a network with N nodes and W wavelengths per link, a route r is represented by a list of nodes as they occur along the path, i.e. $r = (r_1\ r_2\ r_3 ...\ r_{K_r})$ where K_r is the number of nodes in route r ($K_r \leq N$) and r_i is the index of the i^{th} node visited along the route. Each solution is encoded in the following manner.

1. List the node numbers in the order they occur in r from source node to destination node.

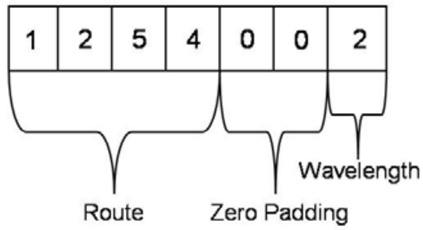

Figure 4-1: Solution encoding

2. Pad the node list with zeros to create a list of length N.

3. Append the assigned wavelength w to the end of the list in position $N+1$.

For example, the route $r = (1\ 2\ 5\ 4)$ with wavelength assignment $w = 2$ in a network with 6 nodes is encoded as [1 2 5 4 0 0 2] as shown in Figure 4-1. This type of encoding maintains a standard length of $N+1$ for each genotype in the population.

Crossover Operation

The crossover operator exchanges sub-routes and wavelengths between two individuals with the same source and destination nodes. Potential crossing sites are randomly selected from nodes that appear in both routes. The wavelength assignment and any extra zeros in the genotype are excluded from the pool of potential crossing sites. The source and destination nodes are also excluded. For a pair of distinct genotypes, g^1 and g^2, crossover is performed as follows:

1. Remove the wavelength assignments and extra zeros from g^1 and g^2 yielding corresponding routes $r^1$and r^2

2. Let N_c be the set of nodes that appear in both $r^1$and r^2.

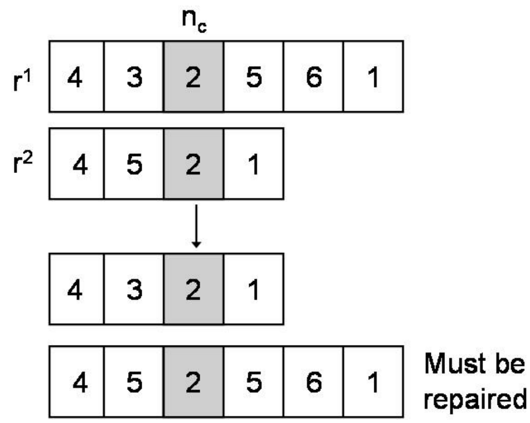

Figure 4-2: Crossover operation

3. Select a crossing node $n_c \in N_c$.

4. Exchange elements after n_c.

An example of the crossover operation is shown in Figure 4-2. Note that the crossover operator may create invalid routes. A repair procedure is described later in this chapter to correct an invalid route.

Mutation Operations

There are two mutations operations performed in our adaptive RWA algorithm. The first operation, *route mutation*, creates a new route with the original assigned wavelength. The second operation, *wavelength mutation*, generates a new solution with the same route assigned to a different wavelength. The procedures for route mutation and wavelength mutation for the genotype g are shown below.

Route Mutation

1. Remove the wavelength assignment and extra zeros from a randomly selected genotype g yielding corresponding route $r = (r_1\ r_2\ r_3\ ...\ r_K)$ where K is the number of nodes visited on route r, $r_1 = s$, and $r_K = d$.

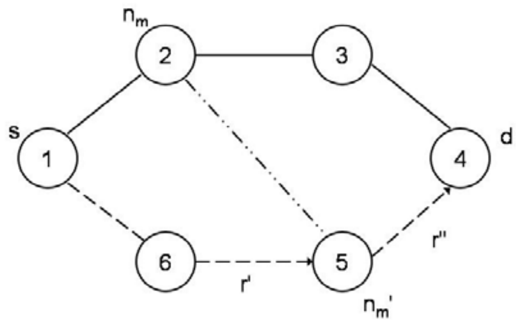

Figure 4-3: Route mutation operation

2. Randomly select a mutation node n_m from the set $\{r_2\ r_3\ ...\ r_{K-1}\}$.

3. Randomly select a node n_m' from the set of nodes adjacent to n_m.

4. Use Dijkstra's algorithm to generate the shortest path r' from r_1 to n_m' and the shortest path r' from n_m' to r_K .

5. Combine sub-routes r' and r' to create the mutated route $\hat{r} = r' + r'$.

Figure 4-3 illustrates route mutation.

Wavelength Mutation

1. Let w be the current wavelength assignment for genotype g. Randomly select a wavelength $w' \in \{1, 2, 3, ..., W\} - \{w\}$.

2. Replace w with w' as shown in Figure 4-4.

Repair Procedure

If the same node appears more than once in a route after the crossover or route mutation operations, a repair procedure is performed. The procedure shown in Figure 4-5 deletes extra copies of the duplicated node and any intermediate nodes between the duplicates.

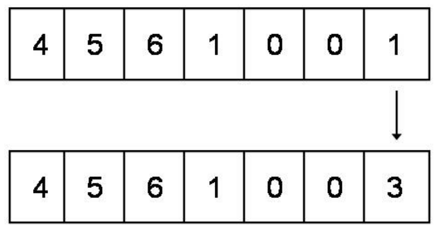

Figure 4-4: Wavelength mutation operation

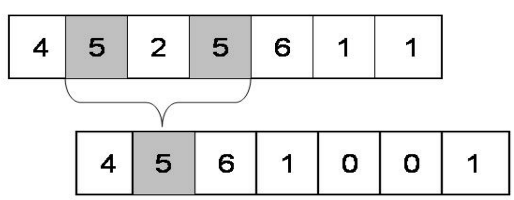

Figure 4-5: Repair procedure

Fitness Evaluation

Each individual in the population consists of a route r and an assigned wavelength w. The fitness of each individual is based on the availability of wavelength w on route r and the length of r. Let L^r be the set of physical links that determine route r. Each route considered during fitness evaluation is assigned a value a_r where

$$a_r = \begin{cases} 1, & \text{if wavelength } w \text{ is available on all links } l \in L^r, \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

The length of r is the cardinality of L^r, denoted by $l_r = |L^r|$ and the fitness value is computed by

$$f_r = \frac{a_r}{l_r}. \tag{4.2}$$

The value of a_r is divided by l_r so that a route with shorter length receives a larger fitness value f_r.

4.3 GA-based Adaptive RWA Algorithm

In a network with N nodes, L links, and W wavelengths per link, each source node s maintains its own routing table RT^s $(s = 1, 2, ..., N)$ that is initially empty. The routing tables have entries for destination, route, and assigned wavelength.

Dijkstra's shortest path algorithm and first-fit wavelength assignment are used to route the first connection request for each (s, d) pair. Uniformly distributed random numbers on the interval $[0, 1]$ are generated to determined if mutation or crossover operations will be performed. Route mutation is performed with probability p_{m1} while the probability of performing wavelength mutation is p_{m2}. The crossover operation is carried out with probability p_c. As additional solutions are generated by the genetic operators, they are placed in the appropriate routing tables. Once the first request for an (s,d) pair is routed using shortest path routing and assigned a wavelength, future requests are assigned the solution with the highest fitness from the routing table. Table 4.1 shows an example of a routing table for the simple network shown in Figure 4-6. The pseudo code for our proposed

GA-based adaptive RWA algorithm is shown in Algorithm 4.2.

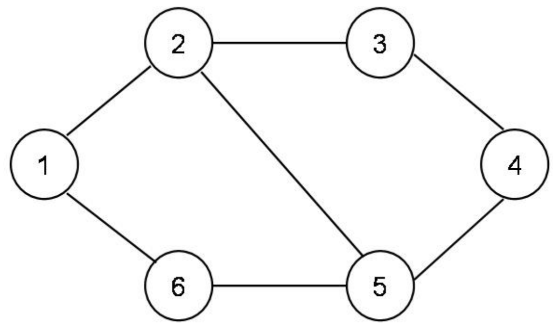

Figure 4-6: Simple network

Table 4.1: Sample routing table for GA-based algorithm

Destination	**Route**	**Wavelength**
2	(1 2)	2
3	(1 2 3)	1
4	(1 2 3 4)	2
4	(1 2 3 4)	3
5	(1 2 5)	2
5	(1 2 5)	1
5	(1 2 5)	3
6	(1 6)	2
6	(1 6)	3
6	(1 6)	1

Algorithm 4.2 GA-based Adaptive RWA

```
Initialize:
    RT^s = [empty table] for s = 1, ··· , N.
    S = L × W zero matrix.
    T = [empty table].
While (termination criterion not fulfilled)
    Wait for a request to arrive (connection or termination).
    If request is a connection request (s, d, h)
        Let R^sd be the set of routes in routing table RT^s to destination d.
        If R^sd is empty
            Use Dijkstra's algorithm to determine route r*.
            Use given wavelength assignment method to assign a wavelength w*.
            If there is no available wavelength on route r*
                Connection request is blocked.
            Else
                Route connection on route r* with wavelength w*.
                Let L* be the set of links that compose route r*.
                Update S to show that wavelength w* is busy on links in L*.
                Add new genotype [r*,w*] to RT^s.
                Update T by adding termination request (L*, w*, t + h).
            End
        Else
            Let G^sd be the set of solutions (genotypes) in routing table RT^s with
                destination d.
            For each solution g ∈ G^sd
                Perform fitness evaluation.
            End
            Use roulette wheel selection to select a solution g*=[ r*,w*] from G^sd.
            Route connection on route r* with wavelength w*.
            Update S to show that wavelength w* is busy on links in L*.
            Update T by adding termination request (L*, w*, t + h).
        End
    Else
        Termination request (L*, w*) arrives.
        Update S to show that wavelength w* is available on links in L*.
    End
```

Algorithm 2 (cont'd)

If random number < p_{m1}
    Perform wavelength mutation to a route in RT^s where s is the source
        node of the last connection request.
    Add genotype $[r, w]$ of new individual to RT^s if not already present.
End
If random number < p_{m2}
    Perform route mutation to a route in RT^s where s is the source
        node of the last connection request.
    Add genotype $[r, w]$ of new individual to RT^s if not already present.
End
If random number < p_c
    Perform crossover operation using two routes with the same destination
        in RT^s where s is the source node of the last connection request.
    Add genotype $[r, w]$ of new individual to RT^s if not already present.
End
End

4.4 Performance Analysis

We evaluated the performance of the GA-based adaptive RWA algorithm on the NSFnet network shown in Figure 3-1 and the 4x4 mesh-torus network illustrated in Figure 3-2. Each method is compared fixed-SP routing and alternate routing using the corresponding wavelength assignment policy. A dynamic traffic model in which connection requests arrive at each node according to a Poisson process with network wide arrival rate β is used for simulations. The connection holding time is exponentially distributed with mean $1/\mu$ resulting in an average load per (s, d) pair $\rho = \beta/N(N-1)\mu$. A node may be involved in multiple sessions and parallel sessions may occur between an (s,d) pair. Wavelength mutation and route mutation are performed with probabilities p_{m1} and p_{m2}, respectively. The probability of performing the crossover operation is p_c. Parameter tests were performed to determine the best values for p_{m1}, p_{m2}, and p_c. The test showed that $p_{m1} = 0.9$, $p_{m2} = 0.9$,and $p_c = 0.9$ yielded the best results. Since the routing table is initially empty, the high mutation and crossover probabilities allow potential solution to be generated faster. Solutions generated by crossover or mutation operations that already appear in a routing table are discarded.

4.4.1 NSFnet with W=4

Figures 4-7, 4-8, 4-9, and 4-10 show the performance of our GA-based algorithm on the NSFnet network with 4 wavelengths per link (W=4) using random, first-fit, least-used, and most-used wavelength assignment, respectively. In each case, our algorithm outperforms the fixed-SP and alternate routing methods. Table 4.2 shows the average blocking rate over all network loads for each algorithm. The average running time per connection for each approach is shown in Table 4.3. The number beside entries in Table 4.2 is the percentage by which the blocking rate increases for the corresponding RWA algorithm in comparison to the GA-based algorithm. The number beside entries in Table 4.3 is the percent decrease in running time when the specified RWA algorithm is used as opposed to the GA-base algorithm. Note that while the average blocking rate is decreased by using the GA-based algorithm, the time spent making a routing decision for each connection increases. A comparison of blocking rates of the GA-based algorithm for the various WAPs is illustrated in Figure 4-11. For network loads less than 0.6, there is no preferred WA strategy. However, the most-used WAP outperforms the other policies when the load is equal to 0.6 or greater.

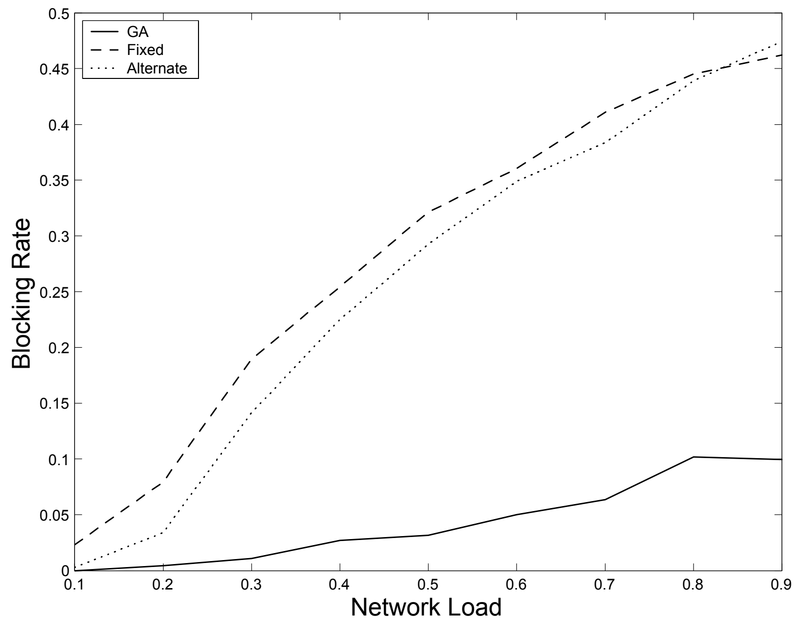

Figure 4-7: Blocking rates for GA-based algorithm using random wavelength assignment for NSFnet with W=4

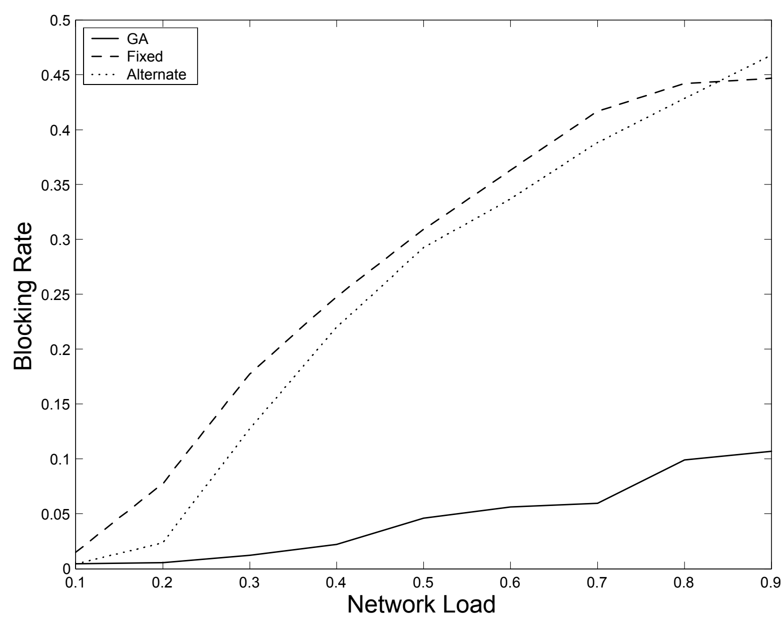

Figure 4-8: Blocking rates for GA-based algorithm using first-fit wavelength assignment for NSFnet with $W = 4$

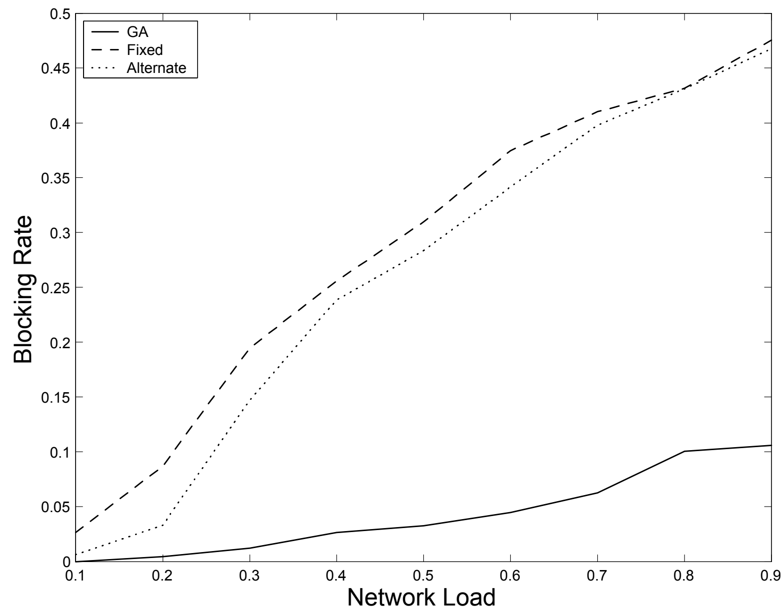

Figure 4-9: Blocking rates for GA-based algorithm using least-used wavelength assignment for NSFnet with W=4

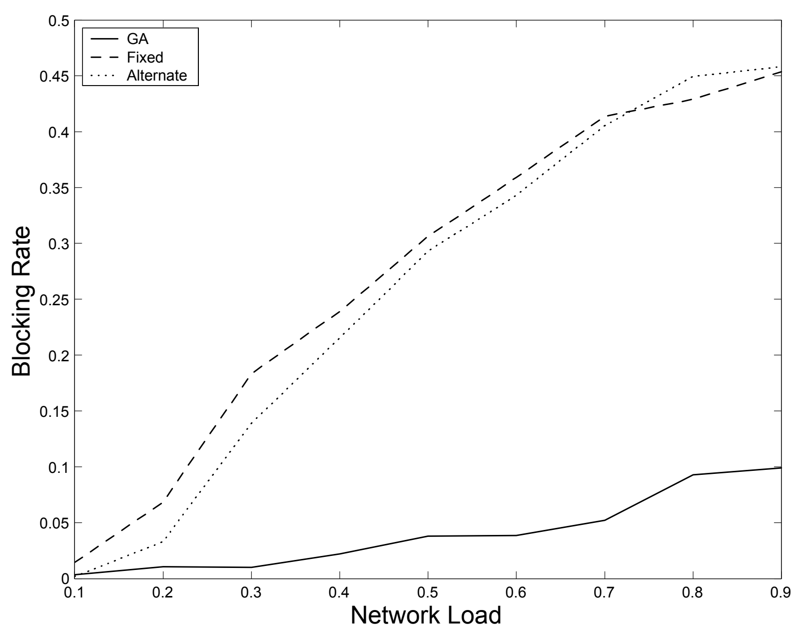

Figure 4-10: Blocking rates for GA-based algorithm using most-used wavelength assignment for NSFnet with W=4

Table 4.2: Average blocking rates for GA, fixed, and alternate routing algorithms for NSFnet with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	**%**	**ABR**	**%**	**ABR**	**%**	**ABR**	**%**
GA	0.0432		0.0457		0.0432		0.0407	
Fixed	0.2829	655	0.2773	607	0.2851	660	0.2742	673
Alternate	0.2602	602	0.2544	557	0.2607	603	0.2598	638

Table 4.3: Average running times for GA fixed, and alternate routing algorithms for NSFnet with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	**%**	**ART**	**%**	**ART**	**%**	**ART**	**%**
GA	0.0094		0.0094		0.0094		0.0095	
Fixed	0.0005	-2019	0.0004	-2226	0.0004	-2297	0.0004	-2132
Alternate	0.0009	-1074	0.0008	-1209	0.0008	-1148	0.0008	-1122

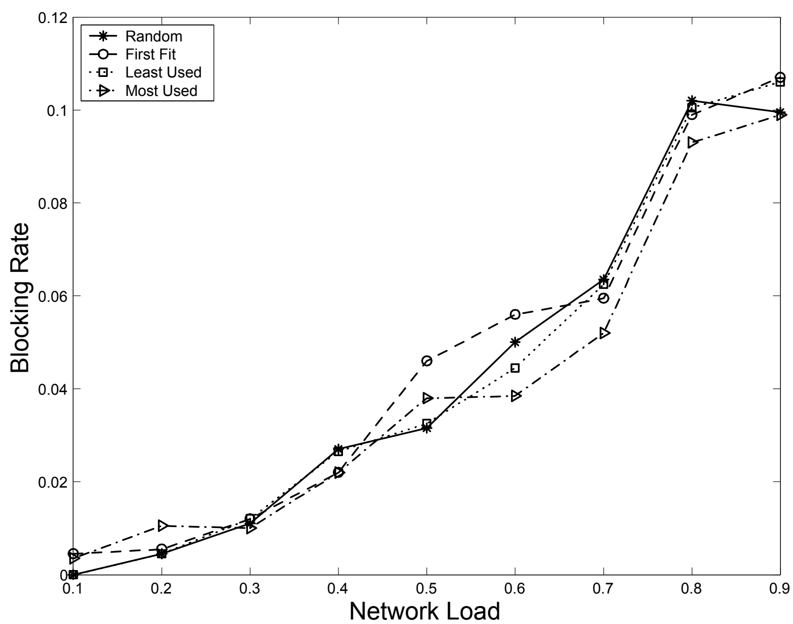

Figure 4-11: Comparison of blocking rates of all wavelength assignment policies for NSFnet with W=4

4.4.2 NSFnet with W=8

Figures 4-12, 4-13, 4-14, and 4-15 show the performance of our GA-based algorithm on the NSFnet network with 8 wavelengths per link ($W = 8$) using random, first-fit, least-used, and most-used wavelength assignment, respectively. Our algorithm outperforms the fixed-SP and alternate routing methods for network loads greater than or equal to 0.6. However the average blocking rate over all network loads for the GA-based algorithm is less than that of fixed and alternate routing as shown in Table 4.4. The average running time per connection for each approach is shown in Table 4.5. As was the case for NSFnet with $W = 4$, the average blocking rate is decreased by using the GA-based algorithm, while the time spent making a routing decision for each connection increases. However, the improvements in blocking rates are not as great. This result is expected since assigning routes and wavelengths is easier for a network with more wavelengths per link. A comparison of blocking rates of the GA-based algorithm for the various WAPs is illustrated in Figure 4-16. Surprisingly, the random WAP has the lowest blocking rate for the majority of the network loads tested.

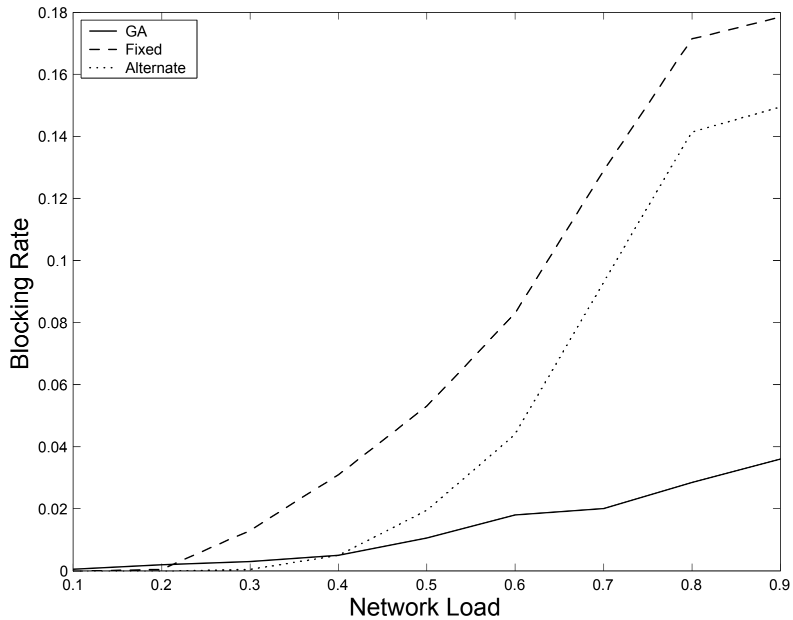

Figure 4-12: Blocking rates for GA-based algorithm using random wavelength assignment for NSFnet with W=8

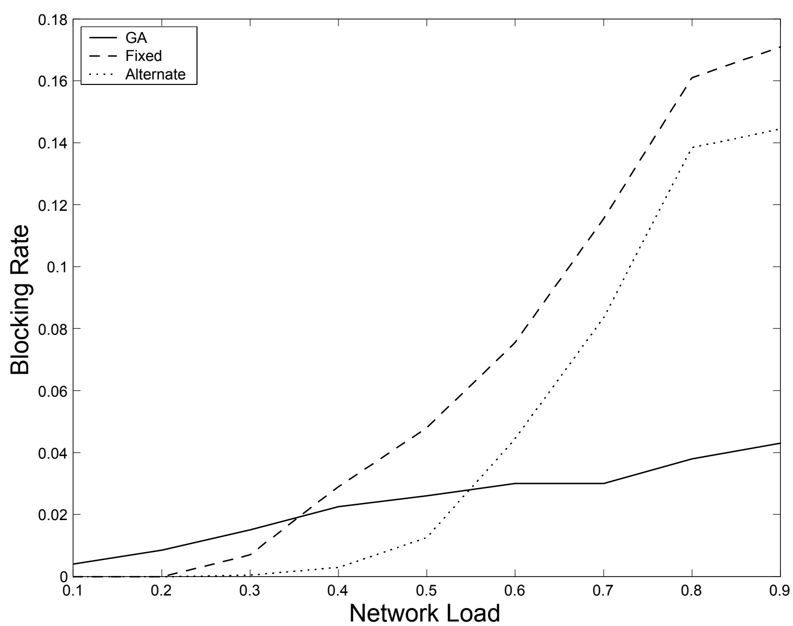

Figure 4-13: Blocking rates for GA-based algorithm using first-fit wavelength assignment for NSFnet with W=8

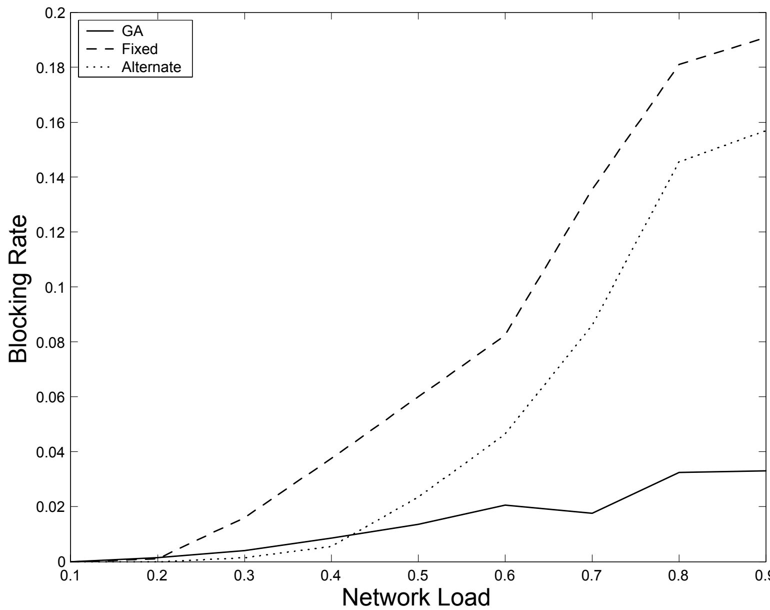

Figure 4-14: Blocking rates for GA-based algorithm using least-used wavelength assignment for NSFnet with W=8

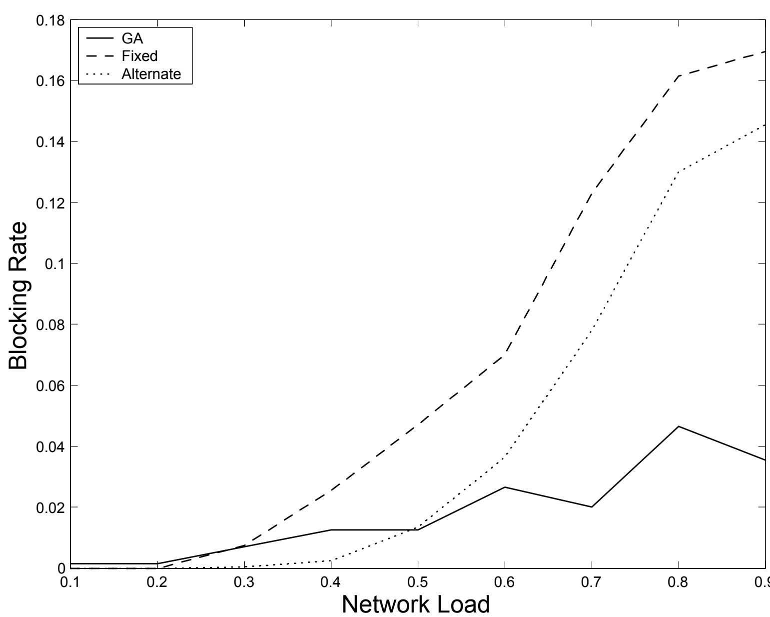

Figure 4-15: Blocking rates for GA-based algorithm using most-used wavelength assignment for NSFnet with W=8

Table 4.4: Average blocking rates for GA, fixed, and alternate routing algorithms for NSFnet with W=8

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	**%**	**ABR**	**%**	**ABR**	**%**	**ABR**	**%**
GA	0.0137		0.0241		0.0146		0.0182	
Fixed	0.0733	534	0.0674	280	0.0783	538	0.0671	369
Alternate	0.0503	367	0.0474	197	0.0517	355	0.0452	249

Table 4.5: Average running times for GA fixed, and alternate routing algorithms for NSFnet with W=8

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	**%**	**ART**	**%**	**ART**	**%**	**ART**	**%**
GA	0.0107		0.0105		0.0105		0.0108	
Fixed	0.0005	-2341	0.0004	-2418	0.0004	-2429	0.0005	-2201
Alternate	0.0006	-1714	0.0006	-1746	0.0006	-1822	0.0007	-1587

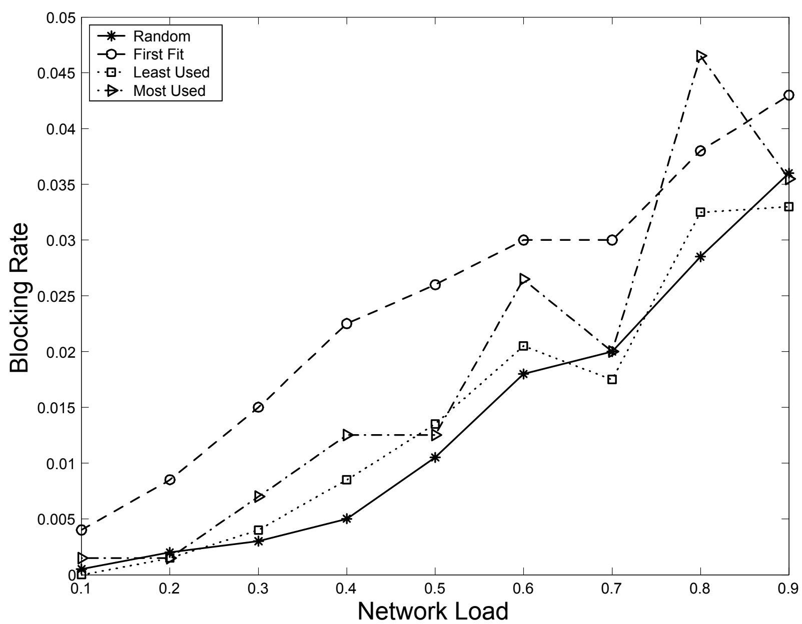

Figure 4-16: Comparison of blocking rates of all wavelength assignment policies for NSFnet with W=8

4.4.3 4×4 Mesh-Torus Network with W=4

Figures 4-17, 4-18, 4-19, and 4-20 show the performance of our algorithm on the 4×4 mesh-torus network with 4 wavelengths per link (W=4) using random, first-fit, least-used, and most-used wavelength assignment, respectively. In each case, our algorithm outperforms the fixed-SP and alternate routing methods. Table 4.6 shows the average blocking rate over all network loads for each algorithm. The average running time per connection for each approach is shown in Table 4.7. Note that while the average blocking rate is decreased by using the GA-based algorithm, the time spent making a routing decision for each connection increases.

A comparison of blocking rates of the GA-based algorithm for the various WAPs is illustrated in Figure 4-21. Although there is no preferred WA strategy that performs best for every load, the random WAP has the lowest average blocking rate over all network loads tested.

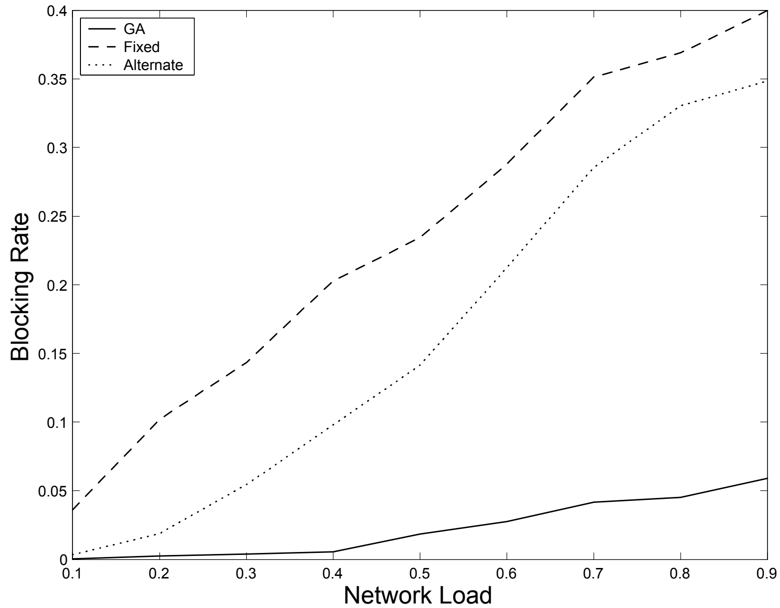

Figure 4-17: Blocking rates for GA-based algorithm using random wavelength assignment for the mesh-torus network with W=4

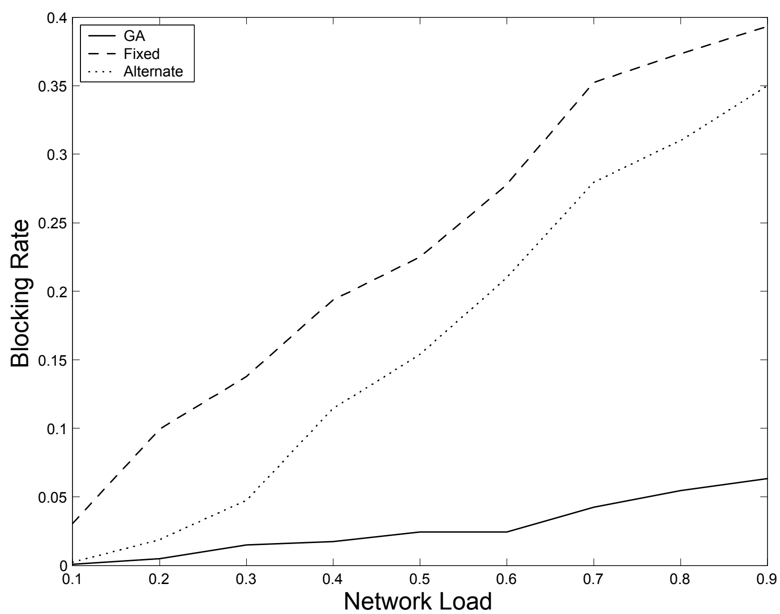

Figure 4-18: Blocking rates for GA-based algorithm using first-fit wavelength assignment for the mesh-torus network with W=4

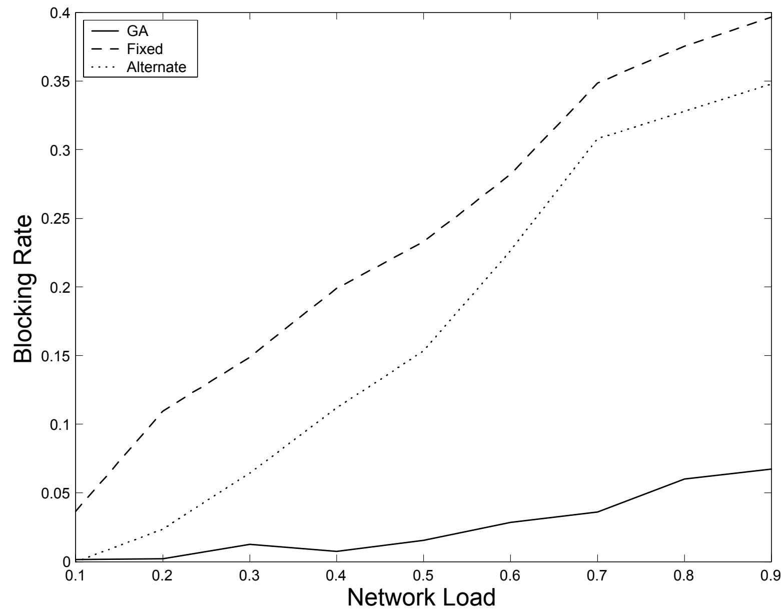

Figure 4-19: Blocking rates for GA-based algorithm using least-used wavelength assignment for the mesh-torus network with W=4

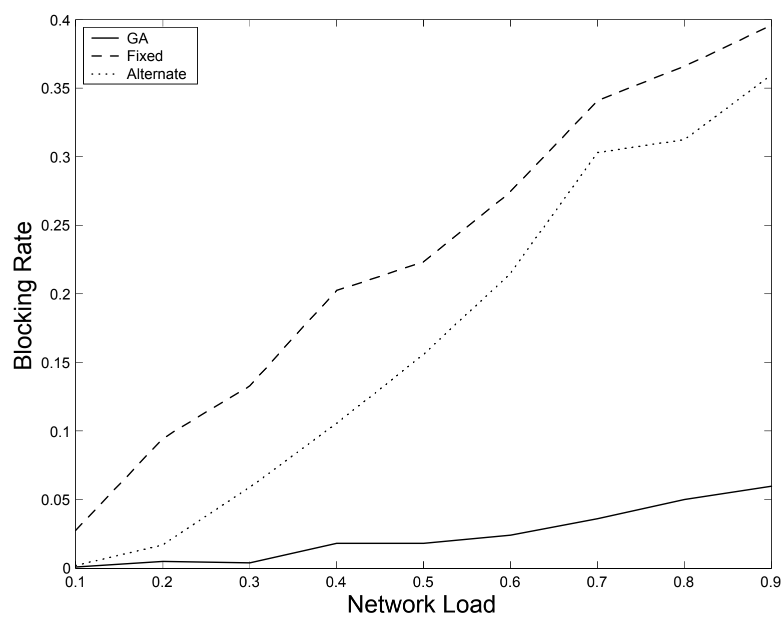

Figure 4-20: Blocking rates for GA-based algorithm using most-used wavelength assignment for the mesh-torus network with W=4

Table 4.6: Average blocking rates for GA, fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	**%**	**ABR**	**%**	**ABR**	**%**	**ABR**	**%**
GA	0.0227		0.0276		0.0257		0.0239	
Fixed	0.2364	1043	0.2316	841	0.2366	922	0.2287	955
Alternate	0.1659	732	0.1652	600	0.1738	677	0.1699	710

Table 4.7: Average running times for GA fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	**%**	**ART**	**%**	**ART**	**%**	**ART**	**%**
GA	0.0146		0.0146		0.0145		0.0146	
Fixed	0.0004	-3366	0.0004	-3549	0.0004	-3443	0.0004	-3290
Alternate	0.0008	-1903	0.0007	-1989	0.0008	-1843	0.0008	-1803

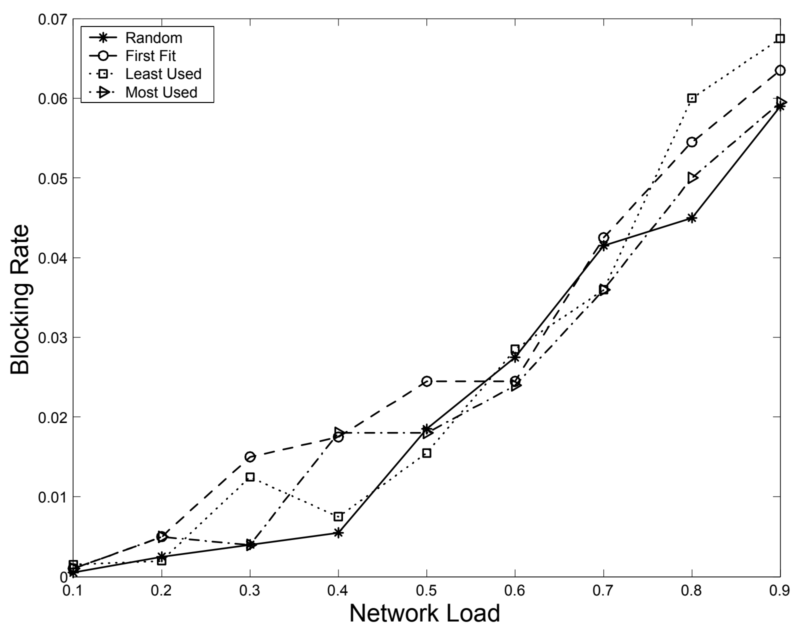

Figure 4-21: Comparison of blocking rates of all wavelength assignment policies for mesh-torus network with W=4

4.4.4 4×4 Mesh-Torus Network with W=8

Figure 4-22 show the performance of our algorithm using random wavelength assignment. The GA-based algorithm outperforms the fixed-SP and alternate routing methods for high network loads. The alternate routing method performs the same or better for loads less than 0.6. The same is true for the algorithms using first-fit and most-used wavelength assignment as shown in Figures 4-23 and 4-25. From the results for least-used wavelength assignment shown in Figure 4-24, we see that the GA-based algorithm outperforms fixed-SP and alternate routing for loads greater than 0.5.

Table 4.8 shows the average blocking rate over all network loads for each algorithm. The average running time per connection for each approach is shown in Table 4.9. Note that the average blocking rate for our algorithm is lower than that of the fixed routing strategy for all WAPs. The average blocking rate for our algorithm is also lower than that of the alternate routing strategy for all WAPs except the first-fit approach.

A comparison of blocking rates of the GA-based algorithm for the various WAPs is illustrated in Figure 4-26. Similar to the results for the mesh-torus network with W=4, there is no preferred WA strategy that performs best for every load. The random WAP has the lowest average blocking rate over all network loads tested.

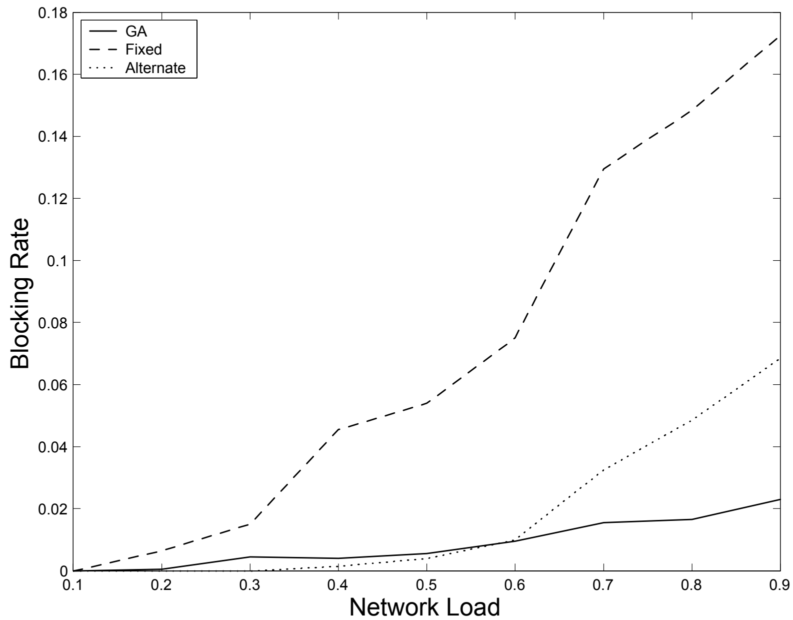

Figure 4-22: Blocking rates for GA-based algorithm using random wavelength assignment for the mesh-torus network with W=8

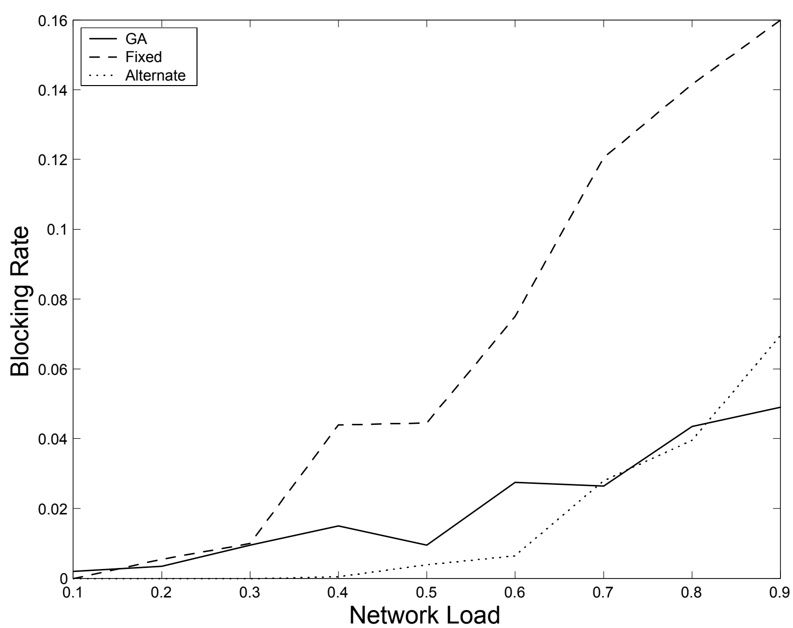

Figure 4-23: Blocking rates for GA-based algorithm using first-fit wavelength assignment for the mesh-torus network with W=8

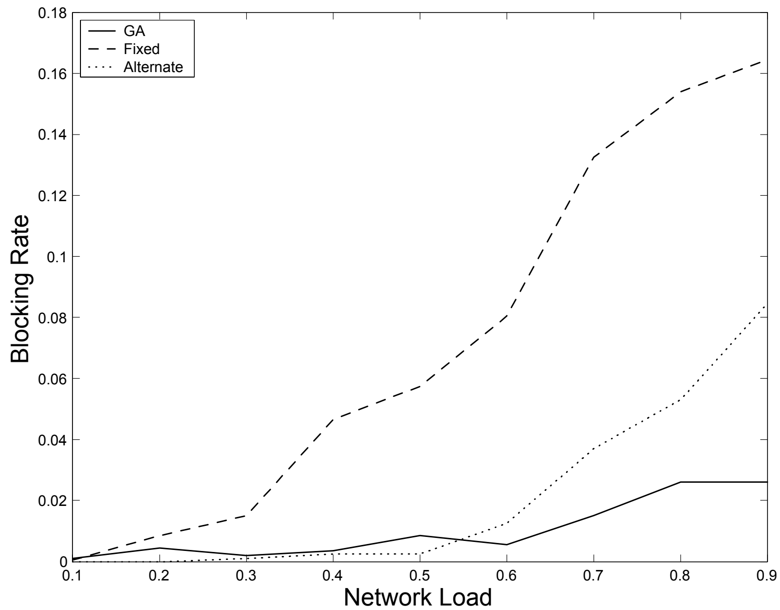

Figure 4-24: Blocking rates for GA-based algorithm using least-used wavelength assignment for the mesh-torus network with W=8

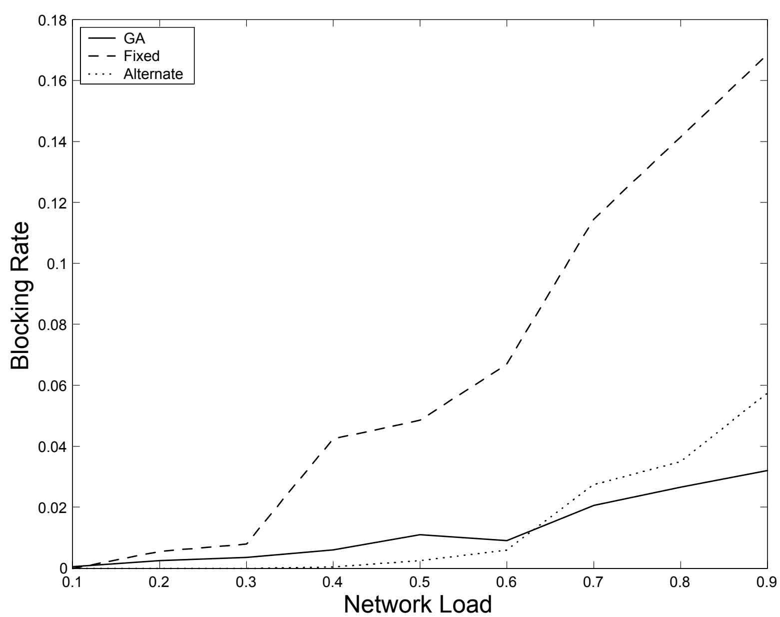

Figure 4-25: Blocking rates for GA-based algorithm using most-used wavelength assignment for the mesh-torus network with W=8

Table 4.8: Average blocking rates for GA, fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=8

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	**%**	**ABR**	**%**	**ABR**	**%**	**ABR**	**%**
GA	0.0088		0.0207		0.0102		0.0124	
Fixed	0.0718	818	0.0668	323	0.0733	717	0.0662	535
Alternate	0.0183	209	0.0164	-126	0.0214	210	0.0143	116

Table 4.9: Average running rimes for GA, fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=8

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	**%**	**ART**	**%**	**ART**	**%**	**ART**	**%**
GA	0.0158		0.0157		0.0157		0.0158	
Fixed	0.0004	-3550	0.0004	-3613	0.0004	-3622	0.0005	-3310
Alternate	0.0006	-2630	0.0006	-2763	0.0006	-2770	0.0006	-2541

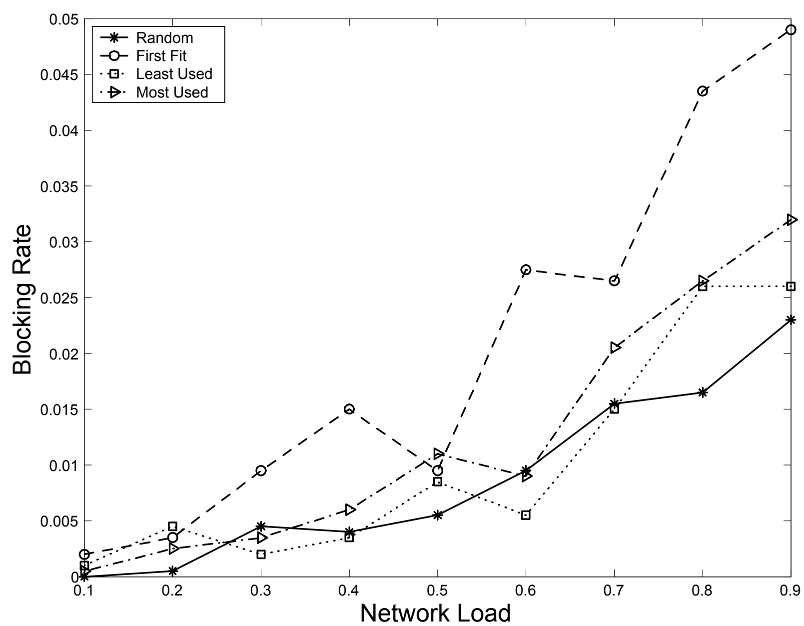

Figure 4-26: Comparison of blocking rates of all wavelength assignment policies for mesh-torus network with W=8

Chapter 5

A Fuzzy Control Approach

The RWA problem in wavelength-routed optical networks consists of setting up lightpaths for each connection request by finding a route from source to destination and assigning a valid wavelength to the route. In this chapter, we propose an adaptive fuzzy-controlled approach to DLE that uses convenient linguistic variables and fuzzy if-then rules.

5.1 Fuzzy Logic

In classic set theory, a *set* is defined as a collection of objects or elements. The fundamental relation from which many others are derived is membership. Each element can either be a member of a set A ($x \in A$) or not ($x \notin A$). One way of describing a set is by its characteristic function μ_A. For a set A defined on domain X, μ_A is defined as

$$\mu_A : X \to \{0, 1\} \qquad \mu_A(x) = \begin{cases} 1, & \text{if } x \in A, \\ 0, & \text{if } x \notin A. \end{cases} \tag{5.1}$$

In fuzzy set theory, ordinary sets are called *crisp sets*. For a *fuzzy set* $\tilde{F}$, it is not required that either $x \in \tilde{F}$ or $x \notin \tilde{F}$. The characteristic function of a fuzzy set $\tilde{F}$ is generalized to a membership function $\mu_{\tilde{F}}$ where each object x in the domain X is assigned a membership value $\mu_{\tilde{F}}(x)$ between 0 and 1. Therefore, a fuzzy set $\tilde{F}$ is completely determined by the set of order pairs

$$\tilde{F} = \left\{ (x,\ \mu_{\tilde{F}}(x)) \mid x \in X \right\}. \tag{5.2}$$

There are numerous operations that can be performed on fuzzy sets as with crisp sets. These include union, intersection, complement, and difference. The standard union and intersection operations on fuzzy sets Ã and B̃ are defined as

$$\mu_{\tilde{A}\cap\tilde{B}}(x) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)), \tag{5.3}$$

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)). \tag{5.4}$$

Since the 1960s, fuzzy logic has been applied in several areas such as graphs, expert systems, data analysis, and mathematical programming. One of the largest areas of application is fuzzy control.

5.2 The Fuzzy Controller

Three essential components of a fuzzy controller are the knowledge base, inference engine, and defuzzification method.

Knowledge Base

The basic concepts used in fuzzy control are *linguistic variables* and *fuzzy if-then (*or *inference) rules.* A linguistic variable is a variable whose values are words or sentences in a natural or synthetic language [61]. For example, '*Temperature*' is a linguistic variable whose values may be '*cold*', '*warm*', or '*hot*'. Each linguistic value is characterized by a membership function. Linguistic variables are used to build fuzzy if-then rules that link input (or state) variables and output (or control) variables. For example, the statement

if x is $\tilde{X}$ **then** y is $\tilde{Y}$

is a fuzzy if-then rule (or fuzzy implication) where x is a linguistic input variable defined on the universal set P, y is a linguistic output variable defined on the universal set Q, and $\tilde{X}$

Table 5.1: Generalized modus ponens

Premise:	x is $\widetilde{X}'$
Implication:	**if** x is $\tilde{X}$ **then** y is $\tilde{Y}$
Conclusion:	y is $\widetilde{Y}'$

and $\tilde{Y}$ are linguistic values associated with the respective linguistic variables. The *if* clause of the rule is known as the antecedent and the *then* clause is called the consequent.

The knowledge base of a fuzzy controller consists of the linguistic variables and the corresponding linguistic values and membership functions for the input and output variables. In addition, the knowledge base includes the set of if-then rules that are used to control the system.

Inference Engine

The inference engine calculates the value of the output variable for each rule and for the whole set of rules. An important inference rule in fuzzy logic is the generalized modus ponens shown in Table 5.1 where $\tilde{X}$, $\tilde{Y}$, $\widetilde{X}'$ and $\widetilde{Y}'$ are fuzzy sets. However, a fuzzy controller may have crisp input values. These crisp values are *fuzzified* or converted into the fuzzy set $\widetilde{X}'$. This conversion makes the input values compatible with the rule antecedents. For a crisp input value x^*, the membership function after fuzzification $\mu_{\widetilde{X}'}(x)$ is defined as

$$\mu_{\widetilde{X}'}(x) = \begin{cases} 1, & \text{if } x = x^*, \\ 0, & \text{otherwise.} \end{cases} \qquad \forall x \tag{5.5}$$

The fuzzy implication," **if** x is $\tilde{X}$ **then** y is $\tilde{Y}$ ", is a fuzzy relation $\tilde{X} \rightarrow \tilde{Y}$ that can be expressed by a fuzzy set $\tilde{R}$ defined on the space $P \times Q$ with membership function $\mu_{\widetilde{R}}(x, y)$. For each pair (x, y), the value of $\mu_{\widetilde{R}}(x, y)$ is determined by $\mu_{\widetilde{X}}(x)$ and $\mu_{\widetilde{Y}}(y)$. There are many proposed implication operators used to compute $\mu_{\widetilde{R}}(x, y)$ [15, 66]. In our research, we will use *Mamdani's implication rule* because it is relatively easy to implement. Mamdani's rule defines

$$\mu_{\widetilde{R}}(x, y) = \min(\mu_{\widetilde{X}}(x), \mu_{\widetilde{Y}}(y)). \tag{5.6}$$

The conclusion $\widetilde{Y}'$ in Table 5.1 can be expressed as the composition of the premise," x is

$\widetilde{X}'$ ", and the implication, " **if** x is $\tilde{X}$ **then** y is $\tilde{Y}$ ". The *compositional rule of inference* proposed by Zadeh [60] computes the membership of $\tilde{Y}'$ as

$$\mu_{\widetilde{Y}'}(y) = \max_x \left(\min\left(\mu_{\widetilde{X}'}(x), \mu_{\widetilde{R}}(x, y)\right)\right). \tag{5.7}$$

If Mamdani's implication rule is used to compute $\mu_{\widetilde{R}}(x, y)$ and the input is fuzzified using (5.5), equation (5.7) reduces to

$$\mu_{\widetilde{Y}'}(y) = \min\left(\mu_{\widetilde{X}}(x^*), \mu_{\widetilde{Y}}(y)\right). \tag{5.8}$$

Fuzzy controllers usually have many if-then rules with several input variables. For a controller with M inputs and 1 output variable, the fuzzy inferences rules have the form

if x_1 is $\tilde{X}_1$ and x_2 is $\tilde{X}_2$ and ... and x_M is $\tilde{X}_M$ **then** y is $\tilde{Y}$

where x_i are linguistic state variables, $\tilde{X}_i$ are linguistic values of x_i (for $i = 1, 2, ..., M$), and y is a linguistic control variable with a corresponding linguistic value $\tilde{Y}$. In this case, the antecedent for each fuzzy rule is the intersection of the M fuzzy input variables with membership function

$$\mu_{\tilde{X}_1 \cap \tilde{X}_2 \cap \cdots \cap \tilde{X}_M}(x_1, x_2, ..., x_M) = \min(\mu_{\tilde{X}_1}(x_1), \mu_{\tilde{X}_2}(x_2), ..., \mu_{\tilde{X}_M}(x_M)). \tag{5.9}$$

Using equation (5.9) in conjunction with equation (5.6), the compositional rule of inference yields

$$\mu_{\widetilde{Y}'}(y) = \min\Big(\min(\mu_{\tilde{X}_1}(x_1^*), \mu_{\tilde{X}_2}(x_2^*), ..., \mu_{\tilde{X}_M}(x_M^*)), \mu_{\widetilde{Y}}(y)\Big) \tag{5.10}$$

where $(x_1^*, x_2^*, ..., x_M^*)$ is the crisp input to the controller.

If the knowledge base contains R rules, the output of each rule $\tilde{Y}'_r$ for $r = 1, ..., R$ is calculated as

$$\mu_{\tilde{Y}'_r}(y) = \min\Big(\min(\mu_{\tilde{X}_{r1}}(x_1^*), \mu_{\tilde{X}_{r2}}(x_2^*), ..., \mu_{\tilde{X}_{rM}}(x_M^*)), \mu_{\tilde{Y}_r}(y)\Big) \tag{5.11}$$

where $\tilde{X}_{r1}, \tilde{X}_{r2}, ..., \tilde{X}_{rM}$ are the linguistic values of input variables $x_1, x_2, ..., x_M$, respectively for rule r. We must then aggregate the output of each rule $\tilde{Y'}_r$ to express the output of the complete set of rules. The fuzzy union operator is normally used for this process resulting in one fuzzy set $\widetilde{Y'}_{all}$. The membership function of $\widetilde{Y'}_{all}$ is given by

$$\mu_{\widetilde{Y'}_{all}}(y) = \max(\mu_{\tilde{Y'}_1}(y), \mu_{\tilde{Y'}_2}(y), ..., \mu_{\tilde{Y'}_R}(y)). \tag{5.12}$$

Defuzzification

Defuzzification converts the fuzzy output of the inference engine into a crisp value. There are several proposed defuzzification methods . The most used methods are *center of area*, *center of sums* and *mean of maxima*. We have chosen the center of area (COA) method for our fuzzy controller because it is often used in conjunction with Mamdani's rule of implication. The COA or Center of Gravity (COG) method determines the center of area below the aggregated membership function $\widetilde{Y'}_{all}$. The defuzzified output value y^* is computed as follows:

$$y^* = \frac{\int_y y \cdot \mu_{\widetilde{Y'}_{all}}(y)dy}{\int_y \mu_{\widetilde{Y'}_{all}}(y)dy} \tag{5.13}$$

5.3 Fuzzy Controller Design

For the RWA problem, the quality of a potential route is determined by the length of the route (i.e. number of links) and the congestion along the route. Path congestion is determined by the number of available (idle) wavelengths on the route. We seek to balance the traffic load among links in the network so it is desirable to select routes with a larger number of idle wavelengths. In addition, shorter routes are preferred over longer routes because they use fewer network resources. In other words, the preferred route is one with a "short" route length and "light" congestion. Thus, the two fuzzy input variables for the fuzzy controller are `Route_Length` and `Congestion`. These variables may be represented as continuous or discrete fuzzy variables since each input value is an integer. Figures 5-1 and 5-2 show the membership functions of `Route_Length` used for NSFnet and the 4×4 mesh-torus network. The membership functions for the input variable `Congestion` are shown in Figure 5-3 for a network with $W = 4$ and Figure 5-4 for a network with $W = 8$. The output variable of the

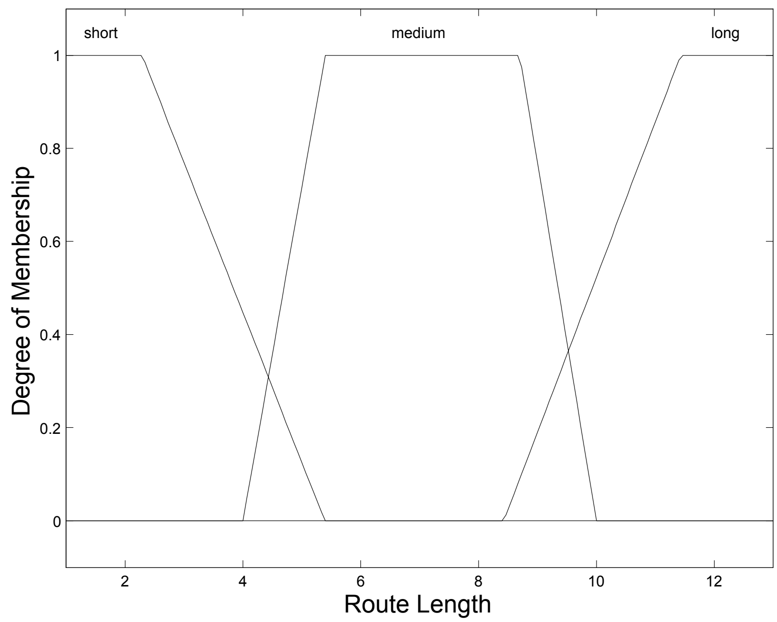

Figure 5-1: Fuzzy input variable ROUTE LENGTH for NSFnet

fuzzy controller is a rating for the path. The fuzzy output variable `Rating` is expressed as a continuous fuzzy variable. The membership functions for `Rating` are shown in Figure 5-5.

Both fuzzy input variables, `Route_Length` and `Congestion,` have three fuzzy values resulting in nine different potential combinations of input values. Therefore, our fuzzy controller contains nine fuzzy if-then rules shown in Table 5.2. The consequent of each rule was chosen to reflect the desired route and wavelength preferences.

The fuzzy controller employs Mamdani's implication rule, Zadeh's compositional rule of

Table 5.2: Fuzzy control rules

If Route Length is short and Congestion is light the Rating is excellent
If Route Length is short and Congestion is medium the Rating is good
If Route Length is short and Congestion is heavy the Rating is poor
If Route Length is medium and Congestion is light the Rating is good
If Route Length is medium and Congestion is medium the Rating is average
If Route Length is medium and Congestion is heavy the Rating is poor
If Route Length is large and Congestion is light the Rating is average
If Route Length is large and Congestion is medium the Rating is poor
If Route Length is large and Congestion is heavy the Rating is poor

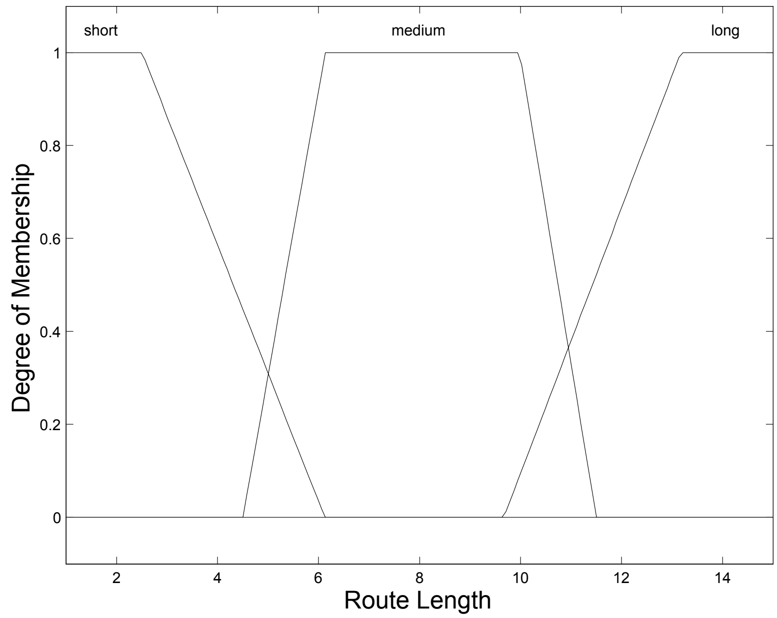

Figure 5-2: Fuzzy input variable ROUTE LENGTH for 4x4 mesh-torus

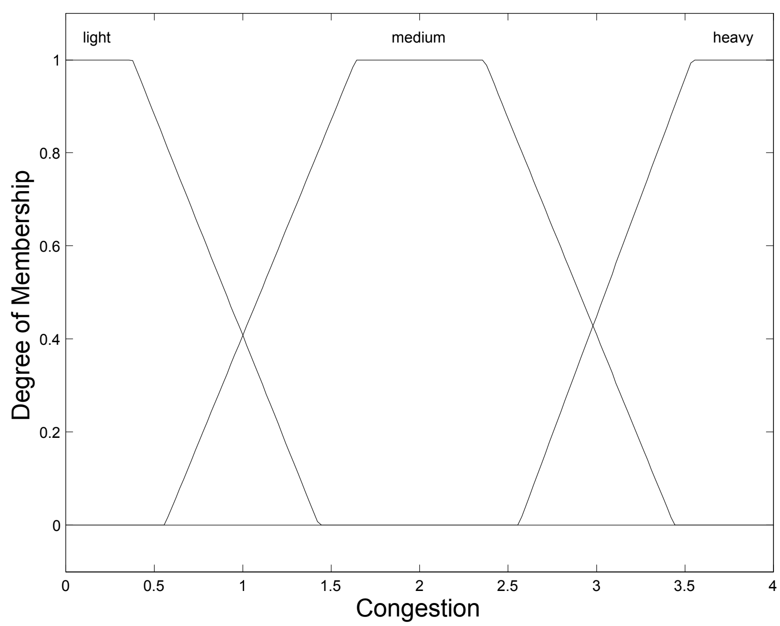

Figure 5-3: Fuzzy input variable CONGESTION for W=4

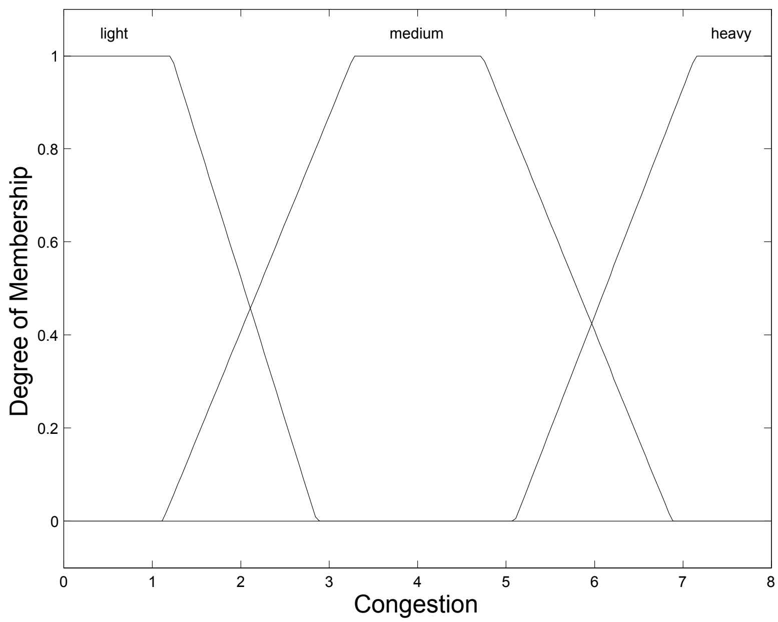

Figure 5-4: Fuzzy input variable CONGESTION for W=8

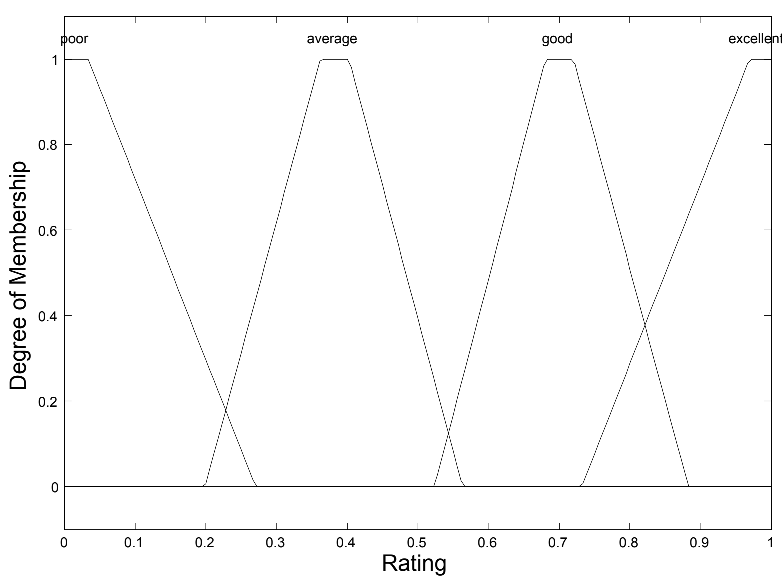

Figure 5-5: Fuzzy output variable - Rating

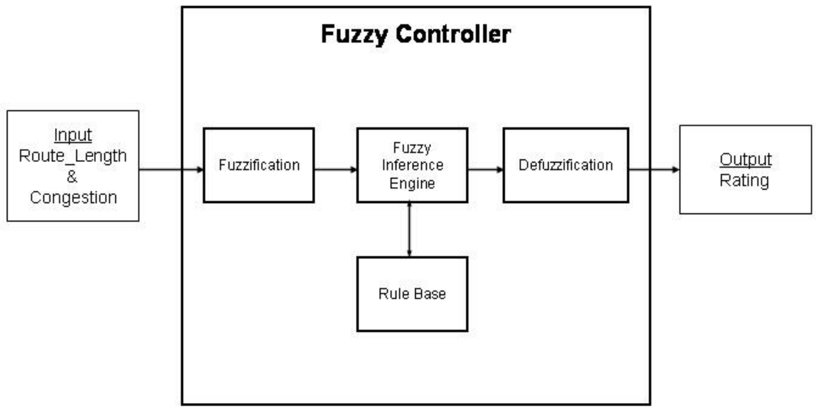

Figure 5-6: Fuzzy controller

inference, and COA defuzzification as described in equations (5.6), (5.7), and (5.13) of the previous section. A diagram of the proposed fuzzy controller is shown in Figure 5-6.

5.4 FC-based Adaptive RWA Algorithm

Our fuzzy-controlled adaptive RWA algorithm is based on a set of fuzzy if-then rules that guides the selection of a physical route and wavelength for each connection request based on the current state of the network. In a network with N nodes, L links, and W wavelengths per link, each source node s maintains its own routing table RT^s ($s = 1, 2, ..., N$) that contains a list of all paths from the source node s to all destination nodes $d \neq s$. For larger networks, the size of the routing table can be reduced by limiting the number of alternate routes for each destination. For our simulations, we limited the routing table to 5 routes per (s, d). Table 5.3 shows an example of a routing table for the simple network shown in Figure 4-6.

The network maintains a $L \times W$ link-wavelength status matrix S where

$$S_{lw} = \begin{cases} 1, & \text{if wavelength } w \text{ is in use on link } l, \\ 0, & \text{otherwise.} \end{cases} \tag{5.14}$$

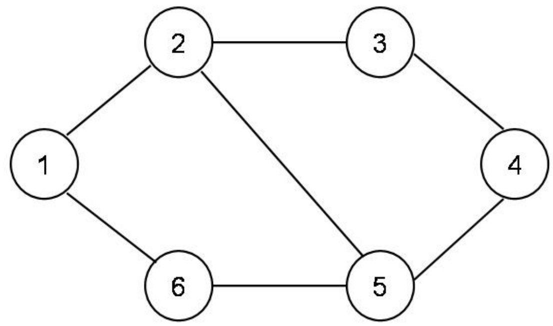

Figure 5-7: Simple network

Table 5.3: Sample routing table for fuzzy-controlled algorithm

Destination	**Route**
2	(1 2) (1 6 5 2) (1 6 5 4 3 2)
3	(1 2 3) (1 6 5 2 3) (1 6 5 4 3) (1 2 5 4 3)
4	(1 2 3 4) (1 6 5 4) (1 2 5 4) (1 6 5 2 3 4)
5	(1 6 5) (1 2 5) (1 2 3 4 5)
6	(1 6) (1 2 5 6) (1 2 3 4 5 6)

This matrix is used by the fuzzy controller to determine the number of available wavelengths on a route.

There are two types of requests used in the algorithm. Connection requests arrive at individual nodes and contain the source node s, destination node d, and holding time h for the connection. Termination requests are setup by each node once a lightpath has been established. These requests include the links that compose the lightpath, the assigned wavelength, and the time at which the lightpath is to be disconnected.

Upon arrival of a connection request, the set of routes from s to d, R^{sd}, is determined from routing table RT^s. For each route $r_i \in R^{sd}$ $(i = 1, 2, ..., |R^{sd}|)$, the links that compose the route and the number of available wavelengths along the route is found. This information is used as crisp input to the fuzzy controller. Once the controller provides a rating for each route in R^{sd}, the route with the highest rating is selected for the connection and a wavelength is assigned using the given wavelength assignment policy. A termination request is then setup to dismantle the lightpath on the specified links and wavelength after the holding time h has elapsed. The pseudo code for the fuzzy-controlled adaptive RWA algorithm is shown in Algorithm 5.1.

5.5 Performance Analysis

We evaluated the performance of the FC-based RWA algorithm on the NSFnet network shown in Figure 3-1 and the 4x4 mesh-torus network illustrated in Figure 3-2. A dynamic traffic model in which connection requests arrive at each node according to a Poisson process with network wide arrival rate β is used for simulations. The connection holding time is exponentially distributed with mean $1/\mu$ resulting in an average load per (s, d) pair $\rho = \beta/N(N-1)\mu$. A node may be involved in multiple sessions and parallel sessions may occur between an (s,d) pair.

Algorithm 5.1 FC-based Dyanmic RWA

Initialize:
    RT^s = [empty table] for $s = 1, \cdots, N$.
    $S = L \times W$ zero matrix.
    T =[empty table].
While (termination criterion not fulfilled)
    Wait for a request to arrive (connection or termination).
    If request is a connection request (s, d, h)
        Let R^{sd} be the set of routes in routing table RT^s to destination d.
        For each route $r_i \in R^{sd}, i = 1, \cdots, |R^{sd}|$
            Let L^i be the set of links that compose route r_i.
            Let $RouteLength^* = |L^i|$.
            Let W^i be the set of available wavelengths on route r_i in ascending order.
            Let $Congestion^* = |W^i|$.
            Invokfuzzy controller
                For each fuzzy rule
                    Fuzzify $RouteLength^*$ and $Congestion^*$ according to
                        Equation 5.5.
                End
                For each fuzzy rule
                    Calculate fuzzy output according to Equation 5.11.
                End
                Aggregate the fuzzy outputs according to Equation 5.12.
                Defuzzify to yield a crisp rating according to Equation 5.13.
                Let $Rating_i$ be the output of the fuzzy controller for route r_i.
            Exit fuzzy controller.
        End
        Let i^* be the index of the route with the highest rating.
        If W^{i^*} is empty
            Connection request is blocked.
        Else
            Set $L^* = L^{i^*}$
            Select $w^* \in W^{i^*}$ according to wavelength assignment policy
            Route request on route r_{i^*} and wavelength w^*.
            Update S to show that wavelength w^* is busy on links in L^*
            Update T by adding termination request $(L^*, w^*, t + h)$
        End

Algorithm 5.1 (cont'd)

Else
    If request is a termination request (L^*, w^*)
        Update S to show that wavelength w^* is available on links in L^*
    End
End
End

5.5.1 NSFnet with W=4

Figures 5-8, 5-9, 5-10, and 5-11 show the performance of our algorithm on the NSFnet network with 4 wavelengths per link (W=4) using random, first-fit, least-used, and most-used wavelength assignment, respectively. In each case, our algorithm dominates over the fixed-SP and alternate routing methods. Table 5.4 shows the average blocking rate over all network loads for each algorithm. The average running time per connection for each approach is shown in Table 5.5. The number beside entries in Table 5.4 is the percentage by which the blocking rate increases for the corresponding RWA algorithm in comparison to the GA-based algorithm. The number beside entries in Table 5.5 is the percent decrease in running time when the specified RWA algorithm is used as opposed to the GA-base algorithm. While the average blocking rate is decreased by using the FC-based algorithm, the time spent making a routing decision for each connection increases. However, the improvement in blocking rate outweighs the increase in time. For the most-used wavelength assignment policy, Table 5.5 shows that the alternate routing method yields an average running time that is 1057% shorter than that of the FC-based approach. However, the FC-based method improves the average blocking rate by over 11,000%.

A comparison of blocking rates of the FC-based algorithm for the various WAPs is illustrated in Figure 5-12. The most-used WAP demonstrates the best performance over all network loads tested It also improves upon the average blocking rate of the alternate routing method by over 11,000%.

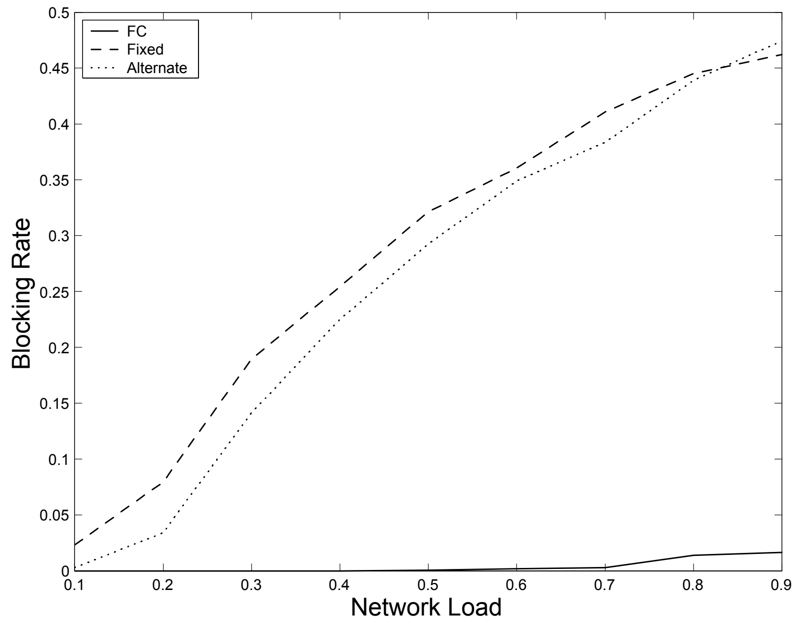

Figure 5-8: Blocking rates for FC-based algorithm using random wavelength assignment for NSFnet with W=4

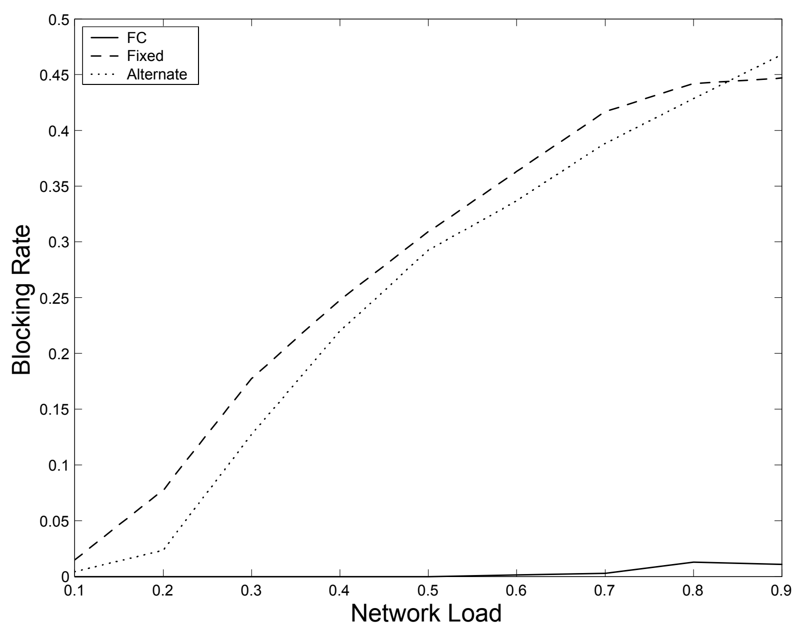

Figure 5-9: Blocking rates for FC-based algorithm using first-fit wavelength assignment for NSFnet with W=4

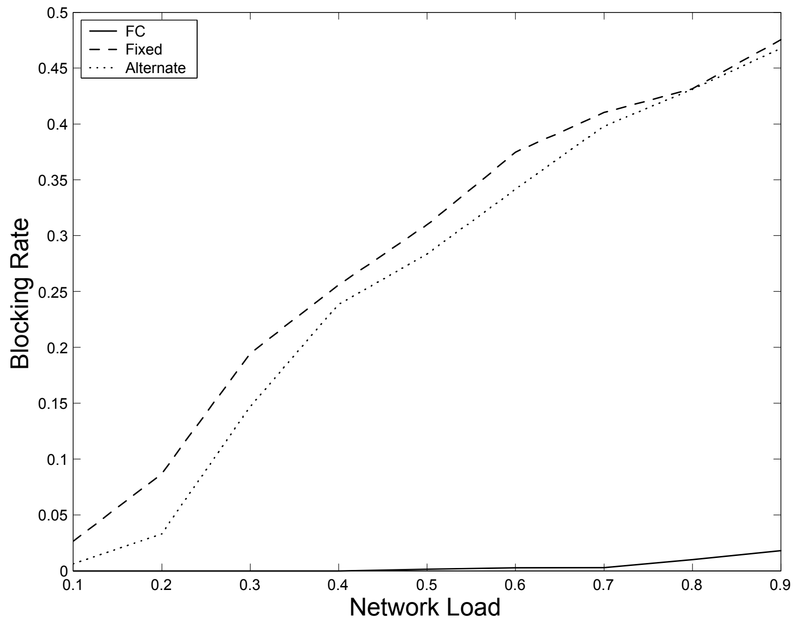

Figure 5-10: Blocking rates for FC-based algorithm using least-used wavelength assignment for NSFnet with W=4

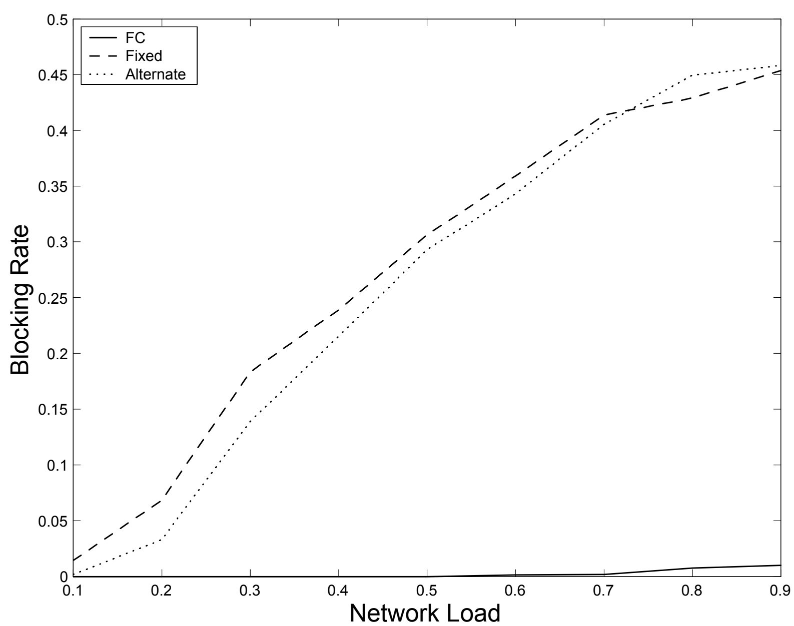

Figure 5-11: Blocking rates for FC-based algorithm using most-used wavelength assignment for NSFnet with W=4

Table 5.4: Average blocking rates for FC, fixed, and alternate routing algorithms for NSFnet with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	**%**	**ABR**	**%**	**ABR**	**%**	**ABR**	**%**
FC	0.0040		0.0032		0.0039		0.0023	
Fixed	0.2829	7074	0.2773	8758	0.2851	7330	0.2742	11750
Alternate	0.2602	6506	0.2544	8035	0.2607	6704	0.2598	11136

Table 5.5: Average running times for FC, fixed, and alternate routing algorithms for NSFnet with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	**%**	**ART**	**%**	**ART**	**%**	**ART**	**%**
FC	0.0087		0.0087		0.0089		0.0089	
Fixed	0.0005	-1872	0.0004	-2056	0.0004	-2169	0.0004	-2007
Alternate	0.0009	-995	0.0008	-1116	0.0008	-1085	0.0008	-1057

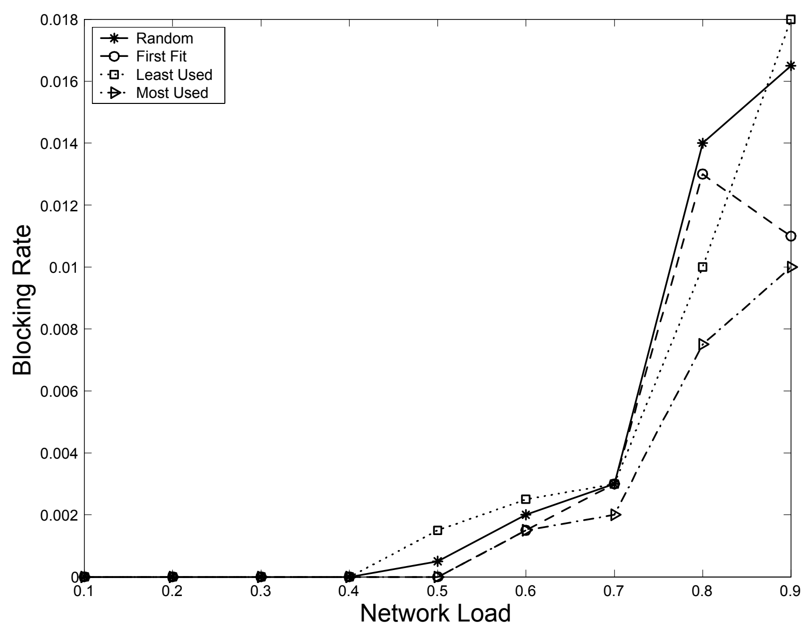

Figure 5-12: Comparison of blocking rates of all wavelength assignment policies for NSFnet with W=4 using FC-based algorithm

5.5.2 NSFnet with W=8

Figures 5-13, 5-14, 5-15, and 5-16 show the performance of our algorithm on the NSFnet network with 8 wavelengths per link (W=8) using random, first-fit, least-used, and most-used wavelength assignment, respectively. In each case, our algorithm outperforms the fixed-SP and alternate routing methods. In fact, the FC-based approach achieved a zero blocking rate in all scenarios. Table 5.6 shows the average blocking rate over all network loads for each algorithm. The average running time per connection for each approach is shown in Table 5.7. As with the NSFnet with W=4, the improvement in blocking rate far outweighs the increase in time.

A comparison of blocking rates of the FC-based algorithm for the various WAPs is illustrated in Figure 5-17. All of the WAPs perform in a superior manner.

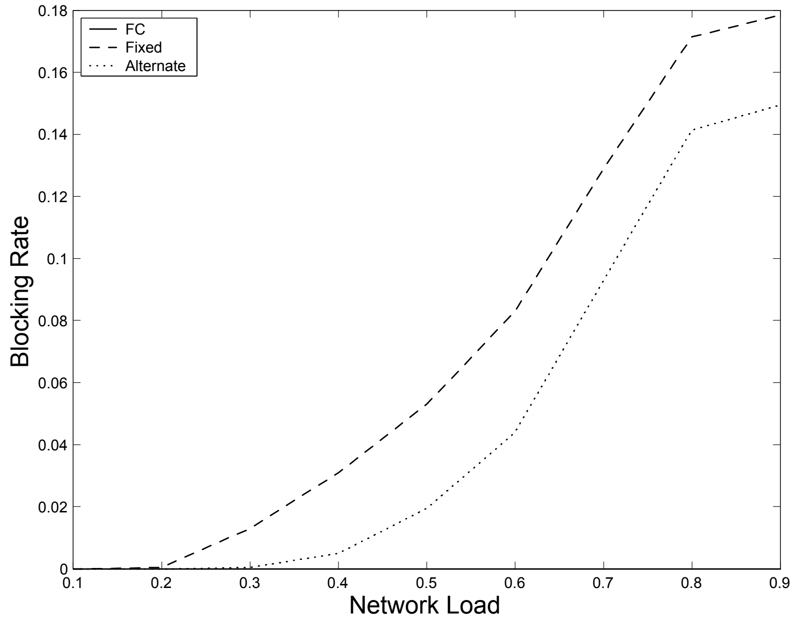

Figure 5-13: Blocking rates for FC-based algorithm using random wavelength assignment for NSFnet with W=8

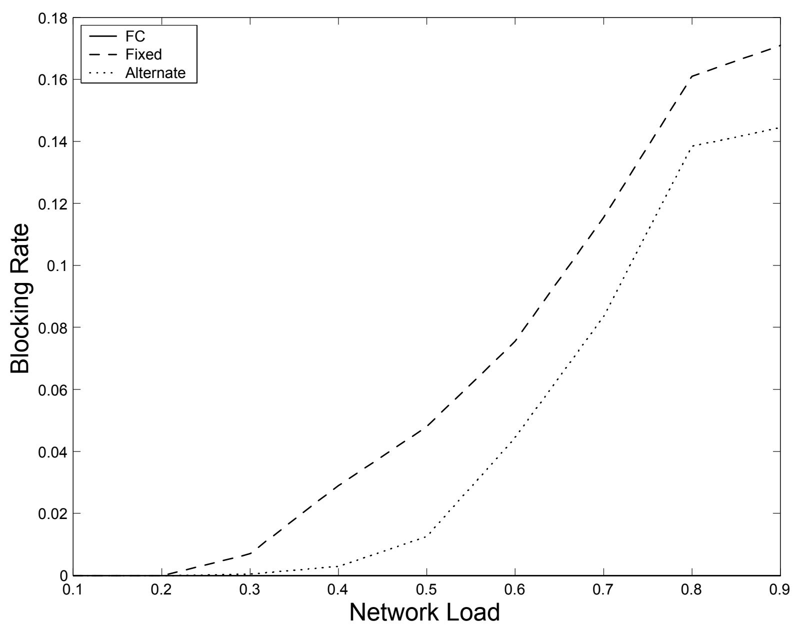

Figure 5-14: Blocking rates for FC-based algorithm using first-fit wavelength assignment for NSFnet with W=8

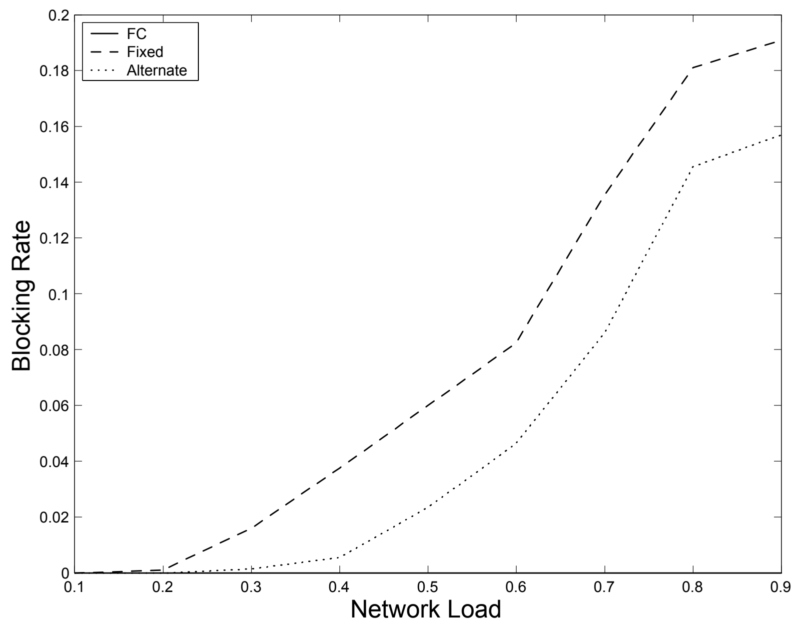

Figure 5-15: Blocking rates for FC-based algorithm using least-used wavelength assignment for NSFnet with W=8

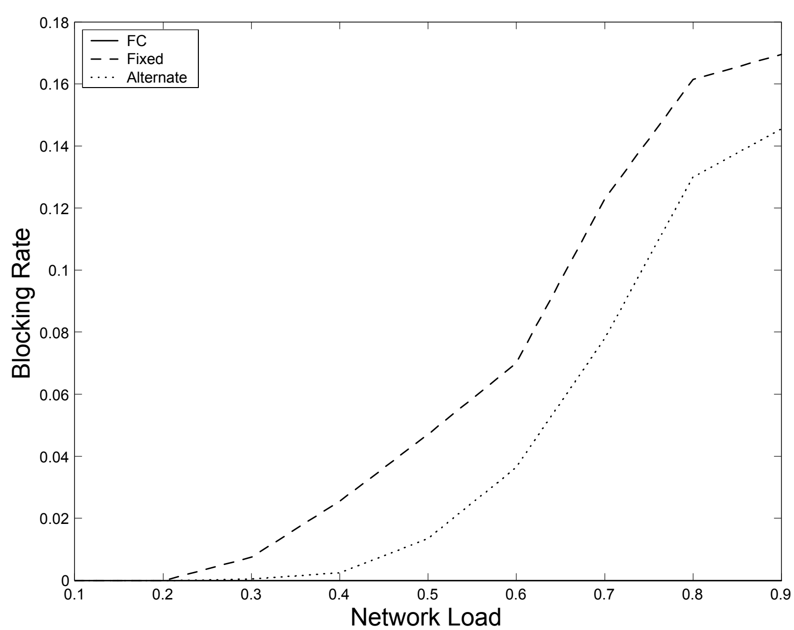

Figure 5-16: Blocking rates for FC-based algorithm using most-used wavelength assignment for NSFnet with W=8

Table 5.6: Average blocking rates for FC, fixed, and alternate routing algorithms for NSFnet with W=8

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	**%**	**ABR**	**%**	**ABR**	**%**	**ABR**	**%**
FC	0.0000		0.0000		0.0000		0.0000	
Fixed	0.0733	∞	0.0674	∞	0.0783	∞	0.0671	∞
Alternate	0.0503	∞	0.0474	∞	0.0517	∞	0.0452	∞

Table 5.7: Average running times for FC, fixed, and alternate routing algorithms for NSFnet with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	**%**	**ART**	**%**	**ART**	**%**	**ART**	**%**
FC	0.0087		0.0086		0.0089		0.0090	
Fixed	0.0005	-1912	0.0004	-1993	0.0004	-2056	0.0005	-1846
Alternate	0.0006	-1400	0.0006	-1439	0.0006	-1542	0.0007	-1331

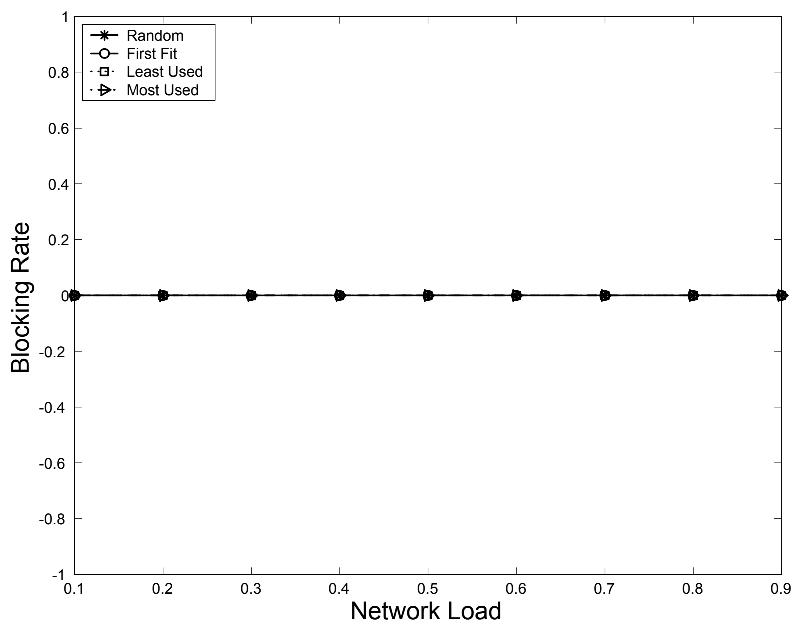

Figure 5-17: Comparison of blocking rates of all wavelength assignment policies for NSFnet with W=8 using FC-based algorithm

5.5.3 4×4 Mesh-Torus Network with W=4

Figures 5-18, 5-19, 5-20, and 5-21 show the performance of our algorithm on the mesh-torus network with 4 wavelengths per link (W=4) using random, first-fit, least-used, and most-used wavelength assignment, respectively. Again, the FC-based approach outperforms the fixed-SP and alternate routing methods. Table 5.8 shows the average blocking rate over all network loads for each algorithm. The average running time per connection for each approach is shown in Table 5.9. As with the other cases, the improvement in blocking rate far is much greater that the increase in time.

A comparison of blocking rates of the FC-based algorithm for the various WAPs is illustrated in Figure 5-22. The most-used WAP has the lowest average blocking rate among all of the policies.

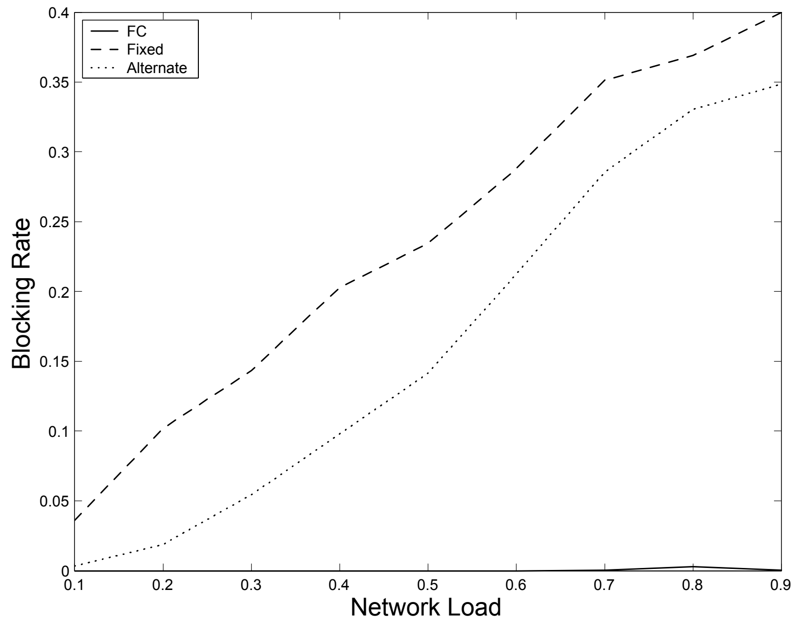

Figure 5-18: Blocking rates for FC-based algorithm using random wavelength assignment for the mesh-torus network with W=4

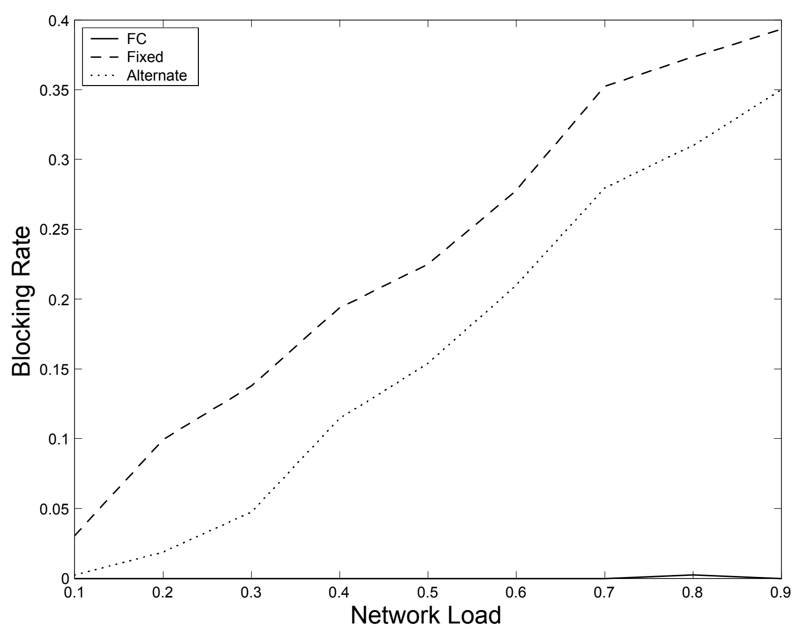

Figure 5-19: Blocking rates for FC-based algorithm using first-fit wavelength assignment for the mesh-torus network with W=4

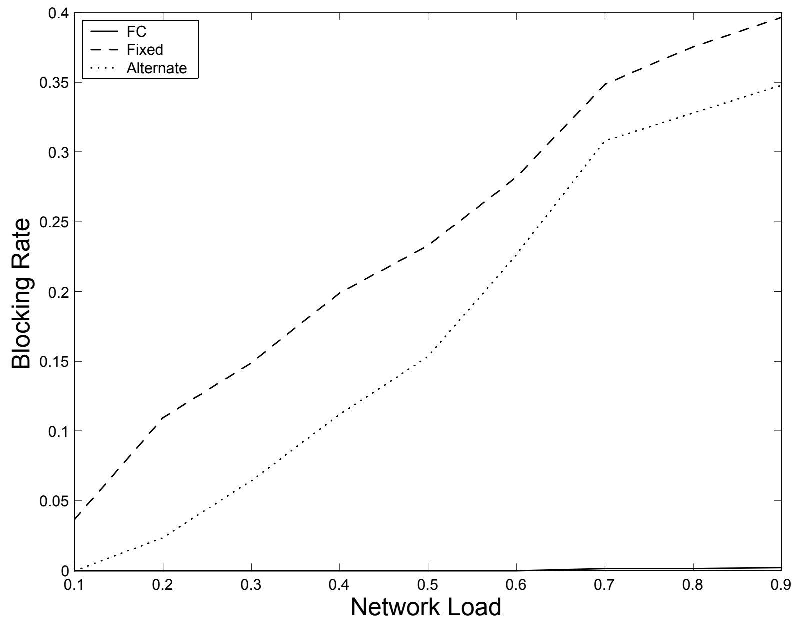

Figure 5-20: Blocking rates for FC-based algorithm using least-used wavelength assignment for the mesh-torus network with W=4

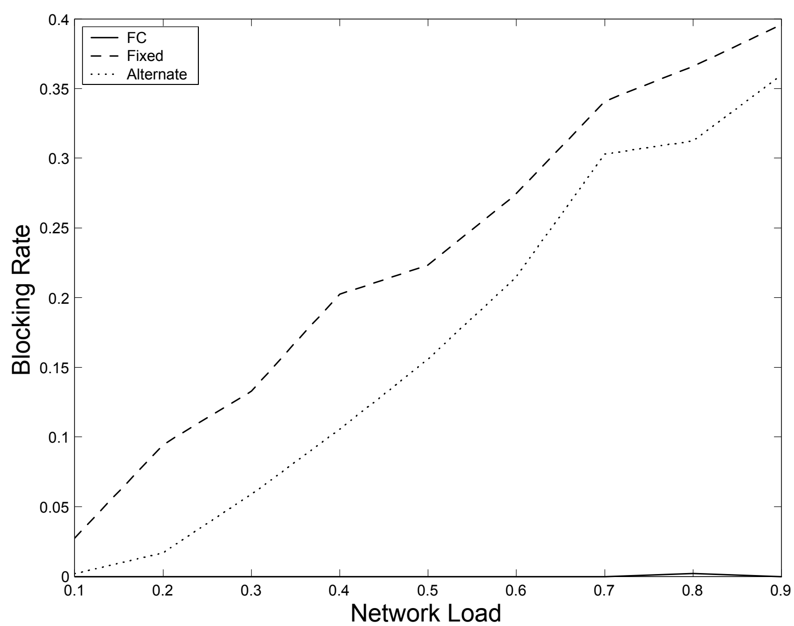

Figure 5-21: Blocking rates for FC-based algorithm using most-used wavelength assignment for the mesh-torus network with W=4

Table 5.8: Average blocking rates for FC, fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	%	**ABR**	%	**ABR**	%	**ABR**	%
FC	0.0004		0.0003		0.0006		0.0002	
Fixed	0.2364	53188	0.2316	83380	0.2366	42590	0.2287	102925
Alternate	0.1659	37338	0.1652	59480	0.1738	31280	0.1699	76475

Table 5.9: Average running times for FC, fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	%	**ART**	%	**ART**	%	**ART**	%
FC	0.0088		0.0087		0.0090		0.0089	
Fixed	0.0004	-2034	0.0004	-2128	0.0004	-2123	0.0004	-2006
Alternate	0.0008	-1150	0.0007	-1193	0.0008	-1136	0.0008	-1099

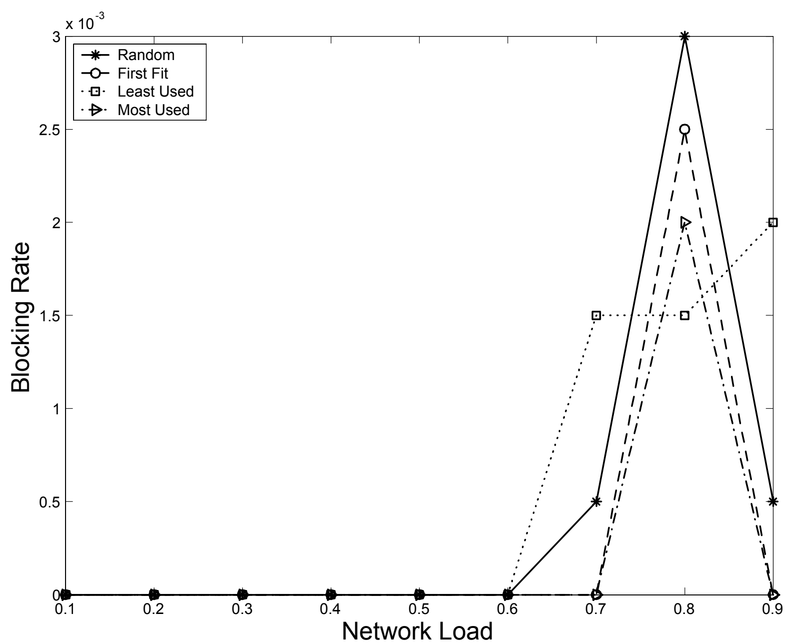

Figure 5-22: Comparison of blocking rates of all wavelength assignment policies for mesh-torus network with W=4 using FC-based algorithm

5.5.4 4×4 Mesh-Torus Network with W=8

Figures 5-23, 5-24, 5-25, and 5-26 show the performance of our algorithm on the mesh-torus network with 8 wavelengths per link (W=8) using random, first-fit, least-used, and most-used wavelength assignment, respectively. Again, the FC-based approach outperforms the fixed-SP and alternate routing methods by achieving a zero blocking rate. Table 5.10 shows the average blocking rate over all network loads for each algorithm. The average running time per connection for each approach is shown in Table 5.11. As with the other cases, the improvement in blocking rate far outweighs the increase in time since the FC-based method is able to achieve zero blocking.

A comparison of blocking rates of the FC-based algorithm for the various WAPs is illustrated in Figure 5-27. All WAPs perform in the same superior manner.

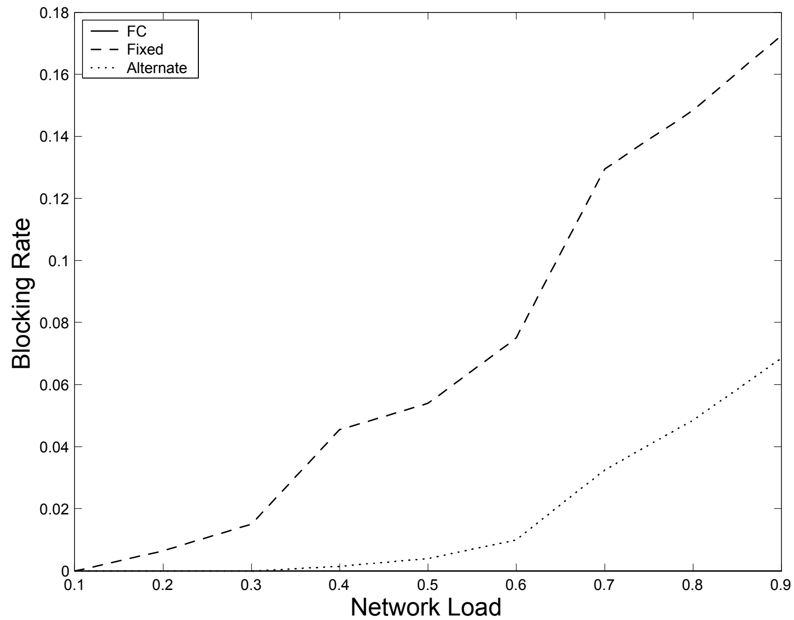

Figure 5-23: Blocking rates for FC-based algorithm using random wavelength assignment for the mesh-torus network with W=8

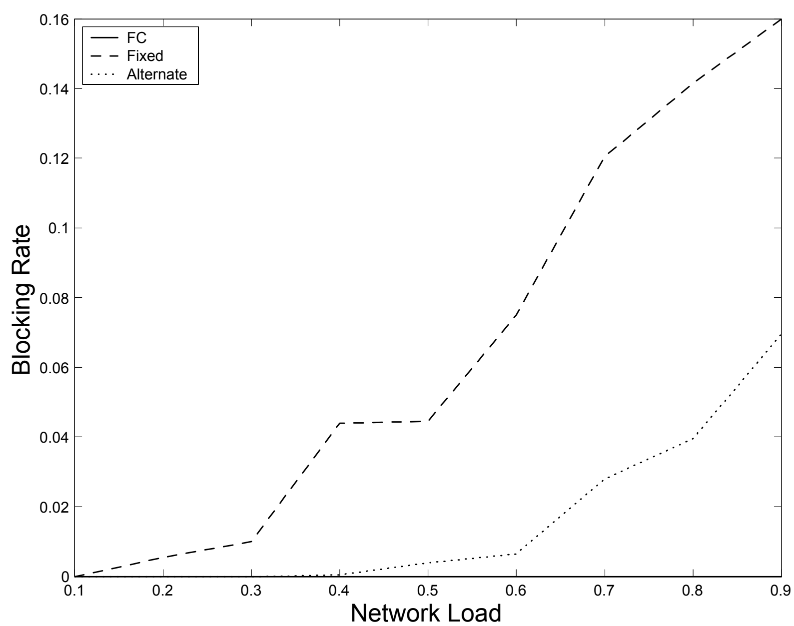

Figure 5-24: Blocking rates for FC-based algorithm using first-fit wavelength assignment for the mesh-torus network with W=8

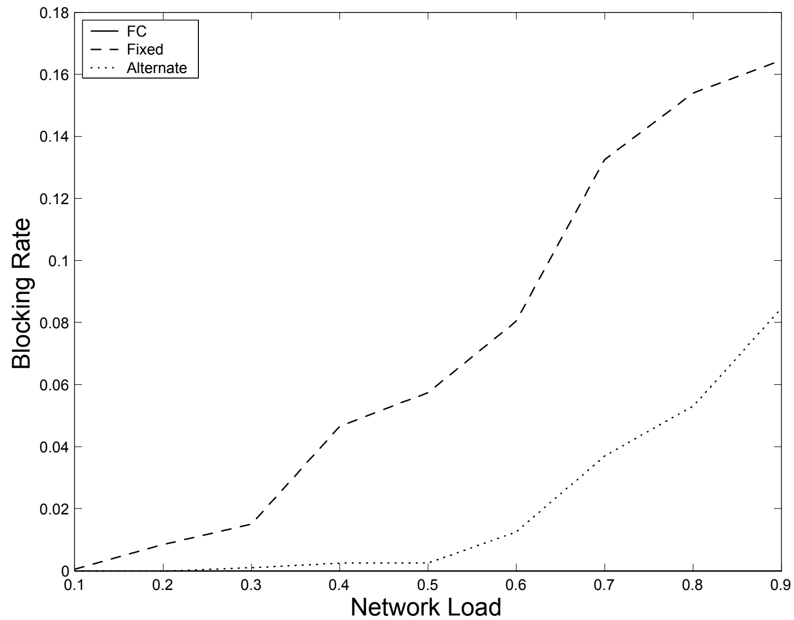

Figure 5-25: Blocking rates for FC-based algorithm using least-used wavelength assignment for the mesh-torus network with W=8

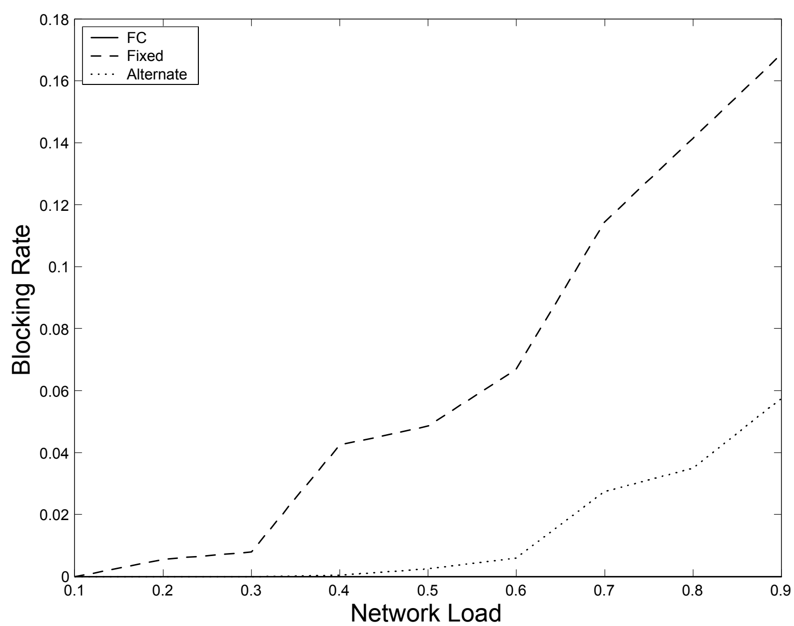

Figure 5-26: Blocking rates for FC-based algorithm using most-used wavelength assignment for the mesh-torus network with W=8

Table 5.10: Average blocking rates for FC, fixed, and alternate routing algorithms for 4x4 mesh-rorus network with W=8

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	**%**	**ABR**	**%**	**ABR**	**%**	**ABR**	**%**
FC	0.0000		0.0000		0.0000		0.0000	
Fixed	0.0718	∞	0.0668	∞	0.0733	∞	0.0662	∞
Alternate	0.0183	∞	0.0164	∞	0.0214	∞	0.0143	∞

Table 5.11: Average running times for FC, fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=8

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	**%**	**ART**	**%**	**ART**	**%**	**ART**	**%**
FC	0.0088		0.0086		0.0089		0.0089	
Fixed	0.0004	-1971	0.0004	-1988	0.0004	-2056	0.0005	-1862
Alternate	0.0006	-1460	0.0006	-1520	0.0006	-1572	0.0006	-1430

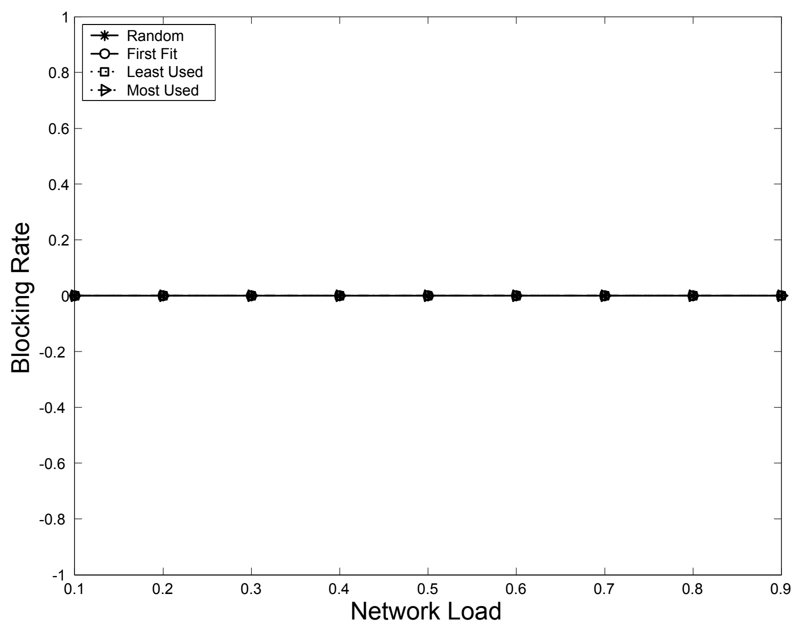

Figure 5-27: Comparison of blocking rates of all wavelength assignment policies for mesh-torus network with W=8 using FC-based algorithm

Chapter 6

A Simulated Annealing Approach

6.1 Introduction

Simulated Annealing (SA) is a heuristic search technique introduced independently by Kirkpatrick et al. [31] in 1982 and Cěrny [12] in 1985. The algorithm exploits the analogy between the annealing process of solids and the search for a solution of a large combinatorial optimization problem. SA has been used by researchers in many fields to find good solutions to a wide variety of problems.

The work of Metropolis et al. [37] in the area of statistical mechanics forms the basis of the ideas presented in the SA algorithm They introduced a simple algorithm for simulating the annealing process of a solid in a heat bath. In general, the annealing process consists of two steps [1]:

1. Increase the temperature of the heat bath to a maximum value at which the solid melts.

2. Carefully decrease the temperature of the heat bath until the solid freezes and no further changes occur.

To simulate this process, the Metropolis algorithm generates a sequence of states of the solid. Given a solid in state i with energy E_i, the next state j is generated by applying a small perturbation to the current state. The energy of the subsequent state is E_j and

Table 6.1: Physical Annealing Simulation vs. Simulated Annealing Algorithm

Metropolis Algorithm	**Simulated Annealing Algorithm**
Current state	Feasible solution
Next state	Neighboring solution
Energy of state	Cost of solution
Temperature	Control parameter
Frozen state	Final solution

the resulting change in energy $\Delta E = E_j - E_i$ is computed. If $\Delta E \leq 0$ the perturbation is accepted and state j is used as the current state in the next iteration. If $\Delta E > 0$ the perturbation is accepted with probability

$$\exp\left(\frac{E_i - E_j}{k_B T}\right) \tag{6.1}$$

where T denotes the temperature of the heat bath and k_B is the Boltzmann constant. This process is repeated for a predetermined number of iterations at each temperature. The temperature is then decreased until the solid freezes into its ground state. It is important to note that the temperature of the heat bath must be reduced slowly in order to maintain thermal equilibrium at each temperature.

6.2 Simulated Annealing

Kirkpatrick et al. [31] and Cĕrny [12] independently showed how the Metropolis algorithm could be applied to combinatorial optimization problems. Elements of the optimization problem are used in place of the states, energy, and temperature found in the annealing simulation algorithm as shown in Table 6.1.

For an instance of a combinatorial optimization problem defined by the solution space S and with cost function f, we will consider the case where a minimal optimal solution is sought. The SA algorithm starts with an initial feasible solution $s_0 \in S$ with cost $f(s_0)$ and an initial temperature $t_0 > 0$. A neighbor s of s_0 is generated using a predetermined procedure and its cost $f(s)$ is computed. A *transition* from the current solution s_0 to s occurs with probability

Algorithm 6.1 Basic SA Procedure

```
Initialize:
    Initial solution: S.
    Initial temperature: T_0.
    Temperature length: L.
    Temperature factor: α.
While (algorithm not frozen)
    Set iteration counter: i = 0.
    While(i < L)
        Select S' ∈ N(S).
        Set Δ = c(S') − c(S).
        If Δ ≤ 0
          Set S = S'.
        Else
          Set x = random number in range (0,1).
          If x < exp(−Δ/t)
            Set S = S'.
          End
        End
        Set i = i + 1.
    End
    Set T_i = αT_{i−1}.
End
```

$$P_t\left(accept\ s\right) = \begin{cases} 1 & \text{if } f\left(s\right) \leq f\left(s_0\right) \\ \exp\left(\frac{f(s_0)-f(s)}{t}\right) & \text{if } f\left(s\right) \geq f\left(s_0\right) \end{cases} \tag{6.2}$$

From Equation 6.2 we see that a transition always occurs when a solution with lower cost is found. Solutions with higher costs are accepted with decreasing frequency as the algorithm progresses. This procedure is repeated for a specific number of iterations at each temperature. A general description of the simulated annealing algorithm for a minimization problem is shown in Algorithm 6.1.

6.3 SA-based Adaptive RWA Algorithm

For a network with N nodes, L links and W wavelengths per link, a potential solution to the RWA problem for a given source s and destination d is the pair (r, w) where r is a route from s to d and $w \in W$ is a valid wavelength. Let L^r be the set of links that compose r. Then solution (r, w) is feasible if w is available on all links $l \in L^r$ The cost of a solution is defined as the sum of the costs for each link $l \in L^r$. For our SA algorithm, the cost of using wavelength w on route r is defined by Equations 6.3 and 6.4. Note that while infeasible solutions are allowed, they are penalized for links where the specified wavelength is unavailable.

$$c(l, w) = \begin{cases} 1 \text{ if } w \text{ is available on } l \\ 10000 \text{ otherwise} \end{cases} \tag{6.3}$$

$$c\,(r, w) = \sum_{l \in L^r} c(l, w) \tag{6.4}$$

Given a solution (r, w), we define route r by a list of nodes ordered as they occur along the path, i.e. $r = (r_1\ r_2\ r_3 \cdots r_K)$ where K is the number of nodes in r $(K \leq N)$.and r_i is the index of the i^{th} node visited. Note that for a route from source s to destination d, $r_1 = s$ and $r_K = d$. A neighboring solution is found using the following procedure for the SA algorithm.

1. Randomly select a node n from the set $\{r_2\ r_3 \cdots r_{K-1}\}$
2. Randomly select a node n' from the set of nodes adjacent to n.
3. Use Dijkstra's algorithm to generate the shortest path r' from r_1 to n' and the shortest path r'' from n' to r_K,
4. If paths r' and r'' do not share any common nodes (other than n'), combine r' and r'' to form neighboring route $\widehat{r} = r' + r''$.

There are many ways to implement the SA algorithm. We have chosen an implementation similar to one proposed by Johnson et. al. [28]. Several modifications are made to adapt the algorithm to the RWA problem for optical networks. Our implementation of the SA algorithm requires several generic parameters that are described below.

START_TEMP

This is the initial temperature for the first iteration of the algorithm. Several shorter trial runs are performed to determine an initial value that balances our need for a fast running time and a good solution. We chose to start at a low initial temperature of 25 in order to shorten the running time of the algorithm.

TEMP_FACTOR

This parameter is the descriptive name for α shown in Algorithm 6.1. It controls the rate at which the temperature decreases at each iteration and is also known as the *cooling ratio*. For our simulations, we set $\alpha = 0.9$.

TEMP_LENGTH

TEMP_LENGTH is the descriptive name for L shown in Algorithm 6.1. It is the number of times the inner loop of the SA algorithm is performed. L is proportional to the expected number of neighbors of a potential solution. In our simulations, we defined TEMP_LENGTH in the following manner

$$L = 2 * \sum_{n \in N} \text{nodes adjacent to } n$$

MIN_PERCENT

This parameter is used to test whether the algorithm is *frozen*. If five temperatures are completed consecutively where the percentage of accepted moves is MIN_PERCENT or less, the algorithm is determined to be frozen and terminated. We chose MIN_PERCENT=2%.

The pseudo code for the SA-based adaptive RWA algorithm is shown in Algorithm 6.2.

Algorithm 6.2 SA-based Adaptive RWA

```
Initialize:
    S = L × W zero matrix.
    T =[empty table].
While (termination criterion not fulfilled)
    Wait for a request to arrive (connection or termination).
    If request is a connection request (s, d, h)
        Set t = current time.
        Use Dijkstra's algorithm to determine route r*.
        Use given wavelength assignment method to assign a wavelength w*.
        If an available wavelength w* is found
            Route connection on route r* with wavelength w*.
            Let L* be the set of links that compose route r*.
            Update S to show that wavelength w* is busy on links in L*.
            Update T by adding termination request (L*, w*, t + h).
        Else
            Use SA algorithm to search for an alternate solution.
            If an alternate solution (r*, w*) is found
                Route connection on route r* with wavelength w*.
                Let L* be the set of links that compose route r*.
                Update S to show that wavelength w* is busy on links in L*.
                Update T by adding termination request (L*, w*, t + h).
            Else
                Connection request is blocked.
            End
        End
    Else
        Termination request (L*, w*) arrives.
        Update S to show that wavelength w* is available on links in L*.
    End
End
```

6.4 Performance Analysis

The performance of the SA-based adaptive RWA algorithm was evaluated on the NSFnet network shown in Figure 3-1 and the 4x4 mesh-torus network illustrated in Figure 3-2 . A dynamic traffic model in which connection requests arrive at each node according to a Poisson process with network wide arrival rate β is used for simulations. The connection holding time is exponentially distributed with mean $1/\mu$ resulting in an average load per (s, d) pair $\rho = \beta/N(N-1)\mu$. A node may be involved in multiple sessions and parallel sessions may occur between an (s,d) pair.

6.4.1 NSFnet with W=4

Figures 6-1, 6-2, 6-3, and 6-4 show the performance of our algorithm on the NSFnet network with 4 wavelengths per link (W=4) using random, first-fit, least-used, and most-used wavelength assignment, respectively. In each case, our algorithm outperforms the fixed-SP and alternate routing methods. Table 6.2 shows the average blocking rate over all network loads for each algorithm. The average running time per connection for each approach is shown in Table 6.3. The average blocking rate is decreased while the average running time increases as a result of using the SA-based algorithm. For the most-used wavelength assignment policy, Table 6.3 shows that the alternate routing method yields an average running time that is 42000% shorter than that of the SA-based approach. However, the SA-based method improves the average blocking rate by over 450%. Unlike the FC-based method, the improvement in blocking rate may not outweigh such a large increase in running time. In many situations, it may be impractical to wait $0.3 - 0.4$ seconds for each routing decision.

A comparison of blocking rates of the SA-based algorithm for the various WAPs is illustrated in Figure 6-5. Although there is no WAP that dominates all of the others, the most-used strategy has the lowest average blocking rate over all loads tested. The random approach also exhibited a good performance.

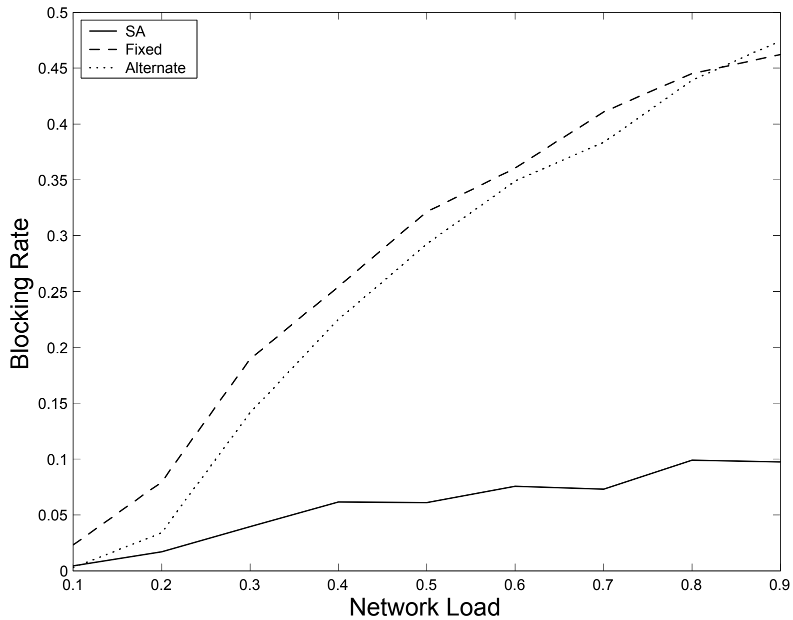

Figure 6-1: Blocking rates for SA-based algorithm using random wavelength assignment for NSFnet with W=4

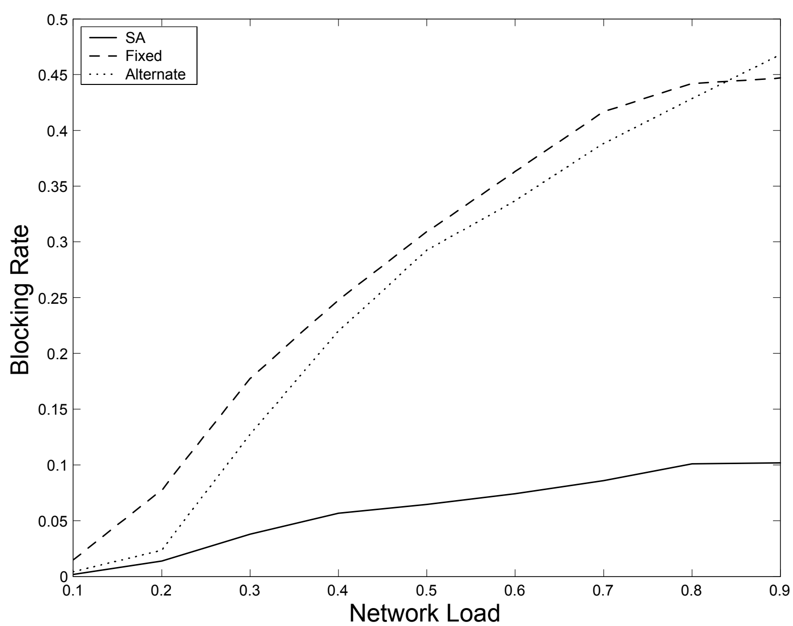

Figure 6-2: Blocking rates for SA-based algorithm using first-fit wavelength assignment for NSFnet with W=4

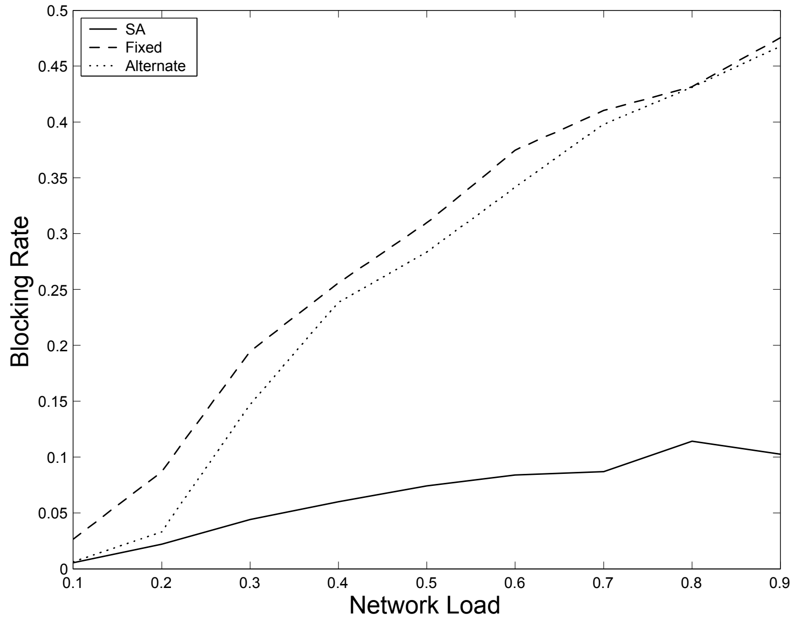

Figure 6-3: Blocking rates for SA-based algorithm using least-used wavelength assignment for NSFnet with W=4

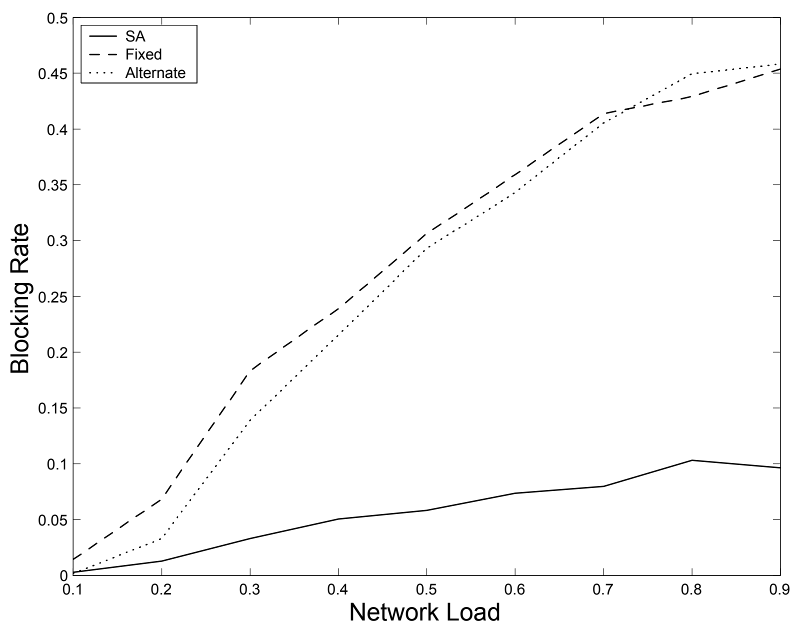

Figure 6-4: Blocking rates for SA-based algorithm using most-used wavelength assignment for NSFnet with W=4

Table 6.2: Average blocking rates for SA, fixed, and alternate routing algorithms for NSFnet with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	%	**ABR**	%	**ABR**	%	**ABR**	%
SA	0.0587		0.0598		0.0659		0.0567	
Fixed	0.2829	482	0.2773	464	0.2851	433	0.2742	483
Alternate	0.2602	443	0.2544	426	0.2607	396	0.2598	458

Table 6.3: Average running times for SA, fixed, and alternate routing algorithms for NSFnet with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	%	**ART**	%	**ART**	%	**ART**	%
SA	0.4571		0.3142		0.3100		0.3549	
Fixed	0.0005	-97955	0.0004	-74413	0.0004	-75400	0.0004	-79863
Alternate	0.0009	-52077	0.0008	-40396	0.0008	-37700	0.0008	-42033

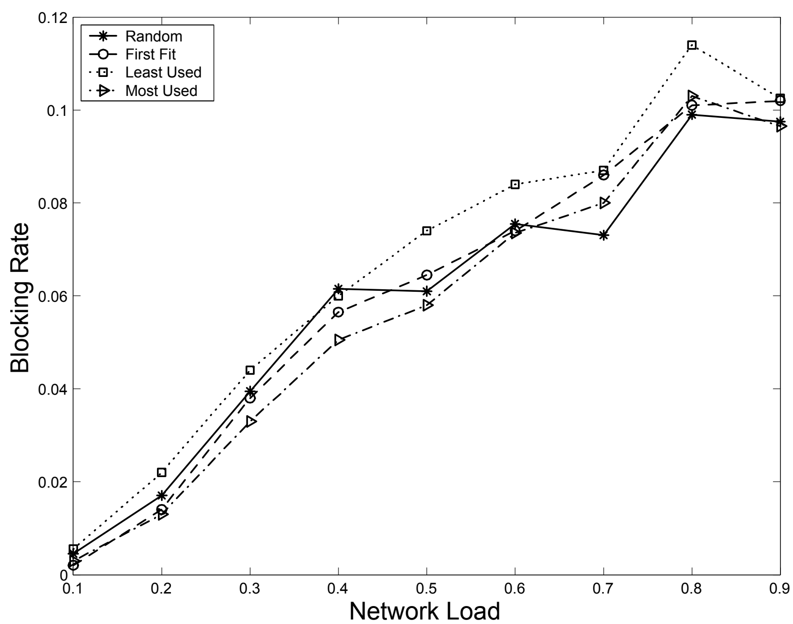

Figure 6-5: Comparison of blocking rates of all wavelength assignment policies for NSFnet with W=4

6.4.2 NSFnet with W=8

Figures 6-6, 6-7, 6-8, and 6-9 show the performance of our SA-based algorithm on the NSFnet network with 8 wavelengths per link (W=8) using random, first-fit, least-used, and most-used wavelength assignment, respectively. Again, the SA-based approach outperforms the fixed-SP and alternate routing methods. Table 6.4 shows the average blocking rate over all network loads for each algorithm. The average running time per connection for each approach is shown in Table 6.5. Again, the percent increase in running time is much greater that the percent decrease in blocking rate obtained by the SA-based method.

A comparison of blocking rates of the SA-based algorithm for the various WAPs is illustrated in Figure 6-10. The most-used WAP shows the best performance out of the four strategies.

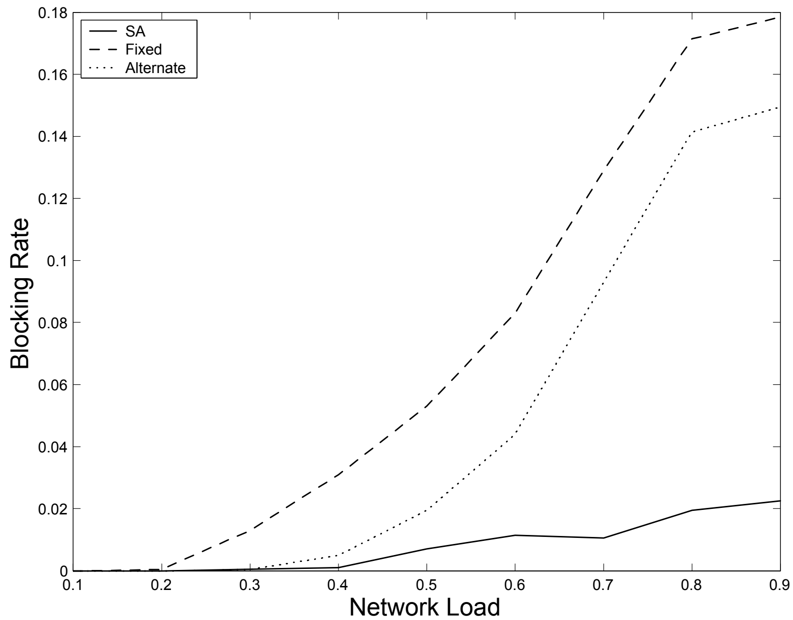

Figure 6-6: Blocking rates for SA-based algorithm using random wavelength assignment for NSFnet with W=8

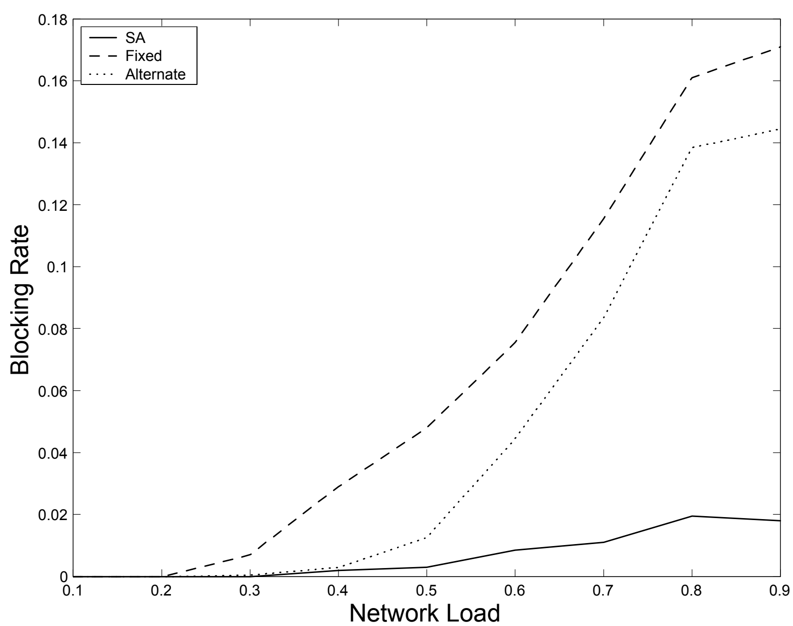

Figure 6-7: Blocking rates for SA-based algorithm using first-fit wavelength assignment for NSFnet with W=8

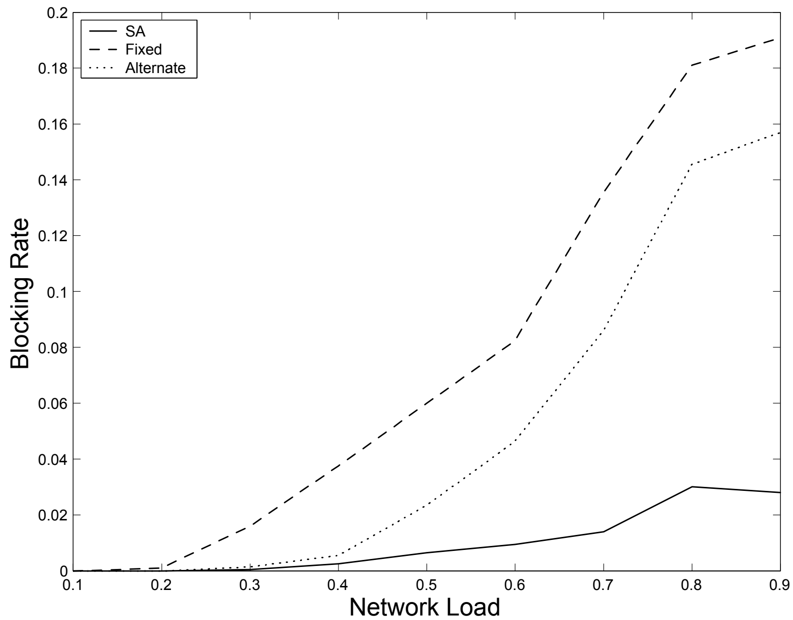

Figure 6-8: Blocking rates for SA-based algorithm using least-used wavelength assignment for NSFnet with W=8

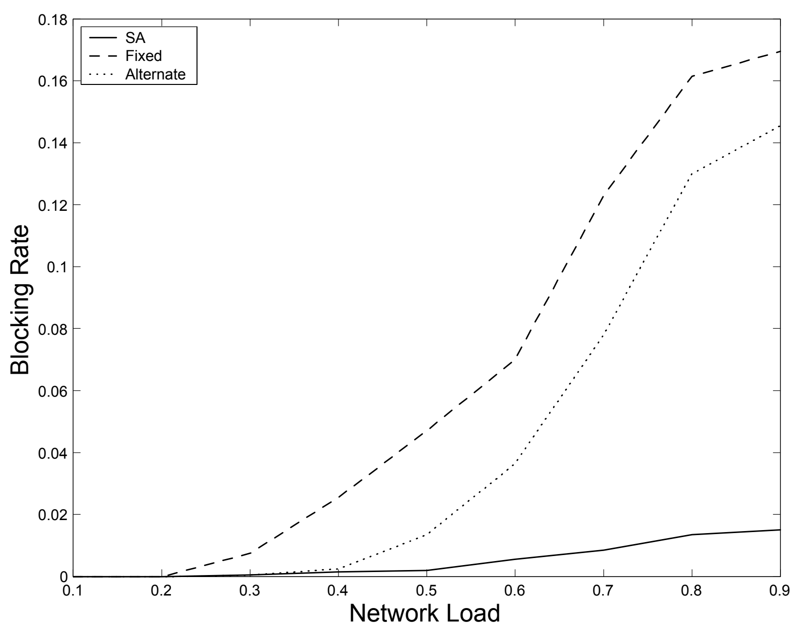

Figure 6-9: Blocking rates for SA-based algorithm using most-used wavelength assignment for NSFnet with W=8

Table 6.4: Average blocking rates for SA, fixed, and alternate routing algorithms for NSFnet with W=8

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	%	**ABR**	%	**ABR**	%	**ABR**	%
SA	0.0081		0.0069		0.0101		0.0052	
Fixed	0.0733	910	0.0674	979	0.0783	774	0.0671	1299
Alternate	0.0503	625	0.0474	689	0.0517	512	0.0452	874

Table 6.5: Average running times for SA, fixed, and alternate routing algorithms for NSFnet with W=8

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	%	**ART**	%	**ART**	%	**ART**	%
SA	0.0967		0.0477		0.0618		0.0510	
Fixed	0.0005	-21222	0.0004	-11010	0.0004	-14272	0.0005	-10423
Alternate	0.0006	-15538	0.0006	-7952	0.0006	-10704	0.0007	-7518

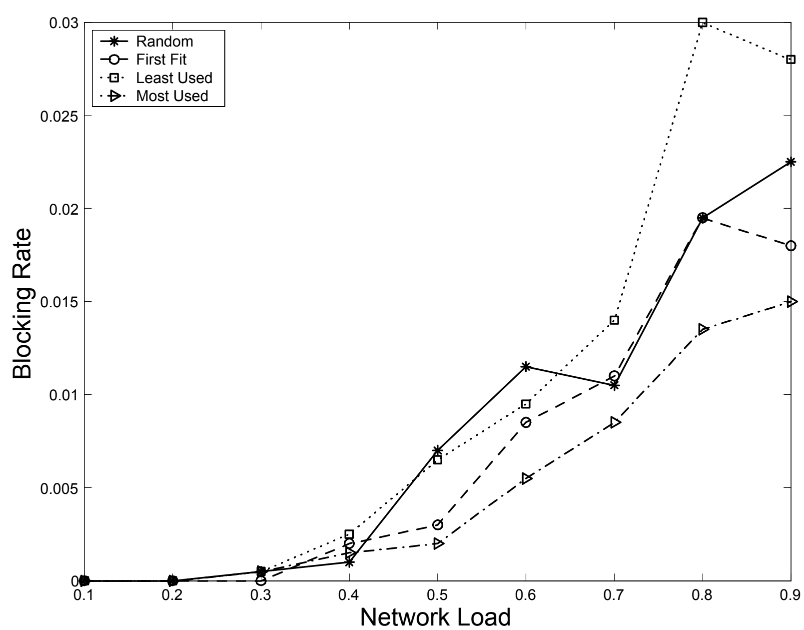

Figure 6-10: Comparison of blocking rates of all wavelength assignment policies for NSFnet with W=8 using SA-based algorithm

6.4.3 4×4 Mesh-Torus Network with W=4

Figures 6-11, 6-12, 6-13, and 6-14 show the performance of our SA-based algorithm on the 4×4 mesh-torus network with 4 wavelengths per link (W=4) using random, first-fit, least-used, and most-used wavelength assignment, respectively. In each case, our algorithm outperforms the fixed-SP and alternate routing methods. Table 6.6 shows the average blocking rate over all network loads for each algorithm. The average running time per connection for each approach is shown in Table 6.7. As with the NSFnet, the average blocking rate is decreased by using the SA-based algorithm while the time spent making a routing decision for each connection increases.

A comparison of blocking rates of the SA-based algorithm for the various WAPs is illustrated in Figure 6-15. The plot shows no clear best performer, but the random and first-fit WAPs have the lowest average blocking rates among the policies tested.

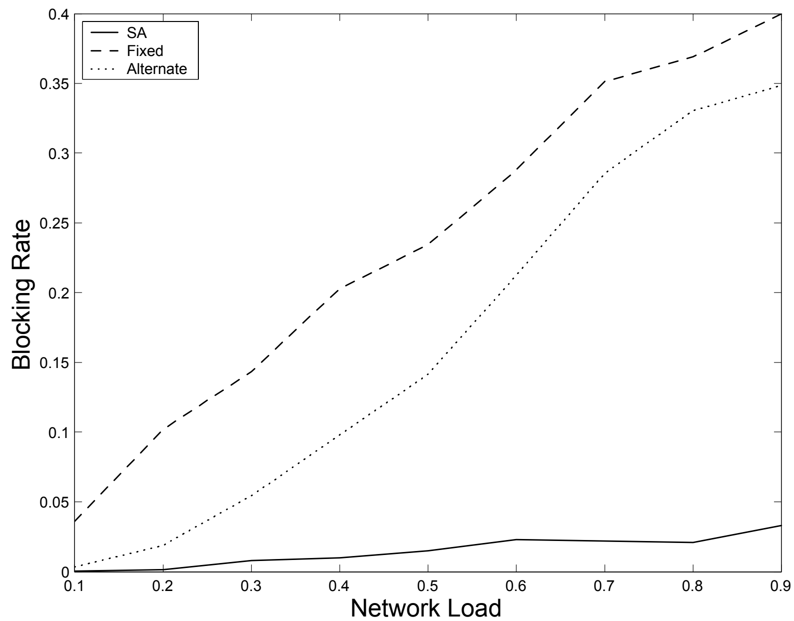

Figure 6-11: Blocking rates for SA-based algorithm using random wavelength assignment for 4×4 mesh-torus network with W=4

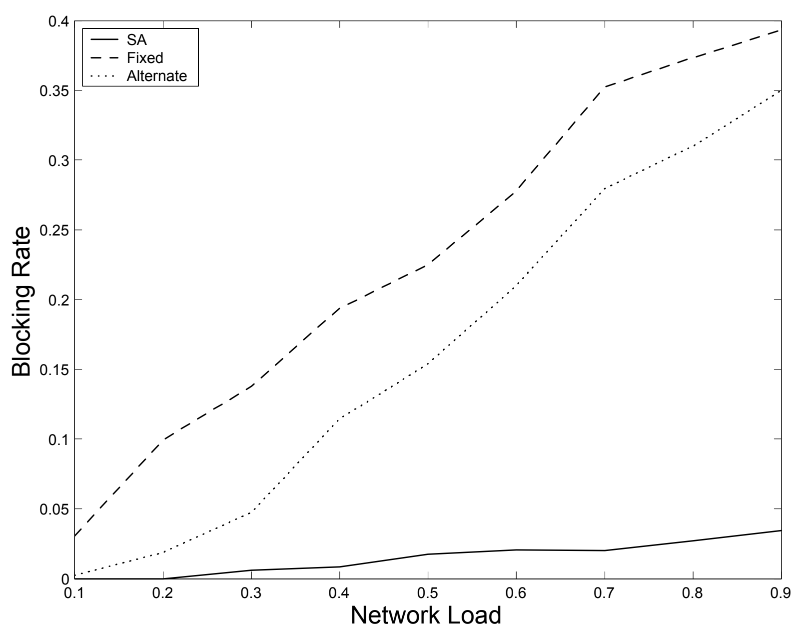

Figure 6-12: Blocking rates for SA-based algorithm using first-fit wavelength assignment for 4×4 mesh-torus network with W=4

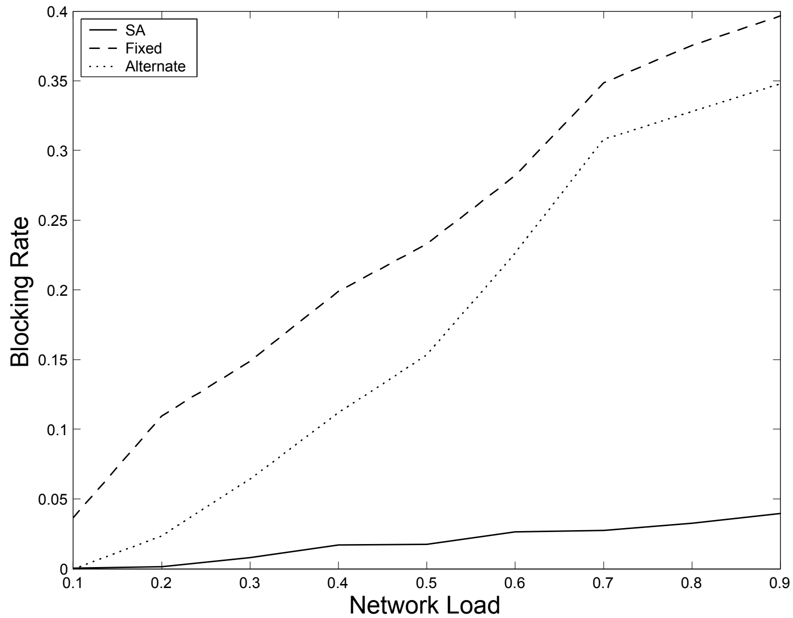

Figure 6-13: Blocking rates for SA-based algorithm using least-used wavelength assignment for 4×4 mesh-torus network with W=4

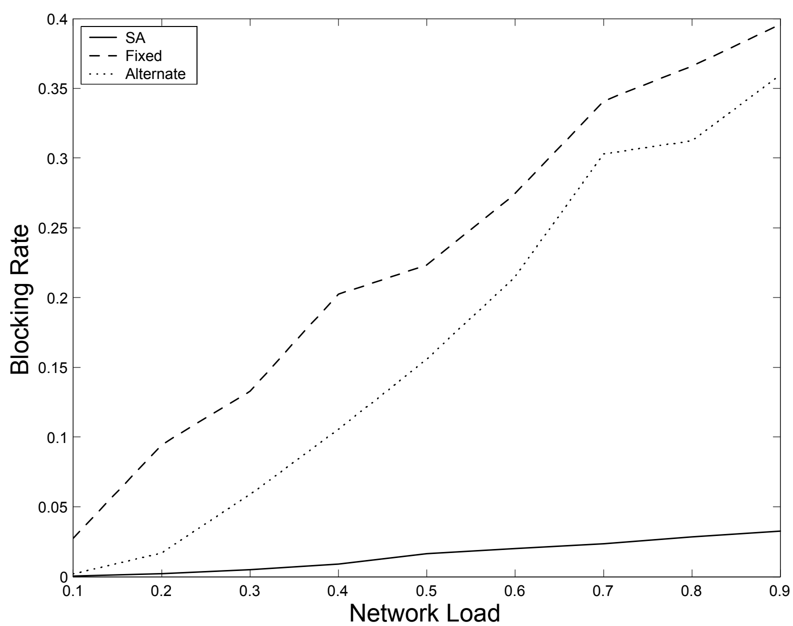

Figure 6-14: Blocking rates for SA-based algorithm using most-used wavelength assignment for 4×4 mesh-torus with W=4

Table 6.6: Average blocking rates for SA, fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	%	**ABR**	%	**ABR**	%	**ABR**	%
SA	0.0149		0.0149		0.0189		0.0153	
Fixed	0.2364	1588	0.2316	1556	0.2366	1249	0.2287	1497
Alternate	0.1659	1115	0.1652	1110	0.1738	917	0.1699	1112

Table 6.7: Average running times for SA, fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	%	**ART**	%	**ART**	%	**ART**	%
SA	0.5514		0.3011		0.2814		0.3396	
Fixed	0.0004	-127254	0.0004	-73249	0.0004	-66647	0.0004	-76400
Alternate	0.0008	-71926	0.0007	-41064	0.0008	-35670	0.0008	-41863

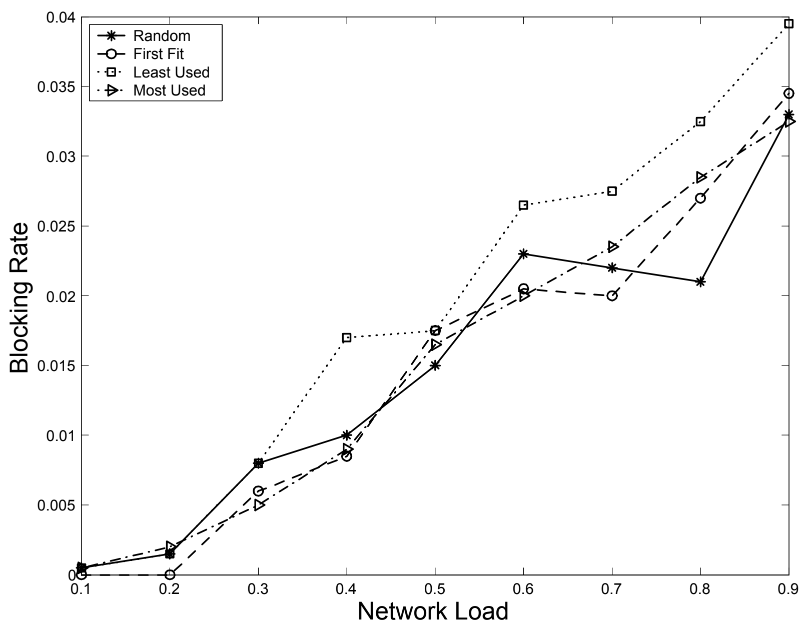

Figure 6-15: Comparison of blocking rates of all wavelength assignment policies for 4×4 mesh-torus network with W=4 using SA-based algorithm

6.4.4 4×4 Mesh-Torus Network with W=8

Figures 6-6, 6-7, 6-8, and 6-9 show the performance of our SA-based algorithm on the NSFnet network with 8 wavelengths per link (W=8) using random, first-fit, least-used, and most-used wavelength assignment, respectively. Again, the SA-based approach outperforms the fixed-SP and alternate routing methods by achieving very low blocking rates. Table 6.8 shows the average blocking rate over all network loads for each algorithm. The average running time per connection for each approach is shown in Table 6.9. For the mesh-torus network with 8 wavelengths per link, the improvement in blocking rate far is greater than the increase in time in most cases.

A comparison of blocking rates of the SA-based algorithm for the various WAPs is illustrated in Figure 6-20. All WAPs demonstrate a good performance. The first-fit strategy provides the lowest average blocking rate.

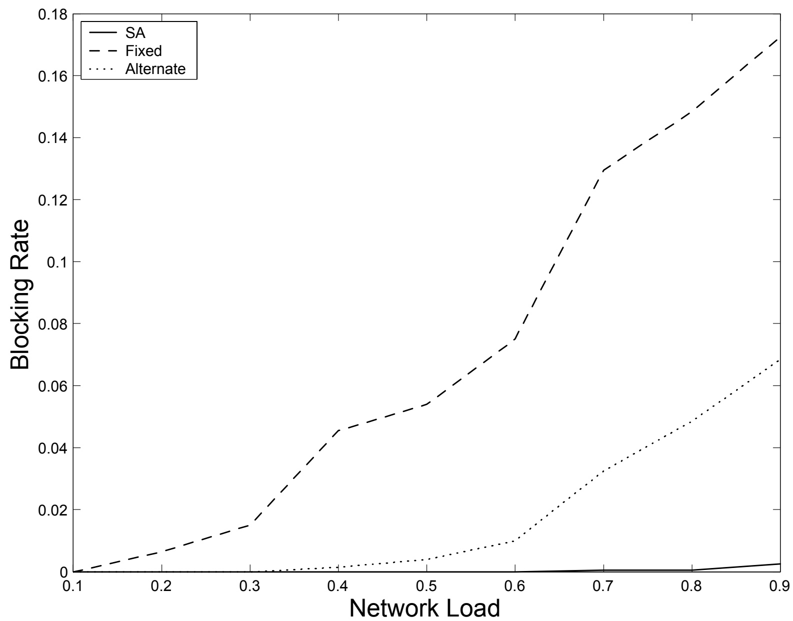

Figure 6-16: Blocking rates for SA-based algorithm using random wavelength assignment for 4×4 mesh-torus network with W=8

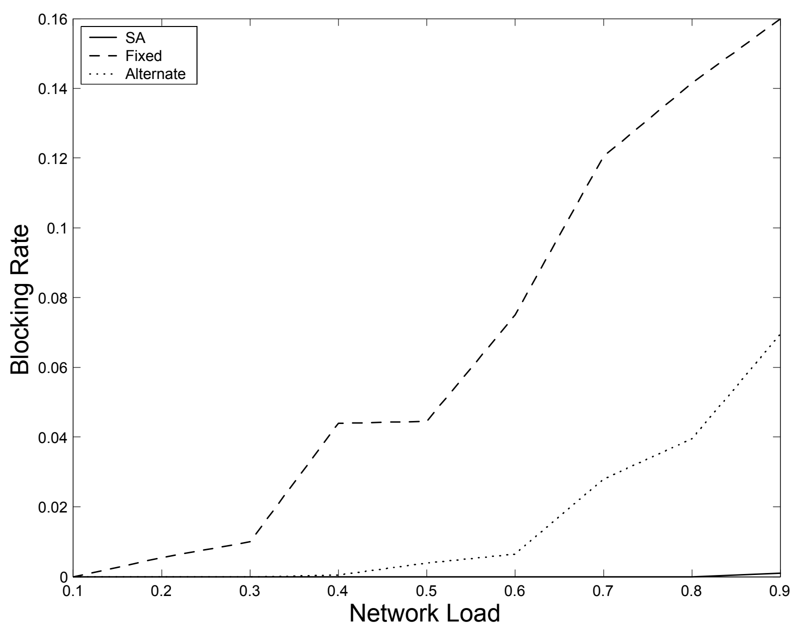

Figure 6-17: Blocking rates for SA-based algorithm using first-fit wavelength assignment for 4×4 mesh-torus network with W=8

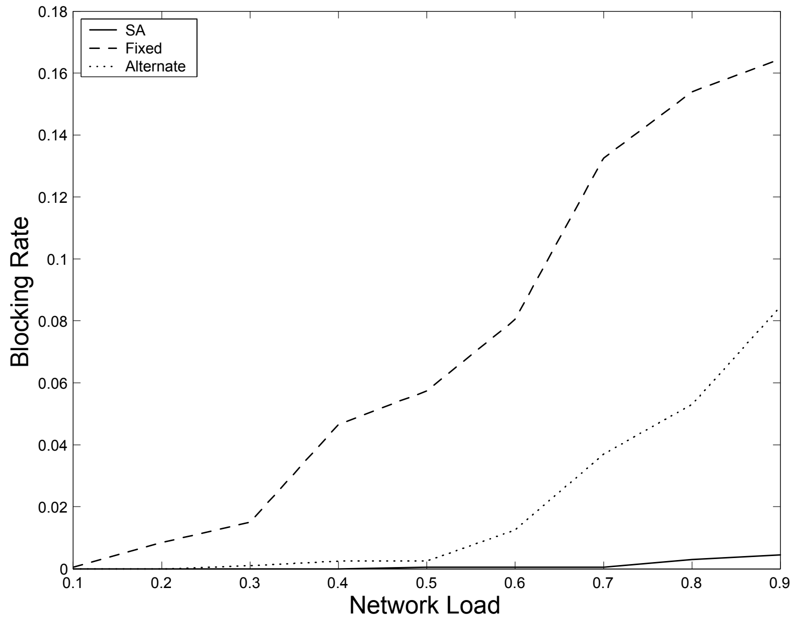

Figure 6-18: Blocking rates for SA-based algorithm using least-used wavelength assignment for 4×4 mesh-torus network with W=8

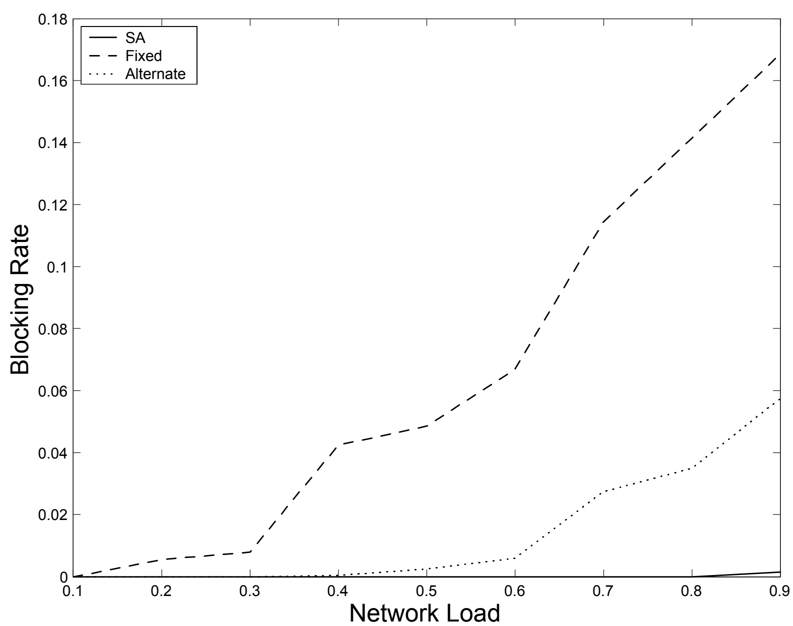

Figure 6-19: Blocking rates for SA-based algorithm using most-used wavelength assignment for 4×4 mesh-torus network with W=8

Table 6.8: Average blocking rates for SA, fixed, and alternate routing algorithms for 4x4 Mesh-Torus Network with W=8

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	%	**ABR**	%	**ABR**	%	**ABR**	%
SA	0.0004		0.0001		0.0010		0.0002	
Fixed	0.0718	18471	0.0668	60100	0.0733	7328	0.0662	39733
Alternate	0.0183	4714	0.0164	14800	0.0214	2144	0.0143	8600

Table 6.9: Average running times for SA, fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=8

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	%	**ART**	%	**ART**	%	**ART**	%
SA	0.0235		0.0077		0.0222		0.0130	
Fixed	0.0004	-5298	0.0004	-1787	0.0004	-5115	0.0005	-2716
Alternate	0.0006	-3924	0.0006	-1367	0.0006	-3912	0.0006	-2086

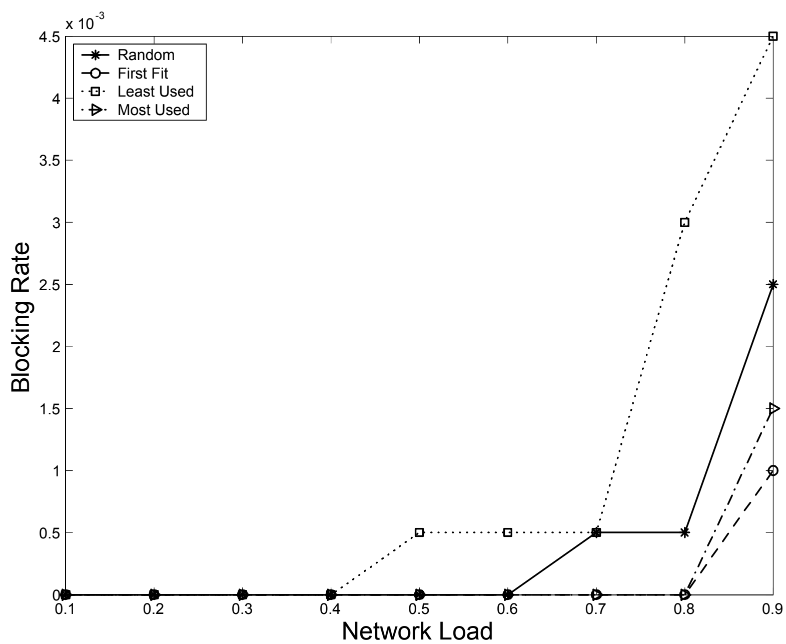

Figure 6-20: Comparison of blocking rates of all wavelength assignment policies for 4×4 mesh-torus network with W=8 using SA-based algorithm

Chapter 7

A Tabu Search Approach

7.1 Tabu Search

Tabu search (TS) is a meta-heuristic that guides a neighborhood search procedure from one potential solution to another. It differs from traditional local search methods because it is capable of escaping a local optima. TS operates by guiding the search through *neighborhood moves*. The TS algorithm considers several moves at each iteration. Some moves are considered *tabu or taboo* unless they result in a highly desirable outcome called the *aspiration criterion*. According to [19],

> Tabu search is based on the premise that problem solving, in order to qualify as intelligent, must incorporate adaptive memory and responsive exploration. ... The adaptive memory feature of TS allows the implementation of procedures that are capable of searching the solution space economically and effectively. ... The emphasis on responsive exploration in tabu search, whether in a deterministic or probabilistic implementation, derives from the supposition that a bad strategic choice can yield more information than a good random choice.

Thus, any implementation of TS contains some form of memory to hold information about the *tabu list*. The *recency-based* memory approach keeps track of recent solutions. Thus, a tabu list TL records the $|TL|$ solutions most recently visited. The size of TL may remain fixed or may vary at each iteration. The basic procedure for TS is described in Algorithm 7.1

Algorithm 7.1 Basic TS Procedure

Initialize:

  Initial solution: s.

  Best solution: $s^* = s$.

  Iteration counter: $i = 1$.

  Tabu list: $Tabu(i) = \oslash$.

While (stopping criterion not met)

  Generate $V^* \subseteq N(s, i)$.

  Choose best $s' \in V^*$.

  While *move* for s' is in $Tabu(i)+$

  If *move* for s' gives a highly desirable solution

    Override $Tabu(i)$ and accept current s'.

  Else

    Choose next best $s' \in V^*$.

  End

  Set $s = s'$.

  If $c(s') < c(s^*)$

    Set $s^* = s'$.

  End

  Update $Tabu(i)$.

  Set $i = i + 1$.

End

7.2 TS-based Adaptive RWA Algorithm

For a network with N nodes, L links and W wavelengths per link, a potential solution to the RWA problem for a given source s and destination d is the pair (r, w) where r is a route from s to d and $w \in W$ is a valid wavelength. Let L^r be the set of links that compose r. Then solution (r, w) is feasible if w is available on all links $l \in L^r$ The cost of a solution is defined as the sum of the costs for each link $l \in L^r$. For our TS-based algorithm, the cost of using wavelength w on route r is defined by Equations 7.1 and 7.2. Note that while infeasible solutions are allowed, they are penalized for links where the specified wavelength is unavailable.

$$c(l, w) = \begin{cases} 1 \text{ if } w \text{ is available on } l \\ 10000 \text{ otherwise} \end{cases} \tag{7.1}$$

$$c\,(r, w) = \sum_{l \in L^r} c(l, w) \tag{7.2}$$

Move Operations

Given a solution (r, w), we define route r by a list of nodes ordered as they occur along the path, i.e. $r = (r_1\ r_2\ r_3 \cdots r_K)$ where K is the number of nodes in r $(K \leq N)$ and r_i is the index of the i^{th} node visited. Note that for a route from source s to destination d, $r_1 = s$ and $r_K = d$ The neighborhood move operation for the TS-based algorithm is performed as follows.

For each node r_i , $i = 2, \cdots, K - 1$

  For each node n' adjacent to r_i

    Use Dijkstra's algorithm to generate the shortest path r' from r_1 to n'

    Use Dijkstra's algorithm to generate the shortest path $r''$from n' to r_K

    If paths r' and r'' share no common nodes (other than $n\prime$)

      Combine r' and r'' to create the mutated route $\hat{r} = r' + r''$.

    End

End

End

If no improvement in the best solution is found after 10 iterations, a second move operation is performed to diversify the search. Unlike the previous move operation, the second operation seeks to build a new route from a node that is not adjacent to any node r_i, $i = 2, 3, 4, \cdots, K - 1$ in r.

Tabu List and Aspiration Criteria

The tabu list records the node r_i around which the most recent neighboring solutions were built. The tabu status of a node r_i can be overridden if the cost of a resulting solution (r, w) is lower than that of any solution obtained earlier. For our algorithm, we chose a tabu list size of 7. Since the cut move seeks to diversify the search, the tabu list is erased after a cut move is performed.

The procedure for the TS-based adaptive RWA algorithm is shown in Algorithm 7.2. The TS portion of the algorithm is performed for only 20 iterations in order to reduce the overall running time.

Algorithm 7.2 TS-based Adaptive RWA

```
Initialize:
    S = L × W zero matrix.
    T =[empty table].
While (termination criterion not fulfilled)
    Wait for a request to arrive (connection or termination).
    If request is a connection request (s, d, h)
        Set t = current time.
        Use Dijkstra's algorithm to determine route r*.
        Use given wavelength assignment method to assign a wavelength w*.
        If an available wavelength w* is found
            Route connection on route r* with wavelength w*.
            Let L* be the set of links that compose route r*.
            Update S to show that wavelength w* is busy on links in L*
            Update T by adding termination request (L*, w*, t + h)
        Else
            Use TS algorithm to search for an alternate solution
            If an alternate solution (r*, w*) is found
                Route connection on route r* with wavelength w*.
                Let L* be the set of links that compose route r*.
                Update S to show that wavelength w* is busy on links in L*
                Update T by adding termination request (L*, w*, t + h)
            Else
                Connection request is blocked.
            End
        End
    Else
        Termination request (L*, w*) arrives.
        Update S to show that wavelength w* is available on links in L*
    End
End
```

7.3 Performance Analysis

The performance of the SA-based adaptive RWA algorithm was evaluated on the NSFnet network shown in Figure 3-1 and the 4×4 mesh-torus network illustrated in Figure 3-2 . A dynamic traffic model in which connection requests arrive at each node according to a Poisson process with network wide arrival rate β is used for simulations. The connection holding time is exponentially distributed with mean $1/\mu$ resulting in an average load per (s, d) pair $\rho = \beta/N(N-1)\mu$. A node may be involved in multiple sessions and parallel sessions may occur between an (s,d) pair. A total of 2000 connection requests were simulated.

7.3.1 NSFnet with W=4

Figures 7-1, 7-2, 7-3, and 7-4 show the performance of our TS-based algorithm on the NSFnet network with 4 wavelengths per link ($W = 4$) using random, first-fit, least-used, and most-used wavelength assignment, respectively. In each case, our algorithm outperforms the fixed-SP and alternate routing methods. Table 7.1 shows the average blocking rate over all network loads for each algorithm. The average running time per connection for each approach is shown in Table 7.2. The TS-based algorithm requires a much greater running time than the fixed and alternate routing methods. Depending upon the service requirements of the network,this great increase in running time may not be acceptable despite the improvement in blocking rates.

A comparison of blocking rates of the TS-based algorithm for the various WAPs is illustrated in Figure 7-5. Most-used WAP demonstrates the best overall performance for the network loads tested.

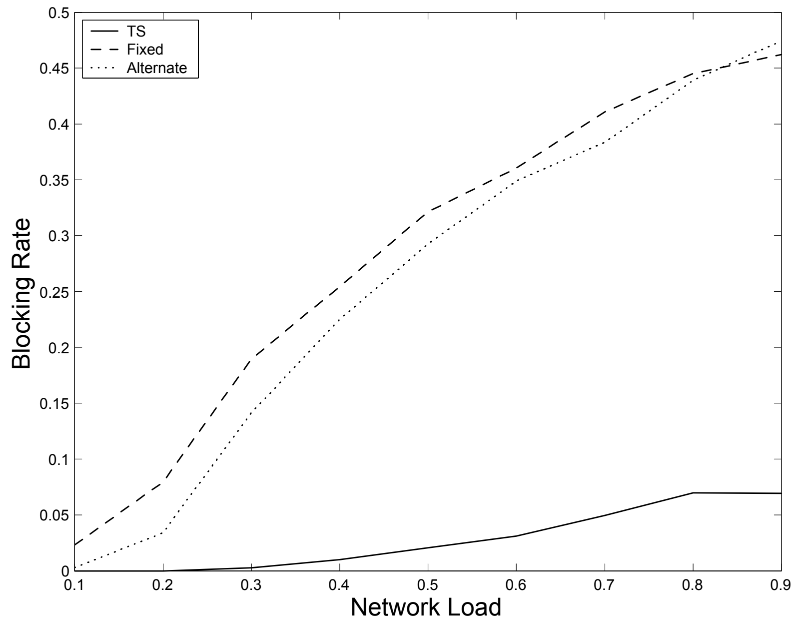

Figure 7-1: Blocking rates for TS-based algorithm using random wavelength assignment for NSFnet with W=4

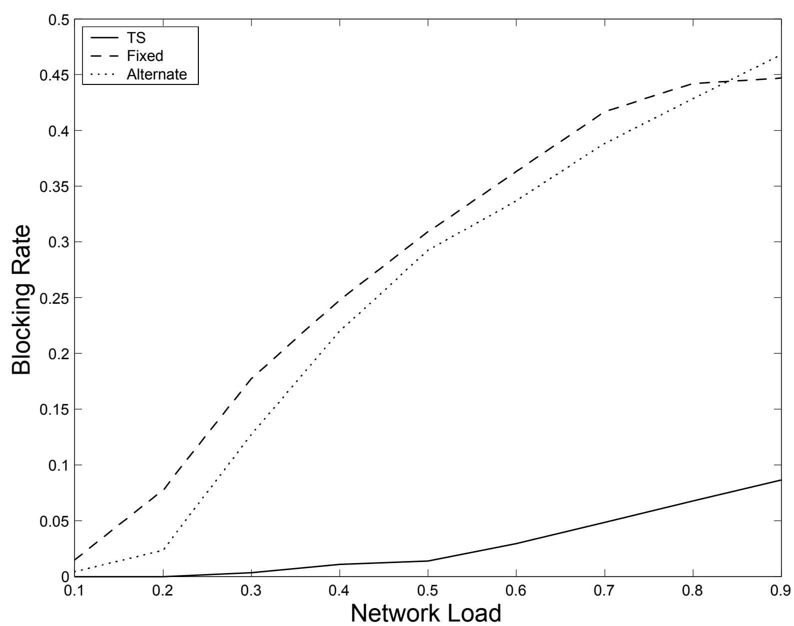

Figure 7-2: Blocking rates for TS-based algorithm using first-fit wavelength assignment for NSFnet with W=4

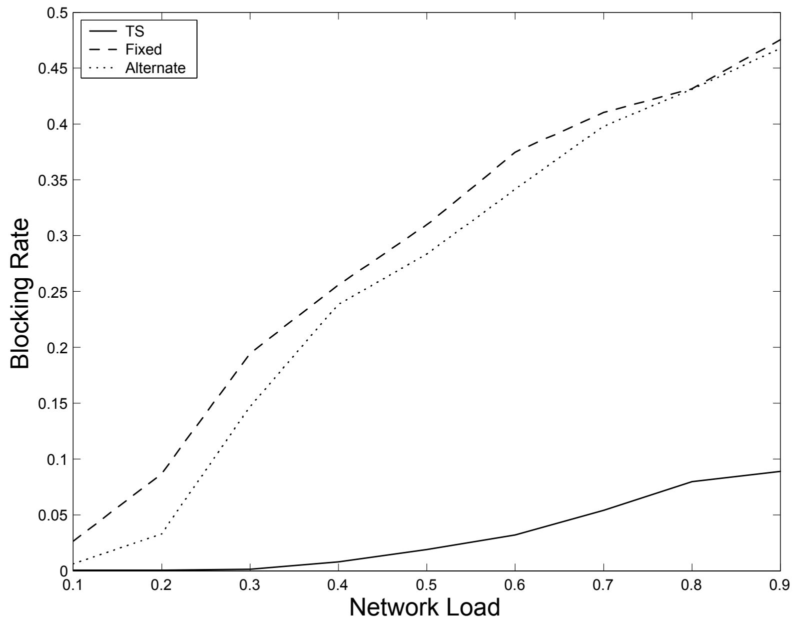

Figure 7-3: Blocking rates for TS-based algorithm using least-used wavelength assignment for NSFnet with W=4

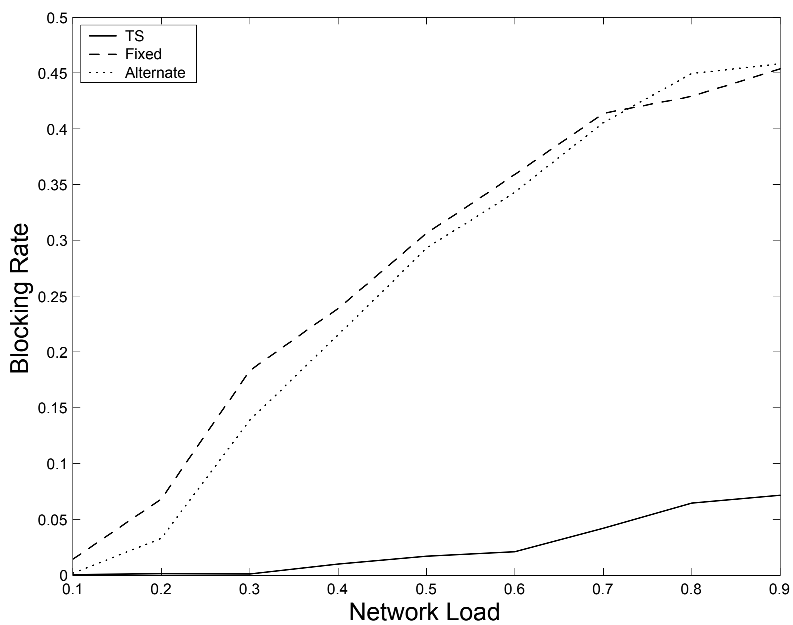

Figure 7-4: Blocking rates for TS-based algorithm using most-used wavelength assignment for NSFnet with W=4

Table 7.1: Average blocking rates for TS, fixed, and alternate routing algorithms for NSFnet with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	%	**ABR**	%	**ABR**	%	**ABR**	%
TS	0.0280		0.0289		0.0316		0.0254	
Fixed	0.2829	1011	0.2773	958	0.2851	902	0.2742	1078
Alternate	0.2602	929	0.2544	879	0.2607	825	0.2598	1021

Table 7.2: Average running times for TS, fixed, and alternate routing algorithms for NSFnet with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	%	**ART**	%	**ART**	%	**ART**	%
TS	0.1238		0.1195		0.1270		0.1189	
Fixed	0.0005	-26526	0.0004	-28305	0.0004	-30900	0.0004	-26758
Alternate	0.0009	-14103	0.0008	-15366	0.0008	-15450	0.0008	-14083

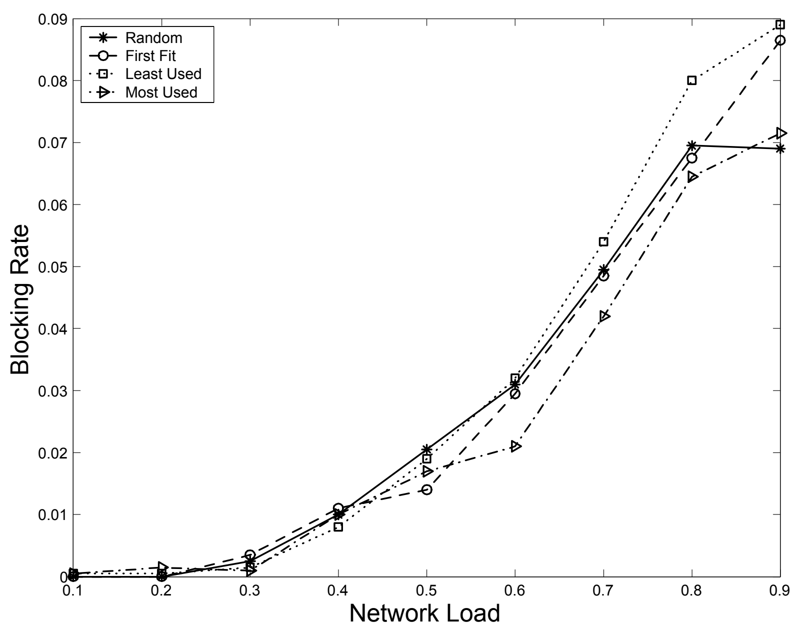

Figure 7-5: Comparison of blocking rates of all wavelength assignment policies for NSFnet with W=4

7.3.2 NSFnet with W=8

Figures 7-6, 7-7, 7-8, and 7-9 show the performance of our TS-based algorithm on the NSFnet network with 8 wavelengths per link ($W = 8$) using random, first-fit, least-used, and most-used wavelength assignment, respectively. Tables 7.3 and 7.4 show the average blocking rates and running times, respectively, of each method. The TS algorithm continues to outperform the other methods. The improvements in blocking rates for the network with $W = 8$ are greater than those for the NSFnet with $W = 4$ and the running times are not as long. A comparison of blocking rates of the TS-based algorithm for the various WAPs is illustrated in Figure 7-10. Random WAP demonstrates the lowest average blocking rate for the network loads tested.

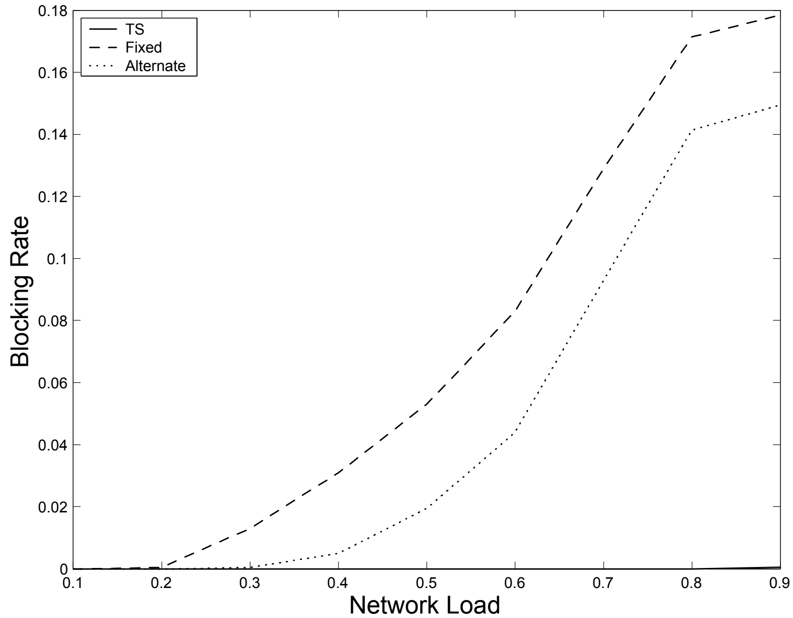

Figure 7-6: Blocking rates for TS-based algorithm using random wavelength assignment for NSFnet with W=8

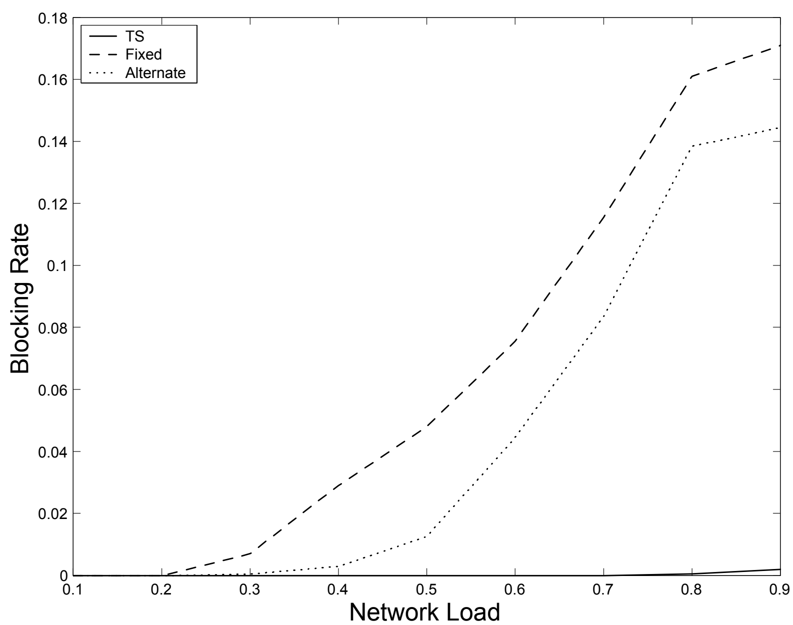

Figure 7-7: Blocking rates for TS-based algorithm using first-fit wavelength assignment for NSFnet with W=8

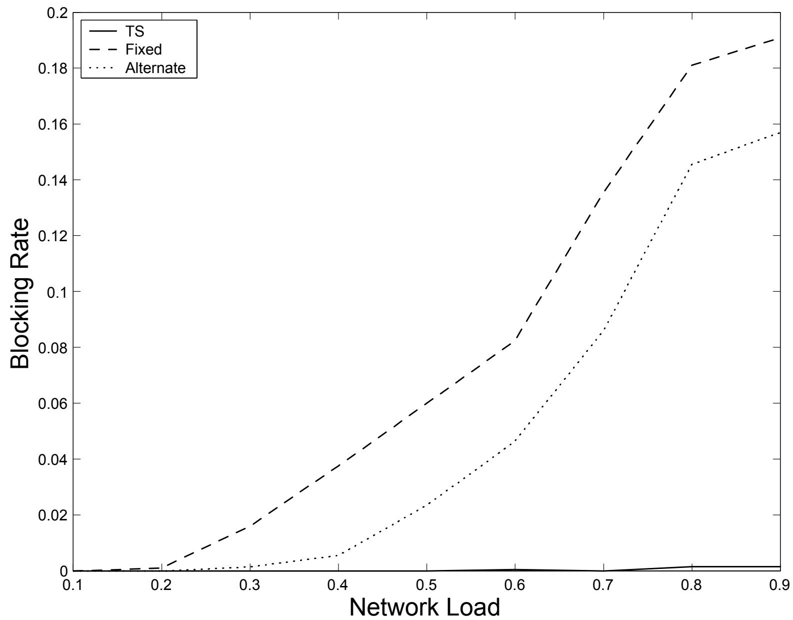

Figure 7-8: Blocking rates for TS-based algorithm using least-used wavelength assignment for NSFnet with W=8

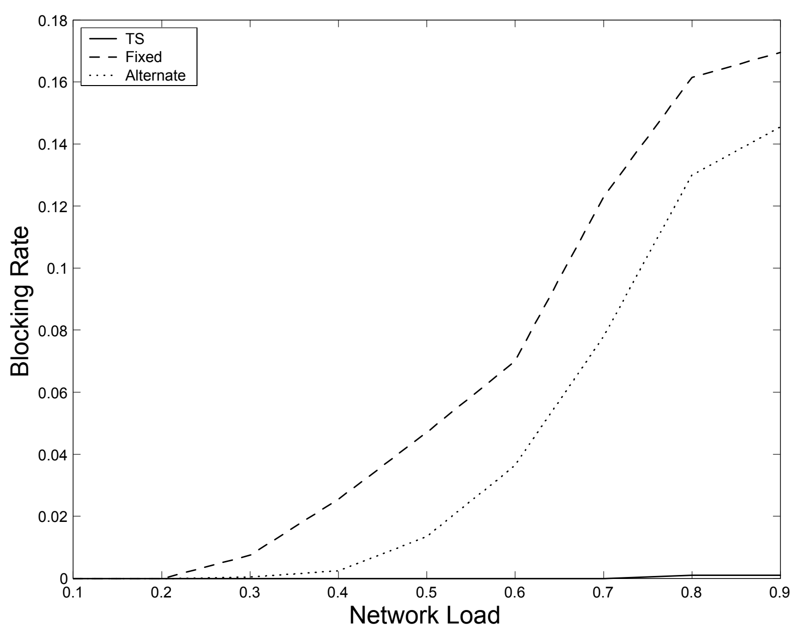

Figure 7-9: Blocking rates for TS-based algorithm using most-used wavelength assignment for NSFnet with W=8

Table 7.3: Average blocking rates for TS, fixed, and alternate routing algorithms for NSFnet with W=8

Routing Method	Wavelength Assignment Policy							
	Random		First-Fit		Least-Used		Most-Used	
	ABR	%	ABR	%	ABR	%	ABR	%
TS	0.0001		0.0003		0.0004		0.0002	
Fixed	0.0733	131900	0.0674	24280	0.0783	20129	0.0671	30200
Alternate	0.0503	90600	0.0474	17080	0.0517	13300	0.0452	20325

Table 7.4: Average running times for TS, fixed, and alternate routing algorithms for NSFnet with W=8

Routing Method	Wavelength Assignment Policy							
	Random		First-Fit		Least-Used		Most-Used	
	ART	%	ART	%	ART	%	ART	%
TS	0.0181		0.0136		0.0210		0.0131	
Fixed	0.0005	-3966	0.0004	-3136	0.0004	-4836	0.0005	-2689
Alternate	0.0006	-2904	0.0006	-2265	0.0006	-3627	0.0007	-1939

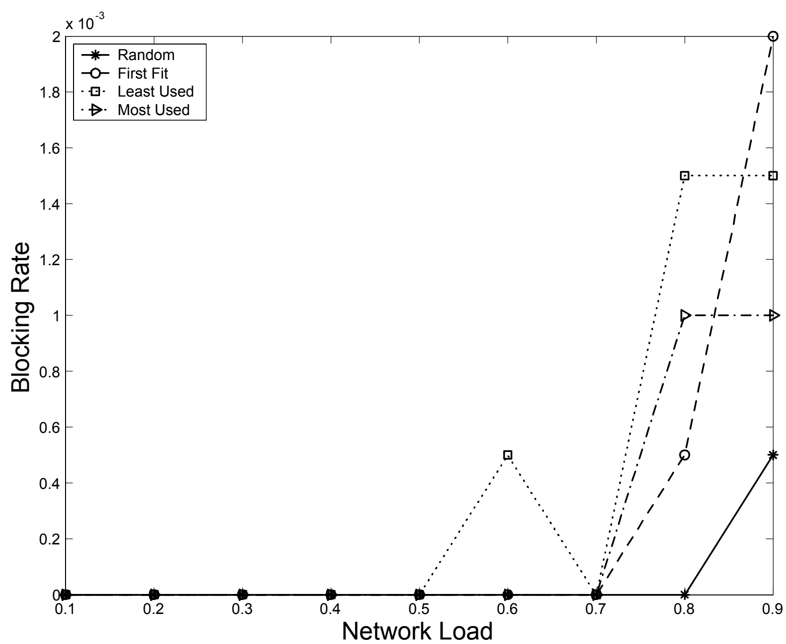

Figure 7-10: Comparison of blocking rates of all wavelength assignment policies for NSFnet with W=8 using TS-based algorithm

7.3.3 4×4 Mesh-Torus Network with W=4

Figures 7-11, 7-12, 7-13, and 7-14 show the performance of our TS-based algorithm on the 4×4 mesh-torus network with 4 wavelengths per link ($W = 4$) using random, first-fit, least-used, and most-used wavelength assignment, respectively. Tables 7.5 and 7.6 show the average blocking rates and running times, respectively, of each method. The TS algorithm outperforms the other methods in all cases but greatly increases the time taken to make a routing decision. A comparison of blocking rates of the TS-based algorithm for the various WAPs is illustrated in Figure 7-15. This plot does not show a dominant WAP but the least-used strategies has the lowest average blocking rate.

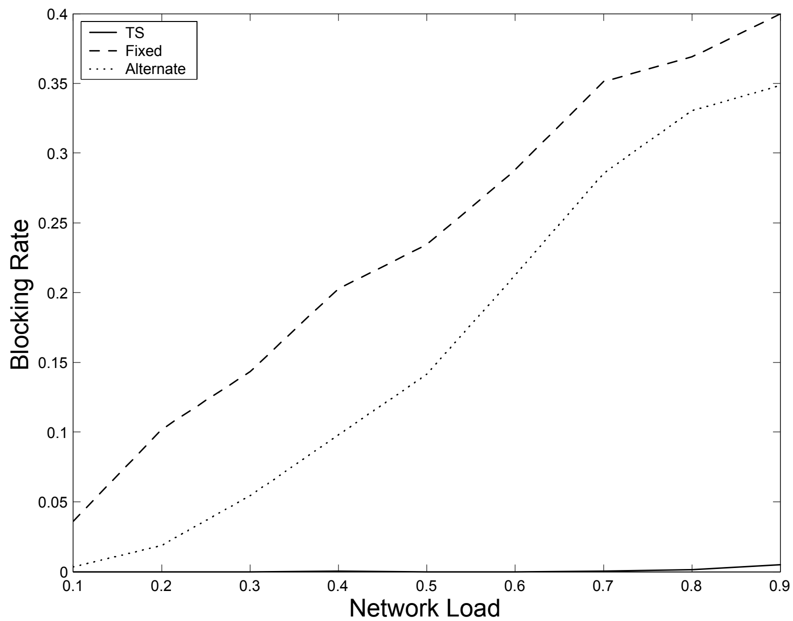

Figure 7-11: Blocking rates for TS-based algorithm using random wavelength assignment for 4×4 mesh-torus network with W=4

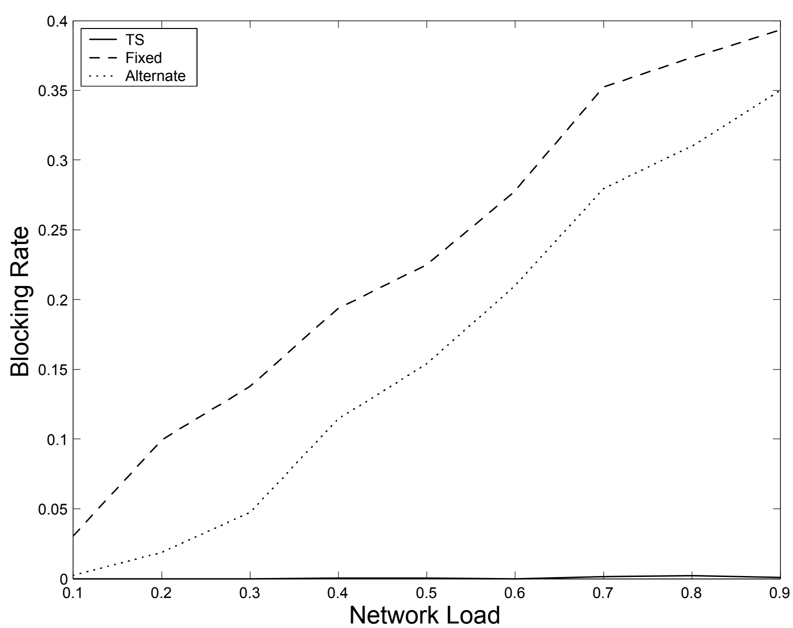

Figure 7-12: Blocking rates for TS-based algorithm using first-fit wavelength assignment for 4×4 mesh-torus network with W=4

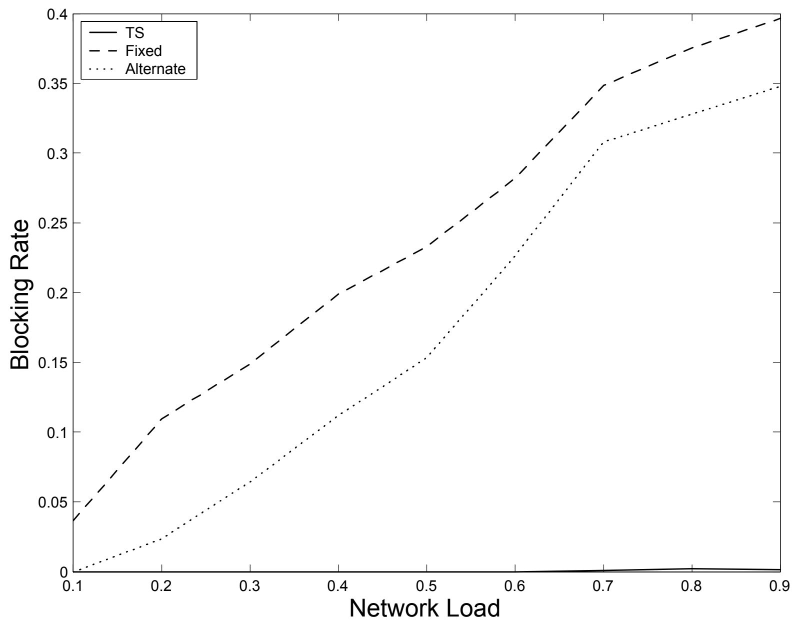

Figure 7-13: Blocking rates for TS-based algorithm using least-used wavelength assignment for 4×4 mesh-torus network with W=4

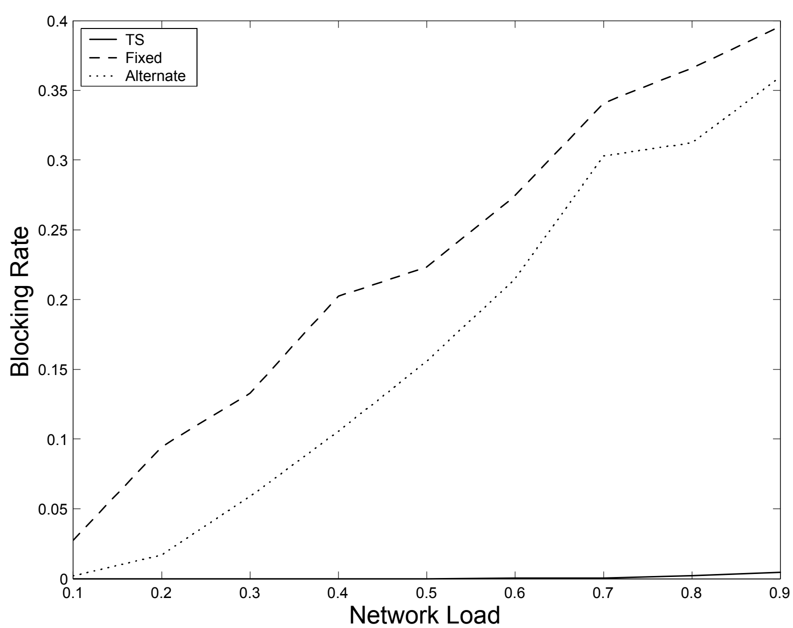

Figure 7-14: Blocking rates for TS-based algorithm using most-used wavelength assignment for 4×4 mesh-torus network with W=4

Table 7.5: Average blocking rates for TS, fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	**%**	**ABR**	**%**	**ABR**	**%**	**ABR**	**%**
TS	0.0008		0.0006		0.0005		0.0008	
Fixed	0.2364	28367	0.2316	37900	0.2366	47322	0.2287	27447
Alternate	0.1659	19913	0.1652	27036	0.1738	34756	0.1699	20393

Table 7.6: Average running times for TS, fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=4

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	**%**	**ART**	**%**	**ART**	**%**	**ART**	**%**
TS	0.0655		0.0571		0.0663		0.0577	
Fixed	0.0004	-15113	0.0004	-13878	0.0004	-15705	0.0004	-12983
Alternate	0.0008	-8542	0.0007	-7780	0.0008	-8406	0.0008	-7114

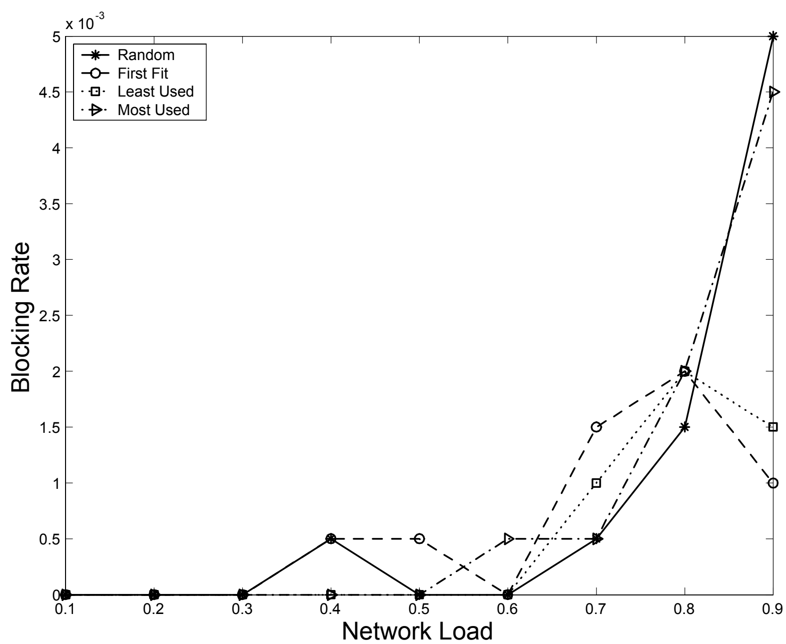

Figure 7-15: Comparison of blocking rates of all wavelength assignment policies for 4×4 mesh-torus network with W=4 using TS-based algorithm

7.3.4 4×4 Mesh-Torus Network with W=8

Figures 7-16, 7-17, 7-18, and 7-19 show the performance of our TS-based algorithm on the 4×4 mesh-torus network with 8 wavelengths per link ($W = 8$) using random, first-fit, least-used, and most-used wavelength assignment, respectively. Tables 7.7 and 7.8 show the average blocking rates and running times, respectively, of each method. The TS algorithm outperforms the other methods in all cases by successfully routing all requests.

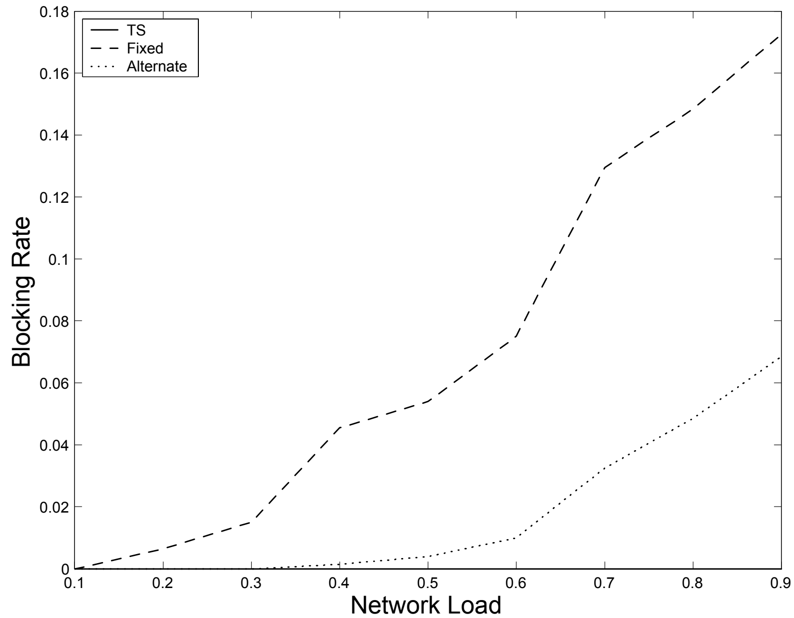

Figure 7-16: Blocking rates for TS-based algorithm using random wavelength assignment for 4×4 mesh-torus network with W=8

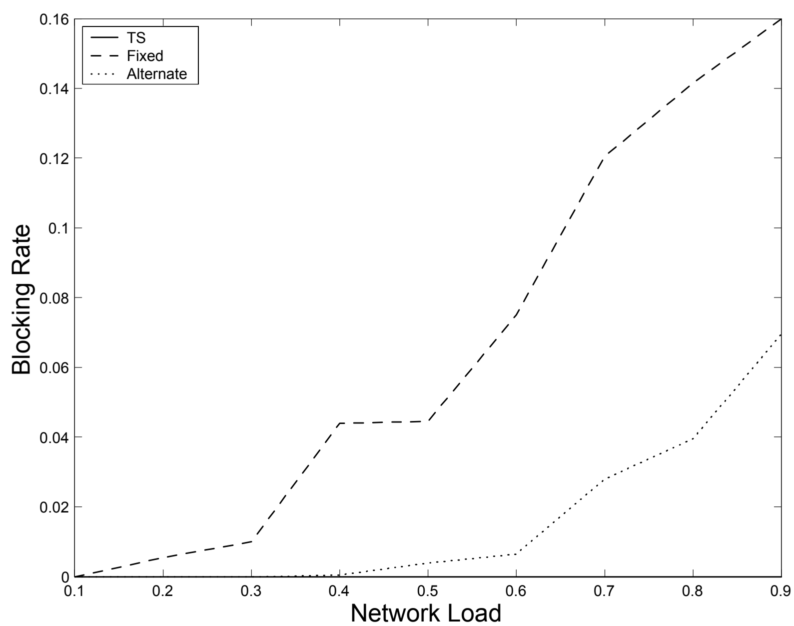

Figure 7-17: Blocking rates for TS-based algorithm using first-fit wavelength assignment for 4×4 mesh-torus network with W=8

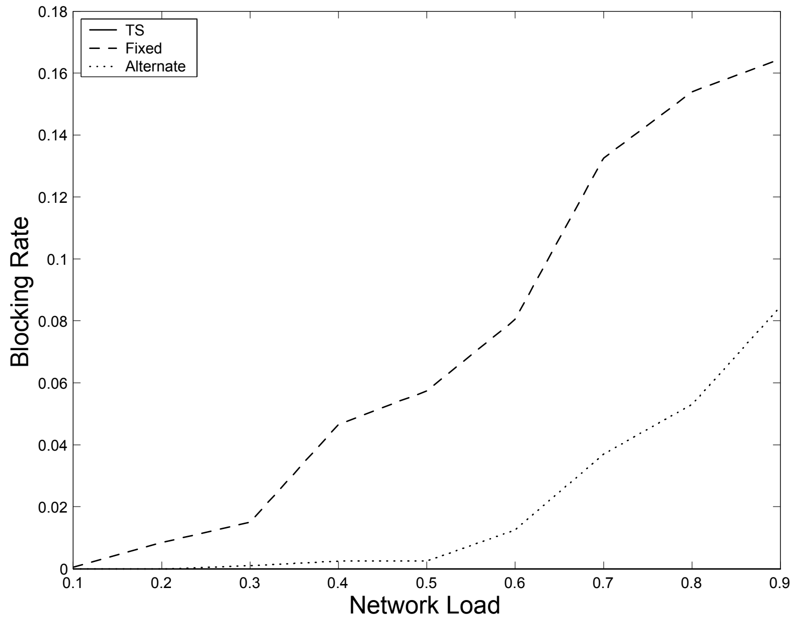

Figure 7-18: Blocking rates for TS-based algorithm using least-used wavelength assignment for 4×4 mesh-torus network with W=8

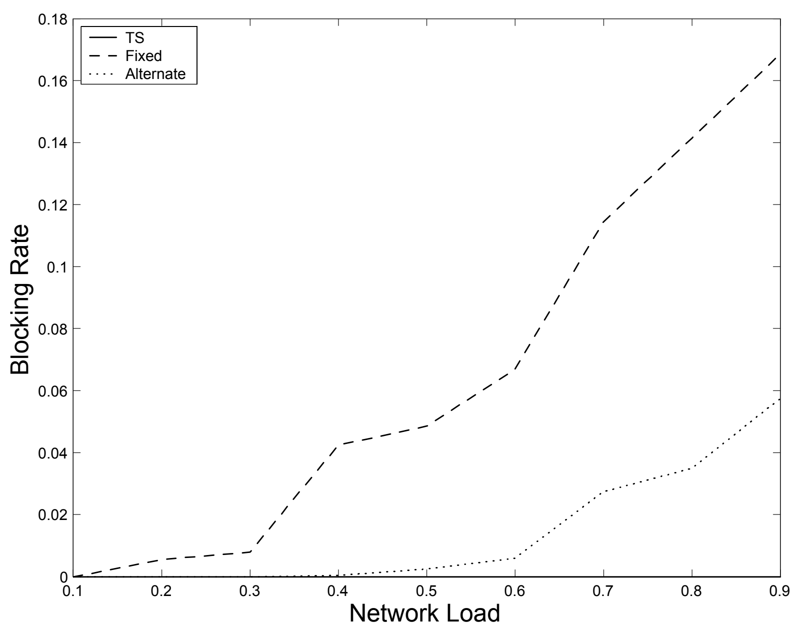

Figure 7-19: Blocking rates for TS-based algorithm using most-used wavelength assignment for 4×4 mesh-torus network with W=8

Table 7.7: Average blocking rates for TS, fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=8

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ABR	**%**	**ABR**	**%**	**ABR**	**%**	**ABR**	**%**
TS	0.0000		0.0000		0.0000		0.0000	
Fixed	0.0718	∞	0.0668	∞	0.0733	∞	0.0662	∞
Alternate	0.0183	∞	0.0164	∞	0.0214	∞	0.0143	∞

Table 7.8: Average running times for TS, fixed, and alternate routing algorithms for 4x4 mesh-torus network with W=8

Routing Method	**Wavelength Assignment Policy**							
	Random		**First-Fit**		**Least-Used**		**Most-Used**	
	ART	**%**	**ART**	**%**	**ART**	**%**	**ART**	**%**
TS	0.0044		0.0036		0.0059		0.0037	
Fixed	0.0004	-1000	0.0004	-831	0.0004	-1367	0.0005	-765
Alternate	0.0006	-741	0.0006	-635	0.0006	-1045	0.0006	-588

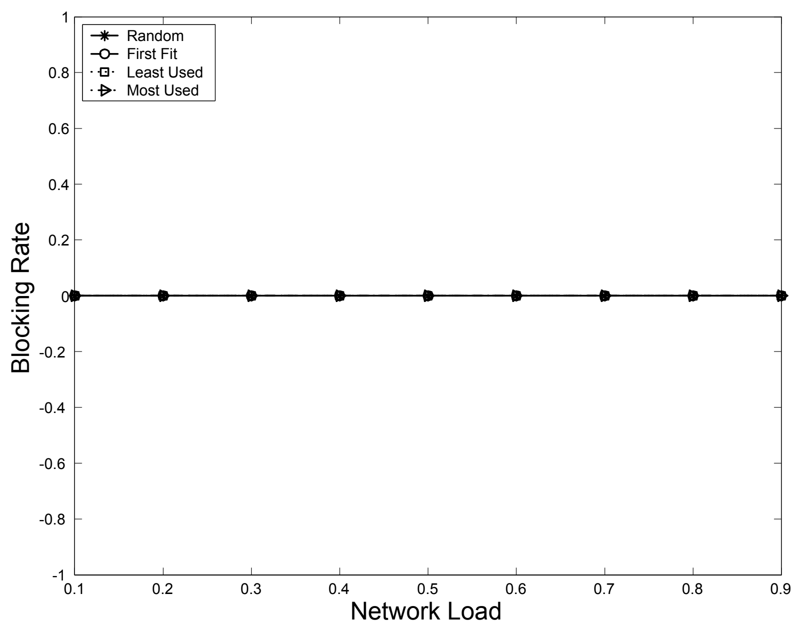

Figure 7-20: Comparison of blocking rates of all wavelength assignment policies for 4×4 mesh-torus network with W=8 using TS-based algorithm

Chapter 8

Comparison of Results

In this chapter, we compare the results of the four RWA methods proposed in our research. For each network, the blocking rate for the WAP with the lowest average blocking rate was plotted for each method. A table of running times for each method is also included. The RWA problem for the mesh-torus network with $W = 8$ may be considered the easiest problem studied in this research because it provides the most wavelengths and physical links on which to route connections. In contrast, the problem for the NSFnet with $W = 4$ may be considered the most complex problem studied in this research.

The FC-based RWA algorithm achieves the lowest average blocking rate for the NSFnet with 4 and 8 wavelengths and for the 4×4 mesh-torus network with 4 wavelengths. The TS-based algorithm is second to the FC-based algorithm in performance, except for the mesh-torus network with $W = 8$ where both algorithms were able to successfully route every request. While both algorithms demonstrate a good performance, the time spent making a routing decision is 13 times as long for the TS-based algorithm when simulating the NSFnet with $W = 4$. The running time for the TS algorithm is 2 times as long for the same network with $W = 8$. This drop is due to the additional wavelengths that make routing connections easier. The FC-based algorithm is 7 times as fast as the TS-based algorithm for the mesh-torus network with $W = 4$. For the mesh-torus network with $W = 8$,the TS-based algorithm demonstrates a shorter running time than the FC algorithm while achieving the same blocking rate.

In [33], Li and Somani study fixed-paths least-congestion routing (FPLC) and neighborhood-information-based routing. As with our research, they perform simulations using the NSFnet and 4×4 mesh-torus network topologies with 8 wavelengths per link. A comparison of the performance of their proposed algorithms with those presented in this study show that our FC-based and TS-based algorithm are able to achieve lower blocking rates for network loads between 0.25 and 0.4. The results for loads less than 0.25 and greater than 0.4 are not presented in [33]. Li's results are very good ranging from 0.00001 to 0.1. However, our FC-based and TS-based algorithms were able to successfully route all of the offered connection requests.

NSFnet with $W = 4$

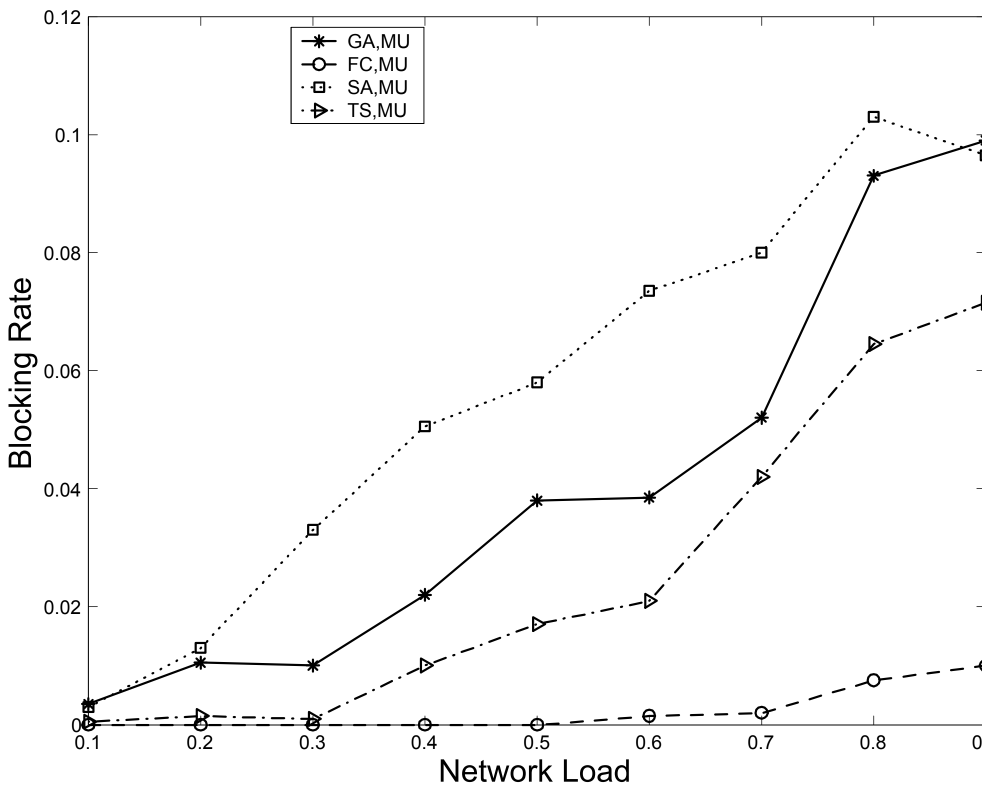

Figure 8-1: Comparison of results for NSFnet with W=4

Table 8.1: Comparison of RWA algorithms for NSFnet with W=4

Performance Measures	**Routing &Wavelength Assignment Policy**			
	GA Most-Used	**FC Most-Used**	**SA Most-Used**	**TS Most-Used**
Average. BR	0.0407	0.0023	0.0567	0.0254
Average. RT per Connection	0.0095	0.0089	0.3549	0.1189

NSFnet with $W = 8$

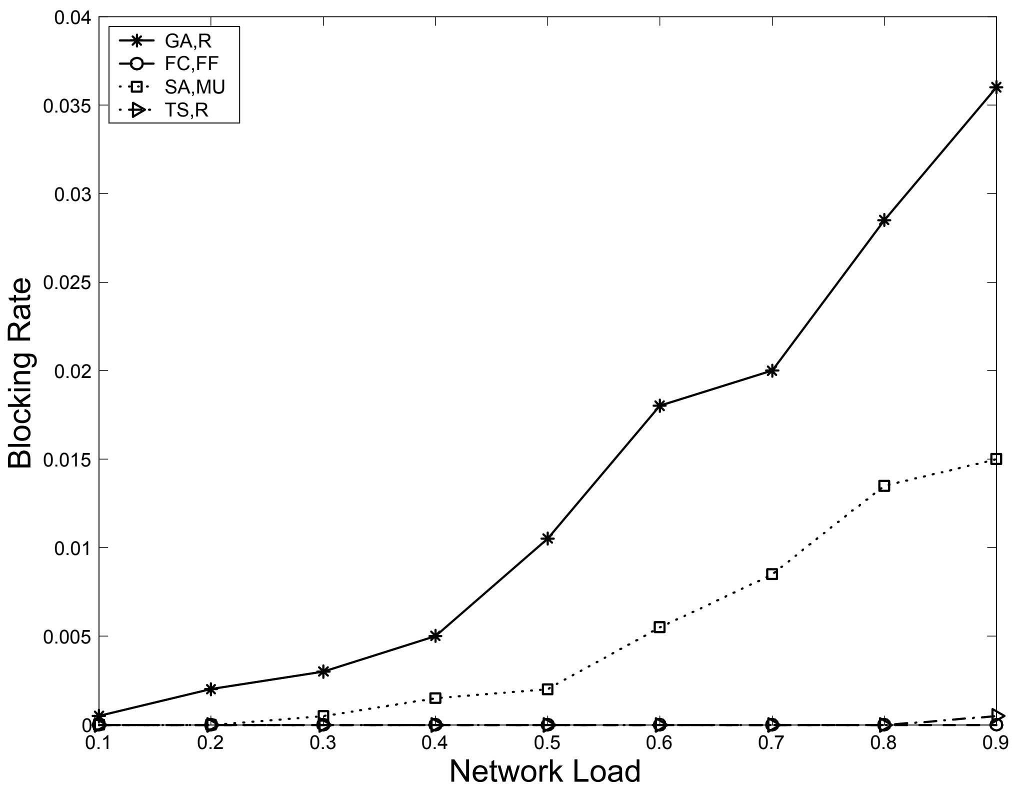

Figure 8-2: Comparison of results for NSFnet with W=8

Table 8.2: Comparison of RWA algorithms for NSFnet with W=8

Performance Measures	**Routing &Wavelength Assignment Policy**			
	GA Random	**FC First-Fit**	**SA Most-Used**	**TS Random**
Average. BR	0.0137	0	0.0052	0.0001
Average. RT per Connection	0.0107	0.0086	0.0510	0.0181

4×4 **Mesh-Torus Network with** $W = 4$

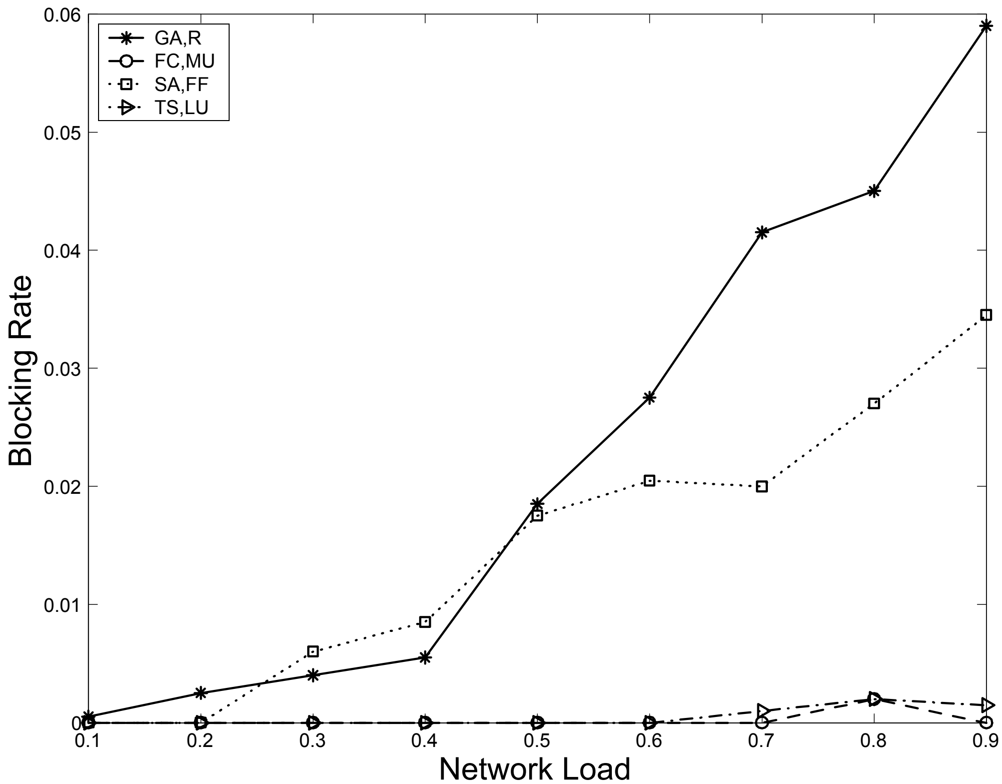

Figure 8-3: Comparison of results for 4×4 mesh-torus network with W=4

Table 8.3: Comparison of RWA algorithms for 4x4 mesh-torus network with W=4

Performance Measures	**Routing &Wavelength Assignment Policy**			
	GA Random	**FC Most-Used**	**SA First-Fit**	**TS Least-Used**
Average. BR	0.0227	0.0002	0.0149	0.0005
Average. RT per Connection	0.0146	0.0089	0.3011	0.0663

4×4 **Mesh-Torus Network with** $W = 8$

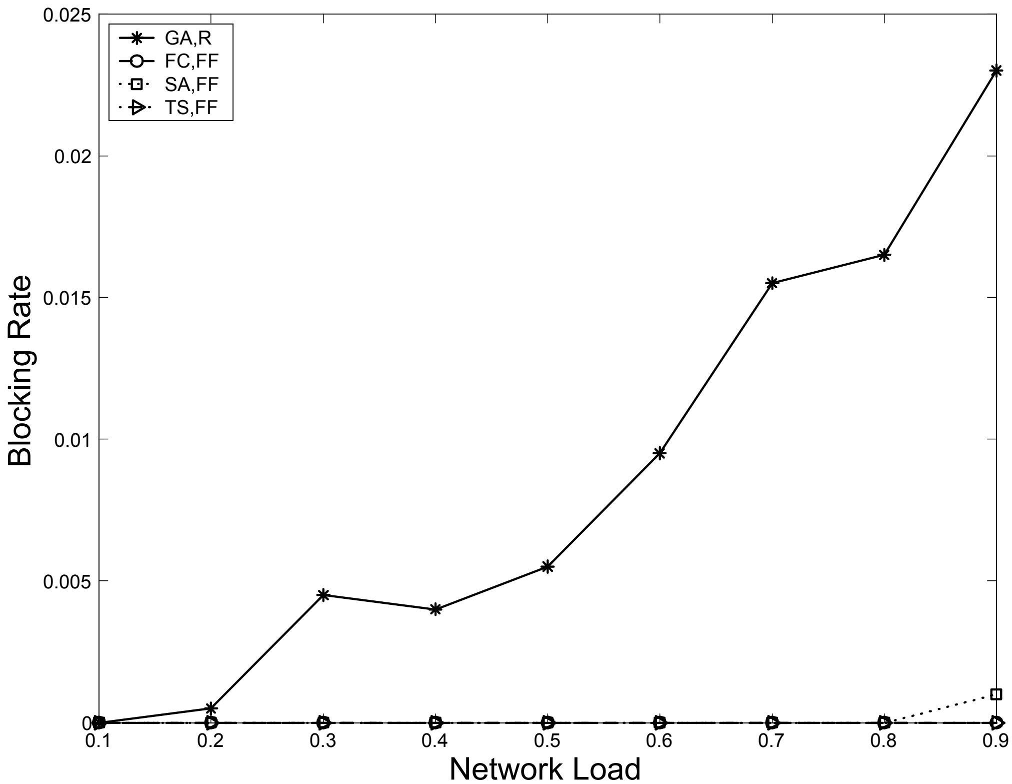

Figure 8-4: Comparison of results for 4×4 mesh-torus network with W=8

Table 8.4: Comparison of RWA algorithms for 4x4 mesh-torus network with W=8

Performance Measures	**Routing &Wavelength Assignment Policy**			
	GA Random	**FC First-Fit**	**SA First-Fit**	**TS First-Fit**
Average. BR	0.0088	0	0.0001	0
Average. RT per Connection	0.0158	0.0086	0.0077	0.0036

In some of the plots of our simulation results, we observe that the data points are not always increasing as one would expect. Intuition tells us that the blocking rate should increase as the traffic load increases. However, the observed behavior does not always follow this trend and the blocking rate may decrease at a higher load as in Figures 5-12 and 5-22 for the FC-based algorithm. We conjecture that this behavior would not occur in simulations with a higher number of repetitions. To verify our conjecture, we simulated the FC-based algorithm for 25 trials as opposed to the 10 trials originally performed. Figures 8-5 and 8-6 show the results of the longer simulations for the NSFnet with $W = 4$ and the 4×4 mesh-torus network with $W = 4$. These figures show that more extensive simulations will yield improved results.

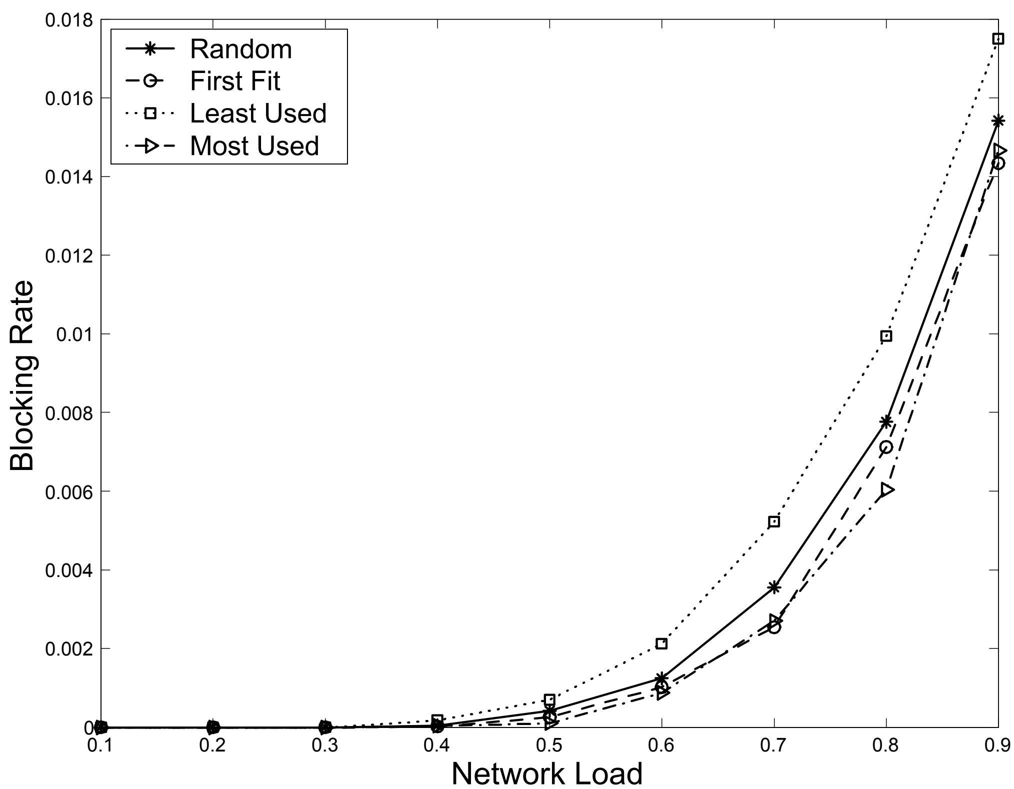

Figure 8-5: Average blocking rates for FC-based algorithm after 25 trials for NSFnet with $W = 4$

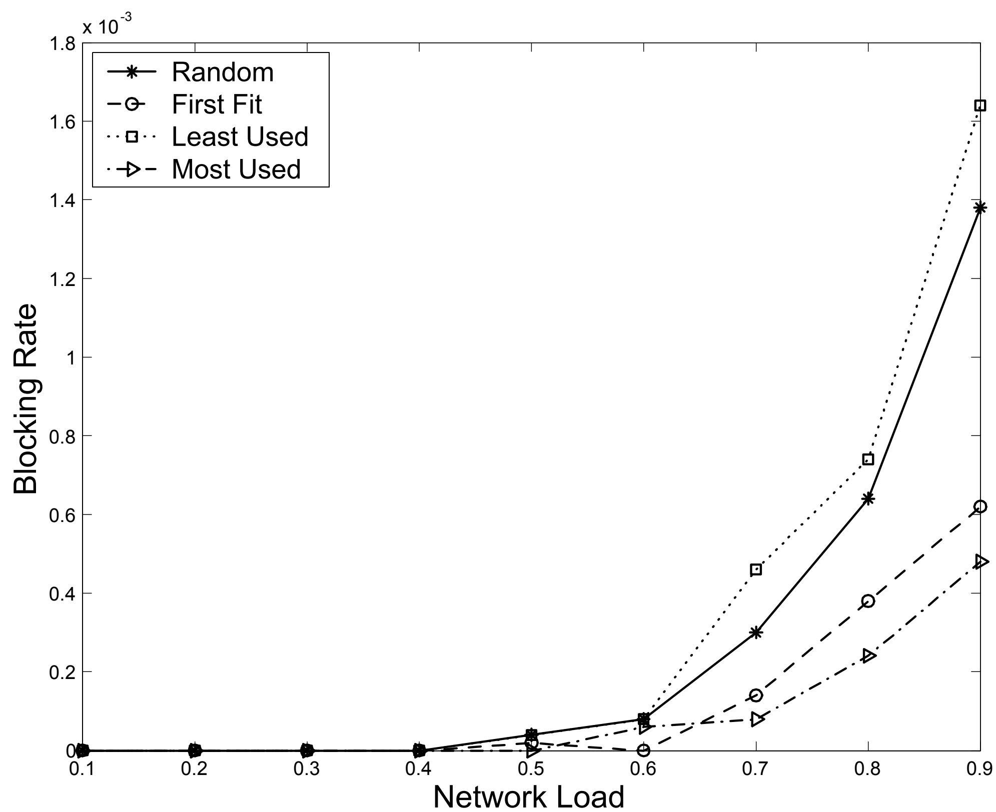

Figure 8-6: Average blocking rates for FC-based algorithm after 25 trials for 4×4 Mesh-torus network with $W = 4$

Chapter 9

Summary and Future Research Directions

In Chapter 1, we introduced wavelength division multiplexed optical networks. We also introduced the routing and wavelength assignment problem (RWA) for wavelength-routed optical networks. This problem has been studied extensively and continues to be an active area of research. However, relatively few studies exist that examine the application of soft computing techniques to the dynamic RWA problem for circuit-switched optical networks. This motivated our study of soft computing approaches to the dynamic RWA problem.

In Chapter 2, we review the existing literature on the RWA problem for both static and dynamic traffic. Many of the routing techniques used for optical networks are borrowed from routing methods used for traditional circuit-switched telephone networks. These include fixed and alternate routing. A routing method is combined with a wavelength assignment policy to form a solution to the RWA problem. The most widely used wavelength assignment policies include random, first-fit, least-used, and most-used wavelength assignment.

In Chapter 3, we describe the methodologies on which our four proposed RWA algorithms are based. Two algorithms are based on the soft computing methods of genetic algorithms (GA) and fuzzy control (FC). The other two methods use heuristic search methods including simulated annealing (SA) and tabu search (TS). Each proposed algorithm relies on global information about the availability of network resources in order to make a routing decision and may employ any wavelength assignment policy. The algorithms are tested on two experimental networks. The 4×4 mesh-torus network has a regular topology with 16 nodes and 32 links and the NSFnet has an irregular topology with 14 nodes and 21 links.

In Chapter 4, an adaptive RWA algorithm based on GA is proposed. The GA-based algorithm employs two mutation operations and a crossover operation to explore the solution space. Wavelength mutation searches the neighborhood of the solution by changing the assigned wavelength while maintaining the same route. Route mutation makes changes to the route by attempting to find an alternative route that includes a node adjacent to one of the nodes of the original route. The original wavelength assignment is kept. In simulations, the GA-based algorithm outperforms fixed routing and alternate routing strategies.

In Chapter 5, an adaptive RWA algorithm based on fuzzy control theory is presented. Linguistic variables, `Route_Length` and `Congestion` are input to a fuzzy controller that utilizes fuzzy if-then rules to determine a rating for each potential route. The route with the highest rating is selected and assigned a wavelength using the specified wavelength assignment method. Simulations performed for dynamic connection requests show that the proposed FC-based algorithm also outperforms fixed and alternate routing algorithms.

In Chapters 6 and 7, we present adaptive RWA algorithms based on SA and TS, respectively. A connection request is routed on a shortest path if one is available. However, in cases where no wavelength is available, SA or TS is employed to search for an alternate route and wavelength assignment. Both methods demonstrate good performance in comparison to the fixed and alternate routing strategies. However, running times for these algorithms are longer in most cases.

In Chapter 8, we compare the performance of the GA, FC, SA, and TS based algorithms. The results in this chapter show that fuzzy control is a promising approach to adaptive routing and wavelength assignment in wavelength-routed networks. The FC-based algorithm showed the best performance (with respect to blocking rate) except for the mesh-torus network with 8 wavelengths per link. The TS-based algorithm also demonstrated good results However, the FC-based algorithm was much faster in most scenarios.

The results from the algorithms presented in this paper show that soft computing can provide promising approaches to adaptive routing and wavelength assignment in wavelength-routed networks. Our research could be extended to examine the following areas.

- The RWA problem for networks with packet-switched traffic.
- The RWA problem for other network architectures such as networks with wavelength converters.
- The use of soft computing methods to reconfigure wavelength-routed networks when a failure has occurred.
- The use of hybrid algorithms that combine two or more soft computing and/or heuristic methods.

Bibliography

[1] Aarts, E., and Korst, J., *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*, John Wiley & Sons, Chinchester, Eng., 1989.

[2] Alanyali, M. and Ayanoglu, E., "Provisioning Algorithms for WDM Optical Networks," *IEEE/ACM Transactions on Networking*, Vol.7 (1999), 767-778.

[3] Al-Fuqaha, A., Chaudhry, G.M., Guizani, M., and Labrador, M.A., "Routing Framework for All-Optical DWDM Metro and Long-Haul Transport Networks with Sparse Wavelength Conversion Capabilities," *IEEE Journal on Selected Areas in Communications,* Vol. 22 (2004), 1443-1459.

[4] Ali, M., Ramamurthy, B., and Deogun,J.S., "Routing Algorithms for All-Optical Networks with Power Considerations: the Unicast Case," *Proceedings of the Eighth International Conference on Computer Communications and Networks* (1999), 237-241.

[5] Anderson, C.A., Fraughnaugh, K., Parker, M., and Ryan, J., "Path Assignment for Call Routing: An Application of Tabu Search," *Annals of Operations Research,* Vol. 41 (1993), 301-312.

[6] Banerjee, D. and Mukherjee, B., "A Practical Approach for Routing and Wavelength Assignment in Large Wavelength-Routed Optical Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 14 (1996), 903-908.

[7] Banerjee, N. and Sharan, S., "An Evolutionary Algorithm for Solving the Single Objective Static Routing and Wavelength Assignment Problem in WDM Networks," *Proceed-*

ings of the International Conference on Intelligent Sensing and Information Processing (2004), 13-18.

[8] Baroni, S. and Bayvel, P., "Wavelength Requirements in Arbitrarily Connected Wavelength-Routed Optical Networks," *IEEE/OSA Journal of Lightwave Technology*, Vol.15 (1997), 242-251.

[9] Birman, A., "Computing Approximate Blocking Probabilities for a Class of All-Optical Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 14 (1996), 852-857.

[10] Birman, A., and Kershenbaum, A., "Routing and Wavelength Assignment Methods in Single-Hop All-Optical Networks with Blocking," *Proc. INFOCOM '95*, Boston, MA, April, 1995, 431-438.

[11] Branke, J., *Evolutionary Optimization in Dynamic Environments*, Kluwer, Boston, 2002.

[12] Cĕrny, V., "A Thermodynamical Approach to the Travelling Salesman Problem," *Journal of Optimization Theory and Applications*, Vol. 45 (1985), 41-55.

[13] Chlamtac, I., Ganz, A. and Karmi, G., "Lightpath Communications: An Approach to High Bandwidth Optical WAN's," *IEEE Transactions on Communications*, Vol. 40 (1992), 1171-1182.

[14] Corne, D., Oates, M., and Smith, G. (eds), *Telecommunications Optimization: Heuristic and Adaptive Techniques*, John Wiley & Sons, Chinchester, Eng., 2000.

[15] Driankov, D., Hellendoorn, H., and Reinfrank, M., *An Introduction to Fuzzy Control*, Springer, Berlin, 1996.

[16] Gen, M. and Cheng, R., *Genetic Algorithms and Engineering Optimization*, Wiley Interscience, New York, 2000.

[17] Girard, A., *Routing and Dimensioning in Circuit-Switched Networks*, Addison-Wesley, Reading, MA.,1990.

[18] Glover, F., "Future Paths for Integer Programming and Links to Artificial Intelligence," *Computer and Operations Research,* Vol. 13 (1986), 533-549.

[19] Glover, F. and Laguna, M., Tabu Search, Kluwer Academic Publishers, Boston, 1997.

[20] Glover, F., Taillard, E., and de Werra, D., "A User's Guide to Tabu Search," *Annals of Operations Research,* Vol. 41 (1993), 3-28.

[21] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA., 1989.

[22] Green, P.E., "The Future of Fiber-Optic Computer Networks," *IEEE Computer*, Vol. 24, No. 9 (1991), 78-87.

[23] Green, P.E., "Optical Networking Update," *IEEE Journal on Selected Areas in Communications*, Vol.14 (1996), 764-779.

[24] Harai, H., Murata, M., and Miyahara, H., "Performance of Alternate Routing Methods in All-Optical Switching Networks," *Proc. INFOCOM '97*, Tokyo, Japan, May, 1997, 516-524.

[25] Holland, J., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI., 1975.

[26] Hwang,I-S., Huang,I-F., and Yu, S-C., "Dynamic RWA scheme using fuzzy logic control (FLC RWA) on IP with GMPLS over DWDM networks," 2004 IEEE International Conference on Networking, Sensing and Control, Vol 2 (2004), 1049 - 1056.

[27] Inkret, R.,Mikac,B., and Podnar, I., "A Heuristic Approach to Wavelength Assignment in All-Optical Networks," *Proceedings of the Ninth Mediterranean Electrotechnical Conference*, Vol. 2 (1998), 759-763.

[28] Johnson, D.S., Aragon, C.R., McGeoch, L.A., and Schevon, C., "Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning," *Operations Research*, Vol. 36, (1989), 865-892.

[29] Keshav, S., *An Engineering Approach to Computer Networking - ATM Networks, the Internet, and the Telephone Network*, Addison-Wesley, Reading, MA., 1997.

[30] Khyda, D.O., Chamberland, S. and Pierre, S., "Improvement of Routing and Wavelength Assignment in WDM Networks Using Tabu Search," *IEEE Canadian Conference on Electrical and Computer Engineering,* Vol. 2 (2003), 765-768.

[31] Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., "Optimization by Simulated Annealing," *Science*, Vol. 220 (1983), 671-680.

[32] Kuri, J., Puech, N., Gagnaire, M., Dotaro, E., and Douville, R., "Routing and Wavelength Assignment of Scheduled Lightpath Demands," *IEEE Journal on Selected Areas in Communications,* Vol. 21 (2003), 1231-1240.

[33] Li, L. and Somani, A.K., "Dynamic Wavelength Routing Using Congestion and Neighborhood Information," *IEEE/ACM Transactions on Networking*, Vol. 7 (1999), 779-786.

[34] Lima, M.A.C., Cesar, A.C., and Araujo, A.F.R., "Optical Network Optimization with Transmission Impairments Based on Genetic Algorithm," *Proceedings of the SBNO/IEEE MTT-S International Microwave and Optoelectronics Conference*, Vol. 1 (2003), 361-365.

[35] Ming-Kang Liu, M., *Principles and Applications of Optical Communications*, Irwin, Chicago, 1996.

[36] Liu, A., Jaekel, A., and Bandyopadhyay, S., "A Genetic Algorithm for Optimization of Logical Topologies in Optical Networks," *Proceedings of the International Parallel and Distributed Processing Symposium* (2002), 202-209.

[37] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E., "Equation of State Calculation by Fast Computing Machines," *Journal of Chemical Physics*, Vol. 21 (1953), 1087-1091.

[38] Michalewicz, Z., *Genetic Algorithms + Data Structure = Evolution Programs*, Springer-Verlag, Berlin, 1996.

[39] Mikac, B., Inkret, R., and Podnar, I., "An Application of Heuristic Search Techniques in Telecommunication System Design," *Proceedings of the Second International Conference on Knowledge-Based Intelligent Electronic Systems*, Vol. 1 (1998). 311-317.

[40] Mokhtar, A. and Azizoglu, M., "Adaptive Techniques for Routing and Wavelength Assignment in All-Optical Networks," *IEEE 39th Midwest Symposium on Circuits and Systems,* Vol.3 *(1996), 1195-1198.*

[41] Mokhtar, A. and Azizoglu, M., "Adaptive Wavelength Routing in All-Optical Networks," *IEEE/ACM Transactions on Networking*, Vol. 6 (1998), 197-206.

[42] Mukherjee, B., "WDM-Based Local Lightwave Networks Part I: Single-Hop Systems," *IEEE Network*, Vol. 6 (1992), 12-27.

[43] Mukherjee, B., *Optical Communication Networks*, McGraw-Hill, New York, 1997.

[44] Munetomo, M., Takai, Y., and Yoshiharu, S., "An Adaptive Network Routing Algorithm Employing Path Genetic Operators," *Proceedings of the Seventh International Conference on Genetic Algorithms* (1997), 643-649.

[45] Qin, H., Liu, Z., Zhang, S., Wen, A., "Routing and Wavelength Assignment Based on Genetic Algorithm," *IEEE Communication Letters*, Vol. 6 (2002), 455-457.

[46] Ramaswami, R. "Multiwavelength Lightwave Networks for Computer Communication," *IEEE Communications Magazine*, February 1993, 78-99.

[47] Ramaswami, R. and Sivarajan, K., "Routing and Wavelength Assignment in All-Optical Networks," *IEEE/ACM Transactions on Networking*, Vol. 3 (1995), 489-500.

[48] Ramaswami, R. and Sivarajan, K., *Optical Networks: A Practical Perspective*, Morgan Kaufmann, San Francisco, 1998.

[49] Reeves, C.(ed.), *Modern Heuristic Techniques for Combinatorial Problems*, Halsted Press, New York, 1993.

[50] Rodriguez-Dagnino, R.M. , Lopez-Caudana, E.O., Martinez-Alfaro, H., and Gonzalez-Velarde, J.L., "Simulated Annealing and Stochastic Ruler Algorithms for Wavelength

Assignment Planning in WDM Optical Networks," *IEEE SMC '99 Conference Proceedings,* Vol. 6 (1999), 1015-1020.

[51] Saha, D., Purkayastha, M., and Mukherjee, A., "An Approach to Wide Area WDM Optical Network Design Using Genetic Algorithm," *Computer Communications*, Vol. 22 (1999), 156-172.

[52] Sinclair, M.C., "Minimum Network Wavelength Requirement Design Using a Genetic-Algorithm/Heuristic Hybrid," *Electronic Letters*, Vol. 34 (1998), 388-389.

[53] Sinclair, M.C., "NoMad: an Optical Network Optimization, Modelling and Design Toolset," *IEE Colloquium on Multiwavelength Optical Networks: Devices, Systems and Network Implementations* (1998), 6/1-6/6.

[54] Sinclair, M.C., "Operator-Probability Adaptation in a Genetic-Algorithm/Heuristic Hybrid for Optical Network Wavelength Allocation," *IEEE International Conference on Evolutionary Computation* (1998), 840-845.

[55] Sinclair, M.C., "Minimum Cost Wavelength-Path Routing and Wavelength Allocation Using a Genetic-Algorithm/Heuristic Hybrid Approach," *IEE Proceedings on Communications*, Vol. 146 (1999), 1-7.

[56] Terano, T., Asai, K., and Sugeno, M., *Fuzzy Systems Theory and Its Applications*, Academic Press, Boston, 1987.

[57] Wang, Y., Li, L., and Wang, S., "A New Algorithm of Design Protection for Wavelength-Routed Networks and Efficient Wavelength Converter Placement," *IEEE International Conference on Communications,* Vol. 6 (2001), 1807-1811.

[58] Zadeh, L.A., "Fuzzy Sets," *Information and Control*, Vol. 8 (1965), 338-353.

[59] Xu, J., Chiu, S.Y., and Glover, F., "Tabu Search for Dynamic Routing Communications Network Design," *Telecommunication Systems*, Vol. 8 (1997), 55-77.

[60] Zadeh, L.A., "Calculus of Fuzzy Restriction," in Zadeh, Tanake, et al., eds., *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, Academic Press, New York, 1975.

[61] Zadeh, L.A., "Fuzzy Logic," *Computer*, Vol. 21, Issue 4 (1988), 83-93.

[62] Zadeh, L.A., "Soft Computing and Fuzzy Logic," *IEEE Software*, Vol. 11, Issue: 6 (1994), 48-56.

[63] Zadeh, L.A., "Fuzzy Logic = Computing with Words," *IEEE Transactions on Fuzzy Systems*, Vol.4 (1996), 103-111.

[64] Zhang, A. and Acampora, A., "A Heuristic Wavelength Assignment Algorithm For Multihop WDM Networks with Wavelength Routing and Wavelength Re-Use," *IEEE/ACM Transactions on Networking*, Vol. 3 (1995), 281-288.

[65] Zhong, L. and Ramamurthy, B., "Optimization of Amplifier Placements in Switch-based Optical Networks," *IEEE International Conference on Communications*, Vol. 1 (2001), 224-228.

[66] Zimmermann, H.-J., *Fuzzy Set Theory-and Its Applications*, Kluwer, Boston, 1996.