An Evolutionary Algorithm for Survivable Virtual Topology Mapping in Optical WDM Networks

Fatma Corut Ergin, Ayşegül Yayımlı, and Şima Uyar

Istanbul Technical University
fcorut@eng.marmara.edu.tr,
{gencata,etaner}@itu.edu.tr

Abstract. The high capacity of fibers used in optical networks, can be divided into many channels, using the WDM technology. Any damage to a fiber causes all the channels routed through this link to be broken, which may result in a serious amount of data loss. As a solution to this problem, the virtual layer can be mapped onto the physical topology, such that, a failure on any physical link does not disconnect the virtual topology. This is known as the survivable virtual topology mapping problem. In this study, our aim is to design an efficient evolutionary algorithm to find a survivable mapping of a given virtual topology while minimizing the resource usage. We develop and experiment with different evolutionary algorithm components. As a result, we propose a suitable evolutionary algorithm and show that it can be successfully used for this problem. Overall, the results are promising and promote further study.

Keywords: Optical networks, WDM, survivable virtual topology design, evolutionary algorithms, constraint optimization.

1 Introduction

Today, optical networking [1] is the most effective technology to meet the high bandwidth network demand. The high capacity of fiber used in optical networks, can be divided into hundreds of different transmission channels, using the WDM (wavelength division multiplexing) technology. Each of these channels work on different wavelengths and can be associated with a different optical connection. The upper layers (IP, Ethernet, etc.) can transmit data using these optical connections. This architecture is known as IP-over-WDM, or Ethernet-over-WDM.

End-to-end optical connections used by the packet layer (IP, Ethernet, etc.) are called lightpaths. Since the fibers on the physical topology allow traffic flow on different wavelengths, more than one lightpath, each operating on different wavelengths, can be routed on a single fiber. All the lightpaths set up on the network form the virtual topology (VT). Given the physical parameters of the network (physical topology, optical transceivers on the nodes, wavelength numbers on the fibers, etc.) and the mean traffic rates between nodes, the problem of designing the lightpaths to be set up on the physical topology is known as

M. Giacobini et al. (Eds.): EvoWorkshops 2009, LNCS 5484, pp. 31–40, 2009.

the VT design problem. VT mapping problem, which is a subproblem of VT design, is to find a proper route for each lightpath of the given VT and to assign wavelengths to these lightpaths.

Any damage to a physical link (fiber) on the network causes all the lightpaths routed through this link to be broken. Since huge amount of data (40 Gb/s) can be transmitted over each of these lightpaths, a fiber damage may result in a serious amount of data loss. Two different approaches can be used to avoid data loss: 1. Survivable design of the physical layer, 2. Survivable design of the virtual layer [1]. The first approach is the problem of designing a backup link/path for each link/path of the virtual layer. The second approach is the problem of designing the virtual layer such that the virtual layer remains connected in the event of a single or multiple link failure. While the first approach provides faster recovery for time-critical applications (such as, IP phone, telemedicine) by reserving more resources; the second approach, i.e. the survivable VT design, which has attracted a lot of attention in recent years, aims to protect data communication using less resources. In this study, our main aim is to design an efficient evolutionary algorithm (EA) to find a survivable mapping of a given VT while minimizing the resource usage.

To illustrate VT mapping problem, assume that we have a physical network topology as in Figure 1.a and a virtual network topology to be routed on this physical topology as in Figrue 1.b. If we route this VT as in Figure 1.c we obtain a survivable mapping, that is, a failure on any physical link does not disconnect the VT. However, if the routing of only one lightpath is changed, e.g., as in Figure 1.d, we end up with an unsurvivable mapping. In this case, if a failure occurs on the physical link between nodes 4 and 5, the nodes connected with lightpaths b and g will not be able to find an alternative path to communicate.

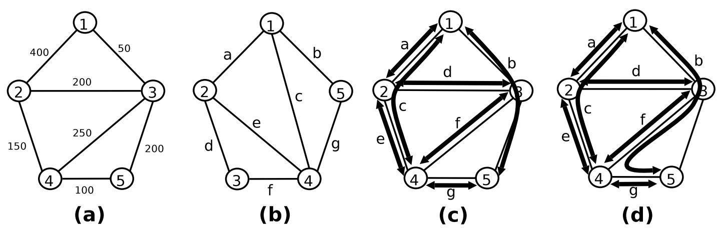

Fig. 1. a. Physical Topology, b. Virtual Topology, c. Survivable Mapping, d. Unsurvivable Mapping

The VT mapping problem is known to be NP-complete [2]. Because of its complexity, for real-life sized networks, it is not possible to solve the problem optimally in an acceptable amount of time using classical optimization techniques. Therefore, heuristic approaches should be used. In this study, we chose EAs due to their successful applications on NP-complete problems. We proposed new mutation operators and fitness evaluation methods and compared their performance to design an effective EA specific to this problem.

The structure of the search space [3] plays an important role on the performance and design of EAs. In order to support our effective EA design, we examined the search space structure of this problem through a correlation length analysis of our different mutation operators and fitness evaluation methods. We performed a comprehensive experimental study for different physical and virtual network topologies. As a result of these experiments, we show that EAs can be successfully used in solving VT mapping problem, and we recommend an appropriate selection of EA components (solution representation, mutation operator, and fitness evaluation method).

2 Problem Definition and Related Literature

In this work, given the physical and the virtual network topologies, our aim is to find a survivable mapping of the VT. Survivable mapping of VT is to find a route for each lightpath, such that in case of a single physical link failure the VT remains connected.

2.1 Formal Problem Definition

The physical topology is composed of a set of nodes $N = \{1..N\}$ and a set of edges E, where (i, j) is in E if there is a link between nodes i and j. Each link has a capacity of W wavelengths. The VT, on the other hand, has a set of virtual nodes N_L, which is a subset of N, and virtual edges (lightpaths) E_L, where an edge (s, t) exists in E_L if both node s and node t are in N_L and there is a lightpath between them.

An ILP formulation of survivable lightpath routing of a VT on top of a given physical topology is given in [2]. Based on this formulation, a number of different objective functions can be considered for the problem of survivable mapping. The simplest objective is to minimize the number of physical links used. Another objective is to minimize the total number of wavelength-links used in the whole physical topology. A wavelength-link is defined as a wavelength used on a physical link. Our choice as the objective is the latter one, since it gives a better idea of the actual resource usage.

The aim of lightpath routing is to find a set of physical links that connect the nodes of the lightpaths. Our objective is to minimize the total number of wavelength-links used in the whole physical topology, subject to two constraints.

a) Survivability constraint: The survivability constraint states that for all proper cuts of the VT, the number of cut-set links flowing on any given physical link is less than the size of the cut-set. This means that all the lightpaths of a cut-set cannot be routed using the same physical link.

b) Capacity constraints: This constraint ensures that the number of wavelengths on a physical link does not exceed its capacity W.

2.2 Related Literature

The survivable VT mapping problem was first addressed as Design Protection [4] in the literature. In this first study, tabu search was used to find the minimum number of source-destination pairs that become disconnected in the event of a physical link failure. Nucci et. al. also used tabu search to solve the survivable VT design problem [5]. The constraints in this study include transmitter and receiver constraints as well as wavelength capacity constraints. In other heuristic approaches to VT mapping Ducatelle et. al. [6] studied the problem using a local search algorithm, while Kurant and Thiran [7] used an algorithm that divides the survivable mapping problem into subproblems. There are a few studies on VT mapping [8] and design [9] using EA, however, the survivability is not considered in any of them.

Modiano and Narula-Tam used ILP (Integer Linear Programming) to solve the VT mapping problem [2]. They added the survivability constraint in the problem formulation, such that, no physical link is shared by all virtual links belonging to a cut-set of the VT graph. However, they did not consider the capacity constraint. Their objective was to minimize the number of wavelengths used. For the cases when ILP cannot find an optimum solution in a reasonable amount of time due to the problem size, Modiano et. al. proposed two relaxations to ILP, which consider only small-sized cut-sets. These relaxations reduce the problem size; however, they may lead to suboptimal solutions. In order to overcome the long execution time problem in ILP formulation, Todimala and Ramamurthy proposed a new ILP formulation and they solved the problem for networks of up to 24 nodes [10]. In [10], besides the physical network and the virtual network topologies, the shared risk link groups should be known in advance.

3 Proposed Evolutionary Algorithm

Designing a solution representation that is well-suited to the problem is crucial in EA performance. VT problem can be seen as a search for the best routing of lightpaths through physical links. Therefore, we use a solution encoding inspired from [8]. For this encoding, first, the k-shortest paths for each lightpath are determined. Then, a solution candidate is represented as an integer string of length l, where l is the number of lightpaths in the VT. Each gene gives the index of the shortest path for the corresponding lightpath. Genes can take on values between $[1..k]$ where k is the predefined number of shortest paths for the lightpaths.

A steady-state EA with duplicate elimination is used where a new individual is generated and inserted into the population at each iteration. After a random initial population generation, the EA operators are applied to the solutions until a predefined number of fitness evaluations are executed. Mating pairs are selected through binary tournament selection. The two selected individuals undergo reproduction. Reproduction consists of uniform crossover and problem specific mutation operators. The offspring replaces the worst individual in the population, if its fitness is better.

3.1 Mutation Operators

We define two different mutation operators. The first one is a simple random-reset mutation, called gene mutation. In this type of mutation, the value of a gene is randomly reset to another value within the allowed range. The second is a problem-specific mutation operator, called path mutation. It considers the physical link similarities between the shortest paths of each lightpath. If mutation occurs on a gene, its current value is replaced by the index of the least similar shortest path for the corresponding lightpath, similarity being defined as the number of common physical links. This mutation operator aims to preserve diversity in the population.

3.2 Fitness Evaluation Methods

Our objective is to minimize the total cost of resources used throughout the network. This cost is evaluated in two different ways: (1) by considering the actual lengths of the physical links (link cost), and (2) by counting the number of physical links used (hop count).

The constraints for the problem, i.e. the survivability and the capacity constraints, are explained in section 2.1. Violations of these constraints are included as penalties to the fitness function. We define three different fitness functions based on three different survivability constraint violation handling approaches.

In order to determine if the solution is survivable or not, each physical link is deleted from the physical network one by one. If any of the lightpaths becomes disconnected in the event of a broken physical link, the solution is taken as unsurvivable. The penalty for an unsurvivable solution is determined in three different ways:

1. The total number of physical links, whose failure results in the network unsurvivability.
2. The sum of the total number of lightpaths that become disconnected in the event of each physical link failure [4,6].
3. The maximum of the total number of lightpaths that become disconnected in the event of each physical link failure [4].

Each of the fitness evaluation methods we define, f_1, f_2, and f_3, use the penalty calculation methods above, respectively. In the first fitness evaluation method (f_1), the connectivity of the graph is checked for the failure of each physical link, which needs an algorithmic complexity of $O(e.n^3)$. On the other hand, for the second and the third fitness evaluation methods (f_2 and f_3), a shortest path algorithm is applied for the failure of each physical link, which means $O(l.(e{+}n).logn)$ algorithmic complexity. Here, e is the number of physical links, n is the number of nodes, and l is the number of lightpaths.

A capacity constraint violation adds a penalty value which is proportional to the total number of physical links which are assigned more lightpaths than the predetermined wavelength capacity. This penalty is multiplied with a penalty factor and added to the fitness of the solution.

The following example illustrates the application of the mutation operators and the fitness evaluation techniques.

Consider the physical and virtual topologies given in Figure 1. The first 4 shortest paths calculated based on hop counts and based on link costs can be seen in Table 1. Here, the first column shows the lightpaths as source-destination node pairs. Four shortest paths found using hop counts are given in the first four columns, and 4 shortest paths found using link costs are given in the next four columns.

Table 1. Four different shortest paths for the lightpaths of the example virtual topology given in Figure 1

	hop count				**link cost**			
lightpath	sp_1	sp_2	sp_3	sp_4	sp_1	sp_2	sp_3	sp_4
1-2 (a)	1-2	1-3-2	1-3-4-2	1-3-5-4-2	1-3-2	1-2	1-3-4-2	1-3-5-4-2
1-4 (b)	1-2-4	1-3-4	1-3-2-4	1-3-5-4	1-3-4	1-3-5-4	1-3-2-4	1-2-4
1-5 (c)	1-3-5	1-2-4-5	1-2-3-5	1-3-4-5	1-3-5	1-3-4-5	1-3-2-4-5	1-2-4-5
2-3 (d)	2-3	2-1-3	2-4-3	2-4-5-3	2-3	2-4-3	2-1-3	2-4-5-3
2-4 (e)	2-4	2-3-4	2-1-3-4	2-3-5-4	2-4	2-3-4	2-3-5-4	2-1-3-4
3-4 (f)	3-4	3-2-4	3-5-4	3-1-2-4	3-4	3-5-4	3-2-4	3-1-2-4
4-5 (g)	4-5	4-3-5	4-2-3-5	4-2-1-3-5	4-5	4-3-5	4-2-3-5	4-2-1-3-5

Assume we have an individual encoded as [1 1 2 3 1 1 2]. This encoding means that the first lightpath uses the 1^{st} shortest path (1-2), the second one uses the 1^{st} shortest path (1-2-4), and the third one uses 2^{nd} shortest path (1-2-4-5), etc. If we sum up the number of wavelength-links used in this solution, we have a total of 12 wavelength-links for hop count evaluation, and 14 wavelength-links for link cost evaluation. For example, in f_1, the number of physical links which leave the network disconnected in the event of a failure are counted. For this sample individual, if a failure occurs on the physical links connecting nodes 1-2, 2-4, and 3-4, the virtual topology becomes disconnected. Thus, a penalty for 3 survivability violations ($p * 100$) is added to the fitness, where p is the penalty factor taken as 100 in this example. Six different fitness values calculated using three different evaluation functions and two different cost metrics are given in Table 2. When calculating the fitness, the link costs given in Figure 1 are normalized dividing by 100.

Table 2. Fitness values for individual [1 1 2 3 1 1 2] calculated using three different evaluation functions and two different cost metrics. penalty factor=100.

	hop count	link cost
f_1	12+100*3=312	22.5+100*2=214
f_2	12+100*9=912	22.5+100*5=514
f_3	12+100*4=412	22.5+100*3=314

Assume a path mutation occurs on the second gene of this sample individual, the new individual becomes [1 2 2 3 1 1 2] or [1 4 2 3 1 1 2] with equal probability, if shortest paths are calculated according to hop count. However, if link costs are used instead, the individual becomes [1 4 2 3 1 1 2].

4 Experimental Design

To determine a good operator and evaluation method combination, we performed a series of experiments on our newly designed mutation operators and fitness evaluation methods. In these experiments, we used three metrics for performance comparisons, namely success rate (sr), first hit time (fht), and correlation length. Success rate is defined as the percentage of program runs in which a survivable mapping that does not violate the capacity constraint is found. First hit time is the first iteration during which the best solution is encountered.

Ruggedness [3] is commonly used to analyze the structure of fitness landscapes and algorithm behavior. The autocorrelation function (ACF) is one of the simplest and the most commonly used techniques for analyzing the ruggedness of a landscape. The ACF looks at the amount of fitness correlation between the points in the search space and takes on values between $[-1, 1]$. Values close to 1 denote a high positive correlation and values close to 0 show low correlation. The ACF value ρ_s is calculated as

$$\rho_s = \frac{\sum_{t=1}^{T-s}(f_t - \bar{f})(f_{t+s} - \bar{f})}{\sum_{t=1}^{T}(f_t - \bar{f})^2} \quad , \quad \lambda = -\frac{1}{ln(\mid \rho_1 \mid)} \tag{1}$$

where, s is the step size, T is the total number of sample points, f_t is the fitness of the t. solution, and $\bar{f}$ is the average fitness of all the points.

To compare the ruggedness of different landscapes, usually the correlation length, λ, as defined in Eq. 1 is used. A high correlation length means a smoother landscape, while a low correlation length shows a more rugged landscape.

4.1 Experimental Setup

For the experiments, we used two different physical topologies: the 14-node 24-link NSF network and a 24-node 43-link network (see [1] chapter 11 pp.557). For each physical topology we created 10 random VTs with average connectivity degrees of 3, 4, and 5. We assumed 10 wavelengths per physical link.

In the EA performance tests, we considered a maximum fitness evaluation count of 5000, mutation probability of $1/l$, where l is the number of lightpaths, crossover probability of 1.0, and population size of 100, and we ran the program 20 times for each parameter set.[1] In the landscape analysis tests, we used 1000 random walks and 50 runs. A penalty factor of 200 is used in the tests using hop count for shortest path calculation, and 300 in the tests using link cost.

4.2 Experimental Results

All EA component combinations we tested were able to solve the problem for NSF network for all 30 VTs, i.e., they were able to find survivable solutions of equal quality for all VTs. Since, the NSF network is a fairly simple and sparse graph, the results do not show meaningful differences between the tested

[1] On average a feasible solution is obtained for this problem in less than half a minute.

combinations. Therefore, in this paper we report the results for the 24-node 43-link network.

The results of the experiments are given in Tables 3, 4, and 5. Table 3 shows the success rates, Table 4 shows the correlation lengths, and Table 5 shows the first hit times. In all the tables, f_i denotes the corresponding results for the i^{th} fitness evaluation method. In the top half of the tables, the results obtained using the gene mutation are given, whereas in the bottom half, the path mutation results are given. Also, the results for three connectivity degrees, 3,4, and 5, can be seen in the tables. For the correlation length and the EA first hit time results, we also showed the standard errors of the means (e) in the tables.

Table 3. Success rates for 24-node network

		5 shortest paths						10 shortest paths						15 shortest paths					
		hop count			link cost			hop count			link cost			hop count			link cost		
		f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3
gene	3	89	62	19.5	71	57	18.5	89.5	95	51.5	89	94	52.5	92	95.5	51	93.5	93.5	54.5
	4	90	90	88	90	90	78.5	100	100	97	100	100	92.5	98.5	99	95.5	99.5	100	93.5
	5	100	100	100	100	100	100	100	100	100	100	99.5	100	100	100	100	100	100	100
path	3	89	62.5	29	73	59.5	23	93	95	61	91.5	91	65.5	95	95.5	63.5	92	93.5	64
	4	90	90	90	90	89.5	88.5	100	100	100	99.5	100	96.5	98.5	99.5	100	99	100	96.5
	5	100	100	100	100	99.5	99	100	100	100	99.5	100	100	100	100	100	100	100	100

From the tables, we can see that there is a difference in all the combinations for smaller shortest path counts and low connectivity degrees. These are relatively difficult problems because the probability of finding potential mappings increases with the node degrees and the number of alternative shortest paths.

As can be seen in Table 3 the performance of the third fitness evaluation method (f_3) is the worst of all. This is confirmed by Table 4, where f_3 has lower correlation lengths than f_1 and f_2, showing a more rugged landscape.

Table 4. Average and standard error of correlation lengths

			5 shortest paths						10 shortest paths						15 shortest paths					
			hop count			link cost			hop count			link cost			hop count			link cost		
			f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3
gene	3	λ	15.3	16.2	9.4	15.5	16.5	11.4	15.2	16.5	10.1	15.5	16.8	11.5	14.8	16.4	10.2	14.7	16.4	11.2
		e	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.2	0.1	0.1	0.2	0.1	0.1	0.2	0.1	0.1	0.2	0.1
	4	λ	18.3	20.5	12.5	18.7	20.4	13.5	17.2	20.4	11.9	17.9	20.5	12.4	16.7	19.7	11.4	17.1	20.7	12.3
		e	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.1	0.2	0.2	0.1	0.2	0.2	0.1	0.2	0.2	0.1
	5	λ	17.0	22.8	13.9	18.7	23.8	14.8	16.4	22.8	13.3	17.1	23.2	14.3	15.5	22.4	12.3	16.0	22.2	13.3
		e	0.2	0.3	0.2	0.2	0.3	0.2	0.2	0.3	0.2	0.2	0.3	0.2	0.2	0.3	0.1	0.2	0.3	0.2
path	3	λ	10.8	11.1	6.6	10.6	10.8	7.0	10.4	10.8	6.7	10.2	10.6	7.1	10.5	11.2	6.9	10.2	10.8	7.1
		e	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	4	λ	12.9	13.4	8.0	12.0	12.7	8.6	11.8	12.4	7.9	11.7	12.5	7.9	11.7	13.0	7.9	12.0	12.5	8.0
		e	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	5	λ	12.1	14.8	9.5	12.4	15.3	9.8	11.8	14.3	9.2	12.0	14.3	9.4	12.0	14.2	9.4	11.9	13.5	9.2
		e	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.2	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1

A difference between f_1 and f_2 can be seen for the 3 connected virtual topology tests. We can say f_1 performs better than the others on relatively difficult problems. In order to confirm this result, we created 100 different virtual topologies of connectivity degree 3 and ran the program 100 times for each of these topologies, for 5 shortest paths, gene mutation, and hop count. We applied a 2 sample 2-tailed t-test with a significance level of 0.05 and saw a statistically significant difference between these two fitness evaluation methods. However, if the algorithmic complexity of fitness evaluation methods as explained in section 3.2 are considered, we can say that f_2 is better. Therefore, f_2 should be preferred for virtual topologies having larger degrees of connectivity.

A difference can be seen between hop count and link cost results for f_1 and f_2 in Table 3. However, as a result of the t-test as in the previous paragraph, we cannot say that there is a statistically significant difference between them. Similarly, there is no difference in their landscapes as given in Table 4. If we consider the first hit counts, we can prefer hop count to link cost.

A difference between gene mutation and path mutation can be seen in Table 3. However, again as a result of the same type of t-test as in the previous paragraphs, we cannot say that there is a statistically significant difference between them. However, if we look at Table 5, we can see that the first hit times of path mutation is lower than gene mutation. In Table 4, it can be seen that the correlation lengths for path mutation is less than the gene mutation. This is an expected result, since the neighborhood definition for path mutation means the most faraway results.

Table 5. Average and standard error of EA first hit times

			5 shortest paths				10 shortest paths				15 shortest paths			
			hop count		link cost		hop count		link cost		hop count		link cost	
			f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2
gene	3	fht	2735	2715	4012	3530	4176	4256	4815	4834	4670	4666	4901	4895
		e	53.56	72.75	42.58	84.23	38.55	34.4	14.56	13.08	20.71	18.69	9.97	8.17
	4	fht	2721	2889	4436	4176	4343	4464	4900	4897	4751	4765	4922	4912
		e	48.15	55	31.35	61.08	33.39	27.9	7.35	8.8	13.94	13.53	5.47	8.29
	5	fht	3250	3382	4828	4573	4691	4764	4916	4921	4850	4871	4940	4923
		e	47.1	50.04	14.04	53.59	19.32	14.77	5.76	6.56	9.1	8.62	5.24	16.92
path	3	fht	2722	2875	3653	3334	3912	3993	4499	4624	4322	4442	4750	4814
		e	56.44	71.42	51.19	74.09	46.68	45.81	31.44	28.04	39.27	33.44	22.85	16.04
	4	fht	2621	2945	3964	3861	3849	4003	4731	4758	4469	4492	4867	4884
		e	55.42	59.4	45.1	62.98	45.65	45.03	19.75	20.03	30.44	25.22	9.37	8.72
	5	fht	2937	3146	4516	4367	4321	4415	4881	4878	4738	4768	4892	4915
		e	48.31	49.07	29.35	53.73	34.82	31	10.35	10.76	16.15	15.61	7.87	6.71

As a summary of the experiments, using path mutation, hop count, f_2 (f_1 in sparse virtual topologies) can be recommended as components for an effective EA for the survivable VT mapping problem.

5 Conclusion and Future Work

Our main aim in this study was to design an efficient evolutionary algorithm to find a survivable mapping of a given virtual topology while minimizing the resource usage. We experimented with different evolutionary algorithm components, and developed three fitness evaluation methods, two cost metrics and two mutation operators. We used a very simple search space structure analysis technique to support our results. As a result of experiments, we designed a suitable evolutionary algorithm and showed that evolutionary algorithms can be successfully used for the survivable virtual topology mapping problem. Overall, the results are promising and promote further study to improve the EA performance. As future work, we plan to use better metrics for landscape analysis and explore more sophisticated nature inspired heuristics for this problem.

References

1. Mukherjee, B.: Optical WDM Networks. Springer, New York (2006)
2. Modiano, E., Narula-Tam, A.: Survivable Lightpath Routing: A New Approach to the Design of WDM-Based Networks. IEEE Journal on Selected Areas in Communications 20(4), 800–809 (2002)
3. Reeves, C.R.: Search Methodologies: Introductory Tutorials i n Optimization and Decision Support Techniques. In: Burke, E.K., Kendall, G. (eds.) Fitness Landscapes, ch. 19. Springer, Heidelberg (2005)
4. Armitage, J., Crochat, O., Le Boudec, J.-Y.: Design of a Survivable WDM Photonic Network. In: INFOCOM 1997. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Driving the Information Revolution (1997)
5. Nucci, A., Sanso, B., Crainic, T., Leonardi, E., Marsan, M.A.: Design of Fault-Tolerant virtual Topologies in Wavelength-Routed Optical IP Networks. In: Proceedings of IEEE Globecom (2001)
6. Ducatelle, F., Gambardella, L.M.: A Scalable Algorithm for Survivable Routing in IP-Over-WDM Networks. In: First International Conference on Broadband Networks (BROADNETS 2004), (2004)
7. Kurant, M., Thiran, P.: Survivable Mapping Algorithm by Ring Trimming (SMART) for Large IP-over-WDM Networks. In: Proceedings of BroadNets (2004)
8. Banerjee, N., Sharan, S.: An EA for Solving the Single Objective Static RWA Problem in WDM Networks. In: ICISIP (2004)
9. Saha, M., Sengupta, I.: A GA Based Approach for Static Virtual Topology Design in Optical Networks, Indicon (2005)
10. Todimala, A., Ramamurthy, B.: A Scalable Approach for Survivable Virtual Topology Routing in Optical WDM Networks. IEEE Journal on Selected Areas in Communications 25(6) (August 2007)