Investigation of Hyper-Heuristics for Designing Survivable Virtual Topologies in Optical WDM Networks

Fatma Corut Ergin, A. Şima Uyar, and Ayşegül Yayimli

Istanbul Technical University
fatma.ergin@marmara.edu.tr,
{etaner,gencata}@itu.edu.tr

Abstract. In optical WDM networks, a fiber failure may result in a serious amount of data loss, hence, designing survivable virtual topologies is a critical problem. We propose four different hyper-heuristic approaches to solve this problem, each of which is based on a different category of nature inspired heuristics: evolutionary algorithms, ant colony optimization, simulated annealing, and adaptive iterated constructive search are used as the heuristic selection methods in the hyper-heuristics. Experimental results show that, all proposed hyper-heuristic approaches are successful in designing survivable virtual topologies. Furthermore, the ant colony optimization based hyper-heuristic outperforms the others.

Keywords: Optical networks, WDM, survivable virtual topology design, hyper-heuristics.

1 Introduction

In today's world, the steady increase in user demands of high speed and high bandwidth networks causes researchers to seek out new methods and algorithms to meet these demands. The most effective medium to transmit data is the fiber. Optical networks [8] are designed for the best usage of the superior properties of the fiber, e.g. high speed, high bandwidth, physical strength, etc. Today, with the help of the wavelength division multiplexing (WDM) technology, hundreds of channels can be built on a single fiber. WDM is a technology in which the optical transmission is split into a number of non-overlapping wavelength bands, with each wavelength supporting a single communication channel operating at the desired rate. Since multiple WDM channels can coexist on a single fiber, the huge fiber bandwidth can be utilized.

A wavelength-routed WDM network provides end-to-end optical connections between two nodes in the network that are not necessarily connected directly by a fiber in the physical layer. These optical connections are called lightpaths. Two nodes become virtually neighbors when a lightpath is set up between them. More than one lightpath, each operating on different wavelengths, can be routed on the same fiber. All the lightpaths set up on the network form the virtual

C. Di Chio et al. (Eds.): EvoApplications 2011, Part II, LNCS 6625, pp. 1–10, 2011.

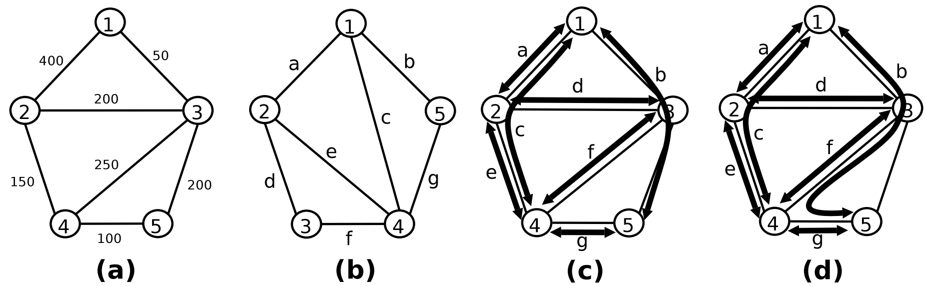

Fig. 1. a. Physical Topology, b. Virtual Topology, c. Survivable Mapping, d. Unsurvivable Mapping

topology (VT). Given the physical parameters of the network (physical topology, optical transceivers on the nodes, number of wavelengths that can be carried on the fibers, etc.) and the mean traffic intensities between the nodes, the problem of determining the lightpaths to be set up on the physical topology is known as the VT design problem.

In a WDM network, failure of a link (fiber) may result in the failure of several lightpaths routed through this link, which leads to several terabits of data loss. Survivable VT design aims to provide a continuous connectivity, using less resources. The continuous connectivity is ensured by designing the VT such that the VT remains connected in the event of a single link failure.

Assume that we have a physical network topology as in Figure 1.a and the virtual network topology to be routed on this physical topology is designed as in Figure 1.b. To obtain a survivable design of this VT, the mapping may be as in Figure 1.c. In this survivable mapping, a single 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 communicate and node 5 will be disconnected from the rest of the network.

Survivable VT design consists of four subproblems: determining a set of lightpaths (forming the VT), routing these lightpaths on the physical topology, so that any single fiber cut does not disconnect the VT, assigning wavelengths, and routing the packet traffic. Each of these subproblems can be solved separately. However, they are not independent problems and solving them one by one may degrade the quality of the final result considerably. Furthermore, the survivable VT design 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. In this study, we try to solve the survivable VT design problem as a whole using different hyper-heuristic (HH) [1] approaches. A HH is a method used to select between the low-level heuristics (LLH) at each step of an optimization process. This way, the best features of different heuristics can be combined.

In this study, we propose four HH approaches to design a survivable VT for a given physical topology, while minimizing resource usage. The proposed HHs use four different methods for heuristic selection: evolutionary algorithms (EA) [4], simulated annealing (SA) [6], ant colony optimization (ACO) [3], and adaptive iterated constructive search (AICS) [6]. From these methods, SA and EA are perturbative search methods, while AICS and ACO belong to the group of constructive search algorithms. Furthermore, SA and AICS are single point search methods, whereas, EA and ACO work on a population of solution candidates.

The rest of the paper is organized as follows: The survivable VT design problem is defined in Section 2. In Section 3, a detailed explanation of the approaches we propose to solve the problem is given. The experimental results are presented in Section 4. Section 5 concludes the paper.

2 Survivable Virtual Topology Design Problem

In optical networks, any damage to a physical link (fiber) on the network causes all the lightpaths routed through this link to be broken. Since huge amounts of data (e.g. 40 Gb/s) can be transmitted over each of these lightpaths, a fiber damage may result in a serious amount of data loss. Several different protection mechanisms have been proposed [7] for the fiber and/or other network equipment failures. Basically, there are two approaches for the fiber failure: 1) Survivability on the physical layer 2) Survivability on the virtual layer.

In the first approach, each connection passing through the fiber, i.e. the lightpath, is protected by assigning backup lightpaths that are disjointly routed from the connection's first lightpath. On the other hand, the second approach ensures the VT to be connected even in the failure of any single physical link. The first approach provides survivability by providing extra routes for each lightpath in the VT, however, with a cost of a high number of unemployed network resources. Thus, it offers an expensive solution for applications which may not need a high level of protection. The second approach, which has attracted attention especially in recent years, is a cost effective solution. Today, most applications are tolerant to latencies of several minutes of repair time needed by the packet layer (web search, file transfer, messaging, etc.), as long as the network connection is not terminated. This approach uses less network resources than the first one, thus, it enables service providers to offer a more economic service to their users.

Solving the survivable VT design problem as a whole is NP-complete [2]. Therefore, most of the previous studies on this problem consider only the survivable VT mapping subproblem [5]. There are only two studies in literature which try to solve all the subproblems together: a tabu search heuristic [9] and an ILP formulation [2]. The tabu search heuristic in [9] is constrained with small nodal degrees (up to 5) of VTs. Similarly, because of the problem complexity, the ILP method in [2] can solve very small problem instances of up to 4 node physical topologies, optimally.

2.1 Formal Problem Definition

The survivable VT design problem is defined as follows:
Given:

- Physical topology: Nodes and physical links that connect the nodes,
- Average traffic rates between each node pair,
- Maximum number of lightpaths that can be established on a node, i.e. the number of transceivers per node,
- Lightpath bandwidth capacity

Find:

- A collection of lightpaths to be established as a VT
- A survivable mapping (routing of the lightpaths over the physical topology, and wavelength assignment)
- A suitable routing of the packet traffic over the VT

The detailed ILP formulation for the survivable VT design problem can be found in [2]. Based on this formulation, the objective is to minimize the resource usage of the network, i.e. the total number of wavelength-links used in the physical topology. A wavelength-link is defined as a wavelength used on a physical link. For example, in Figure 1.c, 2 wavelength-links are used on the link between nodes 1 and 2, 1 wavelength-link is used on the link between nodes 1 and 3, ..., and a total of 9 wavelength-links are used in the physical topology.

3 Proposed Solution to the Survivable Virtual Topology Design Problem

To solve the survivable VT design problem, we use four HH approaches, each of which is based on a different type of nature inspired heuristic (NIH), used as the heuristic selection method. These NIHs are: evolutionary algorithms (EA), ant colony optimization (ACO), adaptive iterated constructive search (AICS), and simulated annealing (SA). Each method belongs to a different category of search approaches:

1. **EA:** population based, perturbative search
2. **ACO:** population based, constructive search
3. **SA:** single point perturbative search
4. **AICS:** single point, constructive search

Given the traffic matrix, the first step is to *determine a suitable VT*. For this subproblem, we selected three commonly used VT design heuristics as LLHs. At each step of the solution construction, a LLH is used to choose the next set of node pairs to establish a lightpath in between. The first LLH chooses the nodes which have the maximum single direction traffic demand between them. The second LLH considers the maximum bidirectional total traffic demand between node pairs. The third LLH chooses a node pair randomly. These three LLHs will be abbreviated as MAX_SNG, MAX_TOT, and RND, respectively for the rest of the paper. The lightpaths are established, such that, the in and out degrees of the nodes do not exceed the maximum number of transceivers on each node. In a preliminary study, we used two more LLHs to determine the nodes to add a lightpath. The first one chooses the node pairs with the minimum single direction traffic demand between them and the second one chooses those with the minimum bidirectional total traffic demand in between. The results showed that these LLHs do not improve the solution quality. Therefore, in this study we do not work with these LLHs.

The *VT routing and wavelength assignment (mapping) subproblem* is solved in our previous work [5] using ACO, which provides high quality solutions in a relatively small amount of time. Therefore, we use the same ACO approach to solve this subproblem in this study, too.

Traffic routing is applied in a straightforward way. The shortest path routing heuristic [8] is used for routing the packet traffic.

In HHs, the quality of the solutions is determined through a fitness function. In this study, the fitness of a solution is measured as the total number of wavelength-links used throughout the network, which is referred to as resource usage. The objective of the survivable VT design problem is to minimize this resource usage while considering the survivability and the capacity constraints. Resource usage is calculated by counting the number of physical links that are used by the lightpaths. An infeasible solution can either be penalized by adding a value to the fitness function, or can be discarded. In our HH algorithms, if a solution is found to be infeasible during the phases of creating a VT and routing the lightpaths, it is discarded. In the next stage, if the traffic cannot be routed over the physical topology, a penalty value proportional to the amount of traffic that cannot be routed, is added to the fitness of the solution.

A solution candidate is represented as an array of integers showing the order of LLHs to select lightpaths to be established. Since there are 3 different LLHs, the integers can have values between 1 and 3. When a lightpath between two selected nodes is established, the lightpath is assumed to be bidirectional. Therefore, a tranceiver at both ends is used. The length of the solution array is equal to the maximum number of lightpaths that can be established, i.e. $number\ of\ transceivers\ on\ each\ node * number\ of\ nodes/2$. For example, in a network with 6 nodes and 3 transceivers per node, each solution candidate is of length 6*3/2=9. If a solution candidate is represented with an array of [2 1 1 3 2 3 2 2 2], this means that, first a lightpath will be selected using the second LLH, then the next two using the first, continuing with the third, second, ... LLHs. While adding the lightpaths, the transceiver capacity constraint is handled. If the lightpath added according to the corresponding LLH results in using more than the existing number of transceivers in one or both ends, this lightpath is not added to the VT and the algorithm continues with the next LLH in the solution array. The lightpath capacity is 40 Gb/s and the traffic flows between the nodes is routed through the established lightpath between them.

Algorithm 1. General flow for the survivable VT design algorithm

Input: physical topology and traffic matrix
Output: a survivable VT

```
repeat
    use HHs to generate a VT
    use ACO to find a survivable mapping of lightpaths on the physical topology
    use shortest path heuristic to route traffic on the physical topology
    calculate fitness of the solution candidate
    apply NIH operators
until a predefined number of solutions are generated
```

The algorithm establishes lightpaths until either the end of the solution array is reached or until no traffic remains in the traffic matrix.

For each solution candidate produced by the HH, the corresponding VT is determined using the method explained above. Then, if the generated VT is not at least 2-connected (at least 2 link-disjoint paths for packet traffic exist between each node pair), new lightpaths are added subject to the transceiver capacity until the VT becomes 2-connected. For the nodes that have a degree lower than two, a new lightpath is added between this node and the node with the highest traffic demand in between. Next, the best mapping for the VT is found using ACO [5]. Then, the packet traffic is routed through the shortest paths starting from the node pair with the largest traffic demand. Finally, the fitness is calculated as the total amount of resource usage, i.e. the number of wavelength-links used throughout the network. The general flow of the algorithm is given in Algorithm 1.

3.1 Evolutionary Algorithms as a HH

We use a steady-state EA with duplicate elimination. After generating an initial set of random solution candidates, the EA operators, i.e. tournament selection, uniform crossover, and gene mutation, are applied and new solution candidates are generated. Gene mutation is defined as changing a LLH in the selected point of the string with another randomly determined LLH. Initial population of solution candidates (individuals) is generated randomly.

3.2 Ant Colony Optimization and Adaptive Iterated Constructive Search as HHs

ACO and AICS are very similar in the way they solve the problem. We can say that AICS is a form of ACO which uses only one ant. We use the elitist ant system (EAS) as the ACO variation, based on the results of our previous study [5], which show that this variation performs better than the others on the survivable VT mapping subproblem. However, there is only one ant in AICS, and the AS is applied as the ACO variation.

Initially, each ant iteratively adds a random LLH to its partial solution. The solution construction terminates when a solution array with a length equal to the maximum number of lightpaths is generated. No constraint is applied to the solution in the construction phase. Since there is no heuristic information, the solution construction only depends on the pheromone trail. The pheromone trails τ_{ij} we use in this paper refer to the desirability of using the j^{th} LLH to add the i^{th} lightpath. Pheromone trails are initialized using the initial random solutions of the ants. Then, they are modified each time that all ants have constructed a solution.

3.3 Simulated Annealing as a HH

We use a non-standard SA where the neighborhood operator is modified over time. The neighborhood operator is defined similar to the mutation operator in the EA-based HH, where with a given mutation probability, a randomly chosen LLH on the solution candidate is replaced by another LLH. The difference is that, we define a larger mutation probability in the beginning of the SA. The mutation probability is decreased by a predefined factor each time after 5 solution candidates are generated. This allows us to have a high exploration rate in the beginning of the search while focusing on exploitation towards the end. In our study, we used the formula given in [10] to calculate the initial temperature.

4 Experimental Results

We present the experimental results for a 24-node 43-link telco network [8], which is a fairly large-sized network for this problem. For the experiments, we use 20 different traffic matrices, randomly generated according to a frequently-used traffic generation method [8], where, 70% of the traffic is uniformly distributed over the range [0, 0.5 Gb/s] and 30% of the traffic is uniformly distributed over the range [0, 5 Gb/s]. The lightpath channel capacity is chosen as 40 Gb/s, which is typical in real-world networks.

First, we performed tests to see the performance of each LLH separately on the problem. For this test, only one LLH is used to generate the complete solution. Tests are run once for each LLH and traffic matrix pair.

Next, for the second group of experiments, we perform tests to find good parameter settings for the approaches used as heuristic selection methods in the HHs. We run the program 20 times for each tried parameter set.

As a result of parameter setting tests of *EA*, we selected 10 as the population size, and $1/l$, as the mutation probability, where l is the solution array length, and 0.8 as the crossover probability. The tests show that no significant improvement is obtained after a total of 100 individuals are created in the EA. Therefore, each run is terminated after 100 individuals are generated.

For the *AICS* parameters, the test results show that the value of q_0 does not affect the solution quality significantly. However, a slightly better solution quality is achieved with a q_0 value of 0.8. We selected 0.1 as the ρ value. To perform a fair comparison between methods, the termination condition is selected as creating 100 solution candidates. Since *ACO* is similar to *AICS* in which more than one ants are used, again, the q_0 value is selected as 0.8, the ρ value as 0.1, and the program is terminated after 100 solutions are generated. We selected 5 as the number of ants.

In *SA* parameter setting tests, the initial temperature selection is performed similar to [10]. As a result of the tests, we use an initial temperature of 195. The termination condition is again the same, i.e. 100 solution generations. The cooling rate is selected as 0.85, since this rate decreases the temperature to 5% of the initial temperature in 100 steps. The initial mutation probability is selected as $30/l$, where l is the solution array length, and is decreased with a factor of 0.85 in every 5 steps of solution generation. The large initial mutation probability is selected because of the large size of the solution array. If we start with a small mutation probability, the *SA* algorithm will search in a small neighborhood of the starting point, which may lead to getting stuck in local minima. The mutation rate is gradually decreased to avoid a random exploration in the search space. We decrease the mutation probability until it reaches a value of $1/l$.

After the parameter tuning of each approach, tests to explore their performance are conducted. In this set of experiments, we include a *Random* heuristic as a baseline for the performance comparisons. In *Random*, 100 random solution strings are generated, which are composed of randomly selected LLHs. For each traffic matrix, the solution candidate with the best fitness is selected as the solution. The tests are run 20 times for each traffic matrix. Each run uses a random initial seed. The results of the experiments are given in Table 1. Success rate of an approach is defined as the percentage of feasible solutions found. In the table, SR stands for success rate. The TM_i values in the first column of the table indicate that, the results in the corresponding row are the results obtained using

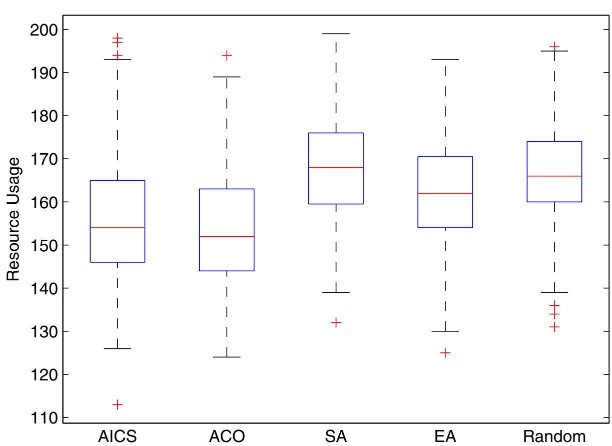

Fig. 2. The box-whisker plot of the results obtained using HHs and *Random* heuristic

traffic matrix i. In the next four columns of Table 1, the results for the four HHs designed in this study are given. The fifth column lists the results of the *Random* heuristic. The last three columns contain the resuls of LLHs applied separately. In the table, the values in the first five columns are the averages of resource usages obtained after 20 runs for each traffic matrix using the corresponding method. The last three columns are the results of using only the corresponding single LLH in the solution. The *NA* values in the table mean that a feasible solution was not found in any of the runs. The corresponding box-whisker plots are given in Figure 2.

From Table 1, we see that all the HHs perform better than the single LLHs. The success rates for the LLH_{RND}, *Random* and all the HHs are 100%, while, this rate is 70% for LLH_{MAX_SNG} and LLH_{MAX_TOT}. The best results obtained for each traffic matrix is marked in bold in the table. While in three of the traffic matrices, a single LLH produces the best result, it should be noted that the success rate for these LLHs is not 100%. The results show that, *ACO* finds the best result for 13 out of 20 traffic matrices, while the next method is *AICS* with 5 best results. To test the statistical significance of the results of the HHs and *Random*, we applied a two-way ANOVA and a Tukey HSD post hoc

Table 1. Resource usage results for different traffic matrices using different approaches

	EA	***ACO***	***AICS***	***SA***	***Random***	LLH_{MAX_SNG}	LLH_{MAX_TOT}	LLH_{RND}
TM_1	**169**	177	181	180	176	186	200	222
TM_2	158	152	149	163	165	**148**	160	229
TM_3	163	**148**	149	168	166	155	*NA*	219
TM_4	160	**146**	152	160	162	172	172	226
TM_5	154	**149**	151	156	157	170	159	220
TM_6	172	**156**	**156**	173	170	183	188	212
TM_7	175	**169**	172	185	180	*NA*	**169**	203
TM_8	167	**159**	162	173	173	167	169	222
TM_9	157	157	**151**	166	163	*NA*	*NA*	235
TM_{10}	172	161	**159**	176	174	194	198	221
TM_{11}	156	**137**	145	165	161	*NA*	155	222
TM_{12}	158	146	**145**	164	164	149	*NA*	218
TM_{13}	157	158	158	156	**155**	182	*NA*	236
TM_{14}	150	**135**	**135**	156	154	152	170	229
TM_{15}	168	**148**	155	176	173	*NA*	161	236
TM_{16}	163	**145**	147	165	165	*NA*	*NA*	219
TM_{17}	178	**175**	182	177	182	195	188	215
TM_{18}	**152**	162	156	160	163	*NA*	*NA*	213
TM_{19}	154	**139**	140	156	157	178	180	223
TM_{20}	170	**163**	166	175	177	**163**	177	221
Average	163	154	156	168	167	171	175	222
SR	1.0	1.0	1.0	1.0	1.0	0.7	0.7	1.0

test at a confidence level of 0.95. The results show that *ACO* and *AICS* produce results which are statistically significantly better than *SA*, *EA*, and *Random*, while a statistically significant difference cannot be observed between *ACO* and *AICS*. Therefore, we can say that, constructive search techniques are more successful for this problem. Furthermore, based only on averages, we conclude that a population based scheme is preferable to a single point one.

5 Conclusion

The HH approaches proposed in this study for survivable VT design is able to solve the problem for large-sized networks having intensive traffic demands with a 100% success rate. The results show that, using HHs can combine the best features of the LLHs and give better results. Furthermore, the HH approach based on ACO outperforms the other HH approaches based on EA, AICS and SA. This suggests that, a population based, constructive search approach as a HH is more suitable for this problem. As future work, the ILP solution to the problem with small-size networks will be investigated and results of the HHS will be compared to the optimum values.

References

1. Burke, E.K., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., Woodward, J.: Handbook of Meta-heuristics. Springer, Heidelberg (2010)
2. Cavdar, C., Yayimli, A.G., Mukherjee, B.: Multilayer resillient design for layer-1 VPNs. In: Optical Fiber Communication Conference and Exposition - OFC 2008, California, USA (February 2008)
3. Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
4. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
5. Ergin, F.C., Kaldirim, E., Yayimli, A., Uyar, A.S.: Ensuring resilience in optical WDM networks with nature inspired heuristics. IEEE/OSA Journal of Optical Communications and Networking 2(8), 642–652 (2010)
6. Hoos, H.H., Stutzle, T.: Stochastic Local Search: Foundations and Applications. Morgan Kaufmann, San Francisco (2005)
7. Kurant, M., Nguyen, H.X., Thiran, P.: Survey on dependable IP over fiber networks. Research Results of the DICS Program, 55–81 (2006)
8. Mukherjee, B.: Optical WDM Networks. Springer, Heidelberg (2006)
9. Nucci, A., Sanso, B., Crainic, T.G., Leonardi, E., Marsan, M.A.: On the design of fault-tolerant logical topologies in wavelength-routed packet networks. IEEE Journal on Selected Areas in Communications 2(9), 1884–1895 (2004)
10. Topcuoglu, H., Ermis, M., Corut, F., Yilmaz, G.: Solving the uncapacitated hub location problem using genetic algorithms. Computers & Operations Research 32(4), 967–984 (2005)