Ant Colony Optimization for Survivable Virtual Topology Mapping in Optical WDM Networks

Elif Kaldırım, Fatma Corut Ergin, Şima Uyar, Ayşegül Yayımlı
Computer Engineering Department
Istanbul Technical University
Istanbul, Turkey
Email: {kaldirime,erginfa,etaner,gencata}@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 determine a suitable ant colony optimization algorithm to solve this problem. Our results show that ant colony heuristics perform remarkably well while producing high quality solutions in less than half a minute.**

I. 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 each channel can be associated with a different optical connection, through which the upper layers (IP, Ethernet, etc.) can transmit data. 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 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 amounts 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. One approach to solve this problem is to design the virtual layer such that the virtual layer remains connected in the event of a single or multiple link failure.

To illustrate the 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 Figure 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.

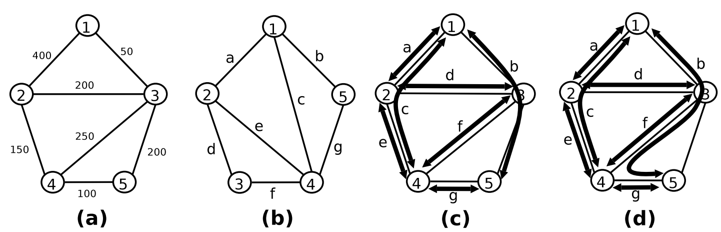

Fig. 1. a. Physical Topology, b. Virtual Topology, c. Survivable Mapping, d. Unsurvivable Mapping

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 ACO algorithms due to their successful applications on NP-complete problems and the way they handle the constraints. Unlike any other algorithms, ACO considers the constraints during solution construction, therefore infeasible solutions are not generated. We implemented six different ACO algorithms and compared their performance to determine which algorithm is more suitable to this problem and we investigated possible reasons.

The rest of the paper is organized as follows. The definition of the problem is given in Section II, followed by the related literature. In Section III the details of the implemented ACO algorithms are given. In Section IV application of ACO to

978-1-4244-5023-7/09/$25.00 ©2009 IEEE

September 14-16, 2009
METU
Northern Cyprus Campus

the survivable VT mapping problem is explained. Finally, in Section V the experimental results are reported and discussed.

II. 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 the VT means finding a route for each lightpath, such that in case of physical link failures the VT remains connected.

A. 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 a given physical topology is given in [2]. Based on this formulation, 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. To illustrate the difference between link and wavelength-link, assume that we have a VT routing as in figure 1.c. Here the number of physical links used is 7, whereas the total number of wavelength-links is 9. 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.

The *survivability constraint* states that for all proper cuts of the VT, the number of cut-set links 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. The *capacity constraint* ensures that the number of wavelengths on a physical link does not exceed its capacity W.

B. Related Literature

The survivable VT mapping problem was first addressed as Design Protection [3] in the literature. In this first study, tabu search was used to minimize the 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 [4]. 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. [5] studied the problem using a local search algorithm, while Kurant and Thiran [6] used an algorithm that divides the survivable mapping problem into subproblems. There are a few studies on VT mapping [7] and design [8] using EAs, however, survivability is not considered in any of them except [9]. Ergin et. al. proposed the only EA based approach for survivable VT mapping problem. Their objective is to minimize the resource usage without violating the survivability and the capacity constraints.

Swarm intelligence algorithms are used in a few studies for the Routing and Wavelength Assignment (RWA) problem. Ant colony optimization is applied to the static [10] and dynamic [11], [12] RWA problem without the survivability constraint. The only study using ACO considering back-up paths on the physical layer is [13]. Particle swarm optimization is applied to the RWA problem in only [14], in which the survivability constraint is not considered.

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 [15]. In [15], besides the physical network and the virtual network topologies, the shared risk link groups should be known in advance.

III. Ant Colony Optimization Algorithms

ACO is one of the most commonly used swarm intelligence techniques and is based on the behavior of real ants. ACO has been applied successfully to many combinatorial optimization problems such as routing problems, assignment problems, scheduling and sequencing problems and subset problems. Ant System (AS) is the first implementation of ACO algorithms, and has been the basis for many ACO variants. ACO algorithms have become the state-of-the-art for many applications, therefore, in this study, we implemented AS, elitist AS (EAS), rank-based AS (RAS), MAX-MIN AS (MMAS), ant colony system (ACS) and best-worst AS (BWAS) for the VT mapping problem since it can be seen in [16] that these direct variants of ACO have been successfully applied to many similar problems in literature.

The basic ACO algorithm is given in Algorithm 1. An iteration consists of the solution construction and pheromone update stages. In each iteration, each ant in the colony constructs a complete solution. Ants start from random nodes and move on the construction graph by visiting neighboring nodes at each step. An ant k chooses the best neighbor with a probability of q_0. Otherwise, the next node to visit is determined through a stochastic local decision policy based

Algorithm 1 Basic ACO outline

```
set ACO parameters
initialize pheromone levels
while stopping criteria not met do
    for each ant k do
        select random initial node
        repeat
            select next node based on decision policy
        until complete solution achieved
    end for
    update pheromone levels
end while
```

on the current pheromone levels and heuristic information between the current node and its neighbors with a probability p_{ij}^k as calculated in Eq. 1.

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^{\alpha} \ \cdot \ \eta_{ij}^{\beta}}{\sum_{l \in N_i^k} \tau_{il}^{\alpha} \ \cdot \ \eta_{il}^{\beta}} & \text{if } j \in N_i^k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where τ_{ij} and η_{ij} are the pheromone level and heuristic information between nodes i and j respectively, α and β are parameters used to determine the effect of the pheromone level and heuristics information respectively, N_i^k is the allowed neighborhood of ant k when it is at node i.

Pheromone trails are modified when all ants have constructed a solution. First the pheromone values are lowered (evaporated) by a constant factor on all edges. Then pheromone values are increased on the edges the ants have visited during their solution construction. Pheromone evaporation and pheromone update by the ants are implemented as given in Eq. 2 and Eq. 3 respectively,

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} \quad (2)$$

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^k \quad (3)$$

where $0 < \rho \leq 1$ is the pheromone evaporation rate and $\Delta\tau_{ij}^k$ is the amount of pheromone ant k deposits on the arcs it has visited. $\Delta\tau_{ij}^k$ is defined as given in Eq. 4, where C_k is the cost of the solution T_k built by the k-th ant.

$$\Delta\tau_{ij}^k = \begin{cases} 1/C_k & \text{if } edge(i,j) \in T_k \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The AS algorithm implements the basic ACO procedure detailed above. The following subsections explain the differences between the ACO variants and AS. For details see [16].

A. Elitist Ant System

The main idea of EAS is to provide additional reinforcement to the edge pairs which belong to T_{bs}, the best solution found since the start of the algorithm. This additional reinforcement of solution T_{bs} is achieved by adding a quantity e/C_{bs} to its arcs, where e defines the weight given to the best-so-far solution T_{bs}, and C_{bs} is its cost.

B. Rank-Based Ant System

The main idea of RAS is to allow each ant to deposit an amount of pheromone which decreases with its solution rank. The ants are sorted in decreasing order according to the quality of the solutions they constructed. The amount of pheromone an ant deposits is weighted according to its rank r. In each iteration only the $(w-1)$ best-ranked ants and the ant which has constructed the best-so-far solution are allowed to deposit pheromones. The best-so-far solution has the largest weight w, while the r-th best ant of the current iteration contributes pheromones with a weight given by $max\ \{0, w-r\}$.

C. MAX-MIN Ant System

The MAX-MIN Ant System(MMAS) has four major differences from AS.

- only either the ant which found the best solution in the current iteration, or the best-so-far ant is allowed to deposit pheromones
- allowed range of pheromone trail values is limited to the interval $[\tau_{min}, \tau_{max}]$
- pheromone trail values are initialized to the upper limit to increase exploration in the beginning
- pheromone trails are initialized when diversity is lost or when no improvement occurs for a given number of consecutive iterations

Pheromones are deposited on the edges according to Eq. 3 and Eq. 4 as in the AS, but the ant which is allowed to add pheromone may be either the best-so-far or the iteration-best. Commonly in MMAS implementations, both the iteration-best and the best-so-far update rules are used alternatively.

D. Ant Colony System

ACS differs from AS in three main points.

- it has a modified action selection rule
- pheromone evaporation and pheromone deposit take place only on the edges belonging to the best-so-far solution
- in addition to the global pheromone update performed at the end of each iteration, the ants also use the local pheromone update rule based on a preset parameter ξ.

E. The Best-Worst Ant System

BWAS differs from AS in three main points.

- while only the best-so-far ant is allowed to deposit pheromones, the worst ant of the current iteration subtracts pheromones on the arcs it does not have in common with the best-so-far solution
- search diversification is achieved through frequently reinitializing the pheromone trails
- to further increase diversity, pheromone mutation is used as explained in [16].

IV. Application of ACO to the Survivable VT Mapping Problem

The VT mapping problem can be seen as a search for the best routing of lightpaths through physical links. Therefore, we use a solution encoding inspired from [7]. For this encoding, first, 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 integer gives the index of the shortest path for the corresponding lightpath. These integers can take values between $[1..k]$ where *k* is the predefined number of shortest paths for the lightpaths. ACO steps are implemented as follows:

Construction graph: The construction graph is identical to the problem graph.

Constraints: Survivable VT mapping problem has two constraints: 1) the number of wavelengths on a physical link should not exceed its capacity, 2) all the lightpaths of a cut-set cannot be routed using the same physical link.

Pheromone trails and heuristic information: Two pheromone trails are implemented in the survivable VT mapping problem: the lightpath pheromone trails τ_{ij}^{l} refer to the desirability of choosing lightpath j directly after i for mapping, the shortest path pheromone trails τ_{ij}^{s} show the desirability of selecting j^{th} shortest path of lightpath i.

The heuristic information η_{ij} is inversely proportional to the length of the j^{th} shortest path of lightpath i, denoted as d_{ij}, i.e., $\eta_{ij} = 1/d_{ij}$. The heuristic information is used together with the shortest path pheromone trails while deciding the proper shortest path of the chosen lightpath.

Solution construction: Each ant is initially placed on a randomly chosen start lightpath and one of its shortest paths is selected. At each step, the ant iteratively adds an unvisited lightpath to its partial solution and decides the shortest path of the selected lightpath. The solution construction terminates once all lightpaths have been visited. Solutions are constructed by applying the following simple constructive procedure to each ant: (1) chosen a start lightpath and one of its shortest paths, (2) use lightpath pheromone information to select the next lightpath to route (3) use shortest path pheromone information together with the heuristic values to probabilistically determine the path between the nodes of the corresponding lightpath, until all lightpaths have been visited. If the ant cannot select a shortest path that makes the solution feasible, this ant is removed from the current iteration.

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 II-A. In order to determine if the solution is survivable or not, each physical link is deleted from the physical network one by one. If the VT graph becomes disconnected in the event of a broken physical link, the solution is considered as unsurvivable.

To illustrate 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 I. Here, the first column shows the lightpaths as source-destination node pairs. Four shortest paths found using hop-counts are given in the next four columns, and 4 shortest paths found using link-costs are given in the last four columns.

Assume we have an individual encoded as [1 1 2 3 1 1 2]. Based on hop-count, 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 the 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 2250 kilometers for link-cost evaluation.

V. Experimental Design

To compare the performance of ACO algorithms, we perform a series of experiments to calculate resource usage based on both hop-count and link-cost on 50 different instances each, for 3, 4, 5 connected virtual topologies. In these experiments, we used three metrics for performance evaluation, namely *success rate*, *first hit iteration* and the *resource usage*. 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 iteration is the first iteration during which the best-so-far solution is encountered. When calculating shortest paths based on hop-count, each wavelength-link is considered to have a length of 1 for each physical link. On the other hand, when calculating the shortest paths based on the link-cost, each wavelength-link is considered to have a length equal to the actual length of the physical path in kilometers.

A. Experimental Setup

ACO algorithm specific parameters given below are used as the default settings in [17].

α	β	ρ	m	q_0
1	1	0.1	10	0.5

Here, m is the number of ants. The parameter e in EAS is set to $e = m$. The number of ants that deposit pheromones in RAS is $w = 6$. In the local pheromone trail update in ACS $\xi = 0.1$. q_0 is taken as 0.5 for all algorithms. τ_0, the initial pheromone value, is set as follows:

	AS	ACS-BWAS	EAS-RAS	MMAS
τ_0	m/C_{min}	$1/nC_{min}$	$1/\rho C_{min}$	$1/2n\rho C_{min}$

For the experiments, we use a physical topology with 24 nodes and 43 links (see [1] chapter 11 pp.557). We created 50 random VTs with average connectivity degrees of 3, 4, and 5 to map onto this physical topology. We assumed that each physical link has a capacity of 10 wavelengths.

In ACO performance tests, for all elements in the problem set, we run each algorithm 10 times. Each run is allowed to continue for 20 seconds.

B. Experimental Results

We present the results of the experiments in Tables II, III, IV, and V. Table II shows the success rates of all ACO algorithms for randomly generated VTs. We have 3 different sets of topologies where the average node degrees are 3, 4, and 5. For each algorithm and node degree, 3

TABLE I

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 (c)	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 (b)	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

different numbers of alternative shortest paths for lightpaths are examined. We tried 5, 10, and 15 shortest path cases. Both hop-count and link-cost results for each case are given in the table. The success probability numbers are averaged over 500 runs(10 runs per VT instance).

Table II shows that ACO algorithms can successfully be used for the survivable VT mapping problem. Success rates are mostly greater than 90% when more than 5 shortest paths are provided to the algorithms. Although algorithms have relatively the same performance, MMAS is the best of all. It is followed by RAS, BWAS and EAS. They have relatively the same performance. ACS is the worst of all, and AS has slightly better performance than ACS. BWAS shows the best performance for 5 connected topologies when shortest path calculation is based on link-cost.

From Table II, we can see that success rates are increasing with the number of shortest paths because the probability of finding feasible potential mappings increases with the number of alternative shortest paths. When ants can use 15 shortest paths, each algorithm finds feasible solutions with success rates of almost 100%. Hop-count based calculation results in better success rates when 5 shortest paths are provided to each ACO algorithm. For 10 shortest path cases, when the virtual topology has an average node degree of 3, link-cost calculation is more successful than hop-count. For average node degrees of 4 and 5, hop-count success rates are greater than link-cost.

Table III and Table IV show resource usage of all ACO algorithms for link-cost and hop-count respectively. Here 3, 4, and 5 on the second column, represent the average node degrees for randomly generated VTs, whereas 5, 10, and 15 are the number of shortest paths provided to ACO algorithms. Both tables show lower and upper bounds of resource usage for the proposed solutions to the VTs when the confidence interval is 95%.

When we analyse Table III and Table IV, we conclude that for each algorithm, resource usage does not increase in the same proportion as the number of shortest paths. For perturbative search algorithms, such as EAs, the search space increases exponentially when the number of alternative shortest paths increases. However for ACO algorithms, the increase in search space size is linear. Therefore ACO algorithms are not affected significantly by the increased search space. Both Table III and IV show that when 10 shortest paths are provided to ACO algorithms, RAS has the best performance whereas when 15 shortest paths are provided to ACO algorithms AS is the best of all as RAS does not decrease in performance. We can conclude that AS and RAS can better react to the increase in the search space size.

As can be seen in Tables III and IV the performance of ACS is the worst of all as resource usage is relatively higher than the other algorithms. The main difference between ACS

TABLE II

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
as	3	0.714	0.552	0.882	0.92	0.97	0.972
	4	0.924	0.84	0.986	0.96	1	1
	5	0.97	0.774	0.998	0.994	1	1
ras	3	0.72	0.596	0.918	0.928	0.98	0.976
	4	0.936	0.886	1	0.96	1	1
	5	0.98	0.806	1	0.998	1	1
eas	3	0.718	0.598	0.914	0.936	0.976	0.978
	4	0.938	0.876	1	0.96	1	1
	5	0.972	0.84	1	0.998	1	1
mmas	3	0.72	0.614	0.92	0.938	0.98	0.98
	4	0.934	0.89	0.996	0.96	1	1
	5	0.98	0.846	1	0.99	1	1
bwas	3	0.716	0.6	0.918	0.922	0.98	0.972
	4	0.938	0.858	0.998	0.96	1	1
	5	0.98	0.914	1	1	1	1
acs	3	0.702	0.516	0.882	0.892	0.97	0.954
	4	0.922	0.83	0.984	0.96	1	1
	5	0.968	0.738	0.996	0.988	1	1

TABLE III

LOWER AND UPPER BOUNDS OF RESOURCE USAGE IN TERMS OF LINK - COST WITH 95% CONFIDENCE INTERVAL

		5 shortest paths	10 shortest paths	15 shortest paths
		lower - upper	lower - upper	lower - upper
as	3	112704 - 114756	120410 - 121964	114186 - 115820
	4	147526 - 148992	154913 - 156361	152280 - 153672
	5	188867 - 191431	201232 - 203530	204232 - 206438
ras	3	111451 - 113341	116939 - 118485	116449 - 117985
	4	147567 - 149001	153958 - 155430	153377 - 154863
	5	189601 - 191823	199971 - 202191	202754 - 204910
eas	3	111563 - 113491	118566 - 120094	120035 - 121503
	4	147718 - 149096	156405 - 157897	160254 - 161696
	5	191199 - 193361	203509 - 205707	208374 - 210400
mmas	3	110969 - 112901	119246 - 120710	116617 - 118045
	4	148038 - 149416	159208 - 160526	161592 - 162862
	5	192017 - 194207	206372 - 208286	214201 - 216025
bwas	3	114628 - 116496	121980 - 123446	119527 - 120969
	4	149520 - 150860	159471 - 160883	161185 - 162561
	5	193942 - 196002	205895 - 207801	210839 - 212749
acs	3	115247 - 117381	124376 - 125826	127217 - 128575
	4	152233 - 153659	163409 - 164745	170216 - 171520
	5	194845 - 197289	210227 - 212131	219958 - 221758

TABLE IV
LOWER AND UPPER BOUNDS OF RESOURCE USAGE IN TERMS OF HOP-COUNT WITH 95% CONFIDENCE INTERVAL

		5 shortest paths	10 shortest paths	15 shortest paths
		lower - upper	lower - upper	lower - upper
as	3	113.36 - 114.77	120.44 - 121.74	113.03 - 114.3
	4	147.71 - 148.83	156.39 - 157.57	152.84 - 154.04
	5	187.08 - 188.90	198.74 - 200.61	202.24 - 204.07
ras	3	111.76 - 113.11	116.58 - 117.88	115.68 - 116.95
	4	146.65 - 147.73	154.52 - 155.79	153.74 - 155.03
	5	187.66 - 189.35	197.34 - 199.11	199.99 - 201.82
eas	3	112.24 - 113.63	118.58 - 119.95	120.53 - 121.76
	4	147.74 - 148.84	157.33 - 158.61	162.44 - 163.73
	5	189.63 - 191.29	201.48 - 203.31	208.03 - 209.74
mmas	3	111.28 - 112.62	118.62 - 119.86	115.86 - 117.05
	4	147.95 - 149.10	160.80 - 161.91	163.42 - 164.56
	5	191.28 - 192.84	205.98 - 207.49	214.64 - 216.07
bwas	3	115.88 - 117.26	122.46 - 123.76	120.19 - 121.39
	4	150.44 - 151.50	161.74 - 162.94	163.77 - 165.03
	5	192.06 - 193.64	205.78 - 207.33	210.96 - 212.55
acs	3	116.57 - 117.99	125.41 - 126.71	128.27 - 129.48
	4	152.99 - 154.09	165.99 - 167.11	173.42 - 174.56
	5	194.6 - 196.23	210.82 - 212.40	221.19 - 222.62

TABLE V
AVERAGE FIRST HIT ITERATIONS

		5 shortest paths		10 shortest paths		15 shortest paths	
		hop	link	hop	link	hop	link
as	3	13.21	14.20	8.27	9.077	29.11	31.57
	4	11.07	11.80	8.69	8.55	16.69	16.48
	5	12.12	15.18	8.92	8.64	10.86	10.96
ras	3	23.74	27.91	14.80	16.01	26.68	29.04
	4	14.74	16.38	9.84	9.84	16.08	16.26
	5	11.61	12.59	9.05	9.08	10.97	11.14
eas	3	23.87	27.42	14.07	15.43	20.64	22.18
	4	15.78	18.16	10.18	10.04	13.41	13.72
	5	11.20	13.31	8.87	8.93	10.97	11.24
mmas	3	28.24	35.24	16.97	15.77	31.82	32.05
	4	18.04	19.01	8.81	8.44	14.95	15.05
	5	11.92	15.66	7.54	7.29	7.09	7.02
bwas	3	11.13	12.41	8.55	9.35	27.17	29.04
	4	16.82	19.22	9.68	10.19	15.59	16.43
	5	15.23	18.26	9.09	9.54	10.49	11.14
acs	3	11.46	13.94	6.91	7.88	21.34	22.03
	4	8.71	9.94	6.46	6.22	10.02	10.14
	5	9.94	12.91	6.68	6.69	6.70	7.78

and the other algorithms is the local pheromone update where each ant decreases the pheromone trail in each step when a lightpath and the shortest path is chosen. For the cases in which $q_0 \leq 0.5$, evaporation of pheromones may make ants forget the heuristic information.

Table V shows the first hit iterations in which the best solutions are retrieved. Average of the first hit iterations are given based on both hop-count and link-cost. From the results, we can see that best solutions are found in earlier iterations when the hop-count calculation method is used. Hop-count calculation method also results in better success rates for 4 and 5 connected virtual topologies (see Table II). We can see from Table V that ACS finds solutions in earlier iterations than the other algorithms. But the quality of the solutions found by ACS is not as good as the solutions found by the other algorithms (see Tables III and IV).

As a summary of the experiments, we recommend ACO algorithms for the survivable VT mapping problem due to their decision policy at each step. They find feasible solutions after each iteration. To increase the performance and the success rates, as many shortest paths as necessary should be used. Based on the results, even though all ACO algorithms perform well, we can recommend RAS using hop-count calculation method due to its overall success and better reaction to the increasing search space.

VI. CONCLUSION AND FUTURE WORK

Our main aim in this study is to find a survivable mapping of a given virtual topology while minimizing the resource usage using various ACO algorithms. We implemented six ACO algorithms and showed that ACO algorithms can be successfully used for the survivable virtual topology mapping problem. When we compare our results with EAs implemented in [9], we see that if we run the algorithms for the same amount of time, we obtain lower resource usage with higher success rates using ACO. Overall, the results are promising and promote further study to improve the ACO performance.

As in most heuristics, ACO performance depends on good parameter settings. As future work, we will examine the effect of ACO parameters and the number of shortest paths on the performance of the algorithms. We will also use classical optimization algorithms to compute lower bounds of the solutions to better asses the quality of our results.

REFERENCES

[1] B. Mukherjee, "Optical WDM Networks", Springer, New York, 2006.
[2] E. Modiano and A. Narula-Tam, "Survivable Lightpath Routing: A New Approach to the Design of WDM-Based Networks," IEEE Journal on Selected Areas in Communications, v.20, no.4, p.800-809, 2002.
[3] J. Armitage, O. Crochat, J.-Y. Le Boudec, "Design of a Survivable WDM Photonic Network," IEEE INFOCOM, 1997.
[4] A. Nucci, B. Sanso, T. Crainic, E. Leonardi, and M.A. Marsan, "Design of Fault-Tolerant virtual Topologies in Wavelength-Routed Optical IP Networks," IEEE GLOBECOM, 2001.
[5] F. Ducatelle, L. M. Gambardella, "A Scalable Algorithm for Survivable Routing in IP-Over-WDM Networks," BROADNETS, 2004.
[6] M. Kurant, P. Thiran, "Survivable Mapping Algorithm by Ring Trimming (SMART) for Large IP-over-WDM Networks," BROADNETS, 2004.
[7] N. Banerjee and S. Sharan, "An EA for Solving the Single Objective Static RWA Problem in WDM Networks," 2nd International Conference on Intelligent Sensing and Information Processing, 2004.
[8] M. Saha and I. Sengupta, "A GA Based Approach for Static Virtual Topology Design in Optical Networks," Indicon, 2005.
[9] F. Ergin, A. Yayımlı, Ş. Uyar, "An Evolutionary Algorithm for Survivable Virtual Topology Mapping in Optical WDM Networks," EvoWorkshops09, LNCS 5484, Springer-Verlag, p. 31-40, 2009.
[10] G.N. Varela, M.C. Sinclair, "Ant Colony Optimisation for Virtual-Wavelength-Path Routing and Wavelength Allocation," Congress on Evolutionary Computation, v.3, p.1809-1816, 1999.
[11] S. Ngo, X. Jiang, S. Horiguchi, "An Ant-Based Approach for Dynamic RWA in Optical WDM Networks," Photonic Network Communications 11, 39-48, 2006.
[12] R.M. Garlick, R.S. Barr, "Dynamic Wavelength Routing in WDM Networks via Ant Colony Optimization," 3rd International Workshop on Ant Algorithms, LNCS 2463, Springer-Verlag, v. 3, p. 250-255, 2002.
[13] S. Ngo, X. Jiang, V. Le, and S. Horiguchi. "Ant-based survivable routing in dynamic WDM networks with shared backup paths," Journal of Supercomputing , v.36, no.3, p.297-307, 2006.
[14] A. Hassan, C. Phillips and J. Pitts, "Dynamic Routing and Wavelength Assignment using Hybrid Particle Swarm Optimization," EPSRC PGNet2007, 2007.
[15] A. Todimala, B. Ramamurthy, "A Scalable Approach for Survivable Virtual Topology Routing in Optical WDM Networks," IEEE Journal on Selected Areas in Communications, v.25, no.6, 2007.
[16] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, 2004.
[17] T. Stützle, Ant Colony Optimization Source Code, http://www.aco-metaheuristic.org/aco-code.