Dynamic Anycast Routing and Wavelength Assignment in WDM Networks Using Ant Colony Optimization (ACO)

Kavitha Bhaskaran*, Joan Triay†*, Vinod M. Vokkarane*
*Department of Computer and Information Science, University of Massachusetts, Dartmouth, MA, USA
†Department of Telematics Engineering, Universitat Politècnica de Catalunya (UPC), Castelldefels, Spain
{kbhaskaran, vvokkarane}@umassd.edu, joan.triay@entel.upc.edu

***Abstract*—Ant colony optimization (ACO) is a probabilistic technique used for solving complex computational problems, such as finding optimal routes in networks. It has been proved to perform better than simulated annealing and genetic algorithm approaches for solving dynamic problems. ACO algorithms can quickly adapt to real-time changes in the system. In this paper, we propose an ACO-based algorithm to solve the dynamic anycast routing and wavelength assignment (RWA) problem in wavelength-routed optical networks. Using extensive simulations, we show that ACO-based anycast RWA significantly reduces blocking probability compared to the fixed shortest-path first (SPF) and other load-balancing and dynamic algorithms.**

Keywords: WDM, RWA, anycast, and ACO.

I. Introduction

Wavelength routing is a promising technology to support high-bandwidth demanding applications on the Internet. It provisions all-optical data paths among the network nodes. Data is transmitted between two nodes using a *lightpath*, which is uniquely identified by a wavelength and a physical path. When a lightpath operates on the same wavelength across all the fiber links that it traverses, it is said to satisfy the *wavelength-continuity constraint* (WCC).

One of the fundamental optimization problems in optical wavelength division multiplexing (WDM) networks is the routing and wavelength assignment problem. RWA involves the selection of a path and a wavelength for every connection request arriving to the network. Moreover, connection requests can be classified as either static or dynamic. In the static case, the connections are given in advance and the problem is to establish lightpaths for all these requests so that the total number of required wavelengths is minimized. In the other case, traffic is dynamic with connection requests arriving randomly, making the dynamic RWA more challenging. RWA problem is proved to be NP-complete [1].

To tackle the difficult problem, RWA can be divided into two sub-problems, the routing problem and the wavelength assignment problem, which can be solved individually. Routing can be further classified as fixed routing, fixed-alternate routing, and adaptive routing. In fixed routing, a connection always chooses the same route for a given source-destination pair. The disadvantage of using this approach is that, if resources are unavailable on this path, it potentially leads to high blocking probability. Also, it is difficult to handle link failures since fixed routing does not provide alternate paths to the destination. Fixed-alternate routing supplies multiple routes, hence one of its advantages is the reduction of the connection blocking probability compared to fixed routing. Finally, in adaptive routing, the route is calculated dynamically depending on the state of the network. In general, adaptive routing results in lower blocking probability than both fixed and fixed-alternate routing at the cost of a longer set up delay and a higher control overhead. Some of the wavelength assignment schemes can be classified as: random, first-fit, least-used, most-used, least-loaded, MAX-SUM, and relative capacity loss [2]. In this paper we use random wavelength assignment with the ACO-based routing algorithm.

Anycast refers to the transmission of data from a source node to any one member in the candidate destination set. In unicasting, each destination address uniquely identifies a single receiver, while in anycasting, each destination address identifies a set of receivers but only one of them is chosen for a given request to receive information from a given sender. If anycast requests are routed carefully, it can help carry additional dynamic traffic demands in the network [3]. Moreover, anycast is a promising communication paradigm for future optical Grid networks where a set of similar Grid resources are available as possible resource providers, out of which only one needs to be selected.

The main contribution of this paper is the use of ant colony optimization [4] to solve the dynamic anycast RWA problem efficiently. ACO is a probabilistic optimization techniques that uses *pheromone trails* laid by ants along different routes to find the optimal paths on the dynamic network.

The rest of the paper is organized as follows. In Section II, we discuss related research in this area. Section III presents the routing table structure and state information dynamic update mechanisms in ACO. Section IV describes the ACO-based anycast RWA algorithm. Section V analyzes the simulation results and, finally, Section VI concludes the paper.

II. Related Work

There are several research works that use ACO-based algorithms for routing in computer networks. AntNet [5] is an adaptive agent-based routing algorithm in which the paths are determined and computed with the help of forward and backward ant agents. The rationale behind using the two-way agents is to help the forward ants utilize the useful information gathered by the backward ants on their round trip between source and destination.

ACO-based algorithms were first introduced to solve RWA for optical networks in [6]. Additional works on ACO include [7] and [8]. In the ACO-based RWA algorithm [7], ants are launched on a separate control plane and forwarded on the WDM network to set up the connection request. Ants leave pheromone trails behind which help other ants find optimal paths in the network. Over time, the pheromone trails evaporate making a path less attractive for the ants to choose.

The longer it takes for an ant to travel the path back and forth, the more the pheromone evaporates. A short path is marched across faster and thus the pheromone density remains high as it is laid on the path as fast as it evaporates. Pheromone evaporation helps avoid the convergence to a local minima and also helps to explore the network dynamics efficiently. If there was no evaporation at all, all the paths chosen by the first ants would tend to be excessively attractive to the following ants, leading to stagnation.

Although there has been extensive research in the area of RWA, current RWA approaches still have problems while tackling with the complex nature of the routing problem in optical WDM networks. The fixed-routing approach has trouble in keeping itself up-to-date with the dynamically changing network state. Though existing adaptive RWA algorithms provide near optimal solutions, they are centralized and require global topology information. To the best of our knowledge, we are the first to propose the use of an ACO-based anycast algorithm to route the connection requests on optical WDM networks. The closest work present in the literature is [9]. However, it is mostly based on resource provisioning in optical grid networks. This motivates us to investigate a new anycast RWA approach that uses the ant-based mobile agents for optical network routing.

III. ACO Routing Mechanisms and Structures

In this section we will describe the basic mechanisms in the ACO-based algorithm, such as the routing table structure, how ants move and collect data, and how this data is updated in the routing tables to reflect the status of the network. The basis of our work is [7], from which we borrow some of its mathematical expressions.

A. Routing Table Structure

Every node in the network has a twin table routing structure comprised of a pheromone table [7] for ants' foraging and a *P-route* table [10] for connection setup. On a network consisting of N nodes, node i, which has k_i neighbors, has a probabilistic pheromone table $r_i = [r^i_{n,d}]_{N-1,k_i}$, where $N-1$ is the number of rows (possible destinations) and k_i is the number of columns (neighbors). The value $r^i_{n,d}$ is used in the pheromone table as a selection probability of neighbor node n, from current node i, when an ant is moving towards a destination node *d*. Similarly, the *P-route* table has $N-1$ rows and two columns, and lists all possible routes from the source to every destination with a corresponding *goodness value*. This value, dr, defines the cost of that route and it is computed using the following equations [10],

$$dr = \phi \frac{1}{(dl+1)} + (1-\phi)dw, \quad (1)$$

where

$$1 > \phi > \frac{[(W-1)(N-1)N]}{[W+(W-1)(N-1)N]} \quad (2)$$

In (1), ϕ is a scalar parameter used to adjust the emphasis between the length of the path, represented by dl (the difference between the length of the current path and the length of the overall shortest path to that destination), and the percentage of free wavelengths on that path, dw. The greater the value of ϕ, the more emphasis is given to shorter routes, whereas for a smaller value of ϕ, the goodness value is primarily weighted by the number of free wavelengths on the same route. The parameter ϕ should be chosen such that shorter routes have a greater goodness value, and in case of a tie (multiple same path length routes), the one with more free wavelengths. To do so, the value of ϕ is set based on the inequality in (2) [10].

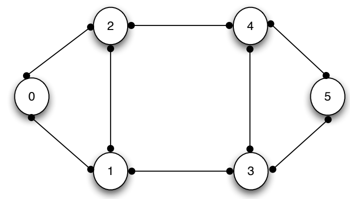


Fig. 1. 6-node network topology.

TABLE I
PHEROMONE TABLE AT NODE 3.

	Neighbor Nodes		
Destination	**1**	**4**	**5**
0	$r^3_{1,0} = 0.6$	$r^3_{4,0} = 0.3$	$r^3_{5,0} = 0.1$
1	$r^3_{1,1} = 0.8$	$r^3_{4,1} = 0.2$	$r^3_{5,1} = 0.0$
....	....	....	.....
5	$r^3_{1,5} = 0.2$	$r^3_{4,5} = 0.2$	$r^3_{5,5} = 0.6$

TABLE II
P-route TABLE AT NODE 3.

0	3.0;3.4.2.0;....
1	3.1;3.4.2.1;3.4.2.0.1;....
....	.....
5	3.5;3.4.5;....

A route and its goodness value is computed and stored in the *P-route* table of the ant's destination node, *d*, corresponding to the source *s* acting as the connection request's destination. For instance, when an ant moves along the path, $(s,\ldots,n-1,n,\ldots,d)$ the reverse path $(d,\ldots,n,n-1,\ldots,s)$ is added in the *P-route* table of *d*, with respect to the destination *s*. Then, this information is used when a connection request arrives from *d* to *s*. Hence, paths explored by the ants are used in the reverse direction by the arriving connection requests.

Fig. 1 shows an example of a 6-node network. The corresponding pheromone table at node 3 is shown in Table I. In this specific example, the pheromone table shows for each possible destination from node 3 (i.e., 0-1-2-4-5), and for each neighbor node (i.e., 1-4-5) the goodness value. That is, every neighbor node has a complementary value that gives information about the goodness of choosing this next hop to get to destination i. The sum of all the values on each row has to be equal to 1.0. Likewise, Table II shows an example of the *P-route* table with possible paths for every destination on the network from the source node 3. These paths are used to launch the needle ants when an anycast request arrives to the node.

In order to compute the complexity of this routing table mechanism, let assume there are N nodes on the network. The worst-case space complexity of the pheromone table is in the order of $O(N^2)$, for a fully meshed network. This is an acceptable complexity, specially on wavelength-routed optical networks, in which the number of nodes do not usually scale above hundreds. Moreover, the storage complexity of the *P-route* table depends on the number of routes to keep for each possible destination node; hence, assuming K routes, its size is $O(NK)$. Furthermore, the proposed algorithm also keeps track of the paths which have low significance and are not considered good candidates, in order to discard them later from the routing tables. In addition to this, pheromone concentration diminishes throughout time for unused paths, which are then removed from the table. This, in turn, permits to keep under control the amount of state information to manage.

B. Data Collection by Ants

To collect route and wavelength availability information, ants are launched from a randomly selected source, *s*, to a randomly selected destination, *d*. During their forwarding process, the objective is to find the best neighbor node toward the destination *d* at every intermediate hop. The neighbor node with the highest node-selection probability is selected as the next hop. Ants are launched from each source node every T time units with a probability ρ. All the ants collect data on their trip and constantly update the routing table on visited nodes. Concerns may arise regarding how this data collection might be implemented on a real network. In this respect, OSPF-TE [11] may be one of the promising options due to its capability to propagate information in the network using link-state advertisements (LSA).

While traversing nodes in a network, ants store every nodes' wavelength availability information using a binary mask, M_{ant} [7]. This binary mask has W bits which correspond to the number of wavelengths in the network. A value on bit $W_i = 1$ means the wavelength λ_i is available, and 0 otherwise. M_{ant} is updated as follows,

$$M_{ant} = M_{ant} \text{ AND } M_{link} \tag{3}$$

As in (3), the M_{ant} is the actual mask which initially (at the source node) has all bits set to 1, implying that all the wavelengths are available. M_{link} is the available wavelengths mask on the next selected link along the path. AND is the binary logical operator.

C. Ant Movement in the Network

Let an ant move from a source s to a destination node d on the following path $(s, \dots, i-1, i, \dots, d)$. At time t, if an ant reaches an intermediate node $(i-1)$, it increases at time $t+1$ the selection probability in its pheromone table of destination node s corresponding to the ant source node s using (4).

Now, assume that $n1$ and $n2$ are two neighboring nodes of node $(i-1)$. The selection probability of node $n1$ and $n2$ is decreased with respect to reaching destination s using (5). Next step is to forward the ant to the next node towards destination d, which is the one with the highest selection probability. This procedure is followed by all the ants until they get to their final destination, d. On arriving at d, the probability of its neighboring node $(i-1)$ in the pheromone table with respect to s as its destination is increased. If d had other neighboring nodes, the selection probability of those nodes would also be reduced using (5) in the pheromone table of d with respect to s as their destination, as explained above.

$$r^i_{i-1,s}(t+1) = r^i_{i-1,s}(t)\frac{\delta r}{(1+\delta r)} \tag{4}$$

$$r^i_{n,s}(t+1) = r^i_{n,s}(t)\frac{1}{(1+\delta r)}, n \neq i-1 \tag{5}$$

In (4) and (5) (taken from [7]), δr is the amount of pheromone trail on a network path and it is computed as,

$$\delta r = \frac{\alpha}{\delta l} + (1-\alpha)\delta w, \tag{6}$$

where, α is a scalar parameter which can be used for adjusting the emphasis of δl and δw. δl corresponds to the length of the path traversed by the ant until now and δw corresponds to the percentage of free wavelengths on the corresponding path. The values of δl and δw are computed as given in (7),

$$\delta l = \beta(e^{(-1.0/l)} - e^{(-1)}) \text{ and } \delta w = e^{(\gamma w)} - 1, \tag{7}$$

where w corresponds to the percentage of free wavelengths on the path [7]. The design parameters, α, β, and γ can be adjusted to improve the performance of the protocol [8]. Specifically, from our analysis, we have experienced that these particular setting, $\alpha = 0.3$, $\beta = 50$, and $\gamma = 0.2$, provided the best performance on a wide variety of network topologies and traffic scenarios under consideration.

In ACO, an ant selects its neighbor to get to the destination according to the selection probability values computed in the pheromone table. In the algorithm, a visited node will never be visited again for the same connection request to avoid loop formation in the network. In order to avoid making only one path constantly attractive for the ants, a random scheme is introduced wherein each ant selects its next hop with an exploiting probability, P_{noise} [7]. Moreover, every ant in a network has a time to live (TTL), after which it is killed regardless of whether or not it reaches the destination. The TTL value also avoids having too many ants in the network, thus reducing the control overhead in the network. Finally, ants are destroyed once they reach the destination node.

IV. Dynamic ACO-based Anycast Routing and Wavelength Assignment

In this section, we discuss the ACO based routing algorithm for dynamic anycast RWA in WDM networks.

Ants are sent from each source node to all possible destination nodes in the network during the ant foraging phase (see functions *AntGeneration()* and *AntForaging()* in Algorithm 1). This initial phase ensures that we fill in the *P-route* table with enough candidate paths. The ant generation and launch probability ρ (line 5) allows to parameterize the control overhead generated by the number of ant packets running concurrently on the network.

After this initial phase, the dynamic anycast connection requests start arriving into the network (see function *RoutingAndWavelengthSelection()*). When a connection request arrives from a source s to a destination set $(d_1, d_2, \ldots, d_m)$, then the algorithm looks into the list of all the paths present in the *P-route* table of s, with respect to every candidate destination d_i. The top k paths are chosen based on the highest goodness value for every candidate destination in the above set (line 26). Afterwards, needle ants packets are sent over these k paths, from the source s to each candidate destination d_i to check if at least one wavelength is available on those k paths (line 27). As the needle ants move along a path, they also keep track of the length of that path and as soon as they reach the destination d_i this information is stored in the node. From these k paths, the preferable shortest path would be the one that has the highest goodness value and has at least one wavelength available (under WCC). In the case needle ants do not find a wavelength on the path that has the highest goodness value, the path with the next-highest goodness value will be chosen if at least one wavelength is available. If the chosen path L_i has more than one wavelength available, a random wavelength assignment among the free wavelengths is executed by d_i (line 31). An acknowledgement is then sent back to the source. This procedure is followed by every node in the destination set in order for the request's source node to calculate a set of lengths for every candidate destination $(L_1, L_2, \ldots, L_m)$ (line 39). As soon as the shortest route is chosen for the connection request, the selected wavelength and path will be reserved by the new arriving connection (line 40).

To derive the computational complexity of the algorithm we must take into account the different functions involved. Consider the network to be modeled as $G = (N, E, W)$, where N is the number of nodes (vertices), E represents the set of links and W the number of wavelengths. Moreover, the *P-route* tables take K paths per destination. On the one hand, the ant foraging, which is solely used to compute possible paths between a source and candidate destinations, implies a complexity of $O(N)$ since ants can run the update functions in a constant time ($O(1)$) but for all the nodes on the path, which at the most can be N. On the other hand, the routing and wavelength selection per destination adds a complexity of $O(KWN^2 + NK \log K + N^2)$, which comes from: for each K possible path, the needle ant will update the wavelength availability bit array, W, for all the nodes in the destination set and at every node in the corresponding path, N^2. Then, at every anycast destination node, which at the most can be in the order of N, one of the K paths and wavelength needs to be chosen, which can be realized in $O(K \log K)$. And the final lightpath selection by the source node can be done with a simple cocktail sorting algorithm with a worst case complexity of $O(N^2)$. Overall, the complexity can be further simplified as $O(N + KWN^2 + NK \log K + N^2) \sim O(KWN^2)$.

V. Simulation Results and Analysis

An event-based C++ simulator was developed to simulate the performance of the proposed anycast extensions to ACO.

Algorithm 1 ACO Algorithm for Dynamic Anycast RWA.

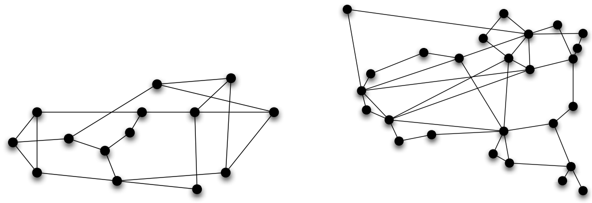
```
1: Function: AntGeneration()
2: repeat
3:     for each node in the network do
4:         select a random destination
5:         launch ants to this destination with probability ρ
6:     end for
7:     increase the time by a time-step for ants' generation
8: until end of simulation
9:
10: Function: AntForaging()
11: for each ant from source s to d (in parallel) do
12:     while current node i ne d do
13:         update routing table elements
14:         push trip's state into the stack
15:         if (found a next-hop) then
16:             move to next-hop
17:         else
18:             kill ant
19:         end if
20:     end while
21: end for
22:
23: Function: RoutingAndWavelengthSelection()
24: for each anycast request do
25:     for each anycast candidate node destination do
26:         select k paths from P-route table starting from source to destination
27:         send needle ant packets over the k paths between source and destination
28:     end for
29:     for at every anycast destination node do
30:         choose path with highest goodness value having one free wavelength and
            track the path length
31:         if more than one free wavelength is available on a path then
32:             choose one free wavelength randomly
33:             acknowledge the source with the path and free wavelength chosen
34:         end if
35:         if no wavelengths are available then
36:             acknowledge to source that no path can be set
37:         end if
38:     end for
39:     source s chooses the destination with the shortest path length
40:     setup lightpath over the chosen path and wavelength
41: end for
```

A dynamic traffic model is used in the simulation experiments where connection requests arrive at each network node according to a Poisson process with an arrival rate of λ (connections/second). Simulations run over a maximum of 10^6 connection requests, after which results are gathered and averaged. Furthermore, in order to assess the performance on different network scenarios with different characteristics, two network topologies have been used: the well-known NSFNET (Fig. 2a), and the 27-node AT&T network (Fig. 2b). To evaluate the performance of the proposed anycast ACO algorithm, we compared it versus the fixed shortest path first (SPF) routing algorithm and the dynamic load-balanced shortest-path (LB-SPF) algorithm [12]. The LB-SPF algorithm uses the following link weight function,

$$W_{(i,j)} = \rho(i,j) + \frac{d_{(i,j)}}{d^{MAX}}, \tag{8}$$

where $\rho(i,j)$ is the average load on the link (i,j), $d_{(i,j)}$ is the physical distance of link (i,j), and d^{MAX} is the maximum link distance in the network (refer [12] for details).

The parameterizations of the ACO functions used throughout the simulations is as follows: $\alpha = 0.3$, $\beta = 50$, $\gamma = 0.2$, $\rho = 0.6$, $\phi = 0.996$ and $P_{noise} = 0.06$. Based on extensive simulations, we found that these values provide a good compromise between performance and protocol control overhead.



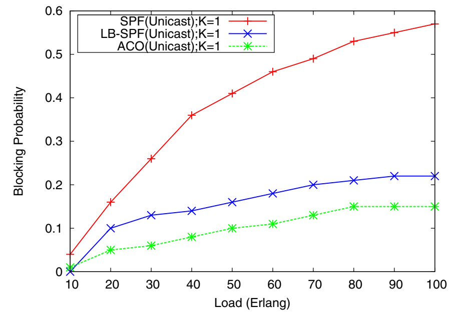
(a) (b)

Fig. 2. Simulation networks (a) 14-node NSFNET, and (b) 27-node AT&T network.

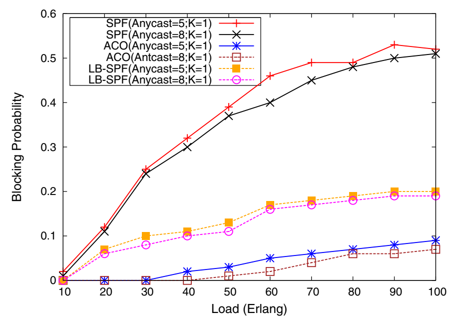
In Fig. 3a we can see the blocking probability as a function of the offered load to the network in the case of unicasting (with K=1) for SPF, LB-SPF, and ACO, on the 14-node NSFNET with 8 wavelengths. Although, the blocking probability increases with the increasing load in the network as expected, the blocking probability for ACO even at higher loads is significantly lower compared to SPF and LB-SPF. The relative improvement in blocking probability using ACO at 100 Erlang compared to SPF and LB-SPF is approximately 77% and 38%, respectively.

Similarly, Fig. 3b plots the blocking probability versus the offered traffic load for anycasting (with K=1) with 5 and 8 candidate destinations. Results are gathered on the same NSFNET with 8 wavelengths. As expected, increasing the number of nodes in the destination set (from 5 to 8), gives slightly better results due to the improved chances to find a destination node in the larger candidate set. Overall, in this case, the relative decrease in blocking probability achieved by ACO (at 100 Erlang load) in comparison with SPF and LB-SPF is approximately 80% and 50%, respectively. These two plots confirm that the ACO-based algorithm is better than SPF and LB-SPF, not only for the anycasting case, but also for unicasting, which corroborates past results.

Now that we have observed the performance improvement of the anycast ACO algorithm in terms of blocking probability, it might also be interesting to check other network parameters. Fig. 4a shows the blocking probability using anycasting with 5 candidate destinations but now with K=2 (i.e., 2 possible paths per destination). Similar to the previous graphs in Fig. 3b, ACO performs consistently better both for unicasting and anycasting with different values of K. The blocking probability at 100 Erlang with 8 wavelengths in the network is in this case, 50%, 18%, and 9% for SPF, LB-SPF, and ACO, respectively. The relative improvement in blocking probability for ACO when compared to SPF is 82% and that of LB-SPF is 50%. Moreover, Fig. 4b plots the average hop-count versus the offered network load for the connection requests. As the load in the network increases, the average hop-count decreases for the three analyzed algorithms due to the fact that connection requests that have fewer number of hops are the ones established successfully when the resources on the network become scarce. It is worth noting that the average path length in anycast tends to be shorter than in unicast, since the destination node is chosen among the candidate destination set. Finally, Fig. 4c plots the blocking probability versus the



(a)



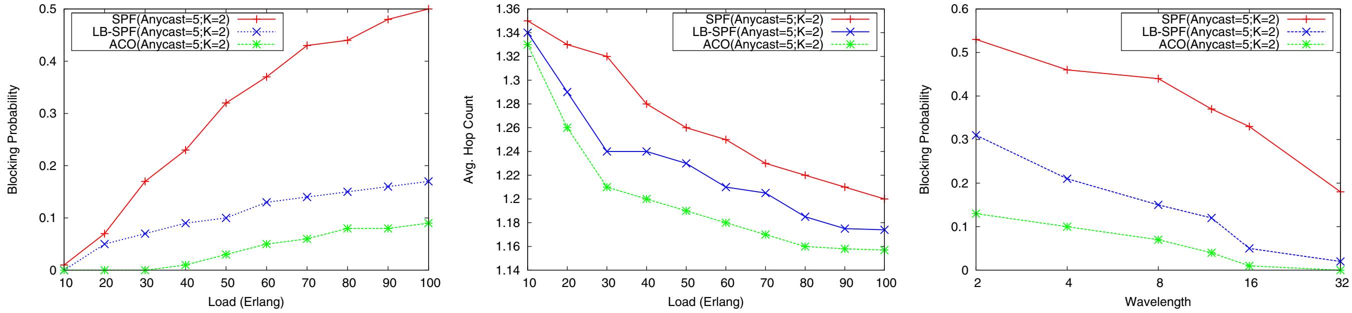
(b)

Fig. 3. Results on the NSFNET and 8 wavelengths for (a) unicast, and (b) anycast with 5 and 8 possible destinations.

number of wavelengths on network. We are aware that as the resources in the network increase, the blocking probability is correspondingly reduced. When the number of wavelengths in the network is 32 with a load of 80 Erlang, the blocking probability for SPF, LB-SPF, and ACO are 19%, 3%, and 0%, respectively. We can clearly observe that ACO outperforms both SPF and LB-SPF.

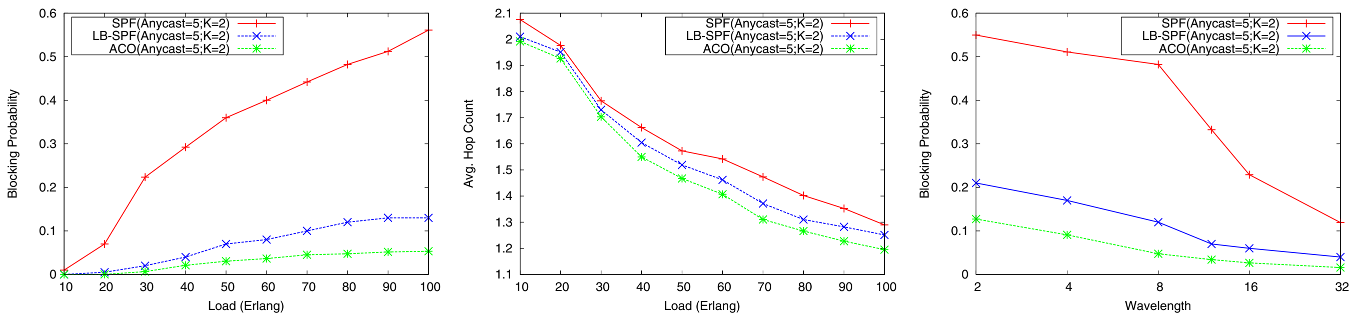
As introduced above, it is important to also assess the protocols on different network topologies. To this end, Fig. 5 shows the results obtained from the larger 27-node AT&T network in the case where K=2 and the number of anycast candidate destinations is 5. Interestingly, we can observe that ACO, again, outperforms SPF and LB-SPF in terms of blocking probability (see Fig. 5a), although in this case, LB-SPF manages to get closer to ACO. ACO outperforms SPF and LB-SPF by reducing the blocking probability by approximately 90% and 58%, respectively. ACO also manages to get shorter paths to the anycast destination as shown in Fig. 5b. Finally, Fig. 5c shows the blocking probability versus number of wavelengths on each link. Again, we observe that as the number of wavelengths in the network increase, the blocking probability reduces considerably. When there are 32 wavelengths in the network, with 80 Erlang load, the blocking probability for SPF, LB-SPF, and ACO are approximately, 12%, 3% and 1%, respectively. The ACO algorithm performs better whatever the number of wavelengths in use, even offering good results by using very few wavelengths.

In summary, it can be observed from all the above simulation results that ACO significantly outperforms SPF and LB-SPF algorithms. ACO helps to find optimal paths in the



(a) (b) (c)

Fig. 4. Results on the NSFNET for anycast with 5 destinations and K=2. (a) Blocking probability for 8 wavelengths, (b) Average path hop-count for 8 wavelengths, and (c) Blocking probability as a function of the number of wavelengths.



(a) (b) (c)

Fig. 5. Results on the AT&T network for anycast with 5 destinations and K=2. (a) Blocking probability for 8 wavelengths, (b) Average path hop-count for 8 wavelengths, and (c) Blocking probability as a function of the number of wavelengths.

network for establishing dynamic connection requests successfully by making use of the available resources more efficiently (i.e,. fewer number of wavelengths is required to get similar results). It is also worth noting that ACO can increase the convergence time on very large networks. However, in our particular case scenario, wide-area networks are typically in the order of tens of nodes (rarely above hundred), which makes ACO a good candidate over other RWA solutions.

VI. Conclusion

In this paper, we have proposed the use of an ACO-based algorithm to solve the anycast RWA problem on optical WDM networks. It has been demonstrated that the ACO-based mobile agents approach is an effective technique in finding optimal paths. We have presented the routing mechanisms and table structures used by the ACO algorithm and how they get updated depending on the changing network states using a suitable number of ants that continuously explore the network. Simulation results show that the ant colony optimization-based dynamic RWA algorithm significantly outperforms the traditional shortest path first algorithm for both unicast and anycast requests, and also performs significantly better than the dynamic load-based routing algorithm.

Acknowledgement

This work was supported in part by the National Science Foundation (NSF) under grant CNS-0626798 and Department of Energy (DOE) COMMON project under grant DE-SC0004909. Joan Triay is a visiting researcher at University of Massachusetts Dartmouth with the support of a Fulbright graduate fellowship.

References

[1] R. Ramaswami and K. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Trans. Netw.*, vol. 3, no. 5, pp. 489–500, Oct. 1995.

[2] H. Zang, J. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Networks Magazine*, vol. 1, no. 1, pp. 47–60, 2000.

[3] D. Din, "A hybrid method for solving ARWA problem on WDM network," *Computer Commun.*, vol. 30, no. 2, pp. 385–395, Jan. 2007.

[4] M. Dorigo and L. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[5] G. D. Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communications networks," *J. of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998.

[6] R. Garlick and R. Barr, "Dynamic Wavelength Routing in WDM Networks via Ant Colony Optimization," in *Proc. of ANTS 2002*, ser. LNCS, vol. 2463, 2002, pp. 27–41.

[7] S. Ngo, X. Jiang, and S. Horiguchi, "An ant-based approach for dynamic RWA in optical WDM networks," *Photonic Netw. Commun.*, vol. 11, no. 1, pp. 39–48, 2006.

[8] F. González and et al., "Lightpath routing and wavelength assignment by means of ant colony optimization," in *Proc. of ONDM 2003*, vol. 2, Budapest, Hungary, Feb. 2003, pp. 855–864.

[9] S. Demeyer, M. D. Leenheer, J. Baert, M. Pickavet, and P. Demeester, "Ant colony optimization for the routing of jobs in optical grid networks," *J. of Optical Netw.*, vol. 7, no. 2, pp. 160–172, 2008.

[10] S. Ngo, X. Jiang, and S. Horiguchi, "Ant-Based Alternate Routing in All-Optical WDM Networks," *IEICE Trans. on Communications*, vol. E89-B, no. 3, pp. 748–755, Mar. 2006.

[11] D. Katz and K. Kompella, "RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2," IETF, Sep. 2003.

[12] G. Thodime, V. Vokkarane, and J. Jue, "Dynamic congestion-based load balanced routing in optical burst-switched networks," in *Proc. of GLOBECOM 2003*, vol. 5, San Francisco, CA, USA, Dec. 2003, pp. 2628–2632.