Routing and Wavelength Assignment with Crankback Re-Routing Extensions by Means of Ant Colony Optimization

Gustavo Sousa Pavani, Member, IEEE, and Helio Waldman, Senior Member, IEEE

Abstract—Crankback re-routing extensions can offer significant improvements in the successful setup of Label Switched Paths (LSPs) by allowing new retries on alternate paths that circumvent blocked links or nodes. These extensions can be incorporated into a fully-distributed algorithm based on Ant Colony Optimization metaheuristics, taking advantage of its self-adapting, emergent behavior. By comparing with traditional fixed-alternate re-routing mechanisms, simulations have demonstrated that the proposed algorithm can efficiently mitigate lightpath blocking, both during normal operation and in case of network failure, by locally repairing failed LSP setups due to blocked/failed resources.

Index Terms—Ant Colony Optimization, GMPLS Control Plane, Routing and Wavelength Assignment, Optical Networks.

I. INTRODUCTION

GENERALIZED Multiprotocol Label Switching (GMPLS) control plane plays an important role to address the complexity in the management and operation of highly-dynamic, reconfigurable optical networks. For instance, it provides a standardized way to support end-to-end lightpath provisioning for routing and wavelength assignment (RWA) algorithms.

Nevertheless, during lightpath provisioning in wavelength-routed networks without conversion capabilities, a lightpath setup can be blocked due to lack of network resources. In addition, when a network failure occurs, an affected lightpath might not be restored if there is insufficient network resources.

A crankback mechanism can be used in these cases to return setup failure information to allow new setup attempts. Many types of networks have used crankback information to reduce call blocking, notably ATM [1], [2], and recent efforts were made to incorporate crankback schemes into RSVP-TE [3].

Thus, crankback information can be obtained from the GMPLS control plane in order to reissue the lightpath establishment to circumvent the blocked/failed resource (link or node).

The proposal of this paper is to present an algorithm based on the Ant Colony Optimization (ACO) metaheuristics with crankback re-routing extensions to reduce the blocking experienced by the RSVP-TE protocol [4] during lightpath setup or restoration. By using local pheromone levels, a Label Switched Path (LSP) setup can be retried on a different route to avoid the blocked resources. Prior works [5], [6] relied on a simple, greedy heuristics to establish the lightpaths, which neither allowed any setup reattempt nor took full advantage of the node’s local information (pheromone levels). The use of this novel crankback scheme achieves a much higher performance, in terms of blocking probability, both under normal operation and under failures of the network.

The indirect communication characterized by modifications in the environment to stimulate subsequent actions is called stigmergy [7] and was first observed in ants. In fact, an ant may leave a type of odor, which is called pheromone, that influences the behavior of nearby ants.

Stigmergy is the base of ACO algorithms, allowing for an emergent and self-organizing behavior of the ant colony even tough ants are very simple agents with limited capabilities. ACO mimics the foraging behavior of natural ants, which can efficiently find a good route from a food source to their nest while avoiding nearby obstacles. In fact, it has been successfully applied in a large number of difficult routing problems [5], [8], [9].

Indeed, the proposed ACO algorithm is an alternative to link-state algorithms [10]–[12], such as OSPF-TE [13], [14]. The artificial ants probe the routes by collecting an identification of the traversed nodes and the available wavelengths in its forward direction and then locally update the same traversed nodes in the backward direction. The information laid down by the ants in the nodes, stored under the form of artificial pheromone levels, can be efficiently used to route the lightpaths in the network.

Simulations were carried to evaluate the performance of the ACO algorithm with and without crankback re-routing extensions when compared to a traditional topologically-driven approach with end-to-end re-routing.

The remaining of the paper is organized as follows. Firstly, in section II, we briefly discuss some related works and the state-of-the-art in bio-inspired networking. In section III, we present our Ant Colony Optimization proposal for the RWA algorithm. Then, we discuss the GMPLS-based architecture considered throughout this paper in Section IV. In Section V, we detail the simulation studies carried out to properly

0733-8716/10/$25.00 © 2010 IEEE

Authorized licensed use limited to: Yuan-Ze University. Downloaded on May 22,2010 at 04:49:21 UTC from IEEE Xplore. Restrictions apply.
characterize the proposed algorithm. Results obtained are presented and discussed in Section VI. Finally, in Section VII, conclusions are drawn.

II. RELATED WORK

A multi-objective ACO algorithm that took into consideration both hop count and the number of wavelength converters for static routing and wavelength assignment in optical networks was presented in [15].

Other ant algorithms have been successfully applied in communication networks. For instance, in [16] a survey of the state-of-the-art on ant colony algorithms for routing and load-balancing was presented.

Stigmergy mechanism was proposed as a software design pattern to build distributed, self-organizing computer networks [17], with an example of the application of the pattern on routing in MANETs (Mobile Ad Hoc Networks).

In [18], an alternative to the ant algorithms was presented for multi-path routing in overlay networks, where the biological model of Adaptive Response Attractor Selection (ARAS) was used to achieve self-adaptiveness in the presence of failure or network load fluctuations.

Other biologically inspired techniques for networking can be found in literature. For instance, in [19], a routing algorithm with survivability capabilities was inspired by chemotaxis, reaction-diffusion and quorum sensing biological processes. An in-depth review of bio-inspired networking based on swarm intelligence, artificial immune system and cell and molecular biology can be found in [20], [21]. In [22], [23], a networking architecture and its services, which follows biological principles and mechanisms in their design, were presented. Finally, a bio-inspired framework for the provisioning of services in pervasive communication environments was shown in [24].

III. ANT COLONY OPTIMIZATION

ACO is used to refer to the class of algorithms that are inspired by the process of foraging for food of natural ants for the optimization of hard-to-solve problems or problems that need distributed control. It relies on artificial stigmergy [7], where a modification of the environment (pheromone levels) indirectly influence the artificial ants’ behavior. Although each artificial ant is individually very limited, the ant colony can collectively exhibit an intelligent, self-organizing behavior [25].

In ACO, by means of both iterative and parallel processes, each ant builds a solution using two types of local information: specific problem information and information added by the ants during previous iterations of the algorithm that are embodied in a single variable per network element and destination, which is called the pheromone level. In fact, while building the solution, each ant gathers information about characteristics of the problem to be solved and about its own performance and uses this information to modify the representation of the problem, i.e., the pheromone level at each node, as seen locally by other ants. In this way, the representation of the problem is modified in such way that the information contained in the previous solutions can be distributively explored to obtain better solutions.

ACO has an important feature for our work: it is explicitly modeled in terms of computational (generally mobile) agents, and is specially suited for routing in telecommunications networks [26]. A (computational) agent can be defined as a computational entity that acts autonomously and can react to external stimuli, to decide according to the environment and to cooperate with other agents. Mobile agents are agents that are capable to migrate autonomously in a network [27], [28].

The AntNet [8] framework is an ACO-based algorithm used for routing of packets in telecommunication networks, successfully adapted to be used in Wavelength-Routed Optical Networks (WRON) [5], [6] and Optical Packet-Switching (OPS) networks [29]. It follows the principles and design paradigms proposed in [25] to achieve self-organized network functions.

The original AntNet framework uses the delay introduced by each hop as the metric for routing. This is not applicable in circuit-switching networks, where the main metric is generally the number of hops. The minimization of the number of hops of a connection following some criteria is a very good heuristic to reduce the overall blocking probability in a network. Then, the ant used in the proposed algorithm will only carry on its memory the node identification of each node that it passes by.

At each intermediate node $i$, the following data structures have to be maintained in the AntNet framework:

1) Pheromone-routing table $T^i$: It is a matrix containing a row for each destination $d$ of the network and a column for each neighbor node $n$, for storing the pheromone values. The sum of each row must be equal to 1, i.e., $\sum_{n \in N_i} \tau^i_{dn} = 1$, where $N_i$ is the set of neighbors of node $i$, $\tau^i_{dn}$ defines the value of pheromone in the link $i \rightarrow n$ for the specified destination $d$ and estimates the probability of reaching $d$ given that $n$ is selected as the next hop. Figure 1 depicts an example of pheromone routing table of an example network, where the length of all links is equal to 1.

2) Statistical parametric model $M^i$: It is a table containing as entries the triplet $<\mu_d, \sigma_d, E_d>$ for each destination $d$ of the network, where $\mu_d$ is the average length of the path followed by the ant from the current node to destination $d$, $\sigma_d$ the standard deviation for this path length and $E_d$ the best value of path length found for this destination within the non-sliding window of $w$ observations [8]. Non-sliding means that when the number of observations reaches $w+1$, all the accumulated values are reset, i.e., $E_d = \mu_d = (\text{path length of the } (w+1)$-
st observation), \( \sigma_d = 0 \) and the observation counter receives 1.

The values of \( \mu_d^i \) and \( \sigma_d^i \) are updated using the following exponential model [30]:

\[
\mu_d^i \leftarrow \mu_d^i + \eta(o_{i \rightarrow d} - \mu_d^i)
\]

\[
\sigma_d^i \leftarrow \sigma_d^i + \eta(o_{i \rightarrow d} - \mu_d^i) - \sigma_d^i ,
\]

(1)

(2)

where \( o_{i \rightarrow d} \) is the new observation of the distance between nodes \( i \) and \( d \), and \( \eta \) is the factor of the exponential model, which weighs the number of the most recent observations that will influence the mean. The number of effective observations is approximately equal to \( 5/\eta \) [8].

The number of values in the window is calculated as \( w = 5(c/\eta) \), where \( c \in (0,1) \) is a reduction factor.

Therefore, the window is updated in a smaller interval than the one used for the mean and standard deviation estimates, in such a way that the value of \( E_d^i \) and these estimates refer to the same set of observations [8].

Figure 2 depicts an example of statistical parametric model of an example network, where the length of all links is equal to 1.

\( M \) and \( T \) store local state information of different aspects of the routing process. The model \( M \) maintains distance estimates to all nodes, while the pheromone routing table \( T \) gives the relative goodness of the next hop to reach a given destination.

The AntNet algorithm has two phases: solution construction and updating of data structures, which are detailed in the next sub-sections.

A. Solution construction

The algorithm starts with the initialization of the pheromone routing tables of each optical node. To speedup the convergence of the algorithms, we used the intelligent initialization of routing tables as described in [31].

Before starting the arrival of requests, only ants are launched to explore the network and populate the routing tables with topology information. In practice, this allows for the configuration of the routing tables with the shortest path in the absence of congestion. After a small amount of time \( I_{warmup} \), the lightpath requests start to arrive randomly at the network nodes.

The solution construction is made by the forward ants, which are launched at regular intervals \( (1/R_{ants}) \) from a random source node \( s \) to another random node \( d \). On its trip from \( s \) to \( d \), each forward ant selects the next hop \( (i+1) \) using a random scheme that accounts for the path selection probabilities, given by the pheromone levels \( \tau_{dn} \) in each neighbor link, and for a heuristics value \( h_n \), calculated from the availability of the neighbor links, as follows [6]:

\[
h_n = \left( \frac{n^a}{W} \right)^f,
\]

(3)

where \( n^a \) is the number of available wavelengths on neighbor \( n \), \( W \) is the total number of wavelengths deployed on the link and \( f \) is a power factor for enhancing the difference in the availability among neighbor nodes [6].

During its trip, the forward ant gathers the label of each node where it passes by, putting it in its memory \( V_{s-i} \), which also serves as tabu list [32].

If among the neighbor nodes of the node that is processing the ant, there are any not visited yet, the choice of the next hop is done using a random scheme and the probability for each candidate node \( n \) to be the next hop \((i+1) \) are given by the following expression [8]:

\[
p_n^d = \begin{cases} 
\left( \frac{1}{1+\alpha} \right) \sum_{k \in T} \tau_{dk} + \left( \frac{\alpha}{1+\alpha} \right) \frac{h_n}{\sum_{k \in T} h_k}, & \forall n \in T \\
0, & \text{otherwise},
\end{cases}
\]

(4)

where \( \alpha \) gives the emphasis between pheromone level (long-term memory) and instantaneous availability state (short-term memory), and \( T = N_i \setminus V_{s-i} \).

However, if all neighbor nodes have already been visited \( (T \) is empty), this indicates that the ant entered a loop. We ignore the heuristic correction given by the link congestion, choosing the next node in a random way, where the probability of each candidate node \( n \) to be the next hop \((i+1) \) is given by the following expression:

\[
p_n^d = \begin{cases} 
\frac{\tau_{dn}}{\sum_{k \in T'} \tau_{dk}}, & \forall n \in T', \text{ if } |N_i| > 1 \\
1, & \text{if } N_i = \{v_{i-1}\} \\
0, & \text{otherwise},
\end{cases}
\]

(5)

where \( v_{i-1} \) represents the last visited node and \( T' = N_i - \{v_{i-1}\} \).

In this case, after the selection of the next hop, all labels that belong to nodes of the cycle are removed from the ant’s memory.

If the ant does not reach its destination node in a number of pre-established hops, it is dropped. This avoids lost ants circulating forever in the network.

B. Updating of data structures

When the forward ant arrives at \( d \), it becomes a backward ant \( B_{d \rightarrow s} \) and returns to \( s \) using the same path followed by the forward ant, but in the opposite direction. At each intermediate node \( i \), it updates the local parametric model \( M^i \) and, after it, the pheromone routing table, for all entries relative to \( d \).

Moreover, this update is also made for all nodes \( d' \in V_{i \rightarrow d} \), \( d' \neq d \) in the sub-paths \( (i \rightarrow d') \) traversed by the forward ant \( F_{s \rightarrow d} \) after visiting \( i \). If this sub-path is statistically good, then the entries of \( M^i \) and \( T^i \) relative to \( d' \) are also updated. This allows for the updating of good paths found by ants that were not intended to those destinations.
A sub-path is considered statistically good if \( \text{dist}(V_{i-d'}) < I_{sup}^d [8] \), where \( \text{dist()} \) is a function that gives the distance, in terms of number of hops, of the path followed by the ant and \( I_{sup}^d \) is a superior estimate calculated from Tchebycheff’s inequalities, which permit the definition of a confidence interval of a random variable that follows any kind of distribution. The inferior estimate is equal to \( E_d \). The superior estimate can be expressed by the following formula:

\[
I_{sup}^d = \mu_d + \frac{1}{(1 - \gamma)} \sqrt{\frac{\sigma_d^2}{w}},
\]

where \( \gamma \) is the confidence level coefficient.

Thus, the local parametric model is updated using Equations 1 and 2, where \( o_{i-d} = \text{dist}(V_{i-d}) \). If \( o_{i-d} < E_d \), then \( E_d' \leftarrow o_{i-d} \). The same process is repeated for all \( d' \), whose sub-paths were considered statistically good.

After the updating of the parametric model, an adaptive reinforcement \( r_d \) is calculated for the updating of the routing table [8]:

\[
r_d = c_1 \left( \frac{E_d}{\text{dist}(V_{i-d})} \right) + c_2 \left( \frac{I_{sup}^d - E_d}{(I_{sup}^d - E_d) + (\text{dist}(V_{i-d}) - E_d)} \right)
\]

The first term of Eq. 7 simply evaluates the ratio between the best route within the non-sliding observation window and the distance traversed by the ant. The second term evaluates how far is this distance from the confidence interval. It is important to note that the second term must be considered equal to zero when \( \text{dist}(V_{i-d}) = I_{sup}^d = E_d \). The \( c_1 \) and \( c_2 \) coefficients weigh the importance of each term.

The obtained \( r \) is limited to 0.9 to avoid stagnation and its value is “squashed” using the following expression:

\[
r_d = s(r) - s(1) \quad \text{where} \quad s(x) = \left(1 + \exp \left(\frac{a}{x|N|}\right)\right)^{-1},
\]

where \( a \) is an amplifier coefficient.

Now, if the neighbor node \( m \) is on the path, then it receives a positive reinforcement:

\[
\tau_{dm} \leftarrow \tau_{dm} + r_d(1 - \tau_{dm})
\]

On the other hand, the other nodes receive a negative reinforcement:

\[
\tau_{dn} \leftarrow \tau_{dn} - r_d\tau_{dn}, \forall n \in N_i, n \neq m.
\]

As already done for the parametric model, the updating process is also repeated for all \( d' \) considered statistically good.

Fig. 3 depicts an example of the updating process by a backward ant \( B_{E-A} \), whose memory is \( V_{A-E} = \{A; D; E\} \). At node D, for destination node E, the backward ant updates the parametric model and reinforces positively the neighbor node E, and negatively the neighbor nodes A and C.

Finally, at network node A, for destination node E, the backward ant updates the parametric model, and reinforces positively the neighbor node D and negatively the node B. Also, for destination node D, which belongs to a sub-path traversed by the forward ant after node A, if \( \text{dist}(V_{A-D}) = 1 < I_{sup}^D \), then the same update process is repeated for the entry relative to destination node D. Node D is marked with a star to indicate that this entry is updated only if it is considered statistically good.

Although the parametric model does not have link availability information, the effect of a higher number of forward ants that choose the path with a link less loaded results in a higher reinforcement of the paths with less blocking probability.

IV. ARCHITECTURE

In our GMPLS architecture, there is a logical one-to-one relationship between an optical node and a Label Switching Router (LSR). Each optical node is Lambda-Switch Capable (LSC) only. For this reason, we also have a one-to-one relationship between a lightpath and a LSP.

Also, each node is equipped with control channels for passing messages/packets/ants in the control plane. These control channels may be implemented using out-of-band signaling.

In our original adaptation of the AntNet framework to wavelength-routed optical networks [5], [6], we used a simple, greedy approach: the ACO routing was done in a hop-by-hop basis, where the next hop of the connection was chosen evaluating the entry of the pheromone routing table that matches the destination. The neighbor with the highest level of pheromone was chosen as the next hop of the Path message. In case of lack of network resources or loop, the setup was blocked.

Although this simple heuristics achieved good results, it does not mimic the behavior of real ants. Indeed, real ants try to circumvent obstacles in their path.

By associating our ACO-based RWA algorithm with a crankback re-routing scheme, we can efficiently use local state information (pheromone levels), thus achieving lower levels of blocking probability, to route around the blockages.

In addition to the pheromone table, each LSR must maintain a history table, with a limited number of entries per LSP, that records each LSP setup attempt [3]. This number of entries determines the maximum number of local LSP re-routing attempts made by the LSR. They are discarded either if the LSP is successfully established by the LSR or if a local timeout has expired.

In the case of ACO routing, any LSR can be a repair point, i.e., any LSR can be responsible for resuing the setup request. This is known as segment-based re-routing [3].

When a LSP request is blocked due to unavailable resources, an error message (PathErr) is returned to the previous node. If there is, at least, one free entry in the history table for that LSP and a link not visited yet (information obtained from the history table), the LSR will make another attempt to establish the LSP.
For instance, a history table with two entries means that a LSP setup can be retried one more time, if the first attempt is blocked. Three entries mean two local re-routing attempts for a given LSP can be made. And a history table with just one entry means that the node cannot reissue a blocked LSP setup, i.e., it is similar to the proposal in [5], [6].

The history table in each node correlates the LSP establishment attempt (by using information present on the SESSION and SENDER TEMPLATE objects, for instance) and the blocked neighbor link. Since this history table is local to the node, there is no need to globally identify the blocked links or nodes, and their location. In case of re-routing, the next hop to be selected is the neighbor not present in the history table, with the highest level of pheromone, given the appropriate destination.

Fig. 4 illustrates the crankback mechanism used by the ACO algorithm for establishing LSP 1234 from node A to node C. At node A, all entries are free for LSP 1234. The neighbor with the highest level of pheromone (node B) is chosen as the next hop and added to the history table (a). At node B, all entries are also free for LSP 1234, then the neighbor with the highest level of pheromone is selected (node C) and added to the history table (b). However, there is not any free wavelengths for link B-C. Then, node B send back to node A a PathErr message indicating blocked resource at node B (c). Node A reissue a new lightpath request using node D as the next hop, because it is the neighbor node with both the highest level of pheromone for destination C and has no entry at the history table for LSP 1234 (d). Finally, node D is also added to the history table of node A for LSP 1234 (not shown for sake of clarity).

If the number of retries of a particular node is exceeded, then the node has to send a PathErr message, which indicates that it has abandoned crankback re-routing, in the upstream direction until a node that is capable to make further re-routing attempts is found. Finally, if the message reaches the ingress node and no more routing attempts can be made, the LSP setup is declared as failed.

In addition, in order to allow for the signalization of wavelength assignment algorithm, the Label Set object for each LSP attempt has to be recorded before the Path message is forwarded to the next hop. As happened with the entries in the history table, this information is discarded after a predetermined time or when the respective Resv message comes back from the destination node.

In the case of fixed, i.e., shortest-path, and fixed-alternate, i.e., shortest path plus alternatives, routing [33], the ingress LSR might attempt a re-routing, when a LSP establishment fail due to resource unavailability, until the number of retries are reached. A new retry follows the next route in the ordered list of fixed routes given by the k-shortest path algorithm [34], i.e., the algorithm starts with the shortest-path, then the second-shortest-path and so on.

In both cases, all crankback extensions are compliant with RSVP-TE and RFC 4920 recommendations. Because of that, the improvements for the signaling protocol that are found in literature, such as [35]–[38], can also be used.

A. Restoration

Restoration is the process of recovering connections or routes affected by a failure, using the free capacity of the network after the failure occurrence.

The use of end-to-end recovery might be the only way to recover from LSP failure in GMPLS-controlled optical networks without wavelength conversion capabilities [3]. Also, end-to-end restoration tends to be more efficient [39].

Moreover, for this work, we have considered only the full re-routing of LSP [40], since it tends to be more efficient in terms of resource usage and more robust in case of multiple failures, but we assumed that the time for calculating alternative paths can be neglected.

The resources of the disrupted LSP are freed before the establishment of the alternative LSP, because of its ease of implementation. This approach is known as “break-before-make” [41].

When a link failure occurs, it takes some time \( I_{detect} \) to detect and localize the failure. A node failure is a special case of multiple link failures, where all links connected to the node fail simultaneously.

The nearest upstream node to the failure have to notify the ingress node(s) of the LSP(s) affected by the failure, sending a PathErr message, which contains an indication of the failure and an indication to free resources of the disrupted LSP.

The nearest downstream node to the failure have to send a PathTear message towards the egress node(s) of the LSP(s) affected by the failure, to free the resource allocated to these LSP(s).

The re-routing of the LSP is done by the ingress node in the same way as a new LSP request, after the arrival of the PathErr message.

In the case of ACO routing, the crankback information is used to calculate the new path and the pheromone routing tables are updated after a network failure as described in [42]. The main difference between [42] and this work is the following: the former try to send ants between the nodes affected by the failure to get a more accurate information about the new state of the network and, then, try to restore the affected lightpaths. This work, on the other hand, rely
on local crankback information, trying to recover from the failure(s), even in the presence of inaccurate information in the pheromone routing tables. Thus, the latter approach tends to restore the lightpaths more promptly, because it does not need that the ants explore new routes in the network to re-route the lightpaths affected by the failure(s).

In the case of fixed and fixed-alternate routing, the nearest upstream/downstream nodes of the failure notify the rest of the network about the topology change. Therefore, when the ingress node has to re-route the LSP, the network has already knowledge of the new topology.

To compare the performance of both algorithms, we used the restorability metric \( R \), which is defined by the following expression [43]:

\[
R = \frac{C^\text{restored}}{C^\text{failed}},
\]

where \( C^\text{failed} \) is the total number of lightpaths affected by the failure(s) and \( C^\text{restored} \) is the number of lightpaths affected by the failure and restored by the recovery process.

When a failure takes place in a node that is the source or destination of one or more lightpath, the lightpath can never be restored. In other words, the restorability can never be equal to 1, if the node directly affected by the failure is the source or destination of one or more lightpaths.

V. SIMULATION

For testing the proposed algorithm, we used two networks in our simulations. The first one is the NSFNet backbone network that is shown in Fig. 5. It is a 14-node network with 21 bidirectional links and it is well-balanced [8], with average shortest-path length between all pairs of nodes equal to 2.2 hops.

The other network tested is the NTTNet (Nippon Telephone and Telegraph) backbone, which is shown in Fig. 6. It is composed by 57 nodes and 81 bi-directional links and is not a well-balanced network [8], with average shortest-path length between all pairs of nodes equals to 6.5 hops.

For this work, we have developed a custom event-driven simulation software, written in Java.

We have considered a homogeneous, Poissonian traffic with uniform spatial profile. The duration of each lightpath has followed an exponential distribution with a mean value of 100 s. The simulations were carried out with 8 wavelengths per link.

All algorithms used a First-fit approach for the wavelength assignment sub-problem [6]. The number of hops allowed for an ant or a signaling message is equal to the number of links in the network.

The parameters used in the simulations are depicted in Table I:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate for launching forward ants</td>
<td>( R_{\text{ants}} )</td>
<td>100 ants/s</td>
</tr>
<tr>
<td>Interval to start the lightpath requests</td>
<td>( I_{\text{warmup}} )</td>
<td>1000 s</td>
</tr>
<tr>
<td>Power factor for availability calculation</td>
<td>( f )</td>
<td>3</td>
</tr>
<tr>
<td>Correction for routing of forward ants</td>
<td>( \alpha )</td>
<td>0.5</td>
</tr>
<tr>
<td>Weight of the window’s exponential model</td>
<td>( \eta )</td>
<td>0.005</td>
</tr>
<tr>
<td>Reduction for parametric model’s window</td>
<td>( c )</td>
<td>0.3</td>
</tr>
<tr>
<td>First weight of the adaptive reinforcement</td>
<td>( c_1 )</td>
<td>0.7</td>
</tr>
<tr>
<td>Second weight of the adaptive reinforcement</td>
<td>( c_2 )</td>
<td>0.3</td>
</tr>
<tr>
<td>Confidence level for reinforcement</td>
<td>( \gamma )</td>
<td>0.65</td>
</tr>
<tr>
<td>Amplifier of the squash function</td>
<td>( a )</td>
<td>5</td>
</tr>
<tr>
<td>Interval to detect and isolate the failure</td>
<td>( I_{\text{detected}} )</td>
<td>10 ms</td>
</tr>
</tbody>
</table>

Table I: Parameters used in the simulations.

The ant launching rate is tuned based on the dynamics of the network. In our case, a value of 100 ants/s was a good value for both networks and increasing its value does not improve the obtained results. In addition, this work, which uses a crankback mechanism, is more efficient when an inadequate ant launching rate is used, since it is more robust to inaccurate information than previous works.

The robustness of parameters \( \alpha, c_1, c_2 \) and \( a \) were evaluated in [5], [8] for the normal operation of the network and in [42] for the failure conditions. There is not much difference when the results obtained keeps \( a \) between 0.5 and 0.7, \( c_1 \) between 0.6 and 0.7, \( c_2 \) between 0.3 and 0.4 or, \( a \) between 5 and 7.

The influence of the parameter \( f \) was evaluated in [6]. Any value of \( f \) between 2.5 and 5 should give similar results.

The rest of parameters are based on the evaluations found in [8].

VI. NUMERICAL RESULTS

First of all, the ACO-based algorithm is compared with a traditional fixed-alternate routing scheme, where the source node undertakes the re-routing job [5]. Each plotted point has a 95% confidence interval for the mean blocking probability of 10 different runs, where each run executed \( 10^5 \) lightpath requests.

Fig. 7 shows that the ACO algorithm with crankback extensions can outperform the fixed-alternate approach in terms of blocking probability until a 60-Erlang load for the NSFNet network. Although the ACO algorithm presents a higher blocking probability for higher loads, it is still competitive when compared to the performance of the fixed-alternate routing.

The same simulations were also carried using the NTTNet network and the results are shown in Fig. 8. Since the NTTNet network provides a much higher number of alternate paths than the NSFNet and the ACO algorithm with crankback re-routing capabilities can intelligently explore those possible paths, observe that the proposed algorithm has a significantly superior performance when compared to fixed-alternate routing.

In both scenarios, there is very little performance improvement for the ACO algorithm when the maximum number of
local re-routing attempts is greater than one, i.e., when we have more than two history table entries per LSP in each node. Note that only an intermediate node with node degree greater or equal to four, or an ingress node with node degree greater or equal to three may take advantage of more than one re-routing attempt per node. Since both networks have an average node degree around three, with few nodes with node degree equal to four, allowing more than one re-routing attempt per node has very little impact in the overall blocking performance.

Afterward we evaluate the impact of the crankback routing in the number of hops for the established lightpaths. In fact, a higher number of hops implies a higher delay for establishing
In Fig. 9, we can observe that the association of crankback re-routing with ACO-based RWA for achieving lower levels of blocking probability has a relative small impact in the number of hops needed for successfully establishing a lightpath.

The dissemination of information is a very important issue in link-state algorithms [10], [12]. Let us consider that the ants are implemented as IPv4 datagrams (20-byte header) with 20 bytes reserved for the ant’s own header (type, source and destination address, wavelength inclusive list, etc) [6], and that each traversed node contributes to 4 bytes to the ant’s memory. In Fig. 10, the average overhead that is generated by the artificial ants in each control channel is depicted for both networks, where the bars indicate the standard deviation. Under the launching rate used throughout the simulations (100 ants/s), we have very little impact over the control channels, since we have less than ten kilobits per second of bandwidth occupied, on average, by the datagrams carrying the ants.

The necessary bandwidth necessary to transport the ants scales linearly with the launching rate and the size of the network is almost not relevant to the overhead in each control channel. Moreover, since the ants are launched at regular intervals, they do not exhibit the unwanted burstiness nature of link state updates.

Finally, we assessed the restorability performance of the proposed ACO algorithm with crankback routing extensions when compared with a traditional fixed-alternate re-routing scheme. All graphs depicts the standard error of the mean, considering four different runs for every possible single failure.

In Fig. 11, we can note that ACO-based algorithm outperforms the fixed-alternate approach most of the time for the single-node failure scenario and it is very competitive in the single-link failure scenario, when the NSFNet network is considered.

The same behavior can also be observed for the NTTNet network, as shown in Fig. 12.

VII. CONCLUSION

In this work, we proposed an ACO-based RWA algorithm with crankback re-routing extensions that is capable of both provisioning lightpaths under the normal operation of the network and to restore lightpaths affected by network failures.

Indeed, the proposed algorithm is capable of mimicking the self-adapting, self-organizing behavior of real ants by circumventing obstacles (blockages due to resource unavailability) or by founding new routes when old ones are not feasible anymore (lightpaths affected by failures).

We demonstrated that only with crankback re-routing extensions the ACO algorithm can outperform the classical topological algorithms in almost all cases during normal network operation and in many times during restoration of failed connections. The algorithm tends to be more efficient when there is a large number of different alternate paths in the network to be explored, like in NTTNet. It is important to note that many cases of lower performance during restoration may be linked to the fact that since the ACO associated with crankback has a lower level of blocking probability, there are also less free resources to accommodate the re-routing of the failed lightpaths.
The cost incurred by the crankback mechanism, in terms of latency due to a larger number of hops explored, although not much higher than the topological approach, is the trade-off to achieve lower levels of blocking probability.

The artificial ants have very little impact on the control channels, needing just a few kilobits per second of the their bandwidth in the proposed scenarios.

Finally, the proposed algorithm can efficiently automate the operation of optical networks, without any kind of centralized control or supervision, being seamlessly integrated into the GMPLS control plane.

REFERENCES


Authorized licensed use limited to: Yuan-Ze University. Downloaded on May 22,2010 at 04:49:21 UTC from IEEE Xplore. Restrictions apply.
PAVANI and WALDMAN: ROUTING AND WAVELENGTH ASSIGNMENT WITH CRANKBACK RE-ROUTING EXTENSIONS 541


Gustavo Sousa Pavan graduated from University of Campinas (UNICAMP) in 2001 with a degree in Computer Engineering. He received his M.Sc. degree and his Ph.D. degree in Electrical Engineering from UNICAMP, in 2003 and 2006, respectively. Currently, he is an assistant professor at Universidade Federal do ABC (UFABC), Brazil. He has interest on the following topics: routing algorithms for packet-switched and circuit-switched optical networks by means of genetic algorithm or ant-colony optimization (ACO), GMPLS control plane, and the optical network support for grid architectures.

Helio Waldman received a BSEE from Instituto Tecnológico de Aeronáutica (ITA) at São José dos Campos, Brazil, in 1966, and the M.S. and Ph.D. degrees from Stanford University in 1968 and 1972, respectively. In 1973 he joined the University of Campinas (UNICAMP), where he was Director of the School of Engineering at Campinas from 1982 to 1986 and Research Vice-President from 1986 to 1990. He is currently Undergraduate Vice-President of UFABC - Universidade Federal do ABC, a new Brazilian Federal University currently under construction in the State of São Paulo. Dr. Waldman is a Senior Member of IEEE and a Senior Member of SBET, where he served as President between 1988 and 1990. Dr. Waldman was active in the investigation of ionospheric physics using satellite radio emissions until 1973, when he engaged on a Brazilian research program on digital communications systems. Since the eighties, his research interests have focused on the fiber optic channel. He has authored three books (all in Portuguese): "Digital Signal Processing" (1987), "Optical Fibers: Technology and System Design" (1991), and "Telecommunications: Principles and Trends" (1997). He has published 26 papers in international journals, and 89 papers in proceedings of scientific meetings. He has supervised 32 Master’s Theses and twelve doctoral Theses. His current research interests are in the areas of Optical Networking and Broadband Communications. He is also interested in discussing the new communication technologies and their impact on labor, education and society.