Future Generation Computer Systems 25 (2009) 257–265

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Co-scheduling in Lambda Grid Systems by means of Ant Colony Optimization☆

Gustavo Sousa Pavani *, Helio Waldman

Federal University of ABC (UFABC), Rua Catequese, 242. Santo André–SP, CEP: 09090-400, Brazil

ARTICLE INFO

Article history:
Received 22 October 2007
Received in revised form
29 August 2008
Accepted 1 September 2008
Available online 18 September 2008

Keywords:
Publish-and-subscribe
Ant Colony Optimization
Grid resource management
Lambda Grid
GMPLS control plane

ABSTRACT

The use of a dynamic reconfigurable optical network is an important requirement for the new advanced resource-intensive, highly distributed Grid applications that begin to emerge on the e-Science field. In this paper, we propose an ACO-based algorithm that is capable of the co-scheduling of both computational and optical network resources. We assess the performance of the proposed algorithm by comparison with traditional publish-and-subscribe grid systems that make use of topological routing of the lightpaths. The proposed algorithm can be used as a viable alternative for grid scheduling in Lambda Grids.

1. Introduction

In recent years, the Grid computing community has largely benefited from the deployment of dedicated optical networks [1–4] that provide the advanced transport infrastructure for new emerging Grid applications.

Indeed, with the advent of the Wavelength Division Multiplexing (WDM) technology, a pair of fibers can carry hundreds of Gibabits of data per second for long distances. This fact linked to advances in the reconfigurability of the optical networks led to the possibility that the Grid users or applications can set up and tear down lightpaths on demand for transporting data to resources located elsewhere in the Grid. These systems are commonly referred to as Lambda Grids [2,5].

The problem arises when the network resources have to be managed by the Grid scheduler, since traditionally the network was considered a static resource and only the computing and storage resources were taken into consideration for scheduling purposes.

A Grid scheduler (or broker) [6] is the entity responsible for assigning tasks or jobs to resources on multiple administrative domains. Scheduling in Grids, as almost any scheduling problem, is known to be NP-hard.

In this work, we propose a Grid scheduling system based on Ant Colony Optimization (ACO) [7] that is capable of the co-scheduling of both computational and optical network resources. We present the comparison of traditional publish-and-subscribe grid systems with an extension of the ACO-based system proposed on [8], and provide important benchmark comparisons.

Since we need to dynamically provision lightpaths, a Routing and Wavelength Assignment (RWA) [9] algorithm has to work closely with the Grid scheduler. The use of a Generalized Multi-Protocol Label Switching (GMPLS) control plane, combined with the RWA algorithm, provides a standardized way to manage the optical network.

Ant-based RWA algorithms have been shown to outperform good conventional algorithms in dynamic, Telecom-oriented networks where requests are made for lightpaths connecting specific source-destination pairs of nodes [10]. In Grid environments, where requests are made for connectivity to a resource to be discovered in the network, ant-based algorithms seem to be even more appropriate, as they naturally combine the solutions to the discovery and routing problems: when the resource is discovered, the pheromone levels produced by the search activity are already an appropriate metric for good routing.

The remaining of the paper is organized as follows. In Section 2, we present related work to ACO, resource discovery and allocation, and managed Lambda Grids. The proposed algorithm and the Lambda Grid architecture used in this paper are described in Sections 3 and 4, respectively. In Section 5, we detail the simulation studies carried out to properly characterize the behavior of the scheduling algorithms. Results obtained are presented and discussed in Section 6. Finally, in Section 7, conclusions are drawn.

☆ This work is an extended version of: G.S. Pavani, H. Waldman, Grid resource management by means of ant colony optimization, in: Third International Conference on Broadband Communications, Network and Systems (BroadNets 2006), San Jose, CA, 2006.

* Corresponding author. Tel.: +55 19 3521 5099; fax: +55 19 3289 1395.
E-mail addresses: gustavo.pavani@ufabc.edu.br (G.S. Pavani), helio.waldman@ufabc.edu.br (H. Waldman).

0167-739X/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2008.09.003

2. Related work

To our knowledge, few works treat the optical network infrastructure as a schedulable resource just like it is currently done with computing and storage resources.

In [11], a general framework, which relies on a GMPLS control plane combined with a RWA algorithm, for providing the optical transport infrastructure to Grid applications is presented. However, the problem of co-scheduling optical network and Grid resources is not directly treated.

The same limitation can be seen in [4], which let for a future work the problem for developing an effective Grid scheduler for both computing and network resources.

On the other hand, in [12] the network resources are considered by the Grid scheduler for estimating the network bandwidth, but they are not dynamically provisioned on demand, i.e., those resources are not managed.

Similar proposal can be found in [13]. This paper accounts for the network bandwidth while allocating jobs in remote clusters. Again, there is no direct control of network resources, such as the provisioning of dedicated lightpaths.

Ant Colony Optimization has been used in conventional (i.e., processing and storage resources only) resource allocation in Grid environments. For instance, resource allocation in computational grids with no global control has been demonstrated in [14] by using ACO. However, only in [8] the optical network resource allocation was incorporated for the first time by the ACO-based Grid scheduler.

3. Ant Colony Optimization (ACO)

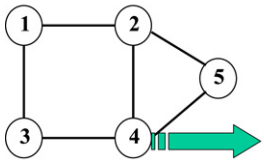
ACO is used to refer to the class of algorithms that are inspired in the process of foraging for food of natural ants for the optimization of hard-to-solve problems or problems that need distributed control.

ACO has an important feature for our work: it is explicitly modeled in terms of computational (generally mobile) agents, which is specially suited for routing in telecommunications networks [15]. (Computational) agents can be defined as computational entities that act 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 [16,17].

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. Indeed, 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 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.

The AntNet [18] framework is an ACO-based algorithm used for routing of packets in telecommunication networks, on which this work is based.

Indeed, 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 heuristics 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.



Neighbor nodes

Destination nodes	3	2	5
1	0.1	0.5	0.4
2	0.2	0.3	0.5
3	0.9	0.1	0.0
5	0.5	0.25	0.25

Fig. 1. Example of pheromone routing table of node 4.

At each intermediate node i, the following data structures have to be maintained in AntNet, which are slightly adapted to our algorithm in order to use the number of hops in lieu of the delay:

(i) Pheromone-routing table $\mathcal{T}^i$: 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., $\sum_{n\in\mathcal{N}_i} \tau^i_{dn} = 1$, where $\mathcal{N}_i$ represents the set of neighbors of node i. τ^i_{dn} defines the value of pheromone in the link $l_{i\to n}$ for the specified destination d and estimates the probability of reaching d given that n is selected as the next hop. In this way, we have only one type of pheromone, but an ant that travels to a specific destination only senses/modifies the pheromone levels associated with its destination (row) in the routing table. Fig. 1 depicts an example of pheromone routing table of an example network.

(ii) Statistical parametric model $\mathcal{M}^i$: It is a matrix containing the triplet $\langle \mu_d, \sigma_d, E_d\rangle$ for each destination d of the network, where μ_d represents the average length of the paths followed by the ant from the current node to destination d, σ_d the standard deviation for these path lengths and E_d the best value of path length found for this destination within the non-sliding window of w observations [18]. Non-sliding means that when the number of observations reaches $w+1$, all the accumulated values are reset, i.e., $E_d = \mu_d =$ (path length of the $(w+1)$-st observation), $\sigma_d = 0$ and the observation counter receives 1.

The values of μ^i_d and σ^i_d are updated using the following exponential model [19]:

$$\mu^i_d \leftarrow \mu^i_d + \eta(o_{i\to d} - \mu^i_d) \quad (1)$$

$$\sigma^i_d \leftarrow \sigma^i_d + \eta(|o_{i\to d} - \mu^i_d| - \sigma^i_d)\,, \quad (2)$$

where $o_{i\to d}$ is the new observation of the distance between nodes i and d, and η 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 $5/\eta$ [18].

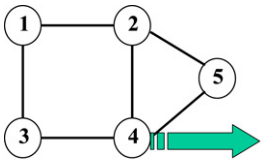
The number of values in the window is calculated as $w = 5(c/\eta)$, where $c \in (0, 1]$ is a reduction factor. Therefore, the window is updated in a smaller interval than the one used for the mean and standard deviation estimates, in such a way that the value of E^i_d and these estimates refer to the same set of observations [18].

Fig. 2 depicts an example of the statistical parametric model of an example network.

(iii) Availability vector $\mathcal{A}^i$: It is a vector containing an availability metric for each destination d of the network.

In this work, for sake of simplicity, we have considered that all nodes have the same number of processors P_{proc}, therefore the expression for the availability metric can be stated as $\mathcal{A}_d = P^{idle}_d$, where P^{idle}_d is the number of idle processors at destination d.

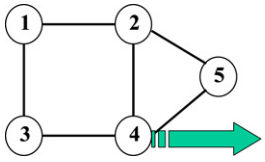
Fig. 3 depicts an example of availability vector of an example network.



Destination nodes

	μ_d	σ_d	E_d
1	2.2	0.5	2.0
2	1.3	0.3	1.0
3	1.2	0.4	1.0
5	1.4	0.5	1.0

Fig. 2. Example of statistical parametric model of node 4.



Destination nodes

1	**2**	**3**	**5**
33	5	8	1

Fig. 3. Example of availability vector of node 4.

$\mathcal{M}$ and $\mathcal{T}$ can be seen as local memories for the nodes, capturing different aspects of the network dynamics. The $\mathcal{M}$ model maintains distance estimates to all nodes, while the pheromone routing table gives the relative goodness of the next hop to reach a given destination.

The availability vector allows for the load-balancing of the grid workload by maintaining a memory for resource selection. It is exclusive of our AntNet adaptation.

Indeed, from the RWA problem perspective, the grid environment differs from WRON mainly because requests do not aim at a connection between a source and a destination node, but between a user and a resource that is located in some node to be discovered. Therefore, the requested resource must be discovered and connected, i.e. routed. In ACO, discovery and routing are naturally entangled, since the route is already determined (by pheromone maximization) upon the discovery of a resource.

Thus, a grid is formed by user and resource nodes, where a node can be both a user and resource node. The algorithm will try to establish a lightpath to connect a user node looking for a resource node somewhere in the network, while reducing the overall lightpath blocking probability. Therefore, the user nodes will be the source nodes and the resource nodes the destination nodes of our algorithm. For the ACO algorithm, it also means that only user nodes will be generating ants.

The AntNet algorithm is composed by two phases [7]: solution construction and updating of the data structures, which are detailed in the next sub-sections.

3.1. 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 [20].

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.

At regular intervals $(1/R_{ants})$, a forward ant $F_{s\to d}$ is launched from a random source node s to another random destination node d. On its trip from s to d, the forward ant selects the next hop $(i+1)$ using a random scheme that accounts for the path selection probabilities, given by the pheromone levels τ_{dn} in each neighbor link, and for a heuristics value h_n, calculated from the availability of the neighbor links.

The availability of each neighbor link is calculated as follows:

$$h_n = \frac{l_n^a}{W}, \tag{3}$$

where l_n^a is the number of available wavelengths on neighbor n and W is the total number of wavelengths deployed on the link.

During its trip, the forward ant gathers the label of each node where it passes by, putting it in its memory $V_{s\to i}$, which also serves as a tabu list [21].

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 probabilities for each candidate node n to be the next hop $(i+1)$ are given by the following expression [18]:

$$p_n^d = \begin{cases} \left(\frac{1}{1+\alpha}\right)\frac{\tau_{dn}}{\sum_{k\in T}\tau_{dk}} + \left(\frac{\alpha}{1+\alpha}\right)\frac{h_n}{\sum_{k\in T}h_k}, & \forall k \in T \\ 0, \quad \text{otherwise}, \end{cases} \tag{4}$$

where α gives the emphasis between pheromone level (long-term memory) and instantaneous availability state (short-term memory)), and $T = \mathcal{N}_i \setminus V_{s\to i}$.

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:

$$p_n^d = \begin{cases} \frac{\tau_{dn}}{\sum_{k\in T'}\tau_{dk}}, & \forall k \in T', \text{ if } |\mathcal{N}_i| > 1 \\ 1, & \text{if } \mathcal{N}_i = \{v_{i-1}\} \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where v_{i-1} represents the last visited node and $T' = \mathcal{N}_i - \{v_{i-1}\}$.

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.

3.2. Updating of data structures

When the forward ant arrives at d, it becomes a backward ant $B_{d\to s}$ 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 $\mathcal{M}^i$ and, after it, the pheromone routing table, for all entries relative to d.

Moreover, this update is also made for all nodes $d' \in V_{i\to d}$, $d' \neq d$ in the sub-paths $(i \to d')$ traversed by the forward ant $F_{s\to d}$ after visiting i. If this sub-path is statistically good, then the entries of $\mathcal{M}^i$ and $\mathcal{T}^i$ 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 $\text{dist}(V_{i\to d'}) < I_{sup}^{d'}$ [18], where dist() is a function that gives the distance, in terms of number of hops, of the path followed by the ant and I_{sup} 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:

$$I_{sup}^{d'} = \mu_{d'} + \frac{1}{\sqrt{(1-\gamma)}} \frac{\sigma_{d'}}{\sqrt{w}}, \tag{6}$$

where γ is the confidence level coefficient.

Thus, the local parametric model is updated using Eqs. (1) and (2), where $o_{i\to d} = \text{dist}(V_{i\to d})$. If $o_{i\to d} < E_d^i$, then $E_d^i \leftarrow o_{i\to d}$. 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 [18]:

$$r_d = c_1 \left(\frac{E_d}{\text{dist}(V_{i\to d})} \right) + c_2 \left(\frac{I_{sup}^d - E_d}{(I_{sup}^d - E_d) + (\text{dist}(V_{i\to d}) - E_d)} \right). \tag{7}$$

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 $\text{dist}(V_{i\to d}) = I_{sup}^d = E_d$. 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:

$$r_d = \frac{s(r)}{s(1)}, \quad \text{where } s(x) = \left(1 + \exp\left(\frac{a}{x|\mathcal{N}_i|}\right)\right)^{-1}, \tag{8}$$

where a is an amplifier coefficient.

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

$$\tau_{dm} \leftarrow \tau_{dm} + r_d(1 - \tau_{dm}). \tag{9}$$

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

$$\tau_{dn} \leftarrow \tau_{dn} - r_d\tau_{dn}, \quad \forall n \in \mathcal{N}_i,\ n \neq m. \tag{10}$$

As already done for the parametric model, the updating process is also repeated for all d' 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 lesser blocking probability.

Finally, when the backward ant returns to source node s, the position d of the availability vector $\mathcal{A}^s$ is updated with the value of availability value gathered by the ant at destination d.

4. Lambda grid architecture

The lambda grid architecture of this paper is built around an integrated GMPLS-controlled WDM optical network.

We consider two different grid scheduling systems: one based on ants for routing the lightpaths and collecting resource availability, and a distributed publish-and-subscribe with topological routing of the lightpath.

In the first system, both the management of Grid resources and lightpaths are naturally combined in the Grid scheduling, since the the ants act on behalf of the user to make resource discovery and allocation, and to route lightpaths. On the other hand, in the second system, the Grid resources and the optical network are separately managed, resulting in an overlay approach [22].

The signaling part relies on the RSVP-TE protocol [23], as described in [10]. In addition, an extra error code is reported by the RSVP-TE protocol when the *Path* message arrives at the resource node and there is no available processor, i.e., the lightpath request is blocked due to the lack of processing resources. This is necessary for the case when a resource node, which had available processors, is now busy and the information about the node's actual status has not reached the user nodes yet.

After a resource node is selected to handle a job, the process of establishing a lightpath between the user node and the resource node is triggered.

Once the lightpath is established, the data related to the job is transferred to the resource node to be processed. After the transfer of the job data, the lightpath is torn down and the job is executed.

However, blocking of a job request can occur due to two main reasons [8]: the first one is caused by lack of a node to process the job, i.e., the allocated resource node or all resource nodes are busy. The second type of blocking happens when there are insufficient network resources to establish a lightpath.

4.1. Node selection policy

In [8], only one node selection algorithm was proposed: the selection of the least-loaded node to execute the job. Although this approach seems interesting for balancing the grid workload, it may waste important network resources, increasing the total blocking.

For this reason, we propose two alternative algorithms in addition to the Least-Loaded (LL) one:

- The Closest Least-Loaded (CLL): This approach chooses the least-loaded node among the closest ones in terms of number of hops. The rationale behind this algorithm is to avoid the sending of jobs to nodes too far away.
- The Best Availability-Distance Ratio (BADR): This strategy selects the node whose ratio between number of processors available and distance in terms of number of hops is the maximum one. Indeed, this policy represents a trade-off between the LL and CLL approaches.

4.2. Routing and wavelength assignment in the grid

In order to assess the performance of the ant routing, we also took into consideration two topological-driven approaches for the routing subproblem: the shortest-path and fixed-alternate algorithms [9].

In the fixed-alternate routing, each node in the network maintains a routing table containing an ordered list of a number of fixed routes from each source node to each destination node. For instance, these routes may include the shortest-path route, the second-shortest-path and so on. In fact, the shortest-path routing is a special case of fixed-alternate routing [9].

For calculating the k-shortest paths in this work, we used Yen's algorithm [24], restraining its number to 3, i.e., we have a shortest path and two other alternative paths.

For the ant-based routing algorithm, the route is calculated hop-by-hop as the Path [23] message travels towards the destination [10], where the next hop with the highest level of pheromone is selected. If the *Path* message enters in a cycle, then a *PathErr* is generated and thus the job request is blocked.

For the wavelength assignment subproblem, we used a first-fit approach for all routing algorithms. In this scheme, all wavelengths are numbered. When searching for an available wavelength, a lower-numbered is considered before a higher-numbered wavelength. The first available wavelength is then selected. This approach is used due to its simplicity and low computation cost. Also, this scheme performs well in terms of blocking probability and fairness [9].

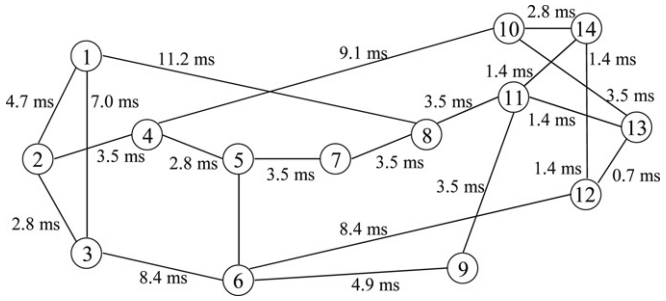


Fig. 4. NSFNet backbone network.

5. Simulation

For testing the scheduling algorithms, we used the NSFNet backbone network, shown in Fig. 4. It is a 14-node network with 21 bidirectional links and it is well-balanced, with an average shortest-path length between all pairs of nodes equal to 2.2. The latencies between neighbor nodes are also depicted in Fig. 4.

We have considered a homogeneous, Poissonian job traffic with uniform spatial profile. We consider a highly asymmetric scenario, with nodes 1 and 2 competing for resources in the grid, which are all located on the other nodes. The execution time of each job follows a negative exponential service time with an average value of 1500 s.

The duration of the lightpaths has a negative exponentially distributed profile with an average value equal to 10 s. Note that this a critical parameter of the simulation, since it directly impacts on the blocking due to lack of optical network resources.

We assume that no additional time is needed after the input and executable files are downloaded to the resource node to start the job execution.

Moreover, we are considering that there is enough storage space on the resources for handling any number of processing requests. Since we are only interested in the steady-state evaluation of the system, we also assume that the system does not allow advance reservation of resources. This consideration is also linked to the fact that the GMPLS standard does not support advance reservation and the definition of a Grid GMPLS standard, which would allow advance reservation, is on its infancy. In addition, adding reservation increases the wait time of queued applications in almost all cases [25], although it may reduce blocking probability.

In order to illustrate the capabilities of the algorithms, two different scenarios were taken into consideration:

- **Scenario 1:** There are 4 wavelengths available on each network link. This scenario is limited by network resources (lambdas) on a 100% grid workload with P_{proc} processors per node.
- **Scenario 2:** There are 8 wavelengths available on each network link. In this case, there is enough network resources for handling all job requests on a 100% grid workload with P_{proc} processors per node.

In both scenarios, without loss of generality, we assume that there is no lightpath blocking when the results of the job execution are sent back to the original user node.

The parameters used in the simulations are depicted in Table 1:

6. Numerical results

First of all, we evaluated the total (resource + lightpath) and resource-only blocking probability for different node selection policies and RWA algorithms considering the Scenario 1. The results are depicted in Figs. 5–8.

Table 1
Parameters used in the simulations

Parameter	Symbol	Value
Number of requests for each run	P_{run}	10^6
Total number of processors at each node	P_{proc}	100
Global rate for launching forward ants	R_{ants}	48 ants/s
Interval to start the lightpath requests	$I_{requests}$	1000 s
Correction for routing of forward ants	α	0.6
Weight of the window's exponential model	η	0.005
Reduction for parametric model's window	c	0.3
First weight of the adaptive reinforcement	c_1	0.6
Second weight of the adaptive reinforcement	c_2	0.4
Confidence level for reinforcement	γ	0.65
Amplifier of the squash function	a	5

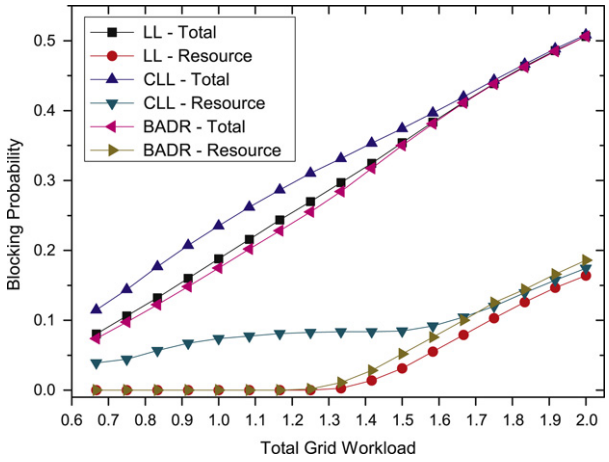


Fig. 5. Blocking probability under different grid workloads for the proposed ant-based algorithm (W = 4).

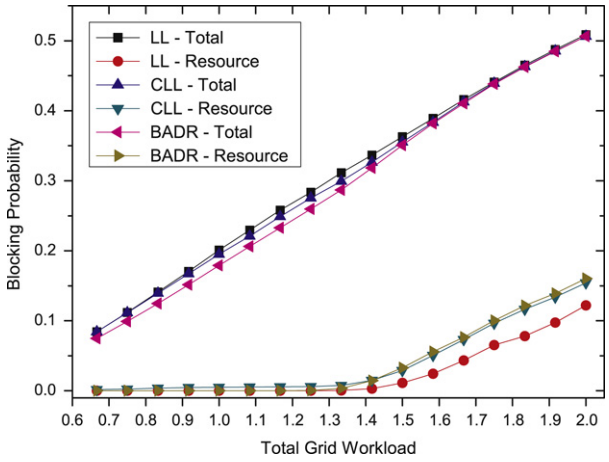


Fig. 6. Blocking probability under different grid workloads for the shortest-path algorithm (W = 4).

The CLL policy has the worst performance in terms of blocking than the other allocation policies for the ant routing. Indeed, the CLL policy is too greedy for using with ant-based routing, because it limits the scope of the node selection procedure while the ants seek for the load-balancing of the network. In other words, the ant routing and the CLL policy have conflicting aims, worsening the overall blocking probability.

On the other hand, for shortest-path and fixed-alternate (with one extra path) routing, the worst performance is credited to the LL policy. Since those algorithms follow a topological approach, it conflicts with the selection policy that does not care about distance, which is exactly the LL policy.

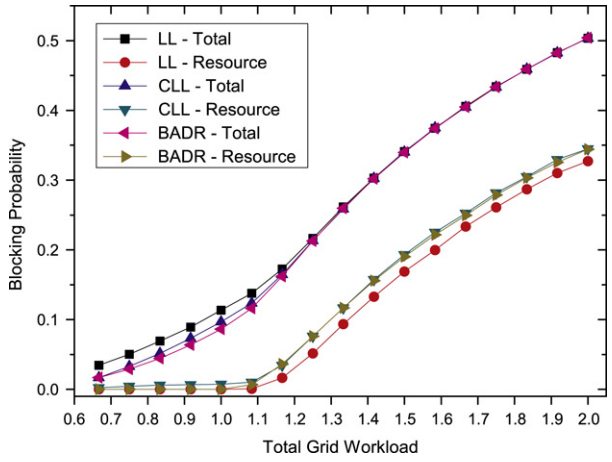


Fig. 7. Blocking probability under different grid workloads for the fixed-alternate (one alternative path) algorithm (W = 4).

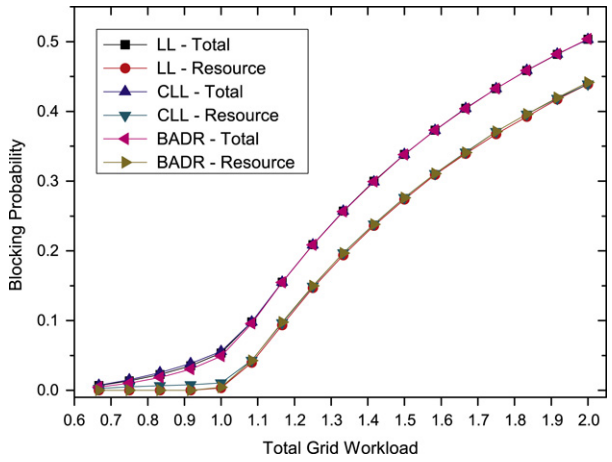


Fig. 8. Blocking probability under different grid workloads for the fixed-alternate (two alternatives paths) algorithm (W = 4).

Although the LL policy achieved the lowest level of resource-only blocking, the total blocking is increased by waste of resources at optical network level.

For the fixed-alternate (with two extra paths) routing, almost no difference can be seen from the different selection policies.

Briefly, after the inspection of all those Figures, the BADR policy has obtained the best performance in terms of blocking probability and, for this reason, the rest of simulations of this work are done using the BADR policy.

In Fig. 9, we compare all RWA algorithms still considering Scenario 1 by showing their total blocking probability under varying grid workload.

The ant-based routing and the shortest-path routing have similar performance in terms of blocking probability. The best one is the fixed-alternate routing with two alternative paths closely followed by the fixed-alternate routing with one alternative path. This behavior was already observed in [10], since the ACO algorithm is more efficient when the network has a large set of alternative paths, which is not the case when we used the NSFNet network.

Also, this is not a totally fair comparison since the ACO algorithm evaluates only one route while the fixed-alternate routing approach evaluates more than one route for each connection request. Indeed, this is an open research topic for AntNet variants of the ACO algorithm, while the use of alternative paths in a different ACO technique has been demonstrated in [26].

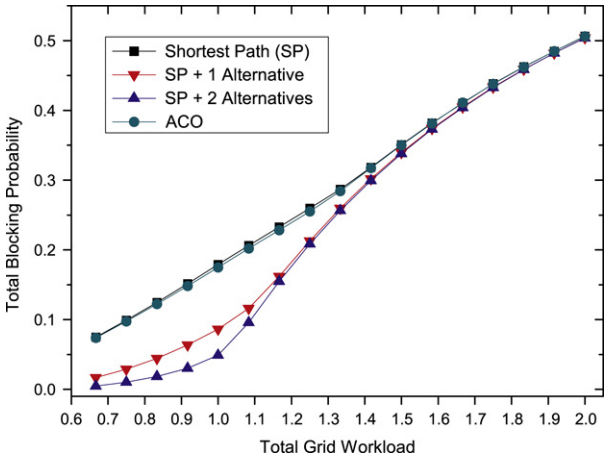


Fig. 9. Blocking probability under different grid workloads (W = 4).

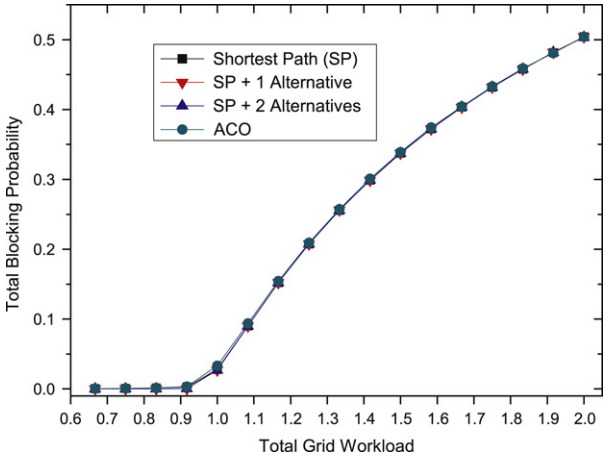


Fig. 10. Blocking probability under different grid workloads (W = 8).

Regarding the fact that this simulation does not account for advance reservation, note that if we have insufficient network resources, the nodes will starve due to the lack of jobs to be processed, which in turn results in the resource nodes operating below their capacity. This is exactly what happens when advance reservation is allowed, with total blocking probability being directly translated as idle capacity of the resource nodes when we consider a 100% grid workload (Fig. 10).

Afterward, the same comparison is repeated for Scenario 2.

As already expected, when the resources of the optical network are not a limiting factor, all RWA algorithms achieve the same level of performance.

In addition, we verified the influence of the number of processors in the total blocking probability. The results are shown in Figs. 11 and 12 for Scenarios 1 and 2, respectively.

As already noted before, the ant-based and shortest-path routing have similar performance in terms of blocking probability. They are surpassed by the fixed-alternate routing, with two alternative paths getting better better results than just one.

Since the fixed-alternate routing makes a better use of the network resources, it can handle a range from 50 to 100 more processors per node with the same blocking of the other routing algorithms.

Another important parameter that should be taken into consideration is the average lightpath duration, which is related to the data rate of the lightpath and the size of the job data. Fig. 13 depicts the impact of different values of this parameter under 100% grid workload for Scenario 1.

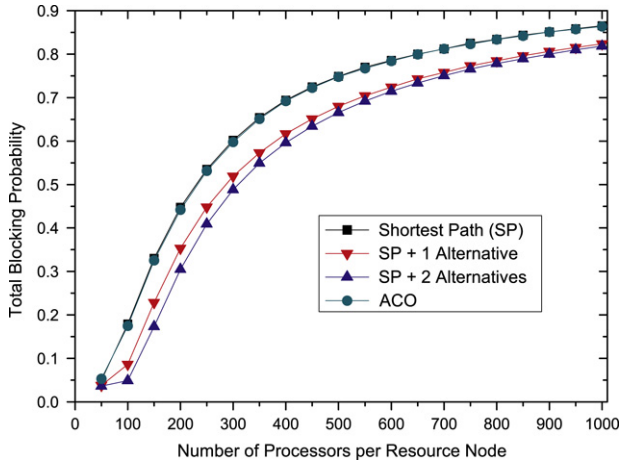


Fig. 11. Blocking probability for different number of processors in the nodes under 100% grid workload (W = 4).

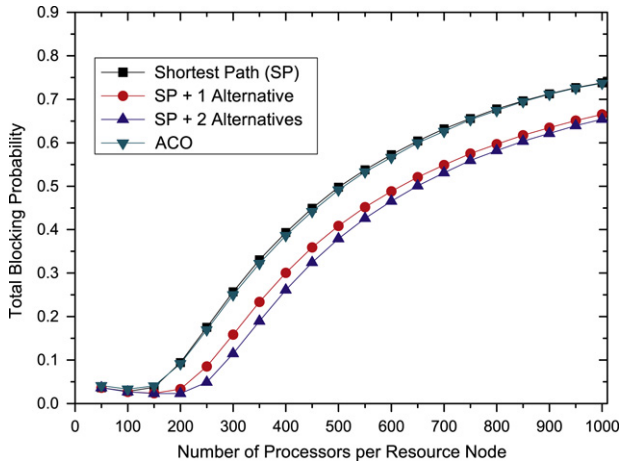


Fig. 12. Blocking probability for different number of processors in the nodes under 100% grid workload (W = 8).

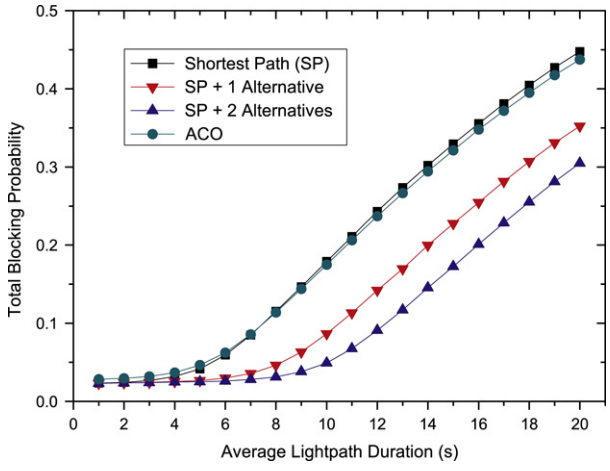


Fig. 13. Total blocking probability for different values of average lightpath duration under 100% grid workload (W = 4).

As we can observe in the Fig. 13, for a value of average lightpath duration above 4 s the blocking due to insufficient network resources becomes important. The performance, in terms of blocking probability, is equivalent to the one observed for the case of variable number of processor as explained before.

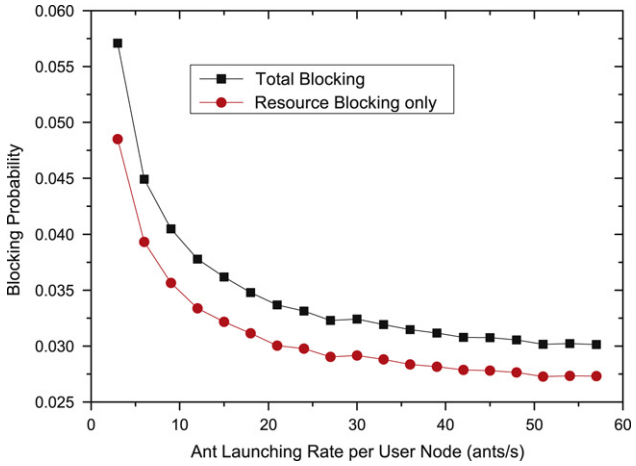


Fig. 14. Blocking probability for different ant launching rates under 100% grid workload (W = 8).

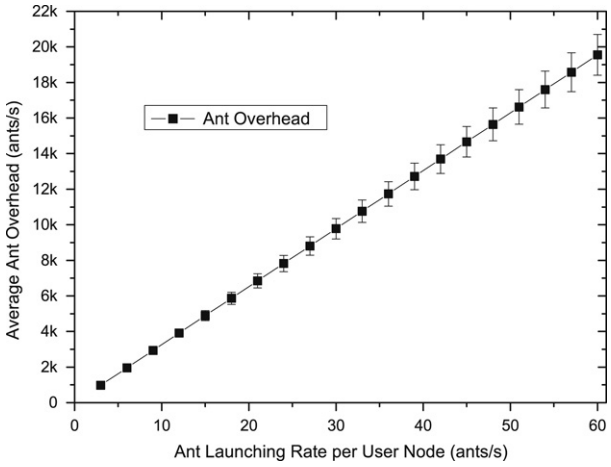


Fig. 15. Overhead of the ant packets on the control channels (W = 8).

Then, we evaluated the influence of the global ant launching rate over the blocking probability for a grid workload equal to 100%, considering Scenario 2. The results are shown in Fig. 14.

Increasing the ant rate improves the performance of the system, but this improvement tends to level off after a certain value. This parameter is very important to obtain a good performance of the ACO algorithm. However, the value of this parameter is intrinsically dependent of the system and it is usually tuned up using a trial and error approach.

The extra overhead due to the ants on the control channels are very dependent on the implementation of the ant packets. Let's suppose that the ant is implemented like a raw packet in a IPv6-capable [27] network. In the simplest case, we have a 40-byte header and each hop contributes a 16-byte address to the ant's payload. Also, the backward ant has a 16-byte field for storing the availability information.

Fig. 15 depicts the overhead caused by the ant packets averaged over all network links, where the bars indicate the standard error of the mean.

For the global rate (R_{ants}) of 48 ants/s used throughout this work, which is equivalent to a rate of 24 ants/s per user node, we have an average 8 kbps of overhead on the control channels. The introduced overhead is much smaller than actual capacity of control channels, which are generally capable of carrying tens of Mbps of data.

Finally, Fig. 16 shows the overhead caused by the publish messages when a shortest-path algorithm is used by the publish-and-subscribe system, which is averaged over all network links. The bars indicate the standard error of the mean.

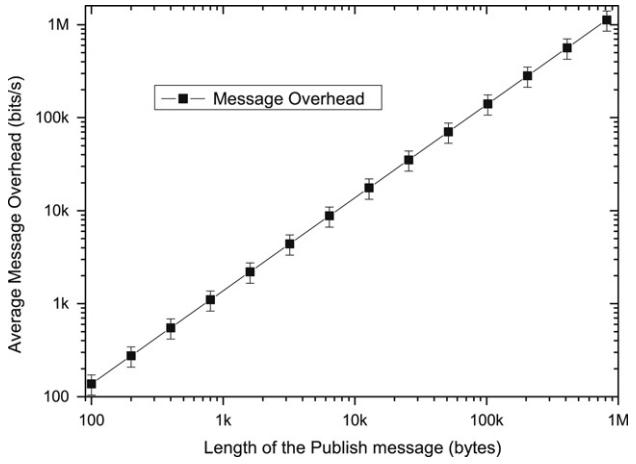


Fig. 16. Overhead of the publish messages on the control channels.

Depending on the the length of publish message, a publish-and-subscribe lambda grid system can demand more bandwidth than the ant approach. In the simulations of the publish-and-subscribe system, we are neglecting the overhead introduced by the routing, supposing that the routing is accomplished by the OSPF-TE protocol [28,29]. Thus, we are underestimating the impact of the overhead introduced on the control channels by those systems, since the ants are both responsible for routing, resource discovery and allocation.

7. Conclusions

In this work, we proposed the use of an ACO-based algorithm as a viable alternative for scheduling in Lambda Grid systems.

Indeed, the proposed algorithm is able to manage the optical network as any other resource, such as processing power or storage space. Moreover, a joint optimization of lightpath routing and resource discovery and allocation is possible without introducing important modifications on the GMPLS control plane.

The ant system presented in this work plays the role of the Grid User Network Interface (G-UNI) without creating a complicated protocol for supporting an integrated management of lightpaths and Grid resources.

In addition, we presented some simulations for assessing the performance of the algorithm when compared with traditional publish-and-subscribe systems.

Further study is needed to enhance the routing capability of the ACO algorithm in order to be competitive with traditional routing strategies.

Acknowledgments

The authors wish to thank Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazil for supporting this research.

References

[1] H.B. Newman, M.H. Ellisman, J.A. Orcutt, Data-intensive e-science frontier, research, Communications of the ACM 46 (11) (2003) 68–77.
[2] O. Yu, A. Li, Y. Cao, L. Yin, M. Liao, H. Xu, Multi-domain lambda grid data portal for collaborative grid applications, Future Generation Computer Systems 22 (8) (2006) 993–1003.
[3] J. Sobieski, T. Lehman, B. Jabbari, C. Ruszczyk, R. Summerhill, A. Whitney, Dynamic provisioning of lightpath services for radio astronomy applications, Future Generation Computer Systems 22 (8) (2006) 984–992.
[4] A. Takefusa, M. Hayashi, N. Nagatsu, H. Nakada, T. Kudoh, T. Miyamoto, T. Otani, H. Tanaka, M. Suzuki, Y. Sameshima, W. Imajuku, M. Jinno, Y. Takigawa, S. Okamoto, Y. Tanaka, S. Sekiguchi, G-lambda: Coordination of a grid scheduler and lambda path service over GMPLS, Future Generation Computer Systems 22 (8) (2006) 868–875.
[5] P. Thysebaert, M.D. Leenheer, B. Volckaert, F.D. Turck, B. Dhoedt, P. Demeester, Scalable dimensioning of resilient lambda grids, Future Generation Computer Systems 24 (6) (2008) 549–560.
[6] J.M. Schopf, Ten actions when grid scheduling: The user as a grid scheduler, in: Grid Resource Management: State of the Art and Future Trends, Kluwer Academic Publishers, 2004, pp. 15–23.
[7] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, 2004.
[8] G.S. Pavani, H. Waldman, Grid resource management by means of ant colony optimization, in: Third International Conference on Broadband Communications, Network and Systems, BroadNets 2006, San Jose, CA, 2006.
[9] H. Zang, J. Jue, B. Mukherjee, A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks, Optical Networks Magazine 1 (1) (2000) 47–60.
[10] G.S. Pavani, H. Waldman, Evaluation of an ant-based architecture for all-optical networks, in: 10th Conference on Optical Network Design and Modelling, ONDM'06, Copenhagen, Denmark, 2006.
[11] F. Palmieri, GMPLS-based service differentiation for scalable QoS support in all-optical Grid applications, Future Generation Computer Systems 22 (6) (2006) 688–698.
[12] E. Elmroth, J. Tordsson, Grid resource brokering algorithms enabling advance reservations and resource selection based on performance predictions, Future Generation Computer Systems 24 (6) (2008) 585–593.
[13] I.D. Scherson, D. Valencia, E. Cauich, J. Duselis, R. Wang, Federated grid clusters using service address routed optical networks, Future Generation Computer Systems 23 (8) (2007) 957–967.
[14] L. Liu, Y. Yang, L. Li, W. Shi, Using ant colony optimization for superscheduling in computational grid, in: IEEE Asia–Pacific Conference on Services Computing, APSCC'06, Los Alamitos, CA, USA, 2006, pp. 539–545.
[15] E. Bonabeau, M. Dorigo, G. Theraulaz, Inspiration for optimization from social insect behaviour, Nature 406 (2000) 39–42.
[16] V.A. Pham, A. Karmouch, Mobile software agents: An overview, IEEE Communications Magazine 36 (7) (1998) 26–37.
[17] D.B. Lange, M. Oshima, Seven good reasons for mobile agents, Communications of the ACM 42 (3) (1999) 88–89.
[18] G. Di Caro, M. Dorigo, AntNet: Distributed stigmergetic control for communications networks, Journal of Artificial Intelligence Research 9 (1998) 317–365.
[19] V. Jacobson, M. Karels, Congestion avoidance and control, ACM Computer Communication Review 18 (4) (1990) 314–329.
[20] B. Barán, R. Sosa, AntNet – Routing algorithm for data networks based on mobile agents, Revista Iberoamericana de Inteligencia Artificial 12 (2001) 75–84.
[21] F. Glover, M. Laguna, Tabu Search, Kluwer Academic Publishers, 1997.
[22] D. Simeonidou, R. Nejabati, G. Zervas, D. Klonidis, A. Tzanakaki, M.J. O'Mahony, Dynamic optical-network architectures and technologies for existing and emerging grid services, Journal of Lightwave Technology 23 (10) (2005) 3347–3357.
[23] L. Berger, Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions, RFC 3473 (Proposed Standard), updated by RFC 4003 (Jan. 2003). http://www.ietf.org/rfc/rfc3473.txt.
[24] J. Yen, Finding the k shortest loopless paths in a network, Management Science 17 (11) (1971) 712–716.
[25] W. Smith, I. Foster, V. Taylor, Scheduling with advanced reservations, in: Proceedings of the 14th International Symposium on Parallel and Distributed Processing, IPDPS'00, Cancun, Mexico, 2000, pp. 127–132.
[26] S.-H. Ngo, X. Jiang, S. Horiguchi, Ant-based alternate routing in all-optical WDM networks, IEICE Transactions on Communications E89-B (3) (2006) 748–755.
[27] S. Deering, R. Hinden, Internet protocol, Version 6 (IPv6) Specification, RFC 2460 (Draft Standard) (Dec. 1998). http://www.ietf.org/rfc/rfc2460.txt.
[28] K. Kompella, and Y. Rekhter (Eds.), Routing extensions in support of Generalized Multi-Protocol Label Switching (GMPLS), RFC 4202 (Proposed Standard) (Oct. 2005). http://www.ietf.org/rfc/rfc4202.txt.
[29] K. Kompella, Y. Rekhter, OSPF extensions in support of Generalized Multi-Protocol Label Switching (GMPLS), RFC 4203 (Proposed Standard) (Oct. 2005). http://www.ietf.org/rfc/rfc4203.txt.

Gustavo Sousa Pavani graduated from State 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. 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 State 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 Research 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 SBrT, 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 22 papers in international journals, and 85 papers in proceedings of scientific meetings. He has supervised 29 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.