Ant Colony Optimisation for Virtual-Wavelength-Path Routing and Wavelength Allocation

Griselda Navarro Varela
Dept. of Electronic Systems Engineering,
University of Essex,
Wivenhoe Park, Colchester, Essex CO4 3SQ, UK

Mark C. Sinclair
Dept. of Electronic Systems Engineering,
University of Essex,
Wivenhoe Park, Colchester, Essex CO4 3SQ, UK
`mcs@essex.ac.uk`

Abstract- Ant Colony Optimisation (ACO) is applied to the problem of routing and wavelength-allocation in a multi-wavelength all-optical virtual-wavelength-path routed transport network. Three variants of our ACO algorithm are proposed: local update (LU), global update/ distance (GU/D) and global update/occupancy (GU/O). All three extend the usual practice that ants are attracted by the pheromone trail of ants from their own colony: in our work, the artificial ants are also repelled by the pheromone of other colonies. Overall, the best ACO variant, GU/O, provides results that approach those of an earlier problem-specific heuristic on small- and medium-sized networks.

1 Introduction

Multi-wavelength all-optical transport networks have attracted considerable interest in recent years, because of their potential, by using multiple wavelengths in both optical transmission and optical switching, to provide the huge bandwidths necessary if broadband services are to be widely adopted [1]. In addition, incorporating wavelength conversion in the nodes (optical cross connects) allows the individual static traffic requirements to be routed through a succession of wavelengths as they traverse the network. Thus considerable flexibility is provided for both restoration in the event of failures, as well as in responding to the increasingly dynamic traffic requirements.

A variety of solution approaches have been adopted for optical-path-layer design (*i.e.* routing, fibre choice and wavelength allocation), including heuristics, evolutionary algorithms and mathematical programming. For example, Sinclair [2, 3] has recently applied a genetic-algorithm/heuristic hybrid approach to routing, fibre and wavelength allocation in transport networks without wavelength conversion. Here, we consider wavelength conversion, and adopt Ant Colony Optimisation (ACO) as the solution approach. This method draws its inspiration from the cooperative problem-solving behaviour of natural ant colonies [4, 5, 6]. However, our ACO algorithms extend the usual practice that ants are *attracted* by the pheromone trail of ants from their own colony: in our work, the artificial ants are also *repelled* by the pheromone of other colonies. (Throughout this paper, ACO is used to refer to the entire class of ant-inspired algorithms which have individually been termed 'ant system', 'ant colony system', *etc.*)

2 Ant Colony Optimisation

2.1 Ants in nature

Individual ants are relatively simple insects that have a very limited amount of memory, are almost blind and their individual behaviour is apparently random [4, 7]. However, they act together as a colony to perform complex tasks such as: regulating nest temperature, forming bridges, searching areas for food, building and protecting their nest, finding the shortest routes to food, and exploiting the richest available food source [7]. To achieve these tasks, they use cooperative behaviour, but without the execution of a joint plan. Ants communicate indirectly through environmental stimuli; this form of communication is termed *stigmergy* [7]. The stimuli are based on two kinds of changes in the environment. With *sematectonic stigmergy*, the stimuli are task-related, actions such as digging a hole or building a ball of mud. These actions change the environment, and other ants react by performing the same or related actions: they remove more material from the hole, or add more mud to the ball. This cooperation with the task is not a consequence of intelligence, but simply a response to stimuli [7]. In *sign-based stigmergy*, the ants deposit a volatile hormone (*pheromone*) to act as a stimulus to other ants. The pheromone thus serves as a signalling system, acting as a means of indirect communication, and leading to cooperative behaviours such as trail following [7]. It is this latter aspect of ant behaviour which is of most interest to us here.

As has been noted, real ants are capable of finding shortest or near-shortest paths between a food source and their colony (nest) [4, 5, 6, 7]. For example, in Fig. 1, two ants at a food source are faced with the choice of two alternative routes back to the nest, one longer than the other (although this is unknown to the ants). Faced with such a decision, and in the absence of other stimuli, the ants appear to take a route at random [7]. We will assume that one takes the upper longer route, the other the lower shorter one. In the same way, two ants at the nest are faced with a similar decision in trying to reach the food source. Again, we will assume one takes the longer, and the other the shorter, route. As the ants walk, they deposit pheromone on the ground, leaving a trail. This is illustrated in Fig. 2, which shows the moment when the two ants on the shorter paths reach their respective destinations; the other two ants are still *en route*. If, at this point in time, an ant was to start out from the nest heading for the food source,

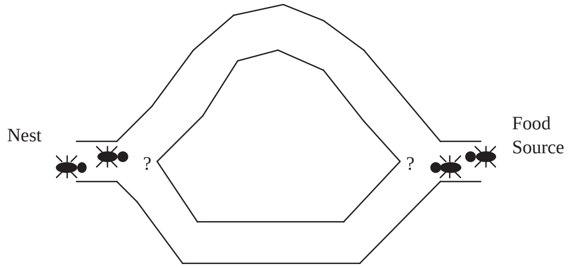


Figure 1: Ant route choice between longer (upper) and shorter (lower) paths (after [7])

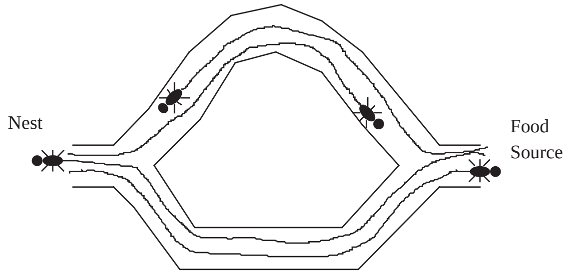


Figure 2: Initial ant pheromone trails (after [7])

it would detect twice as much pheromone on the lower route compared to the upper. In the presence of pheromone, ants appear to select routes with probability proportional to the strength of the pheromone [4]. Consequently, the ant would be twice as likely to take the lower shorter route. It too would lay pheromone along the shorter path, thus further reinforcing its attractiveness to other ants. Although pheromone gradually evaporates, a strong pheromone trail would still be laid fairly rapidly on the lower shorter route, as more and more ants added to it. However, the trail on the longer upper route would soon disappear completely. The ants would then simply follow the shorter path until the food source is exhausted.

Ants, while trail following in this way, can have two problems [7]: the blocking problem and the shortcut problem. The former occurs when a route that was found by the ants is no longer available: it takes time for them to find a short alternative route. The latter corresponds to a new shorter route becoming available: such a route will not easily be found by the ants, because their well-established current route will have a high level of pheromone, and the probability of ants leaving the trail to explore, and thus finding the new route, is vanishingly small (while the food source remains).

2.2 Artificial ants

The foraging behaviour of real ants has been used by Dorigo *et al.* [4, 8, 5] to define several variants of their general-purpose heuristic algorithm, ACO. According to its developers, ACO is versatile, in the sense that it can be easily used on similar versions of the same problem; is robust, because it can be applied with minimal changes to different combinatorial optimisation problems; and is a population-based approach, which allows positive feedback to be used as the primary search mechanism [5].

The ACO approach uses a set of agents which work as artificial ants in a cooperative way to build a solution to a problem by exchanging information via pheromone deposited on the edges of a graph. This graph represents the environment of the artificial ants. While moving, they build solutions and modify the problem representation by adding collected information to the graph [6].

Artificial ants are a particularly simple form of agent, in that while they cannot carry out intelligent actions, they can choose the path they follow. However, this action is not a completely autonomous choice, because the ants choose paths at random influenced by pheromone concentrations. Thus, they are only making a stimulus response to their environment; the higher the concentration of pheromone, the higher the probability that a particular path will be chosen; they are reactive. Nevertheless, the changes in the environment are produced by the interaction between ants; they are collaborative. However, while artificial ants have memory, they cannot learn, as although path reinforcement may seem like learning, it is not a direct characteristic of the individual ants.

Now, while ACO is inspired by real ants, there are important differences, as it is intended as an optimisation tool, not a faithful simulation of ants in nature [5]. Artificial ants have memory for storing actions they have performed or to record places they have been; they are not completely ‘blind’, but have ‘sight’ based on distance, pheromone levels, traffic flow, congestion, *etc.*; and their environment operates on discrete time.

For ACO to be successfully applied to a problem [8, 5], it must be possible to represent it as a search on a graph by simple agents; a positive feedback mechanism must be identified, equivalent to pheromone; a greedy heuristic must be incorporated to allow constructive definition of the solutions; and appropriate constraints provided through the memories of the individual ants.

2.3 ACO applications

Since its first development, ACO has been applied to a variety of problem areas, only a few of which are mentioned here. These include the travelling salesman problem (TSP) [4, 8, 5, 6]; static routing (load balancing) in circuit-switched telecommunications networks [7]; and dynamic routing in packet-switched networks [9, 10].

Two of the earliest published accounts of ACO are by Colorni *et al.* [4, 8]. Their first paper [4] both introduces the ACO approach, and describes three ACO algorithms for the TSP (ant-density, ant-quantity and ant-cycle). The most successful of these (ant-cycle) found a new optimal tour length for a 30-city problem, as well as good tours for larger problems (50 and 75 cities). Their second paper [8], investigates appropriate parameter settings, empirical computational complexity, and result quality against TSP-specific heuristics, for their earlier ant-cycle algorithm (see also §2.4 below).

A more comprehensive treatment of ACO for the TSP is given by the same authors in [5] which, as well as restating their earlier work [4, 8], compares ant-cycle against both tabu search and simulated annealing. In addition, the application of ACO to the asymmetric TSP, the quadratic assignment problem and job-shop scheduling is also described.

The ant-cycle algorithm is extended by Dorigo & Gambardella in [6] to create an ACO algorithm, termed ant colony system, for both TSP and asymmetric TSP. The ant colony system incorporates explicit exploration into the route-choice mechanism of the ants; focuses their search far more onto the best tour found so far; and modifies pheromone evaporation to include aspects of Q-learning. By further modifying their algorithm to include problem-specific heuristics, some of the best ever results are reported for large asymmetric TSP instances.

Schoonderwoerd *et al.* [7] have applied ACO to the static routing (load balancing) problem in circuit-switched telecommunications networks. Their ACO algorithm out-performs both an earlier multi-agent approach [11], as well as an improved variant of the multi-agent system of their own devising.

Finally, in [9, 10], Di Caro & Dorigo describe the application of ACO to dynamic routing in packet-switched networks. Experimental results demonstrate that their AntNet algorithm provides best performance in average delay terms compared to a suite of both Internet standard and state-of-the-art routing algorithms, particularly under difficult traffic conditions. In addition, it demonstrates robust behaviour, always rapidly reaching a good stable level of performance.

2.4 ACO for the TSP

As a further illustration of ACO, we provide a brief informal description of Colorni *et al.*'s ant-cycle algorithm [4] for the TSP.

In each algorithm cycle, one ant starts from every node (city). Then, for each algorithm (time) step, ants choose probabilistically which node to move to next, according to both the pheromone and length of the links they are considering crossing: they prefer more pheromone and shorter links. Each ant is constrained to visit only those nodes it has not visited before (by a tabu list), until it has followed a complete tour. After all the ants have completed their tours, pheromone is deposited, for each ant, on each link of the ant's completed tour. The pheromone deposited is greater on shorter tours (*cf.* real ants); pheromone is then evaporated (a little) on all links. The cycle is then repeated, with all ants starting from their original nodes, for a given maximum number of cycles. The best tour ever found is kept as the final result.

3 Multi-wavelength networks

Multi-wavelength all-optical transport networks are proposed to form the highest layer of national and international networks, providing optically-transparent wavelength channels, each capable of carrying 10 Gbit/s or more. Their optical-path layer is formed by statically routing individual traffic requirements (say as many 10 Gbit/s channels as are necessary between each source-destination pair) using both multi-wavelength links and optical cross connects. Each link potentially consists of many fibres, and each fibre is capable of carrying a certain number of wavelengths. No electronic conversion is used in the optical-path layer. Instead, the individual channels are carried using either wavelength-path (WP) routing, where a single wavelength is used end-to-end, or virtual-wavelength-path (VWP) routing, which employs as many different wavelengths as necessary along each path by using optical wavelength conversion at the nodes. However, there are serious technological limitations and cost implications in carrying more than a few wavelengths over national or even international distances. Consequently, it is important to route requirements through the network in a way that minimises the *network wavelength requirement* (NWR). That is, the total number of different wavelengths that are used in the network, which is in turn equal to the maximum number needed on any one link, or any one of the fibres comprising that link. (For simplicity, we will only consider single-fibre links in the remainder of the paper.)

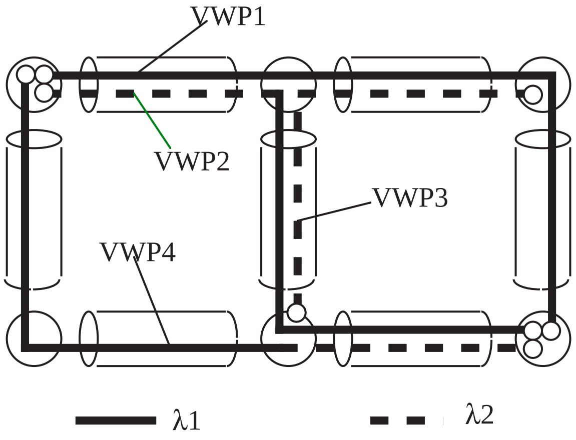


Figure 3: Virtual-wavelength-path routing in a small example network (after [12])

3.1 Virtual-wavelength-path routing

In multi-wavelength networks, every requirement (channel) using a link needs a distinct wavelength. However, in VWP networks it is possible to change the wavelength used by a requirement, through wavelength conversion, at any (and every) node along the path. The lowest-numbered wavelength that is free can thus be taken on each of the links used. This is illustrated in Fig. 3, which shows a small 6-node, 7-link, single-fibre VWP network. Consider laying the four VWP paths shown across the network in numerical order. The first, VWP1, can use wavelength λ_1 end-to-end. However, VWP2 has to employ λ_2 on one link to avoid clashing with VWP1. In the same way, VWP3 has to be on λ_2 throughout, and VWP4 also needs to use λ_2 on one of its links.

Obtaining a low NWR thus amounts to distributing the requirements evenly across the links, as the link with the greatest number of wavelengths determines the NWR, while at the same time keeping the path lengths short (in terms of both distance and number of hops, *i.e.* number of links traversed). This is because, on every link used by a path, a wavelength is required, so paths that use more links than necessary will use up extra wavelengths (potentially increasing NWR). However, following the shortest path will sometimes not be the best choice, as a slightly longer path may be able to use a lower wavelength number on some links (potentially decreasing NWR).

3.2 Nagatsu's VWP heuristic

An example of a conventional heuristic for routing and wavelength allocation in VWP networks is that described by Nagatsu *et al.* [12]. This is a two-stage algorithm, which first establishes reasonable initial routes, and then improves these in a second iterative re-routing stage. Wavelength assignment is simply achieved by a greedy method, which assigns the lowest available wavelength to each channel on a link.

The initial routing is achieved by first determining both the minimum number of hops, and the number of channels required, between each node pair in each direction. Individual channels are then routed in priority order, with the highest priority given to the node pair with the greatest product of minimum hop count and channels remaining to be routed. The routing itself is achieved by maintaining a weight associated with each network link; this is simply the number of channels using it. A channel is actually routed by assigning it to the path with the lowest sum of link weights; any affected weights are then updated.

Once all the channels have been given an initial route, rerouting can begin. For each iteration, two different attempts are made to improve the NWR. Both involve identifying those links which are carrying the highest number of channels, and so have the maximum link wavelength requirement (LWR) which is, of course, equal to the NWR. First, those paths which use the largest number of maximum-LWR links are identified as candidates, and an attempt is made to re-route them using a minimum-link-weight path. If an improvement in NWR is achieved, the new route is adopted, otherwise the candidate reverts to its original route. If, after trying all candidates paths, no improvement is achieved, the second attempt is made. This time, any path which uses a maximum-LWR link is a candidate, and they are all re-routed onto new paths that contain the minimum number of maximum-LWR links (excluding, in each case, those that were used by the original paths).

Throughout, Nagatsu *et al.* impose an upper limit on path length: no path is ever adopted which is more than, say, 2 hops longer than the minimum-hop-count path between that node pair. In addition, they impose a limit, say 50, on the number of iterations of the second stage.

4 Applying ACO

The applicability of ACO for VWP routing and wavelength allocation is obvious. The problem is inherently graph-based, and the routing of wavelength channels clearly requires path-following behaviour. However, there are still several important factors in the algorithm design to consider.

For the TSP, only a single minimum-length tour is required, and a single ant is launched from each node in each cycle [4]. Here, however, there are $n(n-1)/2$ node pairs in an n-node network, with several channels needing to be routed between them in each direction. If the number of channels required *from* node i, *to* node j is c_{ij}, the total number of channels is:

$$C = n(n-1)\overline{c} = \sum_{\forall i,j} c_{ij} \quad i \neq j \tag{1}$$

where $\overline{c}$ is the average number of channels required per node pair in each direction. Even if all the channels between a node pair in one direction are grouped together and thus follow the same route, an ant is required in every cycle for each such group *i.e.* a total of $n(n-1)$ ants, with each ant using its own (distinct) type of pheromone. As an alternative, allowing the possibility that each wavelength channel may be independently routed, an ant and pheromone type is needed for each of the $n(n-1)\overline{c}$ individual channels.

Some previous authors have explicitly introduced noise into link choice to promote exploration [6, 7]. Thus, at a certain probability level, an ant chooses a random link, rather than being guided by pheromone levels or other considerations. Clearly, this is also a possibility for our algorithms.

In following a tour in the TSP, ants employ a tabu list to avoid returning to nodes: they must visit each node (city) only once [4]. However, here, ants are following a path from a particular source node to a particular destination. The paths must not contain loops, and so each ant records the route it has traversed. This route also serves as a tabu list of nodes the ant must not revisit. However, as the ants are restricted to using the actual network topology, rather than being able to move freely between nodes as in the TSP, it is possible for an ant to reach a 'dead end'. For example, in Fig. 4 an ant is attempting to move from source node 4 to destination node 0. So far it has followed the indicated path, 4-1-3-2, as a result of stochastic choices at each node along the way. However, now the ant is unable to move, as it is not allowed to revisit nodes 1 or 3. This problem is overcome in our algorithms by allowing an ant to backtrack: it returns along its route until an allowed choice presents itself. While backtracking, the ant removes nodes from its memory of the route, as well as restoring any modified pheromone on the links traversed. In addition, to enable backtracking to be accomplished correctly, ants not only maintain a record of their route, but also remember the previous node visited. For example, the ant in Fig. 4 could backtrack to node 3, modifying its stored route to 4-1-3. At

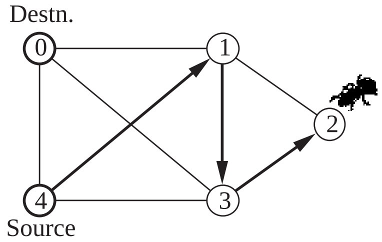


Figure 4: Ant at a dead end (node 2) *en route* from node 4 (source) to node 0 (destination)

that point, both node 0 and node 2 might appear valid choices (neither is on the route), but node 2 is the previous node; hence the only valid choice is node 0.

Now, it still remains to be determined which routes are actually used by the wavelength channels. At the end of each algorithm cycle, each ant will have followed a particular route. However, the choices that resulted in this route were made stochastically: the ant may not necessarily have followed the maximum-probability path through the network. Consequently, two distinct sets of routes are maintained by our algorithm: a *probabilistic* set, which is simply the routes taken by the ants in the last cycle, and a *deterministic* set, which is the routes that would have resulted if each ant had followed the maximum-probability path from its source to destination. This is the path that would have resulted if at each node along its route the ant had deterministically selected the link currently assigned the highest probability (*cf.* [7]).

While the ants directly determine the routes used, the wavelength allocation itself is carried out using a simple greedy method (*cf.* [12]). For each route, in node-pair order, the lowest available wavelength is used on each link traversed. Consequently, the ants must select routes that will promote low NWR. As was explained in §3.1, this requires that the selected routes follow short paths, but at the same time, lead to low and even loading on the network links. To achieve this, the usual application of pheromone as an attractor has had to be extended: the ants will generate shortest paths if they are *attracted* by their own pheromone; but even loading if they are *repelled* by other ants' pheromone.

4.1 Basic algorithm

We have developed three major variants on our ACO algorithm: local update (LU), global update/distance (GU/D), and global update/occupancy (GU/O). However, they all follow the same basic structure, as follows.

During each algorithm cycle, ants move from every source to every destination, one link per algorithm (time) step. The ants make their link choices stochastically, according to the algorithm variant. Ants are destroyed when they reach their destination although, as path lengths differ, this does not happen for all the ants in the same step. When all the ants have died, a new cycle starts. Ants update their pheromone either each step (local) or each cycle (global). In addition, all the different pheromone types are evaporated (a little) each cycle. Also, at the end of each cycle, both the probabilistic and the deterministic routes are used as the basis for a wavelength allocation. The best ever NWR found is stored, and forms the final result of the algorithm.

4.2 Local Update

In order to complete our description of the LU variant of our ACO algorithm, two further elements are required: the mechanism by which an ant chooses which link to take at each step, and the pheromone-updating rule.

To promote shortest-path routes, an ant is attracted to a link, of those attached to its current node and over which it is allowed to move, according to the amount of its own pheromone on the link. (Recall that an ant cannot move over a link to a node which is either the node it has just come from, or is already on its route.) Thus α_{kj}, the *weight of attraction* of link k for ant j, is obtained by normalising the amount of the ant's own pheromone type on the link, p_{kj}, by the sum of its pheromone over all the allowed links (from which it must choose), *i.e.*:

$$\alpha_{kj} = \frac{p_{kj}}{\sum_{i \in A_j} (p_{ij})} \tag{2}$$

where A_j is the set of all links currently allowed to ant j.

In opposition to the attraction by an ant's own pheromone, ant j is also repelled by all the other pheromone types on link k, given by:

$$P_{kj} = \sum_{h \neq j} p_{kh} \tag{3}$$

This is transformed into the *weight of repulsion*, β_{kj}, of link k for ant j by normalising over all the allowed links, *i.e.*:

$$\beta_{kj} = \frac{P_{kj}}{\sum_{i \in A_j} (P_{ij})} \tag{4}$$

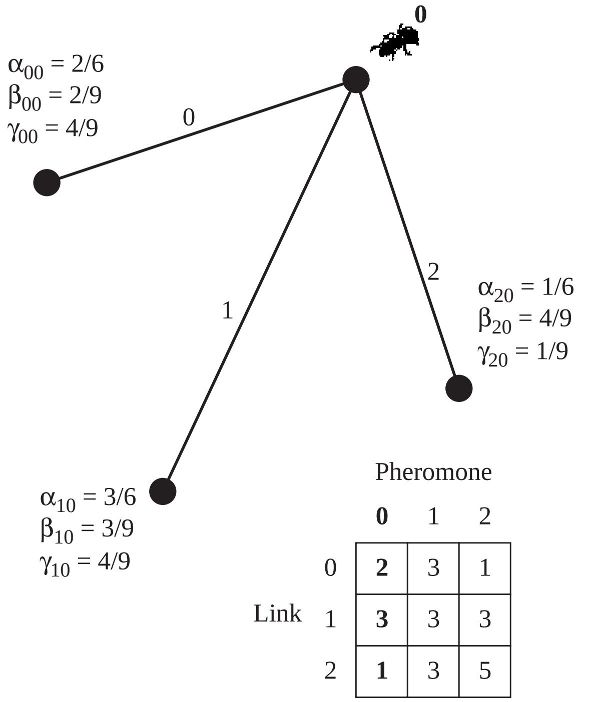
Finally, we combine the two weights into the probability, γ_{kj}, that ant j will take link k:

$$\gamma_{kj} = \frac{\alpha_{kj}/\beta_{kj}^{\varepsilon}}{\sum_{i \in A_j} (\alpha_{ij}/\beta_{ij}^{\varepsilon})} \tag{5}$$

where ε is a constant. By dividing α_{kj} by β_{kj}, we ensure that the probability γ_{kj} increases with the weight of attraction and decreases with the weight of repulsion. The use of ε enables us to vary the relative dependence of probability on the two weights. Further, normalising over all the allowed links makes γ_{kj} a valid probability measure, as:

$$\sum_{i \in A_j} \gamma_{ij} = 1 \tag{6}$$

As an example of the operation of ant choice in local update, consider the ant, type 0, in Fig. 5. It is faced with the choice of three links, labelled 0, 1 and 2, all of which we assume are allowable choices. The table in the lower right of the



Link	**0**	1	2
0	**2**	3	1
1	**3**	3	3
2	**1**	3	5

Figure 5: Example of ant link choice for the LU variant

figure records current pheromone levels, not only for ant 0, but also for two other ants, 1 and 2, on all three links. The sum of pheromone type 0 on all three links (the first column of the table), is 6. Consequently, applying Eq. 2, gives the weights of attraction for the three links as $\alpha_{00} = 2/6$, $\alpha_{10} = 3/6$ and $\alpha_{20} = 1/6$ for the three links 0, 1 and 2, respectively. In the same way, with the sum of other ants' pheromone 18, the weights of repulsion from Eq. 4 are 2/9, 3/9 and 4/9, respectively. Finally, applying Eq. 5 with $\varepsilon = 1$, we arrive at probabilities for the three links of 4/9, 4/9 and 1/9, respectively. Thus, although the greatest concentration of ant 0's own pheromone is on link 1, this is counterbalanced by a concentration of other ants' pheromone, leading to a tie between links 0 and 1. Link 2, with both a low level of pheromone type 0, and high concentrations of other types, is reduced to a poor third.

For the pheromone-updating rule, we adopt that used by Colorni *et al.* [4, 8, 5] for their ant-density algorithm. Thus on each algorithm step, every ant deposits a fixed quantity of pheromone Q of its own type on the link it used (local update). Hence:

$$p_{kj}^{t+1} = p_{kj}^{t} + Q \qquad (7)$$

where ant j used link k in time step t. Pheromone evaporation simply reduces the level of pheromone on all links by a factor ρ, termed the evaporation coefficient, at the end of the last step of each cycle T [4, 8, 5], *i.e.*:

$$p_{kj}^{t+1} = \rho \cdot p_{kj}^{t} \quad \forall j, \forall k, \quad t = \sum_{i=1}^{T} s_i \qquad (8)$$

where s_i is the number of steps in cycle i.

4.3 Global Update/Distance

The second major variant of our ACO algorithm, GU/D, uses the same link-choice mechanism as local update. However, the pheromone-updating rule (Eq. 7) is changed from local updating by each ant on every step, to global update at the end of each cycle. Our rule is similar to that employed by Colorni *et al.* [4, 8, 5] for their ant-cycle algorithm, although here, rather than using TSP tour-length to bias the pheromone quantities added, we use the length of the route followed by the ant. Thus:

$$p_{kj}^{t+1} = p_{kj}^{t} + Q/L_j^T \quad \forall j, \forall k \in R_j^T, \quad t = \sum_{i=1}^{T} s_i \qquad (9)$$

where L_j^T is the length of route R_j^T followed by ant j in cycle T. It should be noted that the path lengths, L_j^T, were assessed here in terms of distance, rather than hop count.

4.4 Global Update/Occupancy

The third and final major variant of our ACO algorithm, while still using global update (Eq. 9), modified the link-choice mechanism by introducing a different assessment for weight of repulsion. Instead of other ants' pheromone, this is based on link utilisation, u_k, which is assessed by counting the number of ants of any type crossing link k. However, u_k is also evaporated, just like pheromone, at the end of each cycle. Thus:

$$u_k^{t+1} = u_k^{t} + \sum_j x_{kj}^{t} \qquad (10)$$

where $x_{kj}^{t} = 1$ if ant j used link k in step t and 0 otherwise. In addition:

$$u_k^{t+1} = \rho \cdot u_k^{t} \quad \forall k, \quad t = \sum_{i=1}^{T} s_i \qquad (11)$$

Furthermore, Eq. 4 for the weight of repulsion of link k for ant j is replaced by:

$$\beta_{kj} = \frac{u_k}{\sum_{i \in A_j} (u_i)} \qquad (12)$$

It should be noted that, unlike in Eq. 4, the weight of repulsion for ant j now includes an indirect contribution from the activities of ant j itself, as well as that of the other ants.

5 Experimental results

To assess the different variants of our ACO approach to routing and wavelength allocation for VWP transport networks, we adopted three test networks. The first of these has 4 nodes spread over a 100 km × 100 km area, an applied traffic of 150 Gbit/s, and requires a total of 20 wavelength channels

(10 Gbit/s each). The topology was hand-crafted, and consists of just 5 links. The second network consists of the 9 central nodes of the European Optical Network, with a topology obtained by Sinclair using a bit-string genetic algorithm [13]. The traffic requirement is 515.4 Gbit/s, equivalent to 98 channels, and it has 17 links. The node locations and traffic demand of the final network were generated using the approach described by Griffith *et al.* [14], although further modified to ensure reasonable node separations. It has 15 nodes, covers a 1,000 km × 1,000 km area and carries an overall traffic of 1,500 Gbit/s, requiring 268 channels. The network topology, of 36 links, was obtained using the genetic-algorithm/ heuristic hybrid described by Sinclair [15].

As a benchmark for the ACO algorithms, we adopted the well-regarded VWP routing and wavelength allocation heuristic described by Nagatsu *et al.* [12] and summarised in §3.2 above.

From initial trial runs, we found that using only one ant for each node pair, in each direction, produced consistently better results than requiring one for each individual wavelength channel. Furthermore, while we used the recommendations of earlier authors [5] and trial runs to establish suitable parameter settings for our algorithm variants, we have yet to carry out an exhaustive investigation. In addition, while we explored introducing noise into our algorithms using a few trial runs, this did not appear to be beneficial in final result quality.

For LU, we adopted $\rho = 0.5$, $Q = 1$, $\varepsilon = 2$ and a maximum of 30 cycles; although this latter figure seems low, stagnation always appeared before this point, with all ants following the same routes as during the previous cycle. The GU/D parameters used were $\rho = 0.5$, $Q = 5$, $\varepsilon = 5$, and again a maximum of 30 cycles due to stagnation. Finally, for GU/O, we took $\rho = 0.9$, $Q = 0.5$, $\varepsilon = 5$, and were able to use 250 cycles without route stagnation. For all three variants, we initialised both the pheromone, for all links and all ants, and the link utilisations (for GU/O), to 10.

We applied the three algorithm variants and Nagatsu *et al.*'s heuristic to all three networks, in each case for ACO using five runs with different pseudo-random seeds. The best of five runs, or the single run for Nagatsu, are recorded in each case in Table 1.

There is a clear progression in solution quality as we move from LU to GU/D to GU/O, with the latter algorithm able to approach the solution quality of Nagatsu *et al.*'s well-regarded and highly problem-specific heuristic. Nevertheless, there is still some room for further improvement.

The progress of a typical ACO run is shown in Fig 6, for GU/O (with an ant for each individual channel) on Network 3, illustrating both the probabilistic and deterministic NWRs for each cycle. In cycle 0, the very high value for deterministic NWR (233) is an artifact of initialisation, as all links start with equal pheromone values, and so there are no maximum-probability routes to follow. After this point, the deterministic NWR is consistently better than the probabilistic, as a consequence of the ants laying down a good pattern of pheromone. Over the course of the run, both types of NWR gradually improve, with the probabilistic converging on the deterministic. This corresponds to the individual ants themselves more closely following the deterministic allocation routes, as their pheromone trails are progressively reinforced.

Net.	Nodes	Nagatsu	LU	GU/D	GU/O
1	4	5	7	5	5
2	9	11	27	17	12
3	15	13	50	32	15

Table 1: NWR results for Nagatsu and the three ACO variants

It should be noted that run times for our ACO approach were quite long. For example, on Network 3, a run of 250 cycles, with ants for individual channels, required about 2 hours 20 minutes on a Sun UltraSPARC 5 using Sun's JIT Java 1.1.3 compiler. However, our focus in developing the implementation was on correctness, rather than program speed, and it is anticipated that this figure could be significantly reduced. Execution of the Nagatsu heuristic takes about 20 seconds on the same platform.

6 Conclusions & further work

We have described the novel application of Ant Colony Optimisation (ACO) to routing and wavelength allocation in multi-wavelength all-optical transport networks employing wavelength converters. Three algorithm variants were presented, and these were compared experimentally against the well-regarded wavelength-allocation heuristic of Nagatsu *et al.* [12]. Our global update/occupancy (GU/O) algorithm approaches the solution quality of the Nagatsu heuristic, although at much higher computational cost.

In further work, it will be possible to carry out more detailed investigation of the appropriate parameter settings for our algorithms. Adopting a hop-count-based, rather than distance-based, global pheromone-updating rule may prove beneficial, as NWR is more sensitive to path hop count than path length. In addition, it is anticipated that incorporating problem-specific heuristic elements into our ACO approach, as has been done by Dorigo & Gambardella [6] for the TSP, would improve both algorithm efficiency and solution quality. For example, rather than simply using ant numbers, as in GU/O, the weight of repulsion could be derived directly from the link wavelength requirement in the previous cycle(s). Also, the second stage of Nagatsu *et al.*'s heuristic, say, could be used to improve the final route choice. Moreover, the more difficult area of wavelength-path routing (*i.e.* no wavelength conversion) remains to be tackled with ACO.

Acknowledgements

The research described in this paper was undertaken by the first author (supervised by the second) as part of an M.Sc. in Telecommunications and Information Systems at the Depart-

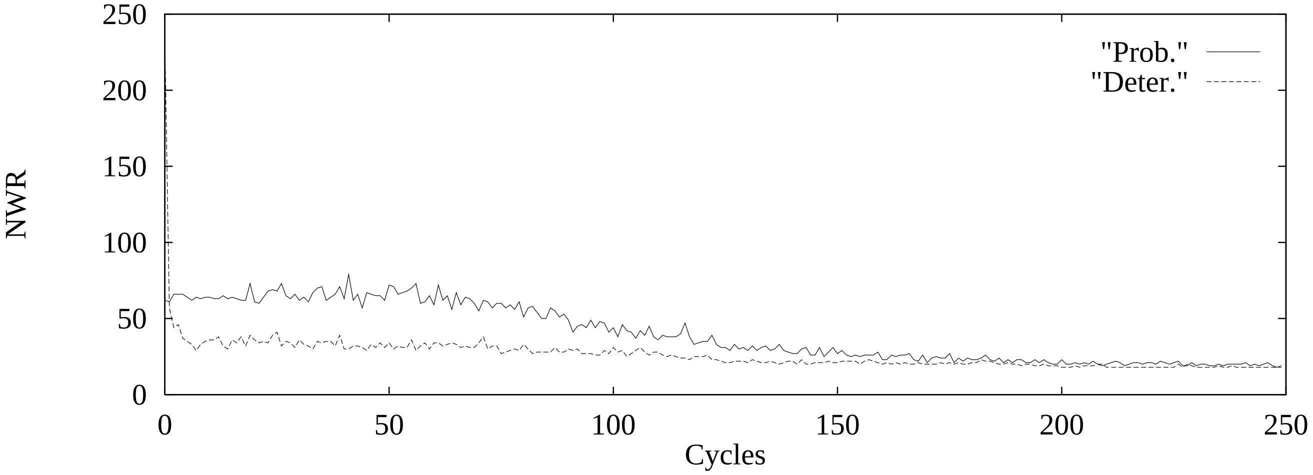


Figure 6: Probabilistic and deterministic NWR for an example run of GU/O on Network 3

ment of Electronics Systems Engineering, University of Essex, sponsored by CONACYT, an agency of the Government of Mexico. The authors are grateful to Noel Parnis (University of Essex) for his comments on an early draft of this paper.

Bibliography

[1] Mukherjee, B. *Optical Communication Networks*, McGraw-Hill, 1997.

[2] Sinclair, M.C. Minimum network wavelength requirement design using a genetic-algorithm/heuristic hybrid, *Electronics Letters*, 34(4):388–389, 1998.

[3] Sinclair, M.C. Minimum cost wavelength-path routing and wavelength allocation using a genetic-algorithm/ heuristic hybrid approach, *IEE Proc. Commun.*, 146, 1999.

[4] Colorni, A., Dorigo, M. & Maniezzo, V. Distributed optimization by ant colonies, *Proc. First European Conf. on Artificial Life (ECAL'91)*, Paris, France, 134–142, 1991

[5] Dorigo, M., Maniezzo, V. & Colorni, A. The ant system: optimization by a colony of cooperating agents, *IEEE Trans. Systems, Man and Cybernetics — Part B*, 26(1):1–13, 1996.

[6] Dorigo, M. & Gambardella, L.M. Ant colony system: a cooperative learning approach to the travelling salesman problem, *IEEE Trans. Evolutionary Computation*, 1(1):53–66, April 1997.

[7] Schoonderwoerd, R., Holland, O.E., Brutten, J.L. & Rothkrantz, L.J.M. Ant-based load balancing in telecommunication networks, *Adaptive Behaviour*, 5(2):169–207, 1997.

[8] Colorni, A., Dorigo, M. & Maniezzo, V. An investigation of some properties of an ant algorithm, *Proc. Parallel Problem Solving from Nature Conference (PPSN'92)*, Brussels, Belgium, 509–520, 1992.

[9] Di Caro, G. & Dorigo, M. Mobile agents for adaptive routing, *Proc. 31st Hawaii Intl. Conf. Systems Sciences (HICSS-31)*, Kohala Coast, Hawaii, 74–83, Jan 1998.

[10] Di Caro, G. & Dorigo, M. AntNet: Distributed stigmergetic control for communications networks, *Journal of Artificial Intelligence Research*, 9:317–365, 1998.

[11] Appleby, S. & Steward, S. Mobile software agents for control in telecommunications networks, *BT Technol. J.*, 12(2):104–113, April 1994.

[12] Nagatsu, N., Hamazumi, Y. & Sato, K.I. Number of wavelengths required for constructing large-scale optical path networks, *Electronics and Communications in Japan, Part I — Communications*, 78(9):1–11, 1995.

[13] Sinclair, M.C. Minimum cost topology optimisation of the COST 239 European optical network, *Proc. 2nd Intl. Conf. on Artificial Neural Networks and Genetic Algorithms (ICANNGA'95)*, Alès, France, 26–29, April 1995.

[14] Griffith, P.S., Proestaki, A. & Sinclair, M.C. Heuristic topological design of low-cost optical telecommunication networks, *Proc. 12th UK Performance Engineering Workshop*, Edinburgh, UK, 129–140, September 1996.

[15] Sinclair, M.C. NOMaD: Applying a genetic-algorithm/ heuristic hybrid approach to optical network topology design, *Proc. 3rd Intl. Conf. on Artificial Neural Networks and Genetic Algorithms (ICANNGA'97)*, University of East Anglia, Norwich, UK, 299–303, April 1997.