Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions

Kwang Mong Sim and Weng Hong Sun, *Member, IEEE*

***Abstract*—Although an *ant* is a simple creature, collectively a colony of ants performs useful tasks such as finding the shortest path to a food source and sharing this information with other ants by depositing pheromone. In the field of *ant colony optimization* (*ACO*), models of *collective intelligence* of ants are transformed into useful optimization techniques that find applications in computer networking. In this survey, the problem-solving paradigm of ACO is explicated and compared to traditional routing algorithms along the issues of routing information, routing overhead and adaptivity. The contributions of this survey include 1) providing a comparison and critique of the state-of-the-art approaches for mitigating *stagnation* (a major problem in many ACO algorithms), 2) surveying and comparing three major research in applying ACO in routing and load-balancing, and 3) discussing new directions and identifying open problems. The approaches for mitigating stagnation discussed include: *evaporation*, *aging*, *pheromone smoothing* and *limiting*, *privileged pheromone laying* and *pheromone-heuristic control*. The survey on ACO in routing/load-balancing includes comparison and critique of *ant-based control* and its ramifications, *AntNet* and its extensions, as well as *ASGA* and *SynthECA*. Discussions on new directions include an ongoing work of the authors in applying *multiple ant colony optimization* in load-balancing.**

***Index Terms*—Ant colony optimization, collective intelligence, mobile agent, swarm intelligence.**

I. Introduction

WITH the growing importance of telecommunication and the Internet, more complex networked systems are being designed and developed. The challenges of dealing with the vast complexity of networking problems such as load balancing, routing and congestion control accentuate the need for more sophisticated (and perhaps more intelligent) techniques to solve these problems. Drawing upon some of the computing techniques inspired by social insects such as ants [1], several mobile agent-based paradigms were designed to solve control and routing problems in telecommunication and networking. Although by itself, an *ant* is a simple and unsophisticated creature, collectively a colony of ants can perform useful tasks such as building nests, and *foraging* (searching for food) [1], [2]. What is interesting is that ants are able to discover the shortest path to a food source and to share that information with another ants through *stigmergy* [2], [3]. Stigmergy is a form of indirect communication used by ants in nature to coordinate their problem-solving activities. Ants achieve stigmergic communication by laying a chemical substance called *pheromone* [3] that induces changes in the environment which can be sensed by other ants. In recent years, computer scientists were able to transform the models of *collective intelligence* of ants into useful optimization and control algorithms [1], [4], [5]. In the interesting and emerging field of *ant colony optimization* (ACO) [2], [4]–[7], a colony of (biological) ants is typically modeled as a society of mobile agents (or artificial ants). Although ACO has been applied in many combinatorial optimization problems such as the asymmetric traveling salesman problem [8], [9], graph coloring problem [10] and vehicle routing problem [2], this manuscript focuses on surveying ACO approaches in network routing and load-balancing. In applying ACO in network routing and load-balancing, an artificial ant is typically realized as a simple program consisting of simple procedures that simulate the laying and sensing of pheromone, and data structures that record trip times and the nodes that it passes. Migrating from node to node, an artificial ant emulates laying of pheromone by updating the corresponding entry in the routing (or pheromone) table in a node which records, for example, the number of ants that pass that node. While a more detailed exposition of the problem-solving paradigm of ACO is given using an example in Section II, the differences between ACO and traditional routing algorithms are discussed in Section III. Comparisons between ACO and traditional routing algorithms along the issues of routing information, routing overhead, and adaptivity are given in Sections III-A–C, respectively. In addition, Section III-C also discusses the issue of *stagnation* [11], [12]—a major weakness of most ACO approaches. Section IV surveys some of the state-of-the-art approaches to mitigate the problem of stagnation. Critique and comparison of the approaches to mitigate stagnation are given in Section IV-D. Section V provides a critical review of three major groups of research in applying ACO for routing and load-balancing, including

1) *Ant-based control* (*ABC*) system [13] and its ramifications (Section V-A);
2) *AntNet* [14] and its extensions [15], [16] (Section V-B);
3) *Ant system with genetic algorithm* (*ASGA*) [17] and *synthetic ecology of chemical agents* (*SynthECA*) [18], [19] (Section V-C).

While Section VI discusses new directions (by the authors) on applying ACO in load-balancing for circuit-switched networks [20], Section VII concludes this survey by identifying some of the open problems in ACO.

Manuscript received July 1, 2002; revised May 1, 2003. This paper was recommended by Guest Editor H. Zhang.

K. M. Sim is the with the Department of Information Engineering, Chinese University of Hong Kong, Shatin, NT, Hong Kong (e-mail: kmsim@ie.cuhk.edu.hk).

W. H. Sun is with Business Information Systems, Faculty of Business Administration, University of Macau, Taipa, Macao (e-mail: whsun1@ie.cuhk.edu.hk).

Digital Object Identifier 10.1109/TSMCA.2003.817391

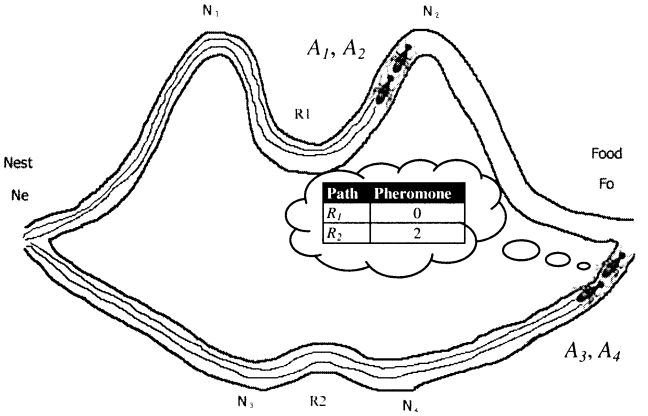


Fig. 1. Problem-solving of ants.

II. Ant Colony Optimization (ACO)

This section describes the problem-solving paradigm of ACO [3], [4], [13], [21] in finding an optimal path. Suppose that there are four ants and two routes leading to a food source: R_1 and R_2 (Fig. 1) such that $R_1 > R_2$. Along the two routes, there are six nodes: N_e (nest), N_1, N_2, N_3, N_4, and F_o (food source). Initially, all ants (A_1, A_2, A_3 and A_4) are at the decision point N_e and they have to select between R_1 and R_2 to reach F_o.

1) At N_e, all ants have no knowledge about the location of food (F_o). Hence, they randomly select from $\{R_1, R_2\}$. Suppose that A_1 and A_2 choose R_1, and A_3 and A_4 choose R_2.
2) As A_1 and A_2 move along R_1, and A_3 and A_4 move along R_2, they leave a certain amount of pheromone along their paths τ_{R1} and τ_{R2}, respectively.
3) Since $R_2 < R_1$, A_3 and A_4 reach F_o before A_1 and A_2. When A_3 and A_4 pass R_2 to reach F_o, $\tau_{R2} = 2$, but A_1 and A_2 have yet to reach F_o and $\tau_{R1} = 0$. To return to N_e from F_o, A_3 and A_4 have to choose between R_1 and R_2. At F_o, A_3 and A_4 detect that $\tau_{R2} > \tau_{R1}$, hence they are *more likely* to select R_2. Suppose A_3 and A_4 choose R_2.
4) As A_3 and A_4 pass R_2 for the second time to reach N_e, τ_{R2} is incremented to 4. The increase in τ_{R2} further consolidates R_2 as the shorter path. When A_1 and A_2 reach F_o, $\tau_{R2} = 4$ and $\tau_{R1} = 2$. Hence, A_1 and A_2 are *more likely* to select R_2 to return to N_e.

In this example, any ant at F_o (respectively, N_e) will be able to determine the optimal path once A_3 and A_4 reach F_o (respectively N_e). If an ant is at a choice point when there is no pheromone (e.g., initially at N_e), it makes a random decision with a probability of 0.5 of choosing R_1 or R_2. However, when pheromone is present (e.g., when the ant is at F_o), there is a *higher probability* that it will choose the path with the higher concentration of pheromone.

The example in Fig. 1 is an adaptation of Schoonderwoerd *et al.* [13] illustration of pheromone trail laying, which was in turn adopted from an example in [22]. In [22], Beckers *et al.* described the pheromone trail laying of a type of ants known as *Lasius Niger*, which deposit pheromone when traveling *both* to and from the nest (see [23, p. 310]). However, it is noted that there are other types of ants that adopt different forms of pheromone trail laying (see for example, [24]). Furthermore, there are some ACO approaches (e.g., AntNet [14]) that adopt *privileged pheromone laying* (see Section IV-C), in which ants only deposit pheromones during their return trips.

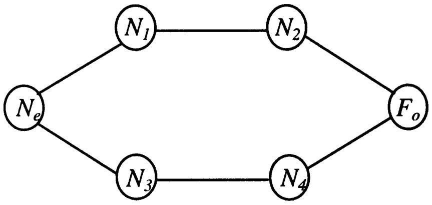


Fig. 2. Simple network.

In addition, most ACO approaches are inspired by the problem solving paradigms of ants [1] rather than building exact replicas of biological ants. In using artificial ants for problem solving, some of the features and capabilities of biological ants (e.g., using visual landmarks [24]) may be omitted, and other additional techniques (e.g., heuristic functions) may be used to complement and supplement the use of pheromone (see Section IV-B).

III. ACO versus Traditional Routing

In this section, the differences between ACO routing and traditional routing algorithms such as the distance vector routing or RIP and link state routing or OSPF are discussed. Of particular interest are the issues of

1) routing information;
2) routing overhead;
3) adaptivity and stagnation.

A. Routing Information

In both RIP and OSPF, a node N_i depends on the routing information furnished by all its neighboring nodes to construct a complete routing table. Furthermore, the neighboring nodes of N_i in turn depend on the routing information of their neighboring nodes which in turn depend on other neighboring nodes. As an example, consider the network shown in Fig. 2. Using RIP, N_e depends on the routing tables sent by N_1 and N_3 to construct a complete routing table (e.g., to determine the distances between N_e and N_2, and N_e and N_4, N_e needs information about the distances between N_1 and N_2, and between N_3 and N_4). N_1 and N_3 in turn depend on the routing tables from N_2 and N_4. Furthermore, N_2 and N_4 depend on the routing table from F_o (e.g., to determine the distances between N_2 and F_o, and between N_4 and F_o). In Fig. 2, using OSPF, N_e needs to know the link-state information (up or down) of all the links in every path leading to F_o before it can compute the shortest path to F_o.

In ACO, the paths from a source to a destination are explored independently and in parallel. For instance, in Fig. 1, four ants travel independently from N_e to F_o via R_1 and R_2 respectively. When the two ants A_3 and A_4 reach F_o first, it can be determined that R_2 is the shortest path without having to wait for A_1 and A_2 to arrive at F_o. As soon as an ant arrives at a node, the corresponding pheromone value for a path is updated, hence, each entry of the pheromone table in a node can be updated independently. In the network in Fig. 2 (which corresponds to the problem-solving situation of Fig. 1), F_o can immediately use the information in its pheromone table to route data packets to N_e when any ant from either path arrives (and updates its pheromone table).

B. Routing Overhead

Routing in RIP involves the transmission of routing tables of each node N_i to every one of its neighbors. For a large network N, the routing table of N_i (which consists of a list of cost vectors to all other nodes in N) is large. Since each N_i needs to transmit its routing table to every of its neighbors, the routing overhead can be very large.

In OSPF, routing is achieved by having each node transmit a link-state-packet (LSP) to every other node in a network N through a *flooding* processing. Although an LSP (which carries information about the costs to all the neighbors of a node) is generally smaller than a routing table, the flooding process ensures that every node in N receives a copy of the LSP. Since an LSP from a node can be disseminated via different paths to other nodes, multiple identical copies of the same LSP may be transmitted to the same node.

Routing in ACO is achieved by transmitting ants rather than routing tables or by flooding LSPs. Even though it is noted that the size of an ant may vary in different systems/implementations, depending on their functions and applications, in general, the size of ants is relatively small, in the order of 6 bytes ([25, p. 2]). This is because ants are generally very simple agents. Table I summarizes the differences between ACO and traditional routing algorithms.

C. Adaptivity and Stagnation

In dynamic networks, transmitting large routing table (in RIP) or flooding multiple copies of LSPs (in OSPF) in short or regular intervals may incur large routing overhead. However, flooding LSPs and transmitting routing table in longer intervals may result in slower responses to changes in network topology. Since ants are relatively small they can be piggybacked in data packets, more frequent transmission of ants to provide updates of routing information may be possible. Hence, using ACO for routing in dynamic network seems to be appropriate.

Related to the issue of adaptivity is stagnation. Stagnation occurs when a network reaches its convergence (or equilibrium state); an optimal path p_o is chosen by all ants and this recursively increases an ant's preference for p_o. This may lead to: 1) congestion of p_o, 2) dramatic reduction of the probability of selecting other paths. The two are undesirable for a dynamic network since:

1) p_o may become nonoptimal if it is congested;
2) p_o may be disconnected due to network failure;
3) other nonoptimal paths may become optimal due to changes in network topology, and iv) new or better paths may be discovered.

Furthermore, Bonabeau *et al.* [1] have pointed out that the success of ants in collectively locating the shortest path is only statistical. If by chance, many of the ants initially choose a nonoptimal p_{no}, other ants are more likely to select p_{no} leading to further reinforcement of the pheromone concentration along p_{no}. This is undesirable for static networks since it is inefficient if ants always choose a stagnant path that is nonoptimal.

TABLE I
ACO ALGORITHMS VERSUS TRADITIONAL ROUTING ALGORITHMS

	RIP / OSPF	**ACO algorithms**
Routing preference	Based on transmission time / delay	Based on pheromone concentration
Exchange of routing information	Routing information and data packet transmitted separately	Can be piggybacked in data packets
Adapting to topology change	Transmit routing table or Flood LSPs at regular intervals	Frequent Transmission of ants
Routing overhead	High	Low
Routing update	Update entire routing table	Update an entry in a pheromone table independently

IV. APPROACHES TO MITIGATE STAGNATION

The approaches to alleviate stagnation are categorized as follows: *pheromone control* (Section IV-A), *pheromone-heuristic control* (Section IV-B) and *privileged pheromone laying* (Section IV-C). Critique and comparison of these approaches are given in Section IV-D.

A. Pheromone Control

Pheromone control adopts several approaches to reduce the influences from past experience and encourages the exploration of new paths or paths that were previously nonoptimal.

1) Evaporation: To reduce the effect of past experience, an approach called evaporation [5] is typically used in conjunction with ACO. Evaporation prevents pheromone concentration in optimal paths from being excessively high and preventing ants from exploring other (new or better) alternatives. In each iteration, the pheromone values τ_{ij} in all edges are discounted by a factor p such that: $\tau_{ij} \leftarrow \tau_{ij} x(1-p)$. An example of evaporation is shown in Fig. 3. Suppose that at some iteration t_i, all ants converge to a path R_i, and deposit a very high concentration of pheromone (represented with larger triangles). In the next iteration t_{i+1}, the pheromone concentration along R_i is reduced by some factor (represented by smaller triangle), and at t_{i+2}, the pheromone concentration is further reduced.

2) Aging: Additionally, past experience can also be reduced by controlling the amount of pheromone deposited for each ant according to its age. This approach is known as *aging* [13]. In aging, an ant deposits lesser and lesser pheromone as it moves from node to node (see Fig. 4). Aging is based on the rationale that "old" ants are less successful in locating optimal paths since they may have taken longer time to reach their destinations. Both aging and evaporation include recency as a factor of routing preference, hence, if a favorable path is not chosen recently, its preference will be gradually eliminated. By making existing pheromone trail less significant than the recent pheromone updates, both aging and evaporation encourage discoveries of new paths that were previously nonoptimal.

3) Limiting and Smoothing Pheromone: Stuzle and Hoos [12] mitigate stagnation by *limiting* the amount of pheromone in every path. By placing an upper bound $\tau_{\max}$ on the amount of pheromone for every edge (i, j), the preference of an ant

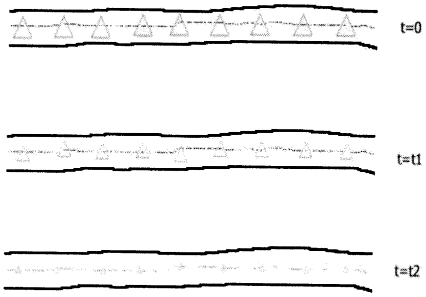


Fig. 3. Evaporation.

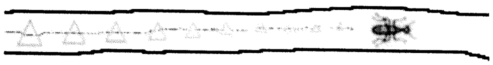


Fig. 4. Aging.

for optimal paths over nonoptimal paths is reduced. This approach prevents the situation of generating a dominant path. A variant of such an approach is *pheromone smoothing* [12]. Using pheromone smoothing, the amount of pheromone along an edge (i, j) is reinforced as follows:

$$\tau_{ij}(t') = \tau_{ij}(t) + \delta \times (\tau_{\max} - \tau_{ij}(t))$$

where δ is a constant between 0 and 1. It can be seen that as $\tau_{ij}(t) \to \tau_{\max}$, a smaller amount of pheromone is reinforced along an edge (i, j). Although not totally identical, pheromone smoothing also bears some resemblance to evaporation. While evaporation adopts a uniform discount rate for every path, pheromone smoothing places a relatively greater reduction in the reinforcement of pheromone concentration on the optimal path(s). Consequently, pheromone smoothing seems to be more effective in preventing the generation of dominant paths.

B. Pheromone-Heuristic Control

Another approach to mitigate stagnation is to configure ants so that they do not solely rely on sensing pheromone for their routing preferences. This can be accomplished by configuring the probability function P_{ij} for an ant to choose an edge (i, j) using a combination of both pheromone concentration τ_{ij} and heuristic function η_{ij}. As noted in [7], [8], an ant selects an edge probabilistically using τ_{ij} and η_{ij} as a functional composition for P_{ij}. In network routing, η_{ij} is a function of the cost of edge (i, j) (which may include factors such as queue length, distance, and delay). While η_{ij} is a function of the arc length d_{ij} in [12], in [7], η_{ij} is determined by the queue length q_{ij} (in terms of bits) along edge (i, j) and η_{ij} is given as follows:

$$\eta_{ij} = 1 - \frac{q_{ij}}{\sum\limits_{l \in N_i} q_{il}}.$$

In [12], P_{ij} at time t is given as follows:

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum [\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}$$

α and β represent the respective adjustable weights of τ_{ij} and η_{ij}. Consequently, the routing preferences of ants can be altered by selecting different values of α and β. If $\alpha > \beta$, ants favor paths with higher pheromone concentrations, and a higher value of β directs ants to paths with more optimistic heuristic values. In general, different values of α and β are suitable to be applied at different states of a network. A lower value of α is generally preferred when pheromone concentration along paths may not necessarily reflect their optimality. Examples of such situations include the initial stage after a network reboots (before the network stabilizes), and when there are frequent and abrupt changes in network status due to either link (or node) failure or introduction of new paths (nodes). However, as a network stabilizes, a higher value of α is preferred. Furthermore, recent research [12] demonstrated that dynamically altering the values of α and β in response to changes in network status may increase the performance of ants.

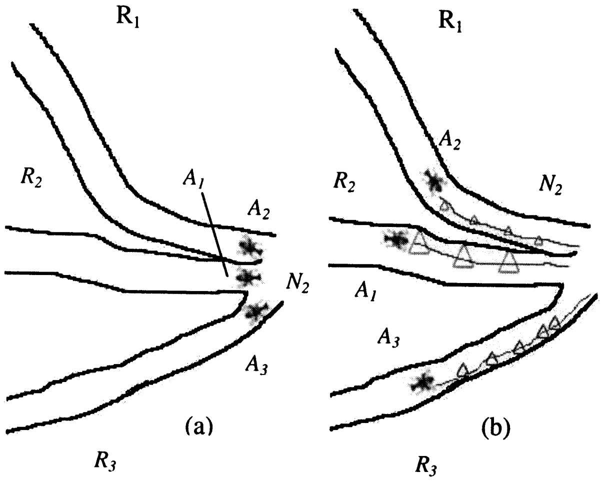


Fig. 5. (a) Three ants arrived at N_2 via different paths. (b) Ants deposit pheromone on the return trips.

C. Privileged Pheromone Laying

One of the early enhancements of the ACO algorithm is to mitigate stagnation by adopting the policy of privileged pheromone laying [14], [26]. Experiments in [14], [26] demonstrated that by permitting a selected subset of ants to have the privilege to deposit extra or more pheromone, the time for ants to converge to a solution reduces significantly. In this approach, two issues are of interest: 1) the assessment of the quality of the solution of ants, and 2) the number of ants to be selected to deposit extra pheromone and the amount of pheromone that ants are permitted to deposit. One of the simplest (and intuitive) approaches to assess the quality of the solutions of ants is to compare their forward-trip times. Alternatively, employing an *FDC* (*fitness-landscape*) approach, Stuzle and Hoos [12] compared the forward-trip time of each ant with the optimal forward-trip time. In FDC, the destination node records the optimal forward-trip time T_{op} (based on statistics), and the fitness of the solution of an ant is determined by a nonlinear function of how close the forward-trip time of an ant is compared to T_{op}. A simple example is shown in Fig. 5. In Fig. 5(a), originating from the same source, three ants A_1, A_2, and A_3, arrive at the destination N_2, each via a different path. Using the FDC approach, at N_2 the optimal forward-trip time T_{op} is recorded (previously based on statistics), and all the forward-trip times of A_1, A_2, and A_3 are compared with T_{op}. In this example, since A_1 arrives first at N_2 it is closest to T_{op} (or perhaps better than T_{op}, in which case its forward-trip time will replace T_{op}). Consequently, A_1 will deposit the most pheromone on its return trip via the same path it came from. Since A_2 arrives after A_1, and A_3 arrives after A_2, on their return trips via the same paths they came from, A_2 will deposit more pheromone than A_3, but less than A_1 (see Fig. 5(b)). In summary, the FDC function evaluates the fitness of a solution by comparing the solution of each ant to the optimal solution. Since the solutions are compared (the fitness of each path is evaluated) before ants deposit their pheromone, Stuzle and Hoos [12] found that using FDC analysis, more accurate results were obtained. In this approach, since the amount of pheromone deposited corresponds to the time delay of the trip, an ant will deposit lesser pheromone along a path that is longer or congested. Consequently, this reduces the chance of ants reinforcing a stagnant path that is nonoptimal or congested. In the above example, all three ants deposit pheromone (albeit, with different amounts) on their return trips. In other approaches, such as [27], only a sub-set of ants deposits pheromone, and in [12], only the *best* ant deposits pheromone on its return trip. Restricting the number of ants to make their return trips and deposit pheromone reduces network traffic.

D. Critique and Comparison

In general, the approaches discussed in Sections IV-A–C do alleviate the problem of stagnation. However, some of their strengths and weaknesses are discussed below.

1) Aging: In aging, the amount of pheromone deposited is based on the age of an ant. Since an ant has to wait for a long time in a congested path p_c, it deposits very little pheromone along p_c. This has the advantage of discouraging other ants from selecting p_c. However, in aging as applied in [13] (see Section V-A), using the pheromone information from ants traveling in one direction (the forward direction) may not be entirely accurate for ants traveling in the opposite (backward) direction in an asymmetric network. For example, in an asymmetric network, the traveling times from a source node N_s to a destination node N_d, and from N_d to N_s are different. Hence, it may seem to be less accurate if ants coming from the opposite direction use pheromone concentration (which is based on the forward traveling time) to determine their routing preferences. However, this can be resolved if ants make a forward and a backward trip for one update.

2) Limiting Pheromone: Placing limits ($\tau_{\max}$ and $\tau_{\min}$) on the pheromone concentration on every path coerces the probability of ants choosing a particular path to a certain range. Although doing so prevents ants from always preferring the dominant path, pheromone limiting must be used in conjunction with evaporation. If pheromone limiting is used alone, the pheromone in every path will have no effect on ants when the pheromone of every path reaches $\tau_{\max}$ eventually through continuous reinforcement.

3) Pheromone Smoothing: In general, pheromone smoothing seems to be effective in preventing the generation of dominant paths because paths with very high pheromone concentrations will be reinforced will lesser pheromone. However, if a dominant path p_o is optimal and remains optimal (e.g., in stabilized network) then continuously reinforcing p_o with lesser pheromone may increase the probability of ants choosing other nonoptimal paths.

4) Evaporation: Evaporation generally does not suffer from any of the weaknesses mentioned above. However, evaporation together with pheromone limiting and pheromone smoothing may not be suitable to be used with ACO for routing in a static network or a network that has already stabilized. In a static (or stabilized) network, optimal paths are rarely changed. Hence, the application of evaporation, and pheromone limiting or smoothing may direct ants to nonoptimal paths leading to possible degradation of network performance. Consequently, it seems prudent to apply evaporation, and pheromone limiting or smoothing on selective intervals (or occasionally) rather than on a frequent basis. Even in dynamic networks, one of the design considerations is to determine the frequency of applying evaporation (and pheromone limiting and smoothing). While too frequent applications of these techniques may destabilize a network, infrequent applications may reduce the adaptiveness of the system. However, unlike evaporation, and pheromone smoothing and limiting, aging is generally suitable for applications in both static and dynamic networks.

5) Privileged Pheromone Laying: Although the privileged pheromone laying approach may generate more accurate results and reduces the probability of ants reinforcing stagnant paths that are nonoptimal or congested, the need for ants to make return trips increases network traffic. Some approaches such as [12] and [27] minimize network traffic by enabling only one ant or a subset of ants to make return trips to deposit pheromone. However, such restriction seems to be inconsistent with the parallel problem-solving paradigm of ACO. Additionally, in the example of Fig. 5, even though A_1, A_2, and A_3 arrive at N_2, they do not provide any routing information for outgoing ants in the source node they originated from because no pheromone is deposited in their forward trips. Furthermore, it is noted that aging can also be used to achieve the effect of reducing the chances of ants reinforcing stagnant paths that are nonoptimal or congested. This is because older ants deposit lesser pheromone with the rationale that solutions of old ants are less likely to be optimal since they have taken longer paths. In general, aging seems to produce similar effect as privileged pheromone laying without increasing network traffic since ants do not need to make backward trips.

6) Pheromone-Heuristic Control: While placing less emphasis on pheromone concentration (by setting a low value of α) may mitigate stagnation, placing too much reliance on heuristic values (by setting too high a value of β) may transform an ACO algorithm into a classical algorithm. Although pheromone-heuristic control does not generally suffer from the weaknesses mentioned above, when the values of and α and β are altered, ants need time to adapt to changes in the emphasis of pheromone concentration and heuristic values. Hence, unlike evaporation, pheromone smoothing and limiting as well as aging, in both the pheromone-heuristic control and the privileged pheromone laying approaches, the system needs time to stabilize. In the case of privileged pheromone laying, time is needed to gather the statistics before an optimal solution can be determined.

V. ACO in Routing and Load Balancing

This section surveys three major groups of research in applying ACO in routing and/or load-balancing. While Section V-A reviews *ant-based control* (*ABC*) and some of its ramifications, *AntNet* and its extensions are discussed in Sections V-B and C surveys the work of Tony White under the systems of *ant system with genetic algorithm* (*ASGA*) and *synthetic ecology of chemical agents* (*SynthECA*).

A. Ant-Based Control and Its Ramifications

Schoonderwoerd *et al.* [13], [23], [28]–[30] ant based control system (ABC) was designed to solve the load-balancing problem in circuit-switched networks much like Appleby and Steward's load management agents [31]. (Since load management agents were not based on ACO, they will not be discussed here and readers are referred to [31] for details). Similar to Appleby and Steward [31], Schoonderwoerd *et al.* [13], [23], [28]–[30] conducted experiments to simulate a 30-node British Telecom SDH network. Each node in the network was configured with

1) capacity C that accommodates a certain number of calls;
2) probability of being a destination (for generating random calls;
3) pheromone (or probabilistic routing) table.

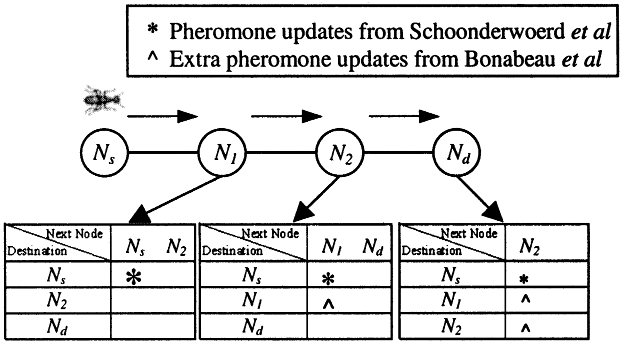
Each row in the pheromone table represents the routing preference for each destination, and each column represents the probability of choosing a neighbor as the next hop. Ants are launched from a node N_s with a random destination N_d. In Schoonderwoerd *et al.* approach, incoming ants update the entries in the pheromone table of a node. For instance, an ant traveling from N_s (source) to N_d (destination) will update the corresponding entry in the pheromone table in N_d. Consequently, the updated routing information in N_d can only influence the routing ants and calls that have N_s as their destination. However, for asymmetric networks, the costs from N_s to N_d and from N_d to N_s may be different. Hence, Schoonderwoerd *et al.* approach for updating pheromone is only appropriate for routing in symmetric networks.

Schoonderwoerd *et al.* adopted three approaches to mitigate stagnation

1) aging;
2) delaying;
3) noise.

Aging (see Section IV-A) is designed to discourage ants from following the trails of an ant that has traveled a longer path to some destination. In contrast to evaporation (see Section IV-A), aging may unintentionally induce an ant to select a nonoptimal link, if the path from a node to its destination is very long.

Used in conjunction with aging, delaying is designed to reduce the flow rates of ants from a congested node to its neighbors. By slowing down the ants originating from a congested node, the amount of pheromone they deposit reduced with time because of the aging process. Consequently, delayed ants have lesser influence on the outgoing ants in the nodes that they will reach. However, delaying may not reduce the influence of those nodes that are not reached by ants.


* Pheromone updates from Schoonderwoerd *et al*
^ Extra pheromone updates from Bonabeau *et al*
N_s
N_1
N_2
N_d

Next Node / Destination	N_s	N_2
N_s	*	
N_2		
N_d		

Next Node / Destination	N_1	N_d
N_s	*	
N_1	^	
N_d		

Next Node / Destination	N_2
N_s	*
N_1	^
N_2	^

Fig. 6. Pheromone update policy of Schoonderwoerd *et al.* and Bonabeau *et al.*.

Noise or exploration enables ants to choose a path randomly not taking into consideration the influence of the pheromone table. This enhances the chances of ants exploring new and better routes, particularly in dynamic networks.

Experimental results showed that the number of dropped calls in the ABC systems were on average less than Appleby and Steward's [31] system. Additionally, ABC required less time to adapt to changes in topology and call probability of nodes.

One of the ramifications of the ABC system is the adaptation of Guérin's [32] *smart ants* to solve the problems of routing and load-balancing in circuit-switched networks by Bonabeau *et al.* [21]. While an ant in ABC updates only the entry corresponding to the source node N_s in the pheromone table of each node it passes, Bonabeau *et al.* [21] smart ants update the pheromone table at each node, *all* entries corresponding to *every* node they pass [21]. Consider the example in Fig. 6. When an ABC ant travels from N_s to N_d via N_1 and N_2, it will *only* update the entry that corresponds to N_s in the pheromone tables of N_1, N_2, and N_d. In the same example, if a smart ant travels from N_s to N_d via N_1 and N_2, it will update the entry corresponding to N_s in the pheromone table of N_1, the entries corresponding to N_s and N_1 in the pheromone table of N_2, and the entries corresponding to N_s, N_1, and N_2 in the pheromone table of N_d. Similar to ants in ABC, pheromone updates in each node are carried out by incoming (smart) ants according to their age. Consequently, smart ants also inherit the same disadvantages of ants in ABC as discussed above. Additionally, by performing more pheromone updates at every intermediate node, smart ants are more complex (and perform more computations) than ants in ABC, but fewer smart ants are needed for achieving the same routing purpose. Using the same testbed as in ABC but augmented with smart ants, Bonabeau *et al.* achieved more favorable results than Schoonderwoerd *et al.* both in terms of the number of dropped calls and the time to adapt network status where there were sudden node failures.

Two other ramifications of the ABC system are the work of Subramanian *et al.* [25] and Heusse *et al.* [33]. Although some of their ideas were based on the ABC approach, Subramanian *et al.* [25] focused on routing in packet-switched networks. In their approach, Subramanian *et al.* [25] devised two kinds of ants: *regular ant* and *uniform ant*. Regular ants are based on ants in ABC, but rather than using the age of an ant, a regular

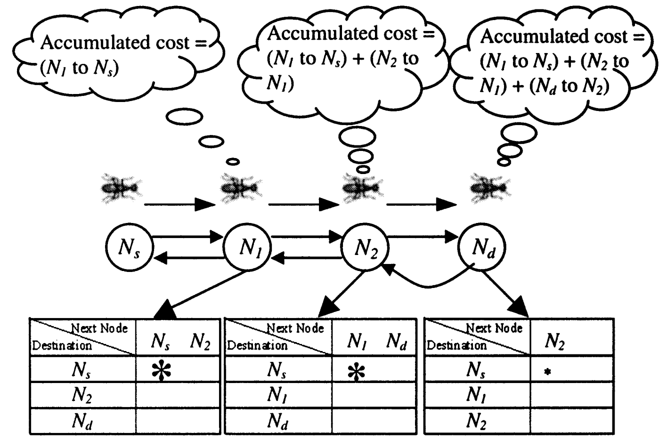

Accumulated cost = (N_1 to N_s)
Accumulated cost = (N_1 to N_s) + (N_2 to N_1)
Accumulated cost = (N_1 to N_s) + (N_2 to N_1) + (N_d to N_2)
N_s
N_1
N_2
N_d
Next Node
Destination
N_s N_2
N_s
N_2
N_d
N_1 N_d
N_s
N_1
N_d
N_2
N_s
N_1
N_2

Fig. 7. Pheromone update policy of Subramanian *et al.*.

ant uses the accumulated cost of a path to determine the amount of pheromone to deposit. A regular ant that travels a higher cost path to reach a destination node deposits lesser pheromone. Unlike regular ants, uniform ants choose their next nodes randomly (or uniformly). Additionally, while regular ants use the accumulated cost in the forward direction (i.e., the direction they travel), uniform ants use the accumulated cost in the reverse direction (i.e., the reverse direction that they travel) to determine the amount of pheromone to deposit. In the example shown in Fig. 7, the accumulated costs of a regular ant traveling from N_s to N_d via N_1 and N_2 is the sum of the costs of N_s to N_1, N_1 to N_2, and N_2 to N_d. For the same example, the accumulated cost of a uniform ant is the sum of the costs of N_d to N_2, N_2 to N_1, and N_1 to N_s. For a network with asymmetric cost, the costs from N_s to N_d, and from N_d to N_s are different. When a regular ant reaches N_d, the cost from N_s to N_d is used to update the pheromone table in N_d. This may not provide accurate routing information for ants traveling from N_d to N_s. On the contrary, when a uniform ant reaches N_d, the cost from N_d to N_s is used to update the pheromone table in N_d. Hence, this method of pheromone update does not assume symmetry in link costs, which appears to be a disadvantage of ants in the ABC system. However, since Subramanian *et al.* focused only on the problem of routing, issues of load-balancing is not addressed. Furthermore, the approach of the uniform ants assumes that a node has the knowledge of the distances to its immediate neighbors. In the example in Fig. 7, when the uniform ant reaches N_1, it gathers the cost of N_1 to N_s from N_1 and the amount of pheromone it deposits in the pheromone table of N_1 is based on that cost. As the uniform ant travels to N_2, it also carries with it the cost of N_1 to N_s. At N_2, it gathers the cost of N_2 to N_1 and the amount of pheromone it deposits at N_2 is based on the accumulated costs of N_2 to N_1 and N_1 to N_s. Similarly, the uniform ant carries the accumulated cost (so far) as it travels to N_d and the amount pheromone it deposits at N_d is based on the total accumulated cost.

Also based on ideas in the ABC system, Heusse *et al.* [33] proposed a cooperative asymmetric forward (CAF) for routing in packet-switched networks with asymmetric path costs. CAF ants update pheromone table much like ants in the ABC system. What discerns CAF ants from ABC ants is that a CAF ant updates an entry in a pheromone table using the cost in the reverse direction (that it travels) recorded by a data packet. For the example shown in Fig. 8, at N_s, the CAF ant obtains the cost from N_1 to N_s previously recorded and piggybacked to N_s by some data packets moving from N_1 to N_s. As it travels to N_1, it also carries the cost from N_1 to N_s with it, and when it reaches N_1, it updates the pheromone table of N_1 using that cost. At N_1, it acquires the cost from N_2 to N_1 (previously piggybacked to N_1 by data packets moving from N_2 to N_1 and recorded in the reverse distance table in N_1). As it travels to N_2, it carries with it the accumulated costs of N_2 to N_1 and N_1 to N_s, and when it reaches N_2 it deposits pheromone in N_2 according to that cost. Similarly when it finally reaches N_d, it uses the accumulated costs of N_d to N_2, N_2 to N_1 and N_1 to N_s to update the pheromone table in N_d. Consequently, CAF ants are not restricted to routing applications in packet-switched networks with symmetric costs only. However, since CAF ants rely on data packets to carry (by piggybacked) and record cost from the opposite direction, this approach may not be appropriate when there is no frequent data packet arriving from the opposite direction. Furthermore, each node needs to maintain a reverse routing table (that records the costs of links in the opposite direction). In addition, like Subramanian *et al.*, the CAF approach [33] focused only on routing problems and did not address the issue of load-balancing.

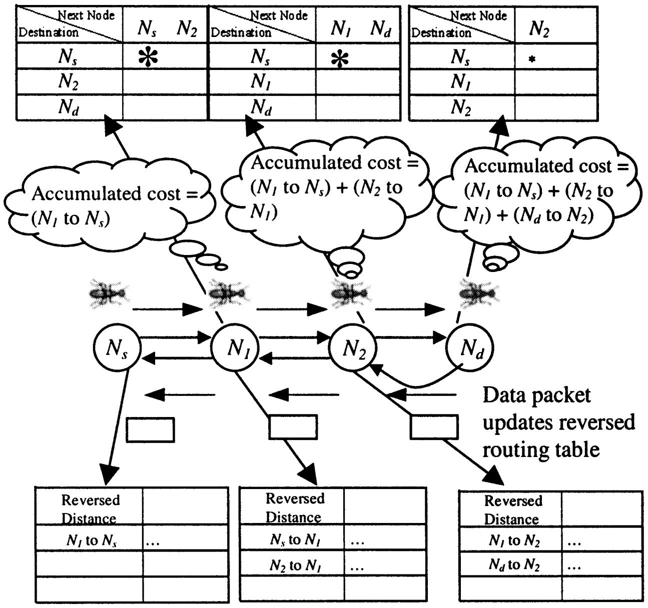
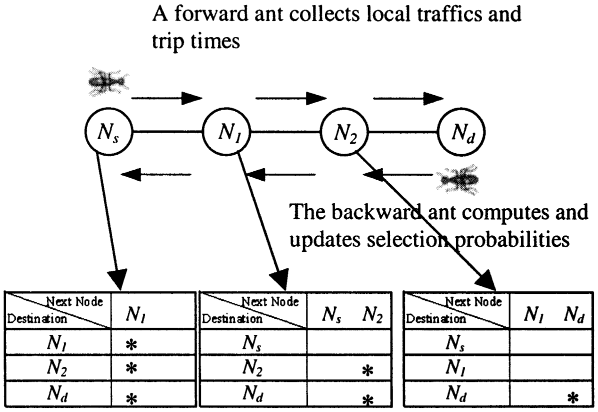

Next Node
Destination
N_s N_2
N_s
N_2
N_d
N_1 N_d
N_s
N_1
N_d
N_2
N_s
N_1
N_2
Accumulated cost = (N_1 to N_s)
Accumulated cost = (N_1 to N_s) + (N_2 to N_1)
Accumulated cost = (N_1 to N_s) + (N_2 to N_1) + (N_d to N_2)
N_s
N_1
N_2
N_d
Data packet updates reversed routing table
Reversed Distance
N_1 to N_s
...
N_s to N_1
N_2 to N_1
N_1 to N_2
N_d to N_2

Fig. 8. Pheromone update policy of Heusse *et al.*.

B. AntNet and Its Extensions

Caro and Dorigo's AntNet [14], [26], [34]–[36] was originally designed for routing in packet-switched networks. Unlike traditional routing algorithms (such as OSPF and RIP) which focused on minimal or shortest path routing, routing in AntNet was carried out with the aim of optimizing the performance of the entire network. In AntNet, routing was achieved by launching forward ants at regular intervals from a source node N_s to a destination node N_d to discover a feasible low-cost path and by backward ants that travels from N_d to N_s to update pheromone tables at each intermediate node. At N_s,


A forward ant collects local traffics and trip times
N_s
N_l
N_2
N_d
The backward ant computes and updates selection probabilities

Next Node / Destination	N_l
N_l	*
N_2	*
N_d	*

Next Node / Destination	N_s	N_2
N_s		
N_2		*
N_d		*

Next Node / Destination	N_l	N_d
N_s		
N_l		
N_d		*

Fig. 9. Pheromone update policy of Caro and Dorigo.

N_d is selected based on the data traffic pattern (in terms of the number of packets traveling from N_s to N_d). From N_s to N_d, a forward ant selects the next hop N_i using a random scheme that take into consideration of both 1) the probability P_{id} of choosing N_i, and 2) a heuristic correction factor l_{ni}. While l_{ni} is based on the queue length at N_i, P_{id} is a selection probability which can be viewed as a pheromone concentration that can be reinforced by other ants. As a forward ant travels from N_s to N_d, it gathers statistics such as the local data traffic condition on each intermediate node N_i and the trip time to N_i. When a forward ant arrives at N_d, a backward ant will be activated. Traveling from N_d to N_s via the same path as a forward ant, a backward ant updates 1) the probabilistic routing (or pheromone) table at each intermediate node N_i, and 2) the estimated trip time from N_s to N_i (gathered by the forward ant). Backward ants reinforce the selection probability by determining 1) the "goodness" of the trip times of forward ants, and 2) the amount of reinforcement using a *squash* function. *goodness* [14] is a relative measure that is determined comparing the current trip time to 1) the current statistical estimates of the best trip time and 2) the confidence interval of the best trip time. Although in [26] and [34], every trip time was recorded, in [14], [35] and [36], only trip times that were within some window time frame were recorded. While more accurate measures may be obtained in the former case, the amount of computation for determining goodness is higher. A squash function is a nonlinear function that is more sensitive in rewarding solutions with higher goodness values while having a lower tendency of reinforcing solutions with lower values of goodness.

Since the goodness of a path is reinforced according to the trip times of forward ants, the selection probabilities updated by backward ants can influence ants traveling in the forward direction from N_s to N_d, rather than from N_d to N_s (like in the cases of both ABC and smart ants). Consequently, AntNet is not restricted to routing application in networks with symmetric costs only. Using the same four nodes example, Fig. 9 shows that in AntNet, as the forward ant travels from N_s to N_d via N_1 and N_2, it gathers the local traffic at each node and trip time to each node. In Fig. 9, the backward ant uses the statistics gathered by the forward ants to compute and update the selection probabilities in those entries marked with "$*$." Using the privileged pheromone laying approach (Section IV-C), backward ants reinforce a selection probability based on the goodness of the corresponding solution. Hence, it also alleviates the problem of stagnation. However, using both forward and backward ants generally doubles the routing overhead.

Using both throughput and average delay as performance measures, Caro and Dorigo [14], [26], [34]–[36] conducted experiments on the SimpleNet, simulated NSFNET and NTTnet. The experimental settings include different traffic patterns such as temporal and spatial conditions [14], [35], [36] and uniformly random distribution [26], [34]. The same simulations carried out using AntNet on the above networks were also conducted using distance vector, link state, Bellmand-Ford, shortest path first and Q-routing. Favorable results in terms of higher throughput and lower average delay were achieved in AntNet as compared to other routing approaches, mentioned above.

In [15], a variant of AntNet was devised, where forward ants travel from a source to a destination in high priority queues, and backward ants estimate the trip time (by size of queueing data, links' bandwidth and delay), update local traffic statistics, and determine and deposit the amount of probability to reinforce. Since backward ants determine the amount of reinforcement using real time statistics, the routing information is comparatively more accurate and up-to-date. Experimental results from [15] demonstrated that the performance of this approach is comparatively better than the original AntNet.

Another ramification of AntNet was the work of Baran and Sosa [16]. Baran and Sosa's system has the five following distinguishing features from AntNet:

1) intelligent initialization of AntNet;
2) intelligent pheromone updates after link or node failures;
3) use of noise to mitigate stagnation;
4) deterministic rather than probabilistic selection of a node;
5) restricting the number of ants inside a network.

The first was generally included to regulate the exploration ants in the initial stage. In the original entries in a routing table consist of a uniform distribution of probabilities which may not reflect the states of the network. Taking into consideration the a-priori knowledge of the network, Baran and Sosa's ants are configured to select neighboring nodes with a higher initial probability. This seems to lead to a saving of network resources since neighboring nodes can typically be reached at lower costs. While AntNet did not consider situations of link failures, the second suggests that in case of link failures, the corresponding probability P_{ij} of a link (i, j) that fails will be set to zero and P_{ij} will be distributed evenly among the remaining neighboring nodes. This enhancement has the advantage of retaining some of the knowledge of the nodes (such as network traffic) before the failure occurs. The third is similar to Schoonderwoerd *et al.* [13], [23], [28], [29] noise where some ants select paths uniformly without taking into consideration the influence of pheromone concentration to reduce the chances of having a nonoptimal path that is stagnant. However, since AntNet adopted the privileged pheromone laying approach, the chance of ants reinforcing a stagnant path that is nonoptimal or congested is low. Hence, such enhancement may not be significant. The fourth adopts a deterministic approach for selecting the next hop. However, this approach may lead to a possible infinite looping. Consider an example of choosing between two nodes N_1 and N_2 to reach a destination N_d. Suppose that in the pheromone table in N_1, the entry for reaching N_d via N_2 as the next hop records the highest pheromone concentration. Similarly, in N_2, the entry for reaching N_d via N_1 as the next hop records the highest pheromone concentration. If an ant selects the next hop deterministically between N_1 and N_2, a recursive situation will result (i.e., N_1 directs the ant to N_2 and N_2 in turns directs the ant to N_1). In addition, if a next hop is selected deterministically, all ants will be directed to the optimal path and this may lead to possible congestion. The fifth suggests that the number of ants inside a network should not exceed four times the number of links. It is unclear whether placing such an upper bound is appropriate since no mathematical proof or empirical result was supplied. Although restricting the number of ants may reduce routing overhead and possible congestion, it also places a restriction on the frequency of launching ants which may lead to possible reduction in the adaptiveness of the routing algorithm.

C. ASGA and SynthECA

Augmenting antnet's with genetic algorithm (GA) [37], Tony White [17], [38], [39] designed *ant system with genetic algorithm(ASGA)* to solve problems of point-to-point, point to multipoint and cycle (multipath) routing in circuit-switched networks. Routing in ASGA is achieved by using explorer ants to update pheromone tables. Although similar to AntNet, explorers travel in a round trip, but unlike backward ants in AntNet, explorers deposit the same amount of pheromones in their return trips [38], [39]. In addition, evaporation agents and pheromone heuristic control were used to mitigate stagnation. The addition of GA was to increase the adaptivity of ants. For instance, if the best path is congested, it increases the likelihood of ants to find an alternative path. However, unlike the ABC system, ASGA was not designed to solve the load-balancing problem in circuit-switched networks.

Subsequently, ASGA was generalized to a framework called *synthetic ecology of chemical agents* (*SynthECA*) [18], [19]. In addition to those problems addressed by ASGA [40], [41], SynthECA was also designed to solve other problems such as fault location detection [42]–[44] in circuit-switched networks. Although SynthECA [18], [19] was not designed with any specific type of ants, all ants in SynthECA are characterized with a combination of the following:

1) emitters;
2) receptors;
3) chemistry;
4) migration decision function;
5) memory.

Emitters are used to generate different types of pheromone (called *chemical* [18]). Using the encoding scheme of GA, pheromones are represented by strings such as "1100" or "10#1." While each type of pheromone corresponds to a *genotype*, each string corresponds to a *chromosome* in GA. An emitter decision function (EDF) is used to generate pheromone. Similar to GA, the operations of crossover and mutation are applied in the EDF to evolve the pheromone types. With different and more pheromone types and pheromone reactions, ants can be designed to send and sense more types of signals in their stigmergic communication.

A *receptor* is used to sense local pheromone changes generated by emitters. Using receptor detection function (RDF), a receptor senses different types of pheromone.

By configuring ants with different EDFs (that produce (different) pheromone at different rate) and RDFs (that are sensitive to different pheromone types), more sophisticated pheromone manipulation techniques such as privileged pheromone laying and pheromone heuristic control can be realized.

Chemistry is a set of rules (inspired by GA) that specifies pheromone reactions. In SynthECA, ants use pheromone reactions to send out control information to other ants. In the set of rules, five types of pheromone reactions are specified as follows:

1) $X \rightarrow$ "nothing:" this is similar to evaporation;
2) $X+Y \rightarrow Y$: this is applied when two ants are competing for a path and only one ant will prevail;
3) $X + Y \rightarrow Z$: this rule is used to report the status (e.g., poor connection quality) of network resources. For example, two ants monitoring the quality of a specific connection can jointly (e.g., $X+Y$) generate another type of pheromone (i.e., Z) to inform other ants detecting faulty locations to check for node/link failures if the quality of service of the connection falls below a threshold;
4) $X+Y \rightarrow X+Z$: this rule, in computational terms, represents a conditional construct. A pheromone type (e.g., Y) is transformed into another type of pheromone (e.g., Z) in the presence of a specific type of pheromone (e.g., X). For example, in the presence of high priority data traffic in a path p, ants routing lower priority data traffic but with larger bandwidth requirement may be diverted to other paths. This increases the likelihood that ants routing data traffic with lower priority and lower bandwidth requirement to select p so that more bandwidth can be reserved for the higher priority ants;
5) $X + Y \rightarrow W + Z$: This rule allows two ants to jointly (e.g., $X+Y$) communicate both inhibitory (e.g., W) and excitatory (e.g., Z) messages to other ants. Although each rule involves the reaction of only two pheromone types, pheromone reactions with multiple types can be achieved by the execution of a series of the interaction rules.

While a *migration decision function* is a set of rules that determines the next hop of an ant, pheromones (i.e., labels and concentrations) and the state of an ant are stored in the ant's *memory*.

Using a combination of the above five components, several types of ants such as *route finding agent* (RFA) [40], [41], *connection monitoring agent* (CMA) and *fault detection agent* (FDA) [18] can be configured to solve different networking problems. In [40], [41], (1) was constructed by augmenting ants in ASGA with privileged pheromone laying. In [42]–[45], (2) and (3) were designed to address the issues of fault location detections [43], [44], fault tolerance [42] and QoS [45].

RFAs include *explorers*, *allocators* and *deallocators*. An explorer is used to find a path from a source N_s to a destination N_d.

It is configured with an emitter for a single type of pheromone and three receptors for sensing pheromone, measuring link costs and detecting quality of links [18], [45]. Using a probability function, an explorer [40]–[45] chooses a path taking into account the pheromone and the cost of the path. As it travels N_s to N_d, it records all the nodes it passed. When it reaches N_d, it returns to N_s via the same path and deposits pheromone along the way, which may influence the pheromone concentration of other types. The functions of explorers bear some resemblance to the combined functions of both forward and backward ants in AntNet. Subsequently, to address the issue of QoS, explorers are programmed to also take into consideration the quality/reliability of the link [18], [19]. While an allocator is used to acquire link resources, a deallocator performs the function of releasing resources that have been previously acquired by an allocator.

CMA's are activated if the quality of service changes. Traversing from node to node, a CMA evaluates the quality of a link using local traffic statistics and it deposits a special type of pheromone (called *q-chemical* [18]) to indicate the quality of the associated link. Consequently, this may be sensed by other ants so that they can take appropriate actions. In particular, CMAs use q-chemical to indirectly communicate the quality of links to FDAs while they circulate the network for diagnostics purposes.

A distinguishing feature of SynthECA from ABC and AntNet is the introduction of chemistry. Through the interaction of different pheromone types, communication and cooperation among different types of ants are possible. In addition, it also provides a framework where different types of ants can be configured to solve several kinds of problems in networking. In general, since there were many types of ants that deposit different pheromone types, the interactions of pheromone and their influences on other types of ants may be difficult to predict. For instance, if evaporation is applied on one type of pheromone that can be sensed by several kinds of ants, the consequences may not be simple to predict.

VI. New Directions

Most of the work surveyed in Section V addressed the problem of routing but not load balancing except for ABC and SynthECA. However, in both ABC and SynthECA, only one probabilistic routing table is maintained in each node. Consequently, if there is more than one optimal path, then it will be more likely for *all* data traffics to be directed into *only one* of the optimal paths. One of the possible solutions is to maintain multiple probabilistic routing tables in a node. An ongoing work of the authors addresses this issue under the topic of *multiple ant colony optimization* (*MACO*) [46]. While the problem-solving paradigm of MACO is briefly presented in Section VI-A, an example suggesting the possible application of MACO in load balancing is given in Section VI-B. Space limitation precludes the description of the MACO testbed and its preliminary empirical results from being included here, but they are reported in [47].

A. Multiple Ant Colony Optimization (MACO)

In MACO, more than one colony of ants are used to search for optimal paths, and each colony of ants deposits a different type of pheromone represented by a different color. Although ants in each colony respond to pheromone from its own colony, MACO is augmented with a *repulsion* mechanism [48], [49] that prevents ants from different colonies to choose the same optimal path.

1) Repulsion: In [48], Varela and Sinclair, adopt ACO to solve problems in virtual wavelength path routing and wavelength allocations. The distinguishing feature of the three variants of their ACO algorithms is that ants are not only *attracted* by the pheromones of other ants in their own colonies, but they are also *repelled* by the pheromones of other colonies. The motivation of their work stems from the fact that virtual wavelength paths can only carry a limited number of different wavelengths because of technological limitations and cost implications. In virtual wavelength path routing, the problem is to allocate the minimum number of wavelengths for each link by evenly distributing the wavelength requirements over different links, while at the same time keeping the path lengths short (e.g., in terms of hop numbers). While pheromone attraction is used in the similar sense as other routing applications of ACO, pheromone repulsion enhances the chance of distributing different wavelengths over different links. The degrees of attraction and repulsion are determined by two weighting parameters, combined into a probability function (details can be found in [48]).

2) MACO Example: An example to illustrate the MACO approach for network routing is given as follows:

Suppose that there are four ants: A_1^r, A_2^r, A_1^b, and A_2^b; where A_1^r, and A_2^r are red colony ants and A_1^b, and A_2^b are blue colony ants. There are three routes R_1, R_2, and R_3 leading to a food source F_o, such that $R_1 > R_3 > R_2$. Initially, all ants are in the nest N_e, and they have to select from among $\{R_1,\ R_2,\ R_3\}$ to reach F_o.

1) At N_e, A_1^r, A_2^r, A_1^b, and A_2^b have no knowledge about the location of F_o. Hence, they randomly select from $\{R_1,\ R_2,\ R_3\}$. Suppose that A_1^b and A_2^b choose R_2 and R_1, respectively while A_1^r and A_2^r select R_3 and R_1 respectively. As they move along their chosen paths, they deposit a certain amount of pheromone. While A_1^b and A_2^b each deposits one unit of blue color pheromone τ_b along R_2 and R_1 respectively, A_1^r and A_2^r each deposits one unit of red color pheromone τ_r along R_3 and R_1, respectively (see Fig. 10).
2) As shown in Fig. 11, since $R_1 > R_3 > R_2$, A_1^b reaches F_o before A_2^b, A_1^r, and A_2^r. To return from F_o to N_e, A_1^b discovers that $\tau_b^{R2} = 1$ and $\tau_b^{R1} = \tau_b^{R3} = 0$ (there is one unit of blue pheromone along R_2, but there is no trace of blue pheromone along R_1 and R_3).
3) Since $\tau_b^{R2} > \tau_b^{R1}$ and $\tau_b^{R2} > \tau_b^{R3}$, A_1^b is *more likely* to choose R_2. Suppose A_1^b chooses R_2. As it moves along R_2, τ_b^{R2} is increased to 2. See Fig. 12.
4) Subsequently, since $R_1 > R_3$, A_1^r reaches F_o before A_2^b and A_2^r. To return from F_o to N_e, A_1^r discovers that $\tau_r^{R3} = 1$ and $\tau_r^{R1} = \tau_r^{R2} = 0$ and $\tau_b^{R2} = 2$. Since $\tau_b^{R3} > \tau_b^{R1}$ and $\tau_b^{R3} > \tau_b^{R2}$, and $\tau_b^{R2} \neq A_1^r$ is *more likely* to select R_3 than R_1 (and has *lesser preference* for R_2 because of repulsion). Suppose A_1^r selects R_3. See Fig. 13. As it moves along R_3, τ_r^{R3} is increased to two.

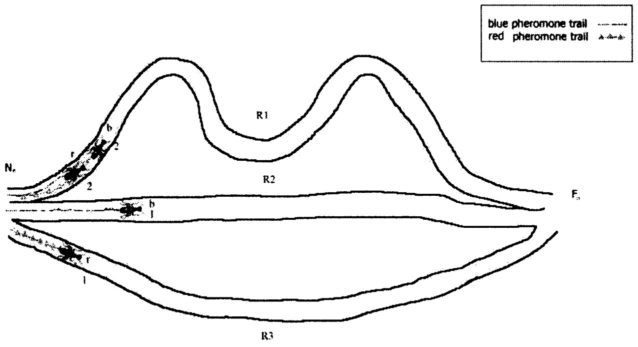


Fig. 10. All ants start to move.

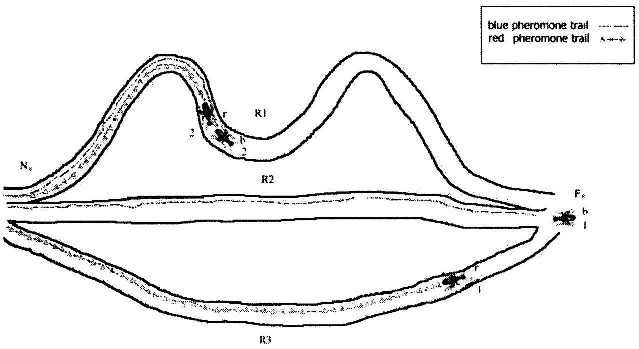


Fig. 11. A_1^b is selecting its return path while the others move toward N_2.

5) When A_2^b and A_2^r finally reach F_o and need to return to N_e, they select their return paths according to their degrees of attraction by τ_b and τ_r, respectively, and repulsion by τ_r and τ_b, respectively. Since A_2^b discovers that $\tau_b^{R2} > \tau_b^{R1}$ and $\tau_b^{R2} > \tau_b^{R3}$, and $\tau_r^{R1} \neq 0$, and $\tau_r^{R3} \neq 0$, it is *more likely* to select R_2 and has *lesser preference* for R_1 and R_3 because of repulsion. In addition, A_2^r discovers that $\tau_r^{R3} > \tau_r^{R1}$ and $\tau_r^{R3} > \tau_r^{R2}$, and $\tau_b^{R1} \neq 0$ and $\tau_b^{R2} \neq 0$ it is *more likely* to select R_3 and has *lesser preference* for R_1 and R_2 because of repulsion. Suppose A_2^b and A_2^r choose R_2 and R_3 respectively. See Fig. 14.
6) When all four ants have returned to N_e, R_2 has the heaviest concentration of τ_b, while R_3 has the heaviest concentration of τ_r. Consequently, ants from the blue colony will be *more likely* to select R_2 if they need to travel to F_o, while ants from the red colony will be *more likely* to select R_3.

Similar to single colony ACO, if an ant is at a choice point when there is no pheromone, it makes a random decision (see step 1). However, when only pheromone from its own colony is present (e.g., see step 3), there is a *higher* probability that it will choose the path with the higher concentration of its own pheromone type. In addition, due to repulsion, an ant is *less likely* to prefer paths with (higher concentration of) pheromone from other colonies (e.g., see step 5). Moreover, it is reminded that the degrees of attraction and repulsion are determined by two weighting parameters as mentioned above (and was suggested in [48].

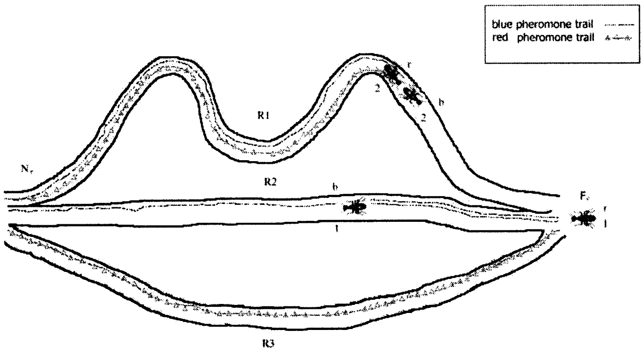


Fig. 12. A_1^b chooses N_2 again, and A_1^r is selecting its return path.

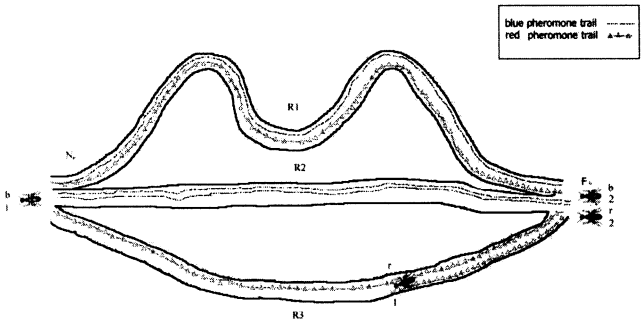


Fig. 13. A_1^r chooses R_3, and A_2^b and A_2^r are selecting their return paths.

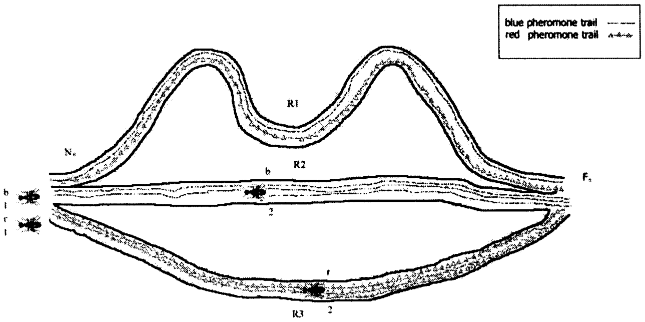


Fig. 14. A_2^r chooses R_3 and A_2^b chooses R_2.

B. Applying MACO in Load-Balancing

Adopting the problem-solving paradigm of MACO, this example illustrates the use of two sets of mobile agents (that act as routing packets) for establishing call connections in a circuit-switched network (see Fig. 15). To establish connections between gateways 1 and 3, the two groups of mobile agents construct, manipulate and consult their own routing tables. In MACO, each group of mobile agents corresponds to a colony of ants, and the routing table of each group corresponds to a pheromone table of each colony. Even though the two groups of mobile agents (MAG1 and MAG2) may have their own routing preferences, they also take into consideration the routing preferences of the other group. (While the routing preferences of ants are recorded in their pheromone tables, the routing preferences of mobile agents are stored in their routing tables). In constructing its routing table, MAG1 (respectively, MAG2) consults the routing table of MAG2 (respectively, MAG1) so as to avoid routing packets to those paths that are highly preferred by the other group. Doing so *increases the likelihood* that two

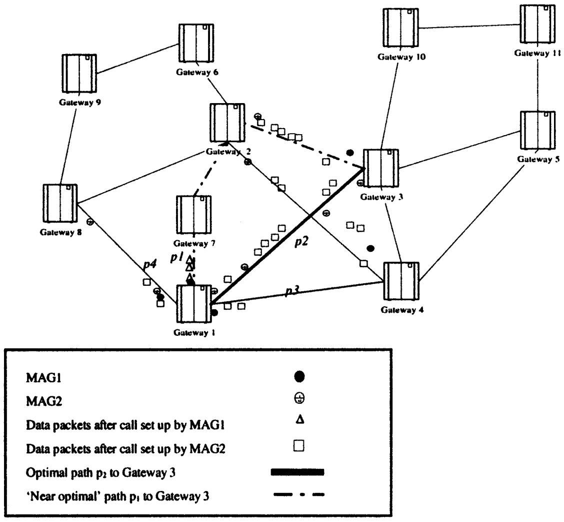
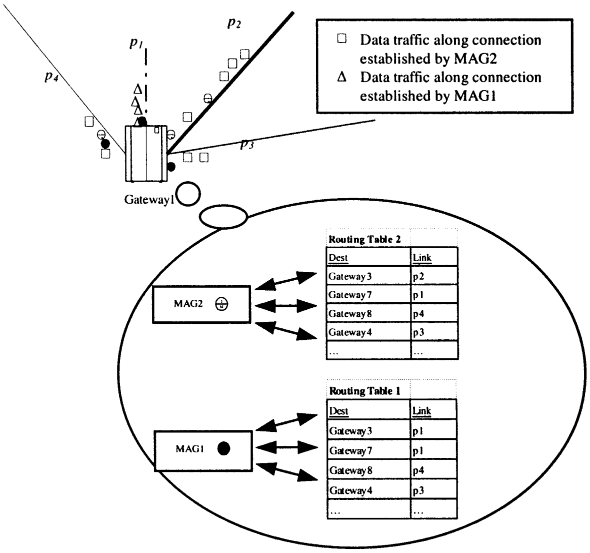

Gateway 9
Gateway 6
Gateway 10
Gateway 11
Gateway 2
Gateway 3
Gateway 5
Gateway 8
Gateway 7
Gateway 4
Gateway 1
p1
p2
p3
p4
MAG1 ●
MAG2 ⊕
Data packets after call set up by MAG1 Δ
Data packets after call set up by MAG2 □
Optimal path p2 to Gateway 3
'Near optimal' path p1 to Gateway 3

Fig. 15. Connection establishment by mobile agents.


p1
p2
p3
p4
□ Data traffic along connection established by MAG2
Δ Data traffic along connection established by MAG1
Gateway1
MAG2 ⊕
MAG1 ●

Routing Table 2

Dest	Link
Gateway3	p2
Gateway7	p1
Gateway8	p4
Gateway4	p3
...	...

Routing Table 1

Dest	Link
Gateway3	p1
Gateway7	p1
Gateway8	p4
Gateway4	p3
...	...

Fig. 16. MAG1 and MAG2 establish connections differently.

different connections between gateways 1 and 3 may be established. This increases the chance of distributing data traffic between gateways 1 and 3 between the two connections p_1 and p_2 (see Fig. 16). In Fig. 16, for the same destination Gateway3, MAG1 is *more likely* to move along p_1 whereas MAG2 is *more likely* to move along p_2. By adopting the MACO approach, it may be possible to *reduce the likelihood* that all mobile agents establish connections using *only* the optimal path. If MAG2 selects the optimal path p_2, the idea of repulsion may *increase the probability* that MAG1 will select an alternative to p_2, (e.g., p_1). The advantage of using MACO in circuit-switched routing is that it is *more likely* to establish connections through multiple paths to help balance the load but does not increase the routing overhead. An on-going work [47] implements *some* of the ideas of MACO (first proposed in [48]) in a testbed, and preliminary empirical results seem to suggest that using the same number of mobile agents (routing packets), it is *more likely* that MACO can establish connections through multiple paths while traditional ACO is *more likely* to establish connections through the optimal path. Space limitation prevents the results from being included here, but they are reported in [47]. Nevertheless, this paper does *not* suggest that MACO is a panacea. Using MACO, an additional routing table is needed at each gateway. In addition, a more complete implementation of MACO and more results are needed to verify the above claim—this is an agenda for future research.

VII. Conclusion

In this survey, an exposition of the basic problem-solving paradigm of ACO was given. The differences between ACO and traditional routing algorithms were compared along the issues of routing information, routing overhead and adaptivity. Furthermore, the issue of stagnation in ACO algorithms was discussed, and the state-of-the-art approaches for mitigating stagnation were analyzed, compared and critiqued. Survey and comparison of three major groups of research (and their ramifications) in applying ACO in routing and/or load-balancing were given. Although many of the extant approaches of applying ACO in routing and load balancing achieved favorable empirical results, these approaches (e.g., *ABC*, *AntNet*, *ASGA* and *SynthECA*) did not provide theoretical analysis on convergence. Issues in convergence analysis by Gutjahr [50] include: 1) proving that the convergence probability of ants' solutions is close to one, and 2) the speed of convergence. However, Gutjahr's analysis was for the application in ACO in the traveling salesman problem. To the best of the authors' knowledge, similar analysis in applying ACO to routing and load balancing has not been carried out.

Acknowledgment

The authors would like to express their thanks and appreciation to the anonymous reviewers for providing comments and suggestions.

References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behavior," *Nature*, vol. 406, pp. 39–42, July 2000.
[2] M. Dorigo, G. D. Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artif. Life*, vol. 5, no. 2, pp. 137–172, 1999.
[3] J. L. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels, "The self-organizing exploratory pattern of the argentine ant," *J. Insect Behav.*, vol. 3, pp. 159–168, 1990.
[4] M. Dorigo, E. Bonabeau, and G. Theraulaz, "Ant algorithms and stigmergy," *Future Gener. Comput. Syst.*, vol. 16, no. 8, pp. 851–871, 2000.
[5] M. Dorigo and G. D. Caro, "The ant colony optimization metaheuristic," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. New York: McGraw-Hill, 1999.
[6] V. Maniezzo and A. Carbonaro, "Ant colony optimization: An overview," in Metaheuristic Int. Conf., Angra dos Reis, Brazil.
[7] M. Dorigo and T. Stutzle, "The ant colony optimization metaheuristic: Algorithms, applications, and advances," in *Handbook of Metaheuristics* , F. Glover and G. Kochenberger, Eds. Norwell, MA: Kluwer.
[8] M. Dorigo, V. Maniezzo, and A. Colorni, "Positive Feedback as a Search Strategy," Dipartimento Elettronica, Politecnico Milano, Italy, Tech. Rep. 91-016, 1991.
[9] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the travelling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 53–66, 1997.

[10] I. A. Wagner, M. Linderbaum, and A. M. Bruckstein, "ANTS: Agents, networks, trees, and subgraphs," in *Future Generation Computer Systems Journal*, M. Dorigo, G. D. Di Caro, and T. Stutzel, Eds. Amsterdam, The Netherlands: North Holland, 2000, vol. 16, pp. 915–926.

[11] K. M. Sim and W. H. Sun, "A comparative study of ant-based optimization for dynamic routing," in *Proc. Conf. Active Media Technology*, Lecture Notes Computer Science, Hong Kong, Dec. 2001, pp. 153–164.

[12] T. Stuzle and H. H. Hoos, "MAX-MIN ant system," *Future Gener. Comput. Syst. J.*, vol. 16, no. 8, pp. 889–914, 2000.

[13] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, "Ants for Load Balancing in Telecommunication Networks," Hewlett Packard Lab., Bristol, U.K., Tech. Rep. HPL-96-35, 1996.

[14] G. D. Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communications networks," *J. Artif. Intell. Res.*, vol. 9, pp. 317–365, 1998.

[15] ——, "Two ant colony algorithms for best-effort routing in datagram networks," in *Proc. 10th IASTED Int. Conf. Parallel Distributed Computing Systems*, 1998, pp. 541–546.

[16] B. Baran and R. Sosa, "A new approach for AntNet routing," presented at the *Proc. 9th Int. Conf. Computer Communications Networks*, Las Vegas, NV, 2000.

[17] T. White, B. Pagurek, and F. Oppacher, "ASGA: Improving the ant system by integration with genetic algorithms," in *Proc. 3rd Genetic Programming Conf.*, July 1998, pp. 610–617.

[18] T. White and B. Pagurek, "Toward multi-swarm problem solving in networks," in *Proc. 3rd Int. Conf. Multi-Agent Systems*, July 1998, pp. 333–340.

[19] T. White, "SynthECA: A Society of Synthetic Chemical Agents," Ph.D. dissertation, Carleton University, Northfield, MN, 2000.

[20] J. Keogh, *The Essential Guide to Networking*. Englewood Cliffs, NJ: Prentice-Hall, 2001, p. 157.

[21] E. Bonabeau, F. Hénaux, S. Guérin, D. Snyer, P. Kuntz, and G. Théraulaz, "Routing in telecommunications networks with ant-like agents," in *Proc. Intelligent Agents Telecommunications Applications*, Berlin, Germany, 1998.

[22] R. Beckers, J. L. Deneubourg, and S. Goss, "Trails and U-turns in the selection of a path by the ant lasius niger," *J. Theoret. Biol.*, vol. 159, pp. 397–415, 1992.

[23] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, "Ant-based load balancing in telecommunications networks," *Adapt. Behav.*, vol. 5, no. 2, 1996.

[24] B. Schatz, S. Chameron, G. Beugnon, and T. S. Collett, "The use of path integration to guide route learning in ants," *Nature*, vol. 399, no. 6738, pp. 769–777, June 1999.

[25] D. Subramanian, P. Druschel, and J. Chen, "Ants and reinforcement learning: A case study in routing in dynamic networks," in *Proc. Int. Joint Conf. Artificial Intelligence*, Palo Alto, CA, 1997, IJCAI-97, pp. 832–838.

[26] G. D. Caro and M. Dorigo, "AntNet: A Mobile Agents Approach to Adaptive Routing," Univ. Libre de Bruxelles, Brussels, Belgium, Tech. Rep. IRIDIA/97-12, 1997.

[27] B. Bullnheimer, R. F. Hartl, and C. Strauss, "A New Rank-Based Version of the Ant System: A Computational Study," Institute Management Science, Univ. Vienna, Vienna, Austria, Tech. Rep. POM-03/97, 1997.

[28] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, "Ant-Based Load Balancing in Telecommunications Networks," Hewlett Packard Lab., Bristol, U.K., Tech. Rep. HPL-96-76, 1996.

[29] R. Schoonderwoerd, O. Holland, and J. Bruten, "Ant-like agents for load balancing in telecommunications networks," in *Proc. Agents*, Marina del Rey, CA, pp. 209–216.

[30] R. Schoonderwoerd and O. Holland, "Minimal agents for communications networks routing: The social insect paradigm," in *Software Agents for Future Communication Systems*, A. L. G. Hayzeldean and J. Bingham, Eds. New York: Springer-Verlag, 1999.

[31] S. Appleby and S. Steward, "Mobile software agents for control in telecommunication networks," *BT Technol. J.*, vol. 12, no. 2, 1994.

[32] S. Guérin, "Optimization Multi-Agents en Environment Dynamique: Application au Routage Dans les Réseaux de Telecommunications," DEA, Univ. Rennes I, Ecole Nat. Supér. Télécommun. Bretagne, , Bretagne, France, 1997.

[33] M. Heusse, D. Snyers, S. Guérin, and P. Kuntz, "Adaptive Agent-Driven Routing and Load Balancing in Communication Networks," ENST Bretagne, Brest, France, Tech. Rep. RR-98 001-IASC, 1998.

[34] G. D. Caro and M. Dorigo, "Mobile agents for adaptive routing," in *Proc. 31st Hawaii Int. Conf. Systems Sciences*, Kohala Coast, HI, Jan. 1998, pp. 74–83.

[35] ——, "Ant colonies for adaptive routing in packet-switched communications networks," in *Proc. 5th Int. Conf. Parallel Problem Solving from Nature*, Amsterdam, The Netherlands, Sept. 27–30, 1998.

[36] ——, "An adaptive multi-agent routing algorithm inspired by ants behavior," in *Proc. 5th Annual Australasian Conf. Parallel Real-Time Systems*, 1998, pp. 261–272.

[37] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[38] T. White, "Routing With Swarm Intelligence," Syst. Comput. Eng. Dept., Carleton Univ., Northfield, MN, Tech. Rep. SCE-97-15, 1997.

[39] T. White, F. Oppacher, and B. Pagurek, "Connection management using adaptive agents," in *Proc. Int. Conf. Parallel and Distributed Processing Techniques Applications*, July 12–16, 1998, pp. 802–809.

[40] T. White, B. Pagurek, and D. Deugo, "Collective intelligence and priority routing in networks," in *Proc. 15th Int. Conf. Industrial Engineering Application Artificial Intelligence Expert Systems*, Cairns, Australia, June 17–20, 2002.

[41] ——, "Biologically-inspired agents for priority routing in networks," in *Proc. 15th Int. FLAIRS Conf.*, Pensacola, FL, May 14–16, 2002.

[42] T. White and B. Pagurek, "Emergent behavior and mobile agnets," in *Proc. Workshop Mobile Agents Coordination Cooperation Autonomous Agents*, Seattle, WA, May 1–5, 1999.

[43] ——, "Distributed fault location in networks using learning mobile agents," in *Proc. 2nd Pacific Rim Int. Workshop Multi-Agents*, H. Nakashima and C. Zhang, Eds., Berlin, Germany, 1999.

[44] ——, "Distributed fault location in networks using learning mobile agents," in *Lecture Notes in Computer Science*. New York: Springer-Verlag, 1999, vol. 1733.

[45] ——, "Application oriented routing with biologically-inspired agents," in Proc. Genetic Evolutionary Computation Conf., Orlando, FL, July 13–17, 1999.

[46] K. M. Sim and W. H. Sun, "Multiple ant-colony optimization for network routing," in *Proc. 1st Int. Symp. Cyberworld*, Tokyo, Japan, November 2002, pp. 277–281.

[47] ——, "A multiple ant colony optimization approach for load balancing," in *Proc. 4th Int. Conf. Intelligent Data Engineering Automated Learning*, Hong Kong, 2003.

[48] N. Varela and M. C. Sinclair, "Ant colony optimization for virtual-wavelength-path routing and wavelength allocation," in *Proc. Congress Evolutionary Computation*, Washington, DC, July 1999, pp. 1809–1816.

[49] S. Fenet and S. Hassas, "An ant system for multiple criteria balancing," in Proc. 1st Int. Workshop Ants Systems, , Brussels, Belgium, Sept 1998.

[50] W. J. Gutjahr, "A graph-based ant system and its convergence," *Future Gen. Comput. Syst.*, vol. 16, pp. 873–888, 2000.

Kwang Mong Sim received the B.Sc. (Hon., summa cum laude) in computer science from the University of Ottawa, Ottawa, ON, Canada, in 1990, and the M.Sc. and Ph.D. degrees in computer science from the University of Calgary, Calgary, AB, Canada, in 1991 and 1995, respectively.

He is currently an Associate Professor with the Department of Information Engineering, the Chinese University of Hong Kong. He is Guest Editor for three special journal issues, including IEEE Transactions, in the areas of negotiation agents and grid computing. He served as Session Chairman and PC Member in many agent conferences and supervised close to 20 graduate students in these areas.

Weng Hong Sun (M'02) received the B.Sc. degree in computer science from the University of Manitoba, Winnipeg, MB, Canada, and the M.B.A. degree from the University of Macau, Taipa, Macao, in 1995 and 2000, respectively. He is currently pursuing the M.Phil. degree in information engineering.

He is currently a Lecturer at the faculty of business information at the University of Macau.