Routing in Telecommunications Networks with Ant-Like Agents

Eric Bonabeau[1], Florian Henaux[2], Sylvain Guérin[3], Dominique Snyers[3], Pascale Kuntz[3], Guy Theraulaz[4]

[1] Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA
bonabeau@santafe.edu

[2] Ecole Nationale Supérieure des Télécommunications de Paris, 46 rue Barrault, 75634 Paris Cédex 13, France

[3] Ecole Nationale Supérieure des Télécommunications de Bretagne, BP 832, 29285 Brest Cédex, France

[4] CNRS - UMR 5550, Laboratoire d'éthologie et psychologie animale, Université Paul Sabatier, 118 route de Narbonne, 31062 Toulouse, France

Abstract. . A simple mechanism is presented, based on ant-like agents, for routing and load balancing in telecommunications networks, following the initial works of Appleby and Stewart [1] and Schoonderwoerd et al. [32,33]. In the present work, agents are very similar to those proposed by Schoonderwoerd et al. [32,33], but are supplemented with the ability to perform more computations at switching nodes, which significantly improves the network's relaxation and its response to perturbations.

1 Introduction

1.1 Routing in Telecommunications Networks

Routing is a mechanism that allows calls to be transmitted from a source to a destination through a sequence of intermediate switching stations or nodes, because not all points are directly connected: the cost of completely connecting a network becomes prohibitive for more than a few nodes. Routing selects routes that meet the objectives and constraints set by the user traffic and the network, and therefore determines which network resources are traversed by which user traffic [26,34]. Why should routing be (1) dynamic, and (2) decentralized?

S. Albayrak and F. J. Garijo (Eds.): IATA'98, LNAI 1437, pp. 60–72

(1) The pathway of a message must be as short as possible. One solution is therefore to design fixed routing tables such that any two nodes in the network are connected through the shortest possible path. Designing such routing tables is a simple optimization problem, which has to be solved when the network topology has been defined. But traffic conditions are constantly changing, and also the structure of the network itself may fluctuate (switching stations or connections can fail). The problems of telecommunications networks is to minimize the number of call failures in any condition. Because there are usually many possible pathways for one message to go from a given node to another node, it is possible in principle to make routing algorithms adaptive enough to overcome local congestions: calls can be rerouted to nodes that are less congested, or have spare capacity. If there is a sudden burst of activity, or if one node becomes the destination or the emitter of a large number of calls, rerouting becomes crucial. Static routing, whereby routing remains fixed independent of the current states of the network and user traffic, is therefore almost never implemented: most routing schemes respond in some way to changes in network or user traffic state. But there exists a wide spectrum of dynamic routing systems, which vary dramatically in their responsiveness, speed of response and in the types of changes they respond to [34]: some routing systems can be seen as "quasistatic", because routing is modified only in response to exceptional events (link or switch failure), and/or on a long time scale, while other routing systems are highly dynamic and autonomously update traffic routing in real time in response to perceived changes in user and network state. Tightly coupled with dynamic routing is load balancing, which is the construction of call-routing schemes that successfully distribute the changing load over the system and minimize lost calls. Load balancing makes it possible to relieve actual or potential local congestion by routing calls via parts of the network that have spare capacity. Dynamic routing and load balancing require more computational resources than static or quasistatic routing. It relies on active participation of by entities within the network to measure user traffic, network state and performance, and to compute routes.

(2) Most routing algorithms today are centralized, with routing tables at switching stations being updated by a central controller at regular intervals. But:

• The controller needs current knowledge about the entire system, necessitating communications links from every part of the system to the controller.

• Central control mechanisms scale badly.

• Failure of the controller leads to failure of the whole system.

• Telecommunications networks are distributed, extended, dynamic, highly unpredictable systems, and central control may simply not be appropriate.

• A routing system's responsiveness to state changes depends upon the load on the central controller and the distance between the central controller and the portion of the network requiring adaptation.

• Centrally controlled systems might have to be owned by a single authority.

• By contrast, with a decentralized implementation of the routing function, multiple entities function independently and exchange information, providing fault tolerance. Moreover, the routing computational load is spread among the multiple entities.

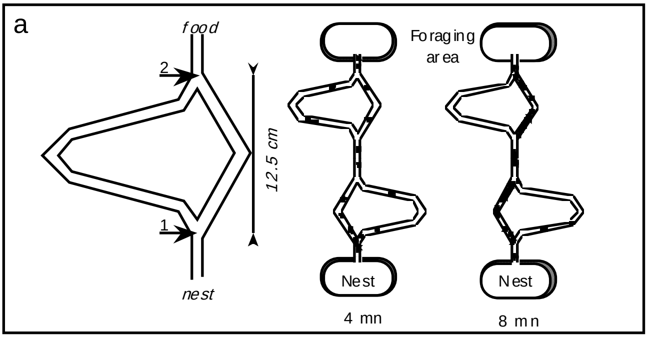
1.2 Agent-Based Routing and Network Control

The idea that ant-like agents, or "mobile software agents", could be used for network control in telecommunications has been introduced by Appleby and Steward [1,22], in a paper that poses the problems clearly but remains vague as regards actual implementation. Schoonderwoerd et al. [32,33] have proposed an interesting version of Appleby and Steward's [1] work, where they use simple agents that modify the routing tables of every node in the network. Their work, being described in more detail, is somewhat easier to reproduce, and contains a set of nice ideas. Routing and load balancing mechanisms based on their algorithm may conceivably be used in some networks in the near future, if it can be proved, one way or another, that *any* network condition can be satisfactorily dealt with. New approaches to routing require intensive testing.

1.3 Ant-Like Agents

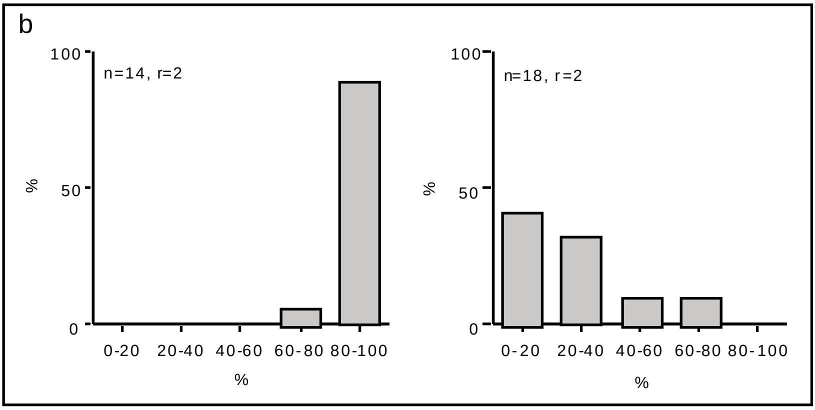
Schoonderwoerd et al.'s [32,33] work relies on the ability of social insects to solve problems, sometimes difficult problems, in a distributed way, without any central control, on the basis on local information. Social insects also often exhibit flexibility (they can respond to internal perturbations and external challenges) and robustness (failure of one or several individuals usually does not jeopardize a colony's functioning). Given these properties and the impressive ecological success of social insects [38], it does not seem unreasonable to try to transfer current knowledge about how insect societies function into the context of engineering and distributed artificial intelligence. This approach is similar to other approaches consisting in imitating the way "nature" (that is, physical or biological systems) solves problems [7,19]. Another possible and apparently very promising pathway is to use economic or financial metaphors to solve problems [24,25].

Figure 1a. Experimental setup and drawings of the selection of the short branches by a colony of *L. humile* respectively 4 and 8 minutes after the bridge was placed.



Recent research in ethology suggests that self-organization is a major component of a wide range of collective phenomena in social insects [15,5]. Theories of self-organization, originally developed in the context of physics and chemistry to describe the emergence of macroscopic patterns out of processes and interactions defined at the microscopic level, can be extended to social insects to show that complex collective behavior may emerge from interactions among individuals that exhibit simple behavior. A good illustration of the self-organized, distributed problem-solving ability of social insects is the binary bridge experiment [2,15]: in experiments with the ant *Linepithema humile*, a food source is separated from the nest by a bridge with two branches A and B, with branch B being r times longer than branch A (Fig. 1a). It is observed that in most experiments, the short branch is selected by the colony if r is sufficiently large (r=2 in Fig. 1b). This is because these ants have a trail-laying/trail-following behavior: individual ants lay a chemical substance, a pheromone, which attracts other ants. The first ants returning to the nest from the food source take the shorter path twice (from the nest to the source and back), and therefore influence outgoing ants towards the short branch which is initially more strongly marked. If, however, the short branch is presented to the colony after the long branch, the short path will not be selected because the long branch is already marked with pheromone: the colony does not exhibit flexibility (Fig. 1b). This problem can be overcome in an artificial system, by introducing the pheromone's lifetime: because pheromone diffuses and evaporates, it is more difficult to maintain a stable pheromone trail on a long path than on a short path. In this way, the short branch can be selected even if presented after the long branch. In the real world, although pheromone concentrations do decay, the lifetime of pheromones is usually so large that it cannot allow such a flexibility: this is a case where there is a clear divergence from biological reality, but solving problems and understanding nature are two activities which have their own criteria of success.

Figure 1b. Distribution of the percentage of ants that selected the shorter branch over all experiments (14 and 18 experiments, respectively). The longer branch is r times longer than the short branch. The second graph (n=18, r=2) corresponds to an experiment in which the short branch is presented to the colony 30 minutes after the long branch: the short branch is not selected, and the colony remains trapped on the long branch. From Ref. 15.



Taking advantage of this ant-based optimizing principle, Colorni et al. [10,11,12], Dorigo et al. [17], Dorigo and Gambardella [18,20] and Gambardella et al. [21] have proposed an ingenious optimization method, the Ant System (or the Ant Colony System, which is more efficient), which they applied to classical optimization problems, such as the traveling salesman problem [30], the quadratic assignment problem [28] or the job-shop scheduling problem, with reasonable success: this method, as a general heuristic, can be compared to simulated annealing [27]. Others have followed the path, and have either extended the original method [8,35] or applied ant-based optimization to the vehicle routing problem [9], the graph coloring problem [13] and the search of continuous spaces [3,39]. Schoonderwoerd et al.'s [32,33] application to routing and load balancing in telecommunications networks is derived from the same metaphor: however, the formulation of their algorithm, the dynamic properties of the problem and the highly distributed nature of the underlying system, make this application original and of special interest. In particular, it is, until now, the only example where the advantage of using the decentralized functioning of social insect colonies is (and could be even more) fully exploited.

Other applications based on the functioning of social insect societies include data analysis [31] and graph clustering [29] inspired by cemetery organization in ants [16], adaptive task allocation [6] inspired by the division of labor in social insects [4,36], and self-assembly inspired by collective building in wasps [37].

1.4 Outline of the Paper

The rest of the paper is dedicated to describing and extending the ant-like-agent-based mechanism introduced by Schoonderwoerd et al. [32,33] to include a form a dynamic programming which makes the agent "smarter".

2 Algorithms

2.1 Original Algorithm [32,33]

If there are n nodes in the network, a node N_i with k(i) neighbors is characterized by a routing table $R_i = [r^i_{l,m}]_{n-1,k(i)}$ that has n-1 rows and k columns: each row corresponds to a destination node and each column to the next node. $r^i_{l,m}$ gives the probability that a given message, the destination of which is node N_l, be routed from node N_i to node N_m. Agents update routing tables of nodes viewing their node of origin as a destination node: links being bidirectional, agents that have a certain "knowledge" about some portion of the network (where they come from) modify routing tables of nodes that influence the routing of messages and agents that have this portion of the network as destination. This avoids agents going back all the way to their node of origin to update all intermediate routing tables, and therefore decreases in pinciple the network's agent load.

Agents can be launched from any node in the network at any time. Destination is selected randomly among all other nodes in the network. The probability of launching an agent per unit time must be tuned to maximize the performance of the system. It appears that too few agents are not enough to reach good solutions, whereas too large a number of agents degrades the performance of the system by adding noise (a more appropriate way of launching agents would be to generate an increasing number of agents as the network becomes more and more congested). The notion of performance of the system here is simply the number of calls that fail to reach their destinations. Agents go from their source node to their destination node by moving from node to node. The next node an agent will move to is selected according to the routing table of its current node. Each visited node's routing table is updated: more precisely, an agent modifies the row corresponding to its source node, which is viewed as a destination node. When the agent reaches its destination node, it updates the local routing table and is deleted from the network. How are routing tables updated? Let N_s be the source node of an agent, N_m the node it just comes from, and N_i its current node at time t. The entry $r^i_{s,m}(t)$ is reinforced while other entries $r^i_{s,l}(t)$ in the same row decay:

$$r^i_{s,m}(t+1) = \frac{r^i_{s,m}(t) + \delta r}{1+\delta r}, \quad (1)$$

$$r^i_{s,l}(t+1) = \frac{r^i_{s,l}(t)}{1+\delta r}, \quad (2)$$

where δr is a reinforcement parameter, that depends on the agent's characteristics. Note that this updating procedure conserves the normalization of $r^i_{s,l}$,

$$\sum_l r^i_{s,l} = 1 \quad (3)$$

if they are initially normalized, so that $r^i_{s,l}$ can always be considered as probabilities, all of which always remain strictly positive. $r^i_{s,m}(t)$ is comparatively more reinforced when it is small (that is, when node m is not on the preferred route to node s) than when it is large (that is, when node m is on the preferred route to node s). This is an interesting feature, as it should allow new routes to be discovered quickly when the preferred route gets congested because established routing solutions become more easily unstable: there is an exploration/exploitation tradeoff to be found, for too much instability might not always be desirable (one way of getting around this problem would be, once again, to increase the number of agents only when the network is congested, because more agents destabilize solutions that use congested routes).

The influence δr of a given agent must depend on how well this agent is performing. Aging can be used to modulate δr: if an agent has been waiting a long time along its route to its destination node, it means that the nodes it has visited and links it has used are congested, so that δr should decrease with the agent's age (measured in time units spent in the network). Aging should in principle be relative to the length (expressed in time units) of the optimal path from an agent's source node to its destination. Schoonderwoerd et al. [32,33] choose to use absolute age T, measured in time units spent in the network, and propose

$$\delta r = \frac{a}{T} + b, \quad (4)$$

where a and b are parameters. Aging is also used in a complementary way: age is manipulated by delaying agents at congested nodes. Delays result in two effects: (1) the flow of ants from a congested node to its neighbors is temporarily reduced, so that entries of the routing tables of neighbors that lead to the congested cannot be reinforced, and (2) the age of delayed agents increases by definition, so that delayed ants have less influence on the routing tables of the nodes they reach, if δr decreases with age. Schoonderwoerd et al. [32,33] suggest that the delay D imposed on an agent reaching a node with spare capacity

S (that is, the percentage of slots left in the node for new messages) should be given by

$$D = c\, e^{-S/S_c}\,, \tag{5}$$

where c is a parameter and S_c is a characteristic spare capacity (expressed in percentage of the node's capacity), so that a node is considered congested if $S >> S_c$.

Finally, one needs to avoid freezing routes in situations that remain static for a long time and then suddenly change. Finding new routes is facilitated by increased reinforcement of small entries, but this may be insufficient. Schoonderwoerd et al. [32,33] suggest the addition of a tunable "noise" or "exploration" factor f $(0<f<1)$. At every node, an agent chooses a purely random path with probability f and chooses its path according to the node's routing table with probability (1-f). Noise allows to maintain information about apparently useless routes to give a head start when the preferred route is blocked. It also allows to rediscover quickly a better route that appears owing to the release from congestion of a node.

2.2 "Smarter" Agents

At each time step, an agent as described by Schoonderwoerd et al. [32,33] only updates the row that corresponds to its source node (viewed as a destination node). An interesting recent addition by Guérin [23] introduces updating of all rows corresponding to all the intermediate nodes visited by the agent, in a way reminiscent of dynamic progamming: reinforcement of an entry associated with a given node is discounted by a factor that depends on the agent's age relative to the time it visited that node. Although Guérin's [23] method relies on agents that perform round trips from their source node to their destination node and back, it can be readily applied to the case of one-way agents. In order to distinguish these more complex or "smarter" agents from those of Schoonderwoerd et al. [32,33], we call them "smart" agents. Let N_i be the ith node visited by an agent on its way to its destination. The agent updates the rows of all intermediate nodes in N_i's routing table. Updating of the row corresponding to node N_m (m<i), the mth visited node, is performed using a relative age instead of its absolute age. Entry $r^i_{m,i-1}(t)$ is reinforced while other entries $r^i_{m,l}(t)$ (l•i) in the same row decay:

$$r^i_{m,i-1}(t+1) = \frac{r^i_{m,i-1}(t) + \delta r}{1+\delta r}\,, \tag{7}$$

$$r^i_{m,l}(t+1) = \frac{r^i_{m,l}(t)}{1+\delta r}\,, \tag{8}$$

$$\delta r = \frac{a}{T_i - T_m} + b, \tag{9}$$

where T_m is the agent's absolute age when reaching node N_m and T_i its age when reaching node N_i. The entries corresponding to nodes visited long ago are weakly updated, which is expressed in equation (9).

The ant-based algorithm based on smarter agents yields significantly better performance. Figure 2a shows the average number of call failures per 500 steps together with error bars representing the standard deviation observed over 10 trials of the simulation with identical parameters, the first 500 steps being discarded, after the network has been initialized. The parameters are identical to those of Schoonderwoerd et al. [32,33]: we use the 30 node interconnection network of British Telecom as an example; each node has a capacity of 40 calls (S=100%); during every time step, an average of one call is generated, with an average duration of 170 time steps; the probabilities that nodes be emitters or receivers are uniform in [0.01,0.07] and normalized; generation and normalization of emission and reception probabilities for all nodes defines a set of call probabilities, and a change in call probabilities means that new emission and reception probabilities have been generated (new nodes become more likely to be emitters or receivers, and others less likely). At initialization, routing tables are characterized by equiprobable routing (all neighboring nodes are equally likely to be selected as next node), and there is no message in the network. Messages are routed independently of the agents' dynamics: when a message reaches a node, it selects the largest entry in the appropriate row it its current table and is routed towards the neighboring node corresponding to this largest entry. During the first phase following initialization (from t=501 to 3000), smart agents perform significantly better than Schoonderwoerd et al.'s (1997) agents (t-test, df=18, t=3.2, $P<0.003$); during the second phase (t=3001 to 7500), when a stationary state in the dynamics of call failures has been reached, smart agents also perform significantly better (df=18, t=3.9, $P<0.001$), with a level of significance comparable to the one obtained during the first phase.

Figure 2b shows the average number of call failures per 500 steps together with error bars representing the standard deviation observed over 10 trials of the simulation with identical parameters, the first 500 steps being discarded, after a change in the call probabilities. Results similar to those obtained after network initialization are observed: during both phases, smart agents perform significantly better than simple agents (phase 1: df=18, t=4.1, $P<0.001$; phase 2: df=18, t=2.3, $P<0.025$), but the level of significance is much lower during the second phase, indicating that smart agents are particularly useful when network conditions are changing.

Figure 2a. Adaptation (500 first steps discarded). Phase 1: from t=501 to 3000, Phase 2: from t=3001 to 7500. a=0.08, b=0.005, c=80, S_c=13.3%, f=0.05, node capacity=40 calls.

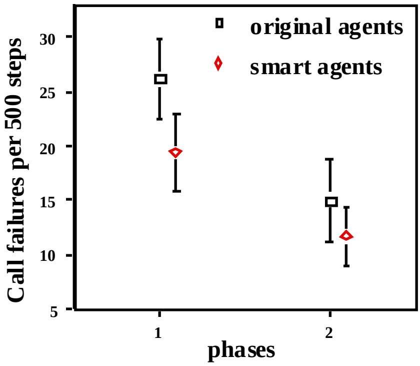
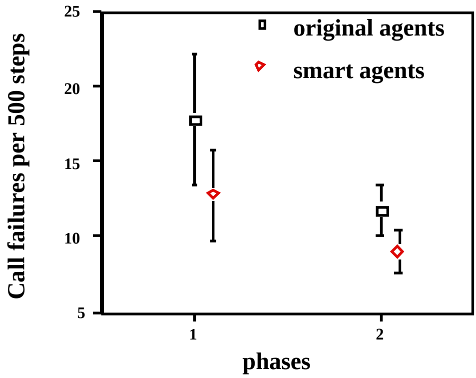


Figure 2b. Change in call probabilities (500 first steps discarded). Phase 1: from t=501 to 3000, Phase 2: from t=3001 to 7500. a=0.08, b=0.005, c=80, S_c=13.3%, f=0.05, node capacity=40 calls.



3 Limitations and Future Work

We have supplemented the agents introduced by Schoonderwoerd et al. [32,33] with a simple extension initially suggested by Guérin [23] with more complex agents. Analysis of call failures indicates that this addition improves significantly the performance of the routing scheme, especially when network traffic is subject to variations. But before this method can actually be implemented in real communications networks, some limitations have to be overcome. First, the model network used in this paper is obviously an oversimplification of reality: the method has to be tested on more realistic, and therefore more complex, network models. But this points to the problem of analyzing the routing's behavior. Routing algorithms are generally difficult to analyze mathematically, especially when the underlying network is complex and/or not fully connected: for example, the properties of the dynamic alternative routing method [26] could be obtained analytically only for fully connected networks. It is crucial for "self-organizing" algorithms, such as the one presented in this paper, where control by humans can only be limited, to be able to prove that they are not going to fall apart in some, possibly pathological but still potential, specific configuration. For example, it would be good to be sure that messages cannot become trapped in infinite cycles without ever reaching their destination. One also needs to have a clear understanding of the limits and constraints of communications networks: for example, if there are sufficient computational power and spare capacity in the network to launch a large number of complex agents without affecting traffic, why bother to design simple agents?

Acknowledgments

E. B. is supported by the Interval Research Fellowship at the Santa Fe Institute. E. B., F. H. and G. T acknowledge support from the GIS (Groupe d'Intérêt Scientifique) Sciences de la Cognition.

4 References

1. Appleby, S. & Steward, S. (1994). Mobile software agents for control in telecommunications networks. *British Telecom Technol. Journal* **12**, 104-113.
2. Beckers, R., Deneubourg, J.-L., Goss, S. and Pasteels, J.-M. (1990). Collective decision making through food recruitment. *Ins. Soc.*, **37**, 258-267.
3. Bilchev, G. & Parmee, I. C. (1995). The ant colony metaphor for searching continuous design spaces. In: *Lecture Notes in Computer Science* (Fogarty, Y., ed.) 993, 25-39, Springer-Verlag.
4. Bonabeau, E., Theraulaz, G. & Deneubourg, J.-L. (1996). Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies. *Proc. Roy. Soc. London B* **263**, 1565-1569.
5. Bonabeau, E., Theraulaz, G., Deneubourg, J.-L., Aron, S. & Camazine, S. (1997a). Self-organization in social insects. *Trends in Ecol. Evol.* **12**, 188-193.
6. Bonabeau, E., Sobkowski, A., Theraulaz, G., Deneubourg, J.-L. (1997b). Adaptive task allocation inspired by a model of division of labor in social insects. In *Bio-Computation and Emergent Computing*, eds. D. Lundh, B. Olsson & A. Narayanan. Singapore: World Scientific.

7. Bounds, D.G. (1987). New optimization methods from physics and biology. *Nature* **329**, 215-219.
8. Bullnheimer, B., Hartl, R. F. & Strauss, C. (1997a). A new rank based version of the ant system: a computational study. Working paper #1, SFB Adaptive Information Systems and Modelling in Economics and Management Science, Vienna.
9. Bullnheimer, B., Hartl, R. F. & Strauss, C. (1997b). Applying the ant system to the vehicle routing problem. *2nd Intnl Conf. on Metaheuristics* (MIC'97).
10. Colorni, A., Dorigo, M. & Maniezzo, V. (1991). Distributed optimization by ant colonies. *Proc. First Europ. Conf. on Artificial Life* (Varela, F. & Bourgine, P., eds), pp. 134-142, MIT Press.
11. Colorni, A., Dorigo, M. & Maniezzo, V. (1992). An investigation of some properties of an ant algorithm. *Proc. of 1992 Parallel Problem Solving from Nature Conference* (Männer, R. & Manderick, B., eds), pp. 509-520, Elsevier Publishing.
12. Colorni, A., Dorigo, M., Maniezzo, V. & Trubian, M. (1993). Ant system for job-shop scheduling. *Belg. J. Oper. Res., Stat. and Comput. Sci.* **34**, 39-53.
13. Costa, D. & Hertz, A. (1997). Ants can colour graphs. *J. Op. Res. Soc.* **48**, 295-305.
14. Croes, G. A. (1958). A method for solving traveling salesman problems. *Oper. Res.* **6**, 791-812.
15. Deneubourg, J.-L. & Goss, S. (1989). Collective patterns and decision making. *Ethol. Ecol. & Evol.* **1**, 295-311.
16. Deneubourg, J.-L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C. and Chretien, L. (1991). The dynamics of collective sorting: Robot-like ant and ant-like robot. In: *Simulation of Adaptive Behavior: From Animals to Animats* (Meyer, J.A. and Wilson, S.W., eds.) pp. 356-365. Cambridge, MA: The MIT Press/Bradford Books.
17. Dorigo, M., Maniezzo, V. & Colorni, A. (1996). The Ant System: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. B* **26**, 1-13.
18. Dorigo, M. & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comp.* **1**, 53-66.
19. Farmer, J. D., Packard, N. H. & Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D* **22**, 187-204.
20. Gambardella, L. M. & Dorigo, M. (1995). Ant-Q: a reinforcement learning approach to the traveling salesman problem. In: *Proc. ML-95, 12th Intnl. Conf. Machine Learning,* pp. 252-260 (Morgan Kaufmann, Palo Alto, CA).
21. Gambardella, L. M., Taillard, E. D. & Dorigo, M. (1997). Ant colonies for the QAP. Technical Report IDSIA-4-97.
22. Geake, E. (1994). How simple ants can sort out BT's complex nets. *New Scientist* **141**, 20-20.
23. Guérin, S. (1997). Optimisation multi-agents en environnement dynamique: application au routage dans les réseaux de télécommunications. DEA Dissertation, University of Rennes I and Ecole Nationale Supérieure des Télécommunications de Bretagne.
24. Huberman, B. A., Lukose, R. M. & Hogg, T. (1997). An economics approach to hard computational problems. *Science* **275**, 51-54.
25. Kephart, J. O., Hogg, T. & Huberman, B. A. (1989). Dynamics of computational ecosystems. *Phys. Rev. A* **40**, 404-421.
26. Kelly, F. P. (1995). The Clifford Paterson Lecture, 1995. Modelling communication networks, present and future. *Phil. Trans. R. Soc. London A* **354**, 437-463.
27. Kirkpatrick, S., Gelatt, C. & Vecchi, M. (1983). Optimization by simulated annealing. *Science* **220**, 671-680.

28. Koopmans, T. C. & Beckman, M. J. (1957). Assignment problems and the location of economic activities. *Econometrica* **25**, 53-76.
29. Kuntz, P., Layzell, P. & Snyers, D. (1997). A colony of ant-like agents for partitioning in VLSI technology. In: *Proc. of 4th European Conference on Artificial Life* (Husbands, P. & Harvey, I., eds), pp. 417-424, MIT Press, Cambridge, MA.
30. Lawler, E. L., Lenstra, J. K., Rinnooy-Kan, A. H. G. & Shmoys, D. B. (eds) (1985). *The travelling salesman problem*. Wiley.
31. Lumer, E. & Faieta, B. (1994). Diversity and adaptation in populations of clustering ants. *Proc. of Third Intl. Conf. on Simulation of Adaptive Behavior*, pp. 499-508, MIT Press.
32. Schoonderwoerd, R. (1996). Collective intelligence for network control. Engineer thesis, Delft University of Technology, The Netherlands.
33. Schoonderwoerd, R., Holland, O., Bruten, J. & Rothkrantz, L. (1997). Ant-based load balancing in telecommunications networks. *Adapt. Behav.* **5**, 169-207.
34. Steenstrup, M. (1995). *Routing in communications networks*. Englewood Cliffs, NJ: Prentice Hall.
35. Stützle, T. & Hoos, H. (1997). The MAX-MIN ant system and local search for the traveling salesman problem. *Proc. IEEE Intnl. Conf. Evolutionary Computation* (ICEC'97), 309-314.
36. Theraulaz G., Goss S., Gervet J. & Deneubourg J.-L. (1991). Task differentiation in *Polistes* wasp colonies: a model for self-organizing groups of robots. In: *From Animals to Animats*, Proc. of the 1st Intnl. Conf. on Simulation of Adaptive Behavior (Meyer, J.A. & Wilson, S.W., eds), 346-355, MIT Press.
37. Theraulaz, G. & Bonabeau, E. (1995). Coordination in distributed building. *Science* **269**, 686-688.
38. Wilson, E.O. (1971). *The Insect Societies.* Cambridge, MA: Harvard University Press.
39. Wodrich, M. (1996). Ant colony optimization. BSc Thesis, Dept. of Electrical and Electronic Engineering, University of Cape Town, South-Africa.