Swarm Intelligence manuscript No.
(will be inserted by the editor)

Principles and applications of swarm intelligence for adaptive routing in telecommunications networks

Frederick Ducatelle, Gianni A. Di Caro, Luca M. Gambardella

"Dalle Molle" Institute for Artificial Intelligence Studies (IDSIA)
Galleria 2, 6928 Manno, Switzerland
e-mail: {frederick,gianni,luca}@idsia.ch

The date of receipt and acceptance will be inserted by the editor

Abstract In the past few years there has been a lot of research on the application of swarm intelligence to the problem of adaptive routing in telecommunications networks. A large number of algorithms have been proposed for different types of networks, including wired networks and wireless ad hoc networks. In this paper we give an overview of this research area. We address both the principles underlying the research and the practical applications that have been proposed. We start by giving a detailed description of the challenges in this problem domain, and we investigate how swarm intelligence can be used to address them. We identify typical building blocks of swarm intelligence systems and we show how they are used to solve routing problems. Then, we present Ant Colony Routing, a general framework in which most swarm intelligence routing algorithms can be placed. After that, we give an extensive overview of existing algorithms, discussing for each of them their contributions and their relative place in this research area. We conclude with an overview of future research directions that we consider important for the further development of this field.

1 Introduction

Swarm intelligence (SI) deals with collective behaviors that result from the local interactions of individual components with each other and with their environment [10]. On the one hand, this includes the study of collective behaviors in nature, such as nest building, foraging, and item sorting in insect societies, and swarming, flocking, herding, and schooling behaviors in vertebrates. On the other hand, from an engineering point of view, it refers to the bottom-up design of distributed systems that display forms of useful and/or interesting behavior at the global level as a result of the actions of a number of units interacting with one another and with their environment at the local level. In this paper, we refer to the latter also as *SI design*.

In recent years, SI design has been applied to a wide variety of problems in combinatorial and continuous optimization, telecommunications, robotics, etc., often with excellent results (e.g., see [10, 28, 35, 37, 39, 41] for extensive literature reviews). Two of the most popular and successful examples of the SI approach are Ant Colony Optimization (ACO) [35–37] and Particle Swarm Optimization (PSO) [59–61]. ACO takes inspiration from the pheromone-mediated ability of ant colonies to find shortest paths between their nest and sources of food [35, 47] to define a metaheuristic for combinatorial optimization based on the use of ant-like agents and stigmergic communication of artificial pheromone information. PSO translates the flocking behavior of birds into a framework based on information-sharing particle-like agents to find extremal points in optimization problems.

In this paper, we focus on the application of SI design to one particular class of optimization problems, namely *adaptive routing in telecommunications networks*. Routing is the task of directing flows or units of data from their source to their destination while optimizing one or more criteria. Examples of optimization criteria are the average or maximum delay experienced by data packets or the variability in delay. An adaptive routing algorithm is one that modifies its routing solution online, in order to account for changes in the network, such as variations in the data traffic load or in the network topology. The typical structure and functioning of telecommunications networks – in which the global routing strategy is the collective result of decentralized decisions made by individual nodes based on local observations – maps surprisingly well to the typical distributed approach advocated in the SI paradigm. Consequently, a lot of successful adaptive routing algorithms have been developed based on SI ideas. Our aim here is to discuss the relationships between SI design and the design of network routing algorithms, and to provide an overview of the theory and practice in this field. Since the large majority of existing SI routing algorithms refers to the ACO framework, we mainly focus on this class of implementations of the SI paradigm. Nevertheless, we also describe other SI approaches, such as, for example, bee-inspired algorithms [43, 44].

The rest of this paper is organized as follows. First, in Section 2, we explain the problem of routing in telecommunications networks. In particular, we discuss typical challenges that are present in different types of networks, and describe traditional approaches to routing. Next, in Section 3, we hold a high level discussion about the application of SI to routing. Then, in Section 4, we discuss AntNet, a prototypical example of a SI algorithm for routing in wired networks, and we derive from it a basic recipe for the construction of SI routing algorithms. Subsequently, in Section 5, we look at this recipe from a different point of view, and present the general Ant Colony Routing framework. This will provide the reader with a deeper insight into the basic workings of the approach and open the door for new applications of these ideas. After that, we give an overview of existing SI algorithms for routing. In Section 6, we discuss algorithms for wired networks, and in Section 7, we discuss algorithms for wireless networks. Finally, in Section 8, we conclude the paper discussing future developments that we consider important for this field.

2 Routing in telecommunications networks

A telecommunications network can be seen as a graph $G = (N, E)$, whereby the nodes N represent routers or end user devices in the network, and the edges E represent the communication links that are available between them [22,104]. Links can be directed or undirected and can be heterogeneous in terms of physical implementation (wired or wireless), transmission capacity, propagation time, reliability, etc. Data load is injected into the network concurrently at source nodes, and needs to be forwarded to assigned destination nodes. The task of a routing algorithm is to find paths through the network graph that connect source and destination nodes, while optimizing predefined criteria and possibly satisfying certain constraints. Routing path information is commonly stored at the nodes in so-called *routing tables*. The best solution for the routing problem can vary, when there are changes to the network graph, such as, for example, the appearance or disappearance of links or nodes, or when there are variations in the network data load. A routing algorithm can either choose to ignore this variability and calculate one set of routes that will be used without change, in which case we speak of *static* routing, or it can choose to adapt and update its routes while the network is in use, in which case we speak of *dynamic* or *adaptive routing*. Most existing routing algorithms are at least to some degree adaptive, and in what follows we will focus on this class of algorithms.

The problem of adaptive routing is most naturally solved with a *distributed approach*. This is because centralized approaches have, in general, inherent limitations in terms of scalability and fault-tolerance. Moreover, in most existing networks nowadays, nodes have at least some capabilities for both network monitoring and data processing, so that the network can form a distributed monitoring and computing system. In this system, nodes locally monitor the network status, exchange this information with their neighbors, and autonomously calculate the routing tables to be used to forward data.

A wide range of different routing algorithms exist in the literature. They can be classified according to several criteria. One criterion concerns whether data need to be routed to a single destination (unicast), a group of destinations (multicast), or all nodes in the network (broadcast). Here we mainly restrict the discussion to unicast routing, which is the most widely employed in practice. Another important criterion concerns the nature of the technology that is used to create the network. Especially the distinction between the use of *wired* and *wireless* links to connect the nodes is relevant. Wired links usually have high capacity, are point-to-point or point-to-multipoint, and are quite reliable. Changes in the network are the result of variations in traffic patterns, and of removal, addition, or failure of network resources, which, however, happen at a much lower rate than the changes at the level of traffic patterns. Wireless links usually have lower capacity than wired links, offer less reliable data transport (i.e., link failures can be quite frequent), and need some arbitration mechanism to control the access to the shared wireless medium. Moreover, they can easily be reconfigured to connect to different nodes. In particular, wireless links can be effectively used to connect mobile users. Because of all these characteristics, networks containing wireless links can present a high rate of change both in terms of topology and traffic patterns.

In what follows, we give a short introduction to routing in both wired and wireless networks. Concurrently, we explain some terms that will be used in the rest of this paper, namely *proactive* versus *reactive* routing, and *top-down* versus *bottom-up* design of routing algorithms. The former terms represent a common way to classify routing algorithms, especially in the area of wireless networks (see [90]). The latter terms refer to a classical distinction in the approach to designing distributed systems (see, for example, [24]).

2.1 Routing in wired networks: top-down approaches

Wired networks implement the core structure of the current Internet, from wide-area backbone networks to metropolitan networks and local-area networks. Routing algorithms in wired networks must deal with potentially very large networks and with continual changes in traffic patterns. Moreover, they must guarantee rapid adaptation in the case of topological variations, which, however, occur relatively infrequently.

Adaptive routing algorithms for wired networks are traditionally developed using a *top-down approach*, where a well-known exact centralized algorithm for route calculation is adjusted to work in an adaptive, decentralized way. An exemplary instance of the top-down approach is the class of *link-state* routing protocols, which includes the widely used Internet routing protocol OSPF [74]. Under link-state routing, each node locally monitors the status of the links to all of its neighbors. Periodically, it constructs a message containing this local view, and floods it to all other nodes in the network. A node receiving this message combines it with similar messages received from all other nodes, in order to obtain a complete view of the network. This view is locally used to derive a weighted graph representation of the entire network and to calculate all the necessary routes using Dijkstra's shortest path algorithm [81]. This way, each node can perform routing based on a complete overview of the system, like a central routing computer would do. A different example of the top-down approach is the class of *distance-vector* routing protocols, to which the Internet protocol RIP [69] belongs. Distance-vector routing is based on the distributed

Bellman-Ford algorithm [8]. This algorithm implements a distributed version of dynamic programming, which is a general solution method for optimization problems [7]. The basic idea behind dynamic programming is to recursively split a problem into subproblems, and to use the solutions to these subproblems to construct an optimal solution for the original problem. Distance-vector routing implements dynamic programming in a distributed and asynchronous way. Nodes incrementally calculate estimates for the cost of the route to each possible destination based on estimates provided by neighboring nodes. Periodically, each node sends its estimates out to its neighbors, so that these can update their own estimates again. This iterative process of reciprocal updating allows all nodes in the network together to eventually converge to a correct set of cost estimates for all possible routes.

Both link-state and distance-vector implementations in wired networks are based on a *proactive* approach: each node periodically transmits its routing information to its neighbors and this information is used to maintain routing information for each possible destination node in the network at all times.

The top-down approach to the design of routing algorithms can naturally lead to the development of algorithms that are highly efficient and that are amenable to theoretical analysis. However, it also makes the system less robust to failures and prone to temporary inconsistencies across the network, as each node depends for its routing information on the correct functioning of all other nodes. Furthermore, it can lead to algorithms that are slow to adapt, since nodes have limited possibilities to take individual actions in response to events; instead local changes need to be propagated through the whole network. Finally, theoretical results for the original centralized algorithm of reference are usually hard to export to the distributed version. This is because these results are based on specific assumptions (e.g., link costs do not vary) that usually only hold when the network dynamics are stationary and known, which is not usually the case during the normal course of operations in modern real-world networks.

2.2 Routing in wireless networks: top-down and bottom-up approaches

In recent years, there has been an increasing interest in *wireless networks*, fueled by the growing availability of a variety of portable devices supporting a wide range of different wireless technologies. Examples of wireless networks of particular scientific interest are *mobile ad hoc networks (MANETs)* [90], *wireless mesh networks (WMNs)* [1] and *wireless sensor networks (WSNs)* [2]. MANETs are characterized by the fact that all nodes can be mobile. Because of this, the topology of MANETs can continuously change. No fixed networking infrastructure is available, so that data packets must be routed in a multi-hop fashion from node to node, with all nodes acting as peers. Routing algorithms for MANETs need to be highly adaptive and scalable. Moreover, they need to be robust to node and communication failures. Finally, they need to work in an efficient way, since mobile devices communicating through wireless connections often have limited resources to rely on, be it in terms of bandwidth, battery power, etc. WMNs are networks with a hybrid architecture: a subset of special nodes, possibly static, form an infrastructure, while the other nodes behave like in a MANET, except that for communications they can take advantage of the presence of the infrastructure nodes. Routing in WMNs usually relies on the same algorithms that are developed for MANETs. Finally, WSNs are special instances of MANETs, in which each node has on-board sensors whose sampled data need to be transported over the network to dedicated sink nodes, where they can be processed. WSNs come with their own specific challenges. Compared to MANETs, they are often much larger, so that scalability of the routing algorithms is more important. Their nodes tend to have less resources, especially in terms of battery power, stressing the need for efficiency. In terms of data traffic, they are usually based on a data-centric rather than an address-based model for data forwarding, which creates very specific traffic patterns (usually in the form of directed and reversed multi-cast trees). Mobility is less of an issue in WSNs, as nodes are normally static. Topology changes result in the first place from failures of the devices, variations over time in radio connectivity, and the fact that new sensor devices may be added. Hereafter we do not discuss WSN routing in detail, but we limit ourselves to routing in MANETs, that are a more general and traditional model of networks. The reader can consult [38] for an extensive discussion on WSNs in relationship to SI and other self-organizing approaches.

There have been some attempts to follow the top-down approach in wireless networks by adapting traditional routing algorithms for wired networks to work also in these dynamic and unstable environments. Notable examples are the *Destination-Sequenced Distance-Vector* (DSDV) protocol [82] and Optimized Link-State Routing (OLSR) [23], which implement, respectively, the distance-vector and the link-state paradigms. These are proactive algorithms. While this way of proceeding is well justified in wired networks, which provide high-bandwidth, good link reliability and topological stability, it might result hard to realize efficiently in wireless networks. This is particularly true as the network gets larger and more dynamic in terms of node mobility and data traffic patterns [14]. There is therefore an increasing interest in a different class of algorithms, called *reactive* algorithms, in which routing information is gathered and maintained only between pairs of nodes that are the source and destination of data packets. Examples of this class are AODV [83] and DYMO [21]. In these algorithms, the search for routing information is triggered by source nodes that have packets to send to unknown destinations. They flood a route request message over the network, and in case this message reaches the destination, a route reply message is sent back to set up a route. This route is then used for message forwarding until it breaks due to changes in the network. After that, the same procedures is repeated.

Reactive routing algorithm depart fundamentally from the top-down approach to the design of routing algorithms described earlier. There is no attempt to build a distributed algorithm that solves the global routing problem. Each node in the network is given the tools to act individually, and the total routing solution that is found results from these individual actions. We can refer to this as a *bottom-up* approach to the design of routing algorithms.

3 Swarm intelligence and its application to routing: a general discussion

In Section 1, we pointed out that the SI approach for control and optimization of distributed systems grew from the practice of designing systems and algorithms taking inspiration from the behavior of animal societies, and in particular of social insects. The rationale behind this lies in the observation that these systems, as well as other biological systems and mechanisms, have a distributed and modular organization that matches the structure of many problems arising in several domains of scientific and technological interest. Moreover, they show a number of properties, such as *self-organization, adaptivity, scalability, and robustness*, that are highly desirable in modern large-scale artificial systems. The interested reader can find a number of examples of animal society and other biological system behaviors that have been studied and adapted to give rise to SI frameworks and algorithms in [4, 10, 28, 34, 35, 37, 44, 60, 86].

In more general terms, the application of the SI paradigm consists in the bottom-up design of systems that mirror the organizational characteristics of the original natural systems of inspiration and aim at obtaining the mentioned set of properties. In the following we first give an informal definition of what SI design means by identifying the architectural and functional building blocks of a generic system designed according to the SI paradigm. To better illustrate the relative role and use of the different building blocks we will also continually refer to their counterpart in some natural system that has provided the basic inspiration. Then, we discuss what it means to apply this approach to the problem of adaptive routing in networking, and finally we also advocate a more pragmatic characterization of SI design, based on the investigation of common practice in the literature.

3.1 The general principles of SI design

From an architectural point of view, a SI system is a composite system made up of a (possibly large) number of partially or totally autonomous units. These units possess some basic processing and interaction capabilities, and we therefore refer to them as *agents*. Examples are the artificial ants used in ACO, which are derived from ants in ant colonies, or the particles used in PSO, which are modeled after birds' behavior. There is usually a certain level of *redundancy* among the agents, so that the system behavior can be robust with respect to decreases and increases in the number of agents.

The agents are expected to have minimal skills to solve the task at hand, and interact with each other using relatively simple protocols. Moreover, they are *situated* entities that only perceive, act, and communicate within a small portion of the environment in which they live (e.g., ants communicate in an indirect way by locally laying and reacting to pheromones). The behavior of the SI system as a whole is the synergistic result of the combination of the actions of these individual agents and their interactions with each other and with their environment. No centralized controller is strictly required. Through interactions and communications the agent system is able to *self-coordinate* and/or *self-organize*.

A key characteristic is the fact that the expected system level behavior transcends the limited capabilities of the composing agents. In this context, the term *emergent behavior* is often used, as a complex behavior of the system emerges from the combination and non-linear superposition of simpler behaviors of the agents. This means that the system's components, the agents, can be relatively simple and computationally light, if this reduction in individual complexity and effectiveness in task solving is counterbalanced by the use of a relatively large population of agents and by the implementation of effective means and protocols for interaction and communication. A good example is the earlier mentioned behavior of ant colonies, where the simple pheromone laying and following behavior of the ants allows the colony as a whole to solve a shortest path problem that is far beyond the capabilities of each individual ant.

Finally, another important aspect in SI design is the use of *stochasticity*. In the example of the ant colonies, ants have a certain preference for going to areas of higher pheromone intensity, but this is by no means a deterministic choice: some ants can still end up on paths that carry less pheromone. Stochastic decision making stimulates exploration, and allows the system to overcome limitations due to the partial and noisy view of the environment that is available to each of the individual, situated agents.

3.2 Principles and properties of SI for network routing

Based on the above definition of SI and the derived properties of SI systems, we can get a general idea of what it means to apply SI to the problem of adaptive routing in telecommunications networks. In essence, the SI way of proceeding emphasizes a bottom-up modular design that relies on self-organization, redundancy and stochasticity in order to obtain the desired global response of the system.

This is in contrast with classical top-down approaches to the design of routing algorithms (see section 2), which start from a centralized solution in which all the units have global knowledge and then decentralize it. In a sense, while top-down approaches require the explicit definition of a strategy which guarantees the solution of the problem at hand, SI approaches only require the definition of the characteristics and interaction rules of simple controllers that can collectively produce the desired solution.

This inverse problem faced by SI design is not easier than the direct problem solved by top-down approaches. In fact, it is often difficult to get a theoretical insight into the dynamics of a SI system and to predict all of its possible behaviors.

However, compared to top-down strategies, a SI strategy is expected to have some important advantages. A first of these is *adaptivity*. Thanks to the stochastic decision making used by the multiple agents working in parallel, the system provides continuous exploration and diversity, which allows it to be adaptive to changes in its environment. A second advantage is *robustness* with respect to individual agent errors, agent losses, communication errors, or other system failures. This is due to the redundancy among the multiple agents in the system. A third advantage is *scalability* with respect to problem size and/or system size. The main ingredients that allow scalability are the intrinsic redundancy and parallelism in the system, as well as the use of simple interaction protocols. Finally, SI systems are usually easily *portable* across different problem scenarios. This is a result of the above-mentioned properties of adaptivity, robustness and scalability.

All these advantages make SI design interesting for modern networks such as the wireless ad hoc networks described in Section 2, for which the design of well performing top-down approaches has proved to be challenging.

3.3 A specific and pragmatic characterization of SI for network routing

The discussion held so far has given us general insight into the application of SI to routing in telecommunications networks. However, it has remained quite broad and has not produced any concrete guidelines for the development of new SI routing algorithms or even for the classification of algorithms as being SI or not. This is on the one hand because its starting point, the basic definition of SI, is rather open and subjective: it just states that the developer should take inspiration from swarm behavior in nature. On the other hand, the properties that can be derived from this definition, such as a bottom-up, modular design, and the use of stochasticity and redundancy, are in general useful in telecommunications networks, and especially in highly distributed and dynamic ones such as wireless ad hoc networks (see Section 2), and are therefore also present in many non-SI routing algorithms (e.g., in the mentioned case of reactive algorithms for MANETs).

For these reasons, it is interesting to consider the field also from a different, more pragmatic point of view, in which we derive a more concrete definition of SI routing based on the common practice in the literature. The next two sections are dedicated to this. First, in Section 4, we describe AntNet [29], a routing algorithm for wired networks based on ACO. AntNet was one of the first SI routing algorithms to be developed, and many other

SI routing algorithms follow a design that is at least to some extent similar to AntNet. Based on the description of this algorithm and of similarities found in other algorithms, we derive some basic guidelines for the development of SI routing algorithms. After that, in Section 5, we describe *Ant Colony Routing*. This is a general framework for the design of ACO and SI routing algorithms. It describes SI routing from a perspective that is complementary to the one that is commonly followed in the field, and therefore allows us to get different insights which can in turn indicate directions for new research.

4 AntNet: lessons taken from a prototypical example of SI routing

AntNet [29] is a SI routing algorithm for packet switched IP networks. It was, together with Ant Based Control (ABC) [95], one of the first routing algorithms that followed the SI approach, and many of its mechanisms have been adopted later by other algorithms. AntNet takes its inspiration from the shortest path behavior of ant colonies and from the related ACO framework for optimization. The task of finding the shortest path between a nest and a food source is mapped onto the task of finding the shortest path between source and destination nodes in the network. The agents solving this task are artificial ants that travel to assigned destinations. Each node in the network maintains a probabilistic routing table, which plays the role of artificial pheromone: ants are stochastically forwarded through the network using the information in these routing tables, and the tables are in turn updated using feedback from the ants about the quality of the paths they have followed. Here, we first give a brief description of the algorithm as a prototypical example of what SI routing algorithms look like. Then, we use this example to derive some basic principles that are present in most SI routing algorithms.

4.1 The AntNet routing algorithm

In AntNet, every node in the network keeps two data structures: a routing table and a traffic statistics table. The routing table is also referred to as *pheromone table*. It contains for every destination a vector with one entry per outgoing link. The entry T^d_{ij} of node i's pheromone table T_i contains the pheromone value τ^d_{ij}, which is a real number indicating the relative goodness of taking outgoing link j on the way to destination d. The traffic statistics table contains general statistics about the paths to each destination, such as the average expected end-to-end delay, its expected variance, and the best end-to-end delay observed over a certain period.

Each node s in the network sends out small control packets (called *forward ants*) at regular intervals, to randomly chosen destinations. They are the agents of the SI system. Their task is to find a path to their destination

d and evaluate it. The route chosen by an ant is the result of stochastic routing decisions taken at every hop. In each intermediate node i, the ant chooses a next hop according to a probabilistic rule. The probability of each next hop j is proportional to a weighted sum of the pheromone value τ_{ij}^d and a heuristic value η_{ij}, where the latter is based on the queue length for j in i. By taking probabilistic routing decisions based on artificial pheromone, the behavior of the artificial ants is similar to that of real ants, which move preferentially in the direction of high pheromone intensities. Moreover, this introduces in the SI system the stochasticity that is needed to provide adaptivity. The heuristic value allows for adaptations based on local traffic variations.

While traveling to d, the forward ant records the delays it experiences. Once it reaches d, it becomes a *backward ant*, which returns to the source node s tracing back the path followed by the forward ant. At each node i along this path, the backward ant updates the entries in i's tables for destination d. First, the estimates in the traffic statistics table are updated using the trip time t experienced by the ant. Then the pheromone entry τ_{ij}^d is updated, with j being the neighbor over which the backward ant arrived in i. To this end, a reinforcement value r is first calculated, which reflects how good t is compared to the information about previous trip times that is stored in the statistics table. Then, τ_{ij}^d is updated with r using a moving average. The same process is repeated for all visited nodes in the path from i to d. For detailed formulas, we refer the interested reader to [28, 29]. The fact that ants update pheromone values in order to guide subsequent ants towards good solutions mimics closely the shortest path behavior of ants in nature.

Data packets are routed in a similar way as forward ants, choosing a next hop stochastically at every hop and using probabilities that depend on pheromone values. Stochastic data forwarding provides load balancing on a per packet basis. Pheromone values are raised to a power higher than 1, in order to increase the probability of taking paths with higher pheromone values. This way, data are only routed over the best paths and are not used for exploration like the ants.

In extensive simulation studies [28, 29] AntNet was compared to traditional routing algorithms, as well as to other adaptive algorithms, such as Q-routing (see [12]). It was shown to give superior performance in a wide range of different test scenarios. AntNet was also tested in real networks, showing good performance [108, 116].

4.2 General SI routing principles derived from AntNet

The description of AntNet has given us a concrete example of the SI approach to routing. Starting from this example, we now extract some general principles that are present in most SI routing algorithms in the literature.

Repeated path sampling. Routing information is gathered through the repeated sampling of full paths. This is quite different from traditional approaches to routing, such as the distance vector approach, where routing information is derived from estimates provided by neighboring nodes, or the link state approach, where routing information is calculated based on update messages received from all other nodes in the network. The use of multiple ant samples gives robustness to the system. This is because the ants, as agents of the SI system, are mutually redundant: each ant is individually unimportant and its loss can be tolerated. Also, the fact that ants always sample full paths between source and destination nodes improves robustness. This is because this way all routing information is based on real experiences, rather than on estimates or updates provided by other nodes. This avoids the creation of errors due to the use of outdated information received from other nodes, and avoids that errors that exist in one node are spread over the network when other nodes base their routing information on it. On the other hand, in case of stationary networks, where adaptivity is not necessary, this mechanism might result to be less effective than those based on global information sharing, such as link-state and distance-vector approaches.

Stochastic pheromone-based decisions. Ants choose a path to sample by constructing it hop by hop in a stochastic way using pheromone information. The use of stochastic decisions allows the exploration of multiple paths. This makes the algorithm adaptive to changes in the network environment. Moreover, it leads to the availability of multiple paths for data routing, each with an associated goodness value (see below). The use of pheromone in the path construction process allows to build on experiences gathered by previous ants. The updating and following of pheromone is the protocol by which the ant agents communicate in an indirect way. This way of communicating by locally changing and sensing the environment is generally called stigmergy [48, 106]. The use of stochastic decisions might result in loops. However, precisely because of the use of stochasticity, these are expected to be short-lived [17].

Multiple data paths. The presence of probabilistic pheromone tables automatically makes multiple paths available, which can be used both to optimize data forwarding and as backup paths in case of failures. By forwarding data probabilistically, the data load is spread over the available paths on a per packet basis. This allows the routing algorithm to make better use of available network resources and obtain better throughput. The use of pheromone in this process ensures that data is focused on the best paths. If pheromone is always kept up-to-date, by using sufficient ants, data load balancing automatically follows the changes in the network. An important aspect in the stochastic forwarding of data is that it uses a different formula than the ants, focusing more on the best pheromone. This way, ants are more explorative, while data packets concentrate on exploiting the routing information provided by the ants.

Using separate mechanisms for exploration and exploitation allows to build a flexible system.

5 Ant Colony Routing: a general template for the practical design of SI routing algorithms

Ant Colony Routing (ACR) [28] is an attempt to provide a general framework for SI routing based on the ant colony metaphor. It considers the ideas presented in Section 4 from a different point of view, which provides new insights into the possibilities of these routing strategies.

ACR considers the network as being populated by three different types of agents. These include *node manager agents*, of which there is one in each node, controlling the node's routing activities, *active perception agents*, which correspond to the ants used in AntNet, traveling through the network and making observations, and *effector agents*, which are similar to the ants, but are meant to execute certain actions in the network rather than just observe it. The node managers are situated at a higher hierarchical level than the other types of agents. They are static agents that control the pheromone tables and other internal data at the nodes. While the pheromone tables contain pheromone values reflecting the relative goodness of particular routing decisions, the remaining data contain information about other properties of the network that is useful for the solution of the routing problem at hand. Based on all this information, the node managers maintain a stochastic policy for routing. Their goal is to *learn a good policy* by adapting the pheromone values and other data to the current state of the network. To find out about the state of the network, they can either make *passive* observations, by looking at the network situation in their local environment (e.g., the amount of data passing through the node, the average transmission delay to reach its neighbors, etc.), or *active* observations, which give information about remote parts of the network. Active observations are done via active perception agents: the node managers generate such agents whenever needed. Active perception agents are similar to the ants in AntNet, with the difference that they carry parameters. These parameters control their behavior, such as, for example, how strong their preference for high pheromone values is, or the extent to which their routing decisions are based on heuristics such as local traffic variations. This means that node managers can control the way active perception agents are processed at other nodes in the network by setting proper parameter values at generation time. In this way, node managers generate active perception agents with the characteristics that they need. Finally, node managers can also generate effector agents. Effector agents are similar to active perception agents, with the difference that they travel through the network not to make an observation, but to execute some action. For example, they can reserve, or free, certain resources, such as bandwidth.

It is clear that ACR views SI routing from a different angle than we have done so far. In this new view, the active components in the SI system are not so much the ants, but rather the static managers in the nodes. They act as a *society of learning agents*, that essentially solve a reinforcement learning [101] problem: they jointly learn a policy based on information about the status of the network, which can be considered a feedback signal about their actions. The ants are a tool in this learning process, used by the managers to get the observations they need. At this point, there is no reason why ants should be generated periodically and towards random destinations, as done in AntNet. The managers can be quite flexible in their use of ants. For example, they could send ants to a specific destination in response to a certain event such as the start of a new session or the failure of a link in the network, or they could increase the ant generation rate when they sense that there is a change in the network status and they want to find out more about it. Moreover, by varying ant parameters, the manager can create heterogeneous ants with different behaviors. Ants can, for example, be made more or less exploratory, they can measure different properties of the network, and so on. Or, they can be created to perform actions in the network, in which case we speak of effector agents. All this leads to a high level of diversity in the society of mobile agents living in the network. Such diversity is important for multi-agent systems operating in a non-stationary environment, as it improves robustness, adaptivity and task distribution. Finally, we point out that the way the node managers use the gathered information to learn their policy can take different forms. For example, the AntNet-SELA algorithm [32], which adapts AntNet for QoS routing, presents an approach where the node managers are stochastic estimator learning automata (SELA) [107]. In general, models for the design of the learning process can be derived from the field of reinforcement learning [101]. One example could be policy search learning [84], which has been applied to network routing [85]. More in general, the application of learning schemes to network control is receiving increasing attention, in particular in the context of the autonomic networks [62] and cognitive networks [45, 68] paradigms.

6 SI implementations for routing in wired networks

Here, we give an overview of SI implementations for routing in wired networks. Routing in wireless ad hoc networks will be treated in the next section. We try to be as complete as possible, but due to the large amount of work that has been presented in this area, we cannot possibly be exhaustive. In what follows, we make a distinction between *connection-oriented networks, connectionless networks providing best effort services*, and *connectionless networks offering QoS*.

In connection-oriented networks, prior to the first packet sending, a path connection (*virtual circuit*) must be established between the session end-points and maintained for the duration of the session. This can be a dedicated physical connection or a logical one, shared among different data sessions. The task of the routing system is to find and use full end-to-end paths. Typical measures of performance in this case are the *session acceptance ratio*, the *delivered throughput*, and statistics of the packet latencies such as the *average end-to-end delay*. The latter two performance metrics are reference metrics for almost any network, since they summarize two basic aspects related to the quantity and the quality of the service a network can deliver. In connectionless (*datagram*) networks, packets are sent in the network without requiring the creation of an end-to-end connection, physical or virtual. Each relay node deals with the packet independently of the other nodes and makes use of packet header information to decide how to route the packet. Each packet can be sent over a different path. This can be done according to a best effort scheme, in which there is no implemented system to control the quality of the data delivery, or according to a QoS scheme, in which certain (hard or soft) guarantees are given regarding the characteristics of the data transport (e.g., in terms of maximum delay or available bandwidth). The AntNet algorithm described in Section 4.1 was developed for connectionless best effort networks.

6.1 Connection-oriented networks

The first SI routing algorithm was Ant-Based Control (ABC) [95], developed in 1996 for circuit-switched telephone networks. As in AntNet, each node s in a network running ABC periodically sends out ants to randomly chosen destinations. Each ant has an associated age, which is increased at each visited node, proportionally to the node's current load. While traveling from its source s to its destination d, the ant updates the pheromone for the path backward to s, based on its age. This is a fundamental difference with AntNet: ants update pheromone for the path leading to their source while going forward, and no backward ants are used. This approach assumes that path costs, in this case the load (i.e., the number of occupied circuits) on the visited nodes, are symmetric. Other important differences with AntNet are that no path statistics are used to evaluate path quality measurements reported by ants, and that no local heuristic is used to help guide the ants. Finally, in ABC, data packets are not routed directly according to the pheromone. This is due to the circuit-switching. Call setup messages are sent out, which follow pheromone and set up circuits; data packets are then forwarded over these circuits. The call setup messages are not routed probabilistically, but follow the best pheromone deterministically. ABC was tested in simulation on a model of the British Telecom network and was shown to give superior performance compared to other approaches [95].

A number of papers propose adaptations and extensions of ABC. The algorithm proposed in [11] combines ABC with a mechanism from dynamic programming, and allows ants to update pheromone not only towards their source node, but also towards intermediate nodes on their path. In [94], the authors extend ABC with probabilistic routing of call setups and the use of anti-pheromone, which allows ants to decrease pheromone in some cases, instead of increasing it.

In [112] Routing By Ants (RBA) is proposed. It addresses the construction of virtual circuits. RBA has similarities with both ABC and AntNet. An interesting difference with these two algorithms is the fact that the parameters which define how routing decisions are derived from pheromone values are carried by the ants, so that they can be different for each ant. This is in accordance with the general ACR framework of Section 5. The parameters are assigned to ants in their source node and are calculated using a genetic algorithm. Some improvements to this algorithm were proposed in [100].

Anti-pheromone and multiple colonies are used in the Multiple Ant Colony Optimization (MACO) algorithm for routing and load balancing [98]. An ant is expected to select paths marked by high values of pheromone of the type laid by the colony it belongs to, and get repulsed by routes marked by high values of pheromone laid by ants of other colonies. This mechanism is expected to favor the establishment of multiple disjoint paths.

A different approach is taken in the CE-ants [114], inspired by the *cross-entropy (CE) metaheuristic* for combinatorial optimization [91]. The CE method is based on the repeated sampling of paths and on the consequent adaptive adjustment of a parameter, that biases path sampling, to minimize the cross-entropy between the used generation probabilities and the optimal importance sampling probabilities. In practice, many CE algorithms have strong similarities with ACO algorithms. The same is true for CE-ants: the general architecture is quite similar to AntNet. The main difference lies in the formulas used for pheromone updating, which bear the signature of the CE method. CE-ants has been applied to a variety of routing related problems, in both connection-oriented and connectionless networks, such as the problem of finding protection cycles [115], and the problem of finding primary and backup paths [113].

The work described in [77] and other papers by the same authors addresses the problem of setting up a primary path between a source and a destination and one or more disjoint backup paths. The authors specifically consider the case of *wavelength-division multiplexing (WDM) networks*. The wavelength assignment for the routing paths is realized in a dynamic on-demand fashion. A similar problem is tackled in [109] using multiple types of ants. Ants cooperate with those of the same type and are in competition with those of different types. In this way the paths found by ants of different types will likely be mutually disjoint and can be used for backup purposes. The first to address the problem of routing and wavelength assignment in WDM networks using a SI approach were the authors of [76], who also were the first to introduce the idea of multiple ant types and the related notion of attraction/repulsion.

6.2 Connectionless networks providing best-effort services

SI routing algorithms for connectionless networks are in the first place based on AntNet (see Subsection 4.1). AntNet-FA [30] is an adaptation of AntNet. It contains an improvement in the behavior of the forward ants: AntNet-FA's forward ants do not use the same queues as data packets, but instead take high priority queues. The trip times experienced by the ants are therefore no longer representative for the end-to-end delay of data packets; the delays for data packets are instead calculated by the backward ants as the sum of local estimates maintained in each of the intermediate nodes. The main advantage is that, in this way, ants travel faster, and therefore updates are more timely.

Other papers propose further improvements over AntNet. In [33] and [78] some mechanisms to enhance the exploratory behavior of AntNet are presented. In [5], the authors propose other improvements such as the possibility to explicitly take link and node failures into account, and a better initialization of the pheromone tables. In [58], adaptive-SDR is proposed. The main difference with AntNet is that the network is divided into clusters, and a distinction is made between inter-cluster routing and intra-cluster routing. This improves scalability, since routing tables do not have to maintain entries for all possible destinations. Scalability issues of AntNet were also investigated in [19]. Finally, in [117], the authors present Adaptive Swarm-based Routing (ASR). Differences with AntNet include the use of a momentum term in pheromone updating, and the fact a node receiving a backward ant updates its pheromone matrix not only for the destination node of the ant, but for each node between itself and the destination node on the path of the backward ant.

The Ants Routing algorithm [99] builds on ABC, and makes use of the same mechanism of updating pheromone toward the source while traveling. It is meant for networks with frequent topology changes, for example due to node and link failures. The main difference with respect to ABC is the use of so-called uniform ants. These are different from regular ants in the sense that they do not have a specific destination and do not follow pheromone. Instead, they wander through the network choosing each next hop according to a uniform distribution, until they reach a maximum time-to-live, after which they are discarded. The use of uniform ants improves exploration, which is important when one wants to keep up with frequent changes. A disadvantage is that the uniform ants can lead to inefficiencies, due to the overhead they cause and the suboptimal paths they follow. The authors of [89] describe ABC-backward, which combines ABC with elements from AntNet, such as the use of backward ants and the use of the ants' trip time for pheromone updating.

In [52], the Co-operative Asymmetric Forward (CAF) mechanism is proposed. It shows how forward ants can update routing information about the path to their source without sending a backward ant and without assuming symmetric path costs. The main advantage is that the overhead created by backward ants is avoided.

In [25] the authors present an AntNet-like algorithm for routing in the specific and relatively novel domain of *networks-on-chip* (NoC). These are sub-micron scale networks that connect the elements on an integrated circuit. The implementation is an adaptation of AntNet to the constraints imposed by NoCs. The algorithm shows superior performance compared to other routing models in terms of ability to effectively balance the load, while minimizing energy consumption and the heating of the integrated circuits.

A number of other algorithms use biological metaphors that are different from the ant foraging behavior, but nevertheless follow a similar approach to routing as described in the general ACR template. The BeeHive algorithm [111] is inspired by the *foraging behavior of honey bees*. Similar to AntNet, it gathers routing information using path probing packets (called bee agents here), and it builds stochastic routing tables for data forwarding. Different from ants in AntNet, bees are deterministically flooded (with a maximum number of hops) instead of unicast along a probabilistically chosen path to a specific destination. Also different from AntNet, the network is divided into regions, so that not all destinations need to be put in the routing table of each node and better scalability can be provided. The basic algorithm and its multiple derivations have been extensively tested both in simulation and in real networks [43]. The GA-agents algorithm [65] is based on the use of a *distributed genetic algorithm (GA)*. Each node maintains a GA population, in which each individual represents a path in the network. Paths are encoded as a sequence of turns. Individuals are evaluated by letting them probe the path they represent. This way, they are similar to the ants in AntNet. Typical GA operations such as mutation and selection are executed to find the best paths.

6.3 Networks offering QoS

In QoS networks, data of different sessions are treated in different ways, in order to provide each session with the specific level of service it needs from the network. Such levels of service can be expressed in terms of various measures, such as end-to-end delay, variation in delay (jitter), bandwidth, etc. When routing is used as a tool in this process, we speak of QoS routing. The algorithms presented in Subsection 6.1 could be used to provide some form of QoS, since they rely on connection-oriented communications. Here we discuss algorithms that were specifically designed to provide QoS.

A first example of a SI-based algorithm for QoS routing is the earlier mentioned AntNet+SELA [32], which deals with QoS routing in (connection-oriented) *ATM networks*. The algorithm integrates AntNet with SELA, a framework for QoS provisioning in ATM networks that uses static reinforcement learning agents to derive routing and application admission strategies. The original SELA uses a link state approach to gather routing information. AntNet+SELA, instead, uses ant based probing. An interesting feature of AntNet+SELA is that nodes have the possibility to reactively send out extra ants in order to search specific information that they need.

A number of other adaptations of SI for QoS routing have been proposed. Most of these aim to provide hard QoS guarantees, following the *IntServ* approach [13], in which virtual circuits are set up and resources are reserved for sessions needing certain levels of service. This is the case for Agent-based Routing System (ARS), proposed in [79], and for Q-Colony, proposed in [102, 103], which makes use of multi-pheromone tables, with each table addressing a single QoS constraint. Other approaches combine SI with a soft approach to QoS, where no hard guarantees for the required levels of service are given. This is the case for Q-Colony, that can be adapted to deal with both hard and soft constraints, for the algorithm proposed in [72] and for AntNet-QoS, presented in [18]. The latter proposes an integration of AntNet with the *DiffServ* framework for QoS [63], in which levels of service are provided by assigning data sessions to certain service classes.

7 SI implementations for routing in wireless ad hoc networks

In this section, we describe SI algorithms for routing in wireless ad hoc networks. Due to the intrinsic dynamic nature of these networks, a large number of SI routing implementations have been proposed in recent years, precisely to exploit the characteristics of adaptivity and robustness of such algorithms. In the following, we briefly discuss some among the most remarkable implementations. We focus our discussion mainly on *routing in MANETs*, since these networks are the most general network model in the class of wireless ad hoc networks. Other types of wireless networks, such as sensor, satellite, and vehicular networks, present very application-specific characteristics and constraints (e.g., see the description of sensor networks in Subsection 2.2). A proper discussion of routing for those networks would require the discussion of application-specific topics that are outside the scope of this general overview. The interested reader can refer, for instance, to [26, 46, 97] for SI implementations for satellite networks, to [16, 75, 80, 92, 93] for sensor networks, and to [54] for vehicular networks.

As previously mentioned, an important issue in this field is the distinction between proactive and reactive algorithms: proactive algorithms try to maintain up-to-date routing information between all pairs of nodes in the network at all times, while reactive algorithms only gather information for nodes that are currently involved in a data communication session [90]. Reactive algorithms are more efficient and are preferred when the network is large or highly dynamic [14]. In what follows, we first discuss proactive SI routing algorithms, and then reactive and hybrid ones (which contain both reactive and proactive elements). Finally, we describe SI algorithms for QoS routing in wireless ad hoc networks.

7.1 Proactive SI routing algorithms

A number of algorithms apply the architecture of known SI routing algorithms for wired networks directly to ad hoc networks. This leads to proactive algorithms, in which all nodes send ants to all possible destinations. A first example is the Accelerated Ants Routing algorithm [71], which is derived from Ants Routing ([99]; see also Section 6.2). It contains small adaptations to this algorithm, such as the no-return rule (which simply states that ants cannot pick their previous hop as next hop, so that simple loops are avoided), and is shown in simulation to perform better than AntNet in MANETs. The ABC-AdHoc algorithm [105] on the other hand is based both on ABC and AntNet. While it uses forward ants that update pheromone for the path to their source, as is done in ABC, it uses formulas of AntNet to calculate pheromone updates and to make probabilistic routing decisions. In a simulation with rather limited mobility, the ABC-AdHoc algorithm was shown to perform better than AntNet. The W_AntNet algorithm [27] is directly derived from AntNet with the addition of periodic beacon messages for neighbor discovery and for keeping the routing tables locally up-to-date. Simulation results show that, for increasing node mobility, performance degrades with respect to AODV and DSR, and the number of looping packets increases.

Other algorithms use elements of SI in a different way, which is however still proactive. The authors of [73] propose to use a set of mobile agents that are quite independent of network nodes or data sessions: these agents are generated at network setup time, and they stay around indefinitely. They perform a continuous random walk through the network, keeping a history of the last N nodes they have visited. At each new node they arrive, paths are extracted from their history list in order to update routing information. In [15], the authors propose an algorithm that combines ants with geographic routing. In geographic routing, nodes are able to figure out their own geographic location (e.g., through use of the GPS system [53]), and data forwarding is based on the relative location of the destination and the next hops. An important issue is how a node can get to know the location of other nodes. In the proposed approach, nodes use ants, sent to randomly chosen destinations, to inform other nodes about their location. The Mobile Ants-Based Routing Protocol (MABR) [51] was designed for large scale ad hoc networks. This algorithm divides the network area in rectangular zones, corresponding to geographical areas. All nodes of a zone together make up a logical router. Long distance routing is done between logical routers, with the aid of location information. Ants are used at this level, to proactively update routing tables between logical routers. In simulation, MABR compared favorably to Terminodes routing [9], a different algorithm for large ad hoc networks. In AntHocGeo [64] the authors propose a modification of AntHocNet (see next section) which implements the concept of geographical localization of knowledge. The environment is partitioned in cells, such that a routing path is first considered at the cell level rather than at the node level. A mechanism is introduced to optimize the exchange of routing information among the location-aware nodes while they move from one cell to the other. The algorithm shows an improvement of performance with respect to AntHocNet in simulation experiments and in tests using a network of wirelessly connected mobile robots.

7.2 Reactive and hybrid SI routing algorithms

The problem with proactive approaches to routing in ad hoc networks is their limited efficiency. In the case of the SI routing algorithms described above, the continuous sending of ant agents between all possible pairs of source and destination nodes can easily saturate the limited bandwidth resources of the network. A solution to this problem is to use SI routing mechanisms reactively, focusing effort on those node pairs between which communication is going on. This is well illustrated in [6]. In this work, the authors first propose an algorithm that is very similar to AntNet. In simulation tests, the algorithm was found to perform worse than AODV, an important reference algorithm in the field, due to inefficient route discovery and large amounts of overhead. Then, the authors propose a new algorithm, called Probabilistic Emergent Routing Algorithm (PERA). This is a purely reactive algorithm: forward ants are only sent out at the start of a communication session, or when all existing routing information is out of date. They are flooded through the network towards the destination. For every copy of the forward ant that reaches the destination, a backward ant is sent to the source, so that multiple paths are created at route setup time. In simulation studies, PERA is found to have a performance that is comparable to AODV. On the downside, it must be noted that the algorithm is not very different from traditional reactive routing algorithms, such as AODV itself. This is on the one hand because, in the efforts to improve efficiency, a lot of the original SI routing mechanisms have been dropped, and on the other hand because some basic elements, such as a bottom up approach and the use of end-to-end path sampling, are generally useful in dynamic networks and have therefore been widely adopted in ad hoc network routing protocols such as AODV.

The approach of building a reactive kind of ACO routing algorithm has been followed by several other researchers in the field. Ant-Colony-Based Routing Algorithm (ARA) [50] is quite similar to PERA. One important difference is that also data packets update pheromone, so that paths which are in use are also reinforced while the data session is going on. This is equivalent to repeated path sampling. In simulation, ARA was found to perform better than AODV. In [20] ARA is enhanced with a timeout to resend forward ants to face ant losses during path discovery, and with a memory buffer to hold packets waiting for a path following a path failure. Also the Termite algorithm [88] follows a reactive approach. An important difference with ARA and PERA is that forward ants are not flooded, but follow a random walk. Moreover, backward ants do not necessarily follow the exact same path of the forward ant back to the source, but are themselves routed stochastically (this can be an advantage if unidirectional links are present). Like in ARA, pheromone updating is also done by data packets. Termite was shown to perform better than AODV for varying values of node speed. Ad hoc Networking with Swarm Intelligence (ANSI) [87] is another algorithm that follows the same approach. It only uses ants at route setup time. A mechanism using forward and backward ants is applied, and like in Termite and ARA, data packets also deposit pheromone, in order to reinforce the paths they use. Data are routed deterministically over the best paths. ANSI evaluates paths based on the congestion rates of nodes along the path. ANSI was shown to perform better than AODV in simulation.

The BeeAdHoc [43, 110] algorithm, from the same authors of BeeHive, is based on the foraging bees metaphor, and adopts a different approach, based on the use of four different types of agents. Scout agents are reactively broadcast using an increasing time-to-live heuristic in order to progressively enlarge the flooding area. When a good path is found, this is held by a different agent and made available at the nodes for data packets, that are source-routed. In this way multiple paths can be used and no routing decision is taken at the intermediate nodes. BeeAdHoc is explicitly aimed at minimizing end-to-end delay and energy consumption. In a set of extensive simulation and real world experiments, BeeAdHoc has shown performance superior to AODV and DSR.

A few algorithms mix the purely reactive approach to routing with some proactive elements, so that they can be labeled hybrid. A first example is the Emergent Ad Hoc Routing Algorithm (EARA) [67]. Like the algorithms discussed above, it uses ants to set up paths at the start of a communication session. However, it does not stop there, and keeps on sending ants to the destination for as long as the session is active. New paths are detected using ants that do random walks through the network. A very similar approach is followed in Ant Routing Algorithm for Mobile Ad hoc networks (ARAMA) [55], that also makes use of pheromone evaporation, composite pheromone metrics, and so-called negative backward ants: when a forward ant incurs in a loop or expires its time-to-live, a backward ant is generated to decrease the pheromone along the path. In AntHocNet [31, 40], the updating of pheromone and the discovery of new paths is not only executed by ants, but is also guided by a secondary process that follows the dynamic programming scheme commonly used in traditional, top-down approaches to routing in wired networks (see Section 2). AntHocNet was shown to outperform AODV and other state-of-the-art algorithms in a wide range of different open space and urban scenarios. In [57], a mechanism is added to AntHocNet in order to favor the establishment of node-disjoint multiple paths. A little improvement in performance has been observed in a set of experiments with 30 personal digital assistants (PDAs). Finally, we mention a slightly different approach presented in [70], which combines AODV and ants. The algorithm uses the basic AODV approach to gather routing information for ongoing communication sessions, while ants performing random walks through the network independently of communication sessions, source or destination nodes, update existing routing tables based on the history of the nodes they have visited. This algorithm was shown to outperform both AODV and Ants Routing.

7.3 SI for QoS routing in wireless ad hoc networks

A few SI routing algorithms for QoS routing in MANETs have also been proposed. Ant-based Distributed Routing Algorithm (ADRA) [119] follows a reactive approach, similar to the PERA algorithm described above. A difference is that, in order to support QoS, nodes check resource availability before they forward an ant, so that paths are only set up when their QoS requirements can be met. In case available resources change and an existing path can no longer rely on the necessary resources, nodes send so-called anti-ants to erase the path and inform downstream nodes that they need to find a new path. In simulations, ADRA was found to outperform the DSR routing algorithm [56], especially in more dynamic scenarios. Ant colony based Multi-path QoS-aware Routing (AMQR) [66] uses ants to set up multiple, link disjoint paths. The source node stores information about the paths followed by different ants, and combines it to construct a topology database for the network. Based on this database, it calculates n different link disjoint paths, and it sends data packets over these different paths. Pheromone is updated by the data packets. The use of a topological database is an approach that is different from most other SI routing algorithms, but can also be found in the AntNet+SELA algorithm for QoS routing in wired networks. It allows the source node to have better control over the paths that are set up. In simulation tests, AMQR was shown to outperform ADRA and DSR, especially in low mobility scenarios. EARA-QoS [67] is derived from the previously mentioned EARA and makes use of cross-layer multiple-criteria metrics to offer DiffServ routing. EARA-QoS includes in the probabilistic rule two different heuristics based on MAC-layer measures for delay and congestion and adopts a mechanism based on sequence numbers to provide multiple loop-free paths. The algorithm is a hybrid one, combining on-demand path finding with periodic ant generation for path maintenance. Simulation results show that EARA-QoS can provide good performance and can outperform AODV. In [3] the authors propose Swarm-based Distance Vector Routing (SDVR), a straightforward on-demand implementation of an AntNet scheme that uses multiple pheromone tables, one for each different QoS parameter, and combines them at decision time. A pheromone evaporation mechanism is used to reduce the attractiveness of old paths. In a number of simulation experiments, which do not involve strict QoS requirements, SDVR systematically outperforms AODV in small networks. Finally, in [118] a PSO-based approach is adopted to effectively search the path space in the case of multiple QoS metrics including power consumption.

8 Conclusions and future perspectives

The control and management of modern computer networks, which are increasingly large, dynamic, and heterogeneous, requires the development of novel algorithms and protocols that are *fully distributed, adaptive, robust, scalable*, and can let the network behave as an *autonomous and self-organizing system*. These properties are the typical fingerprints of well-engineered *swarm intelligence systems*. In the past 10 years, this fact has led a number of researchers from all over the world to apply the principles of SI to design novel algorithms for network control, and in particular, for adaptive routing, which is at the very core of the reliable and effective functioning of any telecommunication network. In this paper we have reviewed a number of these applications of the SI paradigm, considering routing algorithms for wired and wireless networks, best-effort and quality-of-service networks. At the same time, and precisely starting from the practice of algorithm implementations, we laid down the general principles underlying the application of the ideas of SI to the design of routing algorithms.

In the first part of the paper we have discussed what SI is, or, more precisely, what the notion of SI is as it is perceived in the community of its practitioners. We have framed SI as a distributed bottom-up approach that emphasizes locality of interactions and self-organization, possibly by mirroring the organizational characteristics of the original natural systems of inspiration, such as ant or bee colonies. We pointed out that this characterization of SI is nevertheless too general, such that in the case of telecommunication networks, it can encompass a large number of algorithms that have not been developed with an explicit reference to SI. This is the case of many routing algorithms for mobile ad hoc networks developed from the networking community. Therefore, we started from the common core characteristics of the implementations to derive a set of design principles that can be identified as the true fingerprints of SI for routing, and that can be used in practice for future algorithm implementations. As a matter of fact, the large majority of the implementations are based on the Ant Colony Optimization framework. Consequently, the principles that we have identified closely reflect this fact. Moreover, using these principles as building blocks, we went further, and we defined *Ant Colony Routing (ACR)*, which is a general framework for the design of ACO and SI routing algorithms. ACR describes SI routing from a perspective that is complementary to the one that is commonly followed in the field, emphasizing the aspects of distributed and cooperative reinforcement learning, and active and passive information sampling. The network nodes are seen as reinforcement learning agents that adaptively learn about network status through passive monitoring of their local traffic and connection topology, and active gathering of non-local information through, for instance, the generation of ant-like agents.

Analyzing the *performance* delivered by the reviewed routing algorithms, we can safely state that the application of the SI paradigm has so far been particularly successful. A significant number of SI routing algorithms robustly outperform state-of-the-art algorithms for the domain. However, there is still work to do to reach excellence and to face the challenges presented by current and future networks.

First, the number of *real-world implementations* of the proposed algorithms is still too limited. It is not a trivial task to effectively port a SI algorithm from simulation to real devices. For instance, the probabilistic and multi-path aspects are an issue both at the level of kernel implementation and for what concerns the effective tracking and reordering of data packets. Some algorithms such as AntNet and some bee-inspired algorithms have been implemented on physical networks [43, 108, 116], showing good performance. However, the next step should consist of the development of a more systematic policy of physical implementations, in order to, on the one hand, fully test and adapt the algorithms to physical-world constraints, and, on the other hand, favor the actual deployment of SI-based routing in real-world networks used for practical purposes.

Second, starting from the ideas behind the ACR model, it is necessary to make a *cross-over with the machine learning domain* in order to empower the basic building blocks of a SI architecture with the abilities to cooperatively learn in a fast and efficient way about network changes. Due to the very dynamic nature of modern networks, novel and effective algorithms are required to learn about the current network and user context, adapt decision policies to it, self-tune internal parameters, and self-heal the network after failures. This is the approach advocated in the recent views of *autonomic communications* [62] and *cognitive networks* [45, 68]. We believe that a cross-over between SI and reinforcement learning as outlined in the definition of ACR is the way to go to achieve these objectives and to bring a fundamental contribution to the autonomic management and control of the networks of the future.

Finally, there are a few other domains of research that are closely related to SI. It is important to compare the different approaches and promote cross-fertilization of techniques. Of particular interest in this respect is the case of *gossip/epidemics algorithms* [42]. These algorithms, derived from models of spreading of epidemics and of information gossiping in human networks, have a clear link with swarm intelligence. In fact, they are a bottom-up approach based on the metaphor of a biological network and they rely on purely local information exchanges and self-organized behaviors. Because the study of gossip algorithms is a well consolidated and continuously growing domain of research in networking, we have decided not to discuss gossip routing in this paper. On the other hand, gossip-based techniques can be fruitfully integrated into a SI architecture (examples of this way of proceeding can be found in [49, 96]). In particular, we see gossip peer-to-peer information exchanges between neighbor nodes as complementary to the typical path sampling of ACO and SI routing. For instance, we can envisage the use of gossip techniques for the cooperative information exchange and coordination among the node managers of an ACR algorithm.

References

1. I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks Journal*, 47(4):445–487, 2005.
2. I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–116, 2002.
3. R. Asokan, A. Natarajan, and A. Nivetha. A swarm-based distance vector routing to support multiple quality of service metrics in mobile ad hoc networks. *Journal of Computer Science*, 3(9):700–707, 2007.
4. O. Babaoglu, G. Canright, A. Deutsch, G.A. Di Caro, F. Ducatelle, L.M. Gambardella, N. Ganguly, M. Jelasity, R. Montemanni, A. Montresor, and T. Urnes. Design patterns from biology for distributed computing. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 1(1):26–66, 2006.
5. B. Baran and R. Sosa. A new approach for AntNet routing. In *Proceedings of the 9th International Conference on Computer Communications and Networks (ICCCN)*, pages 303–308, Washington, DC, 2000. IEEE Computer Society.
6. J. S. Baras and H. Mehta. A probabilistic emergent routing algorithm for mobile ad hoc networks. WiOpt03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2003.
7. R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
8. R. Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958.
9. L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J.-P. Hubaux, and J.-Y. Le Boudec. Self-organization in mobile ad-hoc networks: the approach of terminodes. *IEEE Communications Magazine*, 39(6), 2001.
10. E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, NY, 1999.
11. E. Bonabeau, F. Henaux, S. Guérin, D. Snyers, P. Kuntz, and G. Theraulaz. Routing in telecommunication networks with "smart" ant-like agents. In *Proceedings of IATA'98, Second Int. Workshop on Intelligent Agents for Telecommunication Applications*, volume 1437 of *Lecture Notes in Artificial Intelligence*, pages 60–71, Berlin, Germany, 1998. Springer.
12. J.A. Boyan and M.L. Littman. Packet routing in dynamically changing networks: a reinforcement learning approach. In *Advances in Neural Information Processing Systems 6 (NIPS6)*, pages 671–678, San Francisco, CA, 1994. Morgan Kaufmann.
13. R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture: an overview. RFC 1633 (Informational), 1994.
14. J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 85–97, New York, NY, 1998. ACM.
15. D. Câmara and A.A.F. Loureiro. GPS/ANT-like routing in ad hoc networks. *Telecommunication Systems*, 18(1–3):85–100, 2001.
16. T. Camilo, C. Carreto, J. Sá Silva, and F. Boavida. An energy-efficient ant-based routing algorithm for wireless sensor networks. In *Proceedings*

of the 5th International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS), volume 4150 of *Lecture Notes in Computer Science*, pages 49–59, Berlin, Germany, 2006. Springer.
17. G. Canright. Ants and loops. In *Proceedings of the 3rd International Workshop on Ant Algorithms (ANTS)*, volume 2463 of *Lecture Notes in Computer Science*, pages 235–242, Berlin, Germany, 2002. Springer.
18. L. Carrillo, C. Guadall, J. L. Marzo, G. Di Caro, F. Ducatelle, and L. M. Gambardella. Differentiated quality of service scheme based on the use of multi-classes of ant-like mobile agents. In *CoNEXT'05: Proceedings of the 2005 ACM conference on Emerging network experiment and technology*, pages 234–235, New York, NY, 2005. ACM.
19. L. Carrillo, J.L. Marzo, D. Harle, and P. Vilà. A review of scalability and its application in the evaluation of the scalability measure of AntNet routing. In *Proceedings of the IASTED Conference on Communication Systems and Networks (CSN)*, pages 317–323, Calgary, Canada, 2003. ACTA Press.
20. N. Cauvery and K. Viswanatha. Enhanced ant colony based algorithm for routing in mobile ad hoc network. *Proceedings of World Academy of Science, Engineering and Technology*, 36:30–35, 2008.
21. I. D. Chakeres and C. E. Perkins. Dynamic MANET on-demand routing protocol. Internet Draft, 2008.
22. S. Chen and K. Nahrstedt. An overview of quality-of-service routing for the next generation high-speed networks: Problems and solutions. *IEEE Network Magazine, Special issue on Transmission and Distribution of Digital Video*, 12(6):64–79, 1998.
23. T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). RFC 3626 (Experimental), 2003.
24. V. Crespi, A. Galstyan, and K. Lerman. Top-down vs bottom-up methodologies in multi-agent system design. *Autonomous Robots*, 24(3):303–313, 2008.
25. M. Daneshtalab, A. Sobhani, A. Afzali-Kusha, O. Fatemi, and Z. Navabi. NoC hot spot minimization using AntNet dynamic routing algorithm. In *Proceedings of the International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pages 33–38, Washington, DC, 2006. IEEE Computer Society.
26. S. Demeyer, M. De Leenheer, J. Baert, M. Pickavet, and P. Demeester. Ant colony optimization for the routing of jobs in optical grid networks. *Journal of Optical Networking*, 7(2):160–172, 2008.
27. S. Dhillon, X. Arbona, and P. Van Mieghem. Ant routing in mobile ad hoc networks. In *Proceedings of the 3rd International Conference on Networking and Services (INCS)*, pages 67–67, Washington, DC, 2007. IEEE Computer Society.
28. G.A. Di Caro. *Ant Colony Optimization and its application to adaptive routing in telecommunication networks*. PhD thesis, Faculté des Sciences Appliquées, Université Libre de Bruxelles, Brussels, Belgium, 2004.
29. G.A. Di Caro and M. Dorigo. AntNet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research (JAIR)*, 9:317–365, 1998.
30. G.A. Di Caro and M. Dorigo. Two ant colony algorithms for best-effort routing in datagram networks. In *Proceedings of the Tenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98)*, pages 541–546, Calgary, Canada, 1998. IASTED/ACTA Press.

31. G.A. Di Caro, F. Ducatelle, and L.M. Gambardella. Theory and practice of Ant Colony Optimization for routing in dynamic telecommunications networks. In N. Sala and F. Orsucci, editors, *Reflecting Interfaces: The Complex Coevolution of Information Technology Ecosystems*, pages 185–216. Idea Group, Hershey, PA, 2008.
32. G.A. Di Caro and T. Vasilakos. Ant-SELA: Ant-agents and stochastic automata learn adaptive routing tables for QoS routing in ATM networks. *ANTS'2000 - From Ant Colonies to Artificial Ants: Second International Workshop on Ant Colony Optimization*, Brussels, Belgium, 2000.
33. S. Doi and M. Yamamura. BntNetL and its evaluation on a situation of congestion. *Electronics and Communications in Japan (Part I: Communications)*, 85(9):31–41, 2002.
34. M. Dorigo, E. Bonabeau, and G. Theraulaz. Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16(8):851–871, 2000.
35. M. Dorigo, G.A. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.
36. M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, 26(1):29–41, 1996.
37. M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.
38. F. Dressler. *Self-Organization in Sensor and Actor Networks*. Wiley, Hoboken, NJ, 2007.
39. F. Ducatelle. *Adaptive Routing in Ad Hoc Wireless Multi-hop Networks*. PhD thesis, Università della Svizzera Italiana (USI), Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Lugano, Switzerland, 2007.
40. F. Ducatelle, G.A. Di Caro, and L.M. Gambardella. Using ant agents to combine reactive and proactive strategies for routing in mobile ad hoc networks. *International Journal of Computational Intelligence and Applications,* Special Issue on Nature-Inspired Approaches to Networks and Telecommunications, 5(2):169–184, 2005.
41. A. P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, Hoboken, NJ, 2006.
42. P. Eugster, R. Guerraoui, A.-M. Kermarrec, and L. Massoulie. From epidemics to distributed computing. *IEEE Computer*, 37(5):60–67, 2004.
43. M. Farooq. *Bee-Inspired Protocol Engineering: from Nature to Networks*. Natural Computing. Springer, Berlin, Germany, 2009.
44. M. Farooq and G.A. Di Caro. Routing protocols for next-generation intelligent networks inspired by collective behaviors of insect societies. In C. Blum and D. Merkle, editors, *Swarm Intelligence: Introduction and Applications*. Springer, Natural Computing Series, Berlin, Germany, 2008.
45. F.H. Fitzek and M. D. Katz, editors. *Cognitive Wireless Networks: Concepts, Methodologies and Visions Inspiring the Age of Enlightenment of Wireless Communications*. Springer, Berlin, Germany, 2007.
46. Z.-H. Gao, Q. Guo, and P. Wang. An adaptive routing based on an improved Ant Colony Optimization in LEO satellite networks. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, pages 1041–1044, Washington, DC, 2007. IEEE Computer Society.
47. S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels. Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76:579–581, 1989.

48. P. P. Grassé. La reconstruction du nid et les coordinations interindividuelles chez *bellicositermes natalensis et cubitermes sp.* La théorie de la stigmergie: essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6:41–81, 1959.
49. Christophe Guéret, Nicolas Monmarché, and Mohamed Slimane. Autonomous gossiping of information in a p2p network with artificial ants. In *Proceedings of the 5th International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS)*, volume 4150 of *LNCS*, pages 388–395. Springer, 2006.
50. M. Günes, U. Sorges, and I. Bouazizi. ARA - the ant-colony based routing algorithm for MANETs. In *Proceedings of the 2002 ICPP International Workshop on Ad Hoc Networks (IWAHN 2002)*, pages 79–85, Washington, DC, 2002. IEEE Computer Society.
51. M. Heissenbüttel, T. Braun, D. Jörg, and T. Huber. A framework for routing in large ad-hoc networks with irregular topologies. *International Journal of Ad Hoc & Sensor Wireless Networks, Special Issue on the Fourth Annual Mediterranean Ad Hoc Networking Workshop*, 2(2):119–128, 2006.
52. M. Heusse, D. Snyers, S. Guérin, and P. Kuntz. Adaptive agent-driven routing and load balancing in communication networks. *Advances in Complex Systems*, 1(2):237–254, 1998.
53. B. Hoffmann-Wellenhof, H. Lichtenegger, and J. Collins. *GPS: Theory and Practice*. Springer, Berlin, Germany, 2001.
54. C.-J. Huang, Y.-T. Chuang, D.-X. Yang, I-F. Chen, Y.-J. Chen, and K.-W. Hu. A mobility-aware link enhancement mechanism for vehicular ad hoc networks. *EURASIP Journal on Wireless Communications and Networking*, 8(3), 2008.
55. O. Hussein, T. Saadawi, and M. Jong Lee. Probability routing algorithm for mobile ad hoc networks' resources management. *IEEE Journal on Selected Areas in Communications*, 23(12):2248–2259, 2005.
56. D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.
57. B. Kalaavathi and K. Duraiswamy. Ant colony based node disjoint hybrid multi-path routing for mobile ad hoc networks. *Journal of Computer Science*, 4(2):80–86, 2008.
58. I. Kassabalidis, M.A. El-Sharkawi, R.J. Marks II, P. Arabshahi, and A.A. Gray. Swarm intelligence for routing in communication networks. In *Proceedings of the IEEE Global Telecommunications Conference (Globecom)*, pages 3613–3617, Washington, DC, 2001. IEEE Computer Society.
59. J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948, Piscataway, NJ, 1995. IEEE Press.
60. J. Kennedy and R. C. Eberhart. The particle swarm: social adaptation in information processing systems. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*. McGraw-Hill, London, UK, 1999.
61. J. Kennedy, R. C. Eberhart, and Y. Shi. *Swarm Intelligence*. Morgan Kaufmann, San Francisco, CA, 2001.
62. J. Kephart and D. Chess. The vision of autonomic computing. *IEEE Computer Magazine*, 36(1):41–50, 2003.
63. K. Kilkki. *Differentiated Services for the Internet*. Macmillan Publishing Co., Inc., Indianapolis, IN, 1999.

64. M. Kudelski and A. Pacut. Ant routing with distributed geographical localization of knowledge in ad-hoc networks. In *Applications of Evolutionary Computing - Proceedings of EvoWorkshops 2009 (EvoCOMNET)*, volume 5484 of *Lecture Notes in Computer Science*, pages 111–116, Berlin, Germany, 2009. Springer.
65. S. Liang, A.N. Zincir-Heywood, and M.I. Heywood. Intelligent packets for dynamic network routing using distributed genetic algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 88–96, San Francisco, CA, 2002. Morgan Kaufmann Publishers.
66. L. Liu and G. Feng. A novel ant colony based QoS-aware routing algorithm for MANETs. In *Proceedings of the First International Conference on advances in Natural Computation (ICNC)*, volume 3612 of *Lecture Notes in Computer Science*, pages 457–466, Berlin, Germany, 2005. Springer.
67. Z. Liu, M. Kwiatkowska, and C. Constantinou. A biologically inspired QoS routing algorithm for mobile ad hoc networks. In *Proceedings of the International Conference on Advanced Information Networking and Applications*, pages 426–431, Washington, DC, 2005. IEEE Computer Society.
68. Q. Mahmoud, editor. *Cognitive Networks: Towards Self-Aware Networks.* Wiley-Interscience, Chichester, UK, 2007.
69. G. S. Malkin. *RIP: An Intra-Domain Routing Protocol.* Addison-Wesley, Reading, MA, 1999.
70. S. Marwaha, C. K. Tham, and D. Srinivasan. Mobile agents based routing protocol for mobile ad hoc networks. In *Proceedings of the IEEE Global Telecommunications Conference (Globecom)*, pages 163–167, Washington, DC, 2002. IEEE Computer Society.
71. H. Matsuo and K. Mori. Accelerated ants routing in dynamic networks. In *Second International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 333–339. ACIS, 2001.
72. T. Michalareas and L. Sacks. Stigmergic techniques for solving multiconstraint routing for packet networks. In P. Lorenz, editor, *Proceedings of the First International Conference on Networking (ICN), Part II, Networking*, volume 2094 of *Lecture Notes in Computer Science*, pages 687–697, Berlin, Germany, 2001. Springer.
73. N. Minar, K. H. Kramer, and P. Maes. Cooperating mobile agents for dynamic network routing. In A. Hayzelden and J. Bigham, editors, *Software Agents for Future Communication Systems*, chapter 12, pages 287–304. Springer, Secaucus, NJ, 1999.
74. J. Moy. *OSPF: Anatomy of an Internet Routing Protocol.* Addison-Wesley, Reading, MA, 1998.
75. R. Muraleedharan and L.A. Osadciw. A predictive sensor network using ant system. In R.M. Rao, S.A. Dianat, and M.D. Zoltowski, editors, *Digital Wireless Communications VI*, volume 5440 of *Proceedings of the SPIE*, pages 181–192, Bellingham, WA, 2004. SPIE.
76. G. Navarro Varela and M. C. Sinclair. Ant Colony Optimisation for virtual-wavelength-path routing and wavelength allocation. In *Proceedings of the Congress on Evolutionary Computation*, pages 1809–1816, Piscataway, NJ, 1999. IEEE Press.
77. S. Ngo, X. Jiang, V. Le, and S. Horiguchi. Ant-based survivable routing in dynamic WDM networks with shared backup paths. *Journal of Supercomputing*, 36(3):297–307, June 2006.

78. K. Oida and A. Kataoka. Lock-free AntNet and its evaluation for adaptiveness. *Journal of IEICE B* (in Japanese), J82-B(7):1309–1319, 1999.
79. K. Oida and M. Sekido. ARS: an efficient agent-based routing system for QoS guarantees. *Computer Communications*, 23(14):1437–1447, 2000.
80. S. Okdem and D. Karaboga. Routing in wireless sensor networks using Ant Colony Optimization. In *Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, pages 401–404, Washington, DC, 2006. IEEE Computer Society.
81. C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization*. Prentice-Hall, Upper Saddle River, NJ, 1982.
82. C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications*, pages 234–244, New York, NY, 1994. ACM.
83. C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, Washington, DC, 1999. IEEE Computer Society.
84. L. Peshkin. *Reinforcement Learning by Policy Search*. PhD thesis, Department of Computer Science, Brown University, Providence, Rhode Island, April 2002.
85. L. Peshkin and V. Savova. Reinforcement learning for adaptive routing. In *International Joint Conference on Neural Networks (IJCNN)*, volume 2, pages 1825–1830, Piscataway, NJ, 2002. IEEE Press.
86. R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization, an overview. *Swarm Intelligence*, 1(1):33–57, 2007.
87. S. Rajagopalan and C. Shen. ANSI: A unicast routing protocol for mobile ad hoc networks using swarm intelligence. In *Proceedings of the International Conference on Artificial Intelligence (ICAI)*, pages 104–110. CSREA Press, 2005.
88. M. Roth and S. Wicker. Termite: A swarm intelligence routing algorithm for mobile wireless ad-hoc networks. In *Stigmergic Optimization*, chapter 7, pages 155–184. Springer, Berlin, Germany, 2005.
89. L. Rothkrantz and R. van der Put. Routing in packet switched networks using agents. First International Workshop on Ant Colony Optimization (ANTS), 1998.
90. E. M. Royer and C.-K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, 1999.
91. R. Y. Rubinstein. Combinatorial optimization, cross-entropy, ants and rare events. In S. Uryasev and P.M. Pardalos, editors, *Stochastic Optimization: Algorithms and Applications*, pages 303–364. Kluwer Academic Publisher, Dordrecht, The Netherlands, 2000.
92. M. Saleem and M. Farooq. BeeSensor: A bee-inspired power aware routing protocol for wireless sensor networks. In *Applications of Evolutionary Computing*, volume 4449 of *Lecture Notes in Computer Science*, pages 81–90, Berlin, Germany, 2007. Springer. Proceedings of EvoWorkshops 2007.
93. M. Saleem and M. Farooq. A framework for empirical evaluation of nature inspired routing protocols for wireless sensor networks. In *Proceedings of the Congress on Evolutionary Computing (CEC)*, pages 751–758, Washington, DC, 2007. IEEE Computer Society.

94. H. Sandalidis, K. Mavromoustakis, and P. Stavroulakis. Ant-based probabilistic routing with pheromone and antipheromone mechanisms. *International Journal of Communication Systems (IJCS)*, 17(1):55–62, 2004.
95. R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz. Ant-based load balancing in telecommunications networks. *Adaptive Behavior*, 5(2):169–207, 1996.
96. C.-C. Shen and S. Rajagopalan. Protocol-independent multicast packet delivery improvement service for mobile ad hoc networks. *Journal of Computer Science*, 5(2):210–227, 2007.
97. E. Sigel, B. Denby, and S. Le Heárat-Mascle. Application of ant colony optimization to adaptive routing in a LEO telecommunications satellite network. *Annals of Telecommunications*, 57(5–6):520–539, 2002.
98. K.M. Sim and W.H. Sun. Ant colony optimization for routing and load-balancing: Survey and new directions. *IEEE Transactions on Systems, Man, and Cybernetics–Part A*, 33(5):560–572, 2003.
99. D. Subramanian, P. Druschel, and J. Chen. Ants and reinforcement learning: A case study in routing in dynamic networks. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 832–838, San Francisco, CA, 1997. Morgan Kaufmann.
100. J. Sum, H. Shen, G. Young, and J. Wu. Analysis on extended ant routing algorithms for network routing and management. *Journal of Supercomputing*, 24(3):327–340, 2003.
101. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.
102. S. Tadrus and L. Bai. A QoS network routing algorithm using multiple pheromone tables. In *Proceedings of the IEEE/WIC International Conference on Web Intelligence*, pages 132–138, Washington, DC, 2003. IEEE Computer Society.
103. S. Tadrus and L. Bai. QColony: a multi-pheromone best-fit QoS routing algorithm as an alternative to shortest-path routing algorithms. *International Journal of Computational Intelligence and Applications (IJCIA)*, 5(2):141–167, 2005.
104. A. S. Tanenbaum. *Computer Networks.* Prentice-Hall PTR, Upper Saddle River, NJ, 4^{th} edition, 2002.
105. B. Tatomir and L. Rothkrantz. Dynamic routing in mobile wireless networks using ABC-AdHoc. In *Ant Colony Optimization and Swarm Intelligence: Proceedings of the fourth International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS)*, volume 3172 of *Lecture Notes in Computer Science*, pages 334–341, Berlin, Germany, 2004. Springer.
106. G. Theraulaz and E. Bonabeau. A brief history of stigmergy. *Artificial Life,* Special Issue on Stigmergy, 5(2):97–116, 1999.
107. A.V. Vasilakos and G.A. Papadimitriou. A new approach to the design of reinforcement scheme for learning automata: Stochastic Estimator Learning Algorithms. *Neurocomputing*, 7(275):649–654, 1995.
108. V. Verstraete, M. Strobbe, E. Van Breusegem, J. Coppens, M. Pickavet, and P. Demeester. AntNet: ACO routing algorithm in practice. In *Proceedings of the INFORMS Telecommunications Conference*, 2006.
109. P. Vrancx, A. Nowé, and K. Steenhaut. Multi-type ACO for light path protection. In *Proceedings of the First International Workshop on Learning and Adaption in Multi-Agent Systems (LAMAS)*, volume 3898 of *Lecture Notes in Computer Science*, pages 207–215, Berlin, Germany, 2005. Springer.

110. H. F. Wedde and M. Farooq. BeeHive: New ideas for developing routing algorithms inspired by honey bee behavior. In *Handbook of Bioinspired Algorithms and Applications*, chapter 21, pages 321–339. Chapman & Hall/CRC, Boca Raton, FL, 2005.
111. H.F. Wedde, M. Farooq, and Y. Zhang. BeeHive: An efficient fault tolerant routing algorithm under high loads inspired by honey bee behavior. In *Ants Algorithms - Proceedings of ANTS 2004, the Fourth International Workshop on Ant Algorithms*, volume 3172 of *Lecture Notes in Computer Science*, pages 83–94, Berlin, Germany, 2004. Springer.
112. T. White, B. Pagurek, and F. Oppacher. Connection management using adaptive mobile agents. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, pages 802–809. CSREA Press, 1998.
113. O. Wittner and B. E. Helvik. Cross entropy guided ant-like agents finding dependable primary/backup path patterns in networks. In *Proceedings of the Congress on Evolutionary Computation (CEC)*, pages 1528–1533, Washington, DC, 2002. IEEE Computer Society.
114. O. Wittner and B. E. Helvik. CE-Ants: Ant-like agents for path management in the next-generation internet. *Ercim News*, 64:31–32, 2006.
115. O. Wittner, B. E. Helvik, and V. Nicola. Internet failure protection using hamiltonian p-cycles found by ant-like agents. In *Proceedings of The Fifth International Workshop on Design of Reliable Communication Networks (DRCN)*, pages 437–444, Washington, DC, 2005. IEEE Computer Society.
116. Y. Yang, A. N. Zincir-Heywood, M. I. Heywood, and S. Srinivas. Agent-based routing algorithms on a LAN. In *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1442–1447, Washington, DC, 2002. IEEE Computer Society.
117. L. Yong, Z. Guang-Zhou, S. Fan-Jun, and L. Xiao-Run. Adaptive swarm-based routing in communication networks. *Journal of Zhejiang University Science*, 5(7):867–872, 2004.
118. X.-H. Zhang and W.-B. Xu. QoS based routing in wireless sensor network with Particle Swarm Optimization. In *Agent Computing and Multi-Agent Systems*, volume 4088 of *Lecture Notes in Artificial Intelligence*, pages 602–607. Springer, Berlin, Germany, 2006.
119. X. Zheng, W. Guo, and R. Liu. An ant-based distributed routing algorithm for ad-hoc networks. In *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS)*, pages 412–417, Washington, DC, 2004. IEEE Computer Society.