Programming and Computer Software, Vol. 31, No. 4, 2005, pp. 167–178. Translated from Programmirovanie, Vol. 31, No. 4, 2005.

Ant Algorithms: Theory and Applications

S. D. Shtovba

Vinnitsa State Technical University, Khmel'nitskoe sh. 95, Vinnitsa, 21021 Ukraine
e-mail: shtovba@ksu.vstu.vinnica.ua, shtovba@svitonline.com

Received March 23, 2004

Abstract—This paper reviews the theory and applications of ant algorithms, new methods of discrete optimization based on the simulation of self-organized colony of biologic ants. The colony can be regarded as a multi-agent system where each agent is functioning independently by simple rules. Unlike the nearly primitive behavior of the agents, the behavior of the whole system happens to be amazingly reasonable. The ant algorithms have been extensively studied by European researchers from the mid-1990s. These algorithms have successfully been applied to solving many complex combinatorial optimization problems, such as the traveling salesman problem, the vehicle routing problem, the problem of graph coloring, the quadratic assignment problem, the problem of network-traffic optimization, the job-shop scheduling problem, etc. The ant algorithms are especially efficient for online optimization of processes in distributed nonstationary systems (for example, telecommunication network routing).

INTRODUCTION

The aim of this paper is to discuss theoretical foundations and examples of practical applications of ant algorithms. These algorithms represent a new promising approach to solving optimization problems that is based on the simulation of the behavior of ant colonies. An ant colony can be regarded as a multi-agent system where each agent (ant) is functioning independently by very simple rules. Unlike the nearly primitive behavior of the agents, the whole system happens to function in an amazingly reasonable way: "… nests of many species of ants surprise us by their dimensions and complex and rational architectonics. There are paths and tunnels scattered on the niche territory for arphides and cochineal, and mushroom gardens… There exist various ways of storing and stoking up with food as well as a real domestication of some species of insects…" [1].

An interesting result of the cooperative behavior of biologic ants is the way they locate the shortest path from food source to the nest. Optimization algorithms imitating such a behavior of ants were proposed in early 1990s in Italy [2]. The first paper on ant algorithms was published in an international journal in 1996 [3], and, it took only a few years after this for a new field of scientific research (Swarm Intelligence and Ant Algorithms) to appear. Currently, many European researchers have successfully been working in this field. Biennially, international workshops on ant colony optimization and swarm intelligence have been organized in Belgium. Special "ant" sections and workshops have been organized in the framework of international congresses and big conferences, and special-purpose issues of international scientific journals have been published.

Ant optimization algorithms have successfully been applied to solving many complex combinatorial problems, such as the traveling salesman problem, the vehicle routing problem, the problem of graph coloring, the quadratic assignment problem, the problem of network-traffic optimization, the problem of job-shop schedule planning, etc. A key event in recognizing promises of the ant optimization was the winning of the 50000-Euro Marie Curie Excellence Award by the inventor of ant algorithms Dr. Dorigo in 2003. In spite of the quick advancement of ant algorithms, the majority of Russian-language specialists in mathematical programming have little idea of this research direction. The first paper on the ant optimization written by Russian scientists in an international journal appeared only in 2004 [4].

This paper consists of four sections. The first section presents the self-organization principles of social insects and explains the way the ants locate the shortest path. The second section brings an example of the traveling salesman problem to demonstrate how the cooperative behavior of ants can be used in algorithms of combinatorial optimization. The third section discusses methods for improving ant algorithms; and the fourth section reviews the applications of the ant algorithms. The first and second sections are based on the works [5–7].

1. PRINCIPLES OF ANT BEHAVIOR

Ants are *social* insects living within a collective (a family or colony). Some two percent of insects are social, and ants account for half of these. The number of ants in a single colony may vary from 30 to tens of millions. Ants are dominant in the Amazon basin, constituting more than 30% of the biomass of the local forests. The behavior of ants in transporting food, overcoming obstacles, building anthills, and other opera-

0361-7688/05/3104-0167 © 2005 Pleiades Publishing, Inc.

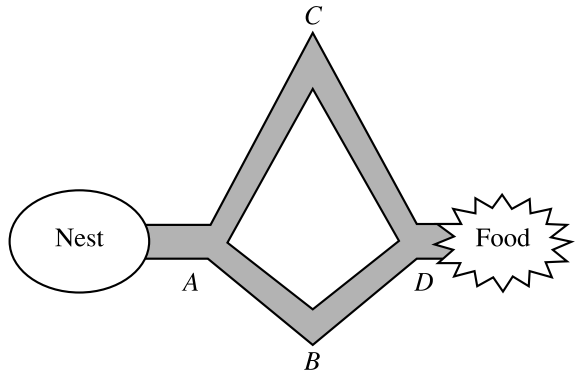

Fig. 1. An asymmetric bridge (taken from [9]).

tions is almost optimal. Principles of ant behavior have withstood the proof of one hundred million years, after they had "colonized" the Earth. Ant colonies are amazingly survivable: a reduction of up to 40% of insects has practically no effect on the functioning of the whole society [8]. A mass destruction of ants (for example, resulting from a chemical treatment of their habitat) leads to the consolidation of insects from the neighboring anthills into one family to save the society [1].

The social behavior of ants is based on *self-organization*, a set of dynamical mechanisms ensuring that the system can achieve its global aim through low-level interactions between its elements. A key feature of this interaction is that the system elements use *only local information.* In this case, any centralized control and reference to the global pattern representing the system in the external world are ruled out. Self-organization is a result of the interaction between the following four components:

(1) multiple renewal;

(2) randomness;

(3) positive feedback;

(4) negative feedback.

There are two ways of information transfer between ants: a direct communication (which includes food exchange and mandible, visual, and chemical contacts) and an indirect communication, which is called stigmergy. *Stigmergy* is a form of communication separated in time, when one participant of the communication modifies the environment, and the others make use of this information later, when they occur in a neighborhood of the modified environment. Biologically, stigmergy is realized through *pheromones*, a special secretory chemical that is deposited as trail by ants when they move. The higher the pheromone concentration on the path, the more the number of ants moving along it. With time, the pheromones evaporate, which allows the ants to adapt their behavior when the environment is modified. The distribution of pheromones is a sort of dynamically varying global memory of the anthill. At any moment, an ant can sense and change only one local cell of this global memory.

On the example of experiments with ants on an asymmetric bridge, we demonstrate how the cooperative behavior of ants makes it possible to find the shortest path to food. The asymmetric bridge (Fig. 1) connects the ant nest with the food source by two branches of different length. The experiments [9] were carried out with a laboratory colony of Argentine ants (*Iridomurmex humilis*), which deposit pheromones on the paths both from and to the nest. The scheme of the experiments was as follows:

(1) the bridge A-B-C-D was constructed;

(2) the gate at point A was opened;

(3) the numbers of ants selecting the longer (A-C-D) and shorter branches of the bridge were counted.

At the early stage of the experiments, both branches have been selected by the ants at about the same rate. In some time, almost all ants choose to move along the shortest route A-B-D, which is explained in the following way. First, the branches were free of pheromones; therefore, the A-C-D and A-B-D branches have been selected with equal rate. The ants that selected the shorter route A-B-D-B-A returned sooner with food to the nest and laid pheromone trails on this shorter branch. When they had to select the next time, the ants preferred to move along the shorter branch of the bridge, since the concentration of pheromones on it is higher. Therefore, the pheromones are accumulated faster on the branch A-B-D, attracting the ants to select the shortest route.

2. ANT APPROACH TO THE TRAVELING SALESMAN PROBLEM

The traveling salesman problem consists in finding the shortest closed route passing once through each town. The choice of this problem for the demonstration of the ideas of the ant algorithms is explained as follows:

(1) the problem can conveniently be interpreted in terms of the ant behavior: intuitively, displacements of a traveling salesman are similar to those of ants;

(2) this is an NP-hard problem;

(3) this is a traditional benchmark problem for combinatorial optimization methods. There is a big library of test traveling salesman problems and methods of their solution, which makes it possible to compare efficiency of the ant algorithms with other optimization approaches;

(4) this is a didactic problem, for which the process of searching for an optimum can be explained without discussing technical details of the algorithm;

(5) this is the first combinatorial problem that was solved by ant algorithms.

Let us consider how the four components of the ant self-organization can be implemented as applied to the optimization of the traveling salesman route.

The *multiple communication* is provided through an iterative search of the traveling salesman route by several ants simultaneously. Each ant is considered as a separate and independent traveling salesman solving his own problem. In the course of one iteration, the ant completes an entire traveling salesman route.

The *positive feedback* is provided through a simulation of the "trail-laying and trail-following" type of ant behavior. The more trails are laid on a path (on a graph edge in the traveling salesman problem), the greater the number of ants choosing this path. This results in new trails on the path, which, at the subsequent iterations of the algorithm, will attract additional ants. For the traveling salesman problem, the positive feedback is provided through the following stochastic rule: the probability that a graph edge is included into the ant route is proportional to its pheromone value. Such a probabilistic rule implements also the next component of the self-organization, the *randomness.* The shorter a path from the food source to the anthill, the more often a biological ant can pass it for a fixed time interval, laying down certain amount of pheromones during each passage. To imitate this behavior of ants, the volume of virtual pheromones laid down on a graph edge is taken to be inversely proportional to the path length. The shorter the path, the more the pheromones laid down on the corresponding edges of the graph, and the more the ants will use pheromones in synthesizing new paths.

The positive feedback alone leads to the stagnation; in this case, all ants choose one suboptimal path. To avoid this, the *negative feedback* through the pheromone evaporation is introduced. The intensity of the evaporation should not be too high; otherwise, the search area will narrow down. The evaporation should not be too fast to avoid the situation when the colony prematurely "forgets" its experience gained in the past (loss of memory), which breaks down the cooperative behavior of ants.

For each ant, the passage from a town i to a town j depends on the following three components: the tabu list, visibility, and trails of virtual pheromones.

The *tabu list* (ant memory) is a data structure that saves the list of the towns already visited, which should not be visited again. This list is growing in size during the tour and is set zero at the start of each iteration of the algorithm. Let us denote by J_{ik} the list of towns yet to be visited by the ant k located in the town i. It is clear that J_{ik} is the complement of the tabu list.

The *visibility* is a quantity reciprocal to the distance: $\eta_{ij} = 1/D_{ij}$, where D_{ij} is the distance between the towns i and j. The visibility is a local static value reflecting the heuristic desire to move to the town j from the town i: the closer the town, the stronger the desire to visit it.

The *trail of virtual pheromones* on the edge $(i-j)$ is the desire based on the experience of the colony to move to the town j from i. Unlike the visibility, the distribution of the pheromones is changed after each iteration, reflecting the experience gained by the ants. The number of virtual pheromones on the edge $(i-j)$ at an iteration t is denoted by $\tau_{ij}(t)$.

The probability that an ant k moves at iteration t from a town i to a town j is calculated by the following probabilistic–proportional rule:

$$P_{ij,k}(t) = \begin{cases} \dfrac{(\tau_{ij}(t))^{\alpha}(\eta_{ij})^{\beta}}{\sum\limits_{l \in J_{ik}} (\tau_{il}(t))^{\alpha}(\eta_{il})^{\beta}}, & \text{if } j \in J_{ik}, \\ 0, & \text{if } j \notin J_{ik}, \end{cases} \tag{1}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are adjustable parameters describing the weights of the pheromone trail and visibility when choosing the route. When $\alpha = 0$, the nearest town is chosen, which corresponds to a greedy algorithm in the classical optimization theory. When $\beta = 0$, only the pheromone trail is taken into account, which implies that all ants select one suboptimal route. To provide a good optimization dynamics, it is recommended in [3] to set $\beta \geq \alpha$.

We note that rule (1) determines the probabilities of choosing a particular town. The very choice is performed according to the "roulette-wheel" principle: each town on it has its own sector with the area proportional to probability (1). The town is chosen like throwing a roulette ball, i.e., by generating a random number to determine the sector where it stops.

When the tour is completed, the kth ant lays down on the edge (i, j) the pheromone value

$$\Delta\tau_{ij,k}(t) = \begin{cases} \dfrac{Q}{L_k(t)}, & \text{if } (i, j) \in T_k(t), \\ 0, & \text{if } (i, j) \notin T_k(t), \end{cases}$$

where $T_k(t)$ is the route of the ant k at iteration t, $L_k(t)$ is the length of the route $T_k(t)$, and $Q > 0$ is an adjustable parameter.

To study the whole space of solutions, the pheromones should evaporate. If the coefficient of evaporation is denoted by $p \in [0, 1]$, the update rule for the pheromones takes the form

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij,k}(t), \tag{2}$$

where m is the number of ants in the colony. At the early stage of the optimization process, the pheromone value on edges is taken to be equal to a small positive number τ_0.

The total number of ants in the colony remains constant. A very large colony leads to a quick growth of suboptimal routes, while a small number of ants may result in a breakdown of their cooperative behavior

```
%Ant system for solving the traveling salesman problem:
<Input of matrix D of distances>
<Initialization of the parameters of the algorithm: α, β, Q, p, and τ0>
m=n %Number of ants is equal to the number of towns
For i=1:n                               %For each edge
    For j=1:n
        If i~=j
            η(i,j)=1/D(i,j)             %Visibility
            τ(i,j)= τ0                  %Pheromone
        Else τ(i,j)=0
        End
    End
End
For k=1:m
    <Allocate ant k in randomly chosen town>
End
<Select a conditionally shortest route T+ and calculate its length L+>;
%Main loop
For t=1:tmax        %tmax - number of iterations
    For k=1:m               %For each ant:
        <Build a route Tk(t) according to rule (1)>
        <Calculate Lk(t), the length of the route Tk(t)>
    End
    If "Is the best solution found?" <Update T+ and L+> End
    For i=1:n
        For j=1:n   %For each edge
            <Update pheromone trails according to rule (2)>
        End
    End
End
<Output the shortest route T+ and its length L+>
```

Fig. 2. Ant system for the optimization of the traveling salesman route.

because of the reduced communication and quick evaporation of the pheromones. Normally, the number of ants is taken equal to the number of towns. In this case, each ant starts moving from its own town.

Unlike biological ants, the virtual agents remember the list of visited towns and live in discrete time. In addition, they are not completely "blind" and choose their routes not only by the pheromone concentration but also using heuristics [3]. These differences are governed by the fact that the virtual ants are used to solve optimization problems, rather than to simulate the insect colonies. Figure 2 presents the ant system for optimizing the traveling salesman route, which implements the above-mentioned principles of self-organization.

In our experiments for a test problem with 29 localities in Bavaria (Bays29) [10], the ant system found (after 100 iterations) an optimal route of length 2020 in two experiments out of ten. To guarantee that the optimum is found, the number of iterations in the algorithm should be increased up to one or two thousand. The algorithm prevents the set of solutions from being degenerated to a single route selected by all ants. In Fig. 3a, the best solutions found at each iteration of the ant algorithm are depicted by the thin lines. The bold lines show the best solutions found from the start of the algorithm. The fact that these lines are different implies that the ant algorithm generates new solutions at each iteration. This is attested also by Fig. 3b, which shows the standard deviations of the lengths of the routes found by the ants at the current iteration. Figure 3c shows the average (by towns) number of branches of the pheromone trails, which is obtained by finding the number of edges incidental to a graph vertex with the pheromone values exceeding some threshold. Throughout the algorithm operation, in any town, there may be found about five promising alternatives for the continuation of the route.

Compared to the exact methods (for example, the dynamic programming or the method of branches and bounds), the ant algorithm is faster in finding suboptimal solutions even for problems of low dimensions. The optimization time in the ant algorithm is a polyno-

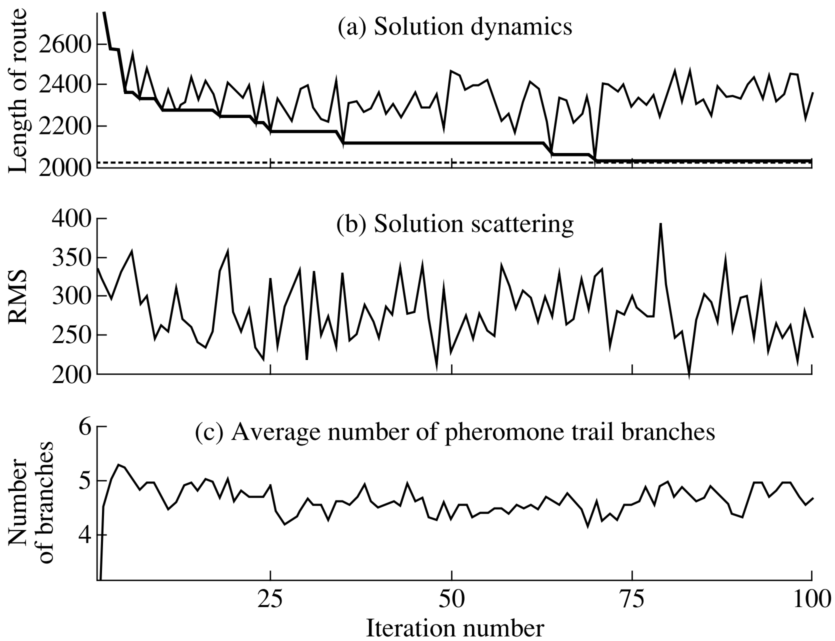

Fig. 3. The process of solving the traveling salesman problem (Bays29) by the basic ant algorithm (for $\alpha = 0.5$, $\beta = 0.5$, $Q = 2000$, and $p = 0.5$).

mial function of the dimension $O(t_{max}, n^2, m)$, whereas the dependence for the exact methods is exponential.

3. METHODS FOR IMPROVING THE ANT SYSTEM

The ant system solves the traveling salesman problems of low dimensions (with the number of towns up to 75) with the same accuracy as other general-purpose heuristic methods, such as genetic algorithms and the simulated annealing [3]. For problems of higher dimensions, the simple ant algorithm cannot compete against modern special-purpose methods for optimization of the traveling salesman route. The insufficient efficiency of the ant system is explained by the following facts:

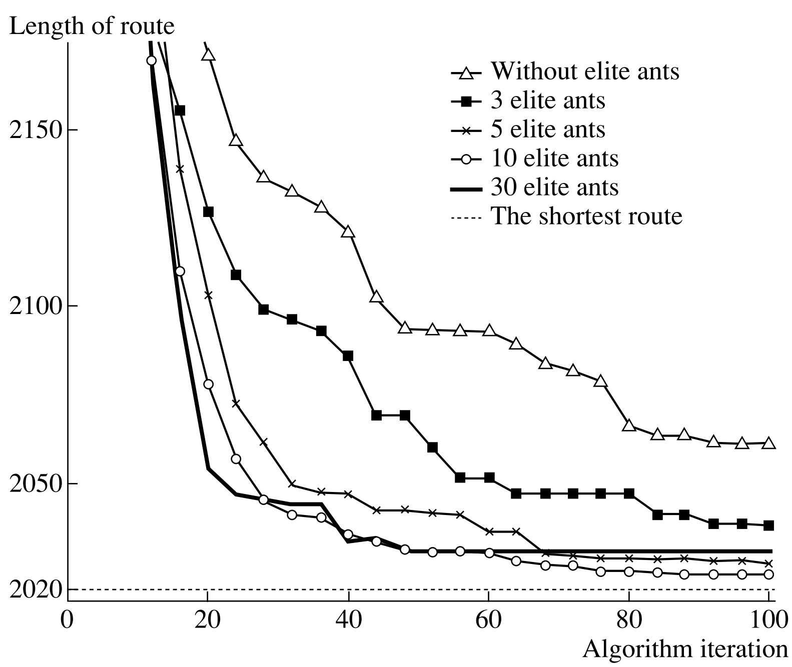

Fig. 4. Comparison of algorithms with different numbers of elite ants for the Bays29 problem.

(1) The best solution found can be lost by virtue of the probabilistic rule of route selection.

(2) The convergence near an optimum is low due to the approximately equal contributions of both the best and worst solutions to the update of the pheromones.

(3) The memory of the colony stores obviously unpromising variants, which leads to a considerable extension of the search area in high-dimensional problems.

Similar problems arise also in the genetic algorithms based on the selection by the roulette-wheel principle, when the areas of the roulette sectors for the best and worst chromosomes are almost equal [11]. To differentiate between good and bad solutions, the genetic algorithms use an adaptive fitness function, ranking of the chromosomes, and truncated and tournament selections. To save the best solution, the selection is performed with an elitism strategy, when the new population first incorporates the best chromosome and, then, the remaining chromosomes are selected. The convergence in the neighborhood of the optimum is enhanced through the use of the local search methods. To reduce the search space, the so-called "building blocks" are used. Below, we consider similar techniques for improving the ant algorithms.

3.1. Elite Ants

In the neighborhood of an optimum, the difference in the lengths of ant routes is insignificant; therefore, according to (2), the contribution of both the best and worst solutions to the update of the pheromones is almost the same. This results in slow convergence in the neighborhood of the optimum. The first improvement of the ant algorithms consists in using elite ants [3]. The elite ants deposit pheromones only on edges of the best route T^+ found.

For the traveling salesman problem, the pheromone value of an elite ant on each edge of the route T^+ is taken to be equal to Q/L^+, where L^+ is the length of the route T^+. The idea of the elitism is to increase the pheromone value in order to attract more ants, forcing them to consider solutions containing edges of the best (at a given time) route T^+. If the anthill has e elite ants, the edges of the route T^+ are additionally strengthened by the value

$$\Delta\tau_{ij,e} = eQ/L^+, \quad \forall (i,j) \in T^+. \tag{3}$$

Figure 4 presents the dynamics of averaged (over 10 runs) solutions of the Bays29 problem by algorithms with different numbers of the elite ants. When this number is high, the ants find very quickly (using 30 to 40 iterations) suboptimal routes of lengths 2033, 2028, and 2026. Then, however, the algorithms are locked for a long time in local optima, which are greatly strengthened by the elite ants. In our ten experiments with 100 iterations, the algorithms found the optimal route three times in the case of three and five elite ants, six times with ten elite ants, and only twice with thirty elite ants.

The elitism ideas are developed in rank-based ant systems [12], ant colony systems [13], MAX–MIN ant systems [14], and best-worst ant systems [15]. In these algorithms, the optimization is achieved owing to increasing the probabilities of selecting the best route fragments.

3.2. Rank-Based Ant Systems

In the rank-based algorithms, the solutions found at each iteration are ranked, and only $(w-1)$ best ants and one elite ant deposit pheromones. Thus, the bad routes are not stored. The pheromone value depends on the ant rank.

For the traveling salesman problem, rule (2) for updating pheromones takes the form

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + \sum_{r=1,\ldots,w-1} (w-r)\Delta\tau_{ij,r}(t) + w\Delta\tau_{ij,e}(t), \tag{4}$$

where

$$\Delta\tau_{ij,e} = \begin{cases} Q/L^+, & \text{if } (i,j) \in T^+, \\ 0, & \text{if } (i,j) \notin T^+ \end{cases}$$

are the pheromones of the elite ant,

$$\Delta\tau_{ij,r} = \begin{cases} Q/L^r(t), & \text{if } (i,j) \in T^r(t), \\ 0, & \text{if } (i,j) \notin T^r(t) \end{cases}$$

are pheromones of the ant with rank r, $T^r(t)$ is the route of the ant with rank r at iteration t, and $L^r(t)$ is the length of the route $T^r(t)$.

In terms of the ant system, rule (4) can be interpreted as follows: w ants move along the best route T^+, $(w-1)$ ants move along the best current route $T^1(t)$, $(w-2)$ ants move along the second-best (by rank) route $T^2(t)$, and so on. The pheromone values on edges of two routes of almost equal length differ significantly, at least by $100/(w-1)\%$. Therefore, in the neighborhood of the optimum, when the route lengths are almost the same, the ranking leads to a significant speed-up in searching for the best solution.

3.3. The Ant Colony System

In the ant colony algorithms, the weight of the best solution is increased if it is in use. At each iteration, the pheromones are updated only at the edges of the best route.

For the traveling salesman problem, the rule of the pheromone updating (2) takes the form

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + p\Delta\tau_{ij,e}(t), \tag{5}$$

where (i, j) is the edge of the best route (either at the current iteration or from the beginning of the algo-

Table 1. Solution of the traveling salesman problem by different ant algorithms [14]

Problem	Eil51	Kroa100	Dl98	Ry48p	Ft70	Kro124p	Ftv70
Optimum	426	21282	15780	14422	38673	36230	2755
MMAS + pts	**427.1**	**21291.6**	**15956.8**	14523.4	**38922.7**	**36573.6**	**2817.7**
MMAS	427.6	21320.3	15972.5	14553.2	39040.2	36773.5	2828.8
ACS	428.1	21420	16054	14565.4	39099	36857	2826.5
ASR	434.5	21746	16199.1	**14511.4**	39410.1	36973.5	2854.2
ASR + pts	428.8	21394.9	16025.2	14644.6	39199.2	37218	2915.6
ASE	428.3	21522.8	16205	14685.2	39261.8	37510.2	2952.4
ASE + pts	427.4	21431.9	16140.8	14657.9	39161	37417.7	2908.1
AS	437.3	22471.4	16702.1	15296.4	39596.3	38733.1	3154.5

rithm). For high-dimensional problems, good results can be obtained by updating the pheromones along the route T^+.

Rule (1) is modified as follows: the kth ant moves with the probability q_0 from town i to the most attractive town $z \in J_{ik}$ and, with the probability $(1 - q_0)$, selects town j by rule (1). The more the q_0, the higher the usage of the experience gained by the ant colony in synthesizing new routes. The most attractive town is determined as

$$z = \underset{j \in J_{ik}}{\operatorname{argmax}}((\tau_{ij}(t))^{\alpha}(\eta_{ij})^{\beta}).$$

The adopted rules force the ants to search for an optimum in a narrow neighborhood of the previous best solution. To support an adequate balance between explanation and exploitation, the ant colony systems include the following rule for the local update of the pheromones. At each iteration, when moving from town i to town j, the ant "eats" some amount of pheromones from the edge $(i - j)$. This edge losses its attractiveness for the other ants, thus forcing them to consider alternative routes from towns i and j. The solutions become more diverse owing to the dynamic update of the pheromone distribution.

3.4. MAX–MIN Ant System

This algorithm differs from the ant system by the following three rules:

(1) At each iteration, the pheromones are added only to the edges of the best route in line with rule (5).

(2) The pheromone value on a graph edge is confined to the range $[\tau_{\min}, \tau_{\max}]$.

(3) Initially, the pheromone value on each edge of the graph is taken to be $\tau_{\max}$.

The restrictions imposed on the pheromone value make it possible to diversify the solutions and, thus, avoid stagnation. To extend the solution domain, the max–min ant systems use a trail smoothing mechanism, according to which the pheromone value $\Delta\tau_{ij}(t)$ on the edge $(i - j)$ is proportional to $(\tau_{\max} - \tau_{ij}(t))$.

Computer experiments with the traveling salesman problem [14] show that the average time of optimization can be significantly reduced, if the pheromones are updated on the route T^+, rather than on the edges of the best route at the current iteration. For high-dimensional problems, the optimization time is reduced through a hybrid strategy, when, at certain iterations, the pheromones are laid down on the route T^+, while, at other iterations, the best current route is used. The rate of the pheromone update on the route T^+ must increase during the execution of the algorithm [14].

Table 1 compares results of solving traveling salesman problems from library [10] by the following algorithms: the ant system (AS), the ant system with elite ants (ASE), the rank-based ant system (ASR), the ant colony system (ACS), and the max–min ant system (MMAS). The symbols "+pts" mean that the trail smoothing mechanism was used. All algorithms synthesized the same number of routes: $10000 \cdot n$ for the symmetric problems Eil51, KroA100, and Dl98, and $20000 \cdot n$ for the asymmetric problems Ry48p, Ft70, Kro124p, and Ftv170. The numbers in the problem names mean the number of the towns (n). The numbers in the table cells denote lengths of the shortest routes

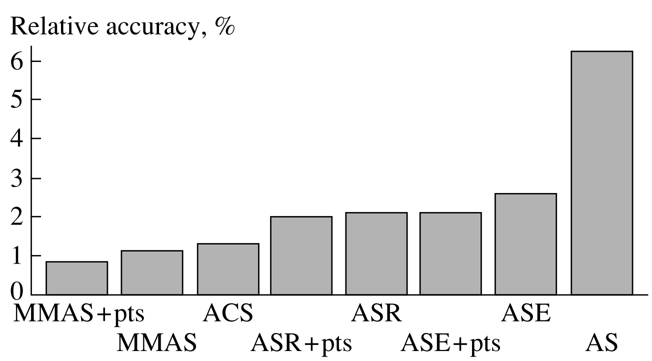

Fig. 5. Comparison of the ant algorithms in terms of relative accuracy (based on the data from Table 1).

Table 2. Comparison of metaheuristic optimization methods for the traveling salesman problem [6]

Test Problem	Eil50	Eil75	KroA100
Simulated Annealing	443	580	Data not available
Genetic Algorithms	428	545	22761
Evolutionary Programming	426	542	Data not available
Ant Colony System	**425**	**535**	**21282**

found (averaged over 25 runs). The best solutions are highlighted by the bold face.

Figure 5 compares the ant algorithms in terms of the average relative accuracy,

$$\varphi = \frac{1}{7} \sum_{i = 1, \ldots, 7} \frac{L_i - opt_i}{opt_i} \cdot 100\%,$$

where opt_i is the length of the optimal route for the ith test problem in Table 1 and L_i is the averaged length of routes found by the corresponding ant algorithm for the ith test problem.

It can be seen from Fig. 5 that the max–min ant system is most efficient, and the ant colony system is the second-best. Note that, according to the data presented in [14], for the two high-dimensional symmetric problems (Att532 and Rat783 [10]), the averaged lengths of routes obtained by the MMAS were even shorter than those obtained by the ACS.

3.5. Best-Worst Ant System

This algorithm differs from the basic ant algorithm by the following three rules:

(1) the pheromones are laid down on edges of the best route T^+ by rule (5);

(2) at a given iteration t, the pheromones evaporate only from the worst route $T^-(t)$:

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t),$$
$$(i, j) \in T^-(t) \text{ and } (i, j) \notin T^+;$$

(3) the pheromone trail on an edge $(i-j)$ is subjected to mutation with a probability p_{mut}:

$$\tau_{ij}(t+1) = \begin{cases} \tau_{ij}(t) + \Delta\tau_{\mathrm{mut}}, & \text{if } a = 0, \\ \tau_{ij}(t) - \Delta\tau_{\mathrm{mut}}, & \text{if } a \neq 0, \end{cases}$$

where $i, j = 1, \ldots, n$, $i \neq j$, $\Delta\tau_{\mathrm{mut}}$ is a random number from a range depending on the iteration number and the average pheromone value on the edges of the route T^+, and $a \in \{0, 1\}$ is a random number.

The first rule increases the contribution of the best solution. The second rule reduces the contribution of the worst current solution. The third rule is similar to the operation of mutation in genetic algorithms and is used to diversify the solutions through extending the search area. When approaching stagnation (when the best and worst solutions differ only by a few edges), the pheromone values on all edges are set equal each other, $\tau_{ij} = \tau_0$; $i, j = 1, \ldots, n$. It was shown experimentally [16] that, among the above-mentioned algorithms, the best–worst ant system is the most efficient one for solving quadratic assignment problems.

3.6. Candidate List

In high-dimensional problems, a candidate list is used. This is a small list of preferential nodes that can be reached by an ant from a given node. The candidate list is generated on the basis of prior knowledge of the problem or data updated dynamically during the solution. An ant selects a node different from those in the list only when the list has been exhausted. The candidate list makes it possible to exclude obviously unpromising variants and force the ants to consider the most promising routes, thus essentially reducing the search area.

For the traveling salesman problem, the candidate list includes neighboring towns. For example, the optimum for the problem Pr2392 with 2392 towns [10] can be found by studying route continuations to eight nearest towns [17]. The candidate list can be used with all considered modifications of the ant algorithms. The candidate list was first implemented in [18].

Table 3. Comparison of metaheuristic methods for solving the quadratic assignment problem [6]

Test Problem	Nugent (12)	Nugent (15)	Nugent (20)	Nugent (30)	Elshafei (19)	Krarup (30)
Simulated Annealing	**578**	**1150**	**2570**	6128	17937024	89800
Tabu Search	**578**	**1150**	**2570**	**6124**	**17212548**	90090
Genetic Algorithms	588	1160	2688	6748	17640584	108830
Evolution Strategies	598	1168	2654	6308	19600212	97880
Ant System	**578**	**1150**	2598	6232	18122850	92490
Ant System with Local Search	**578**	**1150**	**2570**	6128	**17212548**	**88900**

Table 4. Comparison of metaheuristic methods for solving the problem of vehicle scheduling [23]

Number of clients	Rang-based ant system with problem decomposition and local search (D-Ant)			Genetic Algorithm		Granular Tabu Search	
	Solution averaged over 10 runs	Best solution	Solution time (min)	Solution obtained	Solution time (min)	Solution obtained	Solution time (min)
200	**6460.98**	**6460.98**	7.13	**6460.98**	1.04	6697.53	2.38
255	589.28	**586.87**	139.27	596.89	14.32	593.35	11.67
300	1007.81	**1007.07**	32.55	1018.74	39.33	1016.83	21.45
399	932.58	**927.27**	158.93	933.74	78.50	936.04	33.12
420	1836.87	**1834.79**	239.47	1846.55	210.42	1915.83	43.05
480	13958.68	13816.98	240.00	13728.8	187.6	14910.62	15.13

Table 5. Comparison of the routing algorithms for the NSFNET network [42]

Algorithm	AntNet	OSRF	SRF	Daemon	BF
Mean communication delay (s)	0.93 (0.2)	5.85 (1.43)	3.58 (0.83)	0.10 (0.03)	4.27 (1.22)
Capacity ($\times 10^7$ bits/s)	2.392 (0.011)	2.100 (0.002)	2.284 (0.003)	2.403 (0.010)	1.410 (0.047)

3.7. Hybridization of Ant Algorithms

The ant algorithms are most often hybridized by adding local-search techniques. At each iteration of an algorithm, these techniques try to improve solutions found by the ants. Commonly used techniques are applied iteratively: they improve solutions until a local optimum is reached. A successful selection of the local-search method speeds up considerably solution of the optimization problem. For the traveling salesman problem, the 2-opt, 3-opt, and Lin–Kernighan local-search procedures are frequently used, which improve routes by replacing two, three, or a variable number of edges, respectively.

There have been recent efforts in hybridizing the ant algorithms with other metaheuristic optimization methods and natural computing (first and foremost, with genetic algorithms). There are two main directions of such hybridization: the island mechanism and use of genetic operations in ant algorithms. The island mechanism relies on concurrent solving the problem by genetic and ant algorithms, which exchange solutions after some time period. To date, there have been efforts in solving the traveling salesman problem and the vehicle routing problem by the island ant–genetic mechanism [19–21]; however, the time has not yet come to do some generalizations. The second direction of hybridization is based on the best–worst ant system [15, 16], where the pheromone value on the graph edges is modified through mutation.

Of interest are fuzzy <if-then> rules when using the virtual ants for the route selection. This kind of hybridization of ant algorithms makes it possible to draw good transportation schedules under fuzzy source data [22].

4. A REVIEW OF APPLICATIONS OF ANT SYSTEMS OF OPTIMIZATION

After insignificant modifications, the considered ant algorithm for optimizing the traveling salesman route can be used for solving other combinatorial problems, such as the quadratic assignment problem, the vehicle routing problem, the job-shop schedule planning problem, the problem of graph coloring, the problem of the shortest common supersequence, the problem of multiple knapsack, and the like. To solve these problems by the ant algorithms, it is required (1) to reduce them to the search for the shortest path on some graph, (2) to define the mechanisms of pheromone initialization and update, and (3) to assign heuristic rules for the route selection.

The ant algorithms can solve discrete optimization problems as successfully as other metaheuristic techniques and some problem-oriented methods. They ensure a good balance between the solution accuracy and the optimization time. To illustrate this, we compare different metaheuristic methods for solving the traveling salesman problem (Table 2) and the quadratic assignment problem (Table 3). The numbers in the table cells are values of the optimality criteria for the solutions obtained by the corresponding methods. In Table 4, three vehicle scheduling methods for high-dimensional problems are compared. The optimization time was recalculated to fit the power of the Pentium 900 Mhz processor. Boldface letters mark the currently best solutions.

The ant algorithms can also be applied to problems of stochastic combinatorial optimization. The convergence of stochastic ant algorithms to a global optimum has been demonstrated in [24].

The following applications of the ant systems should be emphasized:

• *in engineering*, multiple objective design of water irrigation grids [25], optimization of water distribution [26], optimization of the GPS geodetic grids [27], optimization of the reliability with the help of redundancy [28], ergonomic design of computer keyboards [29], data allocation in memory of supercomputers [30], and dynamic optimization of chemical processes [31];

• *in management*, university course timetabling [32], optimization of the allocation of bus stops [33], balancing transportation timetables [22];

• *in biology*, prediction of the protein folding by its amino-acid chains [34];

• *in arts*, music composition [35] and painting [36].

Good results were obtained by using the ant algorithms for learning Bayesian networks [37], classical logical rules [38], and fuzzy knowledge bases [39], as well as for the extraction of fuzzy rules from experimental data [40]. Based on ant optimization, fuzzy logic, and swarm intelligence, the Siemens Corporation developed a hybrid method for logistics control. A pilot use of this method at the storage facilities in Munich reduced delays in the delivery of goods by 44% [41].

The ant algorithms are highly efficient in the optimization of distributed nonstationary systems. An example of such problems is finding optimal traffics in telecommunication networks [6, 42]. Table 5 presents results of routing in the American network NSFNET, which consists of 14 nodes and 21 bidirectional communication lines. The following algorithms were compared: AntNet (the ant algorithm), OSRF (the official internet routing algorithm), SRF (the algorithm using a dynamic metric in calculating the connection cost), Daemon (the approximation of ideal routing algorithm), and BF (the Bellman–Ford algorithm). Table 5 shows the average delay times and capacities for the case of high network loading. The numbers in parentheses aree values of the rms deviations after ten runs of the algorithms. Data on other applications of ant algorithms can be found in review articles [43, 44] and books [6, 45].

5. CONCLUSIONS

In recent years, there has been a significant impact of biosciences on mathematics and computer technologies, leading to the genesis of a new science, technobiology, which uses biological principles to improve technology and information processes [8]. The ant algorithms can be attributed to technobiology, since they are based on the self-organization mechanisms of social insects. Proposed in the early 1990s, the ant algorithms, for ten years, have turned from "toy" demonstrations to an important field of theory of optimization.

This paper shows (on the example of the traveling salesman problem) how to solve combinatorial optimization problems with the help of the ant algorithms. Basic techniques for improving the ant algorithms, such as the use of elite ants, local-search techniques, trail smoothing mechanisms, and candidate lists, have been discussed. Modern modifications of the ant algorithms—the rank-based, max–min, and best–worst ant systems—have been analyzed.

The ant algorithms can be applied to optimization problems that are reduced to searching for the shortest path on a graph with certain constraints. The virtual ants select routes on the graph by a probability rule, based on the pheromone value and heuristic methods for solving specific problems. Computer experiments have attested that the ant algorithms ensure a good balance between the solution accuracy and the optimization time. The results are especially nice in the case of ant optimization of dynamical systems with nonstationary parameters (for example, telecommunication and computer networks). An important feature is that the ant optimization is non-convergent: even after many iterations, a variety of solutions can simultaneously be investigated, and, thus, the algorithms are not entrapped in local optima. All this allows us to recommend using ant algorithms for solving complex combinatorial optimization problems.

REFERENCES

1. Zakharov, A.A., *Muravei, semya, koloniya* (Ant, Family, Colony), Moscow: Nauka, 1978.
2. Dorigo, M., Optimization, Learning, and Natural Algorithms, *PhD Thesis*, Dipartimento di Elettronica, Politechnico di Milano (Italy), 1992.
3. Dorigo, M., Maniezzo, V., and Colorni, A., The Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Trans. Systems, Man Cybernetics, Part B*, 1996, vol. 26, no. 1, pp. 29–41.
4. Levanova, T.V. and Loresh, M.A., Ant Colony Algorithm and Simulated Annealing for the Problem of p-Median, *Avtom. Telemekh.*, 2004, no. 3, pp. 80–88.
5. Shtovba, S.D., Ant Algorithms, *Exponenta Pro. Matematika v prilozheniyakh*, 2003, no. 4, pp. 70–75.
6. Bonavear, E. and Dorigo, M., *Swarm Intelligence: from Natural to Artificial Systems*, Oxford Univ. Press, 1999.
7. Dorigo, M., Swarm Intelligence, Ant Algorithms and Ant Colony Optimization, *Reader for CEU Summer University Course "Complex System,"* Budapest: Central European University, 2001, pp. 1–38.
8. Shvetsova, N., Evolutionary Biology and High Technologies: Future Symbiosis, http://www.cnews.ru/newcom/index.shtml?2002/09/27/136108.
9. Goss, S., Aron S., Deneubourg J.L., and Pasteels, J.M., Self-Organized Shortcuts in the Argentine Ant, *Naturwissenschaften*, 1989, no. 76, pp. 579–581.

10. TSPLIB, http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

11. Gen, M. and Cheng, R., *Genetic Algorithms and Engineering Design*, Wiley, 1997.

12. Bullnheimer, B., Hartl, R.F., and Strauss, C., A New Rank-Based Version of the Ant System: A Computational Study, *Cent. Eur. J. Oper. Res. Econ.*, 1999, vol. 7, no. 1, pp. 25–38.

13. Dorigo, M., and Gambardella, L.M., Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, *IEEE Trans. Evolutionary Computation*, 1997, vol. 1, no. 1, pp. 53–66.

14. Stutzle, T., and Hoos, H.H., MAX-MIN Ant System, *Fut. Generation Comput. Syst.*, 2000, vol. 16, no. 8, pp. 889–914.

15. Cordon, O., Fernandez de Viana, I., and Moreno, L., A New AGO Model Integrating Evolutionary Concepts: The Best–Worst Ant System, *Proc. of ANTS2000—From Ant Colonies to Artificial Ants: A Series of Int. Workshops on Ant Algorithms*, Brussels, 2000, pp. 22–29.

16. Cordon, O., Fernandez de Viana, I., and Herrera, F., Analysis of the Best–Worst Ant Systems and Its Variants on the QAP, *Lecture Notes in Computer Science* (Proc. III Int. Workshop on Ant Algorithms ANTS 2002), Berlin: Springer, 2002, no. 2463, pp. 228–234.

17. Reinelt, G., The Traveling Salesman: Computational Solutions for TSP Applications, *Lecture Notes in Computer Science*, Berlin: Springer, 1994, vol. 840.

18. Gambardella, L.M. and Dorigo, M., Solving Symmetric and Asymmetric TSPs by Ant Colonies, *Proc. IEEE Conf. on Evolutionary Computation—ICEC96*, Piscataway, USA, 1996, pp. 622–627.

19. Reimann, M., Shtovba, S., and Nepomuceno, E., A Hybrid Ant Colony Optimization and Genetic Algorithm Approach for Vehicle Routing Problems Solving, *Student Papers of Complex Systems Summer School-2001*, Budapest, 2001, pp. 134–141.

20. Pilat, M. and White, T., Using Genetic Algorithm to Optimize ACS-TSP, *Lecture Notes in Computer Science* (Proc. III Int. Workshop on Ant Algorithms ANTS 2002), Berlin: Springer, 2002, no. 2463, pp. 282–287.

21. Acan, A., GAACO: A GA + ACO Hybrid for Faster and Better Search Capability, *Lecture Notes in Computer Science* (Proc. III Int. Workshop on Ant Algorithms ANTS 2002), Berlin: Springer, 2002, no. 2463, pp. 300–301.

22. Lucic, P., Modeling Transportation Problems Using Concepts of Swarm Intelligence and Soft Computing, *PhD Thesis*, Civil Engineering Department, Virginia Polytechnic Institute and State University, Virginia, USA, 2002.

23. Reimann, M., Ant Based Optimization in Good Transportation, *PhD Thesis*, University of Vienna, Vienna, Austria, 2002.

24. Gutiahr, W.J., A Converging ACO Algorithm for Stochastic Combinatorial Optimization, *Lecture Notes in Computer Science* (Proc. of SAFA-2003 (Stochastic Algorithms: Foundations and Applications)), Berlin: Springer, 2003, no. 2827, pp. 10–25.

25. Mariano, C.E. and Morales, E., MOAQ: An Ant-Q Algorithm for Multiple Objective Optimization Problems, *Proc. of Genetic and Evolutionary Computation Conf. (GECCO-99)*, San-Francisco, 1999, vol. 1, pp. 894–901.

26. Maier, H.R., Simpson, A.R., Zecchin, A.C., Wai Kuan Foong, Kuang Yeow Phang, Hsin Yeow Seah, and Chan Lim Tan, Ant Colony Optimization for Design of Water Distribution Systems. *J. Water Resources Planning Manag.*, vol. 129, no. 3, pp. 200–209.

27. Hussian Aziz Saleh, *Ants Can Successfully Design GPS Surveying Networks*, http://www.gpsworld.com/gpsworld/article/articleDetail.jsp?id=30690&pageID=1&sk=&date=. Translated under the title *Povedenie murav'ev mozhno uspeshno ispol'zovat' dlya razrabotki s'emochnykh setei GPS optimal'noi struktury*, http://www.agp.ru/projects/ ants/.

28. Liang Yun-Chia and Smith, A.E., An Ant System Approach to Redundancy Allocation, *Proc. Cong. Evolutionary Computation (CEC-99)*, 1999, vol. 2.

29. Eggers, J., Feillet, D., Kehl, S., Wagner, M.O., and Yannou, B., Optimization of the Keyboard Arrangement Problem Using an Ant Colony Algorithm, *Eur. J. Operational Res.*, 2003, no. 148, pp. 672–686.

30. Rodrigues, A., Application of Ant Colony Optimization to Data Distribution in Memory in Computer Systems, *Proc. VII Annual Swarm Researchers Meeting "SwarmFest-2003,"* Notre Dame, USA, 2003, http://www.nd.edu/arodrig6/.

31. Rajesh, J., Gupta, K., Kusumakar, H.S., Jayaraman, V.K., and Kulkarni, B.D., Dynamic Optimization of Chemical Processes Using Ant Colony Framework, *Comput. Chem.*, 2001, vol. 25, no. 6, pp. 583–595.

32. Socha, K., Knowles, J., and Samples, M., A MAX-MIN Ant System for the University Course Timetabling Problem, *Lecture Notes in Computer Science* (Proc. III Int. Workshop on Ant Algorithms ANTS 2002), Berlin: Springer, 2002, no. 2463, pp. 1–13.

33. De Jong, J., Multiple Ant Colony Systems for the Busstop Allocation Problem, *MS Thesis*, Department of Philology, University of Utrecht, Utrecht, Holland, 2001.

34. Shmygelska, A. and Hoos, H., An Improved Ant Colony Optimization Algorithm for the 2D HP Protein Folding Problem, *Proc. XVI Canadian Conf. Artificial Intelligence (AI-2003)*, Canada, 2003.

35. Gueret, C., Monmarche, N., and Slimane, M., Ants Can Play Music, *Lecture Notes in Computer Science* (Proc. of IV Int. Workshop on Ant Colony Optimization and Swarm Intelligence ANTS-2004), Berlin: Springer, 2004, no. 3172, pp. 310–317.

36. Aupetit, S., Bordeau, V., Monmarche, N., Slimane, M., and Venturini, G., Interactive Evolution of Ant Paintings, *Proc. of IEEE Cong. on Evolutionary Computation*, Canberra: IEEE Press, 2003, pp. 1376–1383.

37. De Campos, L.M., Gamez, J.A., and Puerta, J.M., Learning Bayesian Networks by Ant Colony Optimization: Searching in Two Different Spaces, *Mathware & Soft Computing*, 2002, no. 9.

38. Raspinelli, J.M., Lopes, H.S., and Freitas, A.A., Data Mining with an Ant Colony Optimization Algorithm, *IEEE Trans. Evolutionary Computation (Special issue on Ant Colony Algorithms)*, 2002, vol. 6, no. 4, pp. 321–332.

39. Cassillas, J., Cordon, O., and Herrera, F., Learning Fuzzy Rules Using Ant Colony Optimization Algorithms, *Proc. of ANTS2000—From Ant Colonies to Arti-*

ficial Ants: A Series of Int. Workshops on Ant Algorithms, Brussels, 2000, pp. 13–21.

40. Cassillas, J., Cordon, O., Fernandez de Viana, I., and Herrera, F., Learning Cooperative Linguistic Fuzzy Rules Using the Best–Worst Ant System Algorithm, *Int. J. Intelligent Sys.*, 2005, vol. 20, pp. 433–452.
41. Component Transport Follows the Ant Trail, http://www.siemens.com. September 23, 2004.
42. Caro, G.D. and Dorigo, M., Anet: A Mobile Agents Approach to Adaptive Routing, Tech. Rep. IRIDA 97-12, Brussels: Univ. Libre de Brusseles, 1997.
43. Dorigo, M. and Stutzle, T., The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances, in *Handbook of Metaheuristics*, Glover, F. and Kochenberger, G., Eds., Norwell: Kluwer, 2002.
44. Cordon, O., Herrera, F., and Stutzle, T., A Review on the Ant Colony Optimization Metaheuristic: Basis, Models and New Trends, *Mathware & Soft Computing*, 2002, no. 9.
45. Dorigo, M. and Stutzle, T., *Ant Colony Optimization*, Bradford Book, 2004.