A Concise Overview of Applications of Ant Colony Optimization

Thomas Stützle, Manuel López-Ibáñez and Marco Dorigo
Université Libre de Bruxelles (ULB), Brussels, Belgium
{stuetzle,manuel.lopez-ibanez,mdorigo}@ulb.ac.be

Ant Colony Optimization (ACO) [1, 2, 3] is a metaheuristic for solving hard combinatorial optimization problems inspired by the indirect communication of real ants. In ACO algorithms, (artificial) ants construct candidate solutions to the problem being tackled, making decisions that are stochastically biased by numerical information based on (artificial) pheromone trails and available heuristic information. The pheromone trails are updated during algorithm execution to bias the ants search towards promising decisions previously found. Chapter **[Chapter "Ant Colony Optimization", this encyclopedia]** gives a detailed overview of the main concepts of ACO.

Despite being one of the youngest metaheuristics, the number of applications of ACO algorithms is very large. In principle, ACO can be applied to any combinatorial optimization problem for which some iterative solution construction mechanism can be conceived. Most applications of ACO deal with $\mathcal{NP}$-hard combinatorial optimization problems, that is, with problems for which no polynomial time algorithms are known. ACO algorithms have also been extended to handle problems with multiple objectives, stochastic data and dynamically changing problem information. There are, as well, extensions of the ACO metaheuristic for dealing with problems with continuous decision variables.

This chapter provides a concise overview of several noteworthy applications of ACO algorithms. This overview is necessarily incomplete because the number of currently available ACO applications goes into the hundreds. Our description of the applications follows the classification used in the 2004 book on ACO by Dorigo & Stützle [3] but extending the list there with many recent examples. Tables 1 and 2 summarize those applications.

1 Applications to $\mathcal{NP}$-hard problems

ACO was primarily intended for solving combinatorial optimization problems, among which $\mathcal{NP}$-hard problems are the most challenging ones. In fact, no polynomial-time algorithms are known for such problems, and therefore heuristic techniques such as ACO are often used for generating high-quality solutions in reasonable computation times.

Table 1: Applications of ACO algorithms to $\mathcal{NP}$-hard problems.

Problem type	*Problem name*	*Authors*	*Year*	*References*
Routing	Traveling salesman	Dorigo et al.	1991, 1996	[4, 5]
		Dorigo & Gambardella	1997	[6]
		Stützle & Hoos	1997, 2000	[7, 8]
	Vehicle routing (VRP)	Bullnheimer et al.	1999	[9]
		Reimann et al.	2004	[10]
		Rizzoli et al.	2007	[11]
	VRP with time windows	Gambardella et al.	1999	[12]
	VRPMTWMV	Favoretto et al.	2007	[13]
	VRP with loading constraints	Doerner et al.	2006	[14]
		Fuellerer et al.	2009	[15, 16]
	Team orienteering	Ke et al.	2008	[17]
	Sequential ordering	Gambardella & Dorigo	2000	[18]
	TSP with time windows	López-Ibáñez & Blum	2010	[19]
Scheduling	Single machine	Den Besten et al.	2000	[20]
		Merkle & Middendorf	2000, 2002	[21, 22]
		Meyer & Ernst	2004	[23]
		Liao & Juan	2007	[24]
		Meyer	2008	[25]
	Flow shop	Stützle	1998	[26]
		Rajendran & Ziegler	2004	[27]
	Industrial scheduling	Gravel et al.	2002	[28]
	Project scheduling	Merkle et al.	2002	[29]
	Group shop	Blum	2004	[30]
	Job shop	Blum	2004	[30]
		Huang & Liao	2008	[31]
	Open shop	Blum	2005	[32]
	Car sequencing	Khichane et al.	2008	[33]
		Solnon	2008	[34]
		Morin et al.	2009	[35]
Subset	Multiple knapsack	Leguizamón & Michalewicz	1999	[36]
		Ke et al.	To appear	[37]
	Maximum independent set	Leguizamón & Michalewicz	1999	[36]
	Redundancy allocation	Liang & Smith	1999	[38]
	Weight constraint graph tree partitioning	Cordone & Maffioli	2001	[39]
	Bin packing	Levine & Ducatelle	2002	[40]
	Set covering	Lessing et al.	2004	[41]
	Set packing	Gandibleux et al.	2004	[42]
	l-cardinality trees	Blum & Blesa	2005	[43]
	Capacitated minimum spanning tree	Reimann & Laumanns	2006	[44]
	Maximum clique	Solnon & Fenet	2006	[45]
	Multi-level lot-sizing	Pitakaso et al.	2006, 2007	[46, 47]
		Almeder	2010	[48]
	Edge-disjoint paths	Blesa & Blum	2007	[49]
	Feature Selection	Sivagaminathan & Ramakrishnan	2007	[50]
	Multicasting ad-hoc networks	Hernández & Blum	2009	[51]

(continues . . .)

Table 1: Applications of ACO algorithms to $\mathcal{NP}$-hard problems (continued).

Problem type	*Problem name*	*Authors*	*Year*	*References*
Assignment & layout	Quadratic assignment	Maniezzo et al.	1994, 1999	[52, 53]
		Stützle & Hoos	2000	[8]
	Graph coloring	Costa & Hertz	1997	[54]
	Generalized assignment	Lourenço & Serra	1998	[55]
	Frequency assignment	Maniezzo & Carbonaro	2000	[56]
	Constraint satisfaction	Solnon	2000, 2002	[57, 58]
	Course timetabling	Socha et al.	2002, 2003	[59, 60]
	Ambulance location	Doerner et al.	2005	[61]
	MAX-SAT	Pinto et al.	2007	[62]
	Assembly line balancing	Bautista & Pereira	2007	[63]
	Simple assembly line balancing	Blum	2008	[64]
	Supply chain management	Silva et al.	2009	[65]
Machine learning	Bayesian networks	De Campos et al.	2002	[66, 67]
		Pinto et al.	2009	[68]
	Classification rules	Parpinelli et al.	2002	[69]
		Martens et al.	2007	[70]
		Otero et al.	2008	[71]
Bioinformatics	Shortest common supersequence	Michel & Middendorf	1998, 1999	[72, 73]
	Protein folding	Shmygelska & Hoos	2005	[74]
	Docking	Korb et al.	2006, 2007	[75, 76]
	Peak selection in biomarker identification	Ressom et al.	2007	[77]
	DNA sequencing	Blum et al.	2008	[78]
	Haplotype inference	Benedettini et al.	2008	[79]

Table 2: Applications of ACO algorithms to “non-standard” problems.

Problem type	*Problem name*	*Authors*	*Year*	*References*
Multi-objective	Scheduling	Iredi et al.	2001	[80]
	Portfolio Selection	Doerner et al.	2001, 2004	[81, 82]
	Quadratic assignment	López-Ibáñez et al.	2004, 2006	[83, 84]
	Knapsack	Alaya et al.	2007	[85]
	Traveling salesman	García-Martínez et al.	2007	[86]
	Activity Crashing	Doerner et al.	2008	[87]
	Orienteering	Schilde et al.	2009	[88]
Continuous	Neural networks	Socha & Blum	2007	[89]
	Test problems	Socha & Dorigo	2008	[90]
Stochastic	Probabilistic TSP	Bianchi et al.	2002	[91]
		Bianchi & Gambardella	2007	[92]
		Balaprakash et al.	2009	[93]
	Vehicle routing	Bianchi et al.	2006	[94]
	Screening policies	Brailsford et al.	2006	[95]
Dynamic	Network routing	Di Caro & Dorigo	1998	[96]
		Di Caro et al.	2005	[97]
	Dynamic TSP	Guntsch & Middendorf	2001, 2002	[98, 99]
		Eyckelhof & Snoek	2002	[100]
		Sammoud et al.	To appear	[101]
	Vehicle routing	Montemanni et al.	2005	[102]
		Donati et al.	2008	[103]

1.1 Routing problems

Routing problems involve one or more agents visiting a predefined set of locations, and the objective function and constraints depend on the order in which the locations are visited. Perhaps the best-known example is the traveling salesman problem (TSP) [104, 105]. In fact, the first ACO algorithm, Ant System (AS) [4, 5, 106, 107], was first tested using this problem. Although AS could not compete with state-of-the-art algorithms for the TSP, it was the starting point for the development of various high performing ACO algorithms. The application of AS to the TSP also stimulated the application of ACO to other routing and combinatorial problems.

For instance, ACO has obtained very good results in the sequential ordering problem, an extension of asymmetric TSP with precedence constraints among nodes. At the time it was proposed, the algorithm by Gambardella & Dorigo [18] was the best available algorithm for this problem, improving upon many best-known solutions. Recently, stochastic sampling has been integrated into a Beam-ACO algorithm for the TSP with time windows (TSPTW) [19], which is an extension of the classical TSP with time window constraints; Beam-ACO is a combination of ACO algorithms with beam-search [32].

ACO algorithms have been successful in tackling various variants of the vehicle routing problem (VRP). The first application of ACO to the capacitated VRP was due to Bullnheimer et al. [9]. More recently, Reimann et al. [10] proposed a particular ACO algorithm (D-Ants) for the capacitated VRP. Gambardella et al. [12] introduced MACS-VRPTW, an ACO algorithm for the VRP with time window constraints (VRPTW), which reached state-of-the-art results when it was proposed. Favaretto et al. [13] proposed an ACS algorithm for a variant of the VRP with multiple time windows and multiple visits (VRPMTWMV). Fuellerer et al. [15] used an ACO algorithm for a problem that combines the two-dimensional packing and the capacitated vehicle routing problem (2L-CVRP), showing that it outperforms a tabu search algorithm. In this problem, items of different sizes and weights are loaded in vehicles with a limited weight capacity and limited two-dimensional loading surface, and then they are distributed to the customers. Other variants of VRP with different loading constraints have also been tackled by means of ACO [14, 16].

Ke et al. [17] have recently proposed an ACO approach to the team orienteering problem (TOP), where the goal is to find the set of paths from a starting point to an ending point that maximizes the reward obtained by visiting certain locations taking into account that there are restrictions on the length of each path.

1.2 Scheduling problems

Scheduling problems concern the assignment of jobs to one or various machines over time. Input data for these problems are processing times but also often additional setup times, release dates and due dates of jobs, measures for the jobs' importance and precedence constraints among jobs. Scheduling problems have been an important application area of ACO algorithms, and the currently available ACO applications in scheduling deal with many different job and machine characteristics.

The single-machine total weighted tardiness problem (SMTWTP) has been tackled by both den Besten et al. [20] and Merkle & Middendorf [21, 22] using variants of ACS (ACS-SMTWTP). In ACS-SMTWTP, a solution is determined by a sequence of jobs. The positions of the sequence are filled in their canonical order, that is, first a job is assigned to position one, next a job to position two, and so on, until position n. Pheromone trails are defined as the desirability of scheduling job j at position i, a pheromone trail definition that is used in many ACO applications to scheduling problems [20, 26, 108, 109]. Merkle & Middendorf [21] used sophisticated heuristic information and an algorithmic technique called *pheromone summation rule*, which has proven to be useful in many applications of ACO to scheduling problems. On the other hand, den Besten et al. [20] combined ACS-SMTWTP with a powerful local search algorithm, resulting in one of the best algorithms available for this problem in terms of solution quality. Another application of ACO to a variant of this problem with sequence-dependent setup times has recently been studied by Liao & Juan [24]. Meyer & Ernst [23] and Meyer [25] studied the integration of constraint programming techniques into ACO algorithms using as a case study a single-machine problem with sequence-dependent setup times, release dates and deadlines for jobs.

ACO algorithms have also been proposed for the permutation flow-shop problem. The first approach is due to Stützle [26], who proposed a hybrid between $\mathcal{MM}$AS and ACS. Later, Rajendran & Ziegler [27] improved its performance by introducing the pheromone summation rule. For this problem, however, the results of existing ACO algorithms are behind the current state-of-the-art algorithms. This is also the case for the well-known job-shop problem [30], although recent results hybridizing ACO and tabu search seem promising [31]. Nevertheless, for various other scheduling problems ACO algorithms are nowadays among the best performing algorithms available. Beam-ACO, the hybrid between beam search and ACO, is a state-of-the-art algorithm for open shop scheduling [32]. In addition, a variant of $\mathcal{MM}$AS obtained excellent results in the group shop problem [30].

Another scheduling problem where ACO obtained excellent results is the resource-constrained project scheduling problem, in which a set of activities must be scheduled, subject to resource constraints and precedence constraints among the activities, such that the completion of the last activity is as early as possible. At the time of its publication, the ACO algorithm proposed by Merkle et al. [29] was the best available.

Finally, state-of-the-art results have been obtained in the car sequencing problem by the ACO algorithm proposed by Solnon [34], and these results have been further improved by Morin et al. [35] by means of a specialized pheromone model. The car sequencing problem has also been used as an example application by Khichane et al. [33] to explore the integration of constraint programming techniques into ACO algorithms.

1.3 Subset problems

The goal in subset problems is, generally speaking, to find a subset of the available items that minimizes a cost function defined over the items and that satisfies a number of constraints. This is a wide definition that can include other classes of problems. There are, however, two characteristic properties of the solutions to subset problems: The order of the solution components is irrelevant, and the number of components of a solution may differ from solution to solution.

An important subset problem is the set covering problem (SCP). Lessing et al. [41] compared the performance of a number of ACO algorithms for the (SCP), with and without the usage of a local search algorithm based on 3-flip neighborhoods [110]. The best performance results were obtained, as expected, when including local search. For a large number of instances the computational results were competitive with state-of-the-art algorithms for the SCP.

Leguizamón & Michalewicz [36] proposed the first ACO applications to the multiple knapsack and to the maximum independent set problems, which were, however, not competitive with the state of the art. Currently, the best performing ACO algorithm for the multiple knapsack problem is due to Ke et al. [37]. Levine & Ducatelle [40] adapted $\mathcal{MM}$AS to the well-known bin packing problem and compared its performance with the hybrid grouping genetic algorithm [111], and with Martello & Toth's reduction method [112]. The $\mathcal{MM}$AS algorithm outperformed both, obtaining better solutions in much shorter time. Solnon & Fenet [45] carried out a comprehensive study for the maximum clique problem. Their conclusion was that ACO combined with appropriate local search can match the quality of state-of-the-art heuristics. Blesa & Blum [49] applied ACO to the problem of finding edge-disjoint paths in networks, and found the performance of the proposed ACO superior in terms of both solution quality and computation time when compared with a multi-start greedy algorithm. Another interesting application is the work of Sivagaminathan & Ramakrishnan [50], which shows how ACO may be hybridized with neural networks for optimizing feature selection in multivariate analysis.

Cordone & Maffioli [39] introduced the weight constrained graph tree partition problem, and tested different variants of ACS with and without local search. Blum & Blesa [43] tackled the edge-weighted k-cardinality tree problem (or k-minimum spanning tree), where the goal is to find a tree over a graph with exactly k edges minimizing the sum of the weights. They compared a $\mathcal{MM}$AS variant, tabu search and an evolutionary algorithm. Their results showed that none of the approaches was superior to the others in all instance classes tested, and $\mathcal{MM}$AS was better suited for instances where the value of k was much smaller than the number of vertices.

A subset problem closely related to the capacitated VRP (CVRP) is the capacitated minimum spanning tree problem (CMST), which has been effectively tackled by a hybrid ACO algorithm [44] based on a previous ACO algorithm for the CVRP [10]. More recently, Hernández & Blum [51] considered the minimization of power consumption when multicasting in static wireless ad-hoc networks. This problem can be stated as an $\mathcal{NP}$-hard combinatorial problem, where the goal is to find a directed tree over the network of nodes. Their proposed ACO algorithm outperforms existing algorithms for several variants of this problem.

Finally, a class of problems for which ACO has recently shown competitive results is that of multi-level lot-sizing with [46, 48] and without capacity constraints [47]. In these problems, a subset of items is scheduled for production at each time interval, and the goal is to minimize the cost of producing the items, taking into account several constraints and relations between the items.

1.4 Assignment and layout problems

In assignment problems, a set of items has to be assigned to a given number of resources subject to some constraints. Probably, the most widely studied example is the quadratic assignment problem (QAP), which was among the first problems tackled by ACO algorithms [5, 52, 53]. Various high-performing ACO algorithms for the QAP have followed this initial work. Among them is the approximate nondeterministic tree search (ANTS) algorithm by Maniezzo [113] a combination of ACO with tree search techniques involving the usage of lower bounds to rate solution components and to prune extensions of partial solutions. The computational results of ANTS on the QAP were very promising. Another high-performing ACO algorithm is the $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) proposed by Stützle & Hoos [8], which is among the best algorithms available for large, structured instances of the QAP.

The ANTS algorithm has also been applied to the frequency assignment problem (FAP), in which frequencies have to be assigned to links and there are constraints on the minimum distance between the frequencies assigned to each pair of links. ANTS showed good performance on some classes of FAP instances in comparison with other approaches [56]. Other applications of ACO to assignment problems include university course timetabling [59, 60] and graph coloring [54]. The work of Solnon [57, 58] applies ACO algorithms to the general class of constraint satisfaction problems (CSPs); in fact, decision variants of problems such as graph coloring and frequency assignment can be seen as cases of CSPs. Within this class, Pinto et al. [62] studied the application of ACO to regular and dynamic MAX-SAT problems.

Another notable example is the generalized assignment problem, where a set of tasks have to be assigned to a set of agents with a limited total capacity, minimizing the total assignment cost of tasks to agents. The $\mathcal{MM}$AS algorithm proposed by Lourenço & Serra [55] was, at the time of its publication, close to the state-of-the-art for this problem. More recently, Doerner et al. [61] tackled a real-world problem related to ambulance locations in Austria by means of an ACO algorithm; and Blum [64] has shown that the hybrid between beam search and ACO, Beam-ACO, is a state-of-the-art algorithm for simple assembly line balancing (SALB-1). In Section 2.6, we mention an industrial application of ACO to assembly line balancing. Finally, Silva et al. [65] have used ACO for a complex supply chain management that combines aspects of the generalized assignment, scheduling, and vehicle routing problems.

1.5 Machine learning problems

Diverse problems in the field of Machine Learning have been tackled by means of ACO algorithms. Notable examples are the work of Parpinelli et al. [69] and Martens et al. [70] on applying ACO to the problem of learning classification rules. This work was later extended by Otero et al. [71] in order to handle continuous attributes. De Campos et al. [66, 67] adapted Ant Colony System for the problem of learning the structure of Bayesian networks, and Pinto et al. [68] have recently extended this work. Finally, the work of Socha & Blum [89] for training neural networks by means of ACO is also an example of the application of ACO algorithms for continuous problems.

1.6 Bioinformatics problems

Computer applications to molecular biology (bioinformatics) have originated many $\mathcal{NP}$-hard combinatorial optimization problems. We include in this section general problems that have attracted considerable interest due to their applications to bioinformatics. This is the case of the shortest common supersequence problem (SCSP), which is a well-known $\mathcal{NP}$-hard problem with applications in DNA analysis. Michel & Middendorf [72, 73] proposed an ACO algorithm for the SCSP, obtaining state-of-the-art results, in particular, for structured instances that are typically found in real-world applications.

An important problem in bioinformatics is protein folding, that is, the prediction of a protein's structure based on its sequence of amino acids. A simplified model for protein folding is the two-dimensional hydrophobic-polar protein folding problem [114]. Shmygelska & Hoos [74] have successfully applied ACO to this problem and its three-dimensional variant. The performance of the resulting ACO algorithm is comparable to the best existing specialized algorithms for these problems.

Interesting is also the work of Blum et al. [78], where they propose a *multilevel framework* based on ACO for the problem of DNA sequencing by hybridization. An earlier proposal of multilevel ACO frameworks is due to Korošec et al. [115]. Multilevel techniques [116, 117] solve a hierarchy of successively smaller versions of the original problem instance. The solutions obtained at the lowest level of the hierarchy are transformed into solutions for the next higher level, and improved by an optimization algorithm, such as an ACO algorithm.

Other problems in bioinformatics have been successfully tackled by means of ACO algorithms: Korb et al. [75, 76] considered the flexible protein-ligand docking problem, for which the proposed ACO algorithm reaches state-of-the-art performance, and Benedettini et al. [79] recently studied the problem of haplotype inference under pure parsimony. ACO algorithms are sometimes hybridized with Machine Learning techniques. An example is the recent work of Ressom et al. [77] on a selection problem in biomarker identification, which combines ACO with support vector machines (SVM).

2 Applications to problems with non-standard features

We review in this section applications of ACO algorithms to problems having additional characteristics such as multiple objective functions, time-varying data and stochastic information about objective values or constraints. In addition, we mention applications of ACO to network routing and continuous optimization problems.

2.1 Multi-objective optimization

In many real-world problems, candidate solutions are evaluated according to multiple, often conflicting objectives. Sometimes the importance of each objective can be exactly weighted, and hence objectives can be combined into a single scalar value by using, for example, a weighted sum. This is the approach used by Doerner et al. [118] for a bi-objective transportation problem. In other cases, objectives can be ordered by their relative importance in a lexicographical manner. Gambardella et al. [12] proposed a two-colony ACS algorithm for the vehicle routing problem with time windows, where the first colony improves the primary objective and the second colony tries to improve the secondary objective while not worsening the primary one.

When there is no *a priori* knowledge about the relative importance of objectives, the goal usually becomes to approximate the set of Pareto-optimal solutions—a solution is Pareto optimal if no other solution is better or equal for all objectives and strictly better in at least one objective. Iredi et al. [80] were among the first to discuss various alternatives for extending ACO to multi-objective problems in terms of Pareto-optimality. They also tested a few of the proposed variants on a bi-objective scheduling problem. Another early work is the application of ACO to multi-objective portfolio problems by Doerner et al. [81, 82]. Later studies have proposed and tested various combinations of alternative ACO algorithms for multi-objective variants of the quadratic assignment problem [83, 84], the knapsack problem [85], activity crashing [87] and the bi-objective orienteering problem [88]. García-Martínez et al. [86] reviewed existing multi-objective ACO algorithms and carried out an experimental evaluation of several ACO variants using the bi-criteria TSP as a case study. Angus & Woodward [119] give another detailed overview of available multi-objective ACO algorithms.

2.2 Stochastic optimization problems

In stochastic optimization problems, data are not known exactly before generating a solution. Rather, because of uncertainty, noise, approximation or other factors, what is available is stochastic information on the objective function value(s), on the decision variable values, or on the constraint boundaries.

The first application of ACO algorithms to stochastic problems was to the probabilistic TSP (PTSP). In the PTSP, each city has associated a probability of requiring a visit, and the goal is to find an *a priori* tour of minimal expected length over all cities. Bianchi et al. [91] and Bianchi & Gambardella [92] proposed an adaptation of ACS for the PTSP. Very recently, this algorithm was improved by Balaprakash et al. [93], resulting in a state-of-the-art algorithm for the PTSP. Other applications of ACO to stochastic problems include vehicle routing problems with uncertain demands [94], and the selection of optimal screening policies for diabetic retinopathy [95]. The latter approach builds on the S-ACO algorithm proposed earlier by Gutjahr [120].

2.3 Dynamic optimization problems

Dynamic optimization problems are those whose characteristics change while being solved. ACO algorithms have been applied to such versions of classical $\mathcal{NP}$-hard problems. Notable examples are applications to dynamic versions of the TSP, where the distances between cities may change or where cities may appear or disappear [98, 99, 100, 101]. More recently, Montemanni et al. [102] and Donati et al. [103] discuss applications of ACS to dynamic vehicle routing problems, reporting good results on both artificial and real-world instances of the problem. Other notable examples of dynamic problems are routing problems in communication networks, which are discussed in the following section.

2.4 Communication network problems

Some system properties in telecommunication networks, such as the availability of links or the cost of traversing links, are time-varying. The application of ACO algorithms to routing problems in such networks is among the main success stories in ACO. One of the first applications by Schoonderwoerd et al. [121] concerned routing in circuit-switched networks, such as classical telephone networks. The proposed algorithm, called ABC, was demonstrated on a simulated version of the British Telecom network.

A very successful application of ACO to dynamic network routing is the AntNet algorithm, proposed by Di Caro & Dorigo [96, 122]. AntNet was applied to routing in packet-switched networks, such as the Internet. Experimental studies compared AntNet with many state-of-the-art algorithms on a large set of benchmark problems under a variety of traffic conditions [96]. AntNet proved to be very robust against varying traffic conditions and parameter settings, and it always outperformed competing approaches.

Several other routing algorithms based on ACO have been proposed for a variety of wired network scenarios [123, 124]. More recent applications of these strategies deal with the challenging class of mobile ad hoc networks (MANETs). Due to the specific characteristics of MANETs (very high dynamics, link asymmetry), the straightforward application of the ACO algorithms developed for wired networks has proven unsuccessful [125]. Nonetheless, an extension of AntNet that is competitive with state-of-the-art routing algorithms for MANETs has been proposed by Ducatelle et al. [97, 126]. For recent, in-depth reviews of applications of ACO to dynamic network routing problems, we refer to Farooq & Di Caro [127] and Ducatelle et al. [128].

2.5 Continuous optimization problems

Continuous optimization problems arise in a large number of engineering applications. Their crucial difference from combinatorial problems, which were the exclusive application field of ACO in the early research efforts, is that decision variables in such problems have a continuous, real-valued domain. Recently, various proposals have been made of how to handle continuous decision variables within the ACO framework [129, 130, 131]. In the continuous ACO algorithm proposed by Socha & Dorigo [132], probability density functions, explicitly represented by Gaussian kernel functions, correspond to the pheromone models. Extensions of this approach also exist for mixed-variable—continuous and discrete–problems [133]. A notable application of ACO algorithms for continuous optimization is the training of feed-forward neural networks [89]. Interestingly, there exist also successful applications of ACO to continuous problems that discretize the real-valued domain of the variables. An example is the PLANTS algorithm for the protein–ligand docking problem [76], which combines a discrete ACO algorithm with a local search that works on the continuous domain of the variables.

2.6 Industrial applications

While most research is done on academic applications, commercial companies have started to use ACO algorithms for real-world applications [11]. The company AntOptima (`www.antoptima.com`) develops and markets ACO-based solution methods for tackling industrial vehicle routing problems. Features common to real-world applications are time-varying data, multiple objectives or the availability of stochastic information about events or data. Moreover, engineering problems often do not have a mathematical formulation in the traditional sense. Rather, algorithms have to rely on an external *simulator* to evaluate the quality and feasibility of candidate solutions. Examples of applications of ACO relying on simulation are the design [134] and operation [135] of water distribution networks. Other interesting real-world applications are those of Gravel, Price & Gagné [28], who applied ACO to an industrial scheduling problem in an aluminium casting center, and by Bautista & Pereira [63, 136, 137], who successfully applied ACO to solve an assembly line balancing problem for a bike line assembly.

3 Conclusions

Nowadays, ACO is a well established metaheuristic with hundreds of successful implementations applied to a wide range of optimization problems. Several of these implementations have shown to be, at least at the time of their publication, the state of the art for the respective problems tackled, including problems such as vehicle routing, sequential ordering, quadratic assignment, assembly line balancing, open-shop scheduling, and various others. Applications of ACO to dynamic routing problems in telecommunication networks have been particularly successful, probably because several algorithm characteristics match well the features of the applications.

By analysing the many available ACO implementations, one can identify ingredients necessary for the successful application of ACO. Firstly, an effective mechanism for iteratively constructing solutions must be available. Ideally, this construction mechanism exploits problem-specific knowledge by using appropriate heuristic information. Secondly, the best performing ACO algorithms have specialized features that allow to carefully control the balance between the exploration of new solutions and the intensification of the search around the best solutions. Such control mechanisms are offered by advanced ACO algorithms such as ACS or $\mathcal{MM}$AS. In fact, the original Ant System has been abandoned by now in favor of better performing variants. Thirdly, the usage of local search algorithms for improving the solutions constructed by the ants is very successful in practice. Finally, the integration of other techniques such as constraint programming, tree search techniques or multi-level frameworks often yields a further improvement in performance or increases the robustness of the algorithms.

Further information on ACO and related topics can be obtained by subscribing to the moderated mailing list `aco-list`, and by visiting the Ant Colony Optimization web page (`www.aco-metaheuristic.org`).

Acknowledgments. This work was supported by the META-X project, an *Action de Recherche Concertée* funded by the Scientific Research Directorate of the French Community of Belgium. Marco Dorigo and Thomas Stützle acknowledge support from the Belgian F.R.S.-FNRS, of which they are a Research Director and a Research Associate, respectively.

References

[1] Dorigo M, Di Caro G. The Ant Colony Optimization meta-heuristic. In: Corne D, Dorigo M, Glover F, editors. New Ideas in Optimization. London, UK: McGraw Hill; 1999. p. 11–32.

[2] Dorigo M, Di Caro G, Gambardella LM. Ant algorithms for discrete optimization. Artificial Life. 1999;5(2):137–172.

[3] Dorigo M, Stützle T. Ant Colony Optimization. Cambridge, MA: MIT Press; 2004. 305 p.

[4] Dorigo M, Maniezzo V, Colorni A. Positive feedback as a search strategy. Dipartimento di Elettronica, Politecnico di Milano, Italy; 1991. 91-016.

[5] Dorigo M, Maniezzo V, Colorni A. Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics – Part B. 1996;26(1):29–41.

[6] Dorigo M, Gambardella LM. Ant Colony System: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation. 1997;1(1):53–66.

[7] Stützle T, Hoos HH. The $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System and local search for the traveling salesman problem. In: Bäck T, Michalewicz Z, Yao X, editors. Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97). IEEE Press, Piscataway, NJ; 1997. p. 309–314.

[8] Stützle T, Hoos HH. $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System. Future Generation Computer Systems. 2000;16(8):889–914.

[9] Bullnheimer B, Hartl RF, Strauss C. An improved ant system algorithm for the vehicle routing problem. Annals of Operations Research. 1999;89:319–328.

[10] Reimann M, Doerner KF, Hartl RF. D-ants: Savings based ants divide and conquer the vehicle routing problems. Computers & Operations Research. 2004;31(4):563–591.

[11] Rizzoli AE, Montemanni R, Lucibello E, Gambardella LM. Ant colony optimization for real-world vehicle routing problems. From theory to applications. Swarm Intelligence. 2007;1(2):135–151.

[12] Gambardella LM, Taillard ÉD, Agazzi G. MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In: Corne D, Dorigo M, Glover F, editors. New Ideas in Optimization. McGraw Hill, London, UK; 1999. p. 63–76.

[13] Favaretto D, Moretti E, Pellegrini P. Ant colony system for a VRP with multiple time windows and multiple visits. Journal of Interdisciplinary Mathematics. 2007;10(2):263–284.

[14] Doerner KF, Fuellerer G, Gronalt M, Hartl RF, Iori M. Metaheuristics for the vehicle routing problem with loading constraints. Networks. 2006;49(4):294–307.

[15] Fuellerer G, Doerner KF, Hartl RF, Iori M. Ant colony optimization for the two-dimensional loading vehicle routing problem. Computers & Operations Research. 2009;36(3):655–673.

[16] Fuellerer G, Doerner KF, Hartl RF, Iori M. Metaheuristics for vehicle routing problems with three-dimensional loading constraints. European Journal of Operational Research. 2009;201(3):751–759.

[17] Ke L, Archetti C, Feng Z. Ants can solve the team orienteering problem. Computers and Industrial Engineering. 2008;54(3):648–665.

[18] Gambardella LM, Dorigo M. Ant Colony System hybridized with a new local search for the sequential ordering problem. INFORMS Journal on Computing. 2000;12(3):237–255.

[19] López-Ibáñez M, Blum C. Beam-ACO for the travelling salesman problem with time windows. Computers & Operations Research. 2010;In Press.

[20] den Besten ML, Stützle T, Dorigo M. Ant colony optimization for the total weighted tardiness problem. In: Schoenauer M, et al., editors. Proceedings of PPSN-VI, Sixth International Conference on Parallel Problem Solving from Nature. vol. 1917 of Lecture Notes in Computer Science. Heidelberg: Springer; 2000. p. 611–620.

[21] Merkle D, Middendorf M. An ant algorithm with a new pheromone evaluation rule for total tardiness problems. In: Cagnoni S, et al., editors. Real-World Applications of Evolutionary Computing. vol. 1803 of Lecture Notes in Computer Science. Heidelberg: Springer; 2000. p. 287–296.

[22] Merkle D, Middendorf M. Ant colony optimization with global pheromone evaluation for scheduling a single machine. Applied Intelligence. 2003;18(1):105–111.

[23] Meyer B, Ernst AT. Integrating ACO and constraint propagation. In: Dorigo M, et al., editors. Ant Colony Optimization and Swarm Intelligence: 4th International Workshop, ANTS 2004. vol. 3172 of Lecture Notes in Computer Science. Heidelberg: Springer; 2004. p. 166–177.

[24] Liao CJ, Juan HC. An ant colony optimization for single-machine tardiness scheduling with sequence-dependent setups. Computers & Operations Research. 2007;34(7):1899–1909.

[25] Meyer B. Hybrids of constructive metaheuristics and constraint programming. In: Blum C, Blesa MJ, Roli A, Sampels M, editors. Hybrid Metaheuristics—An Emergent Approach to Optimization. vol. 117 of Studies in Computational Intelligence. Berlin, Germany: Springer; 2008. p. 85–116.

[26] Stützle T. An ant approach to the flow shop problem. In: Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98). vol. 3. Verlag Mainz, Aachen, Germany; 1998. p. 1560–1564.

[27] Rajendran C, Ziegler H. Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. European Journal of Operational Research. 2004;155(2):426–438.

[28] Gravel M, Price WL, Gagné C. Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic. European Journal of Operational Research. 2002;143:218–229.

[29] Merkle D, Middendorf M, Schmeck H. Ant colony optimization for resource-constrained project scheduling. IEEE Transactions on Evolutionary Computation. 2002;6(4):333–346.

[30] Blum C. Theoretical and Practical Aspects of Ant Colony Optimization [Ph.D. Thesis]. IRIDIA, Université Libre de Bruxelles, Brussels, Belgium; 2004.

[31] Huang KL, Liao CJ. Ant colony optimization combined with taboo search for the job shop scheduling problem. Computers & Operations Research. 2008;35(4):1030–1046.

[32] Blum C. Beam-ACO—Hybridizing ant colony optimization with beam search: An application to open shop scheduling. Computers & Operations Research. 2005;32(6):1565–1591.

[33] Khichane M, Albert P, Solnon C. Integration of ACO in a constraint programming language. In: Dorigo M, et al., editors. Ant Colony Optimization and Swarm Intelligence: 6th International Conference, ANTS 2008. vol. 5217 of Lecture Notes in Computer Science. Heidelberg: Springer; 2008. p. 84–95.

[34] Solnon C. Combining two pheromone structures for solving the car sequencing problem with ant colony optimization. European Journal of Operational Research. 2008;191(3):1043–1055.

[35] Morin S, Gagné C, Gravel M. Ant colony optimization with a specialized pheromone trail for the car-sequencing problem. European Journal of Operational Research. 2009;197(3):1185–1191.

[36] Leguizamón G, Michalewicz Z. A new version of Ant System for subset problems. In: Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99). Piscataway, NJ: IEEE Press; 1999. p. 1459–1464.

[37] Ke L, Feng Z, Ren Z, Wei X. An ant colony optimization approach for the multidimensional knapsack problem. Journal of Heuristics. To appear;.

[38] Liang YC, Smith AE. An Ant System approach to redundancy allocation. In: Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99). Piscataway, NJ: IEEE Press; 1999. p. 1478–1484.

[39] Cordone R, Maffioli F. Coloured Ant System and local search to design local telecommunication networks. In: Boers EJW, et al., editors. Applications of Evolutionary Computing, Proceedings of EvoWorkshops 2001. vol. 2037 of Lecture Notes in Computer Science. Heidelberg: Springer; 2001. p. 60–69.

[40] Levine J, Ducatelle F. Ant colony optimisation and local search for bin packing and cutting stock problems. Journal of the Operational Research Society. 2003;55(7):705–716.

[41] Lessing L, Dumitrescu I, Stützle T. A comparison between ACO algorithms for the set covering problem. In: Dorigo M, et al., editors. Ant Colony Optimization and Swarm Intelligence: 4th International Workshop, ANTS 2004. vol. 3172 of Lecture Notes in Computer Science. Heidelberg: Springer; 2004. p. 1–12.

[42] Gandibleux X, Delorme X, T'Kindt V. An ant colony optimisation algorithm for the set packing problem. In: Dorigo M, et al., editors. Ant Colony Optimization and Swarm Intelligence: 4th International Workshop, ANTS 2004. vol. 3172 of Lecture Notes in Computer Science. Heidelberg: Springer; 2004. p. 49–60.

[43] Blum C, Blesa MJ. New metaheuristic approaches for the edge-weighted k-cardinality tree problem. Computers & Operations Research. 2005;32(6):1355–1377.

[44] Reimann M, Laumanns M. Savings based ant colony optimization for the capacitated minimum spanning tree problem. Computers & Operations Research. 2006;33(6):1794–1822.

[45] Solnon C, Fenet S. A study of ACO capabilities for solving the maximum clique problem. Journal of Heuristics. 2006;12(3):155–180.

[46] Pitakaso R, Almeder C, Doerner KF, Hartl RF. Combining exact and population-based methods for the constrained multilevel lot sizing problem. International Journal of Production Research. 2006;44(22):4755–4771.

[47] Pitakaso R, Almeder C, Doerner KF, Hartl RF. A $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System for unconstrained multi-level lot-sizing problems. Computers & Operations Research. 2007;34(9):2533–2552.

[48] Almeder C. A hybrid optimization approach for multi-level capacitated lot-sizing problems. European Journal of Operational Research. 2010;200(2):599–606.

[49] Blesa MJ, Blum C. Finding edge-disjoint paths in networks by means of artificial ant colonies. Journal of Mathematical Modelling and Algorithms. 2007;6(3):361–391.

[50] Sivagaminathan RK, Ramakrishnan S. A hybrid approach for feature subset selection using neural networks and ant colony optimization. Expert Systems with Applications. 2007;33(1):49–60.

[51] Hernández H, Blum C. Ant colony optimization for multicasting in static wireless ad-hoc networks. Swarm Intelligence. 2009;3(2):125–148.

[52] Maniezzo V, Colorni A, Dorigo M. The Ant System applied to the quadratic assignment problem. IRIDIA, Université Libre de Bruxelles, Belgium; 1994. IRIDIA/94-28.

[53] Maniezzo V, Colorni A. The Ant System applied to the quadratic assignment problem. IEEE Transactions on Data and Knowledge Engineering. 1999;11(5):769–778.

[54] Costa D, Hertz A. Ants can colour graphs. Journal of the Operational Research Society. 1997;48:295–305.

[55] Lourenço H, Serra D. Adaptive approach heuristics for the generalized assignment problem. Universitat Pompeu Fabra, Department of Economics and Management, Barcelona, Spain; 1998. Economic Working Papers Series No.304.

[56] Maniezzo V, Carbonaro A. An ANTS heuristic for the frequency assignment problem. Future Generation Computer Systems. 2000;16(8):927–935.

[57] Solnon C. Solving permutation constraint satisfaction problems with artificial ants. In: Horn W, editor. Proceedings of the 14th European Conference on Artificial Intelligence. IOS Press, Amsterdam, The Netherlands; 2000. p. 118–122.

[58] Solnon C. Ants can solve constraint satisfaction problems. IEEE Transactions on Evolutionary Computation. 2002;6(4):347–357.

[59] Socha K, Knowles J, Sampels M. A $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System for the university course timetabling problem. In: Dorigo M, et al., editors. Ant Algorithms: Third International Workshop, ANTS 2002. vol. 2463 of Lecture Notes in Computer Science. Heidelberg: Springer; 2002. p. 1–13.

[60] Socha K, Sampels M, Manfrin M. Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. In: Raidl GR, et al., editors. Applications of Evolutionary Computing, Proceedings of EvoWorkshops 2003. vol. 2611 of Lecture Notes in Computer Science. Heidelberg: Springer; 2003. p. 334–345.

[61] Doerner KF, Gutjahr WJ, Hartl RF, Karall M, Reimann M. Heuristic solution of an extended double-coverage ambulance location problem for austria. Central European Journal for Operations Research. 2005;13(4):325–340.

[62] Pinto P, Runkler T, Sousa J. Ant colony optimization and its application to regular and dynamic MAX-SAT problems. In: Advances in Biologically Inspired Information Systems. vol. 69 of Studies in Computational Intelligence. Berlin, Germany: Springer-Verlag; 2007. p. 285–304.

[63] Bautista J, Pereira J. Ant algorithms for a time and space constrained assembly line balancing problem. European Journal of Operational Research. 2007;177(3):2016–2032.

[64] Blum C. Beam-ACO for simple assembly line balancing. INFORMS Journal on Computing. 2008;20(4):618–627.

[65] Silva CA, Sousa JMC, Runkler TA, da Costa JMGS. Distributed supply chain management using ant colony optimization. European Journal of Operational Research. 2009;199(2):349–358.

[66] de Campos LM, Fernández-Luna JM, Gámez JA, Puerta JM. Ant colony optimization for learning Bayesian networks. International Journal of Approximate Reasoning. 2002;31(3):291–311.

[67] de Campos LM, Gamez JA, Puerta JM. Learning Bayesian networks by ant colony optimisation: Searching in the space of orderings. Mathware & Soft Computing. 2002;9(2–3):251–268.

[68] Pinto PC, Nägele A, Dejori M, Runkler TA, Sousa JMC. Using a local discovery ant algorithm for bayesian network structure learning. IEEE Transactions on Evolutionary Computation. 2009;13(4):767–779.

[69] Parpinelli RS, Lopes HS, Freitas AA. Data mining with an ant colony optimization algorithm. IEEE Transactions on Evolutionary Computation. 2002;6(4):321–332.

[70] Martens D, De Backer M, Haesen R, Vanthienen J, Snoeck M, Baesens B. Classification with ant colony optimization. IEEE Transactions on Evolutionary Computation. 2007;11(5):651–665.

[71] Otero FEB, Freitas AA, Johnson CG. cAnt-Miner: An ant colony classification algorithm to cope with continuous attributes. In: Dorigo M, et al., editors. Ant Colony Optimization and Swarm Intelligence: 6th International Conference, ANTS 2008. vol. 5217 of Lecture Notes in Computer Science. Heidelberg: Springer; 2008. p. 48–59.

[72] Michel R, Middendorf M. An island model based Ant System with lookahead for the shortest supersequence problem. In: Eiben AE, et al., editors. Proceedings of PPSN-V, Fifth International Conference on Parallel Problem Solving from Nature. vol. 1498 of Lecture Notes in Computer Science. Heidelberg: Springer; 1998. p. 692–701.

[73] Michel R, Middendorf M. An ACO algorithm for the shortest supersequence problem. In: Corne D, Dorigo M, Glover F, editors. New Ideas in Optimization. McGraw Hill, London, UK; 1999. p. 51–61.

[74] Shmygelska A, Hoos HH. An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. BMC Bioinformatics. 2005;6:30.

[75] Korb O, Stützle T, Exner TE. PLANTS: Application of ant colony optimization to structure-based drug design. In: Dorigo M, et al., editors. Ant Colony Optimization and Swarm Intelligence: 5th International Workshop, ANTS 2006. vol. 4150 of Lecture Notes in Computer Science. Heidelberg: Springer; 2006. p. 247–258.

[76] Korb O, Stützle T, Exner TE. An ant colony optimization approach to flexible protein-ligand docking. Swarm Intelligence. 2007;1(2):115–134.

[77] Ressom HW, Varghese RS, Drake SK, Hortin GL, Abdel-Hamid M, Loffredo CA, et al. Peak selection from MALDI-TOF mass spectra using ant colony optimization. Bioinformatics. 2007;23(5):619–626.

[78] Blum C, Yábar Vallès M, Blesa MJ. An ant colony optimization algorithm for DNA sequencing by hybridization. Computers & Operations Research. 2008;35(11):3620–3635.

[79] Benedettini S, Roli A, Di Gaspero L. Two-level ACO for haplotype inference under pure parsimony. In: Dorigo M, et al., editors. Ant Colony Optimization and Swarm Intelligence: 6th International Conference, ANTS 2008. vol. 5217 of Lecture Notes in Computer Science. Heidelberg: Springer; 2008. p. 179–190.

[80] Iredi S, Merkle D, Middendorf M. Bi-criterion optimization with multi colony ant algorithms. In: Zitzler E, Deb K, Thiele L, Coello CAC, Corne D, editors. First International Conference on Evolutionary Multi-Criterion Optimization, (EMO'01). vol. 1993 of Lecture Notes in Computer Science. Heidelberg: Springer; 2001. p. 359–372.

[81] Doerner KF, Gutjahr WJ, Hartl RF, Strauss C, Stummer C. Ant colony optimization in multiobjective portfolio selection. In: Proceedings of the Fourth Metaheuristics International Conference; 2001. p. 243–248.

[82] Doerner KF, Gutjahr WJ, Hartl RF, Strauss C, Stummer C. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. Annals of Operations Research. 2004;131:79–99.

[83] López-Ibáñez M, Paquete L, Stützle T. On the design of ACO for the biobjective quadratic assignment problem. In: Dorigo M, et al., editors. Ant Colony Optimization and Swarm Intelligence: 4th International Workshop, ANTS 2004. vol. 3172 of Lecture Notes in Computer Science. Heidelberg: Springer; 2004. p. 214–225.

[84] López-Ibáñez M, Paquete L, Stützle T. Hybrid population-based algorithms for the bi-objective quadratic assignment problem. Journal of Mathematical Modelling and Algorithms. 2006;5(1):111–137.

[85] Alaya I, Solnon C, Ghédira K. Ant colony optimization for multi-objective optimization problems. In: 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007). vol. 1. Los Alamitos, CA: IEEE Computer Society Press; 2007. p. 450–457.

[86] García-Martínez C, Cordón O, Herrera F. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. European Journal of Operational Research. 2007;180(1):116–148.

[87] Doerner KF, Gutjahr WJ, Hartl RF, Strauss C, Stummer C. Nature-inspired metaheuristics in multiobjective activity crashing. Omega. 2008;36(6):1019–1037.

[88] Schilde M, Doerner KF, Hartl RF, Kiechle G. Metaheuristics for the bi-objective orienteering problem. Swarm Intelligence. 2009;3(3):179–201.

[89] Socha K, Blum C. An ant colony optimization algorithm for continuous optimization: An application to feed-forward neural network training. Neural Computing & Applications. 2007;16(3):235–247.

[90] Socha K, Dorigo M. Ant colony optimization for continuous domains. European Journal of Operational Research. 2008;185(3):1155–1173.

[91] Bianchi L, Gambardella LM, Dorigo M. An ant colony optimization approach to the probabilistic traveling salesman problem. In: Merelo JJ, et al., editors. Proceedings of PPSN-VII, Seventh International Conference on Parallel Problem Solving from Nature. vol. 2439 of Lecture Notes in Computer Science. Heidelberg: Springer; 2002. p. 883–892.

[92] Bianchi L, Gambardella LM. Ant colony optimization and local search based on exact and estimated objective values for the probabilistic traveling salesman problem. USI-SUPSI, Manno, Switzerland: IDSIA; 2007. IDSIA-06-07.

[93] Balaprakash P, Birattari M, Stützle T, Yuan Z, Dorigo M. Estimation-based ant colony optimization algorithms for the probabilistic travelling salesman problem. Swarm Intelligence. 2009;3(3):223–242.

[94] Bianchi L, Birattari M, Manfrin M, Mastrolilli M, Paquete L, Rossi-Doria O, et al. Hybrid metaheuristics for the vehicle routing problem with stochastic demands. Journal of Mathematical Modelling and Algorithms. 2006;5(1):91–110.

[95] Brailsford SC, Gutjahr WJ, Rauner MS, Zeppelzauer W. Combined discrete-event simulation and ant colony optimisation approach for selecting optimal screening policies for diabetic retinopathy. Computational Management Science. 2006;4(1):59–83.

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

[97] Di Caro G, Ducatelle F, Gambardella LM. AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. European Transactions on Telecommunications. 2005;16(5):443–455.

[98] Guntsch M, Middendorf M. Pheromone modification strategies for ant algorithms applied to dynamic TSP. In: Boers EJW, et al., editors. Applications of Evolutionary Computing, Proceedings of EvoWorkshops 2001. vol. 2037 of Lecture Notes in Computer Science. Heidelberg: Springer; 2001. p. 213–222.

[99] Guntsch M, Middendorf M. A population based approach for ACO. In: Cagnoni S, et al., editors. Applications of Evolutionary Computing, Proceedings of EvoWorkshops 2002. vol. 2279 of Lecture Notes in Computer Science. Heidelberg: Springer; 2002. p. 71–80.

[100] Eyckelhof CJ, Snoek M. Ant systems for a dynamic TSP: Ants caught in a traffic jam. In: Dorigo M, et al., editors. Ant Algorithms: Third International Workshop, ANTS 2002. vol. 2463 of Lecture Notes in Computer Science. Heidelberg: Springer; 2002. p. 88–99.

[101] Sammoud O, Solnon C, Ghédira K. A new ACO approach for solving dynamic problems. In: 9th International Conference on Artificial Evolution (EA'09). Lecture Notes in Computer Science. Heidelberg: Springer; To appear. .

[102] Montemanni R, Gambardella LM, Rizzoli AE, Donati AV. Ant colony system for a dynamic vehicle routing problem. Journal of Combinatorial Optimization. 2005;10:327–343.

[103] Donati AV, Montemanni R, Casagrande N, Rizzoli AE, Gambardella LM. Time dependent vehicle routing problem with a multi ant colony system. European Journal of Operational Research. 2008;185(3):1174–1191.

[104] Applegate D, Bixby RE, Chvátal V, Cook WJ. The Traveling Salesman Problem: A Computational Study. Princeton University Press, Princeton, NJ; 2006.

[105] Lawler EL, Lenstra JK, Kan AHGR, Shmoys DB. The Travelling Salesman Problem. John Wiley & Sons, Chichester, UK; 1985.

[106] Dorigo M. Optimization, Learning and Natural Algorithms [PhD thesis]. Dipartimento di Elettronica, Politecnico di Milano, Italy; 1992. In Italian.

[107] Dorigo M, Maniezzo V, Colorni A. The Ant System: An autocatalytic optimizing process. Dipartimento di Elettronica, Politecnico di Milano, Italy; 1991. 91-016 Revised.

[108] Bauer A, Bullnheimer B, Hartl RF, Strauss C. An ant colony optimization approach for the single machine total tardiness problem. In: Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99). Piscataway, NJ: IEEE Press; 1999. p. 1445–1450.

[109] Merkle D, Middendorf M, Schmeck H. Ant colony optimization for resource-constrained project scheduling. In: Whitley D, et al., editors. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000). Morgan Kaufmann Publishers, San Francisco, CA; 2000. p. 893–900.

[110] Yagiura M, Kishida M, Ibaraki T. A 3-flip neighborhood local search for the set covering problem. European Journal of Operational Research. 2006;172:472–499.

[111] Falkenauer E. A hybrid grouping genetic algorithm for bin packing. Journal of Heuristics. 1996;2:5–30.

[112] Martello S, Toth P. Knapsack Problems, Algorithms and Computer Implementations. John Wiley & Sons, Chichester, UK; 1990.

[113] Maniezzo V. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. INFORMS Journal on Computing. 1999;11(4):358–369.

[114] Lau KF, Dill KA. A lattice statistical mechanics model of the conformation and sequence space of proteins. Macromolecules. 1989;22:3986–3997.

[115] Korošec P, Šilc J, Robič B. Solving the mesh-partitioning problem with an ant-colony algorithm. Parallel Computing. 2004;30:785–801.

[116] Brandt A. Multilevel computations: Review and recent developments. In: McCormick SF, editor. Multigrid Methods: Theory, Applications, and Supercomputing, Proceedings of the 3rd Copper Mountain Conference on Multigrid Methods. vol. 110 of Lecture Notes in Pure and Applied Mathematics. Marcel Dekker, New York; 1988. p. 35–62.

[117] Walshaw C, Cross M. Mesh partitioning: A multilevel balancing and refinement algorithm. SIAM Journal on Scientific Computing. 2000;22(1):63–80.

[118] Doerner KF, Hartl RF, Reimann M. Are CompetAnts more competent for problem solving? The case of a multiple objective transportation problem. Central European Journal for Operations Research and Economics. 2003;11(2):115–141.

[119] Angus D, Woodward C. Multiple objective ant colony optimization. Swarm Intelligence. 2009;3(1):69–85.

[120] Gutjahr WJ. S-ACO: An ant-based approach to combinatorial optimization under uncertainty. In: Dorigo M, et al., editors. Ant Colony Optimization and Swarm Intelligence: 4th International Workshop, ANTS 2004. vol. 3172 of Lecture Notes in Computer Science. Heidelberg: Springer; 2004. p. 238–249.

[121] Schoonderwoerd R, Holland O, Bruten J, Rothkrantz L. Ant-based load balancing in telecommunications networks. Adaptive Behavior. 1996;5(2):169–207.

[122] Di Caro G, Dorigo M. Mobile agents for adaptive routing. In: El-Rewini H, editor. Proceedings of the 31st International Conference on System Sciences (HICSS-31). IEEE Computer Society Press; 1998. p. 74–83.

[123] Di Caro G. Ant Colony Optimization and its application to adaptive routing in telecommunication networks [Ph.D. Thesis]. IRIDIA, Université Libre de Bruxelles. Brussels, Belgium; 2004.

[124] Sim KM, Sun WH. Ant colony optimization for routing and load-balancing: Survey and new directions. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans. 2003;33(5):560–572.

[125] Zhang Y, Kuhn LD, Fromherz MPJ. Improvements on ant routing for sensor networks. In: Dorigo M, et al., editors. Ant Colony Optimization and Swarm Intelligence: 4th International Workshop, ANTS 2004. vol. 3172 of Lecture Notes in Computer Science. Heidelberg: Springer; 2004. p. 154–165.

[126] Ducatelle F, Di Caro G, Gambardella LM. Using ant agents to combine reactive and proactive strategies for routing in mobile ad hoc networks. International Journal of Computational Intelligence and Applications. 2005;5(2):169–184.

[127] Farooq M, Di Caro G. Routing protocols for next-generation intelligent networks inspired by collective behaviors of insect societies. In: Blum C, Merkle D, editors. Swarm Intelligence: Introduction and Applications. Natural Computing Series. Berlin, Germany: Springer; 2008. p. 101–160.

[128] Ducatelle F, Di Caro G, Gambardella LM. Principles and applications of swarm intelligence for adaptive routing in telecommunications networks. Swarm Intelligence. 2010;In press.

[129] Socha K. ACO for continuous and mixed-variable optimization. In: Dorigo M, et al., editors. Ant Colony Optimization and Swarm Intelligence: 4th International Workshop, ANTS 2004. vol. 3172 of Lecture Notes in Computer Science. Heidelberg: Springer; 2004. p. 25–36.

[130] Tsutsui S. Ant colony optimisation for continuous domains with aggregation pheromones metaphor. In: Proceedings of the The 5th International Conference on Recent Advances in Soft Computing (RASC-04). Nottingham, UK; 2004. p. 207–212.

[131] Tsutsui S. An enhanced aggregation pheromone system for real-parameter optimization in the ACO metaphor. In: Dorigo M, et al., editors. Ant Colony Optimization and Swarm Intelligence: 5th International Workshop, ANTS 2006. vol. 4150 of Lecture Notes in Computer Science. Heidelberg: Springer; 2006. p. 60–71.

[132] Socha K, Dorigo M. Ant colony optimization for continuous domains. European Journal of Operational Research. 2008;185(3):1155–1173.

[133] Socha K, Dorigo M. Ant colony optimization for mixed-variable optimization problems. IRIDIA, Université Libre de Bruxelles, Belgium; 2007. TR/IRIDIA/2007-019.

[134] Maier HR, Simpson AR, Zecchin AC, Foong WK, Phang KY, Seah HY, et al. Ant colony optimization for design of water distribution systems. Journal of Water Resources Planning and Management, ASCE. 2003 May / Jun;129(3):200–209.

[135] López-Ibáñez M, Prasad TD, Paechter B. Ant colony optimisation for the optimal control of pumps in water distribution networks. Journal of Water Resources Planning and Management, ASCE. 2008;134(4):337–346.

[136] Bautista J, Pereira J. Ant algorithms for assembly line balancing. In: Dorigo M, Di Caro G, Sampels M, editors. Ant Algorithms: Third International Workshop, ANTS 2002. vol. 2463 of Lecture Notes in Computer Science. Springer; 2002. p. 65–75.

[137] Blum C, Bautista J, Pereira J. Beam-ACO applied to assembly line balancing. In: Dorigo M, et al., editors. Ant Colony Optimization and Swarm Intelligence. vol. 4150 of Lecture Notes in Computer Science. Springer; 2006. p. 96–107.