Applied Mathematical Sciences, Vol. 3, 2009, no. 23, 1127 - 1140

A Computational Analysis on a Hybrid Approach: Quick-and-dirty ant colony optimization

Paola Pellegrini and Elena Moretti

Abstract

Ant colony optimization (ACO) is a well known metaheuristic. In the literature it has been used for tackling many optimization problems. Often, ACO is hybridized with a local search procedure. All the solutions generated by ants (or some of them) are improved by the local search. In this paper we propose a different framework, called `Quick--and-dirty ant colony optimization`. It is an hybrid approach based on the sequential coupling of ACO and a local search. It exploits the ability of ants to explore the search space. After ants point out the most promising area, the local search procedure is used for analyzing it. Computational experiments on the traveling salesman problem confirm that when the search space is large, by allowing the local search to concentrate on a smaller region it is possible to improve the quality of the performance, provided that this region is properly selected.

1 Introduction

In recent years, the branch of literature related to metaheuristics is focusing more and more on the hybridization of these approaches, either with each other or with more classical procedures from artificial intelligence and operations research. Among the others, also ant colony optimization (ACO) has been object of many studies in this direction. The most widespread hybridization in this context consists in coupling ant algorithms with local search procedures. The ways in which this combination can be made are various. In general it is possible to specify two categories: In the first one, the local search is applied to the tours constructed by ants, and the local optima found are used for the pheromone update as if the ants themselves had found them. In the second one, ACO works in parallel with the local search, sharing information on the search space. Apart from these exchanges of knowledge, the two approaches work independently.

This paper describes an hybridization of ACO and a local search heuristic which cannot be inserted in any of these groups. Here, the two approaches work sequentially, and ant colony optimization is used for pointing out the region of the search space in which to start the local search. The bigger the problem instance to tackle, in general, the more important this point is for the quality of the solution achieved [17]. In this sense, ACO is in the service of the local search.

The success of this idea is linked with the structure of the feasible region. In some problems, there is strong correlation between the cost of a solution and its distance to the global optimum [19]. This phenomenon has been referred to in many ways: clusterization of the best solutions [22], "Massif Central" phenomenon [14], principle of proximate optimality [16], and replica symmetry [27]. It has been shown that this property holds for various combinatorial optimization problems, such as the traveling salesman problem and the graph bi-partitioning problem [7, 19]. Several algorithms base exploitation of this characteristic. Among the others, let us cite iterated local search [22].

If the search space has this peculiarity, for obtaining a good solution, it is sufficient to start the local search heuristic in a promising region, even if no precise information has been gained about the specific (possibly local) optimum whose basin of attraction has to be chosen. Given its searching behavior, ant colony optimization is particularly suitable for exploring, in a short amount of time, very different regions of the search space. After this exploration, ACO can point out the most promising area. The hybrid algorithm proposed is based on this ability of ants of obtaining a rough understanding of the search space in a short amount of time. This understanding is, then, exploited by the local search procedure. In the following, we will refer to this hybridization as `quick-and-dirty ant colony optimization` (`q&dACO`).

We propose an experimental analysis for verifying the validity of the approach. It is based on the traveling salesman problem (TSP). One of the most successful ACO algorithm ($\mathcal{MAX}$–$\mathcal{MIN}$ Ant System [29, 30]) is combined with a modified Lin-Kernighan Heuristic (LKH) [17, 18]. The performance of this hybrid algorithm is compared to the one achievable by the local search alone, which obtained state of the art results for several well known instances of TSP. Moreover, we consider a hybrid procedure of a genetic algorithm with LKH. It is based on the same idea considered for `quick-and-dirty ant colony optimization`, except that the initial phase is committed to a genetic algorithm. This metaheuristic has been chosen as a reference approach owing to its similarities with ant colony optimization.

The paper is organized as follows. Section 2 describes some successful trends of the research on hybrid ant colony optimization. Section 3 describes the `quick-and-dirty ant colony optimization` algorithm, and in Section 4 the experimental analysis is reported. Finally, Section 5 concludes the paper.

2 Literature on hybrid ant colony optimization

The first type of hybridization concerning ant colony optimization algorithms, consists in the incorporation of local search procedures. In particular, in the most classical hybridization, the local search procedure is applied to some (or all) solutions constructed by the ants (see for example [12, 15]). The local optimum returned is then used for the pheromone update. This approach is generally accepted to improve the performance of ACO. It is widely used in the literature, to the extent that often it is not even considered an hybridization.

The second typical way for coupling ACO and local search consists in having the two approaches working in parallel and sharing information. Among the others, Chen and Ting [9] propose to use ant colony system and simulated annealing in parallel. The two approaches share the best solution found, which is used for the pheromone update, on the one hand, and as starting solution of the search, on the other.

Another branch of research focuses on hybridizing ACO with more classical techniques of artificial intelligence and operations research. A short summary of these trends can be found in [3].

An interesting application is proposed by Blum [4]: he describes an algorithm called Beam-ACO. It brings together beam search and ant colony optimization. Beam search is a heuristic search algorithm that uses a heuristic function to estimate the promise of each node it examines. It unfolds the first m most promising nodes at each step, where m is a fixed number, the *beam width* [10]. In some sense it is an incomplete derivative of branch and bound algorithms. The basic algorithmic framework of Beam-ACO is the framework of ant colony optimization. The standard ACO solution construction mechanism is replaced by a solution construction mechanism in which each artificial ant performs a probabilistic beam search. In this context, the extension of partial solutions is done in the ACO fashion rather than deterministically. As the transition probabilities depend on the pheromone values, which change over time, the probabilistic beam searches that are performed by this algorithm are adaptive [3].

Another approach, which in recent years is quite often used, is the coupling of ant colony optimization and constraint programming. The idea is to restrict the search of ants in promising regions of the search space [26]. The hypothesis at the basis of this approach is that, as it happens for various metaheuristics, ACO's performance are better for non-constrained problems, while it degrades for very constrained ones. Constraint programming (CP) is used for specifying the constraints a feasible solution must meet. The CP approach to search for a feasible solution often works by the iteration of constraint propagation and the addition of additional constraints [25]. In the hybrid approaches, at each iteration, first constraint propagation is applied in order to reduce the remaining search space. Then, solutions are constructed in the standard ACO way with respect to the reduced search space. The results are encouraging especially for tackling problems with many constraints which make the search space of feasible solutions very fragmented, though too large for exact methods.

Still working on the search space, although in a different way, Blum and Blesa propose to apply ACO algorithms to an auxiliary search space [5]. The idea that the authors present in the paper is based on replacing the original search space of the tackled optimization problem with an auxiliary search space to which ACO is then applied. This technique needs the existence of a function that maps each element from the auxiliary search space to a solution to the tackled optimization problem. This technique can be beneficial in case the generation of objects from the auxiliary search space is more efficient than the construction of solutions to the optimization problem at hand, and/or the mapping function is such that objects from the auxiliary search space are mapped to high quality solutions of the original search space [3].

Finally, an approach which can be seen as a sort of hybridization, is the application of ACO in a multilevel framework [20, 21]. The basic idea of a multilevel scheme is simple. Starting from the original problem instance, smaller and smaller problem instances are obtained by successive coarsening until some stopping criteria are satisfied. This creates a hierarchy of problem instances in which the problem instance of a given level is always smaller (or of equal size) to the problem instance of the next lower level. Then, a solution is computed to the smallest problem instance and successively transformed into a solution of the next higher level until a solution for the original problem instance is obtained. At each level, the obtained solution might be subject to a refinement process.

Different, and more problem specific, ways for hybridizing ACO are proposed by Di Caro et al. [8], Abraham and Ramos [1], Doerner et al. [11].

3 Quick-and-dirty ant colony optimization

As mentioned in the introduction, the algorithm proposed is an hybridization of an ant algorithm and a local search heuristic. Namely, we consider $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) [29, 30] and a modified Lin-Kernighan Heuristic for the traveling salesman problem (LKH) [17, 18]. We will refer the following arguments to this specific algorithms and problem. Nonetheless the whole reasoning can be extended to a more general case.

We will refer to the hybrid algorithm as `quick-and-dirty ant colony optimization`. This denomination has been chosen for stressing the fact that ants are supposed not to converge on a very high quality solution, but to return quickly a fairly good one. This solution will be appropriate to the aim of the first part of `q&dACO` if the balance of exploration and exploitation allows ants to achieve a good understanding of the feasible region before converging toward a (possibly local) optimum [13, 28]. Moreover, the search space has to present the clusterization of the best solutions [22]. In this case, one can expect the solution returned by ACO to be in the basin of attraction of a local optimum which is *close* to the global one. The local search heuristic is then in charge of selecting the best optimum of the nearby region. This procedure, then, needs to be able to escape from the basin of attraction of an optimum solution to move to the basin of attraction of a different one. It needs to be, then, a quite sophisticated local search, as the modified Lin-Kernighan Heuristic.

3.1 The ant colony optimization phase

Ants construct solutions incrementally, choosing the components (that correspond to edges in the TSP) probabilistically. The choice is biased by an heuristic measure (η_{ij} for arc (i,j)) and by the pheromone trail (τ_{ij} for arc (i,j)). In particular, in $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System, ant k being in node i and not having visited the nodes belonging to the set N_k, chooses to move to node $j \in N_k$ with probability p_{ij}, that is described in formula 1:

$$p_{ij} = \frac{[\tau_{ij}]^{\alpha}\,[\eta_{ij}]^{\beta}}{\sum_{h\in N_k}[\tau_{ih}]^{\alpha}\,[\eta_{ih}]^{\beta}}. \tag{1}$$

α and β are parameters of the algorithm. In the TSP, the heuristic measure η_{ij} corresponds to the inverse of the length of the arc (i,j). At the beginning of the search, the pheromone trail is equally distributed on all the arcs ($\tau_{ij} = \tau_0$ for each arc (i,j)). As a consequence, the choice is influenced only by the heuristic information.

After each iteration, i.e. after the m ants of a colony have constructed their tours, the pheromone trail τ_{ij} is updated on each arc (i,j) according to formula 2:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \Delta\tau_{ij}, \tag{2}$$

where ρ is a parameter such that $0 \leq \rho \leq 1$, and

$$\Delta\tau_{ij} = \begin{cases} \frac{1}{C_b} & \text{if arc } (i,j) \text{ belongs to tour } b, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

C_b is the cost associated to tour b, and tour b is either the iteration-best tour or the best-so-far tour. The schedule according to which the solution to be exploited is chosen, is described by Dorigo and Stützle [13]. Apart from this update, in $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System the pheromone trail is bounded between τ_{min} and τ_{MAX}. Following [13], we use the values reported in formula 4, 5

Table 1: $\mathcal{MM}$AS.

```
Initialize pheromone trails
While stopping criterion is not satisfied
{
      Generate a population of solutions
      Find the iteration best solution
      Update the best so far solution
      Perform the pheromone update
      Check for pheromone bounds
}
```

and 6 for τ_0, τ_{MAX} and τ_{min}, respectively. At the beginning of a run, the best solution corresponds to the one found by the nearest neighbor heuristic (NN).

$$\tau_0 = \frac{1}{\rho C_{NN}} \tag{4}$$

$$\tau_{MAX} = \frac{1}{\rho C_{best-so-far}}, \tag{5}$$

$$\tau_{min} = a\tau_{MAX}, \tag{6}$$

where a is a problem-dependent constant [13]. Remark that τ_{min} is a strictly positive quantity. The general algorithmic framework that characterizes $\mathcal{MM}$AS is reported in Table 1.

This procedure implies, at the beginning, the spot exploration of many different areas of the feasible regions, followed by the concentration in the most promising areas, and finally by the convergence toward the best tour found [13]. `Quick-and-dirty ant colony optimization` needs only the first two phases, allowing the local search procedure to take care of the choice of the best minimum.

A typical measure of the level of convergence of ant algorithms is the average λ-branching factor [12, 13]. In the following, we will refer to this measure as to average γ-branching factor ($\bar{\gamma}$). This choice is due to the fact that, in the second phase of `q&dACO`, a modified version of the Lin-Kernighan Heuristic is used, which is based on λ-opt moves.

The average γ-branching factor measures the distribution of the pheromone trail on the arcs. The basic idea is that the more the algorithm is close to convergence, the fewer the arcs with a significant amount of pheromone are. When the convergence is achieved, the concentration of pheromone is very small on all the arcs incident to each node, but the ones belonging to the tour toward which ants converged [12, 13].

For computing the average γ-branching factor, the first step consists in finding for each node i the maximal (τ^i_{max}) and the minimal (τ^i_{min}) pheromone trail value on arcs incident to i itself. The γ-branching factor for node i is equal to the number of edges (i, j) on which the pheromone trail is greater than a defined threshold. This threshold depends on the maximum and the minimum level of pheromone present on arcs incident on i itself. The inequality that needs to be satisfied is reported in 7:

$$\tau_{ij} \geq \tau^i_{min} + \kappa(\tau^i_{max} - \tau^i_{min}), \tag{7}$$

where κ is a parameter $(0 \leq \kappa \leq 1)$. The γ-branching factor is always bounded between 2 and $n - 1$, with n number of nodes of the graph: When the pheromone trail is equal on all the edges, for all i we have $\tau^i_{min} = \tau^i_{max}$, and then $\tau_{ij} = \tau^i_{min}\, \forall j \in E$. On the other hand, when the algorithm converges, only one solution is visited. After a while, only the two edges incident to each node belonging to this solution will have a high amount of pheromone. On all the other edges the trail will keep decreasing up to τ_{min}. The average γ-branching factor is the average of the γ-branching factors of all nodes. It gives a representation of the size of the region of the search space which has a significant probability of being explored. For example, if $\bar{\gamma}$ is very close to 3, it means that in average 3 arcs incident to each node are likely to be chosen. In case of the TSP it means that most likely the solutions constructed will belong to a set with $O(2^n)$ elements, with respect to the $O(n!)$ elements of the feasible region. In general, the smaller the average γ-branching factor, the smaller the area considered.

Using the average γ-branching factor, then, it is somehow possible to measure the size of the area under analysis in any moment. By using as stopping criterion for the algorithm the average γ-branching factor crossing a certain threshold br, the best solution found $\hat{S}$ will belong to the area still under exploration – and then to the most promising region of the search space –.

3.2 The local search phase

Once ants return a solution belonging to the most promising region of the search space, `quick-and-dirty ant colony optimization` passes to a local search heuristic, which uses as starting solution $\hat{S}$.

The only characteristic that this approach needs to have, is the ability – to some extent – of escaping from local minima. Lin-Kernighan Heuristic for the traveling salesman problem [18] has this feature.

A very detailed description of the heuristic is reported in [17]. It is a modification of a previous algorithm proposed by Lin and Kernighan [24]. It is based on the idea of λ-optimality [23]: a tour is said to be λ-optimal if it is not possible to obtain a shorter tour by replacing any λ of its links by any other set of λ links. While in the typical implementation the value chosen for λ is either 2 or 3, Lin and Kernighan [24] propose a variable λ-opt. At each iteration step the algorithm examines, for ascending values of λ, whether an interchange of λ links may be convenient. The method for selecting which arcs to exchange aims at being efficient in terms of computational cost, without worsening the quality of the solution found. Helsgaun [17, 18] proposes additional rules for restricting and directing the search. The introduction of these elements improves the performance of the algorithm. Thanks to the variability of the factor λ, this heuristic is able to escape from local minima, by defining in different ways the neighborhood of a solution. Various parameters are present in the algorithm. For a complete description we refer the reader to [17].

For large enough instances, the algorithm needs a stopping criterion different from having explored all the possibilities. To this aim, Helsgaun [17, 18] fixes a limit for the number of λ-opt moves allowed. In a substantially equivalent way, we use the consumption of a given computational time.

Clearly, the bigger the instance tackled, the higher the computational time needed to complete the search, since the size of the search space to be explored increases. When the available time is short, the starting point is crucial for the quality of the final solution. Nonetheless, it is not fundamental to start from a solution which is in the basin of attraction of the global optimum: even if the heuristic has as starting point a solution $\hat{S}$ which is in the basin of attraction of a local minimum, the search strategy may get to the best solution quite fast, provided that the the local and the global minimum are not too far [19, 22, 14, 16, 27]. This is due to the fact that the local search heuristic moves in the search space along trajectories implied by the definition of the neighborhoods, and it is able to escape from local minima.

4 Experimental analysis

The experimental analysis proposed is based on the traveling salesman problem. For the first phase of `q&dACO`, we consider the ACOTSP program implemented by Thomas Stützle as a companion software for [13]. The code has been released in the public domain and is available for free download on www.aco-metaheuristic.org/aco-code/. In a similar way, for the second phase, we consider the LKH problem implemented by Keld Helsgaun. Also the latter has been released in the public domain. It is available for free download on www.akira.ruc.dk/∼keld/research/LKH/.

For evaluating the performance of `q&dACO`, we compare its results with those achieved by the LKH heuristic [18] started from random solutions, and with the ones achieved by an hybrid genetic algorithm. This second algorithm is based on the same idea considered for `quick-and-dirty ant colony optimization`, except that the initial phase is committed to a genetic algorithm. This metaheuristic has been chosen as a reference approach owing to its similarities with ant colony optimization: they are both *population based* metaheuristics

Table 2: Number of instances in which each algorithm obtains the best result.

	# best result		
	q&dACO	GA-LKH	LKH
1000 nodes	195	172	154
1500 nodes	168	132	82
2000 nodes	162	113	61
2500 nodes	139	101	48

Table 3: Average percentage difference between each algorithm's result and the best observed one for each instance.

	average % gap		
	q&dACO	GA-LKH	LKH
1000 nodes	3.48e-06	1.44e-05	3.12e-05
1500 nodes	4.77e-05	3.99e-04	4.67e-04
2000 nodes	9.30e-05	5.33e-04	7.09e-04
2500 nodes	1.21e-04	7.14e-04	7.62e-04

[6], and then they may have a similar ability of exploring the search space. The implementation of the genetic algorithm is based on [31] and we refer to the hybrid approach as GA-LKH. In particular, starting from an initial population of solutions, new ones are generated iteratively by edge recombination. At each iteration, some individuals are subject to mutation. Edge recombination [31] consists in generating a solution starting from two different ones, using components that are present in both of them, whenever possible. Mutation swaps probabilistically adjacent customers.

The criterion chosen for shifting from the genetic algorithm to the local search phase is the computational time elapsed. In `q&dACO` the first phase is concluded when the average γ-branching factor reaches a fixed threshold. After preliminary experiments the average time employed by ants has been computed, and it has been used as stopping criterion for GA-LKH.

The instances used are generated through `portgen`, the instance generator adopted in the DIMACS TSP Challenge. In particular, the ones we consider here consist of two dimensional integer-coordinate cities distributed in a square of size $10^6 \times 10^6$.

We consider four sets of instances, with 1000, 1500, 2000, and 2500 nodes, respectively. Two hundreds instances of each set are used for the experiments. The parameters used for the algorithms are the ones suggested in the literature [13, 17, 31]. For `q&dACO`, the value of κ for the computation of the average γ-branching factor is 0.05 [13].

A single run is performed for each instance [2]. The experiments are run on a AMD Opteron 250. The computational time available for each run is 90, 135, 180, and 225 seconds for the instances with 1000, 1500, 2000, and 2500 nodes, respectively. In each run the search is restarted three times. Both the LKH and the GA-LKH heuristic are started each time with a different random seed. `Quick-and-dirty ant colony optimization` is restarted each time considering a different value of br, the threshold for the average γ-branching factor which represents the stopping criterion for the first phase of `quick--and-dirty ant colony optimization`. The values used are $3.0, 3.4$, and 3.8. They have been chosen after some preliminary experiments. A value around 3, as explained in Section 3.1, implies that, on average, a high level of pheromone is present on three edges incident on each node. This means that the search is strongly biased toward $O(2^n)$ solution. This number is quite low, compared with the $O(n!)$ solutions belonging to the feasible region, and then, when this average γ-branching factor is reached, ants have already made a considerable selection. Nonetheless, given that the task assigned to ACO is to point out a promising region of the search space for the local search heuristic, it is not necessary to allow it to converge toward a (possibly local) minimum. For this reason the value of br is fixed higher than 2, but still quite low.

For the four sets of instances, the number of times in which each algorithm obtains the best result (# best result) is reported in Table 2, while the average percentage difference, for each instance, between each algorithm's result and the best observed one (average % gap) is reported in Table 3. For each set, we consider 200 instances. The cases in which the three algorithms achieve the same result are 147, 68, 45 and 28 for the instances with 1000, 1500, 2000 and 2500 nodes, respectively.

The distribution of percentage difference for each class of instances is shown in Figure 1.

The differences between the algorithm is statistically significant according to the Wilcoxon Tests.

As it can be observed, `q&dACO` appears to perform better than both the LKH and the GA-LKH heuristic for all the sets of instances. Furthermore, the hybrid genetic algorithm achieves better results than the local search heuristic.

A trend can be detected in Tables 2 and 3. The relative performance of `q&dACO` improves as the size of the instances increase. In other words, the higher the number of nodes of the instances, the better the relative performance of `q&dACO`. On the one hand, this supports the intuition that, the larger the instance, the more important the starting point of the local search. On the other hand, ant colony optimization appears to be a very effective approach for gaining quickly a rough understanding of the characteristic of the search space.

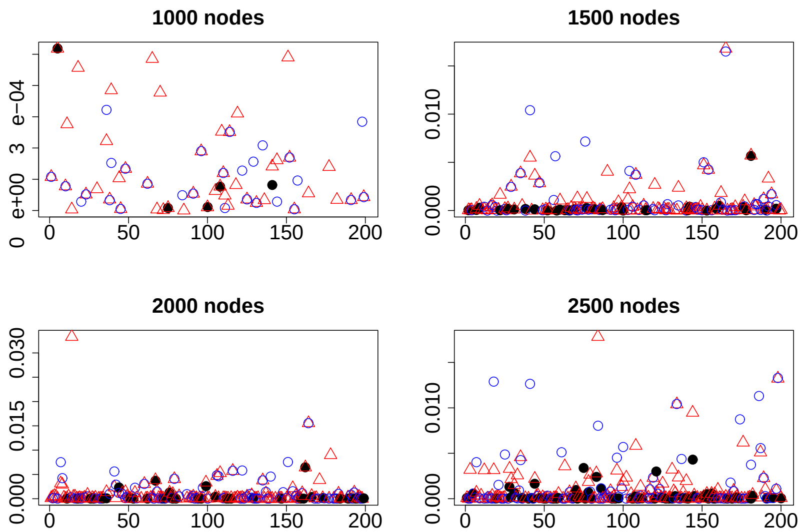

Figure 1: Percentage difference between each algorithm's result and the best observed one, for each instance with 1000, 1500, 2000, 2500 nodes. Triangles represent the LKH heuristic. Circles represent the EA-LKH hybrid. Bullets represent `q&dACO`.

5 Conclusions

In this study a new hybrid algorithm is proposed. It is based on the coupling of an ACO algorithm and a local search heuristic, namely $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System and Lin-Kernighan Heuristic for the traveling salesman problem. It is called `quick-and-dirty ant colony optimization`. The main element which makes this approach different from previous hybridization of ant colony optimization is the fact that, here, the ant algorithm is used as an instrument for the choice of the starting point of the local search procedure. In this way, ants do the dirty job, exploring the search space and selecting the most promising region. The framework of ACO metaheuristic allows to achieve this objective quite fast. From the promising region pointed out, the local search heuristic moves in the search space along trajectories implied by the definition of the neighborhoods. If the feasible region of the problem is such that the best solutions are clustered, as in the traveling salesman problem, starting the local search in the basin of attraction of a good local optimum, makes it likely to get closer and closer to the global one.

Computational experiments support this intuition: the larger the instances considered for the traveling salesman problem, the better the performance achieved by `q&dACO` compared to LKH and to a hybrid genetic algorithm based on the same basic ideas.

Consisting in a metaheuristic and a local search procedure, the approach proposed may be easily applied to many combinatorial optimization problems. Object of future work will be the application of `q&dACO` to different problems. An interesting element that may be investigate is the relative importance of the two phases, when varying the problem tackled. For example, this relevance may vary in case of very constrained problems with respect to an unconstrained one. Moreover, various local search procedures may be tested. Different procedures, considering different neighborhoods, may get to very different results.

References

[1] A. Abraham and V. Ramos. Web usage mining using artificial ant colony clustering and genetic programming. In *Proc. 2003 Congress on Evolutionary Computation (CEC'03)*, pages 1384–1391, Camberra, Australia, 2003. IEEE Press.

[2] M. Birattari and M. Dorigo. How to assess and report the performance of a stochastic algorithm on a benchmark problem: Mean or best result on a number of runs? *Optimization Letters*, 2007.

[3] C. Blum. Ant colony optimization: introduction and recent trends. *Physics of Life Reviews*, 2:353–373, 2005.

[4] C. Blum. Beam-aco — hybridizing ant colony optimization with beam search: An application to opem shop scheduling. *Computers & Operations Research*, 32(6):1565–1591, 2005.

[5] C. Blum and M. Blesa. Combining ant colony optimization with dynamic programming for solving tha k-cardinality tree problem. In J. Cabestany, A. Prieto, and F. Sandoval, editors, *8th international work-conference on artificial neural network, computational intelligence and bioinspired systems (IWANN'05*, volume 3512 of *Lecture Notes in Computer Science*, pages 25–33, Berlin, Germany, 2005. Springer.

[6] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.

[7] K. Boese, A. Kahng, and S. Muddu. A new adaptive multistart technique for combinatorial global optimization. *Operations Research Letters*, 16(2):101–113, 1994.

[8] G. D. Caro, F. Ducatelle, and L. Gambardella. Anthocnet: an ant-based hybrid routing algorithm for mobile ad hoc networks. In *Proceedings of Parallel Problem Solving from Nature (PPSN) VIII, LNCS 3342*, pages 461–470. Springer-Verlag, 2004.

[9] C. Chen and C. Ting. A hybrid ant colony system for vehicle routing problem with time windows. *Journal of the Eastern Asia Society for Transportation Studies*, 6:2822–2836, 2005.

[10] F. D. Croce, M. Ghirardi, and R. Tadei. Recovering beam search: enhancing the beam search approach for combinatorial optimisation problems. In *Proceedings of PLANSIG 2002. 21th workshop of the UK planning and scheduling special interest group*, pages 149–169, 2002.

[11] K. Doerner, R. Hartl, and M. Reimann. A hybrid aco algorithm for the full truckload transportation problem. citeseer.ist.psu.edu/doerner01hybrid.html.

[12] M. Dorigo and L. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1:53–66, 1997.

[13] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, 2004.

[14] C. Fonlup, D. Robilliard, P. Preux, and E.-G. Talbi. *Fitness landscape and performance of meta-heuristics*, pages 257–268. Kluwer Academic Publishers, Boston, Massachusetts, USA, 1999.

[15] L. Gambardella and M. Dorigo. An ant colony system hybridized with a new local search for the sequential ordeting problem. *INFORMS Journal of Computing*, 12(3):237–255, 2000.

[16] F. Glover and M. Laguna, editors. *Tabu Search*. Kluwer Academic Publishers, Norwell, Massachusetts, USA, 1997.

[17] K. Helsgaun. An effective implementation of the lin-kerninghan traveling salesman heuristic. Technical Report 81, Writings on Computer Science, Roskilde University, Copenhagen, Denmark, 1998.

[18] K. Helsgaun. An effective implementation of the lin-kerninghan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000.

[19] H. H. Hoos and T. Stützle. *Stochastic Local Search. Foundations and Applications*. 2004.

[20] P. Korošec, J. Šilc, and B. Robič. Mesh-partitioning with the multiple ant-colony algorithm. In M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. Stützle, editors, *ANTS Workshop*, volume 3172 of *Lecture Notes in Computer Science*, pages 430–431, Brussels, Belgium, 2004. Springer.

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

[22] H. Laurenço, O. Martin, and T. Stützle. *Iterated local search*, volume Handbooks of Metaheuristics vol. 57 of International Series in Operations Research & Management Science, pages 321–353. Kluwer Academic Publishers, Norwell, Massachusetts, USA, 2002.

[23] S. Lin. Computer solutions of the traveling salesman problem. *Bell System Tech. Journal*, 44:2245–2269, 1965.

[24] S. Lin and B. Kernighan. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21:498–516, 1973.

[25] K. Marriott and P. Stuckey. *Programming with constraints*. MIT Press, Cambridge, MA, 1998.

[26] B. Meyer and A.Ernst. Integrating aco and constraint propagation. In M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. Stützle, editors, *ANTS Workshop*, volume 3172 of *Lecture Notes in Computer Science*, pages 166–177, Brussels, Belgium, 2004. Springer.

[27] M. Mézard, G. Parisi, and M. Virasoro. *Spin-glass theory and beyond*. World Scientific Lecture Notes in Physics vol. 9, Singapore, 1987.

[28] P. Pellegrini, D. Favaretto, and E. Moretti. On max-min ant system's parameters. Technical Report TR/132/2006, Department of Applied Mathematics, Ca' Foscari University, Venice, Italy, 2006.

[29] T. Stützle and H. Hoos. Improvements on the ant system: introducing the max-min ant system. In R. Albrecht, G. Smith, and N. Steele, editors, *Proceedings of Artificial Neural Nets and Genetic Algorithms 1997*, pages 245–249, Norwich, U.K., 1998. Springer-Verlag.

[30] T. Stützle and H. Hoos. Max-min ant system. *Future Generation Computer Systems*, 16(8):889–914, 2000.

[31] D. Whitley, T. Starkweather, and D. Shaner. *The traveling salesman problem and sequence scheduling: quality solutions using genetic edge recombination*, volume Handbook of Genetic Algorithms, pages 350–372. Van Nostrand Reinhold, New York, NY, USA, 1991.