An Analysis of Algorithmic Components for Multiobjective Ant Colony Optimization: A Case Study on the Biobjective TSP

Manuel López-Ibáñez and Thomas Stützle

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2009-019

June 2009

IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:

IRIDIA, *Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle*
Université Libre de Bruxelles
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2009-019

An Analysis of Algorithmic Components for Multiobjective Ant Colony Optimization: A Case Study on the Biobjective TSP

Manuel López-Ibáñez manuel.lopez-ibanez@ulb.ac.be
Thomas Stützle stuetzle@ulb.ac.be

IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium

June 2009

Abstract

In many practical problems, several conflicting criteria exist for evaluating solutions. In recent years, strong research efforts have been made to develop efficient algorithmic techniques for tackling such multi-objective optimization problems. Many of these algorithms are extensions of well-known metaheuristics. In particular, over the last few years, several extensions of ant colony optimization (ACO) algorithms have been proposed for solving multi-objective problems. These extensions often propose multiple answers to algorithmic design questions arising in a multi-objective ACO approach. However, the benefits of each one of these answers are rarely examined against the alternative approaches.

This article reports results of an empirical research effort aimed at analyzing the components of ACO algorithms for tackling multi-objective combinatorial problems. The results presented here focus on the bi-objective travelling salesman problem. The main goal is to study the effect of algorithmic components and their possible interactions on the performance of a multi-objective ACO algorithm. Examples of design choices are the use of local search, the use of one versus several pheromone matrices, and the use of one or several ant colonies. The analysis of the results is carried out by means of an exploratory graphical analysis tool based on the attainment function methodology. This tool allows to identify regions of the objective space where there is a strong difference between two alternative strategies.

Keywords: Multiobjective Optimization, Ant Colony Optimization, Travelling Salesman Problem, Quadratic Assignment Problem

1 Introduction

Ant colony optimization (ACO) [1] is a general-purpose stochastic local search (SLS) method [2] that is inspired by the pheromone trail laying and following behavior of some real ant species. Although some few adaptations of ACO to continuous optimization exist, the main application area of ACO is to NP-hard combinatorial optimization problems. Due to the practical relevance of this class of problems and the high performance reached by ACO algorithms in many applications, the number of applications of ACO algorithms has risen strongly over the recent years [1].

In many practically relevant problems it is common to evaluate the quality of candidate solutions with respect to various, often conflicting objectives. It is therefore not surprising that several extensions of ACO algorithms have been proposed to tackle multi-objective combinatorial optimization problems (MCOPs). Typically, these approaches were designed with the intention to tackle multi-objective problems in terms of Pareto optimality. In fact, there have been a few tens of papers that deal with multi-objective ACO (MOACO) algorithms for such problems. For a detailed overview, we refer to [3, 4].

Unfortunately, surprisingly little research on MOACO is targeted towards an understanding of the contribution of specific design choices of MOACO algorithms to performance. Such design options concern the use or not of local search, different ways of directing the search towards the Pareto front (for example, by the use of nondominance criteria or the use of weighted sum aggregations), the use of more than one ant colony to specialize on specific areas of the Pareto front, and so on. In fact, most articles propose one specific way of how to tackle MCOPs by an ACO algorithm [5, 6]; rare are comparisons of several MOACO algorithms [4] or studies that compare few design alternatives [7, 8]. In fact, we believe that our previous article on an experimental analysis of MOACO algorithms for the biobjective quadratic assignment problem (BQAP) [9] is one of the most complete for what concerns the experimental comparison of specific design options of MOACO algorithms.

In this article, we build and extend upon our earlier work in the area of the experimental analysis of MOACO algorithms. As in our previous work, we use a component-wise view of how to design a MOACO algorithm and, hence, analyze the algorithmic performance in an experimental design type style. We extend the earlier analysis in various ways. First, and maybe most important, we use more advanced tools for the empirical analysis of the behavior of multi-objective optimizers. In particular, we base our analysis on the extensive usage of the attainment functions methodology [10, 11], and we use graphical illustrations to examine where in the objective space the various MOACO algorithms differ in performance [12, 13]. This gives us a detailed indication of the impact of specific MOACO components on performance. Second, we tackle another problem, which is the bi-objective traveling salesman problem (BTSP). The BTSP differs from the BQAP very strongly in its search space characteristics [14], and also in the fact that the solution evaluation in the BTSP is much faster than for the BQAP. Moreover, the BTSP can profit much from the usual algorithmic speed-up techniques for the single-objective TSP such as candidate lists [15]. Third, some of the algorithm details studied have not been considered previously. In fact, the results we present here form part of a larger research effort that aims at a detailed analysis of the impact of MOACO components on algorithm performance.

The article is structured as follows. In Section 2 we give some basic definitions and introduce the BTSP. Section 3 reviews concisely available ACO algorithms for MCOPs tackled in the Pareto sense and introduces the algorithmic components we study. The experimental setup is detailed in Section 4 and computational results are presented in Section 5. We end with some concluding remarks in Section 6.

2 Multiobjective Combinatorial Optimization

In this article we focus on multi-objective combinatorial optimization problems (MCOPs), where candidate solutions are evaluated by an *objective function vector* $\vec{f} = (f_1, \dots, f_d)$ with d objectives. MCOPs are often solved without a priori assumptions on the preferences of the decision maker. In such case, the goal is to determine a set of feasible solutions that "minimizes" the objective function vector $\vec{f}$ according to the Pareto optimality criteria. Let $\vec{u}$ and $\vec{v}$ be vectors in $\mathbb{R}^d$. We say that $\vec{u}$ *dominates* $\vec{v}$ ($\vec{u} \prec \vec{v}$) iff $\vec{u} \neq \vec{v}$ and $u_i \leq v_i,\ i = 1, \dots, d$. Furthermore, $\vec{u}$ and $\vec{v}$ are *nondominated* iff $\vec{u} \not\prec \vec{v}$ and $\vec{v} \not\prec \vec{u}$. To simplify the notation, we also say that a feasible solution s dominates another solution s' iff $\vec{f}(s) \prec \vec{f}(s')$. A solution s is a *Pareto optimum* iff no other feasible solution s' exists such that $\vec{f}(s') \prec \vec{f}(s)$. The goal in MCOPs then typically is to determine the set of all Pareto-optimal solutions. However, determining the Pareto optimal set is often computationally intractable, and therefore it is usually preferable to approximate the Pareto set as well as possible in a given amount of time. Such an approximation is always a set of solutions that are mutually nondominated.

The notion of Pareto-optimality can be extended to compare sets of mutually nondominated solutions [16, 17]. When comparing two sets A and B of nondominated solutions, we say that A *is better than* B ($A \lhd B$) iff every $\vec{b} \in B$ is dominated by or equal to at least one $\vec{a} \in A$, and $A \neq B$. If we have that $A \ntriangleleft B$, $B \ntriangleleft A$, and $A \neq B$, A and B are *incomparable* ($A \parallel B$), and none of the two sets is preferred over the other according only to Pareto-optimality.

In this paper we deal with the BTSP, which is a direct extension of the widely studied single-objective TSP with two cost values between each pair of distinct cities. A BTSP instance is defined by a complete graph $G = (V, A)$, with $n = |V|$ nodes $\{v_1, \ldots, v_n\}$, a set of arcs A that fully connects the graph. Each arc has an associated cost vector whose components are $c_1(v_i, v_j)$ and $c_2(v_i, v_j)$, $i \neq j$. Here we assume that instances are symmetric, that is $c_l(v_i, v_j) = c_l(v_j, v_i)$, $i \neq j$, $l = 1, 2$. The goal in the BTSP is to find the set of Hamiltonian tours $p = (p_1, \ldots, p_n)$ "minimizing", in terms of Pareto optimality, the total tour cost, which is given by

$$f(p) = c\left(v_{p(n)}, v_{p(1)}\right) + \sum_{i=1}^{n-1} c\left(v_{p(i)}, v_{p(i+1)}\right).$$

3 Multi-objective ant colony optimization

The first extensions of ACO for MCOPs were proposed by the end of the 90s. While various early approaches targeted problems where the objectives can be ordered lexicographically or preferences of a decision maker are known, most proposals of MOACO algorithms were targeting towards multi-objective problems that are tackled in terms of Pareto optimization. For an overview of most of the available MOACO algorithms we refer to [4, 3].

3.1 Available MOACO algorithms

All proposed MOACO algorithms somehow try to modify underlying ACO components so as to allow the algorithm to direct the search towards the different regions of the Pareto front simultaneously. A main question is how to represent the solution components of different areas of the front in the form of pheromones. Essentially, the possibilities available here are to use one pheromone matrix [18, 6] or several pheromone matrices [19, 5, 18, 7]. In the latter case, typically each objective has associated one pheromone matrix. Related to this decision is the choice of the ants that deposit pheromones. Typically, if one pheromone matrix is used, some or all nondominated solutions are selected for update [18], while in the use of several pheromone matrices some elitist choices w.r.t. the objectives represented in the pheromone matrices are done [19, 5, 7]. During the solution construction, the question arises how to use the pheromones and also the heuristic information. If several pheromone matrices or several forms of heuristic information are used, a common choice is to use some weighted aggregation of these values [19, 5, 18].

Unfortunately, in most papers where MOACO approaches are described, the various possible algorithmic components are not studied in very much details so that little insight about the impact of specific choices is available. To some extend, the following articles provide more details. In the article by Iredi et al. [18] some few design options are studied for a specific scheduling problems. Alaya et al. [7] study various combinations of the number of pheromone matrices and the number of colonies for a multi-objective knapsack problem. The review article by García-Martínez et al. [4] experimentally compares various complete MOACO algorithms, rather than individual algorithmic components, by using the BTSP as a benchmark problem, although not applying usual TSP specific algorithm techniques such as candidate lists or local search. We therefore decided to start a significant research effort to analyze in more detail the impact various MOACO algorithm components may have on performance, extending in a strong way beyond our initial efforts [9].

3.2 Studied algorithm components

Based on our earlier experience with MOACO algorithms and the various proposals made in the literature, we decided to study the following main algorithmic components for composing MOACO algorithms for the BTSP. Note that for some components, only specific levels of other components make sense. This is indicated below where necessary.

Local Search. It is well known that local search has a strong influence on performance for single-objective ACO algorithms [1]. Hence, we study here also the impact of local search for

the BTSP. We use an iterative improvement algorithm based on the 2–exchange neighborhood. Our local search algorithm uses a weighted sum combination of the two objectives and exploits that standard TSP speed-up techniques (candidate sets are reordered for each weight vector).
Pheromone Information. We use two levels for this component, either one pheromone matrix or two pheromone matrices.

If one pheromone matrix is used, the solution construction exploits this pheromone matrix as usual in single-objective ACO algorithms. Since, however, two types of heuristic information η_{ij} are available for each arc (i,j) (one for each distance matrix), these must be exploited in an appropriate way. We decided to do so by giving equal weight to each of these and to choose $\eta_{ij} = 0.5 \cdot \eta^1_{ij} + 0.5 \cdot \eta^2_{ij}$.

When multiple pheromone matrices are utilized, each matrix is typically associated to one objective, and they are aggregated by means of weights. The MOACO literature contains examples of both linear versus non-linear aggregations. Here, we use a linear aggregations, since due to our underlying ACO algorithm we can ensure that the pheromones are in a same range.

$$p^k_{ij} = \frac{\left((1-\lambda)\tau^1_{ij} + \lambda\tau^2_{ij}\right)^\alpha \cdot \left((1-\lambda)\eta^1_{ij} + \lambda\eta^2_{ij}\right)^\beta}{\sum_{l\in\mathcal{N}^k_i} \left((1-\lambda)\tau^1_{il} + \lambda\tau^2_{il}\right)^\alpha \cdot \left((1-\lambda)\eta^1_{il} + \lambda\eta^2_{il}\right)^\beta} \qquad \text{if } j \in \mathcal{N}^k_i \tag{1}$$

where $\mathcal{N}^k_i$ is the feasible neighborhood of ant k, that is, those cities not visited yet by ant k and τ^l_{ij} is the pheromone trail for arc (i,j) for either objective $l = 1, 2$.

Weight Setting Strategies. Whenever weights are used, either for aggregating multiple pheromone matrices or for performing a single-objective local search over a scalarization of the multi-objective problem, several strategies can be applied for setting the weights used at each iteration of the algorithm.

A possible strategy is to define as many weights as ants. Then, each ant may be assigned a different weight from the other ants at each iteration. The first approach was already proposed by Iredi et al. [18]. Alternatively, we can assign to all ants the same weight but modify the weight slightly at each iteration. This latter strategy is more similar to the approach followed by Two-Phase Local Search [20]. In any case, if an ant is assigned a weight vector, the subsequent local search will use the same weight vector for computing the weighted sum combination of the objectives.

Pheromone deposit. In the case of one pheromone matrix, a common choice is to allow nondominated solutions to deposit pheromones. Here, we examine whether we allow all, or a limited number of ants to deposit pheromone. If a limited number is chosen, it is advantageous to choose them distributed along the Pareto front. For this task, we adapt the niching mechanism of SPEA-2 [21].

In the case of multiple pheromone matrices, some elitist strategy is usually followed. Here, we use the method of updating each pheromone matrix with the best solution for each objective.

Concerning the choice of the updating and w.r.t. to the choice between iterated-best and global-best update, we follow directly the strategy followed by the underlying ACO algorithm, which here is $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System [22].

Multiple Colonies. Multiple colonies are typically used to better direct the search towards specific areas of the Pareto front. To do so, each colony has its own pheromone information, be it a single pheromone matrix or two pheromone matrices. The total number of ants is divided equally between the colonies. All solutions generated at each iteration are placed in a common archive, hence, colonies influence each other. The cooperation among the colonies can be made stronger by exchanging solutions among the colonies. One such approach is called *update by region* [18], where the Pareto front used for update is divided into as many parts as colonies, and each part is used to update the pheromone information of a different colony. In the bi-objective case, this can be implemented by sorting the Pareto front according to the first objective, and subdividing the Pareto front into c smaller fronts of equal size.

4 Experimental Setup

All algorithms are implemented in `C` and compiled with `gcc`, version 3.3. The implementation of the multi-objective ACO algorithms is based on ACOTSP [23]. Experiments are carried out on AMD Opteron$^{\text{TM}}$ 2216 dual-core processors running at 2.4 GHz with 2 MB L2-Cache and 4 GB RAM, under Rocks Cluster GNU/Linux. Due to the sequential implementation of the code, only one core is used for running the executable.

We wish to focus on the multi-objective components. Hence, we use a common underlying ACO algorithm, $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) [22], for the management of pheromones (evaporation, pheromone trail limits, etc.). Other basic parameters were $\alpha = 1$, $\beta = 2$, $\Delta\tau = 1$, $\tau_0 = \tau_{\max}$, where $\Delta\tau$ is the amount of pheromone deposited by an ant and τ_0 is the initial value of the pheromone. Note that the usage of $\Delta\tau = 1$ implies that the amount of pheromone deposited is independent of the solution quality of the ant's tour. The evaporation factor is $\rho = 0.2$ if 2-opt local search is used, otherwise it is $\rho = 0.05$. The total number of ants is $m = 30$. In single-objective ACO algorithms for the TSP, the use of candidate lists is essential for obtaining high-quality solutions. Hence, we also use candidate lists for the BTSP. For generating the candidate list, we follow the technique proposed by [15], where edges are sorted according to its dominance ranking. The size of the candidate list is 40.

For the multi-objective components, we compare the following. The use of one pheromone matrix and an update strategy based on the dominance criteria against the use of two pheromone matrices. In the latter case, we use aggregations of the pheromone matrices (and objectives) by m weight vectors which are maximally dispersed and each pheromone matrix is updated by the best ant for the corresponding objective. (Note that each component of the weight vectors is in the interval $[0, 1]$ and that the sum of the two components is equal to one. We tested different number of colonies $c \in \{1, 3, 5\}$, with ants equally divided among the colonies (m/c). The colonies share information in order to identify dominated solutions, which are not used for updating; however, they have their own set of (nondominated) solutions for updating. We tested both *update by origin* (each ant is considered only for updating the pheromone matrix of the colony to which it originally belongs) and *update by region* strategies. We e found only a slight advantage in favor of update by region. Hence, we focus on the result obtained by this approach

We also test the two weight setting strategies explained above. These weight setting strategies are only relevant for components utilizing weights. In fact, they do not apply if only one pheromone matrix is used and no local search performed.

Each experiment was run for a time limit of 300 seconds and repeated ten times with different random seeds. Note that our code for all variants was optimized by using the usual speed-ups and precomputations that are also applied for ACO algorithms that tackle the single-objective TSP [1] and that are also implemented in the ACOTSP software.

We performed experiments on 3 Euclidean BTSP instances of 500 cities. Two were generated taking a single-objective Euclidean instance, and altering the coordinates of the cities by a correlated random noise to create a new distance matrix. This way, we generated a positive-correlated instance, with a correlation between the distance matrices of 0.77, and an estimated correlation between the objectives of 0.81, measured by generating 10 000 random solutions. We also generated a zero correlated instance with a correlation between the distance matrices of 0.01, and an estimated correlation between the objectives of -0.01. In addition, we also considered the instance `euclidAB500` available from `http://eden.dei.uc.pt/~paquete/tsp/`. This instance has a correlation between the matrices of 0.01, and an estimated correlation between the objectives of 0.01.

In contrast to previous results on the BQAP, the different correlation does not lead to large differences in the behavior of MOACO components. Hence, we discuss only results obtained for the zero-correlated instance generated by us, but the same behavior was observed for the other instances.

5 Analysis of Experiments

Our goal is to identify fundamental differences between alternative components of multi-objective ACO algorithms, rather than obtaining the best configuration possible. For this purpose, we analyze the results by means of a graphical technique [12, 13] based on the empirical attainment function (EAF) [10].

The attainment function gives the probability of a particular point in the objective space vector being attained by (dominated by or equal to) the outcome of a single run of an algorithm. This probability can be estimated from several runs of an algorithm, in order to calculate the empirical attainment function (EAF) of an algorithm. The EAFs of two algorithms can be compared by calculating the difference in value of the EAFs for each point in the objective space. Differences in favor of one algorithm indicate that those points are more likely to be attained by that algorithm than by its competitor, and, hence, that the performance of that algorithm is better (in that region of the objective space) than the performance of its competitor. We display this information graphically by plotting side-by-side the positive and negative differences between the EAFs of two algorithms (or two different configurations of the same algorithm). Each point in the objective space obtained by either algorithm is colored according to the value of the difference. Points where this value is lower than 0.2 are not shown. The plots also display, as a continuous line, the grand best and the grand worst attainment surfaces, which respectively delimit the regions never attained by any algorithm and always attained by any run of the two algorithms. The median attainment surface of each algorithm, which delimits the region of the objective space attained by at least 50% of the runs, is also shown in each side of the plot as a dashed line.

Here we discuss the main effects of the components we have studied. We do so by discussing the results in the order of somehow decreasing impact, the most important component in this sense being the use of local search or not. For the other components, we always give results with and without local search, since sometimes there are quite strong interactions observable. In the following, we present the results only for the BTSP. We applied the same type of algorithms also to the BQAP using the same instances as in [9]; below we comment also shortly on the main findings we have for the BQAP and indicate possible differences from the BTSP results.

Component: Local Search The strongest difference is caused by the use of local search. The plots in Fig. 1 compare the same configuration of MOACO with and without local search. The results in the top plot were obtained by using one pheromone matrix, whereas the bottom plot shows results using two pheromone matrices (bottom). In both cases, local search clearly improves the results obtained by ACO. With two pheromone matrices, interestingly, the version without local search obtains solutions that are quite close to those of the version with local search at the extremes of the Pareto front but very poor results in the center. This result suggests that handling the trade-off between objectives is difficult for ACO. In the case of one pheromone matrix, the ACO algorithm tries to handle extreme and trade-off solutions at a same time, and, as a consequence, the results are rather poor along the whole Pareto front. A similar observation was made for the BQAP [9], where local search was also essential in order to obtain high quality solutions.

Component: Weight Setting Strategies Another notable observation is the difference in results obtained by the two weight setting strategies (all ants use the same weight at each iteration—TPLS-like strategy—versus each ant uses a different weight). The plots in Fig. 2 compare these two weight setting strategies with (top plot) and without local search (bottom plot). Although the benefit of the TPLS-like strategy is clearly stronger by a large margin if no local search is used, even in the case with local search there is an important difference along the whole Pareto front. By comparison, for the BQAP, there is no noticeable difference between weight setting strategies.

In our experiments, we consistently noticed that the aggregation of multiple pheromones by using weights, which is necessary for computing the probabilities of adding specific solution components, is an expensive operation. If each ant uses a different weight, this aggregation has to be performed for each of the m individual ant. On the other hand, if all ants use the same weight, the

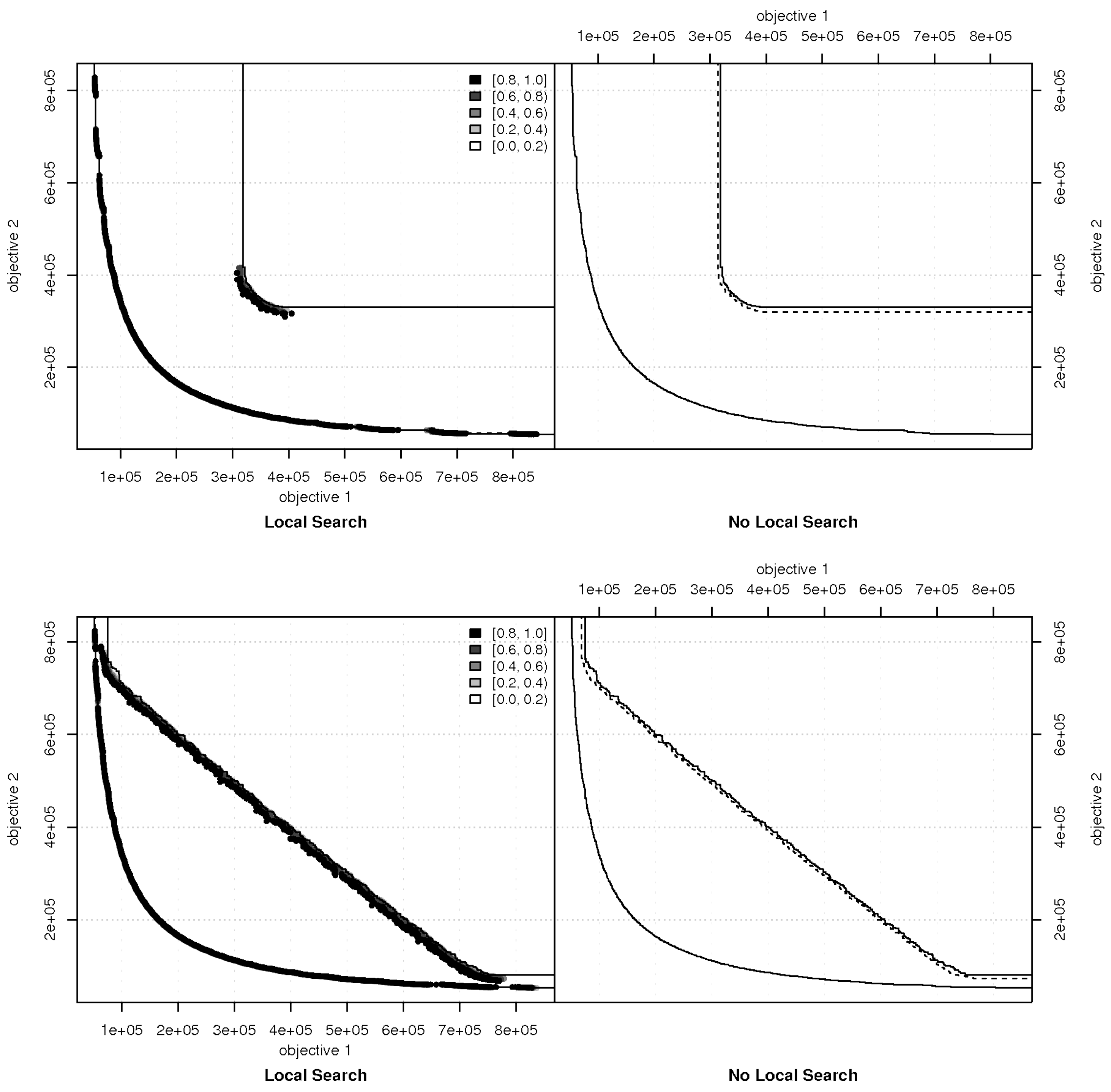

Figure 1: Differences in the EAFs with and without local search, either using one pheromone matrix (top) or two pheromone matrices (bottom), and always using one colony.

probabilities can be pre-computed once at each iteration. The latter approach allows to perform ten times more iterations than the former within the same computation time.

Component: One vs. Two Pheromone Matrices There were also strong differences in the results obtained when using one or two pheromone matrices. The bottom plot of Fig. 3 compares both approaches when not using local search. The first approach with only one pheromone matrix (plus update by non-dominated solutions) tends to produce better results in the center of the Pareto front than the approach using two pheromone matrices. In contrast, when using two pheromones matrices (plus weighted sum aggregations), the resulting Pareto front is much better at the extremes of the Pareto frontier. If local search is used, as shown in the top plot of Fig. 3, these structural differences disappear, due to the fact that the weighted local search approach does explore the whole Pareto front explicitly. Moreover, when using local search, the approach using two pheromone matrices is better along the whole Pareto frontier, and in particular in the center

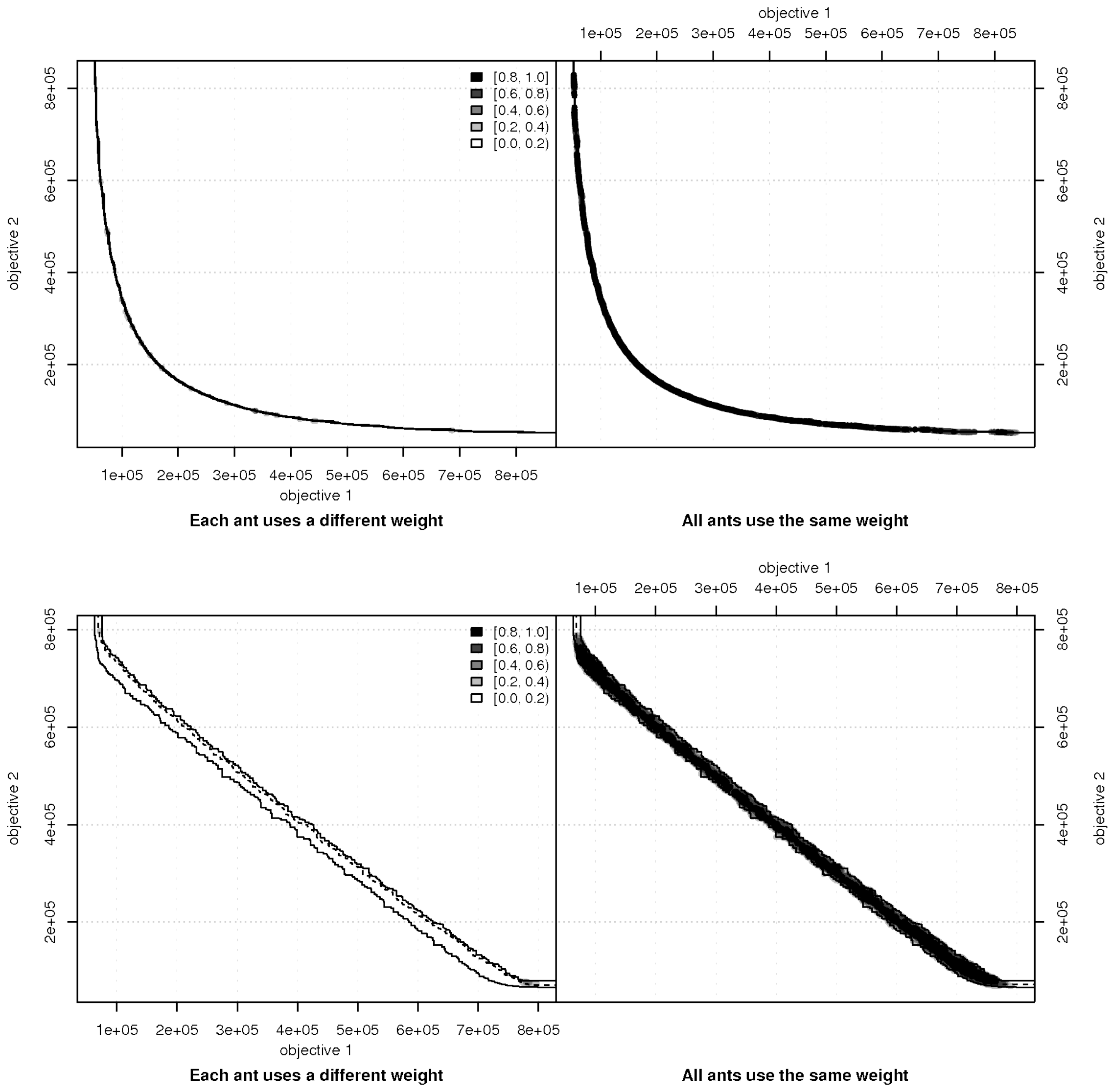

Figure 2: Differences in the EAFs between the two weight setting strategies. The algorithms use one colony and two pheromone matrices, and either local search (top) or not (bottom).

of the Pareto front. Hence, there is a strong interaction between the components concerning the pheromone matrices and local search.

Component: Number of Colonies We observed, as well, differences in the structure of Pareto frontiers obtained when using one or more colonies. Figure 4 compares the use of one or more colonies for the approach based on one pheromone matrix (top) or two pheromone matrices (bottom). As illustrated by the top plot, when using one pheromone matrix, the results obtained with three colonies are better in the extremes than with one colony. On the other hand, when using only one colony better results are obtained in the center of the Pareto front. The opposite behavior is observed in the bottom plot when using two pheromone matrices. In this case, three colonies improve over the results obtained by only one colony, except for the very extremes of the Pareto front.

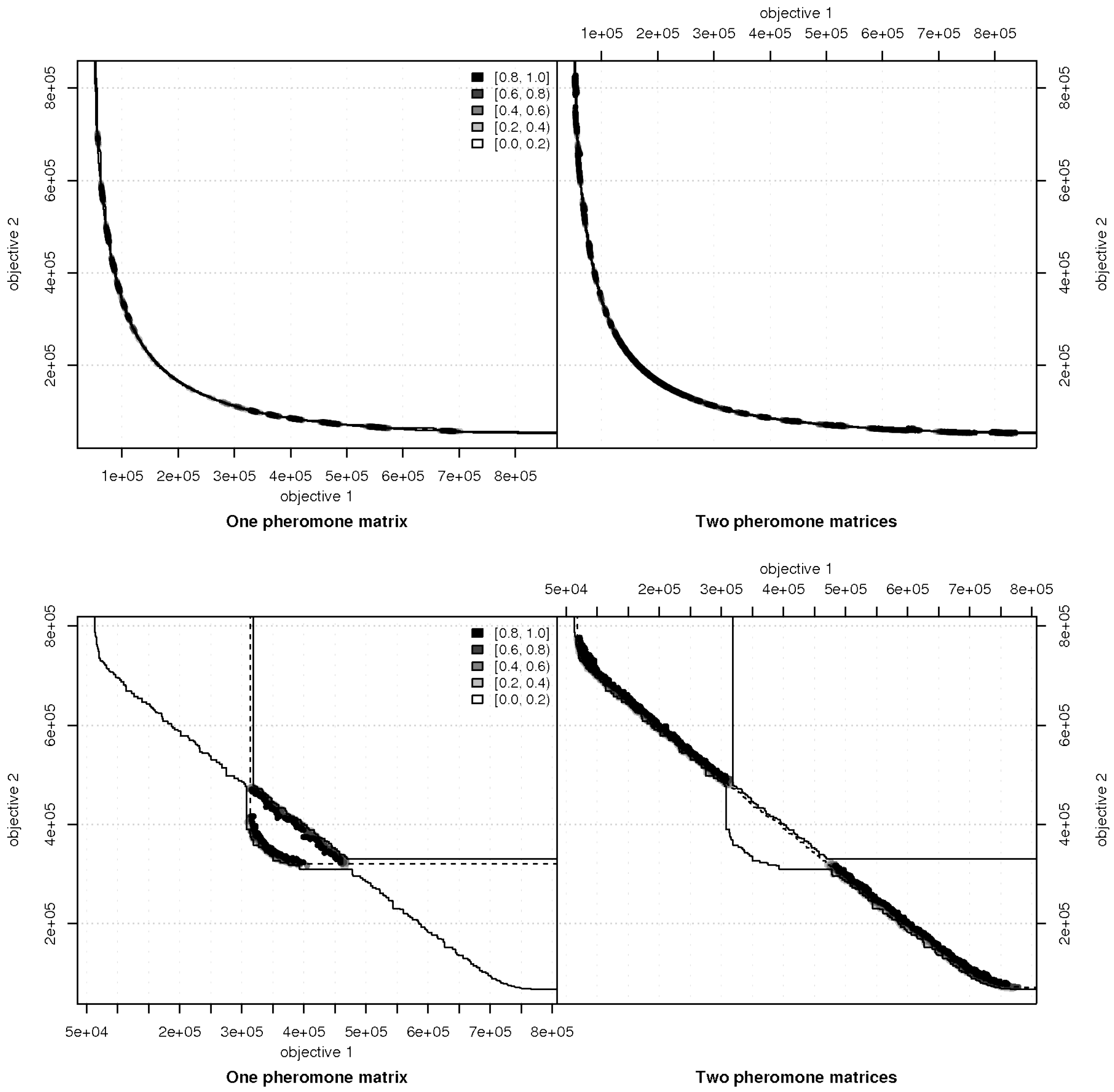

Figure 3: Differences in the EAFs between using one pheromone matrix or two pheromone matrices. The comparison is done with local search (top) and without (bottom), and using one colony.

Taking into account that the approach using one pheromone matrix is good at the center of the Pareto front but very poor at the extremes, the effect of the multiple colonies is to improve the results at the extremes. On the other hand, the approach using two pheromone matrices is very good at the extremes of the Pareto front, but quite poor at the center. In this case, the use of multiple colonies improves the results at the center. Hence, one may conclude that the effect of multiple colonies is to overcome one core weakness of each approach.

However, each additional colony introduces an overhead in computation time caused by the additional pheromone matrices. Since the available CPU time is limited in our experiments, the multiple colonies approach is not better along the whole Pareto frontier than a single colony, since each of the multiple colonies is doing less iterations than a single colony. The benefits of additional colonies were even stronger in the case of the BQAP [9], since the overhead introduced by each additional colony was smaller.

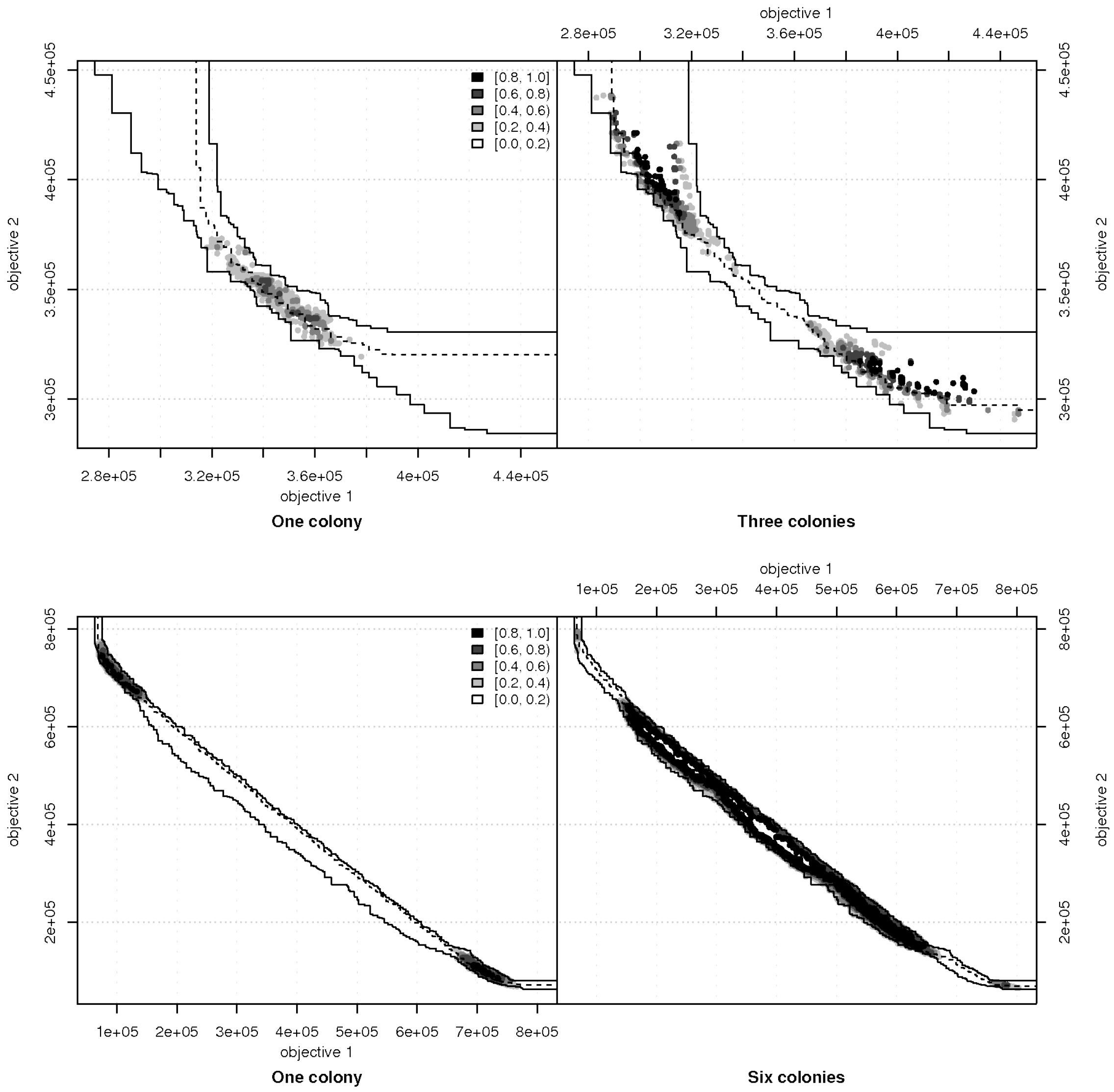

Figure 4: Differences in the EAFs between one and multiple colonies. Results are shown for one pheromone matrix (top) and two pheromone matrices (bottom).

6 Conclusions

In this paper, we have presented an analysis of several key algorithm components for the design of MOACO algorithms for the BTSP. We also have identified specific components, where different behavior is induced as when to compared to the BQAP. There are a number of ways of how this study can be extended. As next steps we plan to investigate the contribution of components of other concrete MOACO algorithms on performance both for the BTSP and BQAP. We also may consider comparisons to ACO algorithms that are embedded into higher-level guidance strategies such as two-phase local search [20]. We hope to have made a step forward towards an understanding of the impact of MOACO algorithm components on performance; ultimately we hope that this leads towards a more directed design of MOACO algorithms for practical applications.

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. Thomas Stützle acknowledges support from the Belgian F.R.S.-FNRS, of which he is a Research Associate.

References

[1] Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press (2004)

[2] Hoos, H.H., Stützle, T.: Stochastic Local Search—Foundations and Applications. Morgan Kaufmann Publishers, San Francisco, CA (2005)

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

[4] 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 **180**(1) (2007) 116–148

[5] Doerner, K., Gutjahr, W.J., Hartl, R.F., Strauss, C., Stummer, C.: Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. Annals of Operations Research (2003) to appear

[6] Barán, B., Schaerer, M.: A multiobjective ant colony system for vehicle routing problem with time windows. In: Proceedings of the Twentyfirst IASTED International Conference on Applied Informatics, Insbruck, Austria (2003) 97–102

[7] 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). Volume 1., Los Alamitos, CA, IEEE Computer Society (2007) 450–457

[8] Angus, D.: Population-based ant colony optimisation for multi-objective function optimisation. In: Progress in Artificial Life, Third Australian Conference, 2007. Volume 4828 of Lecture Notes in Computer Science., Springer Verlag, Berlin, Germany (2007) 232–244

[9] López-Ibáñez, M., Paquete, L., Stützle, T.: On the design of ACO for the biobjective quadratic assignment problem. In Dorigo, M., Gambardella, L.M., Mondada, F., Stützle, T., Birratari, M., Blum, C., eds.: ANTS'2004, Fourth International Workshop on Ant Algorithms and Swarm Intelligence. Volume 3172 of Lecture Notes in Computer Science., Springer Verlag (2004) 214–225

[10] Grunert da Fonseca, V., Fonseca, C.M., Hall, A.O.: Inferential performance assessment of stochastic optimisers and the attainment function. In Zitzler, E., Deb, K., Thiele, L., Coello, C.A., Corne, D., eds.: Evolutionary Multi-criterion Optimization (EMO 2001). Volume 1993 of Lecture Notes in Computer Science. Springer Verlag (2001) 213–225

[11] Shaw, K.J., Fonseca, C.M., Nortcliffe, A.L., Thompson, M., Love, J., Fleming, P.J.: Assessing the performance of multiobjective genetic algorithms for optimization of a batch process scheduling problem. In: Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99). Volume 1. (1999) 34–75

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

[13] López-Ibáñez, M., Paquete, L., Stützle, T.: Exploratory analysis of stochastic local search algorithms for biobjective problems through empirical attainment functions. In Bartz-Beielstein, T., Chiarandini, M., Paquete, L., Preuss, M., eds.: Empirical Methods for the Analysis of Optimization Algorithms. Natural Computing Series. Springer Verlag, Berlin, Germany (2009) To appear.

[14] Paquete, L., Stützle, T.: Clusters of non-dominated solutions in multiobjective combinatorial optimization: An experimental analysis. In: Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications. Volume 618 of Lecture Notes in Economics and Mathematical Systems., Springer Verlag, Berlin, Germany (2009) 69–77

[15] Lust, T., Jaszkiewicz, A.: Speed-up techniques for solving large-scale biobjective TSP. Computers & Operations Research (2009) In press.

[16] Hansen, M.P., Jaszkiewicz, A.: Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark (1998)

[17] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation **7** (April 2003) 117–132

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

[19] Doerner, K., Gutjahr, W.J., Hartl, R.F., Strauss, C., Stummer, C.: Ant colony optimization in multiobjective portfolio selection. In: Proceedings of the Fourth Metaheuristics International Conference. (2001) 243–248

[20] Paquete, L., Stützle, T.: A study of stochastic local search algorithms for the biobjective QAP with correlated flow matrices. European Journal of Operational Research **169**(3) (2006) 943–959

[21] Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In Giannakoglou, K., Tsahalis, D., Periaux, J., Papaliliou, K., Fogarty, T., eds.: Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems. Proceedings of the EUROGEN2001 Conference, International Center for Numerical Methods in Engineering (CIMNE) (2002) 95–100

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

[23] Stützle, T.: ACOTSP: A software package of various ant colony optimization algorithms applied to the symmetric traveling salesman problem (2002) `http://www.aco-metaheuristic.org/aco-code/`.