On the Design of ACO for the Biobjective Quadratic Assignment Problem

Manuel López-Ibáñez, Luis Paquete, and Thomas Stützle

Darmstadt University of Technology, Computer Science Department,
Intellectics Group, Hochschulstr. 10, 64283 Darmstadt, Germany

Abstract. Few applications of ACO algorithms to multiobjective problems have been presented so far and it is not clear how to design an effective ACO algorithms for such problems. In this article, we study the performance of several ACO variants for the biobjective Quadratic Assignment Problem that are based on two fundamentally different search strategies. The first strategy is based on dominance criteria, while the second one exploits different scalarizations of the objective function vector. Further variants differ in the use of multiple colonies, the use of local search, and the pheromone update strategy. The experimental results indicate that the use of local search procedures and the correlation between objectives play an essential role in the performance of the variants studied in this paper.

1 Introduction

Almost all the applications of Ant Colony Optimization (ACO) tackle problems for which solutions are evaluated according to only one objective [1]. However, many real-world problems involve multiple, often conflicting objectives and it is therefore highly desirable to extend the known best ACO techniques to tackle such problems. These multiobjective combinatorial optimization problems (MCOPs) replace the *scalar* value in single objective problems by an objective *vector*, where each component of the vector measures the quality of a candidate solution for one objective.

We consider MCOPs defined in terms of Pareto optimality. Therefore, we first have to introduce a dominance relation between objective vectors. Given two objective vectors u and v, $u \neq v$, we say that u *dominates* v if u is not worse than v for each objective and better for at least one objective. When neither u dominates v nor vice versa, we say that the two objective vectors are nondominated. (We use the same terminology among solutions as for the objective vectors.) The main goal then is to obtain a Pareto global optimum set, that is, a set of feasible solutions such that none of these solutions is dominated by any other feasible solution. As often done in the literature, we call *Pareto front* any set of nondominated objective vectors.

So far, few approaches of ACO algorithms to MCOPs defined in terms of Pareto optimality have been proposed [2,3,4]. (For a concise overview of ACO approaches to MCOPs we refer to [1].) In this article, we examine various of the possible implementation choices for ACO algorithms when applied to MCOPs defined in terms of Pareto optimality using the biobjective Quadratic Assignment Problem (bQAP) as an example application. We also consider instances of MCOPs for which we systematically modified the correlation between the two objectives from high positive to high negative correlation. On the algorithmic side, we examine the influence of two essentially different search strategies of how to tackle MCOPs. A first strategy uses scalarizations of the objective function vector into a single value, while the second uses dominance criteria between solutions. In addition, we test several ACO specific parameters like the number of colonies, the pheromone update strategy, and the influence of the usage of local search. The algorithms are evaluated in terms of the binary and unary ϵ-measures [5] and median attainment surfaces [6]. The main results are that (i) for the bQAP local search is essential to achieve high performance, and (ii) the best choice of parameters and methods for the algorithmic components strongly depend on the correlation between the objectives.

The article is organized as follows. Section 2 introduces the bQAP. Section 3 gives an overview of the ACO components we considered for the algorithm configurations that are described in Section 4. Finally, Section 5 gives details on the experimental results and we conclude in Section 6.

2 The Multiobjective QAP

The quadratic assignment problem (QAP) is a well-known $\mathcal{NP}$-hard problem [7], which can intuitively be described as the problem of assigning a set of facilities to a set of locations with given distances between each pair of locations and given flows between each pair of facilities. The goal is to place the facilities on locations such that the sum of the products between flows and distances is minimal [8].

The multiobjective QAP (mQAP) proposed by Knowles and Corne [9] uses different flow matrices, and keeps the same distance matrix. Given n facilities and n locations, a $n \times n$ matrix A where a_{ij} is the distance between locations i and j, and Q $n \times n$ matrices B^q, $q = 1, ..., Q$, where b^q_{rs} is the q^{th} flow between facilities r and s, the mQAP can be stated as follows:

$$\min_{\phi \in \Phi} \begin{cases} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b^{1}_{\phi_i \phi_j} \\ \vdots \\ \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b^{Q}_{\phi_i \phi_j} \end{cases} \tag{1}$$

where min refers to the notion of Pareto optimality. This problem arises in facilities layout of hospitals [9] and social institutions [10].

Local search algorithms for the biobjective QAP (bQAP) were presented in [11], where it was found that the best search strategy to be followed by the local search algorithms depended strongly on the correlation between the flow matrices (and, hence, objectives), which we denote as ξ. Based on these findings we also study the behavior of the ACO algorithms in dependence of ξ.

3 ACO algorithms for MCOPs

In the following, we discuss some of the main implementation alternatives for tackling MCOPs in terms of Pareto optimality with ACO that are additional to those present for single objective problems. Some of the concepts were already examined in the literature [2,3,4].

Multiple Colonies. In a multi colony approach, the total number of ants is divided into disjoint sets, each of these sets being called a colony. Multiple colonies have previously been used for the parallelization of ACO algorithms. In MCOPs, the main usage of multiple colonies of ants is to allow each colony to specialize on a particular region of the Pareto front [2]. In this case, each colony has its own pheromone information and an ant of a particular colony constructs solutions guided only by the pheromone information of its own colony. We will consider only a *cooperative* case where solutions can be exchanged among colonies, so that the decision of which solution updates the pheromone of a colony is affected by the solutions generated by the other colonies.

Pheromone Information. There exist two main alternatives to define the pheromone information. First, a single pheromone matrix can represent the desirability of the solution components with regard to all objectives [4]. In this case, the stochastic decision at each construction step is done as in single objective problems. The second option uses multiple pheromone matrices, where each of them represents the desirability of the solution components with respect to one objective. Stochastic decisions then may be made at each step according to only one objective, which may be randomly or deterministically chosen according to some criterion, or be based on aggregations of the different pheromone matrices using weights [2]. In the first case, the usual stochastic decisions of ACO algorithms can be made. The second case for the bQAP would be as follows. Given two pheromone matrices $[\tau_{ij}]$ and $[\tau'_{ij}]$, the probability of assigning a facility j to location i is

$$p_{ij}^{k} = \frac{\left[\tau_{ij}^{(1-\lambda)} \cdot {\tau'}_{ij}^{\lambda}\right]}{\sum\limits_{l \in \mathcal{N}_i^k} \left[\tau_{il}^{(1-\lambda)} \cdot {\tau'}_{il}^{\lambda}\right]} \qquad \text{if } j \in \mathcal{N}_i^k \tag{2}$$

where $\mathcal{N}_i^k$ is the feasible neighborhood of ant k, that is, those locations which are still free, and $\lambda \in [0,1]$ weights the relative importance of the two objectives. (In this formula we do not consider any heuristic information, which is anyway not used in state-of-the-art ACO approaches to the QAP [12]).

Weight Setting Strategies. Whenever we use multiple pheromone matrices that should be aggregated, *weights* need to be defined to regulate the influence of the individual terms. The usage of different strategies for setting the weights can then result in different search behaviors.

Iredi *et al.* [2] proposed that each ant uses a different weight (λ in Eq. 2), such that all weights are maximally dispersed in the interval $[0, 1]$. When multiple colonies are used, they proposed that each colony considers either the whole interval, disjoint subintervals, or overlapping subintervals. This weight setting strategy can be characterized as one of 'moving towards the Pareto front' in several directions at once [13]. A different strategy would be to assign all ants the same weight at each iteration and vary the weight between iterations. In this method, weights could be changed in such a way that the ants 'move along the Pareto front', trying to find in each iteration good solutions for a specific aggregation of objectives and then changing slightly the direction for the next iteration. These two strategies were also examined for local search procedures [11].

Pheromone Update Strategy. In single objective problems, the best performing ACO algorithms often use only the best solutions of each iteration (iteration-best strategy) or since the start of the algorithm (best-so-far strategy) for updating the pheromones [1]. One might expect that for MCOPs, similar pheromone update strategies may lead to very good performance. In the multiobjective case one can mimic *iteration-best* and *best-so-far* strategies by selecting among the set of solutions generated in the current iteration or since the start of the algorithm; however, it is more difficult to determine which are the best solutions to be chosen for the pheromone update.

Two different ways of implementing the pheromone update are possible. In *selection by dominance*, only nondominated solutions are allowed to deposit pheromone. An iteration-best strategy would consider the nondominated solutions among those generated in the current iteration; a best-so-far scheme would be obtained by choosing only solutions of an *archive* of the nondominated solutions found since the start of the algorithm. In the *selection by scalarization*, only the best solutions with respect to the scalarization chosen are used, either among the solutions found in the current iteration or since the start of the algorithm.

These possibilities can be applied, at least in principle, when using only one pheromone matrix or several pheromone matrices for each objective. In this paper, we restrict ourselves to a simple form of selection by scalarization that selects to update the pheromones of those solutions with the best value with respect to each of the two objectives. Moreover, when one pheromone matrix is used for each of the objectives, each pheromone matrix is updated by the solution from the candidate set with the best value for the respective objective. Therefore, in this case, only one ant per pheromone matrix will be allowed to deposit pheromone, as done in the best performing ACO algorithms.

A further, important factor to be considered in the pheromone update is whether one or multiple colonies are considered. While in the first case any of the above strategies may be applied, many additional possibilities arise in the latter case. For convenience, in this latter case we focus on the method called *update by region* [2], where first the nondominated ants are sorted according to one objective and then they are partitioned as equally as possible into a number of subsets equal to the number of colonies. Then, all solutions assigned to subset i are assigned to colony i.

Finally, one has to decide on the amount of pheromone to be deposited. In MCOPs, the solution cost cannot be used to define this amount. Therefore, all ants may deposit a same, constant amount of pheromone.

Local Search. For a large number of problems, ACO algorithms obtain the best performance when improving the solutions constructed by the ants through the use of local search procedures [1]. Additionally, it is frequently observed that the best parameter settings for ACO algorithms as well as the configuration of ACO algorithms (ie. how algorithmic features are applied and which features are useful at all) depends strongly on the interaction with the local search. Hence, if there exists the intuition that local search is important for the final performance, then local search should be considered from the first design phase.

We can roughly distinguish two classes of local search methods for MCOPs, one based on dominance criteria and the other based on scalarizations of the objective functions. For the bQAP, we implemented two local search algorithms. Pareto Local Search (PLS) [11,14] is a local search that iteratively adds non-dominated solutions to an archive whereas dominated solutions are discarded. The second local search is a single objective iterative improvement algorithm for a weighted sum scalarization of the objective function vector (W-LS). Both, PLS and W-LS use the same underlying 2-exchange neighborhood, where two solutions are neighbored if they differ in the location of exactly two facilities, and a best-improvement pivoting rule.

4 Configurations and Performance Assessment

Many different configurations of ACO algorithms can be designed mixing the concepts mentioned above. To limit the available choices for the underlying ACO algorithm, we decided to follow the rules of $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) [12] like the management of the pheromones (evaporation, pheromone trail limits, etc.), since $\mathcal{MM}$AS is known to be a state-of-the-art algorithm for the QAP. In our experimental evaluation, we focus on rather simple, straightforward ACO configurations that reflect the different types of search strategies available for MCOPs. The configurations we test are designed in such a way that they share some common principles and concepts to allow an understanding of what happens if some features of the configurations are varied. In this sense, our approach is very much as in experimental design and the goal is to systematically examine the influence of specific algorithm features on the final performance. In particular, we study the following features.

1. use of best-so-far vs. iteration-best pheromone update
2. use of one vs. multiple colonies
3. use of one pheromone matrix and a search strategy based on the dominance criterion (class D) vs. the use of multiple pheromone matrices and selection by scalarization (class S).

For algorithms of class D, the nondominated solutions generated in the current iteration (iteration-best) or since the start of the algorithm (best-so-far) are distributed among the colonies using update by region, and are then allowed to update the pheromone matrix of the corresponding colony. For algorithms of class S, after distributing the nondominated solutions among colonies, only the best solutions with respect to each objective is allowed to deposit pheromone in the respective pheromone matrix of each colony. (The same policy applies, independent of using only one colony or multiple colonies.)

For the class S algorithms we tested two ways of defining the weights for aggregating the pheromone matrices in the solution construction. The first is that every ant has a different weight and uses the same value in every iteration, denoted as S(all); the second is that all ants use the same weight in the current iteration but the weights are modified between the iterations, denoted as S(one). Finally, it must be mentioned that the algorithms tested were run with local search. In particular, for class D we use PLS, while in the case of class S we use W-LS for the local search. (Note that the local searches were chosen to match the main search strategy of either class D or class S.)

The different approaches are evaluated in terms of the unary ϵ-measure using a lower bound for the bQAP and the binary ϵ-measure [5]. The *unary ϵ-measure* gives the factor by which a Pareto front is worse than the Pareto global optimum set with respect to all objectives. Formally, given a Pareto front A and the Pareto global optimum set P, the unary ϵ-measure for Q objectives is defined as follows:

$$I_\epsilon(A, P) = \max_{p \in P} \min_{a \in A} \max_{q \in Q} \left(\frac{a^q}{p^q} \right) \tag{3}$$

For the QAP, typically P is not known and a lower bound is needed. We use a lower bound to P extending the Gilmore-Lawler lower bound [15] to the biobjective case. In the single objective case, this bound is given by the optimal objective value of an associated Linear Assignment Problem (LAP) (see [16] for more details). We defined a biobjective LAP (bLAP) where each objective in the bLAP is associated to an objective in the bQAP. Then we solve several scalarizations of the bLAP to obtain a set of solutions that are not dominated by the Pareto global optimum set. In order to have a set that dominates the Pareto global optimum set, we added points as follows. First, we sort the objective vectors lexicographically and for each successive pair of objective vectors u, v we added a point $w = (\min\{u_1, v_1\}, \min\{u_2, v_2\})$. For the instances considered here we used 5000 weight vectors maximally dispersed in $[0, 1]$.

The *binary ϵ-measure* is the pairwise version of the unary ϵ-measure. Given two Pareto fronts A and B, it considers the values of $I_\epsilon(A, B)$ and $I_\epsilon(B, A)$. If $I_\epsilon(A, B) \leq 1$ and $I_\epsilon(B, A) > 1$ then A is better than B, that is, at least one solution in A is not worse in all objectives than any solution in B and $A \neq B$.

We use the binary ϵ-measure to detect if the Pareto front returned by an ACO algorithm is better than the other. If no conclusion can be drawn from the binary ϵ-measure, we use the unary ϵ-measure with the lower bound as defined above. As a next step, we perform a ANOVA analysis [17] on the values returned by the unary ϵ-measure in order to detect which of the components contribute to the overall performance of ACO. Finally, in order to visualize the differences of performance in the objective space, we also plot the median attainment surfaces [6] of the Pareto fronts obtained by some configurations.

5 Experimental Setup and Results

The experimental setup considers three components: (*i*) the search strategy that can be of class D, S(all) or S(one); (*ii*) iteration-best versus best-so-far pheromone update; and (*iii*) one or multiple colonies. In addition, each algorithm was run with and without local search. The total number of ants (m) was equal to the instance size and we used 5 colonies in the multiple colony approach, where each colony had $m/5$ ants. For the management of the pheromones, we followed the rules of $\mathcal{MM}$AS with $\rho = 0.9$ for the pheromone evaporation; $p_{best} = 0.05$ to derive the factor between the lower and upper pheromone trail limits; and τ_{max} was set to the theoretically largest value [12].

The algorithms were tested on six symmetric bQAP instances, three *unstructured* and three *structured* instances. The three unstructured instances were taken from an earlier experimental study [11]. All instances were generated using the instance generator of Knowles & Corne [9] with size $n = 50$ and $\xi \in \{0.75,\ 0.0,\ -0.75\}$, where ξ is a parameter that influences the correlation between the flow matrices. The QAP specific parameter settings for generating the unstructured instances were set the same as used for generating instances of class `Taixxa` [18]; parameter settings for the structured instances where analogous to those for generating `Taixxb` [18] instances. The instances are available from `http://www.intellektik.informatik.tu-darmstadt.de/~lpaquete/QAP`.

Note that the correlations of the flow matrices also result in different correlations between the objective value vectors especially for the unstructured instances. For unstructured instances, the resulting correlations between the two objectives are of 0.90, -0.01, and -0.90 for the three values of ξ; in the structured case, there is not anymore a clear correlation betwen the objectives (the correlations between the objectives are $0.23, 0.03$, and -0.08, respectively), which is probably due to the many zero entries in the flow matrices. The empirical correlations between objectives were determined through samples of 1,000 random solutions generated for each instance.

Each of the algorithms was run 20 times on each instance for a maximum of 300 CPU-seconds. The algorithms were coded in C and the experiments were run on a Pentium III 933 MHz CPU and 512 MB RAM under Debian Linux.

As a first step in the analysis of the experimental results, every algorithm was compared against each other using the binary ϵ-indicator. This analysis clearly showed that for unstructured as well as structured instances any algorithm that uses local search outperforms all algorithms that do not use local search. Interestingly, the second phase of our analysis, which is described below, also suggested that the best ACO configurations that do not use local search are very different from the best configurations that do make use of local search procedures. This indicates that (i) studying ACO algorithms for MCOPs without local search and

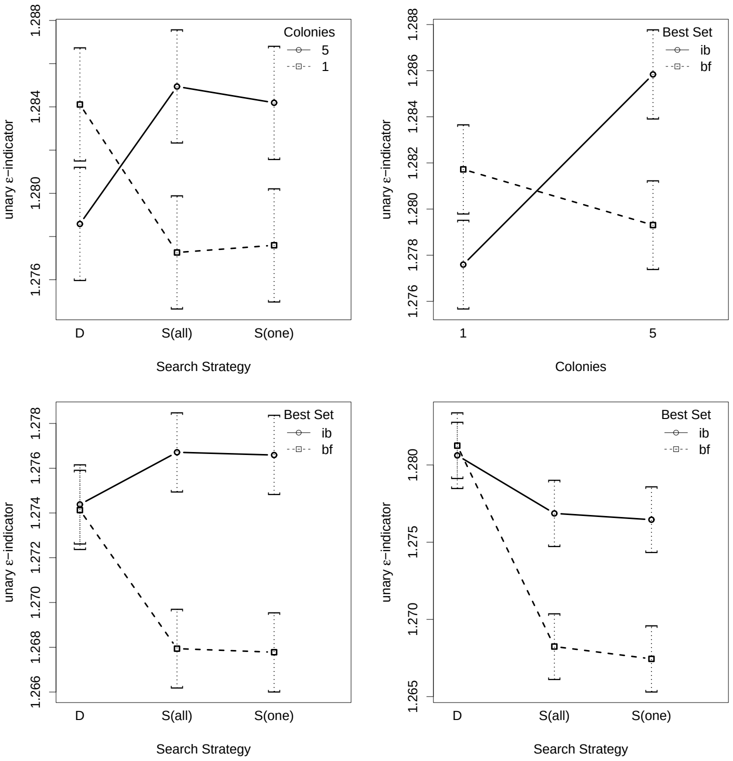

Fig. 1. Interaction plots for three unstructured instances of size 50 and with $\xi = \{$ 0.75 (top), 0.0 (bottom left), −0.75 (bottom right)$\}$. The search strategy can be based on dominance with PLS (D) or on scalarizations with W-LS, either in several directions (*S(all)*) or in one direction at each iteration (*S(one)*). The pheromone update strategy can consider the best solutions found in the current iteration (*ib*, iteration-best) or in all the previous iterations (*bf*, best-so-far). The number of colonies can be one or five

then simply adding local search leads to suboptimal performance and (ii) that local search may be, as it is true for the single objective case, also an essential component of ACO algorithms for MCOPs. In the following analysis, we only consider ACO configurations that use local search.

The results of the first phase of the analysis also showed that for structured instances the configuration of class D with PLS performs worse than the one based on class S with W-LS. The main reason is that PLS becomes unfeasible, because of an enormous number of nondominated solutions, which is much larger than for the unstructured instances, resulting in very high computation times for PLS. In addition, this first phase showed that for unstructured instances with $\xi = 0.75$, the algorithms of class S using five colonies and iteration-best pheromone update were outperformed by the other variants and that the variant of class D using one colony and best-so-far pheromone update was slightly worse than the other variants. For the other unstructured instances the outcomes of the algorithms were mainly incomparable in terms of dominance. Therefore, we proceeded to the second phase of our analysis based on ANOVA.

In the second phase, we evaluated each run of all the algorithms using the unary ϵ-indicator based on the lower bounds and we analyzed the results for the unstructured instances using a multiway ANOVA analysis after verifying that the ANOVA assumptions (homoscedasticity, independence of residuals, and normality of residuals) were satisfied [17]. Unfortunately, for structured instances this analysis was not possible, because some of the ANOVA assumptions were not met. Figure 1 shows the interactions that were found to be significant at the 0.05 level according to the ANOVA analysis (for unstructured instances). Interactions indicate that the value of one factor, that is, one particular algorithm component or parameter setting, depends on the other factors. In other words, the factor cannot be studied independent of the other factors. Additionally, the HSD Tukey intervals are plotted [17], allowing us to infer which combinations of parameter values are significantly different.

From the ANOVA analysis, we can conclude that the best setting of parameters depends on the value of ξ for generating the bQAP instances. In particular,

- for $\xi = 0.75$, the strategies based on scalarizations with a single colony and the one based on dominance with five colonies show the best performance. Also, a better performance was observed when iteration-best pheromone update was combined with a single colony than with five colonies;
- for $\xi = 0.0$ and $\xi = -0.75$, the strategy based on scalarizations and best-so-far pheromone update show a significantly better performance than the other combinations.

These general results suggest that for highly positively correlated instances it is better to use a less aggressive ACO strategy, considering that iteration-best pheromone update allows for a larger diversification than the best-so-far one. The same is true considering the number of colonies: using less colonies does not allow a too strong focus on specific regions of the Pareto front, hence leading to a higher exploration. On the weakly or negatively correlated instances this is not anymore true; here a much stronger exploitation of the search experience appears to be necessary.

Figure 2 shows the median attainment surfaces [6] for unstructured (left side) and structured (right side) instances using search strategy S with W-LS. In addition, we plotted reference solutions (given as points in Fig. 2) that were obtained by a scalarized version of the Robust Tabu Search algorithm (W-RoTS) [11], which gives a high quality approximation to the Pareto global optimum set. For W-RoTS we run as many scalarizations as possible in the given 300 CPU seconds; each run of the underlying tabu search algorithm (RoTS) was stopped after $100 \cdot n$ iterations (see [19] for details on RoTS). In the given CPU-time approximately 136 scalarizations could be run for W-RoTS. All solutions generated in such a way were then filtered to yield the given nondominated points. The result of this comparison is that on the unstructured instances, W-RoTS yields typically better performance than the ACO algorithms, while on the structured

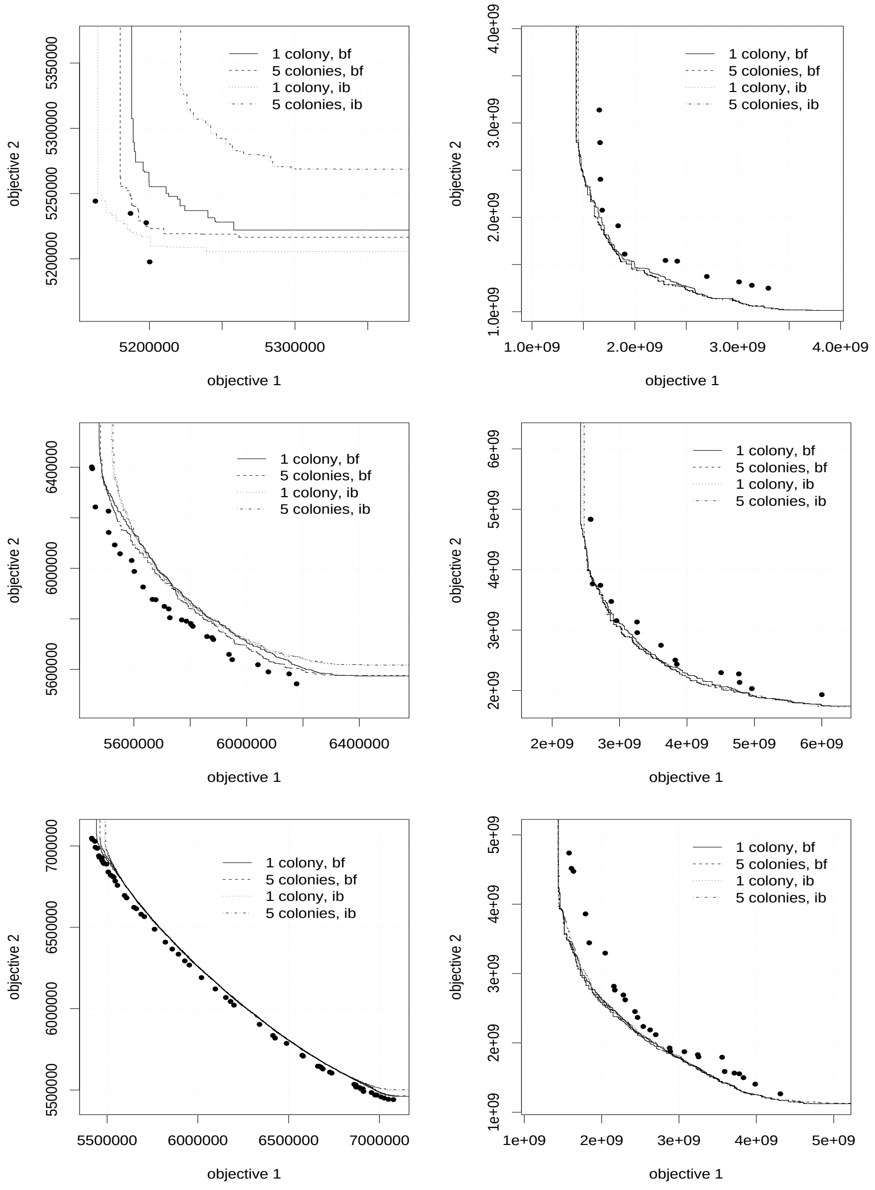

Fig. 2. Median attainment surfaces obtained for unstructured (left column) and structured (right column) instances of size 50 and ξ of 0.75 (top), 0.0 (center) and -0.75 (bottom) when using search strategy based on scalarizations with W-LS. In addition, the objective value vectors obtained by W-RoTS are plotted (points). (For details see text.)

instances, the median attainment surfaces of the ACO algorithms appear to dominate the result of W-RoTS. This result is somehow analogous to the single objective case, where $\mathcal{MM}$AS using a simple 2-opt local search performs worse than RoTS for unstructured instances, whereas the same algorithm outperforms RoTS for structured instances that occur frequently in applications [12].

6 Conclusions and Further Work

We have studied several alternative configurations of ACO algorithms for the bQAP. A first result is that, similar to the single objective case [1], the best performance on the bQAP is obtained by combining ACO with local search. Hence, because of interactions between local search and other parameter settings, for the design of a high performing ACO algorithm for the bQAP, the usage of local search has to be taken into account as the first step in the design process.

A second important observation is that algorithm performance varies strongly with the correlation of the objectives. In fact, depending on this correlation, there are significant differences in the configurations of the best performing algorithms. The tendency appears to be that for high positive correlations the search should not be too focused, while with decreasing correlation, a stronger exploitation of search experience, for example, through the usage of a best-so-far pheromone update strategy, becomes an important ingredient of any high performing ACO algorithm for the bQAP.

There are many possible ways of extending this research. First, the influence of the various possibilities of configuring ACO algorithms for MCOPs need to be studied also on other, differently structured problems. Additionally, more components useful for multiobjective ACO algorithms need to be explored. Furthermore, the exploration of fast and more effective local searches, for example, the use of RoTS instead of W-LS and bounded archiving techniques for PLS, is a highly promising direction to improve performance. Another interesting direction would be to use heterogeneous colonies at least for weakly or negatively correlated bQAP instances: two colonies based on scalarizations can explore the “tails” of the Pareto front while using one or more colonies with a search strategy based on dominance explore the center of the Pareto front.

Acknowledgments. We would like to thank Marco Chiarandini and Rubén Ruiz for the help with the statistical analysis. Luis Paquete and Thomas Stützle were supported by the Metaheuristics Network, a Research Training Network funded by the Improving Human Potential programme of the CEC, grant HPRN-CT-1999-00106. The information provided is the sole responsibility of the authors and does not reflect the Community’s opinion. The Community is not responsible for any use that might be made of data appearing in this publication.

References

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

2. Iredi, S., Merkle, D., Middendorf, M.: Bi-criterion optimization with multi colony ant algorithms. In Zitzler, E., et al., eds.: Proc. of EMO'01. Volume 1993 of Lecture Notes in Computer Science., Springer Verlag (2001) 359–372
3. Guntsch, M.G., Middendorf, M.: Solving multi-criteria optimization problems with population-based ACO. In Fonseca, C.M., et al., eds.: Evolutionary Multi-Criterion Optimization. Volume 2632 of LNCS., Springer Verlag (2003) 464–478
4. Doerner, K., Hartl, R.F., Reimann, M.: Cooperative ant colonies for optimizing resource allocation in transportation. In Boers, E.J.W., et al., eds.: Applications of Evolutionary Computing: Proc. of EvoWorkshops 2001. Volume 2037 of LNCS., Springer Verlag (2001) 70–79
5. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation **7** (2003) 117–132
6. da Fonseca, V.G., Fonseca, C.M., Hall, A.O.: Inferential performance assessment of stochastic optimisers and the attainment function. In Zitzler, E., et al., eds.: Proc. of EMO'01. LNCS 1993. Springer Verlag (2001) 213–225
7. Sahni, S., Gonzalez, T.: P-complete approximation problems. Journal of the ACM **23** (1976) 555–565
8. Çela, E.: The Quadratic Assignment Problem: Theory and Algorithms. Kluwer Academic Publishers (1998)
9. Knowles, J., Corne, D.: Instance generators and test suites for the multiobjective quadratic assignment problem. In Fonseca, C.M., et al., eds.: Proc. of EMO'03. LNCS 2632, Springer Verlag (2003) 295–310
10. Hamacher, H., Nickel, S., Tenfelde-Podehl, D.: Facilities layout for social institutions. In: Operation Research Proceedings 2001, Springer Verlag (2001) 229–236
11. Paquete, L., Stützle, T.: Fast stochastic local searches for the biobjective quadratic assignment problem. Technical Report AIDA-03-02, FG Intellektik, TU Darmstadt (2003)
12. Stützle, T., Hoos, H.H.: $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System. Future Generation Computer Systems **16** (2000) 889–914
13. Knowles, J., Corne, D.: Towards landscape analysis to inform the design of hybrid local search for the multiobjective quadratic assignment problem. In Abraham, A., et al., eds.: Soft Computing Systems: Design, Management and Applications, IOS Press (2002) 271–279
14. Paquete, L., Chiarandini, M., Stützle, T.: Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study. In Gandibleux, X., et al., eds.: Metaheuristics for Multiobjective Optimisation. LNEMS 535. Springer Verlag (2004)
15. Gilmore, P.C.: Optimal and suboptimal algorithms for the quadratic assignment problem. Journal of the SIAM **10** (1962) 305–313
16. Burkard, R.E., Çela, E., Pardalos, P.M., Pitsoulis, L.S.: The quadratic assignment problem. In Pardalos, P.M., Du, D.Z., eds.: Handbook of Combinatorial Optimization. Kluwer Academic Publishers (1998) 241–338
17. Dean, A., Voss, D.: Design and Analysis of Experiments. Springer (1999)
18. Taillard, É.D.: A comparison of iterative searches for the quadratic assignment problem. Location Science **3** (1995) 87–105
19. Taillard, É.D.: Robust Taboo Search for the Quadratic Assingnment Problem. Parallel Computing **17** (1991) 443–455