Journal of Mathematical Modelling and Algorithms (2006)
DOI: 10.1007/s10852-005-9034-x

Hybrid Population-Based Algorithms for the Bi-Objective Quadratic Assignment Problem

MANUEL LÓPEZ-IBÁÑEZ[1,★], LUÍS PAQUETE[2,★★] and THOMAS STÜTZLE[3,★★]

[1] *Centre for Emergent Computing, School of the Built Environment, Napier University, 10 Colinton Rd, Edinburgh, EH10 5DT, UK. e-mail: m.lopez-ibanez@napier.ac.uk.*

[2] *Faculdade de Economia and Centro de Sistemas Inteligentes, Universidade do Algarve, 8005-139 Faro, Portugal. e-mail: lpaquete@ualg.pt.*

[3] *Université Libre de Bruxelles, IRIDIA, Avenue Franklin Roosevelt 50, 1050 Brussels, Belgium. e-mail: stuetzle@iridia.ulb.ac.be*

Abstract. We present variants of an ant colony optimization (MO-ACO) algorithm and of an evolutionary algorithm (SPEA2) for tackling multi-objective combinatorial optimization problems, hybridized with an iterative improvement algorithm and the robust tabu search algorithm. The performance of the resulting hybrid stochastic local search (SLS) algorithms is experimentally investigated for the bi-objective quadratic assignment problem (*b*QAP) and compared against repeated applications of the underlying local search algorithms for several scalarizations. The experiments consider structured and unstructured *b*QAP instances with various degrees of correlation between the flow matrices. We do a systematic experimental analysis of the algorithms using outperformance relations and the attainment functions methodology to asses differences in the performance of the algorithms. The experimental results show the usefulness of the hybrid algorithms if the available computation time is not too limited and identify SPEA2 hybridized with very short tabu search runs as the most promising variant.

Mathematics Subject Classifications (2000): 68W20, 68T20, 90B80, 90C27, 90C29.

Key words: multi-objective quadratic assignment problem, hybrid algorithms, stochastic local search, ant colony optimization, evolutionary algorithms, tabu search.

1. Introduction

Population-based stochastic local search algorithms are widely used for tackling many combinatorial optimization problems. Although single objective problems are most commonly studied, a recent trend is to apply such algorithms also to

★ Corresponding author.

★★ This research was mainly done while Luís Paquete and Thomas Stützle were with the Intellectics Group at the Computer Science Department of Darmstadt University of Technology, Germany.

combinatorial problems that involve multiple objective functions [12–14, 30]. Most approaches make use of either no local search algorithm at all, or of rather simplistic local searches based on some form of iterative improvement algorithm. Only few approaches exist that combine population-based algorithms for multi-objective combinatorial problems (MCOPs) with other stochastic local search (SLS) methods, although for several single objective problems this is known to enhance performance considerably [4, 5, 26].

In this article, we study the performance of such hybrids. In particular, we experimentally study the addition of a simple iterative improvement algorithm and short runs of a tabu search algorithm to an ant colony optimization (ACO) algorithm [18] and to SPEA2 [29]. These algorithms are compared to the underlying local search algorithms applied to scalarizations of the problem for assessing the contribution of the population-based algorithms. By using different local search algorithms for hybridization, we can trade efficiency (that is, speed) of the local search algorithm for efficacy (that is, solution quality). While a simple iterative improvement algorithm is typically very quick in identifying local optima, short runs of a tabu search algorithm typically require more computation time but result in solutions of higher quality. By considering these alternative possibilities for hybridization, we can examine the relative importance of these features for multi-objective optimization problems. In fact, issues of the design of hybrid algorithms may differ from single-objective to multi-objective optimization problems, because of inherent differences in the nature of the solutions searched for – a single solution in the first case, a set of solutions in the second case.

This experimental study is based on the bi-objective quadratic assignment problem (*b*QAP) [16]. This problem is chosen because (*i*) it is well-known that the single objective QAP is considered to be one of the most difficult combinatorial optimization problems to solve, and (*ii*) significant knowledge of the various SLS methods exists for the single objective case. This study also makes use of two differently structured classes of randomly generated instances: Those where the data is generated using uniform distributions, and those where the data distribution is close to that of real-world instances, which is strongly non-uniform. Our experimental results clearly show that the structure of the instance tackled and, to some extent, the correlation of the instance data (and, hence, the objective functions) have a strong influence on the relative performance of the hybrid algorithms.

An additional innovative feature of our experimental comparison is the systematic use of a sound methodology for evaluating and comparing the performance of multi-objective SLS methods. In the first place, we use information on a specific outperformance relation [9] between the outcomes of SLS algorithms. If by these means no clear conclusions can be made on the relative performance of the algorithms, we examine in a next step the attainment functions of the SLS algorithms' outcomes [7]. This is a very useful tool to clearly identify regions in the objective space, where the performance of SLS algorithms differs significantly. It is important to note that the usage of dominance relations and attainment functions in empirical evaluation does not entail the weaknesses of many unary performance measures identified in [31].

The article is organized as follows. In the following section we introduce some basic concepts needed for the rest of the paper and Section 3 presents details on the multi-objective QAP. Section 4 presents the local search algorithms, the ACO algorithm and SPEA2. Next, we introduce the experimental methodology used and discuss the experimental results in Section 6. Finally, Section 7 gives some concluding remarks and outlines directions for future research.

2. Notations and Definitions

Let U and V be vectors in $\mathbb{R}^n$. We denote component-wise order by $U \prec V$, that is, $U \neq V$ and $u_i \leq v_i,\ i = 1, \ldots, n$, and weak component-wise order by $U \leq V$, that is, $u_i \leq v_i,\ i = 1, \ldots, n$. We consider the following general multi-objective program

$$\min_{s \in \mathcal{S}} \begin{cases} p^1 = F^1(s) \\ \quad \vdots \\ p^Q = F^Q(s) \end{cases} \tag{1}$$

where $\mathcal{S}$ is the set of feasible solutions to the given problem instance, Q is the number of objectives, $F^q(s)$ are real-valued functions, and 'min' is understood in terms of Pareto optimality. We say that $F(s) = \left(F^1(s), \ldots, F^Q(s)\right)$ is the *objective function vector* of s.

In the context of optimization, we denote the relation between objective value vectors of two feasible solutions as follows (this relation is often called dominance relation). If $F(s) \prec F(s')$, we say that $F(s)$ *dominates* $F(s')$ and if $F(s) \leq F(s')$, then $F(s)$ *weakly dominates* $F(s')$. Finally, we also say that $F(s)$ and $F(s')$ are *nondominated* if $F(s) \nprec F(s')$ and $F(s') \nprec F(s)$, and are *non-weakly dominated* (*nwd*) if $F(s) \nleq F(s')$ and $F(s') \nleq F(s)$. Note that the latter notion implies that $F(s) \neq F(s')$. We use the same relations also among solutions if they hold between their corresponding objective value vectors.

In order to compare sets of solutions in the objective space, we use relations among sets of objective vectors as proposed by Hansen and Jaszkiewicz [9] and Zitzler et al. [31]. For our purposes, we use the weak outperformance relation [9], also called *better* in [31]; this relation is denoted by the symbol $\triangleleft$ in the following. Given two sets A and B of nondominated points in a Q-dimensional objective space, we have $A \triangleleft B$ if every $b \in B$ is weakly dominated by at least one $a \in A$ and $A \neq B$, i.e., we say A *is better than* B.

Finally, we define the weighted sum scalarization of the multi-objective program (1) as

$$\min_{s \in \mathcal{S}} \sum_{q=1}^{Q} \lambda^q F^q(s), \tag{2}$$

such that λ^q is the q-th component of the weight vector λ taken from the set of weight vectors

$$\Lambda = \{\lambda \in \mathbb{R}^Q : \lambda^q \geq 0, \sum_{q=1}^{Q} \lambda^q = 1,\ q = 1, \ldots, Q\}. \tag{3}$$

In fact, all the algorithms that we are studying in this article tackle the bi-objective QAP using a series of such scalarizations using appropriately chosen strategies for modifying the weight vector.

3. The Multi-Objective QAP

The quadratic assignment problem (QAP) is a well-known $\mathcal{NP}$-hard problem [23], 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 [2].

Knowles and Corne [16] proposed a multi-objective QAP (*m*QAP) which uses different flow matrices, and keeps the same distance matrix. This problem arises in facilities layout of hospitals [16] and social institutions [8]. Here, we focus on the bi-objective variant (*b*QAP), which is defined by n facilities and n locations, a $n \times n$ matrix A where a_{ij} is the distance between locations i and j, and two $n \times n$ matrices B^1 and B^2 where b^1_{rs} is the first flow and b^2_{rs} is the second flow between facilities r and s. Then, the *b*QAP can be formulated as

$$\min_{\phi \in \Phi} \begin{cases} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{\phi(i)\phi(j)} b^1_{ij} \\ \sum_{i=1}^{n} \sum_{j=1}^{n} a_{\phi(i)\phi(j)} b^2_{ij} \end{cases} \tag{4}$$

where Φ is the set of all possible permutations of the set of integers $\{1, 2, \ldots, n\}$, $\phi(i)$ gives the location assigned to facility i in the current solution $\phi \in \Phi$ and 'min' refers to the notion of Pareto optimality.

Using a weight vector λ, we can transform the objective vector of a *b*QAP into a single objective scalar value as defined by Equation (2). It is known that an optimal solution to this *scalarized* problem belongs to the Pareto global optimum set of the corresponding MCOP if the weights are positive. However, there may exist Pareto optimal solutions which are not optimal for any weight vector λ [24].

4. Stochastic Local Search Methods for the bQAP

The local search methods we applied make use of the scalarization given in Equation (2). We implemented two local search algorithms, a simple iterative improvement algorithm and a tabu search algorithm. Both algorithms are considered for hybridization with two population-based algorithms, a multi-objective ACO (MO-ACO) algorithm and the second version of the strength Pareto evolutionary algorithm (SPEA2). These algorithms are described in the following.

4.1. LOCAL SEARCH ALGORITHMS

The iterative improvement algorithm and the tabu search algorithm make use of a two-exchange neighborhood, where two solutions ϕ and ϕ' are neighbored, if they differ in the location assigned to two facilities, that is, $N(\phi, \phi') \Leftrightarrow \exists r, s : (r \neq s \wedge \phi'(r) = \phi(s) \wedge \phi'(s) = \phi(r) \wedge \forall i \notin \{r, s\} : \phi'(i) = \phi(i))$. This neighborhood relation is chosen in almost all local search algorithms for the QAP. Our implementation of both algorithms uses the fast evaluation of neighboring solutions described by Taillard [27].

Iterative improvement. The iterative improvement (II) algorithm searches, at each step, for the best improvement upon the current solution according to the solution quality defined by Equation (2) and a given value for λ. It stops as soon as no better neighbor can be identified.

Robust tabu search. Robust tabu search (RoTS) [27] is one of the best performing tabu search algorithms for the single objective QAP. RoTS chooses the best non-tabu neighboring solution at each local search step. A neighboring solution is tabu if both facilities r and s involved in the respective two-exchange move would become assigned to a location that they occupied in the most recent tt iterations. RoTS also uses an aspiration criterion that overrides the tabu status of a move if it would lead to a new best solution. Additionally, every $2.2n$ iterations, tt is assigned a randomly chosen value within $[0.9n, 1.1n]$. A single run of the RoTS algorithm is stopped after $\ell \cdot n$ iterations, where n is the size of the particular instance and ℓ is a parameter.

Both algorithms, II and RoTS, are considered for hybridization with the population-based SLS algorithms described below. In addition, we tested the two local searches as independent SLS methods, where we run as many different scalarizations as possible within a CPU-time limit. Each scalarization uses a different weight, taken from a set of available weights. The weights are generated by recursively partitioning the interval $[0, 1]$ into smaller subintervals; all subintervals at the same level of the recursion have equal length. Each point where an interval was split into two equal length subintervals is added to the set of available weights. The weights are generated in this way to reach a dispersed set of nondominated objective vectors. Upon termination of the main search process, all solutions returned are filtered to obtain a set of nondominated objective vectors. The two resulting SLS algorithms are called W-II and W-RoTS, respectively. (Note that W-II was also used in [15] for a landscape analysis of the multi-objective QAP, while W-RoTS was used in [22] for generating reference solutions for some *b*QAP instances.) Earlier experiments show that W-RoTS gives a high quality approximation to the optimal Pareto set on unstructured instances of the *b*QAP [18, 22].

4.2. MO-ACO

Ant colony optimization (ACO) is a SLS method for solving combinatorial problems that is inspired by the pheromone-trail-laying behavior of some real ant species. It is essentially a stochastic construction method, where a colony (population) of artificial ants generates solutions to the problem under consideration and the ants communicate indirectly by means of artificial pheromone trails.

The multi-objective ACO (MO-ACO) approach considered in this study uses multiple pheromone information, that is, one pheromone matrix for each objective [11]. A pheromone trail value τ_{ij} in matrix q indicates the desirability of assigning facility i to location j with respect to objective q. This approach was found to achieve high performance in a comparison of various MO-ACO approaches for the *b*QAP [18]. Whenever an ant constructs a solution, the pheromone information for the different objectives is aggregated using a weight vector [11]; for the *b*QAP, ant k would assign a facility i to location j with a probability

$$p_{ij}^k = \frac{\left[\tau_{ij}^{\lambda^1} \cdot \tau_{ij}'^{\lambda^2}\right]}{\sum_{l \in \mathcal{N}_i^k} \left[\tau_{il}^{\lambda^1} \cdot \tau_{ij}'^{\lambda^2}\right]} \qquad \text{if } j \in \mathcal{N}_i^k \tag{5}$$

where $\mathcal{N}_i^k$ is the feasible neighborhood of ant k, that is, those locations which are still available, and $\lambda = (\lambda^1, \lambda^2)$, where λ^1 and λ^2 follow Equation (3). We do not consider any heuristic information, since it is also not used in state-of-the-art ACO algorithms for the single objective QAP [25, 26].

There are different ways to define the weight vector. Here, we follow the proposal of Iredi et al. [11], which is to use for each ant a different vector λ such that all values are maximally dispersed in the interval $[0, 1]$. The weight vectors assigned to each ant do not vary through the execution of the algorithm since weight modifications were not found to be useful for the *b*QAP in earlier experiments [18].

Each pheromone matrix is updated by the solution with the best value for the corresponding objective. Hence, only one solution per pheromone matrix will be allowed to update the pheromone information, and this solution may be taken either from the set of nondominated solutions generated in the current iteration, *iteration-best* (*ib*) strategy, or from the set of nondominated solutions since the start of the algorithm, a *best-so-far* (*bf*) strategy [17]. This pheromone update mimics the typical pheromone update used in the best performing ACO algorithms for single objective problems [3]. The amount of pheromone that each ant deposits is set to a fixed, constant amount of pheromone.

For the MO-ACO algorithm studied here, we also consider the usage of multiple colonies; the number of colonies is denoted by c. If more than one colony is used, each colony uses its own pheromone information and an ant constructs solutions guided only by the pheromone information of its own colony. If $c > 1$, the interval $[0, 1]$ is divided into c equal length subintervals, where the weight vectors of neighboring subintervals overlap by 50%. For each colony the weight vectors in its subinterval are again maximally dispersed. The colonies *cooperate* by using the method called *update by region* [11], where the nondominated solutions of the ants are first sorted according to the first objective before being partitioned as equally as possible into a number of subsets equal to the number of colonies. Then, all solutions assigned to a specific subset i are assigned to colony i, $i = 1, \dots, c$. This sorting takes into account the order of the colonies imposed by the weight intervals: The first colony contains those ants with the lowest values of λ^1, while ants of the last colony have the highest values of λ^2; hence, the first colony is more likely to generate good solutions for the first objective, while with increasing λ^2 the quality of the solutions with respect to the second objective will increase. Once the ants are assigned to the colonies, the best ants with respect to the two objectives update the corresponding pheromone matrix. The remainder of the ACO part of the MO-ACO algorithm follows the rules of $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) [26], which is known to be a state-of-the-art ACO algorithm for the QAP.

The MO-ACO algorithm is combined with the local search algorithms presented in the previous subsection. In this hybridization, each of the ants is improved by the respective local search algorithm. The weight vector of the local search algorithm is exactly the same that was used by the ant that generated the starting solution. If the local search improves over the initial solution, the best solution with respect to the given scalarization is added to the archive of iteration-best solutions. The final set of solutions returned by the algorithm is the set of nondominated solutions found.

4.3. SPEA2

SPEA2 [29] is an evolutionary algorithm with the following features: (i) the fitness of an individual depends on the strength of the individuals by which it is dominated, where the strength of an individual is defined to be the number of other individuals in the current population it dominates; (*ii*) if there are several individuals with the same fitness, ties are broken using a nearest neighbor density estimation technique; (*iii*) the size of the archive of nondominated solutions is fixed to a value α and the nondominated individuals in excess of this value are discarded using a truncation operator which preserves boundary solutions. For details on SPEA2 we refer to the original publication [29].

The search in SPEA2 is organized as follows. SPEA2 starts from an initial population, that is generated according to heuristics for the underlying problem or randomly, and with an empty archive. Then, the fitness of individuals in the current population is calculated and all nondominated individuals are added to the archive. If the resulting archive size is larger than a parameter α, the individual which has the minimum distance to another individual is discarded until archive size is exactly α; when the number of nondominated individuals is less than α, the dominated individual with the minimum fitness value is added to the archive until there are α solutions in the archive. Next, μ individuals are selected as parents using binary tournament selection with replacement. Genetic operators, which have to be defined according to the underlying problem being tackled, are then applied to the parents in order to generate μ new individuals.

Our implementation of SPEA2 is based on the original source code available from the PISA project web-pages at `http://www.tik.ee.ethz.ch/pisa/` [1], but we refined the code by introducing obvious speed-ups to make it more useful for our purposes. For the problem specific part, we use the CX recombination operator described in [19, 20], which is similar to the cycle recombination operator. CX was previously shown to outperform several other recombination operators for the single-objective QAP [19]. In the hybrid SPEA2 algorithm proposed here, every individual of the initial population or generated by recombination is improved using local search, either iterative improvement or RoTS, over a scalarization of the objective function; the weights for scalarization are given by the weight vector $\left(\frac{\mu-w}{\mu}, \frac{w}{\mu}\right)$, where w is the index of the individual that is generated by recombination and, hence, $1 \leq w \leq \mu$. If there is any improvement, the improved solution is added to the current population.

4.4. HYBRID ALGORITHMS

As previously noted, MO-ACO and SPEA2 were both combined with either the II algorithm or short runs of RoTS of length $\ell \cdot n$, $\ell \in \{1, 5, 10\}$. By using different lengths of the tabu search runs (or alternatively iterative improvement), we can trade somehow search exploitation *versus* search exploration: The longer the runs of RoTS, the higher the average quality of solutions returned; however, longer RoTS runs require more computation time and, hence, less iterations of the population-based algorithm can be run. Thus, the tradeoff incurred can be summarized as whether it is preferable to have a local search that, on average, returns high quality solutions but requires long computation times, or whether it is preferable to have a less effective but faster local search. One central aim of the experimental study reported below is to explore this issue and relate the answer to instance characteristics such as structure and correlation between objectives.

For the following we use the local search settings as indices to MO-ACO and SPEA2 to name the hybrid algorithms. For example, we use MO-ACO$_{\mathrm{II}}$, MO-ACO$_n$, MO-ACO$_{5n}$, and MO-ACO$_{10n}$ to name the variants of MO-ACO that use II and RoTS runs of n, $5 \cdot n$ and $10 \cdot n$ iterations, respectively.

5. Performance Assessment Methodology

In this article, we experimentally compare the performance of SLS algorithms for the bQAP. Since the performance of the algorithms depends on parameter settings or discrete choices in their configuration, such as whether II or RoTS is chosen for the local search, at the first stage, each of the algorithms needs to be appropriately tuned. Once we know parameter settings and algorithm configurations that result in good performance, we compare the various algorithms under consideration. Following this approach, on the one hand our goal is to compare peak performance of the different hybrid algorithms. On the other hand, we know that instance features like correlation between the flow matrices can have significant influence on the algorithms' performance [22]. Hence, a second goal of this study is to provide insight into how these features affect design choices for the hybrid algorithms.

Often, this type of approach is reasonably straightforward in the case of single-objective optimization. However, it can be a considerable challenge when tackling multi-objective problems. The main reason is that in the latter case the outcomes of the algorithms are *sets* of nondominated objective vectors, and, even if we were analyzing deterministic algorithms, the best possible way of comparing the outcomes remains unclear. In fact, the results of Zitzler et al. [31] show that one should not expect that the outcome of an algorithm for multi-objective optimization can be summarized in one single value. Therefore, the use of simple statistics such as means and standard deviations of *unary measures* (or combinations thereof) must be interpreted very carefully if it is to give accurate information about the relative performance of algorithms.

We base our experimental analysis on two tools that do not have the known disadvantages of unary performance measures. In a first step, we compare the outcomes of pairs of algorithms with respect to the $\triangleleft$-relation introduced in Section 2. Since our algorithms are stochastic and we run each one multiple times, we count how many times an outcome of an algorithm A weakly dominates an outcome of an algorithm B on one instance and vice-versa; in fact, if each of the algorithms is run r times, we do r^2 comparisons between all possible pairs of outcomes of the algorithms. If the pair of outcomes considered does not weakly dominate each other, i.e., they are incomparable or equal, we do not count such cases. The result of this step will be two numbers, the percentage of comparisons in which the outcomes of A are better than B and the percentage of comparisons in which the outcomes of B are better than A.

If this first step shows no clear advantage of one configuration/algorithm over the other, we have a strong indication that the outcomes could be incomparable most of the time (we do not expect the algorithms to produce equal outcomes, given the stochasticity involved). Therefore, we are now interested in knowing *where* these outcomes differ and *how large* this difference is. This is exactly the information that can be provided by attainment functions that represent the probability that an arbitrary goal in the objective space is attained during a single run of the algorithm [7]. This probability can be estimated using data collected from several independent runs in a way analogous to estimating empirical distributions of one-dimensional random variables; this estimation process leads to the definition of the *empirical attainment function* (EAF) [7]. The EAF can be seen as a distribution of the solution quality (here as a set of nondominated objective value vectors obtained from each run) after running an algorithm for a specific amount of computation time; it can be seen as an extension of the solution quality distribution in the context of performance assessment of SLS algorithms for single objective case [10]. Here, the EAFs are used mainly for giving visual information on the pairwise performance of two SLS algorithms or two configurations of one SLS algorithm by plotting the location of the differences with respect to their EAFs [21]. Given the outcomes of two algorithms/configurations A and B, we only indicate explicitly areas in the objective space, where the difference in the EAFs of the two outcomes is above a value of 0.2. Since the sign of the differences gives information about which algorithm/configuration performed better at that point, we plot positive and negative differences separately. In addition, the size of the differences between the EAFs of A and B are encoded using different shades of grey: The darker a point, the larger is the difference.

Figure 1 illustrates the performance assessment method used in this article by means of EAFs. The two plots in the top part of Figure 1 give the EAFs associated to two algorithms that were run several times in one instance. The lower line on each plot connects the best set of points attained over all the runs of that algorithm, while the upper line connects the sets of points attained by any of those runs. The bottom plots show the location of the differences between the EAFs of the two algorithms; in more detail, on the left are shown those regions where the EAF of Algorithm 1 is larger by at least 0.2 than that of Algorithm 2 and on the right the differences in the opposite direction (positive differences between the EAFs of Algorithm 2 over the one of Algorithm 1). In addition, the lower line on both plots connects now the best set of points attained over all the runs of both algorithms and the upper one the set of points attained by any of that runs. The amount of the differences is encoded in a grey-scale scale also shown in Figure 2. We can clearly observe that Algorithm 1 performs better in the center, while Algorithm 2 performs better towards high quality solutions for the second objective (low values on the y-axis). As can be seen from this example, these plots are useful for identifying differences of performance between al-

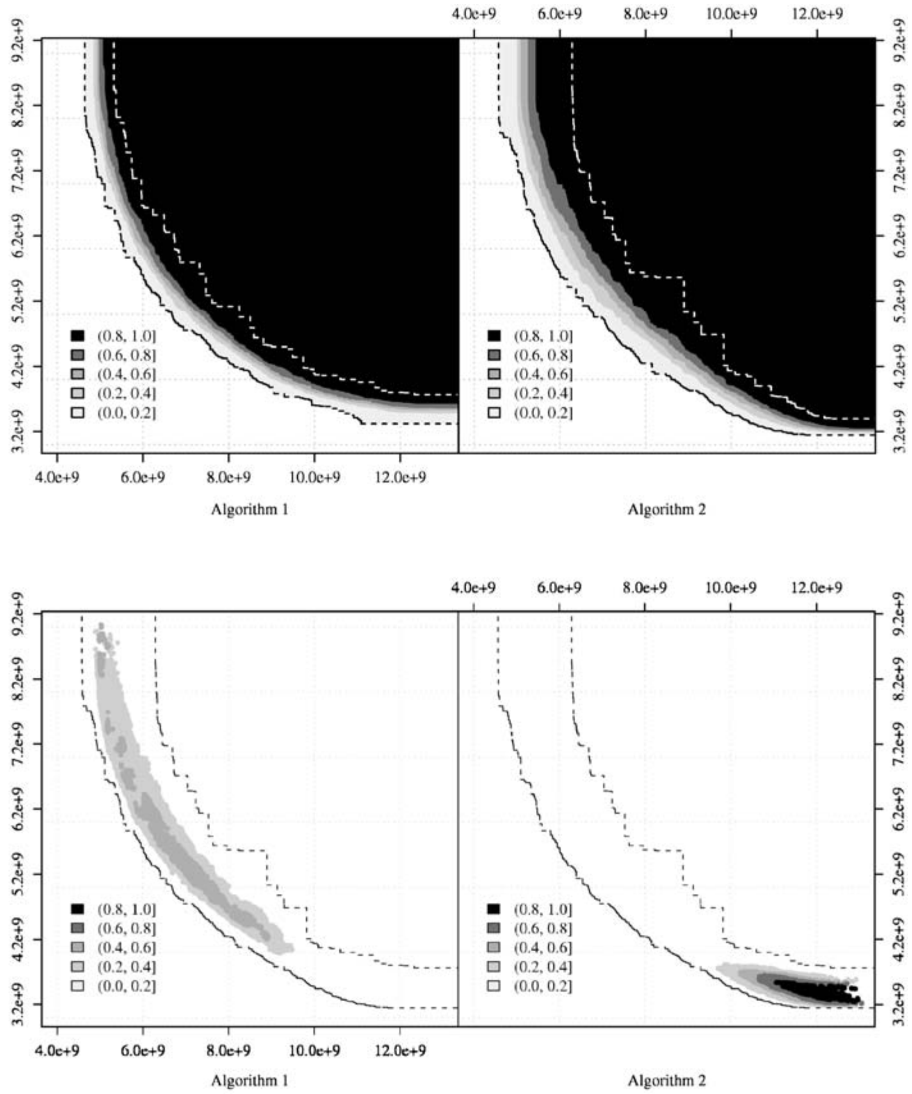

Figure 1. Visualization of the EAFs associated to the outcomes of two algorithms (*top*) and the corresponding differences between the attainment functions (*bottom left*: positive differences for the EAF of Algorithm 1 over the EAF of Algorithm 2; *bottom right*: positive differences of EAF of Algorithm 2 over the EAF of Algorithm 1).

gorithms that would be ignored by most of the current unary and binary measures. The code for computing the EAFs was kindly provided by Carlos Fonseca;⋆ the computation of the EAFs took, on average, almost one minute for

⋆ A new version of the code is now available at http://www.tik.ee.ethz.ch/pisa/.

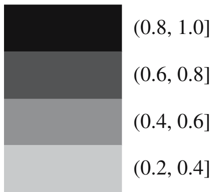

Figure 2. Grey-scale encoding of the differences between attainment functions; differences

each comparison described in the experimental part of this article, within the same computational environment. For the remaining plots of this article, we only use the plots of the differences between the EAFs of pairs of algorithms as indicated in the lower row of Figure 1 and we use the grey-scale encoding given in Figure 2.

6. Experimental Results

The algorithms were coded in `C` and we ran 100 repetitions of each experiment on a Intel Xeon CPU 2.40 GHz with 2 GB of RAM under Debian GNU/Linux. All algorithms were tested on six benchmark instances of size $n = 50$ and their relative performance was then analyzed. The experimental analysis of the algorithms focuses mainly on the hybridization, that is, what strength of the local search gives best performance. To explore the dependence of the relative performance of the algorithms on the available computation time, we used three different time limits as stopping criteria for each experiment. These time limits were defined according to the time required by W-RoTS to perform a particular number of scalarizations using RoTS of length $10n$. We considered $2^7 + 1$, $2^{10} + 1$ and $2^{13} + 1$ scalarizations in order to obtain short, medium and long time limits, which resulted to be of 16.62, 132 and 1055 CPU-seconds, respectively.

6.1. BENCHMARK INSTANCES

The algorithms were tested on 6 symmetric bQAP instances, 3 *unstructured* and 3 *structured* ones. The unstructured instances were taken from an earlier experimental study [22]. All instances were generated using the instance generator of Knowles and Corne [16] with size $n = 50$ and $\xi \in \{0.75, 0.0, -0.75\}$, where ξ is a parameter that influences the correlation between flow matrices. The QAP specific parameter settings for generating the *unstructured* instances were the same as those used for generating instances of class `Taixxa` [28]; parameter settings for the *structured* instances were analogous to those for generating `Taixxb` [28] instances. These two types of instance classes (`Taixxa` and `Taixxb`) are among the most widely studied QAP instances with SLS algorithms for the single-objective QAP, hence we focus on this instance class for the generalization to the bQAP. The instances used in our study are available at 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; the value of ξ and the empirically measured correlation $\widehat{\xi}$ between the objective vectors of randomly generated solutions are given in Table I. The influence of the ξ parameter on the empirical correlation of the objectives for structured instances is rather small, probably because of the high number of zero entries in the flow matrices of structured instances.

6.2. PARAMETER ADJUSTMENT FOR MO-ACO AND SPEA2

For each of the algorithms we initially studied a number of parameter settings, which are detailed below.

MO-ACO. As previously stated, for MO-ACO we used $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) [26] as the underlying ACO algorithm. The total number of ants (m) was set equal to the instance size and we used $c \in \{1, 3, 5\}$ colonies, where each colony had m/c 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 [26]. We also tested both *iteration-best* (*ib*) and *best-so-far* (*bf*) pheromone update strategies.

SPEA2. SPEA2 requires parameter settings for α, the archive size, and μ, the number of new individuals generated in each iteration. For some preliminary experiments, we used the following settings: $\alpha' = 1$ and $\mu' \in \{0.2, 1\}$; $\alpha' \in \{4, 10\}$ and $\mu' \in \{0.2, 1, 5\}$, where we have that $\alpha = \alpha' \cdot n$ and $\mu = \mu' \cdot n$.

From the possible settings for MO-ACO and SPEA2, we tried to identify values resulting in good performance. From some initial experiments it became clear that both MO-ACO and SPEA2 without the use of any local search were clearly outperformed by the hybrid algorithms. In fact, for SPEA2, 100% of the trials without local search were dominated according to the $\lhd$-relation by any of the trials with local search; for MO-ACO we refer to the results presented in [18].

Table I. Given is for each instance the empirical correlation $\widehat{\xi}$ between the components of objective value vectors. ξ is the parameter of the correlation between the flow matrices used in the generation of the instances.

	Unstructured			Structured		
ξ	0.75	0.00	−0.75	0.75	0.00	−0.75
$\widehat{\xi}$	0.90	0.01	−0.91	0.23	0.03	−0.08

When considering only the trials of the hybrid algorithms, we noted that the most significant influence on the final outcomes of the hybrid algorithms was due to the various possible choices of the local search; the choice of the best local search is explained in detail in Section 6.4.

Concerning the algorithm specific parameters, we found that in MO-ACO the influence of the various parameters of the ACO algorithm is not strong if we keep the local search fixed: The outcomes corresponding to different parameter choices for a fixed local search resulted mainly in incomparable sets of solutions; also, analysis of the attainment functions did not lead to strong tendencies for specific configurations. Hence, for the following experiments we chose parameter settings that resulted in good performance overall: $c = 3$ and *ib* were used for unstructured instances, and $c = 5$ and *bf* for structured ones. For SPEA2 the parameters best overall values were $\alpha = 200$ and $\mu = 50$, with the other settings giving slightly worse performance.

6.3. W-RoTS AND W-ll

As a first step in the further analysis, we tested the influence of instance structure and correlation between the flow matrices on the relative performance of W-ll and W-RoTS. As said in Section 5, we do pairwise comparisons of the various algorithm settings for each instance. This results in a total of 4 algorithms tested (W-ll and W-RoTS with settings of $\ell \in \{1, 5, 10\}$) giving 12 pairwise comparisons among algorithms; for each instance and pair of algorithms then $100 \cdot 100$ pairwise comparisons of the outcomes of the trials (100 trials per instance and algorithm combination) with respect to the $\triangleleft$-relation are to be done. The results of this comparison are given in Table II. Here, each entry gives the frequency by which the algorithm indicated on the row is better than the one in the column. In a nutshell, the comparisons based on the $\triangleleft$-relation show that for unstructured instances W-RoTS is clearly superior to W-ll: for these instances, W-RoTS is often better with a frequency of 100%. In fact, for the positively correlated instance, Table II suggests that the longer the length of a tabu trial, the better the performance. Limited experiments with $\ell = 100$ have even shown slightly superior performance of this setting over $\ell = 10$; however, this is not anymore true for other values of the correlation. In the case of unstructured instances with zero and negative correlations, this trend is less obvious. However, an examination of the differences based on EAFs indicates that W-RoTS with a setting of $\ell = 10$ appears still to be superior over $\ell = \{1, 5\}$ (see Figure 3).

For the structured instances, the results based on the pairwise comparisons are not conclusive, since in most cases the outcomes were incomparable (see Table II part on W-RoTS, Structured). However, from an inspection of the EAFs, a rather clear picture arises. W-ll and W-RoTS with $\ell = 1$ perform significantly better across the whole objective space than W-RoTS with $\ell = \{5, 10\}$. W-ll and W-RoTS with $\ell = 1$ are mainly incomparable (see Figure 4).

Table II. Pairwise comparisons of the local search influence on the final performance for three SLS methods, W-RoTS and W-II, SPEA2, and MO-ACO. The results are organized according to correlation (ξ), stopping time criterion (Time), and local search intensity (II and RoTS of length n, $5{\cdot}n$, $10{\cdot}n$). Each entry gives the percentage of the pairwise comparisons for which the outcomes of the algorithm in the row is better than the algorithm in the column

Correlation	0.75												0.0												−0.75											
Time	16.62				132				1055				16.62				132				1055				16.62				132				1055			
Iterations	II	n	$5n$	$10n$	II	n	$5n$	$10n$	II	n	$5n$	$10n$	II	n	$5n$	$10n$	II	n	$5n$	$10n$	II	n	$5n$	$10n$	II	n	$5n$	$10n$	II	n	$5n$	$10n$	II	n	$5n$	$10n$
WROTS																																				
Unstructured																																				
II	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0
n	**57**	–	0	0	**54**	–	0	0	**54**	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0
$5n$	**100**	**99**	–	**1**	**100**	**100**	–	0	**100**	**100**	–	0	**97**	**78**	–	0	**99**	**96**	–	0	**100**	**99**	–	0	0	0	–	0	**67**	**4**	–	0	**95**	**59**	–	0
$10n$	**100**	**100**	**15**	–	**100**	**100**	**20**	–	**100**	**100**	**20**	–	**98**	**85**	0	–	**100**	**99**	0	–	**100**	**100**	0	–	0	0	0	–	**44**	**2**	0	–	**98**	**79**	0	–
Structured																																				
II	–	0	**2**	**6**	–	0	0	**2**	–	0	0	0	–	0	0	**3**	–	0	0	0	–	0	0	0	–	0	0	**1**	–	0	0	0	–	0	0	0
n	0	–	**1**	**5**	0	–	0	**2**	0	–	0	0	0	–	0	**3**	0	–	0	0	0	–	0	0	0	–	0	**1**	0	–	0	0	0	–	0	0
$5n$	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0
$10n$	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–
SPEA2																																				
Unstructured																																				
II	–	0	**25**	**100**	–	0	0	0	–	0	0	0	–	0	0	**5**	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0
n	**82**	–	**99**	**100**	**98**	–	**1**	**2**	**99**	–	0	0	**2**	–	**2**	**47**	**17**	–	0	0	**17**	–	0	0	0	–	0	**1**	0	–	0	0	0	–	0	0
$5n$	**5**	0	–	**100**	**100**	**47**	–	**18**	**100**	**69**	–	**2**	0	0	–	**35**	**16**	0	–	0	**54**	**2**	–	0	0	0	–	**2**	0	0	–	0	0	0	–	0
$10n$	0	0	0	–	**99**	**34**	**4**	–	**100**	**81**	**18**	–	0	0	0	–	**4**	0	0	–	**54**	**2**	0	–	0	0	0	–	0	0	0	–	0	0	0	–
Structured																																				
II	–	0	**99**	**100**	–	0	0	**1**	–	0	0	0	–	0	**99**	**100**	–	0	0	0	–	0	0	0	–	0	**97**	**100**	–	0	0	**4**	–	0	0	0
n	0	–	**98**	**100**	0	–	0	**1**	0	–	0	0	0	–	**99**	**100**	0	–	0	0	0	–	0	0	0	–	**94**	**99**	0	–	0	**2**	0	–	0	0
$5n$	0	0	–	**93**	0	0	–	0	0	0	–	0	0	0	–	**84**	0	0	–	0	0	0	0	0	0	0	–	**69**	0	0	–	0	0	0	–	0
$10n$	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	–	–	0	0	0	–	0	0	0	–	0	0	0	–
MOACO																																				
Unstructured																																				
II	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0
n	**64**	–	0	0	**70**	–	0	0	**82**	–	**27**	**10**	**1**	–	0	0	**1**	–	0	0	**1**	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0
$5n$	**100**	**99**	–	**1**	**97**	**95**	–	0	**74**	0	–	0	**99**	**82**	–	0	**98**	**87**	–	0	**99**	**89**	–	0	0	0	–	0	**59**	**1**	–	0	**93**	**44**	–	0
$10n$	**100**	**100**	**16**	–	**97**	**96**	**18**	–	**80**	**3**	**13**	0	**99**	**90**	0	–	**99**	**95**	0	–	**99**	**93**	0	–	0	0	0	–	**36**	**1**	0	–	**97**	**93**	0	–
Structured																																				
II	–	0	**1**	**12**	–	0	**6**	**57**	–	0	0	0	–	0	0	**3**	–	0	**1**	**21**	–	0	0	0	–	0	0	0	–	0	0	**9**	–	0	0	0
n	0	–	**1**	**10**	0	–	**5**	**52**	0	–	0	0	0	–	0	**3**	0	–	0	**19**	0	–	0	0	0	–	0	0	0	–	0	**6**	0	–	0	0
$5n$	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0
$10n$	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–	0	0	0	–

Main result 1. The main result of this comparison is that the structure of bQAP instances has a very strong influence on the relative performance of the local search algorithms. In fact, for unstructured instances long runs of RoTS inside W-RoTS result in best performance; while for structured instances, W-II and very short RoTS runs in W-RoTS are preferable.

6.4. LOCAL SEARCH FOR MO-ACO AND SPEA2

Next, we investigated which local search is best hybridized with MO-ACO and SPEA2. As indicated by the results for W-RoTS and W-II, the structure of the instances had a very significant influence on the final performance of the hybrids.

SPEA2. From the tables and the inspection of EAFs plots, it can be deduced that for the unstructured instance with $\xi = 0.75$, the choice of an appropriate setting of ℓ depends on the time limit: While for short computation times (of 16.62 seconds) a setting of $\ell = 1$ appears to be preferable over longer runs of RoTS, for longer computation times (of 1055 seconds) it becomes preferable to use larger values for ℓ. This tendency can be observed by means of EAFs on the unstructured instances of $\xi = 0.75$ and $\xi = 0.0$ (see Figure 5 for an illustration of this effect). Furthermore, it should be noted that for negative correlation, SPEA2 with long tabu searches ($\ell = \{5, 10\}$) was performing better than with short tabu searches ($\ell = 1$) at the tails of the outcomes in the objective space, while the outcomes obtained by short tabu searches were slightly better in the center, as

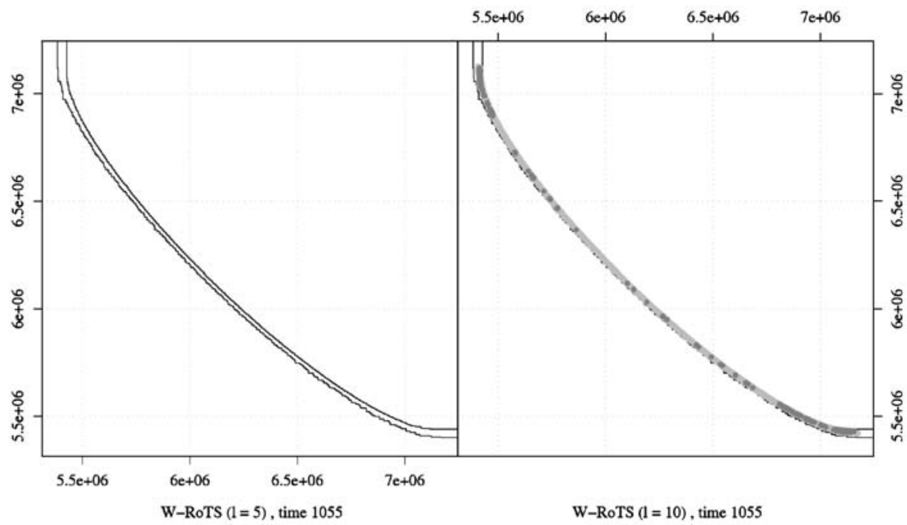

Figure 3. Location of differences with respect to the EAFs for the comparison between W-RoTS with $\ell = 5$ and W-RoTS with $\ell = 10$ (*left side*: positive differences of W-RoTS$_{5n}$ over W-RoTS$_{10n}$; *right side*: positive differences of W-RoTS$_{10n}$ over W-RoTS$_{5n}$) for a computation time of 1055 seconds for an unstructured instance with negative correlation.

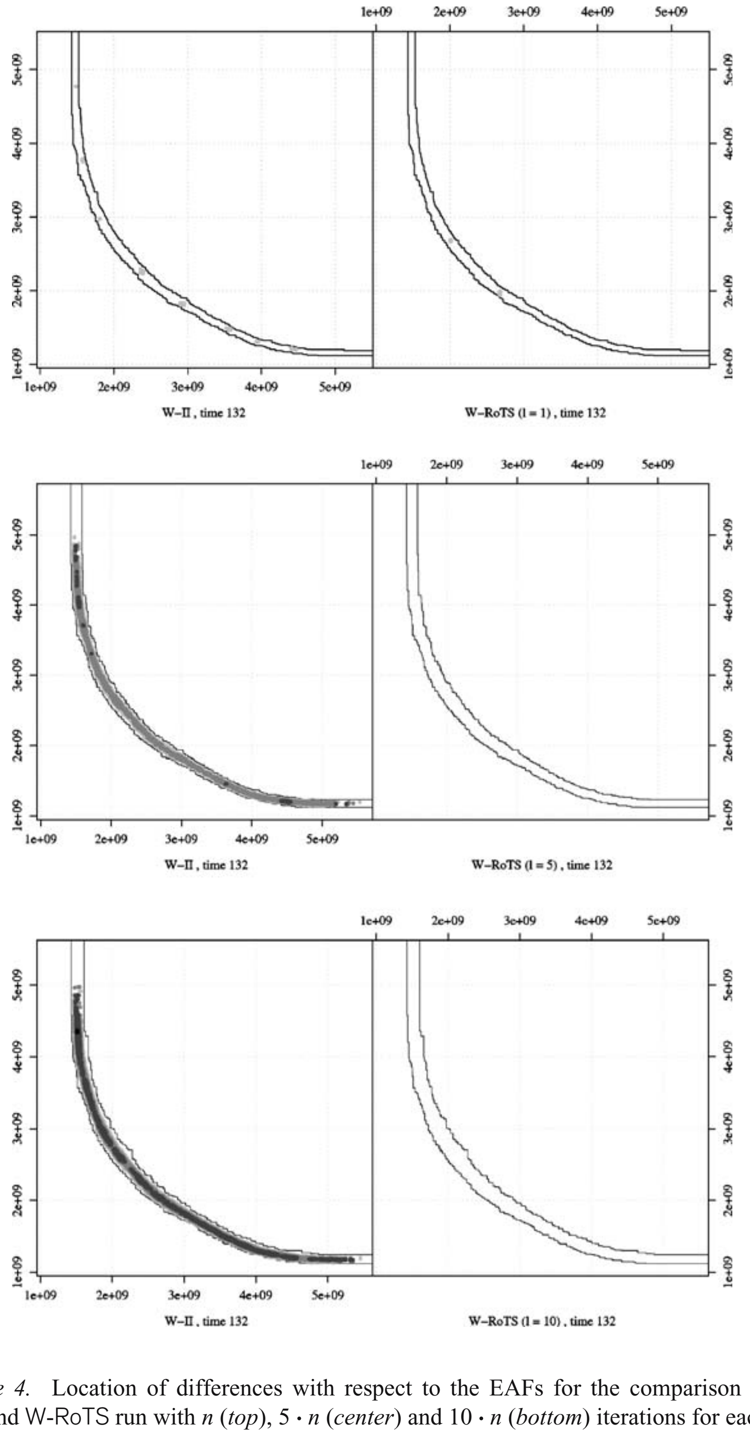

Figure 4. Location of differences with respect to the EAFs for the comparison between W-II and W-RoTS run with n (*top*), $5 \cdot n$ (*center*) and $10 \cdot n$ (*bottom*) iterations for each RoTS trial using a computation time limit of 132 seconds on a structured instance with negative correlation. The name at the bottom of each plot indicates for which of the two algorithms the differences shown are positive.

shown in the bottom plot of Figure 5. For the structured instances using short computation times, SPEA2_{5n} and SPEA2_{10n} are clearly outperformed by SPEA2_n and SPEA2_{II} (see Table II). For larger computation times, the results based on the $\lhd$-relation become inconclusive and therefore, Table II does not give clear results. However, an inspection of the EAFs shows that SPEA2_n and SPEA2_{II} remain highly preferable over the other two variants, while they are mainly incomparable mutually.

MO-ACO. For MO-ACO a very clear picture arises concerning the influence of the local search. For unstructured instances, the longest tabu search length of $10 \cdot n$ resulted in best overall performance. For the positively correlated instance, this is clear from the results given in Table II (entry MO-ACO); however, for the instance with $\xi = 0.75$ and large computation time (1055 seconds), we should notice that MO-ACO_n also obtains very good results. For the remaining unstructured instances with zero or negative correlation, the EAFs give a very strong indication that the larger number of tabu search iterations is clearly preferable: Although this conclusion cannot be obtained directly from Table II, the topmost plot in Figure 6 indicates that MO-ACO_{10n} shows much better performance than even MO-ACO_{5n}. As for structured instances, the results of MO-ACO in Table II are not clear; particularly for large computation times the outcomes of all variants of MO-ACO were incomparable with respect to the $\lhd$-relation, and thus, all entries are zero. Nevertheless, an inspection of the EAFs for the structured instances indicates a strong advantage of MO-ACO_n and $\text{MO-ACO}_{\text{II}}$ over MO-ACO_{5n} and MO-ACO_{10n}. In Figure 6, this is exemplified by the middle plot showing variant $\text{MO-ACO}_{\text{II}}$ *versus* MO-ACO_{5n}. In contrast, MO-ACO_n and $\text{MO-ACO}_{\text{II}}$ are mainly incomparable, showing little differences in various regions of the objective space, as illustrated in the plot at the bottom of Figure 6.

Main result 2. The main result of this section is that the best local search in a hybrid algorithm, be it SPEA2 or MO-ACO, depends strongly on the structure of the *b*QAP instance tackled and to some extent on the correlation ξ and the computation time limit chosen. The general tendency is that differences in the performance of the four local search variants are more pronounced for MO-ACO than for SPEA2, so that, especially on unstructured instances, the MO-ACO_{10n} variant is often better than others, while for SPEA2 the best length of the tabu search depends on the computation time.

6.5. COMPARISON OF SLS METHODS

As a final step, we compared the SLS methods MO-ACO, SPEA2, and W-II or W-RoTS, respectively, using for each instance class good parameter settings identified in the previous analysis.

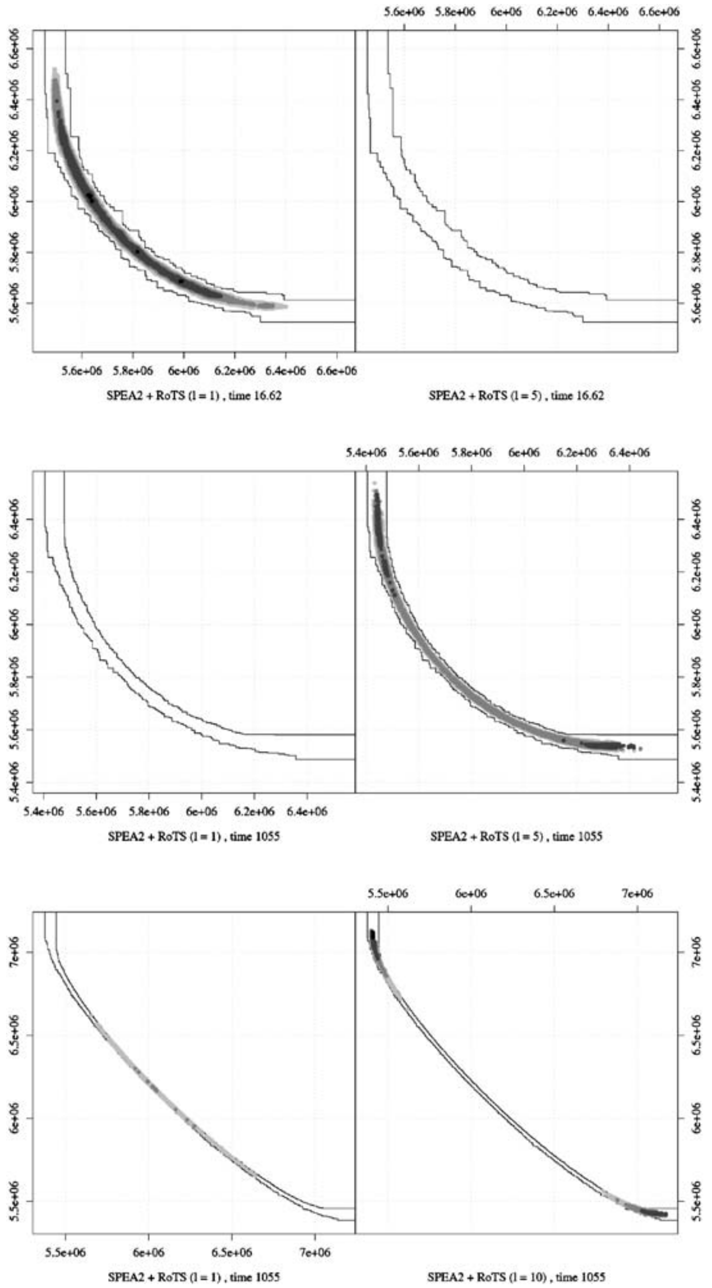

Figure 5. Location of differences with respect to the EAFs for the comparison between SPEA2 using RoTS with n and $5 \cdot n$ iterations for 16.62 (*top*) and 1055 seconds (*center*) for an unstructured instances with $\xi = 0.0$. On the *bottom* is given the comparison between SPEA2 using RoTS with n and with $10 \cdot n$ iterations for 1055 seconds for an unstructured instance with $\xi = -0.75$. The name at the bottom of each plot indicates for which of the two algorithms the differences shown are positive.

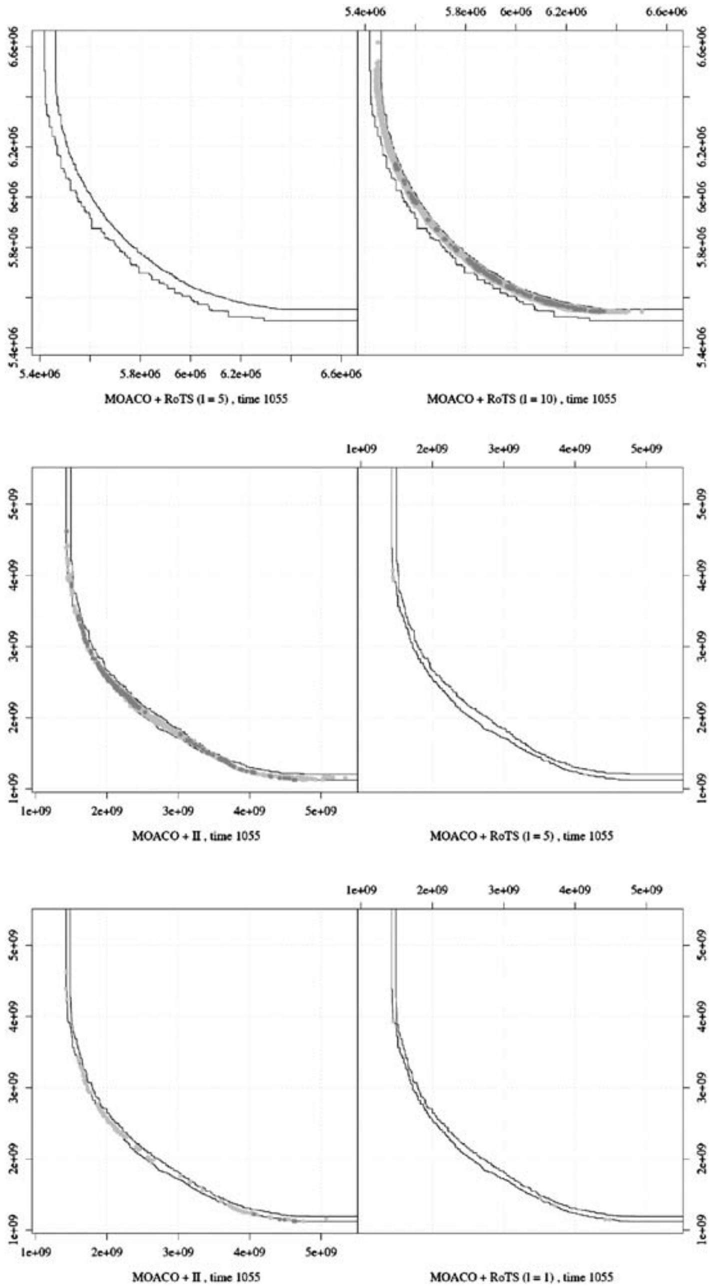

Figure 6. Location of differences in terms of EAFs in the unstructured instance with null correlation between MO-ACO$_{5n}$ and MO-ACO$_{10n}$ (*top*); and in the structured instance with negative correlation, between MO-ACO$_{\mathrm{II}}$ and MO-ACO$_{5n}$ (*middle*), and between MO-ACO$_{\mathrm{II}}$ and MO-ACO$_{n}$ (*bottom*). All the plots refer to experiments with a time limit of 1055 seconds. The name at the bottom of each plot indicates for which of the two algorithms the differences shown are positive.

When comparing SPEA2 to MO-ACO, their relative performance depends strongly on whether the type of instance is structured or unstructured. For structured instances, MO-ACO and SPEA2 show typically incomparable performance from the point of view of the $\triangleleft$-relation as shown in Table III. However, SPEA2 appears to be slightly preferable over MO-ACO for shorter CPU-time, since the differences in terms of EAFs seem to favor SPEA2 (see Figure 7); for longer computation times, the differences between SPEA2 and MO-ACO become less clear. The picture is different on unstructured instances. Although from Table III we see that the results of SPEA2 and MO-ACO are often incomparable with respect to the $\triangleleft$-relation, analysis of the EAFs shows clear advantages for SPEA2. In fact, for $\xi = 0.75$ and $\xi = 0.0$, the EAFs of MO-ACO are in no point larger than the EAFs of SPEA2 by a value of 0.2, while the opposite is true in many points (see Figure 7).

When comparing SPEA2 and MO-ACO to the best versions of W-II and W-RoTS, typically we find better behavior of the hybrid SLS methods. This is particularly true for SPEA2 and MO-ACO on the structured instances; MO-ACO on the unstructured instances appears to perform only slightly better than the constituent local search algorithms. However, the advantage of MO-ACO increases with longer computation time, a fact which suggests that MO-ACO requires long computation times for identifying very high quality approximations to the Pareto global optimum set.

Main result 3. From the various algorithms tested, the overall best performing one appears to be SPEA2. This is clear for the unstructured instances, and to a lesser extent for structured instances. MO-ACO appears to be competitive to SPEA2 in structured instances for long computation times. Additionally, we can conclude that especially for medium and large computation times, the population-based hybrid MO-ACO and SPEA2 algorithms typically show much better behavior than W-II or W-RoTS, which use repeated trials of the same underlying local search algorithms for different scalarizations starting from random initial solutions. Hence, the complexity added to W-RoTS or W-II by introducing an appropriate generation of initial solutions to the local search appears to pay off, at the cost of some fine-tuning of parameters.

7. Conclusions

In this article, we studied the performance of SLS algorithms for the bQAP. We focused on hybrid algorithms between local searches (including iterative improvement algorithms and tabu searches) and the population-based algorithms MO-ACO and SPEA2 (an ant colony optimization algorithm and an evolutionary algorithm). The local search algorithms studied in the hybrids allow to trade, rather arbitrarily, effectiveness of the local search for efficiency. Computational results on benchmark bQAP instances, show that the best tradeoff depends

Table III. Pairwise comparisons between W-RoTS, W-Il, SPEA2 and MO-ACO on their best configurations for each instance. The results are organized according to correlation (ε), stopping time criterion (Time) and instance type. Each entry gives the percentage of the pairwise comparisons for which the outcomes of the algorithm in the row is better than the algorithm in the column.

Correlation	0.75						0.0						−0.75					
Time	16.62		132		1055		16.62		132		1055		16.62		132		1055	
Unstructured																		
	SPEA2	WROTS	SPEA2	WROTS	SPEA2	WROTS	SPEA2	WROTS	SPEA2	WROTS	SPEA2	WROTS	SPEA2	WROTS	SPEA2	WROTS	SPEA2	WROTS
SPEA2	–	**6**	–	**54**	–	**53**	–	0	–	0	–	**6**	–	0	–	0	–	0
WROTS	**14**	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–
	MOACO	WROTS	MOACO	WROTS	MOACO	WROTS	MOACO	WROTS	MOACO	WROTS	MOACO	WROTS	MOACO	WROTS	MOACO	WROTS	MOACO	WROTS
MOACO	–	**4**	–	**5**	–	**3**	–	0	–	0	–	0	–	0	–	0	–	0
WROTS	**6**	–	**2**	–	**1**	–	0	–	0	–	0	–	0	–	0	–	0	–
	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO
SPEA2	–	**6**	–	**54**	–	**48**	–	0	–	0	–	**4**	–	0	–	0	–	0
MOACO	**13**	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–
Structured																		
	SPEA2	W-Il	SPEA2	W-Il	SPEA2	W-Il	SPEA2	W-Il	SPEA2	W-Il	SPEA2	W-Il	SPEA2	W-Il	SPEA2	W-Il	SPEA2	W-Il
SPEA2	–	**29**	–	**5**	–	0	–	**7**	–	0	–	0	–	0	–	0	–	0
W-Il	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–
	MOACO	W-Il	MOACO	W-Il	MOACO	W-Il	MOACO	W-Il	MOACO	W-Il	MOACO	W-Il	MOACO	W-Il	MOACO	W-Il	MOACO	W-Il
MOACO	–	0	–	**12**	–	0	–	0	–	0	–	0	–	0	–	0	–	0
W-Il	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–
	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO	SPEA2	MOACO
SPEA2	–	**28**	–	0	–	0	–	**6**	–	0	–	0	–	0	–	0	–	0
MOACO	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–

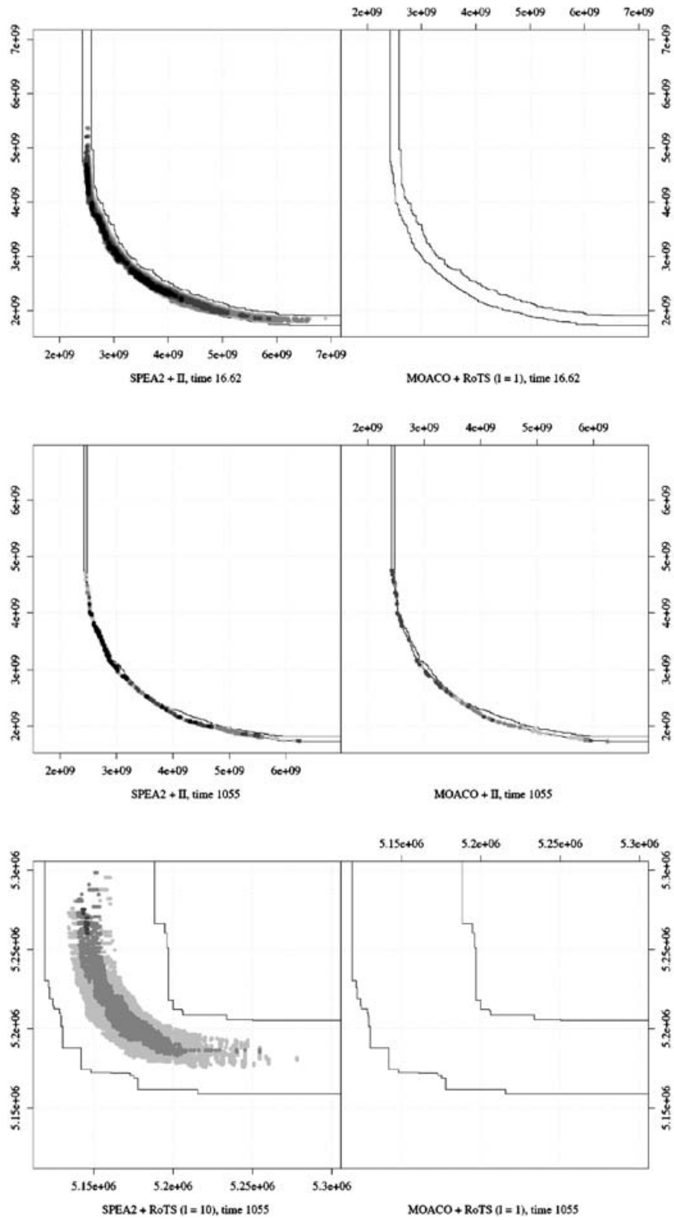

Figure 7. Location of differences in terms of EAFs between SPEA2 and MO-ACO for 16.62 seconds (*top*) and 1055 seconds (*middle*) on a structured instance with $\xi = 0.0$ and for 1055 seconds (*bottom*) on an unstructured instance with $\xi = 0.75$. The name at the bottom of each plot indicates for which of the two algorithms the differences shown are positive.

strongly on the structure of the underlying QAP instances. For instances with unstructured distance and flow matrices, the general tendency is to prefer more effective local searches; while on structured instances similar to real life ones, the efficiency of the local search is more important and the hybrid algorithm relies more heavily on the guiding abilities of the population-based search. However, the best tradeoff for SPEA2 and for MO-ACO apparently lies at different points. Especially for the unstructured instances, MO-ACO profits more from the effectiveness of longer RoTS trials than SPEA2. In general, the tendencies about the configuration of effective hybrid algorithms for the bQAP apparently are strongly related to that observed on the single-objective QAP [6, 26], where the structure of instances has a similar strong influence on the best possible configuration of hybrid algorithms as observed here. In this case, the knowledge obtained on the single-objective QAP would serve as a good guide in designing high-performing hybrid algorithms in the multi-objective case. However, further experimental results on other combinatorial optimization problems are needed to ascertain whether the conclusions obtained for $\mathcal{NP}$-hard single-objective problems can be transfered straightforwardly to the multi-objective counterparts. If this were the case, it would significantly ease the design of algorithms for multi-objective problems.

As stated, the strongest influence on the relative performance of the hybrid algorithms studied here is exerted by the structure of the instances. In earlier research, it was found that the correlation between the flow matrices alone has a very significant influence on the performance of local search algorithms for the bQAP [22]. For the hybrid algorithms applied here, there is still some notable influence of the correlation on the relative performance of the algorithms. For example, from the results in Tables II and III for positively correlated or zero-correlated flow matrices alone, some conclusions can often be drawn about which variant is performing better. This is typically not true for negatively correlated flow matrices, possibly due to the widely spread fronts in this latter case. Nevertheless, the hybrid algorithms are much less affected by the correlation between the flow matrices than the underlying local search algorithms.

Among the two population-based algorithms, SPEA2 hybrids appear to be preferable over MO-ACO hybrids. In fact, our hybrid SPEA2 algorithm at the very moment is a state-of-the-art algorithm for the bQAP. However, it must also be mentioned that evolutionary algorithms for solving multi-objective problems are very widely studied with even specialized conferences like EMO, while effective ACO approaches for these problems are still in their infancy. Hence, further research on ACO may shift that conclusion.

Finally, let us mention that the performance assessment methodology used in this article overcomes the drawbacks of unary measures by comparing algorithms based on the $\triangleleft$-relation; we are able to indicate whether the outcome of one algorithm is *better* than another, a conclusion that cannot be obtained by unary measures or combinations thereof [31]. Moreover, when incomparability is detected between outcomes, we use the attainment functions methodology to pinpoint where differences in the objective space occur and how large these differences are. Further work is required in this area to extend the usefulness of these performance assessment methods for *inferring* conclusions with sound statistical support.

Acknowledgement

We would like to thank Carlos Fonseca for valuable discussions on the attainment functions methodology. Luís 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. Thomas Stützle acknowledges support of the Belgian FNRS, of which he is a research associate.

References

1. Bleuler, S., Laumanns, M., Thiele, L. and Zitzler, E.: PISA – A platform and programming language independent interface for search algorithms in C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb and L. Thiele (eds.), *Evolutionary Multi-Criterion Optimization (EMO 2003)*, Vol. 2632 of *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg New York, 2003, pp. 494–508.
2. Çela, E.: *The Quadratic Assignment Problem: Theory and Algorithms*, Kluwer Academic, Dordrecht, The Netherlands, 1998.
3. Dorigo, M. and Stützle, T.: *Ant Colony Optimization*, MIT, Cambridge, Massachusetts, 2004.
4. Fleurent, C. and Ferland, J. A.: Genetic hybrids for the quadratic assignment problem, in P. M. Pardalos and H. Wolkowicz (eds.), *Quadratic Assignment and Related Problems*, Vol. 16 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, American Mathematical Society, Providence, Rhode Island, 1994, pp. 173–187.
5. Galinier, P. and Hao, J. K.: Hybrid evolutionary algorithms for graph coloring. *J. Comb. Optim.* **3**(4) (1999), 379–397.
6. Gambardella, L. M., Taillard, E. D. and Dorigo, M.: Ant colonies for the quadratic assignment problem. *J. Oper. Res. Soc.* **50**(2) (1999), 167–176.
7. Grunert da Fonseca, V., Fonseca, C. M. and Hall, A.: Inferential performance assessment of stochastic optimisers and the attainment function, in E. Zitzler, K. Deb, L. Thiele, C. C. Coello and D. Corne (eds.), *Evolutionary Multi-criterion Optimization (EMO 2001)*, Vol. 1993 of *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg New York, 2001, pp. 213–225.
8. Hamacher, H., Nickel, S. and Tenfelde-Podehl, D.: Facilities layout for social institutions, in *Operation Research Proceedings 2001, Selected Papers of the International Conference on Operations Research (OR2001)*, Springer, Berlin Heidelberg New York, 2001, pp. 229–236.
9. Hansen, M. P. and 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.

10. Hoos, H. and Stützle, T.: *Stochastic Local Search – Foundations and Applications*, Morgan Kaufmann, San Francisco, California, 2004.
11. Iredi, S., Merkle, D. and Middendorf, M.: Bi-Criterion optimization with multi colony ant algorithms, in E. Zitzler, K. Deb, L. Thiele, C. C. Coello and D. Corne (eds.), *First International Conference on Evolutionary Multi-Criterion Optimization, (EMO'01)*, Vol. 1993 of *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg New York, 2001, pp. 359–372.
12. Ishibuchi, H., Yoshida, T. and Murata, T.: Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Trans. Evol. Comput.* **7**(2) (2003), 204–223.
13. Jaszkiewicz, A.: Genetic local search for multiple objective combinatorial optimization. *Eur. J. Oper. Res.* **137**(1) (2002), 50–71.
14. Knowles, J. and Corne, D.: The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation, in *Proceedings of 1999 Congress on Evolutionary Computation (CEC'99)*, Vol. 1. 1999, pp. 98–105.
15. Knowles, J. and Corne, D.: Towards landscape analyses to inform the design of a hybrid local search for the multiobjective quadratic assignment problem, in A. Abraham, J. R. del Solar and M. Koppen (eds.), *Soft Computing Systems: Design, Management and Applications*, IOS, 2002, pp. 271–279.
16. Knowles, J. and Corne, D.: Instance generators and test suites for the multiobjective quadratic assignment problem, in C. M. Fonseca, P. Fleming, E. Zitzler, K. Deb and L. Thiele (eds.), *Evolutionary Multi-criterion Optimization (EMO 2003)*, Vol. 2632 of *Lecture Notes in Computer Sience*, Springer, Berlin Heidelberg New York, 2003, pp. 295–310.
17. López-Ibáñez, M.: Multi-objective ant colony optimization. Diploma thesis, Intellectics Group, Computer Science Department, Technische Universität Darmstadt, Germany, 2004.
18. López-Ibáñez, M., Paquete, L. and Stützle, T.: On the design of ACO for the biobjective quadratic assignment problem, in M. Dorigo, L. Gambardella, F. Mondada, T. Stützle, M. Birratari and C. Blum (eds.), *ANTS'2004, Fourth International Workshop on Ant Algorithms and Swarm Intelligence*, Vol. 3172 of *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg New York, 2004, pp. 214–225.
19. Merz, P.: Memetic algorithms for combinatorial optimization problems: fitness landscapes and effective search strategies. PhD thesis, Department of Electrical Engineering and Computer Science, University of Siegen, Germany, 2000.
20. Merz, P. and Freisleben, B.: Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Trans. Evol Comput.* **4**(4) (2000), 337–352.
21. Paquete, L., Chiarandini, M. and Stützle, T.: Pareto local optimum sets in the biobjective traveling salesman problem: an experimental study, in X. Gandibleux, M. Sevaux, K. Sörensen and V. T'kindt (eds.), *Metaheuristics for Multiobjective Optimisation*, Vol. 535 of *Lecture Notes in Economics and Mathematical Systems*, Springer, Berlin Heidelberg New York, 2004, pp. 177–200.
22. Paquete, L. and Stützle, T.: A study of stochastic local search algorithms for the biobjective QAP with correlated flow matrices. *Eur. J. Oper. Res*. **169**(3) (2006), 943–959.
23. Sahni, S. and Gonzalez, T.: P-complete approximation problems. *J. ACM* **23** (1976), 555–565.
24. Steuer, R. E.: *Multiple Criteria Optimization: Theory, Computation and Application*, Wiley Series in Probability and Mathematical Statistics, Wiley, New York, 1986.
25. Stützle, T. and Dorigo, M.: ACO algorithms for the quadratic assignment problem, in D. Corne, M. Dorigo and F. Glover (eds.), *New Ideas in Optimization*, McGraw Hill, London, UK, 1999, pp. 33–50.

26. Stützle, T. and Hoos, H. H.: $\mathcal{MAX}$-$\mathcal{MIN}$ ant system. *Future Gener. Comput. Syst.* **16**(8) (2000), 889–914.
27. Taillard, É. D.: Robust taboo search for the quadratic assignment problem. *Parallel Comput.* **17** (1991), 443–455.
28. Taillard, É. D.: Comparison of iterative searches for the quadratic assignment problem. *Location Sci.* **3** (1995), 87–105.
29. Zitzler, E., Laumanns, M. and Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, in K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papaliliou and T. Fogarty (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, pp. 95–100.
30. Zitzler, E. and Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **4**(3) (1999), 257–271.
31. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M. and Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.* **7**(2) (2003), 117–132.