Solving Multi-Criteria Optimization Problems with Population-Based ACO

Michael Guntsch[1] and Martin Middendorf[2]

[1]Institute for Applied Computer Science and Formal Description Methods
University of Karlsruhe, D-76128 Karlsruhe, Germany
guntsch@aifb.uni-karlsruhe.de
[2]Department of Computer Science
University of Leipzig, Augustusplatz 10-11, D-04109 Leipzig, Germany
middendorf@informatik.uni-leipzig.de

Abstract. In this paper a Population-based Ant Colony Optimization approach is proposed to solve multi-criteria optimization problems where the population of solutions is chosen from the set of all non-dominated solutions found so far. We investigate different maximum sizes for this population. The algorithm employs one pheromone matrix for each type of optimization criterion. The matrices are derived from the chosen population of solutions, and can cope with an arbitrary number of criteria. As a test problem, Single Machine Total Tardiness with changeover costs is used.

1 Introduction

The Ant Colony Optimization (ACO) metaheuristic (see Dorigo and Di Caro [8]) has recently been applied to solve multi-criterion optimization problems (see [3, 1] for an overview over metaheuristics for multi-criteria optimization). In most of the earlier works it is assumed that the optimization criteria can be weighted by importance. Mariano and Morales [16] proposed a multi colony ACO approach where for each objective there exists one colony of ants. They studied a problem where every objective is influenced only by parts of a solution, so that an ant from colony i receives a (partial) solution from ant of colony $i-1$ and then tries to improve or extend this solution with respect to criterion i. A final solution that has passed through all colonies is allowed to update the pheromone information when is is part of the non-dominated front. Gambardella et al. [11] developed an ant algorithm for a bi-criterion vehicle routing problem where they also used one ant colony for each criterion. Criterion 1 — the number of vehicles — is considered to be more important than criterion 2 — the total travel time of the tours. The two colonies share a common global best solution which is used for pheromone update in both colonies. Colony 1 tries to find a solution with one vehicle less than the global best solution, while colony 2 tries to improve the global best solution with respect to criterion 2 under the restriction that the solution is not worse than the global best solution with respect to the first criterion. Whenever colony 1 finds a new global best solution both colonies start anew (with the new global best solution).

Gagné et al. [10] tested a multi-criterion approach for solving a single machine total tardiness problem with changeover costs and two additional criteria. In their approach the changeover costs were considered to be most important. Heuristic values for the decisions of the ants were used that take all criteria into account. However, the amount of pheromone that an ant adds to the pheromone matrix depends solely on the changeover costs of the solution. A similar approach was used in [12] for a four criterion industrial scheduling problem.

In [7, 6] Doerner et al. proposed to solve a transportation problem were the aim was to minimize the total costs by searching for solutions that minimize two different criteria. The general approach was two use two colonies of ants where each colony concentrates on a different criterion by using different heuristics. In [7] one criterion was considered the main criterion. Every k iterations, the master population which minimizes the main criterion updates its pheromone information according to the good solutions found in the slave population, which minimizes the minor criterion. However, no information flow occurs from the slave to the master colony. In [6], both criteria were considered to be of equal importance. The size of both populations was adapted so that the colony that found the better solution with respect to costs became larger. Information exchange between the colonies is done by so called spy ants that base their decisions on the pheromone matrices in both colonies.

The only ACO approaches so far that aim to cover the pareto-front of a multiobjective optimization problem have been proposed by Doerner at al. [4, 5] and Iredi at al. [15].

Doerner at al. [4, 5] studied a portfolio optimization problem with more than two criteria. For each criterion, a separate pheromone matrix is used. Instead of a population of ants for each criterion each ant assigns weights to the pheromone information for all criteria according to a random weight vector when constructing a solution. Pheromone update is done by ants that found the best or the second best solution with respect to one criterion. A problem with this approach is that solutions in the non-dominated front that are not among the best with respect to a single criterion do not update the pheromone information.

Iredi et al. [15] studied an approach to solve bi-criterion optimization problems with a multiple colony ant algorithm were the colonies specialize to find good solutions in different parts of the front of non-dominated solutions. Cooperation between the colonies is established by allowing only ants with solutions in the global front of non-dominated solutions to update the pheromone information (i.e. in contrast to [4, 5], all solutions in the non-dominated front influence the future search process). Two methods for pheromone update in the colonies were proposed. In the update by origin method an ant updates only in its own colony. For the other method the sequence of solutions along the non-dominated front is split into p parts of equal size. Ants that have found solutions in the ith part update in colony i, $i \in [1, p]$. This update method is called update by region in the non-dominated front. It was shown that cooperation between the colonies allows to find good solutions along the whole Pareto front. Heterogeneous colonies were used where the ants have different preferences between the criteria when constructing a solution. For the SMTTP with changeover costs test problem, two pheromone matrices were used: $M = (\tau_{ij})$ for the total tardiness criterion, where τ_{ij} is the desirability that job j is on place i of the schedule, and $M' = (\tau'_{ij})$ for the changeover cost criterion, where τ'_{ij} is the desirability to schedule job j immediately after job i.

In this paper, a Population-based Ant Colony Optimization (PACO) approach to solve multi-criteria optimization problems is proposed where the population of solutions is chosen from the set of all non-dominated solutions found so far (see [14] for the concept of Population-based ACO). The aim is to find a set of different solutions which covers the Pareto-optimal front. One advantage of the proposed algorithm is that it can be applied to problems with more than two criteria and is not biased to solutions that are the best for one criterion.

The PACO approach for single-criteria problems is described in Section 2. In Section 3, we introduce the new methods for applying PACO to multi-criterial problems. The test instances and parameters are described in Section 4. The Results are discussed in Section 5 and conclusions are given in Section 6.

2 Monocriterial Optimization and Population-Based ACO

In this section, we describe the general principle employed by ACO to build solutions for single-criteria optimization problems and the modifications to the standard approach by PACO (see [14, 13] for more details). As example problems, we use two Single-Machine Scheduling problems that are also used for later for evaluating the proposed methods. We also describe the Summation Evaluation method for pheromone evaluation as introduced by Merkle and Middendorf in [17], which is included in our algorithm.

2.1 Solution Construction

When constructing solutions to an optimization problem with ACO, (artificial) ants proceed in an iterative fashion, making a number of decisions until the global solution is completed ([9]). Ants that found a good solution mark their paths through the decision space by putting some amount of pheromone along the path. The following ants of the next generation are attracted by the pheromone so that they will search in the solution space near good solutions. For a single machine scheduling problem, an ant will choose an initial job and proceed by deciding which job to place next until all jobs have been scheduled. The decisions an ant makes are probabilistic in nature and influenced by two factors: the pheromone information, which is gained from the choices made by previous good ants, and heuristic information, which indicates the immediate benefit of making the corresponding choice. Depending on the type of problem being processed, the pheromone and heuristic information have different interpretations. Consider the Single Machine Total Tardiness Problem (SMTTP) which is defined as follows:

- Given: n jobs, where job $j \in [1, n]$ has a processing time p_j and a due date d_j.
- Find: A non-preemptive one machine schedule that minimizes the value of $\mathcal{T} = \sum_{j=1}^{n} \max\{0, C_j - d_j\}$, where C_j is the completion time of job j.

$\mathcal{T}$ is called the total tardiness of the schedule. For this problem, the pheromone information τ_{ij} and the heuristic information η_{ij} usually give information about the expected benefit of assigning job j to place i in the schedule, with $i, j \in [1, n]$. In this context we speak of the pheromone being stored in a job×place pheromone matrix. The heuristic information is defined via the modified due date rule ([17]):

$$\eta_{ij} = \frac{1}{\max\{T + p_j, d_j\} - T} \tag{1}$$

where T is the total processing time of all jobs already scheduled. The Single Machine problem with changeover costs is defined as follows:

- Given: n jobs, where for every pair of jobs i, j, $i \neq j$ there are changeover costs $c(i, j)$ that have to be paid when j is the direct successor of i in a schedule.
- Find: A non-preemptive one machine schedule that minimizes the sum of the changeover costs $C = \sum_{i=1}^{n-1} c_{\pi(i)\pi(i+1)}$, where the permutation π is the sequence of jobs in the schedule.

C is called the cost of the schedule. For this problem, the actual place in the schedule is no longer important for a given job. Rather, its predecessor determines the cost incurred. Hence, the pheromone information τ_{ij} and heuristic information η_{ij} refer to placing job j after job i in the schedule, again with $i, j \in [1, n]$. For this case we say that the pheromone is located in a job×job pheromone matrix. The heuristic information is defined by

$$\eta_{ij} = \frac{1}{c_{ij} + 1} \tag{2}$$

for non-negative changeover costs. Note that this problem is closely related to the Travelling Salesman Problem. Also, in contrast to the dynamic heuristic information for tardiness, η_{ij} is constant. Although only the predecessor of a job $j \in [1, n]$ is important for determining the resulting cost, it can still make sense to gain information about which job is placed first in the schedule, since this job has no predecessor and thereby no incurred changeover cost. To realize this, a dummy-job 0 is included, with $\forall j \in [1, n] : c_{0j} = 0$. This job is always scheduled first and given a row in the pheromone matrix, so that τ_{0j} will contain the information how beneficial it has previously been to schedule job $j \in [1, n]$ as the first "real" job.

For any given place, the set of jobs that can still be assigned is denoted by S. With probability q_0, where $0 \leq q_0 < 1$ is a parameter of the algorithm, the ant chooses the job $j \in S$ which maximizes $\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}$, where $\alpha > 0$ and

$\beta > 0$ are constants that determine the relative influence of the heuristic and the pheromone values on the decision of the ant. With the probability of $1 - q_0$, an ant chooses according to the selection probability distribution over S defined by ([9]):

$$\forall j \in S: \; p_{ij} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{h \in S} \tau_{ih}^{\alpha} \cdot \eta_{ih}^{\beta}} \tag{3}$$

2.2 Summation Evaluation

Merkle and Middendorf [17] have proposed an alternative method for evaluating the pheromone information stored in the matrix $[\tau_{ij}]$ when dealing with tardiness minimization scheduling problems. Instead of using only the pheromone value τ_{ij}, the sum over all pheromone values up to and including i, that is $\tau'_{ij} = \sum_{l=1}^{i} \tau_{lj}$ is used. A study of combining a weighted version of this summation evaluation and regular evaluation was performed for the problem of Single Machine Total Weighted Tardiness in [18] and shown to be superior to regular evaluation. In this combination, instead of τ_{ij}, the value

$$\tau_{ij}^{*} = c \cdot x_i \cdot \tau_{ij} + (1 - c) \cdot y_i \cdot \sum_{l=1}^{i} \gamma^{i-l} \tau_{lj} \tag{4}$$

is used in Formula 3. The parameters of τ_{ij}^{*} are c, which determines the relative influence of weighted summation evaluation, γ, which indicates the weight of previous pheromone values, and x_i and y_i, which are used for scaling, which is necessary since the value provided by the weighted summation evaluation can be a lot larger than the standard pheromone value. Specifically, the scaling values are $x_i = \sum_{h \in S} \sum_{l=1}^{i} \gamma^{i-l} \tau_{lh}$ and $y_i = \sum_{h \in S} \tau_{ih}$.

2.3 Pheromone Update

After m ants have constructed solutions, the pheromone information is updated. This is the point where PACO differs from the standard ACO heuristic. The standard ACO employs evaporation to reduce all pheromone values by a relative amount ρ, $\tau_{ij} \mapsto (1 - \rho)\tau_{ij}$ and afterwards performs a positive update with the ant(s) that found the best solutions. For each of these solutions all pheromone values τ_{ij} corresponding to the choices ij of the solution an update is done according to:

$$\tau_{ij} \mapsto \tau_{ij} + \Delta \tag{5}$$

PACO employs a population $P = \{\pi_1, \ldots, \pi_k\}$ of k good solutions, from which the pheromone information τ_{ij} is derived as follows. Each element of the pheromone matrix has an initial value τ_{init}. Whenever a solution enters the population P, a positive update is performed as in Formula 5. If a solution is removed from the population, its influence is explicitly removed from the pheromone matrix by performing a *negative* update, i.e. using $-\Delta$ in Formula 5. As a result, if $\pi(i) = j$ signifies that job j was positioned at place i, then a job×place interpretation of the population P would yield the pheromone matrix $[\tau_{ij}]$ with

$$\tau_{ij} = \tau_{init} + \Delta \cdot |\{\pi \in P | \pi(i) = j\}|. \tag{6}$$

We denote the maximum possible value an element of the pheromone matrix can achieve by Equation 6 as $\tau_{max} := \tau_{init} + k \cdot \Delta$. Reciprocally, if τ_{max} is used as a parameter of the algorithm instead of Δ, we can derive $\Delta := (\tau_{max} - \tau_{init})/k$ so that with Equation 6, τ_{max} is indeed the maximum attainable value for any τ_{ij}. Note that the actual value for τ_{init} is arbitrary, as τ_{max} could simply be scaled in accordance to achieve an identical probability distribution in Equation 3. For reasons of clarity, we wish the row/column-sum of initial pheromone values to be 1, which means that $\tau_{init} = 1/(n-1)$ for the job×job pheromone matrix where the diagonal is 0, and $\tau_{init} = 1/n$ for the job×place matrix.

3 Multi-Criteria Optimization

In this section we introduce a PACO approach for finding solutions in multi-criteria environments. We also propose a new method for ants to make decisions based on pheromone and heuristic information originating from different criteria.

3.1 Population of Solutions

Some methods for updating the population of PACO for single-criteria optimization problems have been studied by Guntsch and Middendorf in [14, 13]. In this subsection, we describe a novel way to employ the population for multi-criteria optimization problems.

Let Q denote the set of non-dominated solutions that have been found so far. This set will act as the super-population for PACO, from which the population $P \subseteq Q$ is derived to construct the pheromone matrices for the ants to work with. First, the algorithm chooses one starting solution π from Q at random. Then, the $k-1$ solutions in Q which are closest to π with respect to some distance measure are determined (if $|Q| \geq k-1$). Here distance is defined simply by the sum of absolute differences in solution quality over all criteria. Together, these k solutions form the population $P = \{\pi_1, \pi_2, \ldots, \pi_k\}$, with $\pi_1 = \pi$, from which the two pheromone matrices are determined according to Formula 6. After a solution has been constructed by an ant the set Q is updated. After m ants have constructed a solution a new population P is chosen.

3.2 Average-Rank-Weight Method

For multi-criteria problems the ants make their decisions based on pheromone and heuristic information originating from different criteria. The method proposed here differs from the one employed by Merkle and Middendorf [17], where each ant is assigned a weight $\lambda \in [0, 1]$ which defines the relative influence of the two criteria on the decisions of an ant. Instead, we calculate a probability distribution p_{ij}^{ζ} for each criterion ζ (according to Formula 3 and using summation evaluation as described in subsection 2.2 when ξ is a tardiness criterion) and from these construct the final selection probability distribution

$$p_{ij}^{\Sigma} = \sum_{\zeta} w_{\zeta} \cdot p_{ij}^{\zeta}, \tag{7}$$

with each individual weight w_{ζ} determining the influence of criterion ζ on the decision process, and $\sum_{\zeta} w_{\zeta} = 1$. This method has the advantage of remaining feasible for an arbitrary number of criteria and not requiring any corrective scaling of pheromone or heuristic values.

Population P is used to determine the weights $w_{\zeta} = w_{\zeta}(P)$ for each criterion ζ needed for Formula 7. The general idea is to give a criterion a higher weight the better the solutions in P are with respect to this criterion compared to all solutions in Q. Formally, to compute these weights we assign each solution $\pi \in P$ a reverse rank $r_{\zeta}(\pi) \in [0, |Q| - 1]$ for each criterion ζ. By reverse rank we mean that $r_{\zeta}(\pi) = 0$ is worst and $r_{\zeta}(\pi) = |Q| - 1$ is best. Let $q_{\zeta}(\pi)$ denote the quality of solution π with respect to criterion ζ, where, since we are minimizing, lesser values of $q_{\zeta}(\pi)$ indicate a better solution. Then

$$r_{\zeta}(\pi) = |Q| - |\{\sigma \in Q | q_{\zeta}(\sigma) < q_{\zeta}(\pi)\}| - 1 \tag{8}$$

and using this reverse rank, we define the solution weights via

$$w_{\zeta}(\pi) = \frac{r_{\zeta}(\pi)}{\sum_{\xi} r_{\xi}(\pi)} \tag{9}$$

Finally, from the individual solution weights, we calculate the combined weight for the population $P = \{\pi_1, \ldots, \pi_k\}$ by aggregating the weights of all solutions in P with respect to the criterion ξ

$$w_{\zeta}(P) = \frac{1}{|P|} \sum_{i=1}^{k} w_{\zeta}(\pi_i) \tag{10}$$

4 Test Setup

We now describe the problem, instances, and parameter settings used to evaluate the methods proposed in Section 2. As mentioned previously, we let the algorithm run on a Single Machine Total Tardiness problem with changeover costs. This problem is a combination of the two scheduling problems defined in Subsection 2.1, and thereby a bi-criterial optimization problem, with one matrix $[c_{ij}]$ for changeover costs when switching from job i to job j, and for each job i a processing time and a due date $[p_i, d_i]$. However, it is possible to scale this problem to more criteria by utilizing several matrices for changeover costs, each representing one criterion, as well as having more than one processing time and due date for each job, which again leads to multiple criteria. Hence, for n jobs the quality $q_\zeta(\pi)$ of a solution π with respect to criterion ζ is defined as

$$q_\zeta(\pi) = \begin{cases} \sum_{i=1}^{n-1} c^\zeta_{\pi(i)\pi(i+1)} & \text{if } \zeta \text{ is a changeover criterion,} \\ \sum_{i=1}^{n} \max(C^\zeta_{\pi(i)} - d^\zeta_{\pi(i)}, 0) & \text{if } \zeta \text{ is a tardiness criterion} \end{cases} \quad (11)$$

where $C^\zeta_{\pi(i)}$ ($c^\zeta_{\pi(i)\pi(i+1)}$,$d^\zeta_{\pi(i)}$) is the completion time (respectively, changeover cost, deadline) of job i according to the processing times of criterion ζ, that is $C^\zeta_{\pi(i)} = \sum_{j=1}^{i} p^\zeta_{\pi(j)}$. In both cases a lower value of $q_\zeta(\pi)$ is better.

We used two bi-criterial test instances, with one changeover and one due date criterion, from Iredi et al. in [15]. From these two instances a four-criterial instance with two changeover and two due date criteria was constructed. For the one bi-criterial instance, called instance A from here on, the changeover costs between the jobs were chosen randomly from the interval $[1, 100]$, while for the other one, dubbed instance B, interval $[50, 100]$ was used. The processing times and due dates were chosen according to an often employed scheme from [2]: for each job $j \in [1, 100]$, an integer processing time p_j is drawn randomly from the interval $[1, 100]$, and after all jobs have been assigned a processing time, the due dates for each job j are drawn randomly from the interval $\left[\sum_{j=1}^{100} p_j \cdot (1 - TF - \frac{RDD}{2})\ ,\ \sum_{j=1}^{100} p_j \cdot (1 - TF + \frac{RDD}{2})\right]$. RDD is the relative *Range* of *Due Dates* and determines the size of the interval from which the due dates are drawn. TF is the $Tardiness\ Factor$ and indicates where the center of the above interval is located. For both instances, $RDD = 0.6$ was used; in instance A, we set $TF = 0.4$ and in instance B, $TF = 0.6$. The four-criterial instance that is a combination of instance A and B and is called instance AB.

For the algorithm, we used several parameter configurations. All combinations of population sizes $k \in \{1, 3, 5\}$, $q_0 \in \{0.0, 0.5, 0.9\}$ (see Subsection 2.1), and $\tau_{max} \in \{1, 5, 25, 125, 500, 2500\}$ (see Subsection 2.3) were tested for each of the two bi-criterial instances. The ants used $\alpha = 1, \beta = 5$ for the changeover based probability distributions and $\alpha = 1, \beta = 1$ for the tardiness based ones respectively. The reason for choosing different values of β is that these values are often used in the literature for the corresponding single-criterion versions of the two problems (i.e., TSP and SMTTP). Unless otherwise stated, only one ant constructed a solution in each iteration before a new population was constructed, i.e. the number of ants m per generation is one. For some configurations, using more than one ant was also tested. The four-criterial problem composed of the two bi-criterial ones was only studied for $q_0 = 0.9$, $\tau_{max} = 1$ and $k \in \{1, 3\}$ since these values performed well for the bi-criterial problems. Each run of the algorithm was stopped after 50000 solutions have been constructed.

In the following section we present and compare the median attainment surfaces of the fronts of non-dominated solutions found for 15 runs of the ant algorithm with different randoms seeds (the median attainment surface is the median line of all the attainment surfaces connecting the fronts of non-dominated solutions in every of the 15 runs).

5 Results

We start the evaluation of the performance of the ant algorithm with the results for problem instance A. Figure 1 shows the influence of different values of q_0 on the behaviour of the algorithm for $k = 1$ and $\tau_{max} = 25$. It can be seen that a higher value of q_0 leads to a set of solutions which completely dominates those attained by a smaller q_0 (This behaviour can also be observed for the other parameter settings of τ_{max} and k). This is a good indication of the ants being able to find new good solutions by making only minor adjustments to a solution already located in Q.

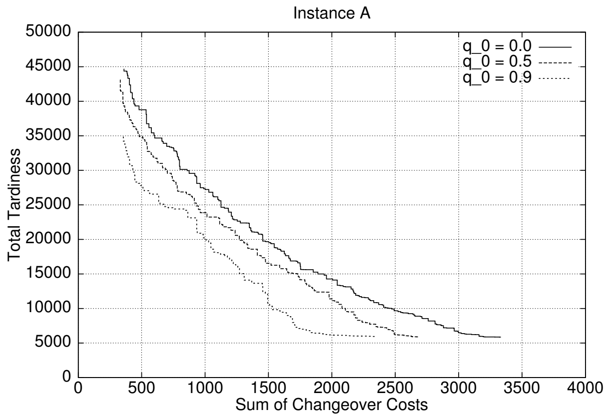

Fig. 1. Median attainment surface for instance A for $q_0 \in \{0.0, 0.5, 0.9\}$ after 50000 ants have built a solution. The other parameters are $\tau_{max} = 25$ and $k = 1$.

The influence of population size k is shown in Figure 2. For instance A, the small population size 1, where exactly one pheromone value in each row equals τ_{max} and all other values are τ_{init}, performed best. Especially when combined with a good heuristic value and a high value of $q_0 = 0.9$, population size 1 keeps the ants very close to the solution from which the pheromone matrices were derived. Note, that the population sizes $k \in \{3, 5\}$ also performed worse than $k = 1$ for $q_0 \in \{0.0, 0.5\}$.

Finally for instance A, we look at the impact of changing the maximum pheromone value τ_{max}, as shown for two different cases in Figures 3 and 4. In Figure 3, where we have a population size of $k = 1$, increasing the maximum pheromone value seems to have a beneficial effect on optimizing the tardiness criterion at the expense of some solution quality in the changeover criterion. A significant effect on the front of non-dominated solutions by different values of τ_{max} is only evident for $k = 1$ however (compare the results for a larger population size $k = 5$ in Figure 4).

It seems that the tardiness criterion and the changeover criterion require different values of τ_{max}. We therefore explored a combination of setting $\tau_{max} = 5$

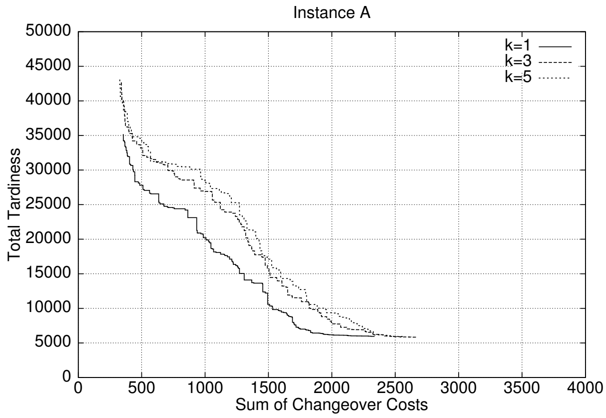

Fig. 2. Median attainment surface for instance A for $k \in \{1, 3, 5\}$ after 50000 ants have built a solution. The other parameters are $\tau_{max} = 25$ and $q_0 = 0.9$.

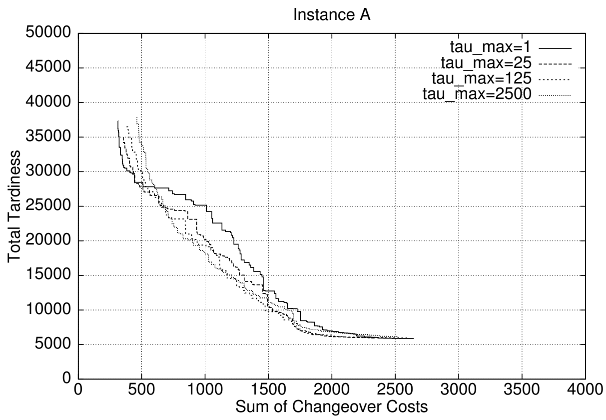

Fig. 3. Median attainment surface for instance A for $\tau_{max} \in \{1, 25, 125, 2500\}$ after 50000 ants have built a solution. The other parameters are $k = 1$ and $q_0 = 0.9$.

for the job×job matrix and $\tau_{max} = 25$ for the job×place matrix, the result of which is shown in Figure 5. The median attainment surface for this case lies between those with the same value of τ_{max} for both criteria.

We now show the results for instance B. The effect of different values of q_0 on the median attainment surface is shown in Figure 6 for $\tau_{max} = 1$ and $k = 3$. Similar as for instance A, a higher value of q_0 outperforms a lower one (this holds also for other values of τ_{max} and k).

Considering the influence of the population size k, the results for instance B differ from those for A (see Figure 7). Whereas for instance A a population size of $k = 1$ performed best, here it performs worst everywhere except in the region of the median attainment surface with the low tardiness values. This

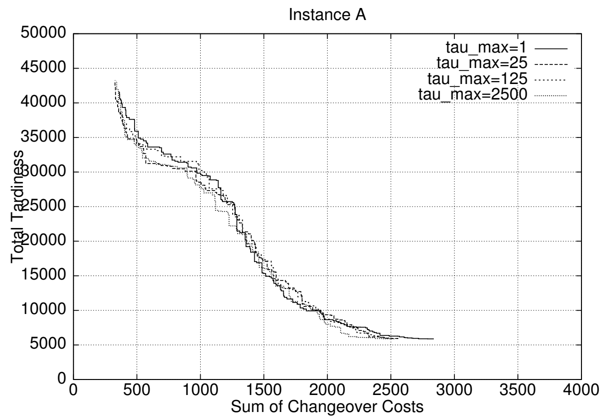

Fig. 4. Median attainment surface for instance A for $\tau_{max} \in \{1, 25, 125, 2500\}$ after 50000 ants have built a solution. The other parameters are $k = 5$ and $q_0 = 0.9$.

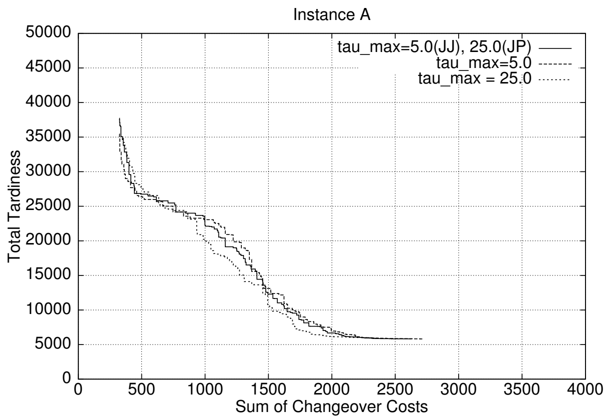

Fig. 5. Median attainment surface for instance A for $\tau_{max} \in \{5, 25\}$ and the combination $\tau_{max}^{JJ} = 5$ and $\tau_{max}^{JP} = 25$) for the job×job and job×place matrix respectively. Results are shown after 50000 ants have built a solution, with $k = 1$ and $q_0 = 0.9$.

behaviour suggests that in comparison to instance A, a more diverse supply of pheromone is necessary to enhance different options for finding schedules with small changeover costs. The reason might be that the changeover costs for instance B are more similar and therefore possibly a relatively large set of different good solutions exist.

The influence of the maximum pheromone value τ_{max} on solution quality for instance B is shown in Figure 8 for $q_0 = 0$ (left) and $q_0 = 0.9$ (right). For $q_0 = 0.0$, a higher maximum pheromone value leads to a better median attainment surface, with $\tau_{max} = 1$ performing comparatively poor (this was observed also for $q_0 = 0.5$). This changes when setting $q_0 = 0.9$, where $\tau_{max} = 1$

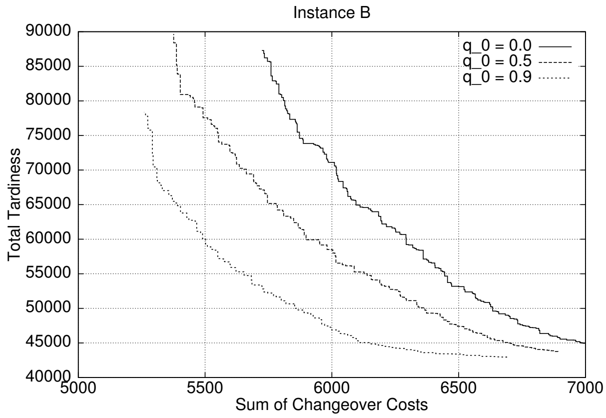

Fig. 6. Median attainment surface for instance B for $q_0 \in \{0.0, 0.5, 0.9\}$ after 50000 ants have built a solution. The other parameters are $\tau_{max} = 1$ and $k = 3$.

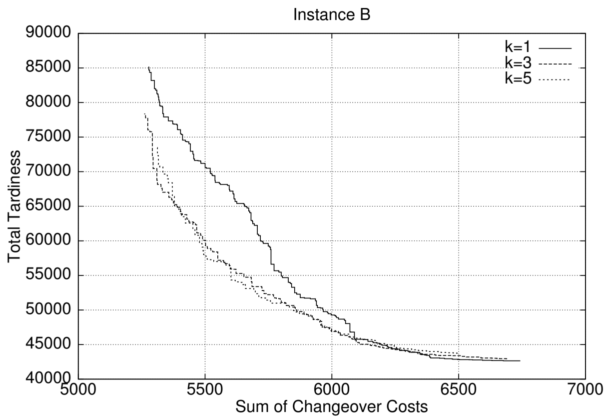

Fig. 7. Median attainment surface for instance B for $k \in \{1, 3, 5\}$ after 50000 ants have built a solution. The other parameters are $\tau_{max} = 1$ and $q_0 = 0.9$.

outperforms the higher maximum pheromone values significantly. This differs from the results for instance A. A reason could be that the possibly different good solutions with respect to changeover costs for instance B might be difficult to find for the ants when using a combination of high τ_{max} and high q_0.

The progression of PACO over time for instance B is shown in Figure 9. It can be seen that in relation to the tardiness criterion, the relative improvement of the changeover costs criterion is by far larger, continuing to explore this outside edge of the set of non-dominated solutions for the entire runtime.

We now move our attention to the four-criterial instance AB. In order to evaluate the performance of the algorithm on this instance, we projected the resulting four-criterial median attainment surface to a 2-dimensional one for

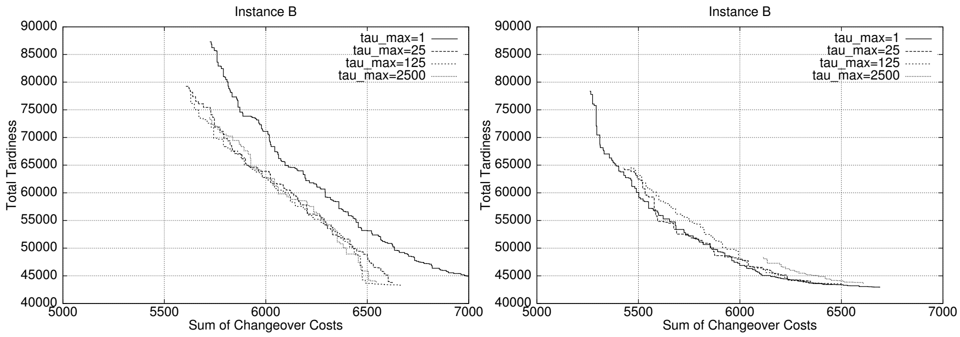

Fig. 8. Median attainment surface for instance B for $\tau_{max} \in \{1, 25, 125, 2500\}$ after 50000 ants have built a solution. The other parameters are $k = 3$ and $q_0 = 0.0$ (left) and $q_0 = 0.9$ (right).

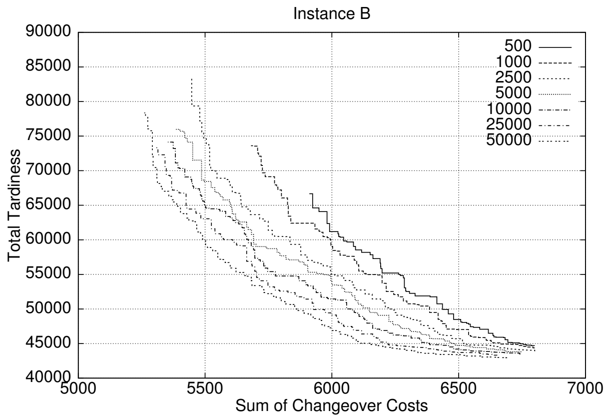

Fig. 9. Median attainment surface for instance B after an indicated number of ants have built solutions. The parameters are $\tau_{max} = 1$, $k = 3$ and $q_0 = 0.9$.

each of the original instances A and B respectively. Thus we can compare the 2-dimensional fronts of the algorithm that ran on instance AB with the algorithm that, with the same parameter settings, was used to solve only A or B exclusively. The results are shown in Figure 10.

As can be seen, the 2-dimensional projection of the 4-dimensional front is worse for both original instances A and B. Note that this comparison is, of course, not completely fair for the algorithm working on instance AB, in the sense that it has to manage a much larger set Q then the algorithm working only on A or B; the size of Q was between 31 and 58 solutions for the bi-criterial instances and ranged from 2739 to 5102 for instance AB, after 50000 iterations. Therefore, it can be argued that each solution of set Q could not be exploited as exhaustively as in the case of a 2-dimensional instance. Despite performing worse than on the bi-criterial instances, the performance of PACO on the four-criterial

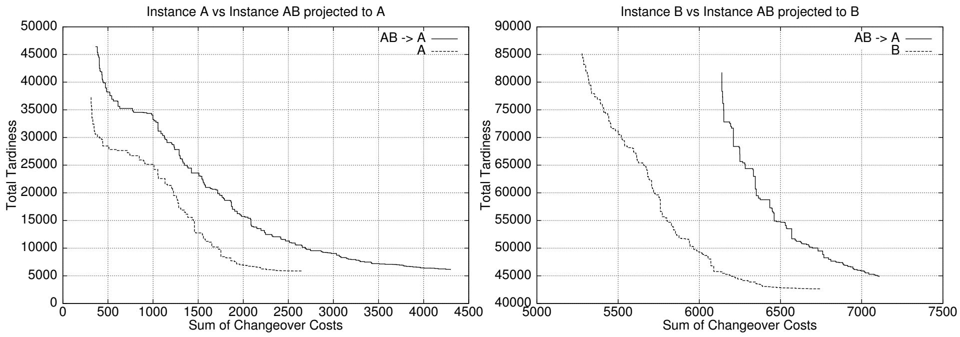

Fig. 10. Median attainment surface for instance A (left) and B (right) and instance AB projected to the corresponding two dimensions after 50000 ants have built a solution. The parameters are $\tau_{max} = 1$, $k = 1$ and $q_0 = 0.9$.

instance is not actually bad, and signifies to us that the general approach is indeed feasible for more than bi-criterial instances.

6 Conclusion

We have successfully modified the PACO algorithm to deal with multi-criteria optimization problems, including the introduction of the Average-Weight-Rank method for constructing the selection probability distribution for the ants and the new derivation of the active population to determine the pheromone matrices. Specifically, the algorithm was tested when used in conjunction with the Single Machine Total Tardiness with changeover costs problem, and the influence of the different parameter settings on the behaviour of the algorithm was investigated.

For future work, we will focus more on many-dimensional problems and the specific handling they require when managing the candidate set for the population, like not biasing the algorithm towards dense parts of the non-dominated set of solutions and limiting the size of the set of solutions from which the population is derived. Also, different possibilities for combining solutions in the pareto-front exist and will be researched further.

References

[1] C. A. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont, editors. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Kluwer Academic Publishers, New York, 2002.

[2] H. Crauwels, C. Potts, and L. V. Wassenhove. Local search heuristics for the single machine total weighted tardiness scheduling problem. *Informs Journal on Computing*, 10:341–350, 1998.

[3] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms.* John Wiley & Sons, Chichester, 2001.

[4] K. Doerner, J. Gutjahr, R. Hartl, C. Strauss, and C. Stummer. Investitionsentscheidungen bei mehrfachen Zielsetzungen und künstliche Ameisen. In Chamoni and et al., editors, *Operations Research Proceedings*, pages 355–362, Berlin, Heidelberg. Springer.

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

[6] K. Doerner, R. Hartl, and M. Reimann. Are COMPETants more competent for problem solving? - the case of a multiple objective transportation problem. In L. S. et al., editor, *Proceedings of the GECCO'01*, page 802, Berlin, Heidelberg, 2001. Morgan Kaufmann.

[7] K. Doerner, R. Hartl, and M. Reimann. Cooperative ant colonies for optimizing resource allocation in transportation. In W. Boers and et al., editors, *Proceedings of the EvoWorkshops 2001*, pages 70 – 79, Berlin, Heidelberg, 2001. Springer.

[8] M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 11–32, London, 1999. McGraw-Hill.

[9] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Systems, Man, and Cybernetics – Part B*, 26:29–41, 1996.

[10] C. Gagné, M. Gravel, and W. Price. Scheduling a single machine where setup times are sequence dependent using an ant-colony heuristic. In *Abstract Proceedings of ANTS'2000,7.-9. September Brussels, Belgium*, pages 157–160, 2000.

[11] L. M. Gambardella, É. Taillard, and G. Agazzi. MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 63–76. McGraw-Hill, London, 1999.

[12] M. Gravel, W. Price, and C. Gagné. Scheduling continous casting of aluminium using a multiple objective ant colony optimization heuristic. *European Journal of Operational Research*, 143:218–229, 2002.

[13] M. Guntsch and M. Middendorf. Applying population based aco to dynamic optimization problems. In *Ant Algorithms, Proceedings of Third International Workshop ANTS 2002*, number 2463 in LNCS, pages 111–122. Springer, 2002.

[14] M. Guntsch and M. Middendorf. A population based approach for ACO. In S. Cagnoni and et al., editors, *Applications of Evolutionary Computing - EvoWorkshops 2002: EvoCOP, EvoIASP, EvoSTIM/EvoPLAN*, number 2279 in LNCS, pages 72–81. Springer, 2002.

[15] S. Iredi, D. Merkle, and M. Middendorf. Bi-criterion optimization with multi colony ant algorithms. In E. Zitzler and et al., editors, *Evolutionary Multi-Criterion Optimization, First International Conference (EMO'01)*, number 1993 in LNCS, pages 359–372, Berlin, Heidelberg. Springer.

[16] C. E. Mariano and E. Morales. MOAQ an ant-Q algorithm for multiple objective optimization problems. In W. Banzhaf and et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, pages 894–901, Orlando, Florida, USA, 13-17 July 1999. Morgan Kaufmann.

[17] D. Merkle and M. Middendorf. An ant algorithm with a new pheromone evaluation rule for total tardiness problems. In *Proceeding of the EvoWorkshops 2000*, number 1803 in LNCS, pages 287–296. Springer Verlag, 2000.

[18] D. Merkle, M. Middendorf, and H. Schmeck. Pheromone evaluation in ant colony optimization. In *Proceedings of the Third Asia-Pacific Conference on Simulated Evolution and Learning (SEAL2000)*, pages 2726–2731. IEEE Press, 2000.