# Simple Probabilistic Population-Based Optimization

Ying-Chi Lin, Martin Clauß, and Martin Middendorf

***Abstract*—A generic scheme is proposed for designing and classifying simple probabilistic population-based optimization (SPPBO) algorithms that use principles from population-based ant colony optimization (PACO) and simplified swarm optimization (SSO) for solving combinatorial optimization problems. The scheme, called SPPBO, identifies different types of populations (or archives) and their influence on the construction of new solutions. The scheme is used to show how SSO can be adapted for solving combinatorial optimization problems and how it is related to PACO. Moreover, several new variants and combinations of these two metaheuristics are generated with the proposed scheme. An experimental study is done to evaluate and compare the influence of different population types on the optimization behavior of SPPBO algorithms, when applied to the traveling salesperson problem and the quadratic assignment problem.**

***Index Terms*—Algorithms, computer science, operations research, population-based ant colony optimization (PACO), population-based metaheuristics.**

## I. Introduction

PROBABILISTIC metaheuristics are very popular methods for solving different types of optimization problems (for overviews, see [6], [8], and [15]). Well-known examples are genetic algorithms (GAs), simulated annealing (SA), population-based incremental learning (PBIL), ant colony optimization (ACO), differential evolution (DE), particle swarm optimization (PSO), and bee colony optimization (BCO). Several of these metaheuristics are motivated or inspired by principles that can be found in biological systems. Typically, the metaheuristics SA, DE, PSO, and BCO are used for function optimization, i.e., for continuous domains, whereas GA, PBIL, and ACO are used for solving combinatorial optimization problems. However, for each of these metaheuristics some variants have also been proposed that can be used to solve the respective other type of optimization problems. It is a common principle of these metaheuristics that they work iteratively and try to find improved solutions in every new iteration.

The metaheuristics that have been mentioned above can be classified into two classes: 1) probability distribution-based metaheuristics, e.g., PBIL and ACO, and 2) population-based metaheuristics, e.g., GA, SA, DE, PSO, and BCO (SA is an extreme case with a population of size one). The probability distribution-based metaheuristics try to learn a probability distribution which is then used to generate the new solutions. Learning in the case of the ACO, for example, means that so-called pheromone information which marks the decisions that correspond to good solutions is aggregated over all iterations [11]. The population-based metaheuristics maintain a population of solutions (sometimes called archive in the literature) from which the new solutions are created by using different types of operations. In the case of the GA, for example, these operations are the mutation of a solution from the population or a crossover that combines different parts of several solutions from the population to create a new solution.

In recent years, the number of new metaheuristics and newly proposed variants and extensions of the above-mentioned metaheuristics has increased strongly. Also, many hybrid metaheuristics which combine principles from several of the above-mentioned metaheuristics have been proposed. Often, new algorithmic ideas are inspired by different natural processes, e.g., biological, physical, or sociological processes. A problem is that many authors use different terms and metaphors to describe algorithmic ideas that are sometimes very similar. Thus, it is difficult even for researchers in this field to keep an overview on all these algorithms and to notice their differences and their common principles. A critical discussion of this "tsunami of 'novel' metaheuristic methods" can be found in a recent paper by Sörensen [36]. He argues that the "time has come for consolidation, to allow the research community to discover the true mechanics underlying these 'novel' methods."

Another problem in the field of metaheuristics is that many algorithms seem to be unnecessarily complicated. One reason could be that authors try to keep their algorithm in close analogy to some corresponding process in nature. Another potential reason is that the quest for novelty leads authors to use complicated combinations of existing algorithmic ideas for designing new algorithms. However, the idea of Occam's razor should not be forgotten which means that more complicated algorithms need a justification and it should be checked if a simpler version might actually work as well. Therefore, it is necessary to show that the different components of an algorithm are useful and it is important to understand their influence on the optimization behavior.

In this paper, we identify a class of simple probabilistic population-based optimization (SPPBO) algorithms for solving combinatorial optimization problems and argue that this class deserves attention for its combination of simplicity and efficiency. The algorithms in this class maintain one or more

Manuscript received December 14, 2014; revised April 14, 2015; accepted June 24, 2015. Date of publication July 1, 2015; date of current version March 29, 2016. This work was supported by the European Social Fund and the Free State of Saxony within the Young Researcher Group "Schwarm-inspirierte Verfahren zur Optimierung, Selbstorganisation und Ressourceneffizienz."

The authors are with the Department of Mathematics and Computer Science, University of Leipzig, Leipzig 04109, Germany (e-mail: yingchilin@gmail.com; mc@informatik.uni-leipzig.de; middendorf@informatik.uni-leipzig.de).

Digital Object Identifier 10.1109/TEVC.2015.2451701

populations of solutions and determine the value of a component of a new solution based on the following two simple operations.

1) SELECT+COPY, i.e., selecting a solution from the population(s) and copying the value of the corresponding component of the solution if this value is feasible.
2) RANDOM, i.e., randomly choosing a (feasible) value from the set of possible values. In addition, some problem-specific heuristic information might also be used to create a new solution.

For describing the identified class of algorithms, we propose a generic scheme that is called SPPBO scheme.

The SPPBO scheme generalizes ideas from population-based ant colony optimization (PACO) and from a newly proposed version of simplified swarm optimization (SSO). PACO was introduced by Guntsch and Middendorf [17] for solving combinatorial optimization problems. SSO was introduced by Yeh [42] for function optimization. The SSO version that is proposed here is suitable for solving combinatorial optimization problems. Both metaheuristics—PACO and SSO—have never been mentioned in the literature as related. We show here that the SPPBO scheme is useful to recognize the similarities (and the differences) between PACO and SSO. Moreover, it is shown that how heuristic information can be used within SPPBO algorithms. Heuristic information has already been used for PACO but not for SSO and its variants.

In accordance with Occam's razor, we perform an experimental study in order to evaluate and compare the influence of different types of populations on the optimization behavior of SPPBO algorithms. Test problems are the traveling salesperson problem (TSP) and the quadratic assignment problem (QAP).

In Section II, a formal definition of optimization problems is given. An overview on population-based optimization methods that are based on the two operations SELECT+COPY and RANDOM is given in Section III. In Section IV, the SPPBO scheme is introduced. The experiments are described in Section V and their results are presented in Section VI. This paper ends with the conclusion in Section VII. Note that this paper significantly extends the preliminary conference version [24].

## II. Optimization Problems

An optimization problem can formally be described by a subset $\bar{V} \subset V^n$, for some set $V$ and a fitness function $f : \bar{V} \to \mathbb{R}$ which assigns to each solution $\mathbf{s} \in \bar{V}$ a positive fitness value $f(\mathbf{s}) > 0$. The aim then is to find a solution from $\bar{V}$ with maximum fitness value. The elements in $V^n$ are solutions and $\bar{V}$ contains the feasible solutions. Thus, a solution $\mathbf{s}$ is a vector $\mathbf{s} = (s_1, \ldots, s_n)$ where each component $s_i$ is an element of $V$. In the case of a combinatorial optimization problem $V = \{v_1, \ldots, v_q\}$ is finite, e.g., in the case of the TSP it is a finite set of cities. Note, that for permutation problems $q = n$ holds.

For a population of solutions $P$, the set of values in $V$ that occur in some vector of $P$ are denoted by $V|P$. $V|P_j$ is the set of all values in $V$ that occur at position $j$ in a vector in $V$. Thus, we write $v \in V|P_j$ if there exists a solution $\mathbf{s} \in P$ with $s_j = v$.

The TSP problem is to find for a given set of cities $V = \{v_1, v_2, \ldots, v_n\}$ and a distance function $d : V^2 \to \mathbb{R}_0^+$, where $d(v_i, v_j)$ is the distance between $v_i$ and $v_j$, a shortest tour which contains each city exactly once, i.e., a vector $\mathbf{s} = (s_1, \ldots, s_n) \in V^n$ which is a cyclic permutation of $V$ so that $\mathbf{s}_i$ is the city that comes after $v_i$ and the length of the tour $\sum_{i=1}^{n} d(v_i, s_i)$ is minimal.

The QAP problem is defined as follows. Given is a set of facilities $V = \{v_1, v_2, \ldots, v_n\}$, a flow function $f : V^2 \to \mathbb{R}_0^+$ such that $f(v_i, v_j) > 0$ is the flow between $v_i$ and $v_j$, $n$ locations $\{1, 2, \ldots, n\}$, and a distance function $d : [1, n]^2 \to \mathbb{R}_0^+$ such that $d(i, j)$ is the distance between locations $i$ and $j$. The problem is then to find an assignment of facilities to locations where each facility is assigned to exactly one location and each location has exactly one facility, i.e., a vector $\mathbf{s} = (s_1, \ldots, s_n) \in V^n$ which is a permutation of $V$ so that $s_i$ is the facility at location $i$ and such that the measure $\sum_{i=1}^{n} \sum_{j=1}^{n} d(i, j) f(s_i, s_j)$ is minimized.

## III. Population-Based Optimization With SELECT+COPY and RANDOM

The minimum functionalities that a population-based metaheuristic needs in order to find new and potentially good solutions for an optimization problem are: 1) that the metaheuristic can use information from solutions that are stored in its population(s) and 2) that the metaheuristic can create new information that is not contained in the solutions of the population(s).

Assume that the value $s_i$ has to be determined to create a new solution $\mathbf{s}$. Then, a simple method to use the information from the population(s) according to 1) would be to select a solution $\mathbf{s}'$ that is a member of a population and copy the value $s'_i$ (i.e., $s_i := s'_i$). In this paper, we consider the case where the selection of $\mathbf{s}'$ is done probabilistically. A simple method for 2) would be to choose a random value for $s_i$ according to some probability distribution over the set of all possible values in $V$. We call the corresponding operations that implement these two methods SELECT+COPY and RANDOM, respectively.

In the literature, there exist two population-based metaheuristics, namely PACO and SSO, that are based on the two operations SELECT+COPY and RANDOM. In the following two sections, we review the literature on PACO and SSO and through our description the similarity of both metaheuristics is elucidated.

### A. Population-Based ACO

The PACO (or P-ACO) metaheuristic was proposed in [17]. The idea of PACO is to keep a small population of solutions that is used to generate the new solutions of the next iteration. This is done by using the operations SELECT+COPY and RANDOM. In addition, some problem-specific heuristic information is used. Formulating PACO within the ACO framework (see [12] and [11]) means that each new solution is generated by an artificial ant which uses artificial pheromone information. A specific property of PACO is that the pheromone

information describes for each solution component essentially how many solutions in the population have a certain value of $V$ at this component. Let $V_i$ be the values of $V$ that occur in the current iteration as the $i$th component of a solution in population $P$.

An ant uses the following probabilistic rule to decide which value should be chosen for the $i$th component of a new solution of some combinatorial optimization problem:

$$p_{ij} = \frac{\left(\tau_{\text{init}} + n_{ij} \times \tau_{\text{solution}}\right)^{\alpha} \eta_{ij}^{\beta}}{\tau_{\text{sum}}} \tag{1}$$

where $p_{ij}$ is the probability to choose a feasible value $v_j \in V$, $\eta_{ij}$ is a heuristic value for the decision, $\alpha \geq 0$ and $\beta \geq 0$ are constants, and $1/\tau_{\text{sum}}$ is the normalization factor to obtain a probability distribution. Thus, $\tau_{\text{sum}} = \sum_{h\text{feasible}} (\tau_{\text{init}} + n_{ih} \times \tau_{\text{solution}})^{\alpha} \eta_{ih}^{\beta}$ where "$h$ feasible" means all indices $h$ for which $v_h$ is feasible.

In the case of the TSP problem, $p_{ij}$ is the probability to choose city $j$ after city $i$ and $\eta_{ij} = 1/d(i,j)$ where $d(i,j)$ is the distance from city $i$ to city $j$. In the case of the QAP problem, $p_{ij}$ is the probability to assign facility $j$ to location $i$. Often for QAP no heuristic is used, i.e., $\beta = 0$ (see [37]).

It has also been proposed for PACO to use—in addition to the normal population—an elitist solution (also called global best solution) which is the best solution found so far by the algorithm [17]. Then, parameter $\tau_{\text{elite}} > 0$ determines the influence of the elitist solution. Moreover, (1) is changed such that the nominator becomes $(\tau_{\text{init}} + \tau_{\text{elitist}} + n_{ij} \times \tau_{\text{solution}})^{\alpha} \eta_{ij}^{\beta}$ for the case that $v_j$ is the $i$th component of the elitist solution and $v_j$ is feasible.

The PACO decision rule can equivalently be stated, as described in the following, for the case that an additional elitist solution is used and $\alpha = 1$ and $\beta = 0$ holds. To compute a new solution $\mathbf{s}^{t+1}$ at iteration $t+1$, the value of the $i$th component $s_i^{t+1}$ is selected probabilistically from: 1) one of the (feasible) values $v_j \in V|P_i^t$ that occur as $i$th component of a solution in the current population $P^t$; 2) the $i$th component $\text{gb}_i^t$ of the current elitist solution $\mathbf{gb}^t$ (if feasible); or 3) a randomly chosen feasible value $v \in V$. The corresponding selection probabilities are $p_{ib,j} := (n_{ij}^t \times \tau_{\text{solution}})/\tau_{\text{sum}}$ for each $v_j \in V|P_i^t$, $p_{\text{elite}} := \tau_{\text{elite}}/\tau_{\text{sum}}$, and $p_r :=$ (number of feasible values)$\times \tau_{\text{init}}/\tau_{\text{sum}}$ where—as before—$1/\tau_{\text{sum}}$ is the normalization factor to obtain a probability distribution. Thus, the decision rule that is used by PACO can be formulated as follows:

$$\mathbf{s}_i^{t+1} = \begin{cases} v_j & \text{with probability } p_{ib,j} \text{ for each} \\ & \text{feasible } v_j \in V|P_i^t \\ \text{gb}_i^t & \text{with probability } p_{\text{elite}}, \text{ if } \text{gb}_i^t \text{ feasible} \\ v & \text{with probability } p_r. \end{cases} \tag{2}$$

Note, that $\tau_{\text{init}}$ determines the probability to select a component randomly from $V$. Thus, the last line in (2) corresponds to operation RANDOM. The probability to make a selection according to operation SELECT+COPY is determined by $\tau_{\text{solution}}$ and possibly by $\tau_{\text{elite}}$. Thus, the first two lines in (2) correspond to SELECT+COPY.

PACO has been proposed for solving combinatorial optimization problems. However, it is straightforward to apply the PACO principle for function optimization. One possibility to create a new solution $\mathbf{s}$ is to choose for component $\mathbf{s}_i$ with probability $p_r$ a (feasible) random value from $V$ according to some probability distribution. Otherwise, a feasible value of $V_i$ is chosen, similarly as in (1), but without $\tau_{\text{init}}$ in the nominator, i.e., $p_{ij} = (n_{ij} \times \tau_{\text{solution}})^{\alpha} \eta_{ij}^{\beta}/\tau_{\text{sum}}$.

In PACO, population $P$ is updated after all new solutions of the current iteration have been generated. Typically, the best of the new solutions (called iteration best solution) is added to the population. An alternative is to add the solution that leads to the strongest increase in the diversity of the population. Several strategies have been proposed in [17] to decide which solution(s) should be removed from the population (see also [20]). Typically, the age-based strategy is used which removes the oldest solution from the population. An alternative is to remove the worst solution.

An empirical comparison has shown that PACO is competitive to ACO (see [17]). Two recent studies compared PACO with several other metaheuristics. The first study is [30] where PACO has been compared with $\mathcal{MAX}$–$\mathcal{MIN}$ ant system (MMAS) from [38]. The conclusion of [30] is that "with the restart procedure and the right configuration, PACO is competitive to the state-of-the-art ACO algorithms with the advantage of finding good solution quality in a shorter computation time." In that study, a variant in PACO was considered that uses local search and a restart mechanism since both are also used by the other state-of-the-art algorithms for the TSP. The second study is [51] where several evolutionary computation methods have been benchmarked by using a newly developed open source framework—called TSP suite—for evaluating and comparing TSP solvers. Weise *et al*. [51] conclude that "prior to these experiments using the TSP suite, some of us would expect a memetic algorithm to be the best hybrid EC method for the TSP. The results, however, tell us that P-ACO is the method of choice."

Considering the fact that PACO is simple, obtains good optimization results, and has been proposed more than ten years ago, it is surprising (as also stated in [30]) that it has been used so far only rarely. Example applications of PACO are single machine scheduling [19], urban waste collection routing problems [5], protein folding [40], DNA sequence optimization [23], sensor placement in water networks [10], and an automotive deployment problem [29]. So far several versions of PACO have been proposed that are suitable for solving dynamic optimization problems [18], [27], [31], [32]. A PACO variant for multiobjective optimization has been proposed in [19] and crowding and fitness sharing strategies for multiobjective PACO have been studied in [1]–[3]. The influence of correlated objectives on multiobjective PACO has been studied in [28]. A PACO variant that is designed for continuous function optimization has been proposed in [35].

### *B. Simplified Swarm Optimization*

SSO is a variant of PSO and has been proposed in [42]. Similar to PACO, SSO also uses only the operations

SELECT+COPY and RANDOM for creating a new solution. SSO uses a swarm of particles $P$ that move within the search space. The current position of a particle $j \in P$ in iteration $t$ is a solution $\mathbf{s}_j^t = (s_{1j}^t, \ldots, s_{kj}^t) \in V^n$. The best found solution so far of particle $j$ at iteration $t$ is called personal best solution and denoted by $\mathbf{pb}_j^t$. The best of the personal best solutions of all particles at iteration $t$ is the global best solution and denoted by $\mathbf{gb}^t$.

To compute a new solution, i.e., to update the position of a particle $j$, the new value $s_{ij}^{t+1}$ of the $i$th component is selected probabilistically from a randomly chosen value $v \in V$ or the $i$th component of one of the following solutions: the current solution $s_{ij}^t$ (i.e., the previous personal position), the personal best solution $\mathrm{pb}_{ij}^t$, and the global best solution $\mathrm{gb}_i^t$. Let $p_r$, $p_{\mathrm{pp}}$, $p_{\mathrm{pb}}$, and $p_{\mathrm{gb}}$ be the corresponding probabilities. SSO uses the decision rule

$$s_{ij}^{t+1} = \begin{cases} s_{ij}^t & \text{with probability } p_{\mathrm{pp}} \\ \mathrm{pb}_{ij}^t & \text{with probability } p_{\mathrm{pb}} \\ \mathrm{gb}_i^t & \text{with probability } p_{\mathrm{gb}} \\ v & \text{with probability } p_r \end{cases} \tag{3}$$

where the first three lines correspond to operation SELECT+COPY and the last line corresponds to operation RANDOM, i.e., $v \in V$ is selected randomly with uniform probability from all possible (and feasible) values.

A simpler version of SSO has been proposed in [43] where the influence of the personal best solution has been removed from the decision rule, i.e., line 2 in (3) is deleted. In another variant of SSO the set of particles is partitioned into groups and for each group a group best solution, i.e., the best of the personal best solutions of all particles in the group, is used [46]. The corresponding decision rule is

$$s_{ij}^{t+1} = \begin{cases} s_{ij}^t & \text{with probability } p_{\mathrm{pp}} \\ \mathrm{pb}_{ij}^t & \text{with probability } p_{\mathrm{pb}} \\ \mathrm{grb}_i^t & \text{with probability } p_{\mathrm{grb}} \\ \mathrm{gb}_i^t & \text{with probability } p_{\mathrm{gb}} \\ v & \text{with probability } p_r \end{cases} \tag{4}$$

where $\mathbf{grb}_i^t$ is the group best solution of the group that contains particle $j$ at iteration $t$.

The probabilities $p_{\mathrm{pp}}$, $p_{\mathrm{pb}}$, $p_{\mathrm{grb}}$, $p_{\mathrm{gb}}$, and $p_r$ can either be fixed or adapted during a run of the SSO algorithm. Fixed probabilities have been used, e.g., in [42] with values $p_{\mathrm{pp}} = 0.1$, $p_{\mathrm{pb}} = 0.3$, $p_{\mathrm{gb}} = 0.5$, and $p_r = 0.1$ and in [46] with values $p_{\mathrm{pp}} = 0.15$, $p_{\mathrm{pb}} = 0.25$, $p_{\mathrm{gb}} = 0.35$, and $p_r = 0.25$. Probabilities that are adapted with respect to the fitness of the solutions found so far were used in [43].

Several applications of SSO and its variants have been described in a series of papers by Yeh and colleagues. An SSO for multiple multilevel redundancy allocation in series systems has been described in [42]. Disassembly sequencing problems have been solved with SSO in [43], [45], and [46]. The mining of classification rules was investigated in [4], [25], and [44]. Image classification with SSO has been studied in [34]. Recent applications of the SSO are the grid-computing reliability and service makespan problem [50], the K-harmonic means problem for mining data [48], and the series–parallel redundancy allocation problem [47], [49].

It should be mentioned that SSO has some similarities to variants of discrete PSO that have been proposed in recent years. An example is the jumping frogs algorithm (JFA) that was proposed in 2008 in [26]. An alternative name for this algorithm is jumping PSO [9]. In JFA, a particle makes one of the following types of moves: a random move with probability $p_r$, a move in direction of the personal best position with probability $p_{\mathrm{pb}}$, a move in direction of the neighborhood best position with probability $p_{\mathrm{nb}}$, and a move in direction of the global best position with probability $p_{\mathrm{gb}}$. If this probabilistic choice is made for each dimension and if "move in direction" means to choose the corresponding value of the solution then JFA is very similar to SSO.

**Algorithm 1:** SPPBO

```
Init
repeat
  for every SCE A ∈ 𝒜 do
    create(A)
  end for
  for every population P ∈ 𝒫 do
    update(P)
  end for
until Stop criterion satisfied
```

## IV. Simple Probabilistic Population-Based Optimization

A general scheme for simple probabilistic population-based metaheuristics is proposed in this section. The scheme is called general SPPBO scheme. A sketch of the general SPPBO scheme is given in Section IV-A before its details are described. Two restricted variants of the general SPPBO scheme and example algorithms are discussed in Section IV-B.

### A. General SPPBO Scheme

The elements of the general SPPBO scheme are: 1) a set $\mathcal{A}$ of solution creating entities (SCEs) where an SCE corresponds to an ant in PACO and ACO or a particle in SSO and PSO; 2) a set of populations of solutions $\mathcal{P}$; 3) the set of possible values $V$ as defined in Section II; 4) heuristic information $\eta$; 5) an initialization method Init; and 6) a stopping criterion Stop. Thus, a general SPPBO algorithm is described by a tuple $(\mathcal{A}, \mathcal{P}, V, \eta, \mathrm{Init}, \mathrm{Stop})$.

An algorithm that is defined by the general SPPBO scheme works as follows (see also Algorithm 1). First, the initialization method Init is applied which creates initial solutions for the populations if necessary. Then, the algorithm executes a sequence of iterations. Each iteration consists of two steps. In the first step, each SCE $A \in \mathcal{A}$ creates one new solution. The solution creation is influenced by $\mathcal{P}$, $V$, and $\eta$. In the second step, each population in $\mathcal{P}$ is updated. The algorithm stops when the stopping criterion Stop is satisfied.

Each SCE $A \in \mathcal{A}$ is characterized in SPPBO by a single attribute: the solution creating function $\text{create}_A$. Let $k_{\text{new}} = |\mathcal{A}|$, i.e., $k_{\text{new}}$ equals the number solutions that are newly created at each iteration. In this paper, we consider the type of solution creating function that is used by the ants in PACO [see (1)].

Each population $P \in \mathcal{P}$ is characterized in SPPBO by three attributes.

1) The range $\text{Range}_P$ defines which SCEs in $\mathcal{A}$ are influenced by $P$. Hence, $\text{Range}_P \subseteq \mathcal{A}$. In this paper, we consider populations with two different types of ranges: a) global and b) personal. A population $P$ with range type global influences every $A \in \mathcal{A}$, i.e., $\text{Range}_P = \mathcal{A}$. A population with range type personal influences only a single SCE in $\mathcal{A}$.
2) The weight $w_P \geq 0$ determines the strength of the influence that $P$ has on the creation of a new solution by an SCE in $\text{Range}_P$.
3) The update rule $\text{update}_P$ determines which solutions are removed from $P$ and which solutions enter $P$. Since we want to consider simple metaheuristics we assume in this paper that only solutions that have been created in the current iteration by an SCE in $\text{Range}_P$ can enter $P$.

The set $V$ has two attributes in the general SPPBO scheme: a probability distribution over $V$ denoted by $\text{Prob}_V$ and a weight $w_r \geq 0$. If an algorithm selects a value only from a feasible subset of $V$, then $\text{Prob}_V$ defines a probability distribution over the feasible subset of $V$. In this paper, we consider only the case where $\text{Prob}_V$ is the uniform distribution over $V$ and its feasible subsets, respectively. As explained later, the weight $w_r$ determines the influence of RANDOM on the creation of a new solution.

### *B. Restricted SPPBO Schemes and Example Algorithms*

Since the aim of this paper is to study simple SPPBO algorithms, we introduce three restricted SPPBO schemes in this section. The restricted SPPBO schemes can be used to describe subclasses of SPPBO algorithms simpler as it is possible with the general SPPBO scheme. Also, several example SPPBO algorithms are discussed. These algorithms are variants of PACO and of a newly proposed version of SSO. First, we make three assumptions that hold for every SPPBO algorithm considered in this section.

1) All SCEs use the same solution creating function.
2) The solution creating function is based on a probabilistic rule that is an extension of (1) from PACO. For an SCE $A \in \mathcal{A}$ to determine the $i$th component of a new solution $\text{create}_A$ uses the probabilistic rule

$$p_{ij} = \frac{\left(w_r + \sum_{P \in \mathcal{P}, A \in \text{Range}_P} w_P \times n_{Pij}\right)^{\alpha} \eta_{ij}^{\beta}}{\tau_{\text{sum}}} \quad (5)$$

where $p_{ij}$ is the probability to choose a feasible value $v_j \in V$, $n_{Pij}$ is the number of solutions in population $P$ that have value $v_j$ in their $i$th component, $\eta_{ij}$ is the heuristic value for the decision, $\alpha \geq 0$ and $\beta \geq 0$ are constants, and $1/\tau_{\text{sum}}$ is the normalization factor for the probability distribution.

3) $\text{Prob}_V$ is the uniform distribution on the feasible subsets of $V$.

Note that (5) generalizes the following principles of PACO: 1) population(s) of solutions are used to generate new solutions from it by operation SELECT+COPY; 2) the selection probabilities are based on the frequencies $n_{Pij}$ of the different values that occur in the corresponding component in the solutions of population $P$; and 3) operation RANDOM is used to integrate new information into new solutions. Also note that parameters $w_r$ and $w_p$, $P \in \mathcal{P}$ can be used to determine the relative influence of RANDOM and that of the different populations. As in ACO and PACO, parameters $\alpha$ and $\beta$ are used to determine the relative influence of RANDOM plus SELECT+COPY and that of the heuristic information.

A simple form of SPPBO algorithms has only global populations, i.e., $\text{Range}_P = \mathcal{A}$ for all $P \in \mathcal{P}$. For such algorithms, a scheme can be used where the range parameter is omitted. This scheme is called SPPBO with global populations and is described by $(\mathcal{A}, \mathcal{P}^{\text{gl}}, V, \eta, \text{Init}, \text{Stop})$ where $\mathcal{P}^{\text{gl}}$ is used instead of $\mathcal{P}$ to differentiate it from the general SPPBO scheme. For the SPPBO with global populations, the probability to choose (a feasible) value $v_j \in V$ for the $i$th component of a new solution is

$$p_{ij} = \frac{\left(w_r + \sum_{P \in \mathcal{P}^{\text{gl}}} w_P \cdot n_{Pij}\right)^{\alpha} \eta_{ij}^{\beta}}{\tau_{\text{sum}}} \quad (6)$$

where the notation is the same as for (5). Different SPPBO algorithms are possible under the SPPBO with global populations scheme that differ in the number and sizes of global populations and the population update rules. Two example algorithms are considered in the following.

Example 1 is an algorithm with only one global population $P$ where $\text{update}_P$ is defined such that population $P$ consists of every iteration best solution from the last $k$ iterations. It is easy to show that this algorithm is exactly the PACO algorithm with the age-based strategy and parameter values $\tau_{\text{init}} = w_r$ and $\tau_{\text{solution}} = w_P$.

Example 2 is an SPPBO algorithm with two global populations $P_1$ and $P_2$. Let $\text{update}_{p_1}$ be defined such that population $P_1$ consists of every iteration's best solution from the last $k$ iterations. Furthermore, let $\text{update}_{P_2}$ be defined such that $P_2$ contains only the best found solution so far, i.e., the elitist solution. Clearly, this algorithm equals the PACO algorithm with an elitist solution where $\tau_{\text{init}} = w_r$, $\tau_{\text{solution}} = w_{P_2}$, and $\tau_{\text{elite}} = w_{P_2}$.

Now, we consider a variant of SPPBO which has only personal populations and where each SCE has the same number and types of personal populations. For such algorithms, it is enough to define each type of personal population only once (and not for each SCE). Moreover, the range parameter of a population can be omitted. To differentiate the corresponding scheme from the general SPPBO scheme, $\mathcal{P}^{\text{ps}}$ is used instead of $\mathcal{P}$. Let $\mathcal{A} = \{A_1, \ldots, A_{k_{\text{new}}}\}$. Each $\mathbf{P} \in \mathcal{P}^{\text{ps}}$ is a list of populations $(\mathbf{P}_1, \ldots, \mathbf{P}_{k_{\text{new}}})$ with $\text{Range}_{\mathbf{P}_i} = \{A_i\}$ for $i \in \{1, \ldots, k_{\text{new}}\}$. The update functions for all populations $\mathbf{P}_1, \ldots, \mathbf{P}_{k_{\text{new}}}$ are the same with the exception that the solutions which are included into $\mathbf{P}_i$ are taken from $A_i$, for $i \in \{1, \ldots, k_{\text{new}}\}$. The scheme is called SPPBO with personal

populations and is described by $(\mathcal{A}, \mathcal{P}^{\text{ps}}, V, \eta, \text{Init}, \text{Stop})$. The probability to choose a (feasible) value $v_j$ for the $i$th component of the new solution $s_l$, $l \in [1, k_{\text{new}}]$ is then

$$p_{ij} = \frac{\left(w_r + \sum_{\mathbf{P} \in \mathcal{P}^{\text{ps}}} w_{\mathbf{P}_l} \cdot n_{\mathbf{P}_l ij}\right)^{\alpha} \eta_{ij}^{\beta}}{\tau_{\text{sum}}}. \quad (7)$$

Now, we consider a scheme which is a combination of the two schemes that were considered before in this section. This scheme is called SPPBO with global populations and personal populations and is described by $(\mathcal{A}, \mathcal{P}^{\text{gl}}, \mathcal{P}^{\text{ps}}, V, \eta, \text{Init}, \text{Stop})$, where $\mathcal{P}^{\text{gl}}$ is the set of global populations and $\mathcal{P}^{\text{ps}}$ is the set of personal populations as defined before. The probability to choose a (feasible) value $v_j$ for the $i$th component of the new solution $s_l$, $l \in [1, k_{\text{new}}]$ is then

$$p_{ij} = \frac{\left(w_r + \sum_{P \in \mathcal{P}^{\text{gl}}} w_P \cdot n_{Pij} + \sum_{\mathbf{P} \in \mathcal{P}^{\text{ps}}} w_{\mathbf{P}_l} \cdot n_{\mathbf{P}_l ij}\right)^{\alpha} \eta_{ij}^{\beta}}{\tau_{\text{sum}}}. \quad (8)$$

Note that (8) is a combination of (6) and (7). In the following, we discuss four examples (Examples 3–6) of SPPBO algorithms with global populations and personal populations. The first two examples can be seen as variants of SSO algorithms. The last two examples can be seen as combinations of PACO and the new variants of SSO.

Example 3 is an SPPBO with one global population $P$ and one personal population $\mathbf{P}^{\text{pers}}$. $P$ contains only the elitist solution and $\mathbf{P}_i^{\text{pers}}$ contains only the solution that its corresponding SCE $A_i$ has created in the last iteration before the current iteration. In the following, we explain that this SPPBO algorithm can be considered as a variant of the simplified SSO, i.e., the SSO which uses (3) but without line 2 (see Section III-B). Consider the case $\alpha = 1$ and $\beta = 0$. In this case, the SPPBO algorithm is very similar to the simplified SSO. However, a difference stems from the fact that for many combinatorial optimization problems not all values in $V$ might be feasible when a component of a new solution is determined. For permutation problems, for example, each value of $V$ has to be chosen as exactly one component of a solution. In the SPPBO scheme, each value in $V$ has weight $w_r$, the global best solution has weight $w_P$, and the personal previous solution has weight $w_{\text{pers}}$. For computing the selection probabilities, only the weights of feasible values in $V$ and only solutions that have a feasible value in their respective component are considered. Thus, the weights $w_r$, $w_P$, and $w_{\text{pers}}$ in SPPBO correspond to the probabilities $p_r$, $p_{\text{gb}}$, and $p_{\text{pp}}$ of the simplified SSO, respectively. For the case $\beta > 0$, the SPPBO scheme shows how heuristic information can be integrated into the proposed version of SSO for combinatorial optimization.

Example 4 is an SPPBO with one global population and two personal populations. The global population $P$ contains only the elitist solution. Personal population $\mathbf{P}_1^{\text{pers}}$ contains only the solution from the last iteration of the corresponding SCE and personal population $\mathbf{P}_2^{\text{pers}}$ contains the personal best solution of the corresponding SCE. Note that this SPPBO algorithm is similar to the original SSO that uses (3). The weights $w_r$, $w_P$, $w_{\mathbf{P}_1^{\text{pers}}}$, and $w_{\mathbf{P}_2^{\text{pers}}}$ of SPPBO correspond to the probabilities $p_r$, $p_{\text{gb}}$, $p_{\text{pp}}$, and $p_{\text{pb}}$ of the SSO, respectively.

Example 5 is an SPPBO algorithm with two global populations $P_1$ and $P_2$ and one personal population $\mathbf{P}^{\text{pers}}$. $P_1$ contains the global best solution and $P_2$ contains the iteration best solutions from each of the last $k$ iterations. $\mathbf{P}^{\text{pers}}$ contains the solution from the last iteration of its corresponding SCE. This algorithm is a combination of the introduced version of the simplified SSO (Example 3) and PACO.

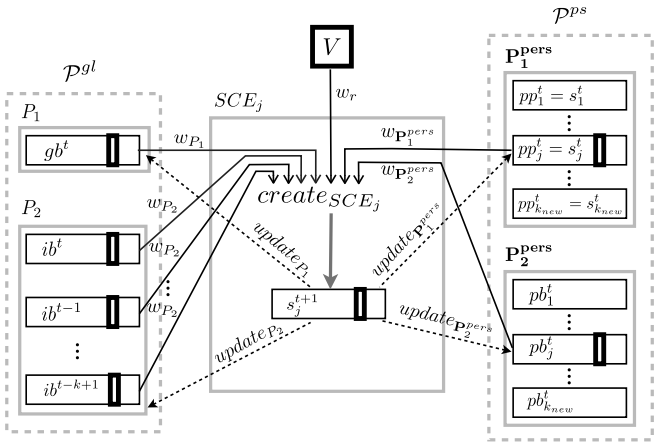


Fig. 1. Algorithm from Example 6 (SSO-PACO): relation between populations and $\text{SCE}_j$; arrows show which solutions from the populations influence the creation of $s_j^{t+1}$; dotted arrows show the populations where $s_j^{t+1}$ might be included; and boxes with broad lines show which components influence a single component of $s_j^{t+1}$.

Example 6 is similar to Example 5, however, with the difference that the algorithm has two personal populations $\mathbf{P}_1^{\text{pers}}$ and $\mathbf{P}_2^{\text{pers}}$. $\mathbf{P}_1^{\text{pers}}$ contains the solution from the last iteration of the corresponding SCE and $\mathbf{P}_2^{\text{pers}}$ contains the personal best solution of the corresponding SCE. This algorithm is a combination of the introduced version of the SSO (Example 4) and PACO. Fig. 1 illustrates this algorithm.

Altogether, it can be seen that the general SPPBO scheme and its restricted versions can be used to describe very basic population-based metaheuristics that combine ideas from PACO and SSO. In order to investigate the influence of different types of populations on the optimization behavior of SPPBO algorithms, we performed an experimental study in this paper.

## V. Experiments

In this section, we describe the experiments that have been done to evaluate various SPPBO algorithms. The algorithms are: 1) the PACO with elitist solution; 2) the SPPBO versions of the SSO algorithms; and 3) the newly proposed combinations of SSO and PACO. We investigate the influence of the different types of populations on the optimization behavior of the algorithms. As test problems the TSP and the QAP are used. For the experiments, the number of SCEs which equals the number of new solutions in each iteration was set to $|\mathcal{A}| = k_{\text{new}} = 10$. This is a typical value for the number of ants in ACO or PACO (see [13]). For a global population that contains the best solution of each of the last $k$ iterations, the value $k = 5$ was used for the TSP. The reason is that such a small value is known to be good for solving the TSP with PACO (see [17]) and additional tests have shown that this also holds

TABLE I
EXAMPLE OF TEST PARAMETER SETTING WITH $w_{\text{total}} = 1.5$ AND $k_{\text{new}} = 10$ NEW SOLUTIONS, AND GLOBAL POPULATION OF SIZE $k = 5$

| | **SPPBO-1** | **SPPBO-2** | **SPPBO-3** | **SPPBO-4** | **SPPBO-5** | **SPPBO-6** | **SPPBO-7** |
|---|---|---|---|---|---|---|---|
| $\boldsymbol{w_r}$ | $1/(n-1)$ | $1/(n-1)$ | $1/(n-1)$ | $1/(n-1)$ | $1/(n-1)$ | $1/(n-1)$ | $1/(n-1)$ |
| $\boldsymbol{w_{elite}}$ | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| $\boldsymbol{w_{ib}}$ | 0.3 | 0.15 | 0.15 | 0.1 | – | – | – |
| $\boldsymbol{w_{pb}}$ | – | 0.075 | – | 0.05 | 0.15 | – | 0.075 |
| $\boldsymbol{w_{pp}}$ | – | – | 0.075 | 0.05 | – | 0.15 | 0.075 |

TABLE II
BEST COMBINATION OF $w_{\text{total}}$ AND $w_{\text{elite}}$ VALUES FOR EACH ALGORITHM AT ITERATION 10 000 FOR EACH TSP INSTANCE AND CORRESPONDING AVERAGE SOLUTION QUALITY WITH STANDARD DEVIATION (SD). EACH NUMBER IN THE S-MATRIX HEADER CORRESPONDS TO AN SPPBO ALGORITHM, E.G., 1 = SPPBO-1. A MARKING "X" IN S-MATRIX DENOTES THAT THE CORRESPONDING ALGORITHM IN THE ROW HAS A SIGNIFICANTLY SMALLER AVERAGE TOUR LENGTH THAN THE CORRESPONDING ALGORITHM IN THE COLUMN (WILCOXON-SIGNED RANK TEST WITH $p$-VALUE = 0.01/42)

| **Instance** | **Algorithm** | $w_{total}$ | $w_{elite}$ | **Average±SD** | **S-Matrix** | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| rat195 | SPPBO-1 | 0.75 | 0.1 | 2340.1±10.1 | - | - | - | - | X | X | - |
| | SPPBO-2 | 1.5 | 0.1 | 2339.7±4.5 | - | - | - | - | X | X | - |
| | SPPBO-3 | 1.5 | 1 | 2338.9±4.5 | - | - | - | - | X | X | - |
| | SPPBO-4 | 3 | 0.1 | 2339.7±6.3 | - | - | - | - | X | X | - |
| | SPPBO-5 | 6 | 0.1 | 2346.9±8.6 | - | - | - | - | - | - | - |
| | SPPBO-6 | 12 | 1 | 2355.0±15.4 | - | - | - | - | - | - | - |
| | SPPBO-7 | 12 | 0.1 | 2338.3±4.7 | - | - | - | - | X | X | - |
| a280 | SPPBO-1 | 1.5 | 0.1 | 2604.4±29.1 | - | - | - | - | - | X | - |
| | SPPBO-2 | 1.5 | 10 | 2607.8±34.1 | - | - | - | - | - | X | - |
| | SPPBO-3 | 3 | 0.1 | 2591.7±19.0 | - | - | - | - | X | X | - |
| | SPPBO-4 | 12 | 1 | 2606.5±26.3 | - | - | - | - | - | X | - |
| | SPPBO-5 | 12 | 0.1 | 2615.1±16.2 | - | - | - | - | - | X | - |
| | SPPBO-6 | 96 | 1 | 2641.5±43.2 | - | - | - | - | - | - | - |
| | SPPBO-7 | 24 | 1 | 2603.8±12.6 | - | - | - | - | - | X | - |
| pr439 | SPPBO-1 | 6 | 0.1 | 109709.6±1279.1 | - | - | - | - | - | X | - |
| | SPPBO-2 | 12 | 10 | 109930.9±1198.7 | - | - | - | - | - | X | - |
| | SPPBO-3 | 24 | 10 | 109729.8±1096.7 | - | - | - | - | - | X | - |
| | SPPBO-4 | 48 | 10 | 110088.1±1206.1 | - | - | - | - | - | X | - |
| | SPPBO-5 | 48 | 1 | 110190.9±783.5 | - | - | - | - | - | X | - |
| | SPPBO-6 | 96 | 10 | 112453.9±2012.9 | - | - | - | - | - | - | - |
| | SPPBO-7 | 96 | 10 | 110230.9±833.5 | - | - | - | - | - | X | - |

for the other test algorithms with such a global population. For the QAP, the values $k \in \{5, 10, 25\}$ have been tested.

To describe the setting of the weight parameters, let $w_{\text{total}}$ be the total weight of all solutions in all populations, however, excluding the weight of the elitist solution if it is used. In other words, $w_{\text{total}}$ is the total weight of the solutions in a global population that contains the $k$ last iteration best solutions, the personal previous solutions, and the personal best solutions (if the algorithm has these populations). Since we want to find out how important these populations are for the optimization behavior of SPPBO algorithms, it was decided to split their influence on the solution construction equally between all these populations that an algorithm has. Therefore, for the case of the PACO $w_{\text{total}}$ is the total weight of the last $k$ iteration best solutions, i.e., $w_{ib} = w_{\text{total}}/k$. For algorithms with additional personal previous solutions and/or personal best solutions, the total weight $w_{\text{total}}$ is split equally between the corresponding three or two types of solutions. For example, when a global population with the last $k$ iteration best solutions and a personal population with the personal previous solution are used then $w_{ib} = 0.5 \cdot w_{\text{total}}/k$ and $w_{\text{pp}} = 0.5 \cdot w_{\text{total}}/k_{\text{new}}$. Note that $w_{\text{pp}} = 0.5 \cdot w_{ib}$ because there are $k_{\text{new}} = 10$ personal best solutions (one in each of the $k_{\text{new}} = 10$ personal populations) but only $k = 5$ solutions in the global population. In the experiments, the values $w_{\text{total}} \in \{0.75, 1.5, 3, 6, 12, 24, 48, 96, 192\}$ were tested. For SPPBO-5 and SPPBO-7 and the QAP where large values of $w_{\text{total}}$ are good, the values $w_{\text{total}} \in \{384, 768, 1536\}$ were tested. The reason that these large values were not tested for the other cases is that the results clearly indicate that such large values are not good for those cases.

For the weight of the elitist solution, the values $w_{\text{elite}} \in \{0.1, 1, 10\}$ have been tested. The random influence sets to $w_r = 1/(n-1)$ for the TSP and $w_r = 1/n$ for the QAP.

TABLE III
BEST COMBINATION OF POPULATION SIZE $k$ WITH $w_{\text{total}}$ AND $w_{\text{elite}}$ VALUES FOR EACH ALGORITHM AT ITERATION 10 000 FOR EACH QAP INSTANCE AND CORRESPONDING AVERAGE SOLUTION QUALITY WITH STANDARD DEVIATION (SD). EACH NUMBER IN THE S-MATRIX HEADER CORRESPONDS TO AN SPPBO ALGORITHM, E.G., 1 = SPPBO-1. A MARKING "X" IN S-MATRIX DENOTES THAT THE CORRESPONDING ALGORITHM IN THE ROW HAS A SIGNIFICANTLY BETTER AVERAGE SOLUTION QUALITY THAN THE CORRESPONDING ALGORITHM IN THE COLUMN (WILCOXON-SIGNED RANK TEST WITH $p$-VALUE = 0.01/42)

| Instance | Algorithm | $k$ | $w_{total}$ | $w_{elite}$ | Average±SD | S-Matrix | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| lipa60a | SPPBO-1 | 5 | 12 | 10 | 108473.5±65.7 | - | - | - | - | X | X | X |
| | SPPBO-2 | 25 | 24 | 0.1 | 108445.3±63.8 | - | - | - | - | X | X | X |
| | SPPBO-3 | 5 | 48 | 10 | 108476.5±62.9 | - | - | - | - | X | X | X |
| | SPPBO-4 | 25 | 48 | 0.1 | 108468.9±62.2 | - | - | - | - | X | X | X |
| | SPPBO-5 | - | 192 | 10 | 108668.6±47.0 | - | - | - | - | - | X | X |
| | SPPBO-6 | - | 0.75 | 10 | 108879.9±76.7 | - | - | - | - | - | - | - |
| | SPPBO-7 | - | 768 | 1 | 108826.7±47.6 | - | - | - | - | - | X | - |
| lipa70a | SPPBO-1 | 5 | 24 | 1 | 171544.4±77.8 | - | - | - | - | X | X | X |
| | SPPBO-2 | 5 | 48 | 0.1 | 171545.0±76.3 | - | - | - | - | X | X | X |
| | SPPBO-3 | 15 | 48 | 1 | 171532.8±75.0 | - | - | - | - | X | X | X |
| | SPPBO-4 | 25 | 48 | 0.1 | 171513.9±92.4 | - | - | - | - | X | X | X |
| | SPPBO-5 | - | 192 | 10 | 171832.3±63.1 | - | - | - | - | - | X | X |
| | SPPBO-6 | - | 0.75 | 10 | 172135.9±98.3 | - | - | - | - | - | - | - |
| | SPPBO-7 | - | 768 | 1 | 172046.3±71.0 | - | - | - | - | - | X | - |
| lipa90a | SPPBO-1 | 25 | 24 | 0.1 | 363744.0±124.9 | - | X | - | X | X | X | X |
| | SPPBO-2 | 25 | 48 | 1 | 363846.1±113.9 | - | - | - | - | X | X | X |
| | SPPBO-3 | 25 | 48 | 10 | 363806.6±97.5 | - | - | - | X | X | X | X |
| | SPPBO-4 | 15 | 96 | 0.1 | 363928.2±120.3 | - | - | - | - | X | X | X |
| | SPPBO-5 | - | 384 | 10 | 364420.5±63.3 | - | - | - | - | - | X | X |
| | SPPBO-6 | - | 0.75 | 10 | 365101.3±145.5 | - | - | - | - | - | - | - |
| | SPPBO-7 | - | 1536 | 0.1 | 364834.6±101.4 | - | - | - | - | - | X | - |

Note that the small difference between these values for TSP and QAP is not very relevant. The reason for this setting is to make the results comparable with the literature (see [17]).

The different types of algorithms with their identifiers and the corresponding test parameter values can be seen in Table I for the case of $w_{\text{total}} = 1.5$. Note that SPPBO-1 is PACO (Example 2 in Section IV-B), SPPBO-6 is the new version of the simplified SSO (sSSO) (Example 3 in Section IV-B), and SPPBO-7 is the new version of SSO (Example 4 in Section IV-B). The proposed combination of simplified SSO and PACO is SPPBO-3 (sSSO–PACO) (Example 5 in Section IV-B), and the proposed combination of SSO and PACO (SSO–PACO) is SPPBO-4 (Example 6 in Section IV-B). SPPBO-5 is a new simplified form of SSO that is proposed here (nsSSO) and its combination with PACO is SPPBO-2 (nsSSO–PACO).

As heuristic information for the TSP $\eta_{ij} = 1/d_{ij}$ has been used where $d_{ij}$ is the distance between cities $i$ and $j$. Parameters $\alpha$ and $\beta$ have been set to $\alpha = 1$ and $\beta = 5$ which are typical values for the TSP (see [13]). For the QAP, no heuristic information has been used which is typical for many ACO algorithms for the QAP (see [37]). Thus, $\alpha = 1$ and $\beta = 0$ have been used for the QAP.

As test instances, the following TSP instances have been taken from the standard library TSPLIB (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/): rat195, a280, and pr439. For the QAP, the following test instances have been taken from the standard library QAPLIB (http://www.opt.math.tu-graz.ac.at/qaplib): lipa60, lipa70, and lipa90. For all test instances, the number in the name denotes the number of cities and locations, respectively. Each test run was terminated after 10 000 iterations and has been repeated 50 times.

## VI. RESULTS

In this section, we present the experimental results. In Section VI-A, the different SPPBO algorithms are compared. Section VI-B shows more details on the optimization behavior of algorithms SPPBO-1 and SPPBO-7. In particular, the influence of parameters $w_{\text{total}}$ and $w_{\text{elite}}$ on the optimization behavior is investigated. A general discussion on the results can be found in Section VI-C.

### A. Comparison of SPPBO Algorithms

The solution quality (i.e., the best found solution and averaged over the different test runs) of the different SPPBO algorithms is shown in Table II for the TSP instances and in Table III for the QAP instances. For each algorithm, the results that are shown have been obtained with its best combination of parameter values, to make the comparison fair. The tables also show if the differences between the algorithms are significant. For the TSP, it can be seen that algorithm SPPBO-6 is the worst of all algorithms for nearly every problem instance. Algorithm SPPBO-5 is the second worst algorithm because it is worse than SPPBO-1 to -4 and SPPBO-7 for

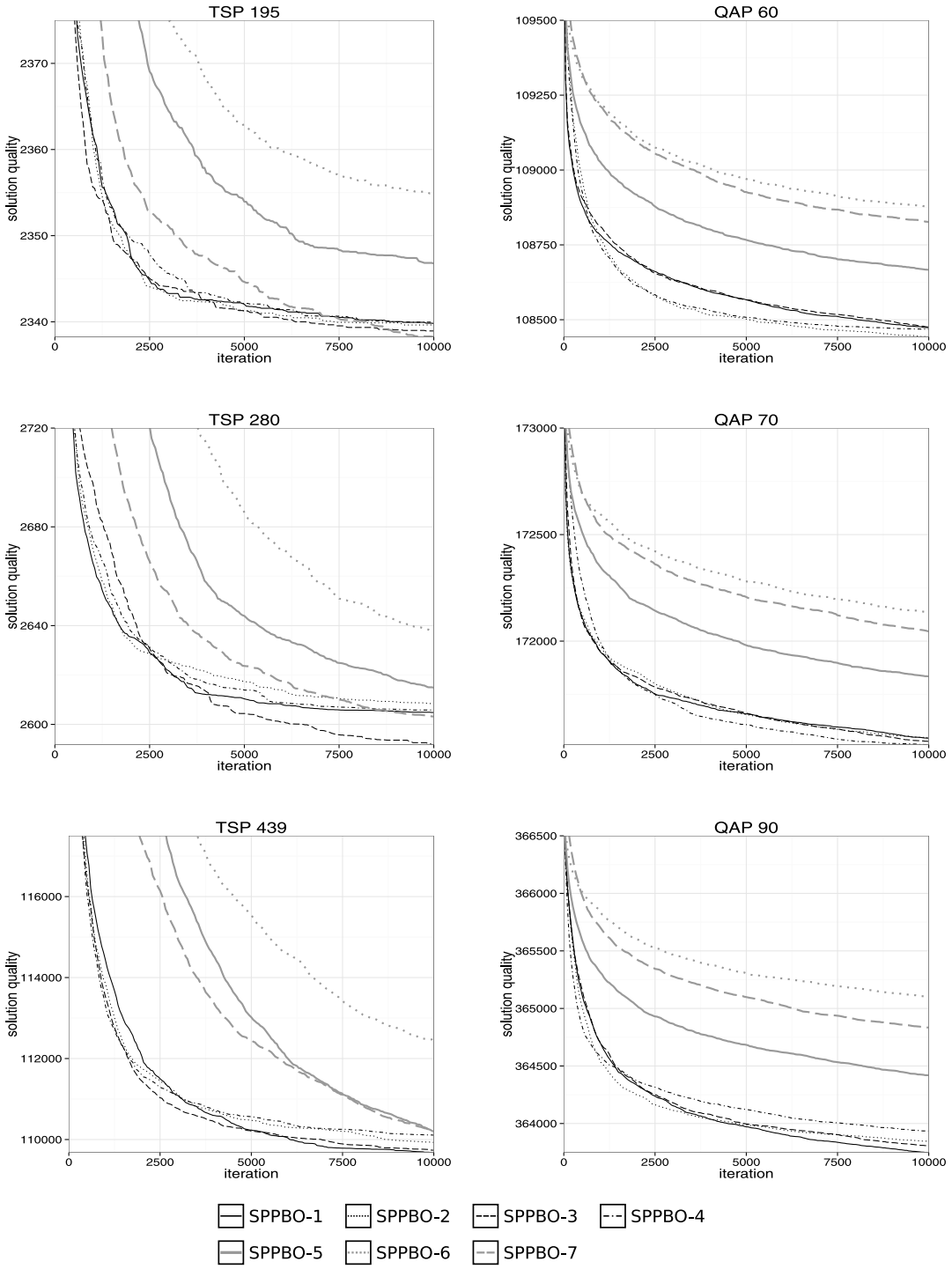


Fig. 2. Average solution quality of the SPPBO algorithms for the TSP (left) and the QAP instances (right); for each algorithm and instance, the best parameter values according to Tables II and III are used.

rat195 and worse than SPPBO-3 for a280. All other differences between algorithms are not significant with respect to the strict threshold of significance that we have used (Wilcoxon-signed rank test with $p$-value $= 0.01/42$ where the correction value 42 equals the number of test cases). For the QAP, it can be seen that algorithms SPPBO-5 to -7 are worse than each of the algorithms SPPBO-1 to -4. Similarly as for the TSP, SPPBO-6 is the worst algorithm for the QAP. However, different as in the case of the TSP, algorithm SPPBO-7 is the second worst algorithm for the QAP. The best algorithms for the QAP are SPPBO-1 and -3 because these are the only algorithms for which there exists not a single case where another algorithm is significantly better.

When considering the parameter values that worked best for the TSP (Table II), it can be seen that there is a tendency for larger values of parameter $w_{\text{total}}$ for the larger problem instances. Similarly, but with the exception of SPPBO-1, there is also a tendency for larger values of $w_{\text{elite}}$ for the larger

TABLE IV
INFLUENCE OF $w_{\text{total}}$ ON SOLUTION QUALITY (AVERAGE) WITH STANDARD DEVIATION (SD) OF SPPBO-1 AT ITERATION 10 000 FOR THE TSP. EACH AVERAGE IS CALCULATED OVER THREE $w_{\text{elite}}$ VALUES AND FOR EACH $w_{\text{elite}}$ VALUE OVER 50 REPEATS. EACH NUMBER IN THE S-MATRIX HEADER CORRESPONDS TO A $w_{\text{total}}$ VALUE. A MARKING "X" IN S-MATRIX DENOTES A SIGNIFICANTLY SMALLER AVERAGE LENGTH OF RESULTS WITH $w_{\text{total}}$ IN THE ROW THAN THOSE WITH $w_{\text{total}}$ IN THE COLUMN (WILCOXON-SIGNED RANK TEST WITH $p$-VALUE = 0.01/72). THE OPTIMAL RESULTS ARE: 2323 FOR RAT195, 2579 FOR A280, AND 107 217 FOR PR439

| Instance | $w_{total}$ | Average±SD | S-Matrix | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0.75 | 1.5 | 3 | 6 | 12 | 24 | 48 | 96 | 192 |
| rat195 | 0.75 | 2354.3±20.9 | - | - | - | X | X | X | X | X | X |
| | 1.5 | 2353.3±18.4 | - | - | - | X | X | X | X | X | X |
| | 3 | 2357.4±19.3 | - | - | - | X | X | X | X | X | X |
| | 6 | 2363.4±17.2 | - | - | - | - | - | X | X | X | X |
| | 12 | 2369.4±19.2 | - | - | - | - | - | X | X | X | X |
| | 24 | 2381.9±24.1 | - | - | - | - | - | - | - | X | X |
| | 48 | 2391.7±23.9 | - | - | - | - | - | - | - | X | X |
| | 96 | 2406.2±24.8 | - | - | - | - | - | - | - | - | X |
| | 192 | 2424.9±29.5 | - | - | - | - | - | - | - | - | - |
| a280 | 0.75 | 2626.6±37.4 | - | - | - | - | - | X | X | X | X |
| | 1.5 | 2628.6±40.3 | - | - | - | - | - | X | X | X | X |
| | 3 | 2629.9±32.9 | - | - | - | - | - | X | X | X | X |
| | 6 | 2627.5±31.2 | - | - | - | - | - | X | X | X | X |
| | 12 | 2637.0±29.5 | - | - | - | - | - | X | X | X | X |
| | 24 | 2650.1±32.5 | - | - | - | - | - | - | - | X | X |
| | 48 | 2661.9±32.4 | - | - | - | - | - | - | - | X | X |
| | 96 | 2680.9±38.5 | - | - | - | - | - | - | - | - | - |
| | 192 | 2692.4±42.4 | - | - | - | - | - | - | - | - | - |
| pr439 | 0.75 | 111528.2±2260.0 | - | - | - | - | - | - | - | - | - |
| | 1.5 | 110950.4±1746.1 | - | - | - | - | - | - | - | - | X |
| | 3 | 110269.1±1473.5 | X | - | - | - | - | - | - | X | X |
| | 6 | 110070.1±1443.1 | X | X | - | - | - | - | X | X | X |
| | 12 | 110148.8±1399.3 | X | X | - | - | - | - | X | X | X |
| | 24 | 110158.5±1363.2 | X | X | - | - | - | - | X | X | X |
| | 48 | 110840.2±1597.2 | - | - | - | - | - | - | - | - | X |
| | 96 | 111510.0±1672.3 | - | - | - | - | - | - | - | - | X |
| | 192 | 112282.4±1674.2 | - | - | - | - | - | - | - | - | - |

problem instances. Thus, the influence of randomness should not be too high for the larger problems. For the QAP, the problem size dependence is not so clear. It is interesting that larger global populations seem to be advantageous for solving the QAP. (Note that this applies only to algorithms SPPBO-1 to -4, since the other algorithms have no global population).

The convergence behavior of all SPPBO algorithms is shown in Fig. 2. For the TSP and the QAP, it can be seen that SPPBO-1 to -4 converge fastest and find good solutions already after 5000 iterations. The other three algorithms converge much slower. It is interesting that SPPBO-7 also finds very good solutions after 10 000 iterations for the TSP, but not for the QAP where it cannot find good solutions.

Overall, the results show that it is profitable for both test problems for an SPPBO algorithm to have a global population with the last $k$ iteration best solutions. Such a global population also leads to a faster convergence. However, for the TSP algorithm SPPBO-7 which does not have a global population is also good. One reason might be that the TSP is easier than the QAP because for the TSP the good solutions are typically similar which is not the case for the QAP (for references where the fitness landscapes of TSP and QAP have been compared (see [14], [22], [39], and [41]). Therefore, for the TSP it is not so important to have a global population which influences the generation of every new solution and therefore can push the algorithm into one direction.

### B. Influence of Parameters

In the following, we investigate the behavior of SPPBO-1 and SPPBO-7 in more detail. The reasons to choose these two algorithms are: 1) SPPBO-1 is one of the best performing algorithms and the results of the other best performing algorithms (SPPBO-2 to -4) are similar and 2) SPPBO-7 is the only algorithm that performs well for the test problems (at least for the TSP) and has no global population with iteration best solutions.

The influence of parameter $w_{\text{total}}$ on the optimization behavior of SPPBO-1 and SPPBO-7 can be seen in Tables IV and V for the TSP and in Tables VI and VII for the QAP. The results show that for SPPBO-1 and the TSP, the optimal values of $w_{\text{total}}$ depend on the size of the problem instance. Small values (between 0.75 and 3) are good for the small instance

TABLE V
INFLUENCE OF $w_{\text{total}}$ ON SOLUTION QUALITY (AVERAGE) WITH STANDARD DEVIATION (SD) OF SPPBO-7 AT ITERATION 10 000 FOR THE TSP. EACH AVERAGE IS CALCULATED OVER THREE $w_{\text{elite}}$ VALUES AND FOR EACH $w_{\text{elite}}$ VALUE OVER 50 REPEATS. EACH NUMBER IN THE S-MATRIX HEADER CORRESPONDS TO A $w_{\text{total}}$ VALUE. A MARKING "X" IN S-MATRIX DENOTES A SIGNIFICANTLY SMALLER AVERAGE LENGTH OF RESULTS WITH $w_{\text{total}}$ IN THE ROW THAN THOSE WITH $w_{\text{total}}$ IN THE COLUMN (WILCOXON-SIGNED RANK TEST WITH $p$-VALUE = 0.01/72). THE OPTIMAL RESULTS ARE: 2323 FOR RAT195, 2579 FOR A280, AND 107 217 FOR PR439

| **Instance** | $w_{total}$ | **Average±SD** | **S-Matrix** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0.75 | 1.5 | 3 | 6 | 12 | 24 | 48 | 96 | 192 |
| rat197 | 0.75 | 2384.0±30.5 | - | - | - | - | - | - | - | - | - |
| | 1.5 | 2365.0±21.4 | X | - | - | - | - | - | - | - | - |
| | 3 | 2347.8±8.7 | X | X | - | - | - | - | - | - | X |
| | 6 | 2341.4±4.9 | X | X | X | - | - | - | - | X | X |
| | 12 | 2339.1±4.9 | X | X | X | X | - | - | X | X | X |
| | 24 | 2340.8±6.0 | X | X | X | - | - | - | - | X | X |
| | 48 | 2342.6±5.8 | X | X | X | - | - | - | - | X | X |
| | 96 | 2348.1±7.1 | X | X | - | - | - | - | - | - | X |
| | 192 | 2354.3±10.0 | X | X | - | - | - | - | - | - | - |
| a280 | 0.75 | 2732.9±102.6 | - | - | - | - | - | - | - | - | - |
| | 1.5 | 2708.3±93.8 | X | - | - | - | - | - | - | - | - |
| | 3 | 2676.1±58.6 | X | X | - | - | - | - | - | - | - |
| | 6 | 2642.7±42.7 | X | X | X | - | - | - | - | - | - |
| | 12 | 2610.5±18.5 | X | X | X | X | - | - | - | - | X |
| | 24 | 2605.2±12.6 | X | X | X | X | - | - | - | X | X |
| | 48 | 2607.0±12.5 | X | X | X | X | - | - | - | - | X |
| | 96 | 2611.0±13.6 | X | X | X | X | - | - | - | - | X |
| | 192 | 2622.4±14.6 | X | X | X | X | - | - | - | - | - |
| pr439 | 0.75 | 118368.5±5194.6 | - | - | - | - | - | - | - | - | - |
| | 1.5 | 117221.0±4612.0 | X | - | - | - | - | - | - | - | - |
| | 3 | 116466.8±3747.7 | X | - | - | - | - | - | - | - | - |
| | 6 | 115123.3±3290.1 | X | X | X | - | - | - | - | - | - |
| | 12 | 113655.5±2544.2 | X | X | X | X | - | - | - | - | - |
| | 24 | 112226.2±1870.3 | X | X | X | X | X | - | - | - | - |
| | 48 | 110946.9±1114.1 | X | X | X | X | X | X | - | - | - |
| | 96 | 110395.8±882.6 | X | X | X | X | X | X | X | - | - |
| | 192 | 110494.4±812.7 | X | X | X | X | X | X | X | - | - |

and medium values (6–24) are good for the large instance. A similar tendency holds for SPPBO-7 and the TSP with the difference that the values of $w_{\text{total}}$ should be larger (12 for the small instance and 96–192 for the large instance). For the QAP instances, the best values of $w_{\text{total}}$ for SPPBO-1 are 24 and 48. For SPPBO-7 and the QAP, the best value of $w_{\text{total}}$ is 768. A potential reason why the values of $w_{\text{total}}$ should be larger for larger problem instances is that otherwise the influence of operation RANDOM becomes too large.

If one wants to choose a single value of $w_{\text{total}}$ for all test instances of both problems, $w_{\text{total}} = 12$ is a good choice for SPPBO-1. For SPPBO-7, there is no single value that works well for all test instances of both problems. However, $w_{\text{total}} = 48$ might be an acceptable choice.

The influence of parameter $w_{\text{elite}}$ on the optimization behavior of SPPBO-1 and SPPBO-7 can be seen in Table VIII for the TSP and in Table IX for the QAP. For both algorithms, the value of $w_{\text{elite}}$ should be large (i.e., 10) for the QAP. This is different for the TSP where values of 0.1 or 1 are good for SPPBO-1 and values of 1 and 10 are good for SPPBO-7. In both cases, larger values are better for larger problem instances. The reason that larger values of $w_{\text{elite}}$ are needed for the QAP might be that the good solutions for the QAP are often very different from each other. Therefore, in order to be able to converge to one subarea of the search space a large value of $w_{\text{elite}}$ seems helpful because it concentrates the search more around the elitist solution.

A combined evaluation of $w_{\text{total}}$ and $w_{\text{elite}}$ is done for SPPBO-1 and SPPBO-7 for the largest instances of both test problems (the results can be found in Appendixes A and B, respectively). It should be noted that a very strong significance threshold has been used for the tests (Wilcoxon-signed rank test with $p$-value = 0.01/702 or $p$-value = 0.01/1260, the large correction values 702 and 1260 are due to the large number of test cases). The results show that in the case of the TSP for SPPBO-1, a large range of medium values (3–24) of $w_{\text{total}}$ are good and the values of $w_{\text{elite}}$ are not so important. In the case of the QAP problem, the range of good values for $w_{\text{total}}$ is narrower, but also medium values (24–48) are good and the values of $w_{\text{elite}}$ are not so important in this range. For SPPBO-7, large values of $w_{\text{total}}$ are good (48–192 for TSP and 768–536 for QAP) and the values of $w_{\text{elite}}$ are not very relevant.

TABLE VI
INFLUENCE OF $w_{\text{total}}$ ON SOLUTION QUALITY (AVERAGE) WITH STANDARD DEVIATION (SD) OF SPPBO-1 AT ITERATION 10 000 FOR THE QAP. EACH AVERAGE IS CALCULATED OVER THREE $w_{\text{elite}}$ VALUES, THREE POPULATION SIZES AND FOR EACH COMBINATION OVER 50 REPEATS. EACH NUMBER IN THE S-MATRIX HEADER CORRESPONDS TO A $w_{\text{total}}$ VALUE. A MARKING "X" IN S-MATRIX DENOTES A SIGNIFICANTLY BETTER SOLUTION QUALITY OF RESULTS WITH $w_{\text{total}}$ IN THE ROW THAN THOSE WITH $w_{\text{total}}$ IN THE COLUMN (WILCOXON-SIGNED RANK TEST WITH $p$-VALUE $= 0.01/72$). THE OPTIMAL RESULTS ARE: 107218 FOR LIPA60A, 169755 FOR LIPA70A, AND 360630 FOR LIPA90A

| Instance | $w_{total}$ | Average±SD | S-Matrix | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0.75 | 1.5 | 3 | 6 | 12 | 24 | 48 | 96 | 192 |
| lipa60a | 0.75 | 109145.1±426.9 | - | - | - | - | - | - | - | - | - |
| | 1.5 | 109131.9±407.8 | X | - | - | - | - | - | - | - | - |
| | 3 | 109070.3±387.7 | X | X | - | - | - | - | - | - | - |
| | 6 | 108886.8±295.9 | X | X | X | - | - | - | - | - | - |
| | 12 | 108572.0±146.5 | X | X | X | X | - | - | - | - | X |
| | 24 | 108485.7±59.0 | X | X | X | X | X | - | X | X | X |
| | 48 | 108501.2±61.8 | X | X | X | X | X | - | - | X | X |
| | 96 | 108548.4±62.8 | X | X | X | X | - | - | - | - | X |
| | 192 | 108619.5±64.4 | X | X | X | X | - | - | - | - | - |
| lipa70a | 0.75 | 172517.6±567.3 | - | - | - | - | - | - | - | - | - |
| | 1.5 | 172492.6±547.9 | X | - | - | - | - | - | - | - | - |
| | 3 | 172429.6±509.8 | X | X | - | - | - | - | - | - | - |
| | 6 | 172236.0±437.2 | X | X | X | - | - | - | - | - | - |
| | 12 | 171811.9±247.3 | X | X | X | X | - | - | - | - | - |
| | 24 | 171571.6±83.6 | X | X | X | X | X | - | - | X | X |
| | 48 | 171576.1±77.0 | X | X | X | X | X | - | - | X | X |
| | 96 | 171639.0±82.4 | X | X | X | X | X | - | - | - | X |
| | 192 | 171730.4±83.1 | X | X | X | X | X | - | - | - | - |
| lipa90a | 0.75 | 365655.2±867.5 | - | - | - | - | - | - | - | - | - |
| | 1.5 | 365626.5±850.8 | X | - | - | - | - | - | - | - | - |
| | 3 | 365537.1±834.9 | X | X | - | - | - | - | - | - | - |
| | 6 | 365300.6±746.4 | X | X | X | - | - | - | - | - | - |
| | 12 | 364710.9±516.0 | X | X | X | X | - | - | - | - | - |
| | 24 | 363943.8±258.3 | X | X | X | X | X | - | - | X | X |
| | 48 | 363924.0±113.0 | X | X | X | X | X | - | - | X | X |
| | 96 | 364010.8±109.9 | X | X | X | X | X | - | - | - | X |
| | 192 | 364150.8±132.9 | X | X | X | X | X | - | - | - | - |

### C. Discussion

The experimental results have shown that, overall, the algorithms SPPBO-1 (PACO) and SPPBO-3 (sSSO-PACO) find the best solutions and are the fastest converging algorithms for TSP and QAP. Algorithm SPPBO-1 has the advantage that it is simpler than SPPBO-3 because it has only global populations.

However, this does not mean that the other algorithms are useless. SPPBO-7 (SSO), e.g., is a good candidate for a distributed algorithm for the TSP. The reason is that SPPBO-7 shows good optimization behavior for the TSP and uses only personal populations (with the exception of the small global population that contains only the elitist solution). Thus, for a distributed implementation, SPPBO-7 can be realized such that the different SCEs run on different processors. Then, it is necessary to exchange information between the processors only when a new elitist solution has been found. A corresponding distributed implementation of SPPBO-1 or SPPBO-3 would require an information exchange at every iteration, since each SCE has to know the iteration best solution.

With respect to the different types of populations, the experimental results indicate the following guidelines for algorithm designers. Global population(s) are particularly advantageous for problems with a complex fitness landscape, i.e., where the good solutions can be very different from each other. Also, the strength of the relative influence of the different populations is particularly important for such problems. The particular choice of the values of parameters $w_{\text{total}}$ and $w_{\text{elite}}$ is more important for algorithms which rely more on private populations. Algorithms that have only global populations seem to be more robust. This indicates the fact that for SPPBO-1, there exists a single value for $w_{\text{total}}$ that is good for all test instances of both test problems which is not the case for SPPBO-7.

Of course, care has to be taken with general statements and more investigations seem to be necessary. It would be interesting to investigate SPPBO algorithms with other types of populations, e.g., SPPBO algorithms where the SCEs are partitioned into groups and each group has a population that influences only the solutions within the group (similar as in the extended SSO). Also, SPPBO algorithms where the influence and/or range of the populations changes during the run of

TABLE VII
INFLUENCE OF $w_{\text{total}}$ ON SOLUTION QUALITY (AVERAGE) WITH STANDARD DEVIATION (SD) OF SPPBO-7 AT ITERATION 10000 FOR THE QAP. EACH AVERAGE IS CALCULATED OVER THREE $w_{\text{elite}}$ VALUES, THREE POPULATION SIZES, AND FOR EACH COMBINATION OVER 50 REPEATS. EACH NUMBER IN THE S-MATRIX HEADER CORRESPONDS TO A $w_{\text{total}}$ VALUE. A MARKING "X" IN S-MATRIX DENOTES A SIGNIFICANTLY BETTER SOLUTION QUALITY OF RESULTS WITH $w_{\text{total}}$ IN THE ROW THAN THOSE WITH $w_{\text{total}}$ IN THE COLUMN (WILCOXON-SIGNED RANK TEST WITH $p$-VALUE $= 0.01/132$). THE OPTIMAL RESULTS ARE: 107218 FOR LIPA60A, 169755 FOR LIPA70A, AND 360630 FOR LIPA90A

| **Instance** | $w_{total}$ | **Average±SD** | **S-Matrix** | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0.75 | 1.5 | 3 | 6 | 12 | 24 | 48 | 96 | 192 | 384 | 768 | 1536 |
| lipa60a | 0.75 | 109295.0±304.3 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 1.5 | 109296.7±303.4 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 3 | 109290.8±298.9 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 6 | 109286.9±294.7 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 12 | 109272.9±275.2 | X | X | X | X | - | - | - | - | - | - | - | - |
| | 24 | 109231.8±245.7 | X | X | X | X | X | - | - | - | - | - | - | - |
| | 48 | 109165.6±194.8 | X | X | X | X | X | X | - | - | - | - | - | - |
| | 96 | 109065.3±131.0 | X | X | X | X | X | X | X | - | - | - | - | - |
| | 192 | 108949.0±73.8 | X | X | X | X | X | X | X | X | - | - | - | - |
| | 384 | 108863.0±53.6 | X | X | X | X | X | X | X | X | X | - | - | - |
| | 768 | 108834.1±47.6 | X | X | X | X | X | X | X | X | X | X | - | X |
| | 1536 | 108862.1±48.3 | X | X | X | X | X | X | X | X | X | - | - | - |
| lipa70a | 0.75 | 172715.4±397.3 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 1.5 | 172712.8±395.2 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 3 | 172709.5±394.0 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 6 | 172702.6±381.6 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 12 | 172681.7±368.8 | X | X | X | X | - | - | - | - | - | - | - | - |
| | 24 | 172638.3±324.9 | X | X | X | X | X | - | - | - | - | - | - | - |
| | 48 | 172556.0±262.7 | X | X | X | X | X | X | - | - | - | - | - | - |
| | 96 | 172438.9±182.5 | X | X | X | X | X | X | X | - | - | - | - | - |
| | 192 | 172283.1±100.3 | X | X | X | X | X | X | X | X | - | - | - | - |
| | 384 | 172129.9±71.4 | X | X | X | X | X | X | X | X | X | - | - | - |
| | 768 | 172062.8±64.1 | X | X | X | X | X | X | X | X | X | X | - | X |
| | 1536 | 172088.1±63.3 | X | X | X | X | X | X | X | X | X | X | - | - |
| lipa90a | 0.75 | 365964.1±598.4 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 1.5 | 365960.5±580.1 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 3 | 365954.6±593.4 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 6 | 365941.7±566.0 | X | - | - | - | - | - | - | - | - | - | - | - |
| | 12 | 365914.1±544.1 | X | X | X | X | - | - | - | - | - | - | - | - |
| | 24 | 365851.5±492.5 | X | X | X | X | X | - | - | - | - | - | - | - |
| | 48 | 365743.5±410.8 | X | X | X | X | X | X | - | - | - | - | - | - |
| | 96 | 365567.0±310.3 | X | X | X | X | X | X | X | - | - | - | - | - |
| | 192 | 365348.1±184.1 | X | X | X | X | X | X | X | X | - | - | - | - |
| | 384 | 365050.8±118.1 | X | X | X | X | X | X | X | X | X | - | - | - |
| | 768 | 364864.8±97.6 | X | X | X | X | X | X | X | X | X | X | - | - |
| | 1536 | 364854.0±93.2 | X | X | X | X | X | X | X | X | X | X | - | - |

an algorithm can be interesting. The proposed generic SPPBO scheme might help to design and classify such algorithms. However, we like to repeat that the introduction of more complex algorithms needs a justification. The field of metaheuristics might also profit from other generic schemes that can be used to identify other classes of algorithms which have some common principles. This is particularly useful, when the algorithms have so far not been identified as related.

## VII. CONCLUSION

It was shown that PACO and SSO can be characterized as simple population-based metaheuristics that are based on two simple operations, namely SELECT+COPY and RANDOM, for creating a new solution. To determine a component of a new solution, SELECT+COPY selects a solution from the population(s) and copies the value of the corresponding component if this is feasible. RANDOM chooses a (feasible) value from the set of possible values randomly according to a given probability distribution. A generic scheme—called SPPBO scheme—was proposed for classifying, designing, and investigating such algorithms systematically. The scheme is not only useful for describing the close relation that exists between PACO and SSO (a fact that has not been mentioned before in the literature) but it also suggested several algorithms that are natural hybrids between

TABLE VIII
INFLUENCE OF $w_{\text{elite}}$ ON SOLUTION QUALITY (AVERAGE) WITH STANDARD DEVIATION (SD) OF SPPBO-1 AND SPPBO-7 AT ITERATION 10 000 FOR THE TSP. EACH AVERAGE IS CALCULATED OVER NINE $w_{\text{total}}$ VALUES AND FOR EACH $w_{\text{total}}$ VALUE OVER 50 REPEATS. A MARKING "X" IN S-MATRIX DENOTES A SIGNIFICANTLY BETTER QUALITY OF THE ROW PARAMETER VALUE COMPARED TO THE COLUMN PARAMETER VALUE (WILCOXON-SIGNED RANK TEST WITH $p$-VALUE $= 0.01/6$)

| Instance | Algorithm | $w_{elite}$ | Average±SD | S-Matrix | | |
|---|---|---|---|---|---|---|
| | | | | 0.1 | 1 | 10 |
| rat195 | SPPBO-1 | 0.1 | 2371.5±34.1 | - | X | X |
| | | 1 | 2374.2±32.3 | - | - | X |
| | | 10 | 2388.4±28.4 | - | - | - |
| | SPPBO-7 | 0.1 | 2354.8±26.5 | - | - | - |
| | | 1 | 2347.0±10.3 | X | - | X |
| | | 10 | 2352.5±17.6 | - | - | - |
| a280 | SPPBO-1 | 0.1 | 2637.9±45.4 | - | - | X |
| | | 1 | 2642.3±39.8 | - | - | X |
| | | 10 | 2664.8±36.7 | - | - | - |
| | SPPBO-7 | 0.1 | 2680.8±101.5 | - | - | - |
| | | 1 | 2625.5±37.9 | X | - | X |
| | | 10 | 2632.4±33.7 | X | - | - |
| pr439 | SPPBO-1 | 0.1 | 111178.1±2006.9 | - | - | - |
| | | 1 | 110747.1±1719.1 | X | - | - |
| | | 10 | 110660.7±1606.3 | X | - | - |
| | SPPBO-7 | 0.1 | 116182.0±5266.4 | - | - | - |
| | | 1 | 113636.1±3402.9 | X | - | - |
| | | 10 | 111814.7±2098.8 | X | X | - |

TABLE IX
INFLUENCE OF $w_{\text{elite}}$ ON SOLUTION QUALITY (AVERAGE) WITH STANDARD DEVIATION (SD) OF SPPBO-1 AND SPPBO-7 AT ITERATION 10 000 FOR THE QAP. EACH AVERAGE IS CALCULATED OVER 9, RESPECTIVELY 11, $w_{\text{total}}$ VALUES, THREE POPULATION SIZES, AND FOR EACH COMBINATION OVER 50 REPEATS. A MARKING "X" IN S-MATRIX DENOTES A SIGNIFICANTLY BETTER QUALITY OF THE ROW PARAMETER VALUE COMPARED TO THE COLUMN PARAMETER VALUE (WILCOXON-SIGNED RANK TEST WITH $p$-VALUE $= 0.01/6$)

| Instance | Algorithm | $w_{elite}$ | Average±SD | S-Matrix | | |
|---|---|---|---|---|---|---|
| | | | | 0.1 | 1 | 10 |
| lipa60a | SPPBO-1 | 0.1 | 108945.7±442.4 | - | - | - |
| | | 1.0 | 108836.6±361.3 | X | - | - |
| | | 10.0 | 108538.0±73.2 | X | X | - |
| | SPPBO-7 | 0.1 | 109248.6±282.3 | - | - | - |
| | | 1.0 | 109226.5±265.0 | X | - | - |
| | | 10.0 | 108878.2±74.5 | X | X | - |
| lipa70a | SPPBO-1 | 0.1 | 172238.6±614.8 | - | - | - |
| | | 1.0 | 172096.7±520.6 | X | - | - |
| | | 10.0 | 171666.2±103.3 | X | X | - |
| | SPPBO-7 | 0.1 | 172649.7±381.5 | - | - | - |
| | | 1.0 | 172621.4±360.9 | X | - | - |
| | | 10.0 | 172158.8±104.4 | X | X | - |
| lipa90a | SPPBO-1 | 0.1 | 365149.8±1072.2 | - | - | - |
| | | 1.0 | 364949.3±946.5 | X | - | - |
| | | 10.0 | 364187.5±237.5 | X | X | - |
| | SPPBO-7 | 0.1 | 365847.0±607.7 | - | - | - |
| | | 1.0 | 365809.0±577.4 | X | - | - |
| | | 10.0 | 365097.7±191.0 | X | X | - |

PACO and SSO. Using the SPPBO scheme, it was shown how heuristic information can be included into SSO and its variants. This led to a systematic experimental investigation of the influence that different types of populations have on the optimization behavior of SPPBO algorithms. Test problems were the TSP and the QAP. The results show that the original PACO and some of its variants which all have a global population performed well for both test problems. The original SSO algorithm which uses only a small global population with the elitist solution performs very well for the TSP. The results indicate that global population(s) are particularly advantageous for SPPBO algorithms that solve

TABLE X

INFLUENCE OF ($w_{\text{total}}$, $w_{\text{elite}}$) COMBINATION ON SOLUTION QUALITY (AVERAGE) WITH STANDARD DEVIATION (SD) OF SPPBO-1 AT ITERATION 10 000 FOR TSP INSTANCE PR439. EACH AVERAGE IS CALCULATED OVER 50 REPEATS. A MARKING "X" IN S-MATRIX DENOTES A SIGNIFICANTLY BETTER QUALITY OF THE ROW PARAMETER VALUE COMPARED TO THE COLUMN PARAMETER VALUE (WILCOXON-SIGNED RANK TEST WITH $p$-VALUE = 0.01/702)

| SPPBO-1 | | Average±SD | S-Matrix | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $w_{total}$ | $w_{elite}$ | | 0.75 | | | 1.5 | | | 3 | | | 6 | | | 12 | | | 24 | | | 48 | | | 96 | | | 192 | | |
| | | | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 |
| 0.75 | 0.1 | 113691.5±1740.9 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 0.75 | 1 | 110785.3±1774.7 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X |
| 0.75 | 10 | 110107.8±1397.6 | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | X | X |
| 1.5 | 0.1 | 111804.1±1691.6 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 1.5 | 1 | 111058.1±1828.2 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 1.5 | 10 | 109988.9±1169.6 | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | X | - | X | X | X |
| 3 | 0.1 | 110454.9±1575.5 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | X | X |
| 3 | 1 | 110080.0±1538.5 | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | X | X |
| 3 | 10 | 110272.5±1296.6 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | X | X |
| 6 | 0.1 | 109709.6±1279.1 | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | X | X | X | X | X |
| 6 | 1 | 110048.8±1350.8 | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | - | X | X | X |
| 6 | 10 | 110451.8±1609.5 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | X |
| 12 | 0.1 | 110069.1±1214.9 | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | - | X | X | X |
| 12 | 1 | 110199.3±1516.5 | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | X | X |
| 12 | 10 | 110178.1±1472.6 | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | - | X | X | X |
| 24 | 0.1 | 110100.0±1537.0 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | X | X |
| 24 | 1 | 110164.5±1259.5 | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | X | X |
| 24 | 10 | 110210.9±1302.1 | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X | X | X |
| 48 | 0.1 | 110987.1±1680.7 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 48 | 1 | 110596.8±1663.8 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X |
| 48 | 10 | 110936.7±1439.3 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | X |
| 96 | 0.1 | 111620.8±1849.1 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 96 | 1 | 111600.9±1671.5 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 96 | 10 | 111308.4±1492.7 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 192 | 0.1 | 112166.0±1886.3 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 192 | 1 | 112190.2±1533.4 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 192 | 10 | 112491.1±1595.8 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |

TABLE XI

INFLUENCE OF ($w_{\text{total}}$, $w_{\text{elite}}$) COMBINATION ON SOLUTION QUALITY (AVERAGE) WITH STANDARD DEVIATION (SD) OF SPPBO-7 AT ITERATION 10 000 FOR TSP INSTANCE PR439. EACH AVERAGE IS CALCULATED OVER 50 REPEATS. A MARKING "X" IN S-MATRIX DENOTES A SIGNIFICANTLY BETTER QUALITY OF THE ROW PARAMETER VALUE COMPARED TO THE COLUMN PARAMETER VALUE (WILCOXON-SIGNED RANK TEST WITH $p$-VALUE = 0.01/702)

| SPPBO-7 | | Average±SD | S-Matrix | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $w_{total}$ | $w_{elite}$ | | 0.75 | | | 1.5 | | | 3 | | | 6 | | | 12 | | | 24 | | | 48 | | | 96 | | | 192 | | |
| | | | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 |
| 0.75 | 0.1 | 124191.1±2132.3 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 0.75 | 1 | 117362.4±3305.8 | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 0.75 | 10 | 113552.1±2697.9 | X | X | - | X | X | - | X | - | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 1.5 | 0.1 | 122348.1±1908.3 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 1.5 | 1 | 116555.9±2883.1 | X | - | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 1.5 | 10 | 112758.8±2265.4 | X | X | - | X | X | - | X | X | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 3 | 0.1 | 120301.3±2051.8 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 3 | 1 | 115991.6±2556.8 | X | - | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 3 | 10 | 113107.4±2282.8 | X | X | - | X | X | - | X | X | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 6 | 0.1 | 118029.4±1865.3 | X | - | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 6 | 1 | 115068.7±2863.9 | X | - | - | X | - | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 6 | 10 | 112271.7±2077.7 | X | X | - | X | X | - | X | X | - | X | X | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 12 | 0.1 | 115912.3±1769.2 | X | - | - | X | - | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 12 | 1 | 113529.9±2097.6 | X | X | - | X | X | - | X | X | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 12 | 10 | 111524.2±1518.3 | X | X | - | X | X | - | X | X | X | X | X | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 24 | 0.1 | 112725.4±1640.3 | X | X | - | X | X | - | X | X | - | X | X | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 24 | 1 | 112303.3±2181.5 | X | X | - | X | X | - | X | X | - | X | X | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 24 | 10 | 111650.0±1608.8 | X | X | X | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 48 | 0.1 | 111168.0±1266.0 | X | X | X | X | X | - | X | X | X | X | X | - | X | X | - | X | - | - | - | - | - | - | - | - | - | - | - |
| 48 | 1 | 110937.7±1099.9 | X | X | X | X | X | - | X | X | X | X | X | - | X | X | - | X | - | - | - | - | - | - | - | - | - | - | - |
| 48 | 10 | 110735.1±930.6 | X | X | X | X | X | X | X | X | X | X | X | - | X | X | - | X | - | - | - | - | - | - | - | - | - | - | - |
| 96 | 0.1 | 110476.0±894.4 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - |
| 96 | 1 | 110480.4±912.5 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - |
| 96 | 10 | 110230.9±833.5 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - | - | - | - |
| 192 | 0.1 | 110486.5±866.9 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - |
| 192 | 1 | 110494.6±902.3 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - |
| 192 | 10 | 110502.0±665.3 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - |

optimization problems with a complex fitness landscape. Global populations also seem to make SPPBO algorithms more robust with respect to the relative influence of the different populations. However, more investigations are necessary to confirm or refute these observations for other test problems.

TABLE XII

INFLUENCE OF ($w_{\text{total}}$, $w_{\text{elite}}$) COMBINATION ON SOLUTION QUALITY (AVERAGE) WITH STANDARD DEVIATION (SD) OF SPPBO-1 AT ITERATION 10 000 FOR QAP INSTANCE LIPA90A. EACH AVERAGE IS CALCULATED OVER THREE POPULATION SIZES AND FOR EACH POPULATION SIZES OVER 50 REPEATS. A MARKING "X" IN S-MATRIX DENOTES A SIGNIFICANTLY BETTER QUALITY OF THE ROW PARAMETER VALUE COMPARED TO THE COLUMN PARAMETER VALUE (WILCOXON-SIGNED RANK TEST WITH $p$-VALUE = 0.01/702)

| SPPBO-1 | | Average±SD | S-Matrix | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $w_{total}$ | $w_{elite}$ | | 0.75 | | | 1.5 | | | 3 | | | 6 | | | 12 | | | 24 | | | 48 | | | 96 | | | 192 | | | | |
| | | | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | | |
| 0.75 | 0.1 | 366387.7±70.1 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 0.75 | 1.0 | 366130.2±80.6 | X | - | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 0.75 | 10.0 | 364447.7±145.4 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 1.50 | 0.1 | 366354.5±70.4 | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 1.50 | 1.0 | 366080.1±82.0 | X | X | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 1.50 | 10.0 | 364445.0±152.6 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 3.00 | 0.1 | 366271.2±74.7 | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 3.00 | 1.0 | 365958.3±88.4 | X | X | - | X | X | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 3.00 | 10.0 | 364381.8±149.8 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 6.00 | 0.1 | 366011.3±84.1 | X | X | - | X | X | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 6.00 | 1.0 | 365604.9±110.6 | X | X | - | X | X | - | X | X | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 6.00 | 10.0 | 364285.5±142.8 | X | X | X | X | X | X | X | X | X | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 12.00 | 0.1 | 365225.3±230.8 | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 12.00 | 1.0 | 364814.1±270.7 | X | X | - | X | X | - | X | X | - | X | X | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 12.00 | 10.0 | 364093.5±121.6 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 24.00 | 0.1 | 364035.9±369.4 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 24.00 | 1.0 | 363860.9±185.7 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | X | - | X | X | X | X | X | X | | |
| 24.00 | 10.0 | 363934.6±120.8 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - | X | X | X | X | X | X | | |
| 48.00 | 0.1 | 363907.6±114.8 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - | X | X | X | X | X | X | | |
| 48.00 | 1.0 | 363937.9±111.3 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - | X | X | X | X | X | X | | |
| 48.00 | 10.0 | 363926.5±111.5 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - | X | X | X | X | X | X | | |
| 96.00 | 0.1 | 364003.8±110.8 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - | - | - | - | X | X | X | | |
| 96.00 | 1.0 | 364004.4±108.2 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - | - | - | - | X | X | X | | |
| 96.00 | 10.0 | 364024.1±110.2 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - | - | - | - | X | X | X | | |
| 192.00 | 0.1 | 364150.8±134.4 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - | | |
| 192.00 | 1.0 | 364153.0±136.5 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - | | |
| 192.00 | 10.0 | 364148.7±128.6 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - | | |

TABLE XIII

INFLUENCE OF ($w_{\text{total}}$, $w_{\text{elite}}$) COMBINATION ON SOLUTION QUALITY (AVERAGE) WITH STANDARD DEVIATION (SD) OF SPPBO-7 AT ITERATION 10 000 FOR QAP INSTANCE LIPA90A. EACH AVERAGE IS CALCULATED OVER THREE POPULATION SIZES AND FOR EACH POPULATION SIZES OVER 50 REPEATS. A MARKING "X" IN S-MATRIX DENOTES A SIGNIFICANTLY BETTER QUALITY OF THE ROW PARAMETER VALUE COMPARED TO THE COLUMN PARAMETER VALUE (WILCOXON-SIGNED RANK TEST WITH $p$-VALUE = 0.01/1260)

| SPPBO-7 | | Average±SD | S-Matrix | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $w_{total}$ | $w_{elite}$ | | 0.75 | | | 1.5 | | | 3 | | | 6 | | | 12 | | | 24 | | | 48 | | | 96 | | | 192 | | | 384 | | | 768, 1536 | | |
| | | | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 | 0.1 | 1.0 | 10 |
| 0.75 | 0.1 | 366424.6±61.8 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 0.75 | 1.0 | 366332.4±79.4 | X | - | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 0.75 | 10.0 | 365135.3±165.0 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - |
| 1.50 | 0.1 | 366402.2±67.8 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 1.50 | 1.0 | 366325.3±74.8 | X | - | - | X | - | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 1.50 | 10.0 | 365153.9±136.7 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - |
| 3.00 | 0.1 | 366406.6±61.0 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 3.00 | 1.0 | 366328.1±68.1 | X | - | - | X | - | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 3.00 | 10.0 | 365129.1±142.8 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - |
| 6.00 | 0.1 | 366368.3±75.4 | X | - | - | - | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 6.00 | 1.0 | 366300.6±70.0 | X | - | - | X | - | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 6.00 | 10.0 | 365156.2±144.5 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - |
| 12.00 | 0.1 | 366326.8±60.0 | X | - | - | X | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 12.00 | 1.0 | 366255.2±84.6 | X | X | - | X | X | - | X | X | - | X | X | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 12.00 | 10.0 | 365160.3±143.8 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - |
| 24.00 | 0.1 | 366211.5±62.0 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 24.00 | 1.0 | 366172.2±79.8 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 24.00 | 10.0 | 365170.9±143.1 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - |
| 48.00 | 0.1 | 366031.4±72.9 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 48.00 | 1.0 | 366014.4±75.1 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 48.00 | 10.0 | 365184.7±160.8 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - |
| 96.00 | 0.1 | 365770.7±76.1 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 96.00 | 1.0 | 365759.0±91.0 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 96.00 | 10.0 | 365171.4±199.1 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - |
| 192.00 | 0.1 | 365432.5±91.8 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - |
| 192.00 | 1.0 | 365428.2±96.4 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - |
| 192.00 | 10.0 | 365183.6±208.7 | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | X | X | - | - | - | - | - | - | - |
| 384.00 | 0.1 | 365073.1±100.9 | X | X | - | X | X | X | X | X | - | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - |
| 384.00 | 1.0 | 365065.5±99.3 | X | X | - | X | X | X | X | X | - | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - |
| 384.00 | 10.0 | 365013.8±141.2 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - | - | - | - |
| 768.00 | 0.1 | 364869.3±89.5 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - |
| 768.00 | 1.0 | 364868.9±102.1 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - |
| 768.00 | 10.0 | 364856.3±100.8 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - |
| 1536.00 | 0.1 | 364846.5±94.1 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - |
| 1536.00 | 1.0 | 364858.3±85.1 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - |
| 1536.00 | 10.0 | 364857.3±99.9 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | - | - | - |

## APPENDIX A
## TABLES SHOWING INFLUENCE OF $w_{\text{total}}$ ON SPPBO-1 AND SPPBO-7

See Tables IV–VII.

## APPENDIX B
## TABLES SHOWING INFLUENCE OF $w_{\text{elite}}$ ON SPPBO-1 AND SPPBO-7

See Tables VIII and IX.

## APPENDIX C
## TABLES SHOWING INFLUENCE OF ($w_{\text{total}}$, $w_{\text{elite}}$) ON SPPBO-1 AND SPPBO-7

See Tables X–XIII.

## REFERENCES


[1] D. Angus, "Population-based ant colony optimisation for multi-objective function optimisation," in *Proc. 3rd Aust. Conf. Progr. Artif. Life (ACAL)*, Gold Coast, QLD, Australia, 2007, pp. 232–244.

[2] D. Angus, "Crowding population-based ant colony optimisation for the multi-objective travelling salesman problem," in *Proc. IEEE Symp. Comput. Intell. Multi Criteria Decis. Mak. (MCDM)*, Honolulu, HI, USA, 2007, pp. 333–340.

[3] D. Angus, "Niching for population-based ant colony optimization," in *Proc. 2nd IEEE Int. Conf. e-Sci. Grid Comput.*, Amsterdam, The Netherlands, 2006, p. 115.

[4] C. Bae, W.-C. Yeh, N. Wahid, Y. Y. Chung, and Y. Liu, "A new simplified swarm optimization (SSO) using exchange local search scheme," *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 6, pp. 4391–4406, 2012.

[5] J. Bautista, E. Fernández, and J. Pereira, "Solving an urban waste collection problem using ants heuristics," *Comput. Oper. Res.*, vol. 35, no. 9, pp. 3020–3033, 2008.

[6] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Inf. Sci.*, vol. 237, pp. 82–117, Jul. 2013.

[7] L. T. Bui, J. M. Whitacre, and H. A. Abbass, "Performance analysis of elitism in multi-objective ant colony optimization algorithms," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Hong Kong, 2008, pp. 1633–1640.

[8] E. Burke and G. Kendall, Eds., *Search Methodologies—Introductory Tutorials in Optimisation and Decision Support Methodology*, 2nd ed. New York, NY, USA: Springer, 2014.

[9] S. Consoli, J. A. Moreno Pérez, K. Darby-Dowman, and N. Mladenović, "Discrete particle swarm optimization for the minimum labelling Steiner tree problem," in *Nat. Comput.*, vol. 9, no. 1, pp. 29–46, Mar. 2010.

[10] K. Diwold, T. Ruhnke, and M. Middendorf, "Sensor placement in water networks using a population-based ant colony optimization algorithm," in *Proc. 2nd Int. Conf. Comput. Collect. Intell. Technol. Appl. (ICCCI)*, vol. 6423. Kaohsiung, Taiwan, 2010, pp. 426–437.

[11] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[12] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: An autocatalytic optimizing process," Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milan, Italy, Tech. Rep. 91-016, 1991.

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

[14] D. Garrett and D. Dasgupta, "Multiobjective landscape analysis and the generalized assignment problem," in *Proc. 2nd Learn. Intell. Optim. Workshop (LION-II)*, vol. 5313. Trento, Italy, 2008, pp. 110–124.

[15] M. Gendreau and J.-Y. Potvin, Eds., *Handbook of Metaheuristics* (International Series in Operations Research & Management Science), vol. 146. New York, NY, USA: Springer, 2010.

[16] M. Gómez-González, A. López, and F. Jurado, "Hybrid discrete PSO and OPF approach for optimization of biomass fueled micro-scale energy system," *Energy Convers. Manage.*, vol. 65, pp. 539–545, Jan. 2013.

[17] M. Guntsch and M. Middendorf, "A population based approach for ACO," in *Proc. 2nd Eur. Workshop Evol. Comput. Comb. Optim. (EvoCOP)*, vol. 2279. Kinsale, Ireland, 2002, pp. 72–81.

[18] M. Guntsch and M. Middendorf, "Applying population based ACO to dynamic optimization problems," in *Proc. 3rd Int. Workshop Ant Algorithms (ANTS)*, vol. 2463. Brussels, Belgium, 2002, pp. 111–122.

[19] M. Guntsch and M. Middendorf, "Solving multi-objective permutation problems with population based ACO," in *Proc. 2nd Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, vol. 2636. Faro, Portugal, 2003, pp. 464–478.

[20] M. Guntsch, "Ant algorithms in stochastic and multi-criteria environments," Ph.D. dissertation, Inst. AIFB, Univ. Karlsruhe, Karlsruhe, Germany, 2004.

[21] W. J. Gutjahr, "Stochastic search in metaheuristics," in *Handbook of Metaheuristics* (International Series in Operations Research & Management Science), vol. 146. New York, NY, USA: Springer, 2010, pp. 573–597.

[22] H. H. Hoos and T. Stützle, *Stochastic Local Search—Foundations and Applications*. Burlington, MA, USA: Elsevier, 2004.

[23] T. B. Kurniawan, N. K. Khalid, Z. Ibrahim, M. Khalid, and M. Middendorf, "Evaluation of ordering methods for DNA sequence design based on ant colony system," in *Proc. Asia Int. Conf. Model. Simulat.*, Kuala Lumpur, Malaysia, 2008, pp. 905–910.

[24] Y.-C. Lin and M. Middendorf, "Simple probabilistic population based optimization for combinatorial optimization," in *Proc. IEEE Symp. Swarm Intell.*, Singapore, 2013, pp. 213–220.

[25] Y. Liu, Y. Y. Chung, and W.-C. Yeh, "Simplified swarm optimization with sorted local search for golf data classification," in *Proc. IEEE Congr. Evol. Comput.*, Brisbane, QLD, Australia, 2012, pp. 1–8.

[26] F. J. Martínez-García and J. A. Moreno-Pérez, "Jumping frogs optimization: A new swarm method for discrete optimization," Dept. Stat. O.R. Comput., Univ. La Laguna, Tenerife, Spain, Tech. Rep. DEIOC 3/2008, 2008.

[27] M. Mavrovouniotis and S. Yang, "Ant colony optimization with memory-based immigrants for the dynamic vehicle routing problem," in *Proc. IEEE Congr. Evol. Comput.*, Brisbane, QLD, Australia, 2012, pp. 2645–2652.

[28] R. L. V. Moritz, E. Reich, M. Bernt, and M. Middendorf, "The influence of correlated objectives on different types of P-ACO algorithms," in *Proc. 14th Eur. Conf. Evol. Comput. Comb. Optim. (EvoCOP)*, vol. 8600. Granada, Spain, 2014, pp. 230–241.

[29] I. Moser and J. Montgomery, "Population-ACO for the automotive deployment problem," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Dublin, Ireland, 2011, pp. 777–784.

[30] S. Oliveira, M. S. Hussin, T. Stützle, A. Roli, and M. Dorigo, "A detailed analysis of the population-based ant colony optimization algorithm for the TSP and the QAP," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Dublin, Ireland, 2011, pp. 13–14.

[31] O. Sammoud, C. Solnon, and K. Ghedira, "A similarity based P-ACO version for solving dynamic problems," in *Proc. Int. Conf. Metaheuristics Nat. Inspired Comput. (META)*, Hammamet, Tunisie, 2008.

[32] O. Sammoud, C. Solnon, and K. Ghedira, "A new ACO approach for solving dynamic problems," in *Proc. 9th Int. Conf. Artif. Evol.*, Strasbourg, France, 2009.

[33] B. Scheuermann *et al.*, "FPGA implementation of population-based ant colony optimization," *Appl. Soft Comput.*, vol. 4, pp. 303–322, Aug. 2004.

[34] M. A. M. Shukran, "Image classification technique using modified particle swarm optimization," *Mod. Appl. Sci.*, vol. 5, no. 5, pp. 150–164, 2011.

[35] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *Eur. J. Oper. Res.*, vol. 185, pp. 1155–1173, Mar. 2008.

[36] K. Sörensen, "Metaheuristics—The metaphor exposed," *Int. Trans. Oper. Res.*, vol. 22, pp. 3–18, Jan. 2015.

[37] T. Stützle and M. Dorigo, "ACO algorithms for the quadratic assignment problem," in *New Ideas in Optimization*, D. Corne, F. Glover, and M. Dorigo, Eds. Maidenhead, U.K.: McGraw-Hill, pp. 33–50, 1999.

[38] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future Gener. Comput. Syst.*, vol. 16, no. 9, pp. 889–914, 2000.

[39] E.-G. Talbi, *Metaheuristics: From Design to Implementation*. Hoboken, NJ, USA: Wiley, 2009.

[40] T. Thalheim, D. Merkle, and M. Middendorf, "Protein folding in the HP-model solved with a hybrid population based ACO algorithm," *IAENG Int. J. Comput. Sci.*, vol. 35, pp. 291–300, Sep. 2008.

[41] S. Volke, D. Zeckzer, G. Scheuermann, and M. Middendorf, "Visual analysis and comparison of search landscapes of TSP, QAP, and SMTTP," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Madrid, Spain, 2015, pp. 497–504.

[42] W.-C. Yeh, "A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems," *Expert Syst. Appl.*, vol. 36, no. 5, pp. 9192–9200, Jul. 2009.

[43] W.-C. Yeh, "Simplified swarm optimization in disassembly sequencing problems with learning effects," *Comput. Oper. Res.*, vol. 39, no. 9, pp. 2168–2177, 2012.

[44] W.-C. Yeh, "Novel swarm optimization for mining classification rules on thyroid gland data," *Inf. Sci.*, vol. 197, pp. 65–76, Aug. 2012.

[45] W.-C. Yeh, "Optimization of the disassembly sequencing problem on the basis of self-adaptive simplified swarm optimization," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 1, pp. 250–261, Jan. 2012.

[46] W.-C. Yeh, C.-M. Lin, and S.-C. Wei, "Disassembly sequencing problems with stochastic processing time using simplified swarm optimization," *Int. J. Innovat. Manage. Technol.*, vol. 3, no. 3, pp. 226–231, 2012.

[47] W.-C. Yeh, "Orthogonal simplified swarm optimization for the series parallel redundancy allocation problem with a mix of components," *Knowl.-Based Syst.*, vol. 64, pp. 1–12, Jul. 2014.
[48] W.-C. Yeh and C.-L. Huang, "Simplified swarm optimization to solve the K-harmonic means problem for mining data," in *Proc. 18th Asia Pac. Symp. Intell. Evol. Syst.*, vol. 2. 2015, pp. 429–439.
[49] W.-C. Yeh and C.-L. Huang, "Self-adaptive orthogonal simplified swarm optimization for the series-parallel redundancy allocation problem," in *Proc. 18th Asia Pac. Symp. Intell. Evol. Syst.*, vol. 2. 2015, pp. 191–200.
[50] S.-C. Wei, W.-C. Yeh, and T.-J. Yen, "Pareto simplified swarm optimization for grid-computing reliability and service makspan in grid-RMS," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, 2014, pp. 1593–1600.
[51] T. Weise *et al.*, "Benchmarking optimization algorithms: An open source framework for the traveling salesman problem," *IEEE Comput. Intell. Mag.*, vol. 9, no. 3, pp. 40–52, Aug. 2014.

**Ying-Chi Lin** received the bachelor's and master's degrees in entomology from National Chung-Hsin University, Taichung, Taiwan; a second bachelor's degree in computer science from University of Leipzig, Leipzig, Germany; and the Ph.D. degree in environmental sciences from University of East Anglia, Norwich, U.K. She is currently working toward the master's degree in computer science with University of Leipzig.

She was a Consultant with a software company.

**Martin Clauß** received the Diploma degree in computer science from University of Leipzig, Leipzig, Germany.

He is a Researcher with the Research Group of Prof. Dr. M. Middendorf, University of Leipzig. His research interests include ant colony optimization, combinatorial optimization, permutation problems, and graphics processing unit computing.

**Martin Middendorf** received the Diploma degree in mathematics and the Dr. rer. nat. degree in computer science from University of Hannover, Hannover, Germany, in 1988 and 1992, respectively, and the Habilitation degree from University of Karlsruhe, Karlsruhe, Germany, in 1998.

He was with University of Dortmund, Dortmund, Germany, and a Visiting Professor of Computer Science with University of Hannover. He was a Professor of Computer Science with Catholic University of Eichstätt, Eichstätt, Germany. He is currently a Professor of Parallel Computing and Complex Systems, University of Leipzig, Leipzig, Germany. His research interests include nature inspired algorithms, reconfigurable architectures, parallel algorithms, swarm intelligence, and bioinformatics.