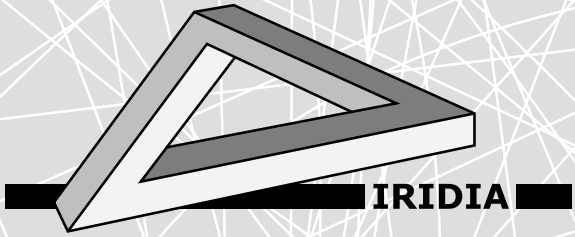

Université Libre de Bruxelles

Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle

Stochastic Local Search Algorithms for Multiobjective Combinatorial Optimization: A Review

Luís Paquete and Thomas Stützle

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2006-001

January 2006

IRIDIA – Technical Report Series
ISSN 1781-3794

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

Technical report number TR/IRIDIA/2006-001

Stochastic Local Search Algorithms for Multiobjective Combinatorial Optimization: A Review

Luís Paquete
Faculdade de Economia &
Centro de Sistemas Inteligentes
Universidade do Algarve
Campus de Gambelas
8005 Faro, Portugal
E-mail: lpaquete@ualg.pt

Thomas Stützle
IRIDIA, CP194/6
Université Libre de Bruxelles
Avenue Franklin Roosevelt 50
1050 Brussels, Belgium
E-mail: stuetzle@iridia.ulb.ac.be

1 Introduction

Multiobjective Combinatorial Optimization Problems (MCOPs) are combinatorial problems that involve the optimization of several, typically conflicting objectives. An MCOP arises, for example, when planning a trip from a city A to some city B in a holiday trip. Besides the minimization of the overall distance between the two sites, one may also be interested in minimizing the cost, the overall travel time, etc. An unequivocal solution to such problems is the one which is *optimal* with respect to all objectives. But is there such a solution? The shortest tour is not necessarily the fastest nor the fastest needs to be the cheapest one–just consider tolled highways.

Which is then the optimal solution to such a multiobjective problem? It depends on the notion of *optimality*. In this paper, we focus on the notion of Pareto optimality, which arises when the decision maker is not able to express his preferences *a priori*, simply because he is not present in the process or not able to give an *a priori* formula or ranking of the objectives. In this case, one is interested in obtaining a set of solutions that represents the *optimal trade-off* between the objectives, *i.e.*, solutions which are not worse than any other and strictly better in at least one of the objectives.

The set of available algorithms for computing high quality approximations to the Pareto optimal set has grown enormously over the recent years as witnessed by a large number of papers at international conferences and workshops [15, 26, 30, 66], special issues of scientific journals [14, 27, 39] and numerous regular papers at other conferences about algorithms or on topics of multiple criteria decision making. The majority of these approaches are based on stochastic local search (SLS) algorithms (see chapter R-18 about more details on SLS), a trend that reflects the enormous success of these algorithms for single-objective problems.

Here, we review the main developments in the application of SLS algorithms to MCOPs. SLS techniques range from simple constructive algorithms and iterative improvement algorithms to general algorithm frameworks that can be adapted to a specific problem under concern. These latter general-purpose SLS methods (also often called metaheuristics) include simulated annealing, tabu search, evolutionary algorithms, ant colony optimization and many others. While these techniques can result in rather complex algorithms already for single-objective problems, when applied to MCOPs they become even more complicated because they need to return a set of solutions instead of a single one.

For tackling MCOPs with SLS algorithms, two fundamentally different approaches can be distinguished. The first is to base the search on the component-wise ordering of the objective value vectors of solutions (or some *ranking* derived from these orderings). We will say that SLS algorithms that mainly focus on this approach follow the *component-wise acceptance criterion search model* (CWAC search model). The second approach is based on the usage of parameterized scalarization methods by aggregating the objectives; the SLS algorithms following such lines use the *scalarized acceptance criterion search model* (SAC search model). These two choices define somehow the two main *schools* for the design of SLS algorithms for MCOPs. Different choices for the remaining components of an SLS algorithm for MCOPs crucially depend on the choice taken for the search model.

In the following sections, we review available SLS algorithms in dependence of these two choices, which also makes this review different from several earlier ones [12, 13, 22, 25, 45, 40]. In addition, some proposals combine these two search models. We will review these latter *hybrid* approaches separately from the others. Despite the view taken here, it is inevitable to further discuss the proposed algorithms in dependence of the analogy to known SLS methods for the single objective problems. However, since we are more interested in the differences between the main search strategies, we do not consider specific problem-dependent implementation choices in detail and also try to avoid the highly specialized jargon found in the context of various SLS methods.

2 Basics

The main goal of solving MCOPs in terms of Pareto optimality is to find solutions which are not worse than any other solution and strictly better in at least one of the objectives. Let Q be the number of objectives and $\mathcal{S}$ be the set of all candidate solutions; then the objective function for a solution $s \in \mathcal{S}$ to MCOPs can be defined as a mapping $\vec{f} : s \mapsto \mathbb{R}^Q$. The following orders hold for objective function vectors in $\mathbb{R}^Q$. Let $\vec{u}$ and $\vec{v}$ be vectors in $\mathbb{R}^Q$; we define the (i) *weak component-wise order* as $\vec{u} \leq \vec{v}$, *i.e.*, $u_i \leq v_i,\ i = 1, \ldots, Q$; and (ii) the *component-wise order* as $\vec{u} \prec \vec{v}$, *i.e.*, $\vec{u} \neq \vec{v}$ and $u_i \leq v_i,\ i = 1, \ldots, Q$. In the context of optimization, we denote the relation between objective function value vectors of two feasible solutions s and s' as follows: (i) if $\vec{f}(s) \prec \vec{f}(s')$, we say that $\vec{f}(s)$ *dominates* $\vec{f}(s')$; and (ii) if $\vec{f}(s) \leq \vec{f}(s')$, then $\vec{f}(s)$ *weakly dominates* $\vec{f}(s')$. In addition, we say that $\vec{f}(s)$ and $\vec{f}(s')$ are *non-dominated* if $\vec{f}(s) \not\prec \vec{f}(s')$ and $\vec{f}(s') \not\prec \vec{f}(s)$, and they are *non-weakly dominated* if $\vec{f}(s) \not\leq \vec{f}(s')$ and $\vec{f}(s') \not\leq \vec{f}(s)$. Note that the latter implies that $\vec{f}(s) \neq \vec{f}(s')$. For simplification purposes, we shall use the same notation among solutions when the above relations hold between their objective function value vectors.

Since the notion of optimal solution clearly differs from the single-objective counterpart, we need to define the notion of a *Pareto global optimum solution* and a *Pareto global optimum set*: A solution $s \in \mathcal{S}$ is a Pareto global optimum if and only if there is no $s' \in \mathcal{S}$ such that $\vec{f}(s') \prec \vec{f}(s)$; we say that $\mathcal{S}' \subseteq \mathcal{S}$ is a Pareto global optimum set if and only if it contains *only* and *all* Pareto global optimum solutions. We call the image of the Pareto global optimum set in the objective space *the efficient set*. In most cases, solving an MCOP in terms of Pareto optimality would correspond to finding solutions that are representative of the efficient set.

When solving an MCOP in terms of scalarized optimality, it is assumed that the decision maker is able to *weigh* the importance of each objective; the objective function vector is then *scalarized* according to some weight vector $\vec{\lambda} = (\lambda_1, \ldots, \lambda_Q)$. We will denote the scalarized objective function value by $f_\lambda(s)$. We then say that a solution $s \in \mathcal{S}$ is a scalarized global optimum solution if and only if there is no $s' \in \mathcal{S}$ such that $f_\lambda(s) < f_\lambda(s')$ with respect to a given $\vec{\lambda}$. The weight vector $\vec{\lambda}$ is usually normalized such that $\sum_{q=1}^{Q} \lambda_q = 1$. Thus, $\vec{\lambda}$ is an element from the set of normalized weight vectors Λ given by

$$\Lambda = \{\vec{\lambda} \in R^Q : \lambda_q > 0, \sum_{q=1}^{Q} \lambda_q = 1, q = 1, \ldots, Q\} \tag{1.1}$$

In the case of different ranges of values between objectives, a normalization by range equalization factors must be considered [63]. The scalarization of the objective function vector is usually based on the family of weighted L_p-metrics as

$$f_\lambda(s) = \Big[\sum_{i=1}^{Q} \big(\lambda_i |f_i(s) - y_i|\big)^p \Big]^{1/p}, \tag{1.2}$$

where $s \in \mathcal{S}$, $p > 0$, and $\vec{y} = y_1, \ldots, y_Q$ is the *ideal* vector, where we have $y_i = \min f_i,\ i = 1, \ldots, Q$. Settings of $p = 1$ or $p = \infty$ are most often used. When $p = 1$, we have the well-known *weighted sum* formulation given by

$$f_\lambda(s) = \sum_{i=1}^{Q} \lambda_i f_i(s). \tag{1.3}$$

It is well known that a scalarized global optimum solution for Equation (1.2) with $p \neq \infty$ is also a Pareto global optimum solution, either if it is a unique solution or if the components of the weight vectors are all

positive [63].

Obviously, the great advantage of using a scalarized objective function is that the same SLS algorithm for solving the single-objective problem can be used for tackling the multiobjective version. When using such scalarized objective functions, the available algorithms typically change the weight vectors to generate solutions that are of high quality for different weights; A disadvantage is that, when finding optimal solutions with respect to the scalarized objective function, only *supported solutions*, that is, solutions on the convex hull of the efficient set, are obtained.

Finally, let us mention that (with few exceptions) common to all SLS algorithms for MCOPs is that they return a set of non-dominated solutions. Therefore, most of these algorithms include an additional data structure that maintains a set of solutions during the search process that we call *archive* and that is returned when the algorithm is terminated at an arbitrarily chosen time. In our review, we consider that the best non-dominated solutions found during the algorithm's run are maintained in the archive, if not expressed otherwise. During the search process, the algorithm needs to *update* the archive and, if nothing else is said, we assume that this update consists of (i) adding new non-dominated solutions, and (ii) removing dominated ones.

Many of the available SLS algorithms make use of two further techniques. The first is *archive bounding*; it is used because the archive may grow very strongly and the operations for manipulating the archive become increasingly time consuming. The second are techniques for maintaining the solutions in the archive spread in the objective space, since it is assumed that clusters of solutions are not informative for the decision maker.

3 Component-wise acceptance criterion

We first give an overview of SLS algorithms that make direct or indirect use of the component-wise ordering when deciding about the acceptance of new candidate solutions. By *direct* we mean that this decision is exclusively based on the component-wise ordering introduced in Section 2; by *indirect* we understand that from this component-wise ordering some *ranking* of candidate solutions is derived that is then finally used for deciding on which solutions to accept or choose for further manipulation. The algorithms that fall into this latter category are mainly evolutionary algorithms.

3.1 Direct use of the component-wise ordering

When designing an SLS algorithm for single-objective problems, one typically starts by implementing some form of an iterative improvement algorithm. In fact, it is relatively straightforward to apply iterative improvement algorithms also to multiobjective problems by modifying the acceptance criterion, making use of the component-wise ordering of solutions, and the use of an archive of non-dominated solutions found so far. With such modifications, iterative improvement algorithms under the CWAC search model can iteratively improve the current set of candidate solutions in the archive by adding non-dominated neighboring solutions to it [57]. Such an algorithm can be seeded either by one single solution that may be generated randomly, or by a a set of candidate solutions generated by, for example, an exact algorithm. Despite their wide spread for single-objective problems, iterative improvement algorithms for MCOPs were proposed only recently; nevertheless, almost all SLS algorithms that make direct usage of the component-wise ordering are such iterative improvement algorithms or extensions thereof.

Iterative improvement algorithms. Among the first such approaches is *Pareto local search (PLS)* by Paquete *et al.* [54, 55, 59]. PLS applies iteratively the two following steps. First, it selects randomly one candidate solution s from the archive that has not been visited and examines all neighbors of s. Second, it adds all neighbors of s that are non-weakly dominated with respect to the archive. It stops when the neighborhood of all candidate solutions has been examined. Independent of PLS, Angel *et al.* [4] proposed a similar approach called bicriteria local search (BLS). The main difference between PLS and BLS is that the latter examines the neighborhood of all candidate solutions in the archive, while PLS chooses only one and updates the archive immediately after examining a candidate solution's neighborhood. Angel *et al.* also proposed an extension of BLS by using an archive bounding technique that only accepts neighboring

solutions whose objective function value vectors do not lie in a same partition of the objective space; in addition, also a restart version of BLS was presented. A similar idea was also proposed by Laumanns *et al.* [47] for a simple evolutionary multiobjective optimizer (SEMO) that examines only one randomly chosen solution in the neighborhood. A variant of SEMO [47], called fair evolutionary multiobjective optimizer (FEMO), selects the candidate solution of the archive whose neighborhood was examined least often.

An iterative improvement algorithm with a more complex acceptance criterion, called *Pareto archived evolution strategy (PAES)*, was proposed by Knowles and Corne [42] which induces explicitly some dispersion among the solutions of a bounded-size archive. The acceptance criterion of PAES works as follows: a neighboring candidate solution s' is chosen randomly from the neighborhood of a current candidate solution s of the archive; if s' is dominated by any of the candidate solutions in the archive it is discarded, while if s' dominates candidate solutions in the archive, s' is added and the dominated candidate solutions are removed. If s' is non-dominated with respect to s and to the archived candidate solutions, one of the two following possibilities is applied: (i) if the archive is not full, s' is added to the archive; (ii) if the archive is already full, s' is only added if there exists another solution s^* lying in a partition of the objective space that contains more solutions. In that case, s^* is removed from the archive. A more complex version of PAES, called M-PAES, is proposed in [43].

Extensions of tabu search. An obvious further extension of the iterative improvement algorithms described above is to incorporate features of other general-purpose SLS methods. Few approaches in that direction have been proposed so far and the ones we are aware of are based on multiobjective tabu search. The central idea is to examine the neighborhood of a set of solutions, extract non-dominated solutions and accept among those only some non-tabu ones for inclusion into an archive. Examples of such an approach have been presented by Baykasoglu *et al.* [7, 8] and by Armetano and Arroyo [5]. In the latter approach, a bounded size archive is used and special bounding and dispersion techniques are used that are based on the location of centroids of clusters of solutions in the objective space.

3.2 Indirect use of the component-wise ordering

Many current population-based SLS algorithms rely on a mapping of the objective function value vector of each candidate solution in the archive into a single value, a *rank*, where the lower a candidate solution's rank is, the higher are its chances of being chosen. This indirect use of the component-wise ordering is mainly made by a number of *multiobjective evolutionary algorithms* (MEAs). Here, we describe the most relevant of these MEAs with particular emphasis on the ranking procedure. All these approaches consider an archive of bounded size. An illustration of the ranking procedure used in the algorithms presented next is given in Figure 1.1; the brighter the color of an objective function value vector the better is the solution considered by the ranking procedure. Note that, as said before, we do not discuss details of these approaches such as crossover or mutation operators, since these are problem-specific.

- Fonseca and Fleming [24] proposed the multiple objective genetic algorithm (MOGA), which assigns to each solution the number of solutions that weakly dominate it in the archive. Then, solutions are ranked according to those values, where ties result in ranks being averaged. A sampling algorithm chooses the next set of solutions to remain in the archive, even if some of them are dominated. The top-left plot in Figure 1.1 shows this ranking procedure, before averaging tied ranks.

- Srinivas and Deb [62] proposed the non-dominated sorting genetic algorithm (NSGA), which extended a ranking procedure initially proposed (but not tested) by Goldberg [31]: the lowest rank is assigned to the set of candidate solutions in the archive that are non-dominated. These solutions are then removed and the non-dominated solutions among the remaining candidate solutions are assigned the next rank level. This procedure is iterated until no candidate solution remains to be assigned a rank. Next, a sampling procedure is used for choosing the next set of solutions according to the ranking. Similarly to MOGA, also dominated solutions could be chosen. The top-right plot in Figure 1.1 illustrates this ranking procedure. Some further improvements are found in NSGA-II [19]; these include a faster computation of the ranks and a strategy for maintaining a dispersed set of solutions called *crowding* (this technique is explained later in this section).

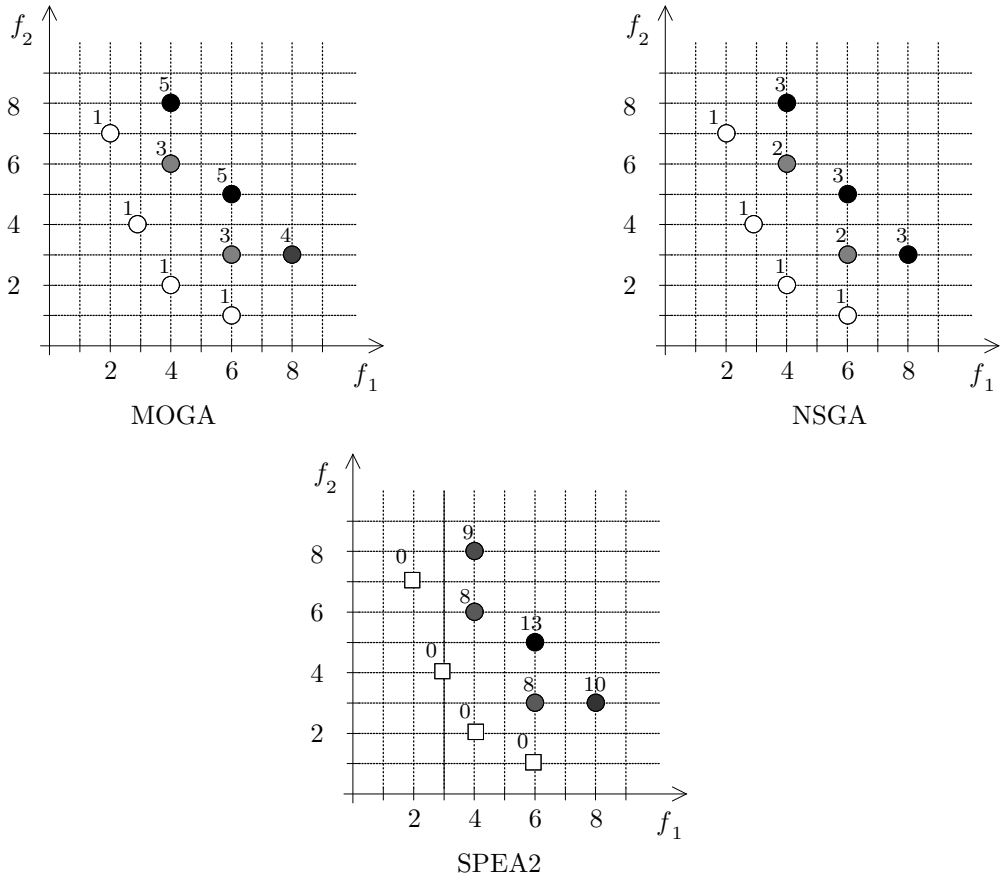

Figure 1.1: Graphical illustration of several ranking procedures in MEAs where the numbers at the points indicate the respective rank values (see text for more details.)

- Zitzler *et.al.* [68] proposed SPEA2 that maintains two sets of solutions, one being the archive of the best non-dominated solutions found so far and the second being the set of current candidate solutions that play the usual role of the population in evolutionary algorithms. Some of the solutions from the archive might be removed by a clustering algorithm if their number exceeds the maximum allowable size. The ranking procedure of SPEA2 works as follows:

  1. To each solution in the archive and in the set of current solutions is assigned the number of current solutions that are dominated by it;
  2. For each solution in both sets, its rank is given by the sum of the values computed for the solutions from both sets that weakly dominate it.

  Solutions, whose rank are less than one, are added to the archive. The size of the archive is maintained fixed at some value j in two ways: if the number of solutions becomes larger than j, solutions which are clustered in the objective space are removed (except those that are the best to each objective); if the archive is not full, the best current solutions are added to it until the total number of solutions reaches j. The bottom plot in Figure 1.1 illustrates this ranking procedure, where circles correspond to the set of current solutions and squares correspond to the solutions in the archive. The ranking procedure used by SPEA2 corrects the previous version called SPEA [69], in which a solution that is dominated by another one could be assigned a lower rank.

Since these approaches typically use an archive of fixed size, it is desirable that the existing solutions are dispersed in the objective space. Therefore, this aspect also affects how solutions are going to be chosen.

Both MOGA and NSGA use a *fitness sharing* strategy, which decreases a solution's rank depending on how large is the number of solutions in the archive within a certain radius in the objective space. NSGA-II applies a *crowding* strategy, which modifies a solution's rank based on an average distance from the nearest non-dominated solutions in the objective space. A similar approach is also used in SPEA2, where to each solution rank the inverse of the distance from the kth nearest neighbor solution in the objective space is added (k is a parameter).

Recently, some MEAs have been proposed that add a further exploration step by making direct use of the component-wise ordering. For instance, Talbi *et al.* [64] proposed an algorithm that starts with an MEA, whose final set of solutions is used afterwards as starting solutions for an iterative improvement similar to PLS; similar approaches are found also in Brizuela *et al.* [11] and Basseur *et al.* [6]. In Jozefowiez *et al.* [41], the further step after the termination of a MEA consists of a Tabu Search algorithm, with similar principles to [7], that is run several times for different regions of the objective space. Finally, Morita *et al.* [50] proposed a more complex combination by maintaining two sets of solutions such as SPEA2. At each iteration, some solutions are chosen from each set and are recombined and mutated; the resulting solutions are then added to the current set of solutions and to the archive. Then, the CWAC step consists on the examination of the neighboring solutions to a chosen one from the archive. Finally, the archive is updated and some solutions from the current set of solutions are removed in order to maintain a given cardinality at the end of each iteration.

4 Scalarized acceptance criterion

The main principle underlying the SAC search model is to use the value returned by the scalarization of the objective function vector with respect to some weight vector to distinguish between *better* and *worse* solutions. Obviously, using only one weight vector is not enough for obtaining a reasonable approximation to the efficient set. Hence, most approaches consider to change the components of the weight vector while running the algorithm in order to attain different regions of the objective space.

In the following, we divide the approaches in *non-proprietary* and *proprietary* ones. In non-proprietary approaches, an SLS algorithm is embedded into a general framework that mainly says how an underlying SLS algorithm is applied to tackle an MCOP. In the proprietary approaches, some specific, general-purpose SLS method is enhanced by additional features or specific search strategies that make it adapted for tackling MCOPs. Several examples of such proprietary approaches are therefore discussed in dependence of the underlying general-purpose SLS method.

4.1 Non-proprietary approaches

In the simplest case, a single-objective SLS algorithm could be run several, say k, times using k different weight vectors and one could for each scalarization return the best solution found by the SLS algorithm. Then, the set of objective value vectors of the k returned final solutions forms an approximation to the efficient set. In order to output a set of non-dominated solutions, the dominated solutions from the final set are removed.

Surprisingly, such a very basic approach is very rarely used, not even for a comparison to more complex algorithms. Exceptions are found in Borges and Hansen [10] and Knowles and Corne [44], who used the set of solutions returned by such an approach to get insight into certain instance features. In the following, we describe some approaches that further extend these ideas:

- Borges [9] proposed a general framework, called CHESS, whose acceptance criterion is based on a function of the distance between a new solution and an archive; in particular, the neighboring solution that is accepted to the archive is the one that maximizes the minimum difference between each component of the objective function value vector to any solution in the archive. However, this distance does not take into account any weight vector. This general rule for the acceptance criterion can be applied to SLS methods such as Simulated Annealing or Tabu Search.

- Paquete and Stützle [58] proposed two-phase local search (TPLS), which works as follows: in a first phase, a high quality solution for one objective is obtained by some high performance SLS algorithm.

Then, in the second phase, a sequence of scalarizations is solved; the initial solution of each scalarization is the one returned by the previous scalarization, the first scalarization being initialized by the solution returned from the first phase. The weight vectors that define the scalarizations in the second phase are modified according to some strategy. This strategy could consist of a random sequence of weight vectors, or of a sequence such that a small change is incurred between components of successive weight vectors. This latter strategy is the most applied one so far. For a detailed description of this algorithm see [58, 59, 53].

4.2 Proprietary approaches

Proprietary simulated annealing. Differently from the CWAC model, many algorithms that follow the SAC model use simulated annealing principles. Here, we describe some of the most relevant ones:

- Serafini [60] proposed several ways of modifying the usual probabilistic acceptance criterion of simulated annealing for tackling multiobjective problems. Given the current solution s and a neighboring solution s', he gives guidelines that should be applied to the computation of the probability p of accepting s': if $\vec{f}(s') \prec \vec{f}(s)$, then $p = 1$; if $\vec{f}(s) \prec \vec{f}(s')$, then $p < 1$; otherwise, p depends on the value returned from a function of a parameter called "temperature" and the weighted distance between $\vec{f}(s')$ and $\vec{f}(s)$ (or between $\vec{f}(s')$ and the ideal vector). In order to attain more solutions, Serafini proposed to use small random variations on the components of the weight vector during the run.

- Ulungu [65] proposed MOSA, where a set of weight vectors is defined *a priori* and for each scalarization one run of a simulated annealing algorithm is done. The probabilistic acceptance criterion in MOSA follows similar principles to those proposed by Serafini [60]. Each time a neighboring solution is accepted, an archive A_λ of non-dominated solutions for the current weight vector λ is updated; the final set of solutions returned by the algorithm is obtained after removing the dominated solutions from the union of the resulting sets A_λ.

Proprietary tabu search. Hansen [33] proposed MOTS that uses an archive, which is improved during the search process. In MOTS, only neighboring solutions that are non-tabu and the best with respect to a given scalarization can be added to the archive. In order to obtain a final set of solutions that is dispersed in the objective space, the current weight vector is updated such that neighboring solutions that are isolated in the objective space are preferably chosen. This update is done as follows: given a solution s from the archive, the i-th weight vector component increases by a fixed amount for each solution in the archive that is worse in the i-th objective; then, a *non-tabu* neighbor of s that has the best scalarized objective function value with respect to the new weight vector is chosen, added to the archive if it is non-dominated with respect to all solutions in the archive and, in that case, the tabu list is updated. Further features of MOTS are described in [33].

Proprietary memetic algorithms Several memetic algorithms for MCOPs have been proposed in analogy to principles of the single-objective case. Usually, these algorithms consist of a sequence of runs of an SLS algorithm using a scalarized objective function, each one seeded by solutions that are generated by recombination and mutation procedures applied to elements of the current set of solutions. We describe the following main approaches:

- Murata and Ishibushi [36] proposed MOGLS, which is a straightforward extension of MEAs to memetic algorithms. At each iteration, two solutions are selected from the archive by a sampling procedure that takes into account the ranking of all solutions based on a scalarization of the objective function vector with respect to a randomly generated weight vector. These two solutions are then recombined, generating a new one, which is then further improved by a local search algorithm that uses the current weight vector. While in the first proposals only one randomly chosen solution from the neighborhood is considered at each iteration of the local search, later also analyses on the influence of the strength of the local search on the overall performance have been done [37].

- Jaszkiewicz [38] proposed another memetic algorithm also called MOGLS. This algorithm maintains two sets of solutions as SPEA2, where one set maintains the current solutions while a second one corresponds to the archive of the best solutions found. At each iteration, two solutions are taken from a subset of the best current solutions with respect to a certain randomly chosen weight vector and then recombined. Next, a local search algorithm starts from this new solution using the scalarized objective function defined by the generated weight vector. The solution returned by the local search algorithm is then added to the two sets of solutions according to some acceptance rules.

5 Combination of search models

Several recently proposed algorithms combine the SAC and the CWAC search model, giving place to hybrid algorithms. Two main trends can be identified. Either the overall SLS algorithm applies two clearly distinct phases based on the two search models in sequence (sequential combination), or the SLS algorithm combines components of the two search models and iteratively changes between those in the overall search process (iterative combination).

5.1 Sequential combination

All the sequential combinations we are aware of first apply algorithms based on the SAC search model. The reason for this choice may be due to the fact that optimal solutions with respect to scalarizations of the objective function vector can identify only supported solutions, possibly leaving large gaps in the final set of solutions returned; in addition, the number of solutions returned by some methods following the SAC model is limited to the number of scalarizations. Algorithms that follow the CWAC search model can easily be applied after the SAC step with the aim of filling these gaps and increasing the number of returned solutions. Note that in the SAC step of such a combination, not necessarily SLS algorithms are required. In fact, if the scalarized problem is polynomially solvable, it is more useful to use an exact algorithm; this is also the case in the first two combinations described below.

- Hamacher and Ruhe [32] and Andersen *et al.* [3] proposed an algorithm that combines the two search models for tackling the multiobjective minimum spanning tree problem. In their approaches, the SAC step consists in obtaining several supported solutions for different scalarizations (note that a minimum spanning tree can be computed in polynomial time) and in the CWAC step, candidate solutions in the neighborhood of the supported ones are added to the archive if they are non-dominated by any other candidate solution.

- Gandibleux *et al.* [29] also proposed an algorithm to search for non-dominated solutions with respect to a set of supported solutions that was previously obtained in the SAC step. (As in [3, 32], an efficient algorithm is known to the single objective version of the MCOP tackled in that paper, the multiobjective assignment problem). The CWAC step follows principles from evolutionary algorithms: several pairs of solutions are chosen randomly from the archive and are recombined in order to generate a new set of solutions; the recombination operator used here takes into account some information about the components of the supported solutions. The so obtained solutions are then changed by a mutation procedure and, if the mutated solutions are not dominated by some previously calculated bounds, their neighborhood is explored and the non-dominated neighbors are added to the archive. This step is repeated for a given number of iterations.

- Paquete and Stützle [58] proposed a Pareto Double Two-Phase Local Search (PDTPLS) as a further extension of the TPLS approach (see page 7). For each scalarization, a solution s is returned and a CWAC step examines all neighboring solutions of s; all non-dominated solutions found in this way are returned once all scalarizations have been examined.

5.2 Iterative combination

Iterative combinations of the SAC and CWAC search models could be conceived in various ways and, hence, it is probably not surprising that the approaches in this category are more varied than when only

sequential combinations are considered. In the following we restrict ourselves to give some examples of such combinations that range from local search algorithms like tabu search and simulated annealing to population-based algorithms like memetic algorithms or ant colony optimization algorithms. These examples illustrate the range of possibilities that are opened by such iterative combinations.

- Gandibleux *et al.* [28] use tabu search principles in an algorithm called MOTS. This algorithm iteratively changes between the neighborhood exploration based on a SAC search model and CWAC search model. In the SAC step, a non-tabu solution that minimizes the distance from a *local* utopian point[1] or a tabu but aspired solution replaces the current one. Once a new solution in the neighborhood of the current one, s, is accepted, a CWAC step adds the non-dominated neighbors (or a subset thereof) of s to the archive. In order to favor more isolated regions of the objective space, the weight vector is updated periodically; to maintain the diversity of the search process, a new weight vector is declared tabu for a given number of iterations.

- Abdelaziz and Krichen [2] developed an algorithm that was also called MOTS. Given some solution s, the SAC step works as follows: a constructive algorithm is run several times for several scalarizations in order to generate a set S of solutions; for each run of the constructive algorithm, the weight vector takes into account the two worst components of the objective function value vector of s multiplied by some random values. The neighboring solutions to s are added to S, from which then the dominated solutions are removed and the resulting set S' is used to update the archive. Then, a CWAC step that follows similar principles to [7] (see also page 4) is applied for a given number of iterations. The iterative combination of the SAC and CWAC steps is repeated until no solution can be added to the archive during some iterations. A further extension, using a MEA, has been proposed in [1].

- Czyzak and Jaszkiewicz [16] proposed an algorithm which further extends the simulated annealing principles for tackling MCOPs. First, a set of solutions S is generated randomly and each solution from this set is assigned a weight and added to the archive. For each candidate solution s from S, a neighboring solution s' is added to the archive if not dominated by s (CWAC step). The SAC step is then applied for determining s' as the new current solution according to probabilistic rules similar to the ones proposed by Serafini; the weight vector is changed automatically during the search process in order to favor the acceptance of neighboring solutions that are more *isolated* in the objective space. This iterative process is repeated for all solutions in S.

- López-Ibáñez *et al.* [49] proposed a combination of SPEA2 with SLS algorithms using a scalarized acceptance criterion, which combines a CWAC step, as executed by SPEA2, with a SAC step used for defining the search direction for the SLS algorithm. In that proposal, the SLS algorithm is applied to every new solution obtained from a recombination of solutions in the archive using randomly generated weights.

Recently, there has been some interest in applying ant colony optimization (ACO) algorithms [21] to MCOPs. ACO algorithms are based on the repeated, construction of solutions that is stochastically biased by artificial *pheromone* trails (pheromone trails are essentially some numerical information attached to solution components that are used in the solution construction) and heuristic information on the problem under being solved. The pheromone trails are updated during the algorithm's run in dependence of the search experience. Interestingly, all applications of ACO algorithms to MCOPs that are solved under the notion of Pareto optimality combine the two search models: The solution construction is typically based on a SAC step, where weights are defined to join various types of pheromone information with respect to the various objectives, while the artificial ants (representing the generated solutions) that update the pheromone information are typically chosen based on a CWAC step. Some approaches are described next:

- Iredi *et al.* [35] proposed several extensions of ACO algorithms where the pheromone and heuristic information are associated to different weight vectors, defined according to subintervals within the range $[0, 1]$. Thus, different weight vectors direct the constructive steps towards different regions of the objective space. The artificial pheromones are then updated with the non-dominated solutions

[1] An utopian point dominates the ideal point; In MOTS, a local utopian point dominates all neighbors of the current solution.

found. Extensions of these algorithms were also proposed by López-Ibáñez *et al.* [48, 49], where an SLS algorithm based on a scalarization of the objective function vector is applied after each new constructed solution.

- Dörner *et al.* [20] describe an ACO algorithm that uses one pheromone matrix for each objective but only one same type of heuristic information. Each solution is constructed based on the heuristic information and on the weighted aggregation of the pheromone matrices according to a randomly chosen weight vector. Some of the best non-dominated solutions found are then used for updating the pheromones.

6 Conclusions

We have reviewed the research on SLS algorithms for tackling MCOPs with respect to the notion of Pareto optimality. The existing approaches were classified according to whether they use the SAC and CWAC search models or some combinations thereof. For each of the resulting main classes of algorithms we have shortly described the main representatives without the intention of providing a fully comprehensive enumeration of all existing proposals, which would be almost impossible, given their large number and the limited amount of space. While we described here only the main features of available approaches, several other interesting details can be found in the original papers.

Among the possible approaches for tackling MCOPs, a recent and interesting one is to use an acceptance criterion based on solution quality indicators. Such an algorithm has been proposed by Knowles et al. [46]; it applies at each step the hypervolume indicator [69] to decide which solutions are added to the archive. A similar approach has also been followed in [67], where binary performance indicators are used, that is, quality indicators that return a pair of values for each non-dominated set. We remark that the performance of these algorithm strongly also depends on the performance of the underlying algorithm for computing the quality indicators and we expect further development on ways of computing these more efficiently.

There are a few important areas related to SLS algorithms for MCOPs that were not covered here. One of the most important is the empirical assessment of the performance of these algorithms. This is far from being a trivial issue since, as shown by Zitzler et al. [70], frequently used quality indicators for comparing non-dominated sets obtained by SLS algorithms have severe problems. A recognized exception is the use of attainment functions. Initially proposed in [18], the attainment function characterizes the performance of the SLS algorithms by describing the distribution of the outcomes. This function has shown to be a first-order moment measure of these outcomes and it can be seen as generalization of the multivariate empirical distribution function [17]. This allows the use of statistical inference and experimental design techniques to *infer* conclusions on the performance of SLS algorithms [61, 56]. Some further extension of attainment functions for second-order moments can be found in [23].

Finally, we remark that little is known on the dependence between the performance of SLS algorithms and certain features of the MCOPs. Exceptions can be found in Mote et al. [51] and Müller-Hannemann and Weihe [52] who identified several features of the Multiobjective Shortest Path problem and variations that translate in a tractable number of Pareto optimal solutions, which contrasts with the known worst case for the same problem in [34]. In [10, 55], it is conjectured that most efficient solutions and approximations thereof are strongly clustered in the solution space for the multiobjective traveling salesman problem with respect to a small sized neighborhood. This means that local search algorithms using this neighborhood are very suitable for this problem, which is confirmed in [58]. In addition, the correlation between objectives seems to have a strong effect on the choice for the search model; in [59], experimental results indicate that simple SLS algorithms based on the SAC and CWAC search model, respectively, can behave strongly differently for the multiobjective quadratic assignment problem as the correlation between objectives is changed. Therefore, an interesting future line of research is to investigate, both experimentally and analytically, which and how instance features affect the performance of SLS algorithms.

While the research field of applying SLS algorithms to MCOPs poses many open questions and significant research issues, the set of available SLS algorithms shows that nowadays it is becoming increasingly feasible to tackle MCOPs with respect to the notion of Pareto optimality and that therefore these approaches are very likely to receive more attention for the solution of difficult real-world multiobjective problems.

Acknowledgments

This work was supported in part by the "Metaheuristics Network", a Research Training Network funded by the Improving Human Potential programme of the CEC. The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data appearing in this publication. Thomas Stützle acknowledges support of the Belgian FNRS, of which he is a research associate.

References

[1] F. B. Abdelaziz, J. Chaouachi, and S. Krichen. A hybrid heuristic for multiobjective knapsack problems. In S. Voss, S. Martello, I. Osman, and C. Roucairol, editors, *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pages 205–212. Kluwer Academic Publishers, Boston, MA, 1999.

[2] F. B. Abdelaziz and S. Krichen. A tabu search heuristic for multiobjective knapsack problems. Technical Report RRR 28-97, Rutgers Center for Operations Research, Piscataway, New Jersey, USA, December 1997.

[3] K. Andersen, K. Jörnsten, and M. Lind. On bicriterion minimal spanning trees: An approximation. *Computers & Operations Research*, 23(12):1171–1182, 1996.

[4] E. Angel, E. Bampis, and L. Gourvés. A dynasearch neighborhood for the bicriteria traveling salesman problem. In Gandibleux et al. [30], pages 153–176.

[5] V. A. Armetano and J. E. Arroyo. An application of a multi-objective tabu search algorithm to a bicriteria flowshop problem. *Journal of Heuristics*, 10(5):463–481, 2004.

[6] M. Basseur, F. Seynhaeve, and E. G. Talbi. Design of multi-objective evolutionary algorithms: Application to the flow-shop. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)*, volume 2, pages 151–1156. IEEE Press, Piscataway, NJ, 2002.

[7] A. Baykasoglu, S. Owen, and N. Gindy. A taboo search based approach to find the Pareto optimal set in multiobjective optimization. *Journal of Engineering Optimization*, 31:731–748, 1999.

[8] A. Baykasoglu, L. Özbakir, and A. Sönmez. Using multiple objective tabu search and grammars to model and solve multi-objective flexible job shop scheduling problems. *Journal of Intelligent Manufacturing*, 15:777–785, 2004.

[9] P. Borges. CHESS – Changing Horizon Efficient Set Search: A simple principle for multiobjective optimization. *Journal of Heuristics*, 6(3):405–418, 2000.

[10] P. C. Borges and M. P. Hansen. A study of global convexity for a multiple objective travelling salesman problem. In C. C. Ribeiro and P. Hansen, editors, *Essays and Surveys in Metaheuristics*, pages 129–150. Kluwer Academic Publishers, Boston, MA, 2000.

[11] C. A. Brizuela, N. Sannomiya, and Y. Zhao. Multi-objective flow-shop: Preliminary results. In Zitzler et al. [66], pages 443–457.

[12] C. C. Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1:269–308, 1999.

[13] C. C. Coello. An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys*, 32:109–143, 2000.

[14] C. C. Coello. Special issue on "Evolutionary Multiobjective Optimization". *IEEE Transactions on Evolutionary Computation*, 7(2), 2003.

[15] C. C. Coello, A. H. Aguirre, and E. Zitzler, editors. *Evolutionary Multi-criterion Optimization (EMO 2005)*, volume 3410 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, Germany, 2005.

[16] P. Czyzak and A. Jaszkiewicz. Pareto simulated annealing - a metaheuristic technique for multiple objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7:34–47, 1998.

[17] V. Grunert da Fonseca and C. M. Fonseca. A link between the multivariate cumulative distribution function and the hitting function for random closed sets. *Statistics & Probability Letters*, 57(2):179–182, 2002.

[18] V. Grunert da Fonseca, C. M. Fonseca, and A. Hall. Inferential performance assessment of stochastic optimizers and the attainment function. In Zitzler et al. [66], pages 213–225.

[19] K. Deb, S. Agrawal, A. Prataand, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[20] K. Doerner, W. J. Gutjahr, C. Strauss, and C. Stummer. Pareto ant colony optimization: A metaheuristic approach to portfolio selection. *Annals of Operations Research*, 131:79–99, 2004.

[21] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.

[22] M. Ehrgott and X. Gandibleux. Approximative solution methods for combinatorial multicriteria optimization. *TOP*, 12(1):1–90, 2004.

[23] C. M. Fonseca, V. Grunert da Fonseca, and L. Paquete. Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In Coello et al. [15], pages 250–264.

[24] C. M. Fonseca and P. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrester, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[25] C. M. Fonseca and P. Fleming. An overview of evolutionary algorithms in multiobjective algorithms. *Evolutionary Computation*, 3:1–16, 1995.

[26] C. M. Fonseca, P. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors. *Evolutionary Multi-criterion Optimization (EMO 2003)*, volume 2632 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, Germany, 2003.

[27] X. Gandibleux, A. Jaszkiewicz, A. Fréville, and R. Slowinski. Special issue on "Multiple Objective Metaheuristics". *Journal of Heuristics*, 6(3), 2000.

[28] X. Gandibleux, N. Mezdaoui, and A. Fréville. A tabu search procedure to solve multiobjective combinatorial optimization problems. In R. Caballero, F. Ruiz, and R. Steuer, editors, *Advances in Multiple Objective and Goal Programming*, volume 455 of *Lecture Notes in Economics and Mathematical Systems*, pages 291–300. Springer Verlag, Berlin, Germany, 1997.

[29] X. Gandibleux, H. Morita, and N. Katoh. Use of a genetic heritage for solving the assignment problem. In Fonseca et al. [26], pages 43–57.

[30] X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt, editors. *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*. Springer Verlag, 2004.

[31] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[32] H. V. Hamacher and G. Ruhe. On spanning tree problems with multiple objectives. *Annals of Operations Research*, 52:209–230, 1994.

[33] M. P. Hansen. Tabu search for multiobjective optimisation: MOTS. In *Proceedings of the 13th International Conference on Multiple Criteria Decision Making (MCDM 1997)*, Cape Town, South Africa, January 1997.

[34] P. Hansen. Bicriterion path problems. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Application*, volume 177 of *Lecture Notes in Economics and MAthematical Systems*, pages 109–127. Springer Verlag, Berlin, Germany, 1979.

[35] S. Iredi, D. Merkle, and M. Middendorf. Bi-criterion optimization with multi colony ant algorithms. In Zitzler et al. [66].

[36] H. Ishibuchi and T. Murata. A multi-objective genetic local search algorithm and its application to flow-shop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics – Part C*, 28(3):392–403, 1998.

[37] H. Ishibuchi, T. Yoshida, and T. Murata. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transactions on Evolutionary Computation*, 7(2):204–223, 2003.

[38] A. Jaszkiewicz. Genetic local search for multiple objective combinatorial optimization. Technical Report RA-014/98, Institute of Computing Science, Poznań University of Technology, Poznań, Poland, December 1998.

[39] A. Jaszkiewicz. Special issue on "Evolutionary and Local Search Heuristics in Multiple Objective Optimization". *Foundations of Computing and Decision Sciences Journal*, 26(1), 2001.

[40] D. Jones, S. Mirrazavi, and M. Tamiz. Multi-objetive meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137(1-9), 2002.

[41] N. Jozefowiez, F. Semet, and E .-G. Talbi. Parallel and hybrid models for multi-objective optimization: Application to the vehicle routing problem. In H. G. Beyer J. J. Merelo, P. Adamidis, editor, *Proceedings of PPSN-VII, Seventh International Conference on Parallel Problem Solving from Nature*, volume 2439 of *Lecture Notes in Computer Science*, pages 271–280. Springer Verlag, Berlin, Germany, 2002.

[42] J. Knowles and D. Corne. The Pareto archived evolution strategy: A new base line algorithm for multiobjective optimisation. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, pages 98–105, Washington, 1999.

[43] J. Knowles and D. Corne. M-PAES: A memetic algorithm for multiobjective optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation (CEC'00)*, volume 1, pages 325–332, Piscataway, New Jersey, USA, July 2000.

[44] J. Knowles and D. Corne. Towards landscape analyses to inform the design of a hybrid local search for the multiobjective quadratic assignment problem. In A. Abraham, J. Ruiz del Solar, and M. Koppen, editors, *Soft Computing Systems: Design, Management and Applications*, pages 271–279. IOS Press, Amsterdam, The Netherlands, 2002.

[45] J. Knowles and D. Corne. Memetic algorithms for multiobjective optimization: Issues, methods and prospects. In N. Krasnogor, J. Smith, and W. Hart, editors, *Recent Advances in Memetic Algorithms*, volume 166 of *Studies in Fuzziness and Soft Computing*, pages 313–352. Springer Verlag, 2004.

[46] J. Knowles, D. Corne, and M. Fleischer. Bounded archiving using the lebesgue measure. In R. Sarker et al., editors, *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'03)*, pages 2490–2498. IEEE Press, Piscataway, NJ, 2003.

[47] M. Laumanns, L. Thiele, E. Zitzler, E. Welzl, and K. Deb. Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In J. M. Guervos et al., editors, *Proceedings of PPSN-VII, Seventh International Conference on Parallel Problem Solving from Nature*, volume 2439 of *Lecture Notes in Computer Science*, pages 44–53. Springer Verlag, Berlin, Germany, 2002.

[48] M. López-Ibáñez, L. Paquete, and T. Stützle. On the design of ACO for the biobjective quadratic assignment problem. In M. Dorigo, L. Gambardella, F. Mondada, and T. Stützle, editors, *Proceedings of ANTS 2004 – From Ant Colonies to Artificial Ants: Fourth International Workshop on Ant Algorithms*, volume 3172 of *Lecture Notes in Computer Science*, pages 214–225. Springer Verlag, Berlin, Germany, 2004.

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

[50] H. Morita, X. Gandibleux, and N. Katoh. Experimental feedback on biobjective permutation scheduling problems solved with a population heuristic. *Foundations of Computing and Decision Sciences*, 26(1):23–50, 2001.

[51] J. Mote, I. Murthy, and D. L. Olson. A parametric approach to solving bicriterion shortest path problem. *European Journal of Operational Research*, 53:81–92, 1991.

[52] M. Müller-Hannemann and K. Weihe. Pareto shortest paths is often feasible in practice. In G. S. Brodal, D. Frigioni, and A. M. Spaccamela, editors, *Algorithm Engineering, 5th International Workshop (WAE 2001)*, volume 2141 of *Lecture Notes in Computer Science*, pages 185–197. Springer Verlag, 2001.

[53] L. Paquete. *Stochastic Local Search Algorithms for Multiobjective Combinatorial Optimization: Methods and Analysis*. PhD thesis, FB Informatik, TU Darmstadt, Germany, 2005.

[54] L. Paquete, M. Chiarandini, and T. Stützle. A study of local optima in the multiobjective traveling salesman problem. Technical Report AIDA-02-07, Fachgebiet Intellektik, Fachbereich Informatik, Technische Universität Darmstadt, 2002.

[55] L. Paquete, M. Chiarandini, and T. Stützle. Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study. In Gandibleux et al. [30], pages 177–200.

[56] L. Paquete, M. López-Ibáñez, and T. Stützle. Towards an empirical analysis of SLS algorithms for multiobjective combinatorial optimization problems by experimental design. In *Proceedings of the Fifth Metaheuristics International Conference*, 2005.

[57] L. Paquete, T. Schiavinotto, and T. Stützle. On local optima in multiobjective combinatorial optimization problems. Technical Report AIDA–04–11, FG Intellektik, TU Darmstadt, November 2004.

[58] L. Paquete and T. Stützle. A two-phase local search for the biobjective traveling salesman problem. In Fonseca et al. [26], pages 479–493.

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

[60] P. Serafini. Simulated annealing for multiobjective optimization problems. In G. H. Tzeng, H. F. Wang, V. P. Wen, and P. L. Yu, editors, *Multiple Criteria Decision Making. Expand and Enrich the Domains of Thinking and Application*, Lecture Notes in Economics and Mathematical Systems, pages 283–292. Springer Verlag, Berlin, Germany, 1992.

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

[62] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 3(2):221–248, 1994.

[63] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley & Sons, New York, NY, 1986.

[64] E. G. Talbi. A hybrid evolutionary approach for multicriteria optimization problems: Application to the flow shop. In Fonseca et al. [26], pages 416–428.

[65] E. L. Ulungu. *Optimisation combinatoire multicritére: Détermination de l'ensemble des solutions efficaces et méthodes interactives*. PhD thesis, Université de Mons-Hainaut, Mons, Belgium, 1993.

[66] E. Zitzler, K. Deb, L. Thiele, C. C. Coello, and D. Corne, editors. *Evolutionary Multi-criterion Optimization (EMO 2001)*, volume 1993 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, Germany, 2001.

[67] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In X. Yao et al., editors, *Proceedings of PPSN-VIII, Eight International Conference on Parallel Problem Solving from Nature*, volume 3242 of *Lecture Notes in Computer Science*, pages 832–842. Springer Verlag, Berlin, Germany, 2004.

[68] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. Technical Report TIK-Report No. 103, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Zürich, May 2001.

[69] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 4(3):257–271, 1999.

[70] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.