The Multiobjective Traveling Salesman Problem: A Survey and a New Approach.

Thibaut Lust[*] and Jacques Teghem

Faculté Polytechnique de Mons
Laboratory of Mathematics and Operational Research
9, rue de Houdain
7000 Mons, Belgium
thibaut.lust@fpms.ac.be, jacques.teghem@fpms.ac.be

Summary. The traveling salesman problem (TSP) is a challenging problem in combinatorial optimization. In this paper we consider the multiobjective TSP for which the aim is to obtain or to approximate the set of efficient solutions. In a first step, we classify and describe briefly the existing works, that are essentially based on the use of metaheuristics. In a second step, we propose a new method, called two-phase Pareto local search. In the first phase of this method, an initial population composed of an approximation to the extreme supported efficient solutions is generated. The second phase is a Pareto local search applied to all solutions of the initial population. The method does not use any numerical parameter. We show that using the combination of these two techniques—good initial population generation and Pareto local search—gives, on the majority of the instances tested, better results than state-of-the-art algorithms.

1 Introduction

Since the 70ies, *multiobjective optimization* problems (MOP) became an important field of operations research. In many real applications, there exists effectively more than one objective to be taken into account to evaluate the quality of the feasible solutions.

A MOP is defined as follows:

$$\begin{array}{lll} \text{“min”} & z(x) = z_k(x) \quad k = 1, \ldots, p & \text{(MOP)} \\ \text{s.t} & x \in X \subset \mathbb{R}^n_+ & \end{array}$$

where n is the number of variables, $z_k(x) : \mathbb{R}^n_+ \rightarrow \mathbb{R}$ represents the k^{th} objective function and X is the set of feasible solutions. We will denote by $Z = \{z(x) : x \in X\} \subset \mathbb{R}^p$ the image of X in the objective space.

[*] T. Lust thanks the “Fonds National de la Recherche Scientifique” for a research fellow grant (Aspirant FNRS).

Due to the typically conflictive objectives, the notion of optimal solution does not exist generally anymore for MOPs. However, based on the dominance relation of Pareto (see definition 1), the notion of optimal solution can be replaced by the notion of efficient (or Pareto optimal) solution (see definition 2).

Definition 1. *A vector $z \in Z$ dominates a vector $z' \in Z$ if, and only if, $z_k \leq z'_k$, $\forall\, k \in \{1, \ldots, p\}$, with at least one index k for which the inequality is strict. We denote this dominance relation by $z \prec z'$.*

Definition 2. *A feasible solution $x \in X$ is efficient if there does not exist any other solution $x' \in X$ such that $z(x') \prec z(x)$. The image of an efficient solution in objective space is called a non-dominated point.*

In the following, we will denote by X_E, called efficient set, the set of all efficient solutions and by Z_N, called the Pareto front, the image of X_E in the objective space.

The following additional definitions have been introduced by Hansen [22].

Definition 3. *Equivalent solutions: two solutions $x_1, x_2 \in X_E$ are equivalent if $z(x_1) = z(x_2)$.*

Definition 4. *Complete set: a complete set X_{Ec} is a subset of X_E such that each $x \in X \setminus X_{Ec}$ is either dominated by or equivalent to at least one $x \in X_{Ec}$. In other words, for each non-dominated point $z \in Z_N$ there exists at least one $x \in X_{Ec}$ with $z(x) = z$.*

Definition 5. *Minimal complete set: a minimal complete set X_{Em} is a complete set without equivalent solutions. Any complete set contains at least a minimal complete set.*

Even if other approaches exist to tackle a MOP problem (aggregation of the objectives with a utility function, hierarchy of the objectives, goal programming, interactive method to generate a “ good compromise ”: see [53]), in this paper we are only interested in the determination, or the approximation, of X_E and Z_N. It should be noted that in all heuristics presented in this work, only an approximation of a minimal complete set is determined: no equivalent solution generated will thus be retained.

It is first the problems with continuous variables which called the attention of the researchers: see the book of Steuer [52] for multiobjective *linear programming* (MOLP) problems and of Miettinen [43] for multiobjective *non-linear programming* [43] (MONLP) problems.

However it is well-known that discrete variables are often unavoidable in the modeling of many applications and for such problems the determination of X_E and Z_N becomes more difficult.

Let us consider for instance a multiobjective *integer linear programming* (MOILP) problem of the form:

$$\begin{array}{lll} \text{“min”} & z(x) = z_k(x) = c^k x \quad k = 1, \ldots, p & \text{(MOILP)} \\ \text{s.t} & x \in X = \{x \in \mathbb{Z}^n_+ : Ax = b\} & \end{array}$$

In such case, we can distinguish two types of efficient solutions:

- The supported efficient solutions are optimal solutions of the weighted single-objective problem
$$\begin{array}{ll} \min \sum_{k=1}^{p} \lambda_k z_k(x) \\ \text{s.t} \quad x \in X \end{array}$$
where $\lambda \in \mathbb{R}^p_+$ is a weight vector with all positive components λ_k, $k = 1, \ldots, p$. We denote by X_{SE} and Z_{SN} respectively the set of supported efficient solutions and the set of corresponding non-dominated points in $\mathbb{R}^p$. The points of Z_{SN} are located on the frontier of the convex hull of Z.
- Contradictorily with a MOLP problem Z_{SN} is generally a proper subset of Z_N due to the non-convex character of Z: there exist efficient solutions which are non-supported. We denote by $X_{NE} = X_E \backslash X_{SE}$ and $Z_{NN} = Z_N \backslash Z_{SN}$ respectively the set of non-supported efficient solutions and the set of the corresponding non-dominated points in $\mathbb{R}^p$.

Inside X_{SE}, it is useful to distinguish the extreme supported efficient solutions, i.e., the solutions $x \in X_{SE}$ such that $z(x)$ is a vertex of the convex hull of Z, and the others, the non-extreme supported solutions. We denote by X_{SE1} and $X_{SE2} = X_{SE} \backslash X_{SE1}$ respectively the set of extreme and non-extreme supported solutions and by Z_{SE1} and Z_{SE2} the corresponding sets of non-dominated points in the objective space.

Already in the 80ies, several methods have been proposed to generate X_E for MOILP problems [54]. The two main difficulties to overcome are that:

- The sets X_E and Z_N, formed of discrete points, can be of very large cardinality;
- The sets X_{NE} and Z_{NN} are more difficult to determine.

Later, various multiobjective *combinatorial* optimization (MOCO) problems have been considered. Most of them are of the type

$$\begin{array}{lll} \text{“min”} & z(x) = c^k(x) \quad k = 1, \ldots, p & \text{(MOCO)} \\ \text{s.t} & x \in X = D \cap \{0,1\}^n & \end{array}$$

where D is a specific polytope characterizing the particular CO problem.

During the last 15 years, there has been a notable increase of the number of studies on MOCO problems. From the first survey [55] in 1994 till [8] in 2002, a lot of papers have been published and this flow is still increasing. The main reason of this phenomenon is the success story of the metaheuristics [16].

Effectively, it is quite difficult to determine exactly the sets X_E and Z_N for MOCO problems. This is a NP-hard problem even for CO problems for which a polynomial algorithm for the single-objective version exists such as linear assignment problem. Therefore, there exist only few exact methods able to determine the sets X_E and Z_N and we can expect to apply these methods only for small instances. For this reason, many methods are heuristic methods which produce approximations $\widehat{X_E}$ and $\widehat{Z_N}$ to the sets X_E and Z_N. Due to the succes of metaheuristics for single-objective CO, multiobjective *metaheuristics* (MOMH) became quickly a classic tool to tackle MOCO problems and it is presently a real challenge for the researchers to improve the results previously obtained.

The two main difficulties of MOMH are related to the basic needs of any metaheuristics [16]:

- To assure sufficient intensity, i.e. to produce an approximation $\widehat{Z_N}$ as close as possible to Z_N;
- To assure sufficient diversity, i.e. to cover with $\widehat{Z_N}$ all the parts of Z_N.

Unfortunately, measuring the quality of an approximation or to compare the approximations obtained by various methods remains a difficult task: the problem of the quality assessment of the results of a MOMH method is in fact also a multicriteria problem. Consequently, several indicators have been introduced in the literature to measure the quality of an approximation (see [60] for instance).

Some of them are unary indicators:

- The hypervolume $\mathcal{H}$ (to be maximized) [59]: the volume of the dominated space defined by $\widehat{Z_N}$, limited by a reference point.
- The R measure (normalized between 0 and 1, to be maximized) [26]: evaluation of $\widehat{Z_N}$ by the expected value of the weighted Tchebycheff utility function over a set of normalized weight vectors.
- The average distance D_1 and maximal distance D_2 (to be minimized) [7, 57] between the points of a reference set and the points of $\widehat{Z_N}$, by using the Euclidean distance. Ideally, the reference set is Z_N itself, but generally it is not available ; otherwise, it can be the non-dominated points existing among the union of various sets $\widehat{Z_N}$ generated by several methods, or a lower bound of Z_N [9].

Other indicators are introduced to compare two approximation sets A and B of Z_N. In particular, the $I_\epsilon(A,B)$ [61] indicator gives the ϵ factor by which the approximation A is worse (if $\epsilon > 1$) or better (if $\epsilon < 1$) than the approximation B with respect to all objectives:

$$I_\epsilon(A,B) = \inf_{\epsilon \in \mathbb{R}^+} \{\forall\, z \in B, \exists\, z' \in A : z'_k \leq \epsilon \cdot z_k,\, k = 1, \ldots, p\}$$

If B is a reference set, this indicator can also be used as a unary indicator, denoted by $I_{\epsilon 1}$.

Unfortunately, none of these indicators allows to conclude that an approximation is better than another one (see [61] for details). Nevertheless, an approximation that finds better values for these indicators is generally preferred to others.

In the first time, the MOCO problems treated in the literature are those for which there exist efficient algorithms in the single-objective case, essentially linear assignment problem, shortest path problem or knapsack problem.

More recently, a large attention has been devoted to the multiobjective *traveling salesman* problem (MOTSP). In the next section, we survey briefly the existing MOTSP literature. Section 3 is dedicated to a new heuristic method applied to the biobjective TSP. Its results are presented in section 4 before a short conclusion in section 5.

2 The multiobjective TSP literature

The traveling salesman problem is certainly the best-known and most studied NP-hard single-objective CO problem.

The book of Gutin and Punnen [18] analyzes various methods for the TSP and for some of its variants; the chapters of Johnson and McGeoch [29, 30] are devoted to heuristics respectively for the symmetric and asymmetric versions of the TSP.

Other interesting reviews are related to the use of metaheuristics for the TSP: [28] is a survey of local search (LS) methods, [35] of genetic algorithms (GA) and [42] of memetic algorithms applied to TSP.

We recall the formulation of the MOTSP: given N cities and p costs $c^k_{i,j}\,(k = 1, \ldots, p)$ to travel from city i to city j, the MOTSP consists of finding a tour, i.e. a cyclic permutation ρ of the N cities, minimizing

$$\text{“}\min\text{”}\, z_k(\rho) = \sum_{i=1}^{N-1} c^k_{\rho(i),\rho(i+1)} + c^k_{\rho(N),\rho(1)},\; k = 1, \ldots, p$$

The majority of the instances treated in the following cited papers concern the symmetric biobjective case ($p = 2$); sometimes $p = 3$.

One of the first paper concerning MOTSP is the study of Borges and Hansen [6]. It is devoted to the “global convexity” in MOCO problems in general, MOTSP in particular. The authors analyzed the landscape of local optima, considering the classical 2-opt neighborhood (two edges are deleted and replaced with the only possible pair of new edges that does not break the tour) and using well-known scalarizing functions like the weighted sum of the objectives or the Tchebycheff function. They indicated the implications of the distribution of local optima in building MOMH to approximate Z_N.

Almost all the existing methods are MOMH. In these methods, a set $\widehat{X_E} \equiv \widehat{X_{Ec}}$ of potential efficient solutions is updated each time a new solution x is generated: x is added to $\widehat{X_E}$ if no solution of $\widehat{X_E}$ dominates x or is equivalent to x and all the solutions of $\widehat{X_E}$ dominated by x are removed. At the end of the procedure, $\widehat{X_E}$ is the proposed approximation.

In order to present the MOMH suited to solve the MOTSP, we made a distinction between MOMH based on evolutionary algorithms, based on local search and we finally present particular studies and variants of the MOTSP.

2.1 Evolutionary Algorithms (EAs)

We first present MOMH essentially based on EAs.

- One of the best performing MOMH is the MOGLS (multiple objective genetic local search) method proposed by Jaszkiewicz [26] so that the performance of MOGLS is often taken as reference for comparisons.
  MOGLS combines GA and LS and works as follows:
  - An initial population P is built: each solution is optimized by a local search applied to a weighted sum of the objectives.
  - At each iteration, a random utility function is selected. The best solutions in P for this function form a temporary population TP. Inside TP, two solutions x_1 and x_2 are randomly selected and recombined to generate a solution x_3; the local search is applied to x_3 obtaining x_3'. If x_3' is better that the worst solution of P according to the random utility function, x_3' is added to P. To treat a MOTSP problem, the local search is a standard 2-opt exchange and the recombination operator is the distance preserving crossover DPX: all edges common to both parents are put in the offspring which is then completed by randomly selected edges to form a Hamiltonian cycle. The considered instances—with two or three objectives—are based on the TSPLIB library [49].
- The memetic random key genetic algorithm described in [51] is inspired by MOGLS. The crossover operator is different: first a random number is associated to each city of each parent and the chromosome is represented alternatively by these numbers. A classical one-point crossover is applied but the random numbers are used as "sort keys" to decode the new solutions: the cities are sorted in the ascending order of their corresponding keys to indicate the travel order of the salesman. Unfortunately, the results are not compared with those of [26] (but are better than those of [20]).
- Very recently, Jaszkiewicz and Zielniewicz [27] analyzed the idea of path relinking (search for new efficient solutions along path in the decision space connecting two potential efficient solutions) for MOTSP. Whenever two parents are selected for recombination, a path in the decision space linking the two solutions is built by the local search that starts from one of the parents. The local search uses lexicographic objective function: first it minimizes the distance (measured by the number of common arcs) to the second parent. In case of several solutions having the same distance they are compared with the current scalarizing function. This Pareto memetic algorithm (PMA) with path-relinking is further extended with the use of one iteration of the Pareto local search method of Paquete *et al.* [45] (see

section 2.2). Both Pareto local search and path-relinking use a simple 2-opt neighborhood. However, they use for the generation of the initial population and for improving the solutions after recombination a greedy version of the Lin-Kernighan algorithm [37]. The results of PMA improve those of PD-TPLS [46] (see section 2.3) but with higher running time and higher number of parameters.

- A quite different MOEA is proposed by Yan *et al.* [58].
  The main characteristics are that:
  - A reproduction operator, called “inver-over”, is defined to replace the tradional crossover and mutation operators: a son is created from a father solution making use of a particular comparison with a solution randomly selected in the population.
  - Niches are defined with a sharing function (see chapter 6 of [8]) to preserve the diversity of the population.
  - Rather than assigning a fitness value to solutions, a boolean function “better” is defined to sort the individuals of the population, from the subset of best solutions to the subset of worst solutions.
  - To generate a new population, a competition is chosen randomly, based on a fixed probability, between two types: a “family competition”, to replace the father solution by the son solution if the second is better than the first; a “population competition” which compares the son solution with a randomly chosen solution in the population.

  Various two objectives instances are randomly generated and the results obtained with this new MOEA outperforms those obtained by the classical MOEA called SPEA (see chapter 6 of [8]).

- Kumar and Singh [34] proposed a Pareto converging genetic algorithm (PCGA) which is a hybridization of Pareto-rank GA with a LS.
  Each solution of the population is evaluated by its rank equal to one plus the number of individuals dominating it. Then two individuals are determined by a conventional roulette wheel selection and are crossed-over using the DPX operator to produce offspring. The offspring is inserted into the population according to its rank against the whole set of individuals. Mutation is also applied. An original convergence criterion based on “rank-histograms” is defined. Inside a population, the rank of an individual is equal to one plus the number of individuals dominating it so that all the non-dominated individuals are assigned to one. Two successive populations P_{t-1} and P_t are considered and the union $P_{t-1} \cup P_t$ is ranked. Taking each rank in turn, the histogram is the fraction of the solutions from P_t among those from $P_{t-1} \cup P_t$ having this rank. The perfect convergence corresponds to a rank histogram equal to a single non-zero entry of 0.5 for rank equal to 1 indicating that no solutions superior to those in P_{t-1} have been generated in evolving the later generation P_t.
  The hybridization of the PCGA is made with the 3-opt LS. To form the initial population, the LS is applied with each objective separately from in-

dividuals generated randomly. For the other populations, the LS considers the objectives simultaneously and uses Pareto-ranking.
Comparison is made with the results of MOGLS and PD-TPLS (see next section) and, roughly speaking, the results appear equivalent between the three methods.

- Very recently, Elaoud *et al.* [10] proposed a multiple crossover GA for the MOTSP. As many types of crossover and mutation operators have been proposed for TSP (see [35]), all of them are included in the algorithm.
At each iteration of the GA, a pair (crossover, mutation) is randomly selected based on a probability distribution. At the beginning, this distribution is uniform. Then, depending on the performance of the iteration—measured by the numbers of new solutions entering in $\widehat{X_E}$ and solutions rejected from $\widehat{X_E}$— the probability distribution is modified into a dynamic scheme to reward or not the selected pair of operators. Experimental results and comparison with those of MOGLS show the synergy effects among the different operators and prove the efficiency of the proposed approach.

- Another more recent and quite different metaheuristic is ant colony optimization (ACO) which also has been adapted to a multiobjective framework. Recently in [14], the authors review and classify the different existing MOACO algorithms by proposing a taxonomy and developing a large and systematic experimental study, comparing them with two well-known MOEA algorithms (see chapter 6 of [8]) on several MOTSP instances.
From this paper, it results that MOACO algorithms are very competitive against those two MOEA algorithms. Among their conclusions, the authors propose as future development to add a local search to the MOACO algorithms with the aim to compare their performance with those of MOGLS.

2.2 Local search algorithms

Another class of metaheuristics abundantly applied to MOTSP are sophisticated local search algorithms.

- In [3], Angel *et al.* consider a so-called "dynasearch neighborhood". Given a solution σ and a neighborhood $\mathcal{N}(\sigma)$, the first idea is to determine the set $U\mathcal{N}(\sigma)$ containing all neighboring solutions not dominated by σ. The authors apply this idea to the particular ds-2opt (dynasearch) neighborhood $\mathcal{N}_{dyna}(\sigma)$ which consists to apply a series of independent 2-opt moves: two such moves are independent if the edges involved do not overlap. They use a dynamic programming method to compute $U\mathcal{N}_{dyna}(\sigma)$ in an efficient way.
The neighborhood is integrated in a Pareto local search method, that we call PLS1. The method starts with an initial population P and $U\mathcal{N}_{dyna}(\sigma)$

is determined for each solution $\sigma \in P$. The next population is formed of all the new potentially efficient solutions. With each new solution, $\widehat{X_E}$ is possibly updated. In this algorithm, neighbors are generated from all $\sigma \in P$ even if σ does not belong anymore to $\widehat{X_E}$ (i.e. if σ is dominated by a new neighbor). The initial population is composed of only one random solution. The authors also use a rounding technique to limit the number of potential non-dominated points generated. The results on biobjective instances are of good quality but obtained in a high resolution time.

- The Pareto local search of Paquete *et al.* [45], noted PLS2, is very similar to PLS1. The method starts with a randomly generated solution, and $\widehat{X_E}$ is initialized with this solution. At each iteration a solution σ is chosen randomly in $\widehat{X_E}$ and its neighborhood $\mathcal{N}(\sigma)$ is completely explored. If a non-dominated solution $\sigma' \in \mathcal{N}(\sigma)$ is found, $\widehat{X_E}$ is updated. When all the neighborhood $\mathcal{N}(\sigma)$ has been examined, a new solution σ is chosen in $\widehat{X_E}$. Therefore, contrary to PLS1, in PLS2 the neighbors are never generated from a dominated solution. The method stops when all the solutions σ of $\widehat{X_E}$ have been examined. The results depend on the order in which the solutions of $\widehat{X_E}$ are examined. Three neighborhoods are considered: 2-opt, a so-called 2h-opt (considering in addition of 2-opt, moves of a single city from its position to another inside the tour) and 3-opt. With this last neighborhood, the results are a little bit better than those of MOGLS but at the price of a higher computation time.

- Paquete and Stützle [46] presented three versions of a local search method, all presenting two phases. The methods are called two-phase LS (TPLS), double two-phase LS (P-TPLS) and Pareto double two-phase LS (PD-TPLS). The two phases are as follows. In the first phase, the TPLS method generates a single initial solution by optimizing only the first objective (z_1) with a LS (the weight of z_1 is 1 and the weight of z_2 is equal to zero). In the second phase, the LS is applied with a sequence of different weighted sum a_i forming a chain. The LS with the function a_{i+1} starts with the best solution found at the preceding iteration i, the function a_{i+1} is a slight modification of a_i: the weight of objective z_1 is decremented and the weight of objective z_2 is incremented.
  P-TPLS is simply TPLS applied two times, considering for the first phase respectively z_1 and z_2.
  PD-TPLS tries to determine additional potential non-dominated points which are missed by the aggregation used. For this, a LS that accepts solutions which are not dominated by the current local optimum in both objectives, is applied after a local optimum is found with each function a_i. The neighborhood used is 2-opt exchange.

A comparison of the results obtained shows that they are competitive with those of MOGLS on biobjective instances.

- A quite different approach is proposed by Li [36], based on the notion of "attractor" of each objective. Such attractor contains a set of the best solutions found for the corresponding objective. The number of hits of each edge in all these solutions is recorded. Using this information, the procedure combines these edges to produce potential efficient solutions. Results are provided for randomly generated costs, but no comparison with other algorithms, either assessments of the quality of the approximations is realized.

2.3 Particular studies and variants of the MOTSP

To be as far as possible exhaustive, we mention here other papers related to the MOTSP.

- Gupta and Warburton [17] described a heuristic method to approximate an optimal solution minimizing a weighted Tchebycheff function.

- Tung [50] proposed an exact labeling method to determine $\widehat{X_E}$. Of course, only very small instances can be treated.

- Special models are also treated by different authors:
  - Fisher and Richter [12] considered a biobjective particular TSP, in which the first objective is a classical sum objective but the second is a multiplicative reliability objective, and they proposed a dynamic programming approach.

  - In the biobjective problem treated by Melamed and Sigal [41] the first objective is also a classical sum objective but the second is the maximum of the cost of the edges selected for the tour; their method is based on the linear convolution of the two objective functions.

  - The TSP model treated by Angel *et al.* [2] is called TSP(1,2) because the cost associated to each edge can only take the two values 1 and 2. For a biobjective TSP(1,2), the authors proposed a LS based on 2-opt to obtain an approximation $\widehat{Z_N}$ with a guarantee of performance equal to 3/2.

  - Manthey and Ram [40] also described heuristics with performance guarantees for a MOTSP in the particular case where the costs satisfied the $\gamma-$triangle inequality:

$$c_{ij}^k \leq \gamma(c_{il}^k + c_{lj}^k) \quad \forall\, i,j,l \in \{1,\dots,n\}, \forall\, k \in \{1,\dots,p\}, \gamma \in [\frac{1}{2},1].$$

They also treat the case of MOTSP(1,2).

Some variants of the MOTSP are also considered.

- An interesting variant of the TSP is the TSP with profits. In such a problem, the traveling salesman is not required to visit all cities, but a profit is associated to each visited city. It can thus be modelized as a biobjective problem: it is necessary to find a cyclic permutation over a subset of the n cities such that the collected prize is maximized while the travel cost is minimized. The old paper of Keller and Goodchild [33] already introduced such a model for which a heuristic is proposed. Very recently, Bérubé *et al.* [15] proposed an exact ϵ-constraint method to generate X_E and Jozefowiez *et al.* [31] developed a MOEA combined with an ejection chain local search to approximate X_E.

- Other related models are analyzed: Jozefowiez *et al.* [32] considered multiobjective vehicle routing problems and Huang *et al.* [25] a multiobjective route planning problem.

3 New approach: the Two-Phase Pareto Local Search Method

3.1 Presentation

From the preceding survey about MOMH, we have seen that many methods, using sometimes complex components, have been developed for approximating X_E. We show here that it is however possible to produce very good approximations $\widehat{X_E}$ for different types of instances of the MOTSP, by using simple components coming from state-of-the-art results for the single-objective TSP. The new method, called two-phase Pareto local search (2PPLS) [39], thoroughly makes use of the Lin-Kernighan heuristic [37] and speed-up techniques [28].

The 2PPLS method presents no numerical parameters and is mainly based on the conclusion of Borges and Hansen [6] in their work about the global convexity in MOTSP: efficient solutions of the MOTSP are very close in the decision space and 2-opt moves from efficient solutions allow to generate many other efficient solutions. To take this fact into account, the 2PPLS method is decomposed into two phases: in the first, an approximation of supported efficient solutions is generated and in the second, a local search is used to generate non-supported efficient solutions. The spirit of the two phases of the 2PPLS method is thus similar to that of the exact two-phase method developed by Ulungu and Teghem [56], but here, approximation methods are used in both phases. The adaptation of the two phases in the case of finding approximations to the efficient solutions of the biobjective TSP (bTSP) is as follows:

1. Phase 1: Find a good approximation of the supported efficient solutions. These solutions can be generated by resolution of weighted sum single-objective problems obtained by applying a linear aggregation of the objectives. Only a good approximation of a minimal complete set of the extreme supported efficient solutions is sought. To this aim, we have heuristically adapted the method of Aneja and Nair [1], initially proposed for the resolution of a biobjective transportation problem. The method consists in generating all the weight sets which make it possible to obtain a minimal complete set of extreme supported efficient solutions of a biobjective problem. Each single-objective problem is solved with one of the best heuristics for the single-objective TSP: the Lin-Kernighan heuristic [37]. Two different improvements of the original version of Lin and Kernighan, with a published implementation, are well-known : the version of Helsgaun [23] and the version of Applegate *et al.* [4]. We use the chained Lin-Kernighan version of Applegate since as shown in [39], the results obtained with these two versions are similar but with the version of Applegate *et al.*, no numerical parameters have to be tuned to keep reasonable resolution times.
2. Phase 2: Find non-supported efficient solutions located between the supported efficient solutions. In this phase, we use the Pareto local search (PLS) method, used and developed by different authors [3, 5, 45]. Two different versions of PLS are known depending on how the population is updated: the version of Angel *et al.* [3] and the version of Paquete *et al.* [45] (see section 2.2). We use the version of PLS of Angel *et al.* (PLS1), which has also been used by Basseur [5] as local search in a memetic algorithm. In PLS, we use the well-known 2-opt neighborhood, as suggested by Borges and Hansen in their work about the global convexity in bTSP [6].

It should be noted that 2PPLS is close to the method of Hamacher and Ruhe [19], developed in 1994, to find a well distributed approximation of the efficient solutions of the multiobjective spanning tree problem. However, two differences can be noticed: they use the exact set of extreme efficient solutions in phase 1 and a stop criterion related to the distance between two consecutive solutions is applied in phase 2. This method has only been applied to the multiobjective spanning tree problem.

3.2 Speed-up technique

As generating all the 2-opt moves for each solution of the population will be very time-consuming ($\frac{N(N-3)}{2}$ neighbors to generate for each solution), we will only consider a small proportion of all the possible 2-opt moves. Otherwise, large instances could not be solved in a reasonable time [38]. We show below how to select the 2-opt moves to make.

Data analysis

Before presenting the speed-up technique, let's take a look at the edges used by the solutions of an efficient set. As for biobjective instances, the edges can be represented in a two-dimensional graph (the x-coordinate and y-coordinate of the graph are respectively given by the cost 1 and 2 of the edges), we will employ such representation to study what are the edges of a biobjective instance used by the efficient set.

We have presented in Fig. 1, on the left, all the 4950 edges of a biobjective Euclidean instance of 100 cities (see section 4). On the right, we have represented the edges used by a near-efficient set, which is a very good approximation of the efficient set.

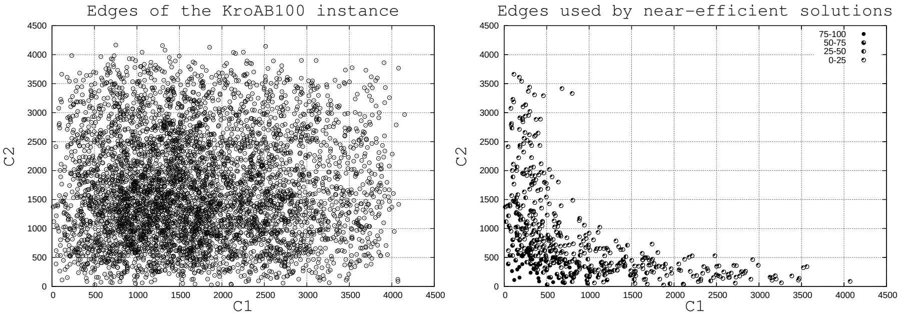


Fig. 1. Edges of the KroAB100 Instance.

It is noted that only a small proportion of the edges are used by the near-efficient set, and the edges that are bad for both costs are not used at all. This observation is important to develop appropriate speed-up techniques. In the right graph, we also add frequencies with which the edges are used by the solutions of the near-efficient set, and we remark that well-located edges (both costs are low) are intensively employed (near to 100% for certain edges, what means that almost all solutions use these edges) while others are only rarely used. But the relation between the location of the edges and the frequencies is not clear and would be difficult to take into account.

Also, after application of the first phase of 2PPLS, a set of potential efficient solutions is already discovered and it would be relevant to retrieve information from this set. Gandibleux *et al.* [13] have also exploited this idea in the case of scheduling and knapsack problems.

We have represented in Fig. 2 all the edges used in at least one solution generated during the first phase and edges used by the near-efficient set. We can see that both sets of candidate edges are very close. It thus seems that using only the edges found after the first phase for the 2-opt moves will already give good results.

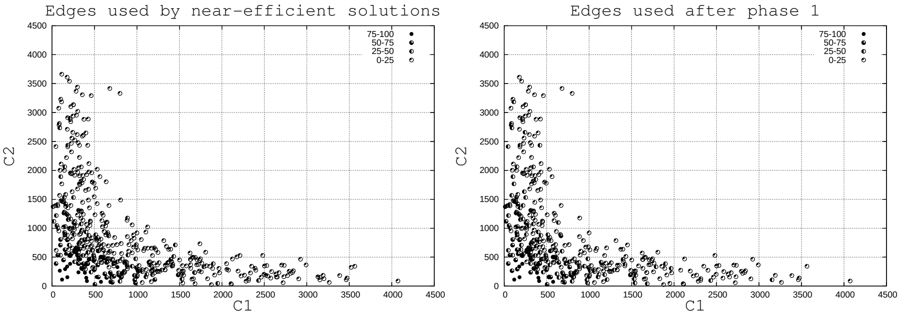


Fig. 2. Edges used by a near-efficient set and by the set obtained after the first phase (KroAB100 instance).

2-opt moves with candidate lists

We will explain now how to take into account the reduced set of edges used by the efficient solutions.

We have represented in Fig. 3 a 2-opt move, where (t_1, t_2) and (t_3, t_4) are the leaving edges and (t_1, t_4) and (t_2, t_3) are the entering edges. For the leaving edge (t_1, t_2), if we intend to generate all the 2-opt moves, they are $(n - 3)$ possibilities for the city t_3.

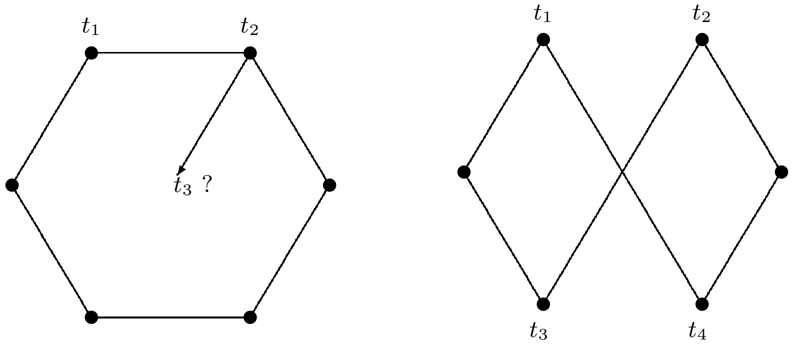


Fig. 3. 2-exchange move.

In order to limit the number of possibilities for t_3, a classic speed-up technique for solving the TSP is the candidate list. For a starting city t_1, with t_2 the next city in the tour, to consider candidates for t_3, we only need to start at the beginning of the candidate list of t_2 and proceed down it until the end of the list has been reached. The size of the list is limited to a reasonable size, compromise between quality and running time. For example, in [28], Johnson and McGeoch recommend a size equal to 15.

In the single-objective case, the candidate lists are created on the basis of the single cost. In the biobjective case, as there is no more a total order between the cost vectors c, the candidate lists are created on the basis of the edges used by the solutions found during the phase 1 of the 2PPLS method. To do that, we explore the set of candidate edges of phase 1, and for each candidate edge $\{v_i, v_j\}$, we add the city v_j to the candidate list of the city v_i, if v_i is not already in the candidate list of v_j (to avoid to realize the same move twice).

All the cities are considered as starting cities for the 2-opt moves.

4 Results

We test 2PPLS with the speed-up technique on several biobjective TSP instances. We consider four types of instances:

- Euclidean instances: the costs between the edges correspond to the Euclidean distance between two points in a plane.
- Random instances: the costs between the edges are randomly generated from a uniform distribution.
- Mixed instances: the first cost corresponds to the Euclidean distance between two points in a plane and the second cost is randomly generated from a uniform distribution.
- Clustered instances: the points are randomly clustered in a plane, and the costs between the edges correspond to the Euclidean distance.

In this work, we use the following biobjective symmetric instances:

- Euclidean instances: three Euclidean instances of respectively 100, 300 and 500 cities called EuclAB100, EuclAB300 and EuclAB500.
- Random instances: three random instances of respectively 100, 300 and 500 cities called RdAB100, RdAB300 and RdAB500.
- Mixed instances: three mixed instances of respectively 100, 300 and 500 cities called MixedAB100, MixedAB300 and MixedAB500.
- Clustered instances: three clustered instances of respectively 100, 300 and 500 cities called ClusteredAB100, ClusteredAB300 and ClusteredAB500.

We have generated ourselves the clustered instances with the generator available from the 8th DIMACS Implementation Challenge site[2] with the following property: the number of clusters is equal to the number of cities divided by 25 and the maximal coordinate for a city is equal to 3163 (as done by Paquete for the Euclidean instances). The other instances have been generated and published by Paquete [44]. Paquete and Stützle [47] have very recently published the results of the comparison of their PD-TPLS method to MOGLS on these instances and they show that they obtain better results on most of the instances.

[2] http://www.research.att.com/~dsj/chtsp/download.html

All the algorithms tested in this work have been run on a Pentium IV with 3 Ghz CPUs and 512 MB of RAM. We present the average of the indicators on 20 executions. The resolution time of our implementation of the algorithms corresponds to the wall clock time.

To compute the distances D_1 and D_2 (see section 1), reference sets based on the notion of ideal set [39] have been generated for all the instances experimented. The ideal set is defined as the best potential Pareto front that can be produced from the extreme supported non-dominated points. This is a lower bound of the Pareto front [9]. For generating the extreme supported non-dominated points, we have used the method proposed by Przybylski *et al.* [48]. However, for the instances of more than 200 cities, numerical problems were encountered. Thus, for these instances, we have generated the extreme supported non-dominated points of the biobjective minimum spanning tree problem (bMST) associated to the same data than the bTSP. The ideal set is then produced on the basis of the extreme supported non-dominated points of the bMST. As the biobjective minimum spanning tree problem is a relaxation of the bTSP, all feasible solutions of the bTSP remain dominated or equivalent to the solutions of the ideal set of the bMST.

For the computation of the R and $\mathcal{H}$ indicators, the reference points are determined according to the reference sets. For the R indicator, the number of weight sets used is equal to 101 for all instances. This indicator has been normalized between 0 and 1, where 1 corresponds to the best value.

As state-of-the-art results are not known for all the instances used in this work, we have implemented ourselves the PD-TPLS method of Paquete and Stützle [46], a method presenting few parameters contrarily to MOEA algorithms. This allows to produce a comparison as fair as possible with the PD-TPLS method, since 2PPLS and PD-TPLS are run on the same computer and share the same data structures. However, our implementation of PD-TPLS is presumable not as good as the original implementation and the execution time has to be seen as an indicator but not as an objective comparison factor. For this implementation, we have fixed the parameters of this method as follows, as done in [46]:

- The number of iterations for the number of perturbation steps of the iterated local search method used in PD-TPLS is equal to the number of cities n minus 50.
- The number of aggregations is equal to n, except for the 100 cities instances where the number of aggregations is higher than n in order to obtain comparable running times between 2PPLS and PD-TPLS. The number of aggregations is equal to 250 for the 100 cities instances, excepted for the ClusterredAB100 instance where this number has been set to 400.

4.1 Comparison based on the mean of indicators

We use five different indicators to measure the quality of the approximations obtained: the $\mathcal{H}$, $I_{\epsilon 1}$, R, D_1 and D_2 indicators (see section 1). We also add as additional information the number $|PE|$ of potential efficient solutions generated and the running time in seconds.

We can see the results of the comparison in Table 1.

Table 1. Comparison with PD-TPLS.

Instance	Algorithm	$\mathcal{H}(10^8)$	$I_{\epsilon 1}$	R	D_1	D_2	$\|PE\|$	Time(s)
EuclAB100	2PPLS	**217.62**	**1.007141**	**0.930081**	**0.345**	5.091	**1196.90**	**25.53**
	PD-TPLS	217.55	1.013423	0.930009	0.460	**4.072**	811.35	29.47
EuclAB300	2PPLS	**2309.67**	**1.124524**	**0.941701**	**16.868**	**21.633**	**14050.90**	258.56
	PD-TPLS	2308.97	1.124950	0.941598	17.020	21.921	4415.40	**255.13**
EuclAB500	2PPLS	**7165.35**	1.129026	**0.948995**	**17.247**	22.072	**33017.30**	**692.88**
	PD-TPLS	7163.87	**1.128899**	0.948927	17.395	**21.829**	9306.75	1056.43
RdAB100	2PPLS	**529.77**	**1.025300**	**0.953339**	**0.932**	**14.325**	**438.30**	**25.46**
	PD-TPLS	528.92	1.042416	0.953061	1.195	30.620	419.40	30.89
RdAB300	2PPLS	**4804.48**	**1.728586**	**0.966925**	**20.910**	**36.424**	**1766.95**	**305.36**
	PD-TPLS	4790.35	1.893817	0.966152	22.848	60.815	1238.70	305.41
RdAB500	2PPLS	**13987.92**	**1.790290**	**0.973697**	**19.502**	**55.098**	**3127.85**	**816.40**
	PD-TPLS	13951.01	2.057356	0.972940	21.954	95.618	2081.55	1324.55
MixedAB100	2PPLS	**331.96**	**1.017122**	**0.936782**	**0.579**	**6.006**	**793.10**	**26.24**
	PD-TPLS	331.74	1.032013	0.936741	0.659	12.754	585.10	30.15
MixedAB300	2PPLS	**3410.40**	**1.778747**	**0.956492**	**20.135**	**28.422**	**5202.05**	**238.45**
	PD-TPLS	3406.12	1.943372	0.956254	20.722	37.980	2299.80	288.04
MixedAB500	2PPLS	**10440.35**	**1.710601**	**0.960652**	**22.858**	**31.865**	**12925.40**	**865.55**
	PD-TPLS	10429.11	1.907713	0.960452	23.517	35.546	4816.00	1303.07
ClusteredAB100	2PPLS	**267.28**	**1.007686**	**0.949999**	**0.274**	10.426	**2184.05**	**50.02**
	PD-TPLS	267.21	1.019305	0.949887	0.442	**6.015**	989.45	53.56
ClusteredAB300	2PPLS	2565.34	1.243181	**0.956617**	17.763	24.363	**15511.10**	366.25
	PD-TPLS	**2566.38**	**1.232286**	0.956616	**17.702**	**21.917**	4540.15	**293.72**
ClusteredAB500	2PPLS	8516.35	1.183524	**0.962412**	**16.974**	28.404	**40100.60**	1517.68
	PD-TPLS	**8516.71**	**1.181800**	0.962394	**16.974**	**23.230**	9678.15	**1239.56**

We remark that the 2PPLS method finds better results than the PD-TPLS method on all the indicators for the random instances and mixed instances. For the Euclidean instances, 2PPLS finds better results on all the indicators only for the EuclAB300 instance. For the EuclAB100 instance, PD-TPLS is better on the D_2 indicator, and for the EuclAB500 instance, PD-TPLS is better on the $I_{\epsilon 1}$ and D_2 indicators. For the ClusteredAB100 instance, 2PPLS finds better results except on the D_2 indicator. For the ClusteredAB300 instance, PD-TPLS finds better results except on the R indicator. For the ClusteredAB500 instance, 2PPLS finds better or equivalent results on the R and D_1 indicators while PD-TPLS finds better results on the $\mathcal{H}$, $I_{\epsilon 1}$ and D_2 indicators. The 2PPLS method always generates more potential efficient solutions than PD-TPLS. The running time of PD-TPLS is higher than the running time of

2PPLS, except on the EuclAB300, ClusteredAB300 and ClusteredAB500 instances where the running time of PD-TPLS is slightly lower. The PD-TPLS method appears thus more performing than 2PPLS on the clustered instances.

4.2 Mann-Whitney test

To take into account the variations in the results of the algorithms, as we do not know the distributions of the indicators, we carried out the non-parametric statistical test of Mann-Whitney [11]. This test allows to compare the distributions of the indicators of 2PPLS with these of PD-TPLS.

We test the following hypothesis: "the two samples come from identical populations" for the $\mathcal{H}$, $I_{\epsilon 1}$, R, D_1 or D_2 indicators on a given instance. When the hypothesis is satisfied, the result "=" is indicated (no differences between the indicators of the algorithms). When the hypothesis is not satisfied, there are differences between the indicators of the algorithms: the sign ">" indicates that the mean value obtained with 2PPLS is better than the mean value obtained with PD-TPLS and the sign "<" indicates that the mean value obtained with 2PPLS is worse than the mean value obtained with PD-TPLS.

As five hypothesis are tested simultaneously, the levels of risk α of the tests have been adjusted with the Holm sequential rejective method [24]. This method works as follows: the p-values obtained with the Mann-Whitney test for each indicator are sorted by increasing value and the smallest p-value is compared to α/k (where k is the number of indicators, equal to five in our case). If that p-value is less than α/k, the hypothesis related to this p-value is rejected. The second smallest p-value is then compared to $\alpha/(k-1)$, the third to $\alpha/(k-2)$, etc. The procedure proceeds until each hypothesis has been rejected or when a hypothesis cannot be rejected. The starting level of risk of the test has been fixed to 1%.

The results of the comparison of 2PPLS with PD-TPLS are given in Table 2. The results show that we can affirm with a very low risk that, for the indicators considered in this work, 2PPLS is better or equal to PD-TPLS on all the instances tested in this work, except for the ClusteredAB100 instance (on D_2), for the ClusteredAB300 instance (on $\mathcal{H}$, $I_{\epsilon 1}$ and D_1) and for the ClusteredAB500 instance (on $\mathcal{H}$ and $I_{\epsilon 1}$).

4.3 Outperformance relations

We now compare the solutions obtained with 2PPLS and PD-TPLS in term of outperformance relations [21].

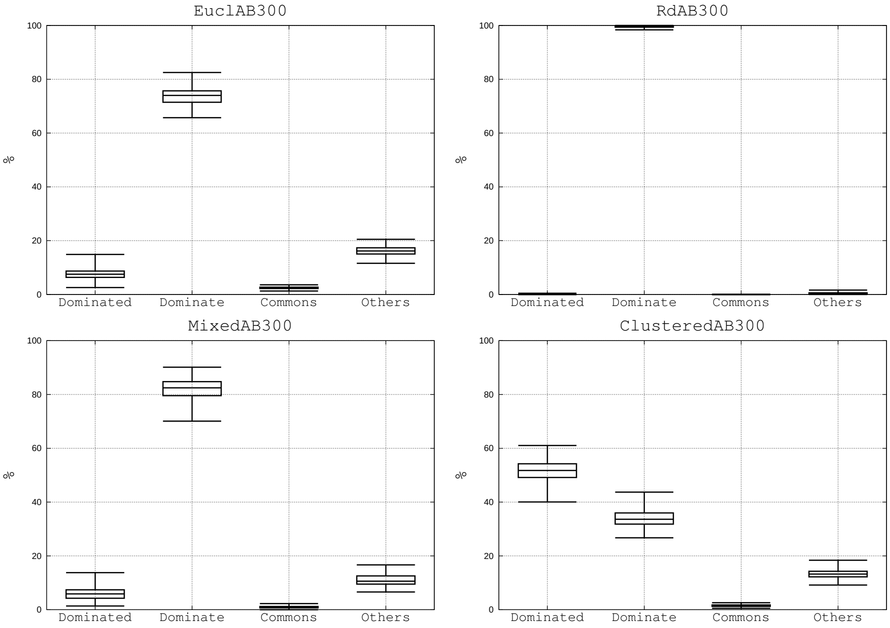
We show in Fig. 4 the results of the comparison between the potential non-dominated points obtained with 2PPLS and by PD-TPLS, for the 300 cities instances.

To create these box-plot graphs, we compare the points obtained with the 2PPLS method to the points obtained with the PD-TPLS method. Four cases can occur: a point of 2PPLS is dominated by at least one point of

Table 2. Results of the Mann-Whitney test for the D_1 and $I_{\epsilon 1}$ indicators.

Instance	$\mathcal{H}$	$I_{\epsilon 1}$	R	D_1	D_2
EuclAB100	$>$	$>$	$>$	$>$	$=$
EuclAB300	$>$	$>$	$>$	$>$	$>$
EuclAB500	$>$	$=$	$>$	$>$	$>$
RdAB100	$>$	$>$	$>$	$>$	$>$
RdAB300	$>$	$>$	$>$	$>$	$>$
RdAB500	$>$	$>$	$>$	$>$	$>$
MixedAB100	$>$	$>$	$>$	$>$	$>$
MixedAB300	$>$	$>$	$>$	$>$	$>$
MixedAB500	$>$	$>$	$>$	$>$	$>$
ClusteredAB100	$>$	$>$	$>$	$>$	$<$
ClusteredAB300	$<$	$<$	$=$	$<$	$=$
ClusteredAB500	$<$	$<$	$>$	$=$	$=$

PD-TPLS (`Dominated`), a point of 2PPLS dominates at least one point of PD-TPLS (`Dominate`), a point of 2PPLS is equal to another point of PD-TPLS (`Commons`), or the result of the comparison belongs to none of these three possibilities (`Others`).



EuclAB300

RdAB300

MixedAB300

ClusteredAB300

Fig. 4. Comparison between the potentially non-dominated points obtained with 2PPLS and with PD-TPLS.

These box-plots show that many solutions obtained with the 2PPLS method dominate solutions of PD-TPLS, for the EuclAB300, RdAB300 and MixedAB300 instances. However, on the ClusteredAB300 instance, there are more solutions of 2PPLS that are dominated than solutions that dominate solutions of PD-TPLS; nevertheless the difference remains low.

4.4 Summary

The previous results show that 2PPLS gives better results for the indicators considered than PD-TPLS, except on the clustered instances. In 2PPLS, aggregations are first realized to generate a good initial population, by using the method of Aneja and Nair and solving the weighted single-objective problems with a powerful implementation of the Lin-Kernighan heuristic. The initial population is then improved by a Pareto local method using a 2-opt neighborhood with candidate lists. In PD-TPLS, a higher number of weighted single-objective problem are solved, with the iterated local search, and the solutions found are improved with a 2-opt neighborhood. It appears thus more efficient for the Euclidean, Random and Mixed instances, to use an efficient number of weighted single-objective problems (determined by the Aneja and Nair method) and to improve the solutions with a full PLS (and not only one iteration of PLS as in done in PD-TPLS). However, to keep reasonable resolution times, speed-up techniques based on candidate lists are necessary. On the clustered instances, to improve 2PPLS, we have already tried to use the iterated local search in the place of the Lin-Kernighan heuristic for the generation of the initial population, or also to explore completely the neighborhood (no speed-up techniques), but the results remained worse than PD-TPLS, for some of the indicators considered. Till now, it seems thus that for these particular instances, it is better to solve a high number of weighted single-objective problems, before applying the Pareto local search, which appears not as efficient than for the other types of instances. Nevertheless, other ways to improve 2PPLS for this type of instances will be considered in the future.

5 Conclusion

In this work, we have shown that the MOTSP is a very challenging problem, presenting many particularities and variants. Many methods have been developed for tackling this problem, but they are often quite complicated and require many parameters. In this work, we have thus introduced the 2PPLS method, a simple method with no numerical parameters, taking benefit from an efficient single-objective solver and from information contained in potential supported efficient solutions. With this method, we have obtained state-of-the-art results for different types of instances of the biobjective TSP, except on the particular clustered instances where PD-TPLS is more performing.

Despite of these good results, the MOTSP still needs further study: very few results are known for the TSP with more than two objectives (at the exception of some results for the three objective case), and no efficient exact method has been developed for solving this problem, even in the biobjective case. Moreover, the instances size for which approximations have been generated with a reasonable running time are still small compared to the size of the single-objective instances that we can approximate.

References

1. Y.P. Aneja and K.P.K. Nair. Bicriteria transportation problem. *Management Science*, 25:73–78, 1979.
2. E. Angel, E. Bampis, and L. Gourves. Approximating the Pareto curve with local search for the bicriteria TSP(1,2) problem. *Theoritical Computer Science*, 310:135–146, 2004.
3. E. Angel, E. Bampis, and L. Gourvès. A dynasearch neighborhood for the bicriteria traveling salesman problem. In X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, pages 153–176. Springer. Lecture Notes in Economics and Mathematical Systems Vol. 535, Berlin, 2004.
4. D. Applegate. Chained Lin-Kernighan for large traveling salesman problems. *INFORMS Journal on Computing*, 15:82–92, 2003.
5. M. Basseur. Design of cooperative algorithms for multi-objective optimization: application to the flow-shop scheduling problem. *4OR*, 4(3):255–258, 2006.
6. 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, 2000.
7. 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.
8. M. Ehrgott and X. Gandibleux. Multiobjective Combinatorial Optimization—Theory, Methodology, and Applications. In Matthias Ehrgott and Xavier Gandibleux, editors, *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*, pages 369–444. Kluwer Academic Publishers, Boston, 2002.
9. M. Ehrgott and X. Gandibleux. Bound sets for biobjective combinatorial optimization problems. *Computers and Operations Research*, 34:2674–2694, 2007.
10. S. Elaoud, J. Teghem, and T. Loukil. Multiple crossover genetic algorithm for the multiobjective traveling salesman problem: University of Sfax (tunisia). *Submitted for publication*, 2008.
11. T.S. Ferguson. *Mathematical Statistics, a decision theoretic approach.* Academic Press, New-York and London, 1967.
12. R. Fisher and K. Richter. Solving a multiobjective traveling salesman problem by dynamic programming. *Mathematische Operationsforschung und Statistik, Series Optimization*, 13(2):247–252, 1982.
13. X. Gandibleux, H. Morita, and N. Katoh. The supported solutions used as a genetic information in a population heuristics. In Eckart Zitzler, Kalyanmoy Deb,

Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, volume 1993 of *Lecture Notes in Computer Science*, pages 429–442. Springer, 2001.

14. C. Garcia-Martinez, O. Cordon, and F. Herrera. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research*, 180(1):116 – 148, June 2007.
15. M. Gendreau, J-F. Bérubé, and J-Y. Potvin. An exact epsilon-constraint method for bi-objective combinatorial optimization problems: Application to the traveling salesman problem with profits. *European Journal of Operational Research*, 194(1):39–50, 2009.
16. F. Glover and G. Kochenberger. *Handbook of Metaheuristics.* Kluwer, Boston, 2003.
17. A. Gupta and A. Warburton. Approximation methods for multiple criteria traveling salesman problems, towards interactive and intelligent decision support systems. In Y. Sawaragi, editor, *Proc. 7th Internat. Conf. on Multiple Criteria Decision Making*, pages 211–217, Berlin, 1986. Springer.
18. G. Gutin and A. Punnen. *The Traveling Salesman Problem and its Variations.* Kluwer, Dordrecht / Boston / London, 2002.
19. H.W. Hamacher and G. Ruhe. On spanning tree problems with multiple objectives. *Annals of Operations Research*, 52:209–230, 1994.
20. M.P. Hansen. Use of Substitute Scalarizing Functions to Guide a Local Search Based Heuristic: The Case of moTSP. *Journal of Heuristics*, 6(3):419–430, 2000.
21. M.P. Hansen and A. Jaszkiewicz. Evaluating the quality of approximations of the nondominated set. Technical report, Technical University of Denmark, Lingby, Denmark, 1998.
22. P. Hansen. Bicriterion path problems. *Lecture Notes in Economics and Mathematical Systems*, 177:109–127, 1979.
23. K. Helsgaun. An effective implementation of the lin-kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126:106–130, 2000.
24. S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.
25. B. Huang, L. Yao, and K. Raguraman. Bi-level GA and GIS for multi-objective TSP route planning. *Transportation Planning and Technology*, 29(2):105–124, April 2006.
26. A. Jaszkiewicz. On the Performance of Multiple-Objective Genetic Local Search on the 0/1 Knapsack Problem—A Comparative Experiment. *IEEE Transactions on Evolutionary Computation*, 6(4):402–412, August 2002.
27. A. Jaszkiewicz and P. Zielniewicz. Pareto memetic algorithm with path-relinking for biobjective traveling salesman problem. *European Journal of Operational Research*, 193(3):885–890, 2009.
28. D.S. Johnson and L.A. McGeoch. The Traveling Salesman Problem: A Case Study in Local Optimization. In E. H. L. Aarts and J.K. Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 215–310. John Wiley and Sons Ltd, 1997.
29. D.S. Johnson and L.A. McGeoch. Experimental analysis of heuristics for the ATSP. In G. Gutin and A. Punnen, editors, *The Traveling Salesman Problem and its Variations.* Kluwer, 2002.

30. D.S. Johnson and L.A. McGeoch. Experimental analysis of heuristics for the STSP. In G. Gutin and A. Punnen, editors, *The Traveling Salesman Problem and its Variations*. Kluwer, 2002.
31. N. Jozefowiez, F. Glover, and M. Laguna. Multi-objective meta-heuristics for the traveling salesman problem with profits. *Journal of Mathematical Modelling and Algorithms*, 7(2):177–195, 2008.
32. N. Jozefowiez, F. Semet, and E-G. Talbi. Multi-objective vehicle routing problems. *European Journal of European Research*, 189(2):293–309, 2008.
33. C.P. Keller and M. Goodchild. The multiobjective vending problem: A generalization of the traveling salesman problem. *Environment and Planning B: Planning and Design*, 15:447–460, 1988.
34. R. Kumar and P.K. Singh. Pareto evolutionary algorithm hybridized with local search for biobjective TSP. In C. Grosan, A. Abraham, and H. Ishibuchi, editors, *Hybrid Evolutionary Algorithms*, chapter 14. Springer-Verlag, 2007.
35. P. Larranaga, C.M.H. Kuijpers, R.H. Murga, I. Inza, and S. Dizdarevic. Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13(2):129–170, 1999.
36. W. Li. Finding Pareto-optimal set by merging attractors for a bi-objective traveling salesmen problem. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, volume 3410 of *Lecture Notes in Computer Science*, pages 797–810. Springer, 2005.
37. S. Lin and B.W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21:498–516, 1973.
38. T. Lust and A. Jaszkiewicz. Speed-up techniques for solving large-scale biobjective TSP. *To appear in Computers and Operations Research*, 2009.
39. T. Lust and J. Teghem. Two-phase Pareto local search for the biobjective traveling salesman problem. *To appear in Journal of Heuristics*, 2009.
40. B. Manthey and L. Shankar Ram. Approximation algorithms for multi-criteria traveling salesman problems. *CoRR*, arxiv:abs/cs/0606040, 2007.
41. I.I. Melamed and K.I. Sigal. The linear convolution of criteria in the bicriteria traveling salesman problem. *Computational Mathematics and Mathematical Physics*, 37(8):902–905, 1997.
42. P. Merz and B. Freisleben. Memetic algorithms for the traveling salesman problem. *Complex Systems*, 13:297–345, 2001.
43. K. Miettinen. *Nonlinear multiobjective optimization*. Kluwer, Boston, 1999.
44. L. Paquete. *Stochastic Local Search Algorithms for Multiobjective Combinatorial Optimization: Methods and Analysis*. PhD thesis, FB Informatik, TU Darmstadt, 2005.
45. L. Paquete, M. Chiarandini, and T. Stützle. Pareto Local Optimum Sets in the Biobjective Traveling Salesman Problem: An Experimental Study. In X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, pages 177–199, Berlin, 2004. Springer. Lecture Notes in Economics and Mathematical Systems Vol. 535.
46. L. Paquete and T. Stützle. A Two-Phase Local Search for the Biobjective Traveling Salesman Problem. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 479–493, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science Vol. 2632.

47. L. Paquete and T. Stützle. Design and analysis of stochastic local search for the multiobjective traveling salesman problem. *Computers and Operations Research*, 36(9):2619–2631, 2009.
48. A. Przybylski, X. Gandibleux, and M. Ehrgott. Two-phase algorithms for the biobjective assignement problem. *European Journal of Operational Research*, 185(2):509–533, 2008.
49. G. Reinelt. Tsplib - a traveling salesman problem library. *ORSA Journal of Computing*, 3(4):376–384, 1991.
50. F. Samanlioglu, W.G. Ferrell Jr., and M.E. Kurz. A multicriteria Pareto-optimal algorithm for the traveling salesman problem. *Asia-Pacific Journal of Operational Research*, 11:103–115, 1994.
51. F. Samanlioglu, W.G. Ferrell Jr., and M.E. Kurz. A hybrid random-key genetic algorithm for a symmetric travelling salesman problem. *Computers and Industrial Engineering*, 55(2):439–449, 2008.
52. R. Steuer. *Multiple Criteria Optimization: Theory, Computation and Applications.* John Wiley & Sons, New-York, 1986.
53. J. Teghem. La programmation linéaire multicritère. In D. Dubois and M. Pirlot, editors, *Concepts et méthodes pour l'aide à la décision*, pages 215–288. Hermès, 2006.
54. J. Teghem and P. Kunsch. A survey of techniques for finding efficient solutions to multi-objective integer linear programming. *Asia-Pacific Journal of Operational Research*, 3(2):95–108, 1986.
55. E.L. Ulungu and J. Teghem. Multiobjective combinatorial optimization problems: A survey. *Journal of Multi-Criteria Decision Analysis*, 3:83–104, 1994.
56. E.L. Ulungu and J. Teghem. The two phases method: An efficient procedure to solve biobjective combinatorial optimization problems. *Foundation of Computing and Decision Science*, 20:149–156, 1995.
57. E.L. Ulungu, J. Teghem, Ph. Fortemps, and D. Tuyttens. MOSA Method: A Tool for Solving Multiobjective Combinatorial Optimization Problems. *Journal of Multi-Criteria Decision Analysis*, 8(4):221–236, 1999.
58. Z. Yan, L. Zhang, L. Kang, and G. Lin. A new MOEA for multiobjective TSP and its convergence property analysis. In C.M. Fonseca, P.J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *EMO*, volume 2632 of *Lecture Notes in Computer Science*, pages 342–354. Springer, 2003.
59. E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications.* PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.
60. E. Zitzler, M. Laumanns, L. Thiele, C.M. Fonseca, and V. Grunert da Fonseca. Why Quality Assessment of Multiobjective Optimizers Is Difficult. In W.B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M.A. Potter, A.C. Schultz, J.F. Miller, E. Burke, and N. Jonoska, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002)*, pages 666–673, San Francisco, California, July 2002. Morgan Kaufmann Publishers.
61. 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, April 2003.