INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 00 (2012) 1–26
DOI: 10.1111/j.1475-3995.2011.00840.x

The multiobjective multidimensional knapsack problem: a survey and a new approach

Thibaut Lust and Jacques Teghem

Laboratory of Mathematics and Operational Research, Faculté Polytechnique de Mons, University of Mons, place du parc 20, 7000 Mons, Belgium
E-mail: lust.thibaut@gmail.com [Lust]; jacques.teghem@umons.ac.be [Teghem]

Received 26 November 2010; received in revised form 3 October 2011; accepted 5 November 2011

Abstract

The knapsack problem (KP) and its multidimensional version (MKP) are basic problems in combinatorial optimization. In this paper, we consider their multiobjective extension (MOKP and MOMKP), for which the aim is to obtain or approximate the set of efficient solutions. In the first step, we classify and briefly describe the existing works that are essentially based on the use of metaheuristics. In the second step, we propose the adaptation of the two-phase Pareto local search (2PPLS) to the resolution of the MOMKP. With this aim, we use a very large scale neighborhood in the second phase of the method, that is the PLS. We compare our results with state-of-the-art results and show that the results we obtained were never reached before by heuristics for biobjective instances. Finally, we consider the extension to three-objective instances.

Keywords: multiple objective programming; knapsack problems; metaheuristics; local search

1. Introduction

The multiobjective multidimensional knapsack problem (MKP) is a particular case of multiobjective linear integer programming (MOILP) of the form

$$\begin{aligned} &\text{“max”} \quad z(x) = z_k(x) = c^k x \quad k = 1, \ldots, p \quad \text{(MOILP)} \\ &\text{s.t} \quad x \in X = D \cap \mathbb{Z}_p^n. \end{aligned}$$

We first recall some basic definitions for any type of multiobjective optimization problems (MOPs) (Miettinen, 1999; Steuer, 1986). We consider that all the objectives have to be maximized.

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

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

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

Although other approaches exist to tackle an MOP problem (see Teghem, 2009), 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 paper, only an approximation of a minimal complete set (Hansen, 1979) is determined: no equivalent solution generated will be thus retained. In the case of an MOILP, we can distinguish between two types of efficient solutions.

- The supported efficient solutions are optimal solutions of weighted single-objective problems

$$\begin{aligned} \max \quad & \sum_{k=1}^{p} \lambda_k z_k(x) \\ \text{s.t} \quad & x \in X, \end{aligned}$$

  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 nondominated points in $\mathbb{R}^p$. The points of Z_{SN} are located on the frontier of the convex hull of Z.
- As opposed to a multiobjective linear programming problem, Z_{SN} is generally a proper subset of Z_N due to the nonconvex character of Z: there exist efficient solutions that are nonsupported. We denote by $X_{NE} = X_E \backslash X_{SE}$ and $Z_{NN} = Z_N \backslash Z_{SN}$, respectively, the set of nonsupported efficient solutions and the set of the corresponding nondominated points in $\mathbb{R}^p$.

The two main difficulties to overcome while generating X_E for MOILP problems (Teghem and Kunsch, 1986) are as follows:

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

In this paper various multiobjective *combinatorial* optimization (MOCO) problems have been considered. MOCO problems are particular cases of MOILP in which the variables are binary and D is a specific polytope characterizing the particular CO problem.

During the last 15 years, there has been a significant increase in the number of studies on MOCO problems. From the first survey (Ulungu and Teghem, 1994) in 1994 until 2002 (Ehrgott and Gandibleux, 2002), a large number of papers have been published and this flow is still continuing. The main reason for this phenomenon is the success of the metaheuristics (Glover and Kochenberger, 2003).

Effectively, it is quite difficult to determine exactly the sets X_E and Z_N for MOCO problems, which are $\mathcal{NP}$-hard. Therefore, there exist only few exact methods to determine the sets X_E and Z_N and we can apply these methods to small instances only. For this reason, many methods are heuristic in nature and produce approximations $\widetilde{X}_E$ and $\widetilde{Z}_N$ to the sets X_E and Z_N. Due to the success

of metaheuristics for single-objective CO, multiobjective metaheuristics (MOMHs) fast became a classic tool to tackle MOCO problems and presently this is a real challenge for the researchers to improve the results previously obtained.

The two main difficulties of MOMHs are related to the basic needs of any metaheuristics (Glover and Kochenberger, 2003).

- To assure sufficient intensity, that is to produce an approximation $\widetilde{Z}_N$ as close as possible to Z_N.
- To assure sufficient diversity, that is to cover all the parts of Z_N with $\widetilde{Z}_N$.

Nevertheless 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 an MOMH is in fact a multicriteria problem. Consequently, several indicators have been introduced in the literature to measure the quality of an approximation (see Zitzler et al., 2002 for instance).

Some of them are unary indicators.

- The hypervolume $\mathcal{H}$ (to be maximized) (Zitzler, 1999): The volume of the dominated space defined by $\widetilde{Z}_N$, limited by a reference point.
- The R measure (to be minimized) (Jaszkiewicz, 2002): The evaluation of $\widetilde{Z}_N$ 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) (Czyzak and Jaszkiewicz, 1998; Ulungu et al., 1999) between the points of a reference set and the points of $\widetilde{Z}_N$, by using the Euclidean distance. Ideally, the reference set is Z_N itself, but generally it is not available; otherwise, it can be the nondominated points existing among the union of various sets $\widetilde{Z}_N$ generated by several methods, or an upper bound of Z_N (Ehrgott and Gandibleux, 2007).
- The ϵ factor $I_{\epsilon 1}$ (to be minimized) by which an approximation A is worse than a reference set B with respect to all objectives:

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

- The proportion P_{Z_N} (to be maximized) of nondominated points generated.

Unfortunately, none of these indicators conclude that one approximation is better than the other (see Zitzler et al., 2003 for details). However, an approximation that finds better values for these indicators is generally preferred to others.

In the next section, we survey the existing literature on the multiobjective KP (MOKP) and then on its multidimensional version (MOMKP). We also present the new heuristic approach in Section 3, and the data used for this study in Section 4. In Section 5, we provide the results.

2. The multiobjective knapsack literature

The single-objective KP is certainly one of the most studied $\mathcal{NP}$-hard CO problem (Kellerer et al., 2004; Martello and Toth, 1990). In the book by Martello and Toth (1990) and recently by Kellerer et al. (2004), various methods—essentially branch and bound and dynamic programming (DP)

approaches—are analyzed, for the KP and MKP and for some of its variants; in Kellerer et al. (2004), a chapter is devoted to approximation algorithms for the KP and another chapter presents different heuristics for the MKP.

We recall the formulation of the multiobjective MKP (MOMKP): given n items ($i = 1, \ldots, n$) having m characteristics w^i_j ($j = 1, \ldots, m$)—such as weight, volume, etc.—and p profits c^i_k ($k = 1, \ldots, p$), some items should be selected to maximize the p total profits while not exceeding the m knapsack capacities W_j regarding the different characteristics.

The MOMKP problem is formulated as follows:

$$\begin{aligned} \text{“max”}\ z_k(x) &= \sum_{i=1}^{n} c^i_k x_i \qquad k = 1, \ldots, p \\ \text{subject to}\ \sum_{i=1}^{n} w^i_j x_i &\le W_j \qquad j = 1, \ldots, m \\ x_i &\in \{0, 1\} \qquad i = 1, \ldots, n, \end{aligned}$$

where $x_i = 1$ means that the item i is selected to be in the knapsack. It is assumed that all coefficients c^i_k, w^i_j, and W_j are nonnegative.

The majority of the instances treated in the following cited papers concern the biobjective case ($p = 2$), sometimes the three-objective case ($p = 3$). The particular case of a single constraint ($m = 1$) corresponds to the MOKP.

We first recall a simple and important result for the single-objective KP ($m = p = 1$), following Dantzig (1957). If the items are ordered by nonincreasing efficiencies c^i/w^i, an optimal solution $\widehat{x}$ of the linear relaxation of the KP is given by

$$\widehat{x}_i = \begin{cases} 1 & \text{if } i < s \\ \frac{W - \sum_{j=1}^{s-1} w^j}{w^s} & \text{if } i = s \\ 0 & \text{if } i > s, \end{cases}$$

where the “split” item s is defined by:

$$\sum_{i=1}^{s-1} w^i \le W < \sum_{i=1}^{s} w^i.$$

Balas and Zemel (1980) noticed that the optimal solution for random instances of the KP was very close to the optimal solution of the linear relaxation of the KP. The notion of *core* has thus been defined and the most efficient algorithms for solving the single-objective KP are based on this concept. We recall that a core $\mathcal{C}$ is a subset of items defined around the split item, that is

$$C = \{i \,|\, n_1 \le i \le n_2\} \text{ with } n_1 < s < n_2.$$

Only the variables x_i, $i \in \mathcal{C}$ are considered, the others are fixed, respectively, to 1 if $i < n_1$ and to 0 if $i > n_2$.

2.1. The MOKP

We first survey the literature concerning the MOKP ($m = 1$). We decompose the presentation of the methods into three groups (Sections 2.1.1–2.1.3): exact methods, approximation algorithms, and heuristic methods based on metaheuristics. We end this section by reviewing particular studies (Section 2.1.4). Note that it is impossible in this survey to present the basic elements of the metaheuristics cited in this section; for a description of these metaheuristics, the reader can refer to Glover and Kochenberger (2003) or Talbi (2009), also the same can be referred for the presentation of the numerous adaptations of metaheuristics to the multiobjective framework. Nevertheless we indicate in the bibliography the references of the corresponding papers.

2.1.1. Exact methods

We recognize four approaches of exact methods:

1. *Two-phase method.*
   This method, proposed by Ulungu and Teghem (1995), was initially used to solve biobjective CO problems. The first phase generates the set X_{SE} of supported efficient solutions by solving a sequence of single-objective KP obtained by several linear aggregations of the two-objective functions. The second phase explores the interior of all the right-angled triangles defined by two consecutive points of Z_{SN} to obtain the set X_{NE} of nonsupported efficient solutions. This second phase is more complex and problem dependent. Ulungu and Teghem (1997) applied this method to a biobjective MOKP using a branch and bound method in the second phase. This method is improved by Visée et al. (1998) and they were able to solve randomly generated instances with up to 500 items. Jorge and Gandibleux (2007) proposed an improvement by using better bounds in the branch and bound and a ranking algorithm in the second phase.
2. *Transformation into biobjective shortest path problems.*
   This approach was proposed by Captivo et al. (2003). The obtained biobjective shortest path problem was solved with an adapted version of the labeling algorithm of Martins and Dos Santos (1999). They used biobjective instances of three different types, such as random, weakly correlated, and strongly correlated objectives. They solved these three types of instances with, respectively, up to 320, 800, and 900 items. They showed that their results was better in term of computational time than the method used by Visée et al (1998). This approach was extended further by Figuera et al. (2006).
3. *Dynamic programming.*
   An attempt to extend a DP algorithm to the MOKP was first suggested by Klamroth and Wiecek (2000). Bazgan et al. (2009b) developed this idea by adding several complementary dominance relations to discard partial solutions. They obtained an efficient method by testing on different types of instances, such as random (type A), strongly correlated (type B), weakly uncorrelated (type C), and strongly uncorrelated (type D) biobjective. In less than 2 hours of computational time, biobjective instances of type A, B, C, and D with, respectively 700, 4,000, 500, and 250 items were solved. When they compared their results with the method used by Captivo et al. (2003) and with that of an ϵ-constraint method (Haimes et al., 1971) coupled with the ILOG CPLEX 9.0 solver, they obtained better results and solved instances of higher

size. They also tested their method on three-objective instances. Because of the explosion of the cardinality of X_E, they could only solve instances of type A with up to 110 items and of type C with up to 60 items. It is still remarkable since this is the first exact method that was adapted to three-objective instances of the MOKP.

4. *Hybridization.*
   Delort and Spanjaard (2010) proposed a hybridization (called *two-phasification*) between the two-phase method and a DP procedure for solving biobjective instances. The DP procedure is applied in the second phase. Shaving procedures and bound sets are integrated to reduce the size of the problems. The results obtained were better than those of Bazgan et al. (2009b) on random and correlated instances, and comparable to those of uncorrelated instances.

2.1.2. Approximation algorithms

Following Erlebach et al. (2002) and Vazirani (2001), we first recall some definitions (in case of maximization).

- An ϵ-*efficient set* is a set Y_ϵ of feasible solutions such that for all $x_e \in X_E$ there exists $y \in Y_\epsilon$ satisfying

$$z_k(y)(1+\epsilon) \geq z_k(x_e) \quad \forall k \in \{1, \ldots, p\}$$

  with the same ratio $r = (1+\epsilon)$ for all objectives.
- A $(1+\epsilon)$-*approximation algorithm* A_ϵ is an algorithm producing an ϵ-efficient set in polynomial time.
- A *polynomial-time approximation scheme* (PTAS) is a family of algorithms that contains, for each $\epsilon > 0$, a $(1+\epsilon)$-approximation algorithm A_ϵ.
- A *fully polynomial-time approximation scheme* (FPTAS) is a PTAS for which A_ϵ is polynomial in ϵ^{-1}.

The following few papers deal with approximation algorithms.

- Erlebach et al. (2002) presented an efficient and applicable FPTAS for the MOKP. They also presented a PTAS for the MOMKP based on linear programming.
- Kumar and Banerjee (2006) proposed a restricted evolutionary multiobjective optimizer, called REMO, which gives $(1+\epsilon)$ approximations. It is based on a restricted mating pool with a separate archive to store the remaining population. They presented a rigorous running time analysis of the algorithm.
- Bazgan et al. (2009a) proposed a new FPTAS with a practical behavior. The main idea is the same as that in their exact method (Bazgan et al., 2009b). They compared their FPTAS with that of Erlebach et al. (2002) and obtained better results.

2.1.3. Heuristic methods

Here the aim is to find a good approximation of X_E using metaheuristics.

(a) Simulated annealing (SA)

- Ulungu (1993) presented the first adaptation of SA to MO through multiobjective SA (MOSA) (Ulungu et al., 1999). SA is simply applied several times with a well-diversified set of weight sets aggregating the objective functions. For each weight set, the nondominated solutions are kept and all these solutions are finally merged to achieve a unique approximation. He solved random biobjective instances with up to 500 items (using the same instances as in the exact method of Visée et al., 1998)
- Czyzak and Jaszkiewicz (1998) proposed the Pareto SA (PSA) to solve random bi-, three-, and four-objective instances with up to 800 items. In PSA, an initial set of solutions is generated. Weight sets are associated with each of these solutions that are optimized in the same way as in MOSA. For a given solution, the weight set is changed in order to induce a repulsion mechanism, allowing dispersion of the solutions over all the regions of the Pareto front.

(b) Tabu search (TS)

- Ben Abdelaziz and Krichen (1997) proposed a TS-based method. They solved random biobjective instances with up to 100 items. They extended their method by integrating the tabu search into a genetic algorithm (Ben Abdelaziz et al., 1999).
- Gandibleux and Fréville (2000) proposed a TS, called MOTS. In MOTS, an augmented weighted Tchebycheff scalarizing function is used to select the best neighbor. The weight set is dynamically updated such that the search process is oriented toward a region of the objective space where few nondominated points have been generated. They also used different techniques to reduce the decision space to an interesting area. They tested MOTS on random biobjective instances of 50 and 100 items.

(c) Genetic algorithm (GA)
Gandibleux et al. (2001) developed a two-phase method. In the first phase an exact algorithm to compute the set X_{SE} has been used and in the second phase the traditional components of a GA has been applied.

(d) Scatter search (SS)

- Gomes da Silva et al. (2006) proposed SS-based method, following the usual structure of SS. They tested their method on large size random biobjective instances with a number of items ranging from 100 to 6,000. They compared their method with the exact method of Visée et al. (1998) and showed that for the 100, 300, and 500 items instances, they generate on average, respectively, 33.13%, 9.75%, and 5.12% of the efficient solutions. Obviously, the running time of their method is much lower: for $n = 500$, the exact method takes about 400 times the computational time required by their heuristic.
- Gomes da Silva et al. (2007) presented an improvement of their previous method by modifying some elements of the SS technique. In particular, in the solution combination method, which combines solutions from each subset of a reference set to create new solutions, they used an adapted version of the exact method of Visée et al. (1998) to solve small size residual problems. They improved their previous results since they generate for the same instances with 100, 300,

and 500 items, respectively, 87.3%, 95.0%, and 91.2% of the efficient solutions. On the other hand, the improvement of running time was not significant: for the 500 items instance, the ratio of running times is only equal to 1.5 instead of 400. They showed that their results were better than those generated by the MOSA of Ulungu et al. (1999) and the genetic approach of Gandibleux et al. (2001) presented earlier.

(e) Linear relaxation
Zhang and Ong (2004) proposed a simple method essentially based on an efficient heuristic for the linear relaxation. They tested their method on random biobjective instances. They solved instances from 500 to 50,000 items and showed that they obtained better results if their method is coupled with the ILOG CPLEX 7.5 solver.

2.1.4. Particular studies

There are other studies concerning the MOKP.

- Gandibleux and Klamroth (2006) studied cardinality bounds for the MOKP based on weighted sum scalarizations. They showed that these bounds can be used to reduce the feasible set of the biobjective MOKP.
- Ehrgott and Gandibleux (2007) introduced the concept of bound sets for MOCO problems. Indeed, well-known bounds for multiobjective problems are the ideal point (lower bound) and the nadir point (upper bound) but first of all these bounds are not easy to compute, especially the nadir point, and second these values are very far from the Pareto front. They thus generalized the notion of a bound value to that of a bound set and applied these concepts to, among others, the MOKP. They obtain upper bound sets by using the linear relaxation of the MOKP and lower bound sets by using a simple greedy algorithm.
- Jorge and Gandibleux (2007) presented new properties aiming to reduce the size of the biobjective MOKP. Based on a lower and upper bound on the cardinality of a feasible solution for the KP introduced by Glover (1965) and on dominance relations in the data space of the MOKP, they reduced the size of the biobjective instances of the MOKP by fixing, *a priori*, about 10% of the variables, on random instances.
- Gomes da Silva et al. (2008) studied the interesting notion of core problems for the MOKP. Indeed, this concept has never been directly used in the MO framework. They thus investigated the existence of the core structure in the MOKP and defined the notion of *biobjective* core. They reported computational experiments related to the size of the biobjective core on different types of instances. The results show that, on average, the biobjective core is a very small percentage of the total number of items. Then, they proposed a heuristic and an exact method based on these results, but without testing these methods experimentally.

2.2. The MOMKP

For the MOMKP, as far as we know, these are mainly heuristic methods that have been developed. Although many of them are hybrid methods, we have tried to make a distinction between those based on evolutionary algorithms (Section 2.2.1) and those based on local search (Section 2.2.2). Some particular studies are analyzed in Section 2.2.3.

2.2.1. Evolutionary algorithms

- It seems that Zitzler and Thiele (1999) were the first to tackle the MOMKP. They performed a comparative study of five different multiobjective GAs (MOGAs) for the MOMKP, and also introduced the strength Pareto evolutionary algorithm (SPEA). They showed that this method outperforms the other methods. In their paper, they introduced the instances of the MOMKP that were later used by many others. For these instances (that we will call the ZMKP instances), the number of objectives is equal to the number of constraints ($m = p$). Nine different instances with two, three, and four objectives, in combination with 250, 500, and 750 items have been created. The profits and the weights were randomly generated in the interval [10, 100]. The knapsack capacities were set to half the total corresponding weight. In all the works presented in the following, if nothing is mentioned about the instances used, that means that the ZMKP instances are considered. An improved version of SPEA, called SPEA2, was further developed by Zitzler et al. (2001). In SPEA2, the rank of an individual takes into account the number of individuals that dominate it and the number of individuals dominated by the individual. Clustering in the objective space is used to reduce the number of individuals, and it allows to obtain only one representative individual in small regions of the objective space. The new algorithm has been tested on a subset of the ZMKP instances and it was concluded that SPEA2 performed better than its predecessor for all instances.
- Knowles and Corne (2000b) compared the memetic-Pareto archived evolution strategy (M-PAES), based on Pareto ranking of the solutions, with the random directions-multiobjective genetic local search (RD-MOGLS) (Jaszkiewicz, 1998), based on random scalarization functions. They showed that both algorithms work well and produce better results than (1+1)-PAES (Knowles and Corne, 1999).
- Jaszkiewicz (2000) applied MOGLS and compared it with SPEA. In MOGLS, the best solutions in the population P for the scalarizing function form a temporary population TP of small size. Both parents are randomly selected in TP. He showed that MOGLS outperforms SPEA. Jaszkiewicz (2001b) continued his experiments and compared five algorithms: MOGLS, M-PAES (Knowles and Corne, 2000b), Serafini's MOSA (Serafini, 1992), MOSA (Ulungu et al., 1999), and the PSA (Czyzak and Jaszkiewicz, 1998). Jaszkiewicz concluded that MOGLS outperforms all the other methods. In Jaszkiewicz (2002), a comparison between MOGLS, SPEA, M-PAES and the Ishibuchi and Murata's MOGLS (IMMOGLS) (Ishibuchi and Murada, 1998) was shown and concluded that MOGLS outperforms all the other methods. Jaszkiewicz published the results obtained with MOGLS, which became the new reference for testing new methods.
- Jaszkiewicz (2004) gave up his MOGLS algorithm and compared three MOMHs: the Pareto memetic algorithm (PMA) (Jaszkiewicz, 2001a), SPEA, and the controlled elitist nondominated sorting GA (CENSGA) (Deb and Goel, 2001). In PMA, the selection is based on a tournament. The two parents selected are the winners of a tournament between solutions coming from a sample of size T randomly drawn from the population. The size of T is set in order to guarantee that this selection procedure gives the same quality of offspring than with the selection procedure of MOGLS. Indeed, PMA has been developed by Jaszkiewicz to reduce the running time of the selection process of MOGLS, while keeping the same quality results. PMA has been adapted to the MOMKP in a more sophisticated way than MOGLS in Jaszkiewicz (2000). SPEA and CENSGA have also been adapted to the MOMKP by Jaszkiewicz in the same way as PMA. The three methods thus share some identical components. Jaszkiewicz showed that PMA performs

better than SPEA and CENSGA on instances with more than two objectives. After this work, it seems that this adaptation of PMA became the new reference method, since this method is an evolution of MOGLS and is adapted with more sophisticated operators. Unfortunately, the results of PMA were not published, and the previous results obtained with MOGLS remained the reference, although it was possible to generate the results of PMA with the source code of PMA that Jaszkiewicz published, through the MOMHLib++ library (Jaszkiewicz, 2000). This library makes it possible to run various existing MOMHs on the MOMKP. In this library, MOGLS has also been adapted following PMA and shown that it gives better results than the initial MOGLS (Jaszkiewicz, 2000).

- Li et al. (2004) studied a hybrid adaptation of the estimation of distribution algorithm (EDA) (Larranaga and Lozano, 2002; Pelikan et al., 2006) to the MOMKP, by using a local search based on weighted sums, a random repair method, and a population sampled from a probability distribution. They compared their results with those of MOGLS and showed that they obtained better results for some indicators.
- Alves and Almeida (2007) presented a GA based on the Tchebycheff scalarizing function, called multiobjective Tchebycheff GA (MOTGA). Several stages were performed; each one intended for generating potentially nondominated points in different parts of the Pareto front. Different weight vectors act as pivots to define the weighted Tchebycheff scalarizing functions and direct the search for each stage. Linear relaxation of the integer program based on weighted Tchebycheff scalarizing functions followed by a repair procedure has been used to produce initial solutions. They compared MOTGA with SPEA, SPEA2, and MOGLS and showed that MOTGA obtains better quality indicators than the other MOMHs, with a lower running time, although the number of potentially nondominated solutions obtained by MOTGA is not very high.
- Lust and Teghem (2008) presented a memetic algorithm integrating TS (MEMOTS). The particularity of the method is in the selection of the parents for recombination. They used a dynamic hypergrid created in the objective space to select parents located in a region of minimal density. Then they applied a TS to the offspring. They showed that they obtained better results than MOGLS and PMA for various indicators for the instances with two or three objectives.

2.2.2. Local search

- Barichard and Hao (2002) proposed a TS that gives better results than SPEA but worse results than MOGLS. Their method is a classic TS with the Tchebycheff scalarizing function to select the best neighbor, except that an interesting way to measure the diversity has been added. In 2003, they improved the TS by integrating it into a GA (Barichard and Hao, 2003) and with this new hybrid method, they concluded that they obtained better results than MOGLS.
- Vianna and Arroyo (2004) proposed an adaptation of the GRASP metaheuristic to the MOMKP. The adaptation of GRASP follows the following frame: at each iteration, a weighted linear sum of the objectives is defined and a solution is built considering this as a linear sum. The solution is then improved by a local search that also uses the linear sum. They compared their method with SPEA2 and MOGLS and showed that they obtained better results.
- Gomes da Silva et al. (2004) adapted their SS for the MOKP (Gomes da Silva et al., 2006) to also tackle the MOMKP. Surrogate relaxation was used to convert the MOMKP into an MOKP

by generating adequate surrogate multipliers. They first compared their method with SPEA and MOGLS on the ZMKP instances with only two objectives. They showed that by comparing the potentially nondominated points generated in objective space better results are obtained on these instances. Then they tested their method on new instances of bigger size: the number of items ranged between 1,000 and 3,000 and the number of constraints was in the range of 10–100. The values of the profits and the weights were randomly generated between 1 and 100 and constraint capacity of each knapsack was equal to 50% of the total corresponding weight. They evaluated the quality of their results by measuring different distances between the approximations and the upper bound obtained with the surrogate relaxation. They showed that the approximations generated are very close to the upper bound. On the other hand, the running time was close to 1 hour, but given the size and the number of constraints taken into account, that is reasonable.

- Alsheddy and Tsang (2009) proposed a guided Pareto local search (GPLS). This method combines the GLS (guided LS) of Voudouris and Tsang (1999) and the PLS of Paquete et al. (2004), with a penalization of the objective functions to escape from local optima. They compared their method with general MOGA (SPEA2 and NSGA-II; Deb et al., 2002). In Alsheddy and Tsang (2010), they improved their approach by using better initial solutions.

2.2.3. *Particular studies*

We conclude the review of the works concerning the MOMKP using four particular studies.

- Sato et al. (2007) studied and compared the effects on performance of local dominance and local recombination applied to different locality. They used the NSGA-II algorithm (Deb et al., 2002) to run their experiments. They showed that the optimum locality of dominance is different from the optimum locality of recombination. They also pointed out that when local dominance and local recombination with different locality are applied, the performances are significantly better than the performances when either local dominance or local recombination is applied , or even when local recombination and dominance with the same locality are applied.
- Beausoleil et al. (2008) applied a multistart search method combined with path-relinking to generate potentially efficient solutions in the so-called *balanced* zone of the Pareto front. However, this concept is not defined at all: this means that the method focuses on solutions that establish a good compromise between all the objectives. The method is relatively sophisticated and integrates many ideas developed by Glover (1994, 2000): ghost image process, strategic oscillation approach, and use of conditional-exclusion memory or developed by Glover and Laguna (1997) with the path-relinking technique. In addition, the method presents many parameters. The results of the method are compared with many MOGAs, including SPEA, SPEA2, NSGA, and NSGA-II but not with MOGLS or PMA that gives better performances on the MOMKP. They showed that their approach is competitive regarding those algorithms.
- Florios et al. (2010) published an exact method for solving the MOMKP. The method is a simple branch and bound method based on the ideal point as fathom rule and on branching heuristics. They experimented their method on instances with $p = m = 3$. They compared the results with the "ϵ-constraint" approach of Laumanns et al. (2006) and showed that their approach is faster.
- Mavrotas et al. (2009) adapted the notion of core to the MOMKP, as Gomes da Silva et al. (2008) did for the MOKP. They used linear relaxations to define different weight intervals that allow

to create different MOMKP problems of smaller size than the original. For this, they used the core concept developed by Puchinger et al. (2006) for solving the MKP. They employed the exact method developed in Florios et al. (2010) to solve the residual problems. They used biobjective instances, with n going from 6 to 50, and m from 5 to 10. They also used some of the ZMKP instances. They studied the influence of the size of the core on the quality of the results and computational time. For example, for the ZMKP instance of 250 items, with two constraints and two objectives, they generated 81% of Z_N in about 30 minutes. But needed 21 hours to generate 93% of Z_N.

3. New approach

3.1. Presentation

From the preceding survey, the most promising methods are those based on the notion of core. The adaptations of Mavrotas et al. (2009) of the method proposed by Gomes da Silva et al. (2008) allow to obtain high-quality results, but unfortunately with a substantial computational time. This high computational time is the result of attempt to compute exactly the core or to solve exactly the residual problems. We show that with this new approach, it is possible to use these notions in a complete heuristic way to obtain simultaneously high-quality results and small computational times.

We have used the two-phase PLS (2PPLS) (Lust and Teghem, 2010), coupled with a very large scale neighborhood (VLSN) (Ahuja et al., 2002). With 2PPLS, similar to the methods based on the notion of core, the "divide and conquer" principle is applied by solving many residual problems. However, all those residual problems will be managed by the VLSN and selected in a heuristic way.

Local search with VLSN uses a large neighborhood combined with an efficient method to explore the neighborhood (otherwise it takes too much time to explore the neighborhood). This technique is very popular in single-objective optimization. For example, one of the best heuristics for solving the single-objective traveling salesman problem (TSP), the Lin-Kernighan heuristic (Lin and Kernighan, 1973), is based on VLSN. On the other hand, there is almost no study of VLSN for solving MOCO problems. The only known result is the local search of Angel et al. (2004), which integrates a dynasearch neighborhood (the neighborhood is solved with DP) to solve the biobjective TSP. We now present 2PPLS. This method only needs an initial population and a neighborhood (the VLSN in our case, which will be present later).

The aim of the first phase of 2PPLS is to generate a good approximation of the supported efficient solutions, by using weighted sums and efficient single-objective solvers for optimizing the weighted single-objective problems. The aim of the second phase is to improve the approximation by generating potentially nonsupported efficient solutions. For that we use the PLS (Angel et al., 2004; Paquete et al., 2004). This method is a straightforward adaptation of local search to MO and only needs a neighborhood function $\mathcal{N}(x)$. At the end, a local optimum, defined in an MO context, is found (Paquete et al., 2004) (called a Pareto local optimum set). The main part of the adaptation of 2PPLS to MOCO problems concerns the definition of the neighborhood (a VLSN in our case).

The pseudo-code of 2PPLS is given by Algorithm 1.

Algorithm 1 2PPLS

Parameters $\downarrow$: an initial population P_0, a neighborhood function $\mathcal{N}(x)$.
Parameters $\uparrow$: an approximation $\widetilde{X}_E$ of the efficient set X_E.
- -| Initialization of $\widetilde{X}_E$ and a population P with the initial population P_0
$\widetilde{X}_E \leftarrow P_0$
$P \leftarrow P_0$
- -| Initialization of an auxiliary population P_a
$P_a \leftarrow \varnothing$
while $P \neq \varnothing$ **do**
  - -| Generation of all neighbors p' of each solution $p \in P$
  for all $p \in P$ **do**
    for all $p' \in \mathcal{N}(p)$ **do**
      if $z(p) \not\geq z(p')$ **then**
        AddSolution($\widetilde{X}_E \updownarrow$, $p' \downarrow$, $z(p') \downarrow$, $Added \uparrow$)
        if $Added = true$ **then**
          AddSolution($P_a \updownarrow$, $p' \downarrow$, $z(p') \downarrow$)
  - -| P is composed of the new potentially efficient solutions
  $P \leftarrow P_a$
  - -| Reinitialization of P_a
  $P_a \leftarrow \varnothing$

The method starts with the population P composed of potentially efficient solutions given by the initial population P_0. Then, all the neighbors p' of each solution p of P are generated. If a neighbor p' is not weakly dominated by the current solution p, we try to add the solution p' to the approximation $\widetilde{X}_E$ of the efficient set, which is updated with the procedure AddSolution. This procedure is not described in this paper but simply consists of updating an approximation $\widetilde{X}_E$ of the efficient set when a new solution p' is added to $\widetilde{X}_E$. This procedure has four parameters: the set $\widetilde{X}_E$ to actualize, the new solution p', its evaluation $z(p')$, and an optional Boolean variable called $Added$ that returns $True$ if the new solution has been added and $False$ otherwise. If the solution p' has been added to $\widetilde{X}_E$, $Added$ is true and the solution p' is added to an auxiliary population P_a, which is also updated with the procedure AddSolution. Therefore, P_a is only composed of (new) potentially efficient solutions. Once all the neighbors of each solution of P have been generated, the algorithm starts again, with P equal to P_a, until $P = P_a = \varnothing$. The auxiliary population P_a is used such that the neighborhood of each solution of the population P is explored, although some solutions of P become dominated following the addition of a new solution to P_a. Thus, sometimes, neighbors are generated from a dominated solution.

3.2. Adaptation of 2PPLS to the MOMKP

The 2PPLS requires only two elements to be adapted to an MOCO problem: an initial population and a neighborhood.

3.2.1. Initial population

We use a greedy heuristic in this case. To create a new solution, the items are added one by one to the knapsack. At each iteration, the item s that maximizes the following ratio (R_1) (Michalewicz and Arabas, 1994) is selected:

$$R_1 = \frac{\sum_{k=1}^{p} \lambda_k c_k^s}{\sum_{j=1}^{m} \left(\frac{w_j^s}{W_j - \sum_{i=1}^{n} w_j^i x_i + 1} \right)}. \tag{1}$$

The greedy procedure is run S times, S being the number of weight sets λ used. The weight sets are uniformly distributed for biobjective instances and randomly generated for three-objective instances.

3.2.2. Very large scale neighborhood

The aim of the VLSN is to define the function $\mathcal{N}(x)$, which generates a set of neighbors from a current solution x. For that, two lists are created, both are of size L: one list (L1) containing the items candidates to be removed (thus present in x) and another list (L2) containing the items candidates to be added (thus missing in x).

To create L1, the items, in x, minimizing the ratio R_2, defined by

$$R_2 = \frac{\sum_{k=1}^{p} \lambda_k c_k^s}{\sum_{j=1}^{m} w_j^s}, \tag{2}$$

are selected.

To create L2, the items, not in x, maximizing the ratio R_1 (defined by (1)) are selected. For biobjective instances, the weight set λ necessary for the computation of these ratios is determined according to the relative performance of the potentially efficient solution x selected, for the different objectives, among the population P. That is, better the evaluation of the solution x according to an objective, higher the value of the weight. For three-objective instances, the weight set is randomly generated.

Once both lists containing each L items have been created, we merge them to create a new MOMKP instance, called the residual problem, which is composed of $(L * 2)$ items. The capacities W_j of the residual problem are equal to $W_j - \sum_{\substack{i=1 \\ i \notin L1}}^{n} w_j^i x_i$ with $j = 1, \ldots, m$.

We then solve the residual problem. As this problem is of small size, we can use an exact method. We have implemented a branch and bound method based on the method of Florios et al. (2010). In addition, to be able to use a higher value of L while keeping reasonable running times, we have also used a heuristic method. For that, we have employed a simplified version of MEMOTS (Lust and Teghem, 2008): no hypergrid will be used to select the parents. The reason is that the number of potentially efficient solutions generated will not be high, and thus managing a hypergrid to select a solution of minimal density is not worthwhile. The advantage of not using the hypergrid is the

simplification of the method and the elimination of parameters to tune. Consequently, both parents will be selected randomly in the set of potentially efficient solutions.

Once the residual problem has been solved, we merge the efficient solutions (or potentially efficient depending on the method used) of this small problem with the current solution to obtain the neighbors.

4. Data and reference sets

Similar to the work done by many authors previously, we use the ZMKP instances with 250, 500, or 750 items, two objectives and two constraints, or three objectives and three constraints. It should be noted that, hereafter, when we speak about, for example, the 250-2 instance, it will mean the instance with 250 items and two objectives. In order to assess the quality of the approximations generated, we have used the unary indicators presented in the Introduction.

Some of these indicators need a reference set. For the biobjective instances, we use the nondominated points generated by Tuyttens (2006) by applying the ϵ-constraint method coupled with the commercial CPLEX solver. For the three-objective instances, we approximate the efficient solutions by applying heuristic methods several times (the MEMOTS; Lust and Teghem (2008) and MOGLS; Jaszkiewicz (2002) methods) during a high number of iterations. Then we retain only the potentially nondominated points.

The values of the reference points to compute the R indicator are the same as the ones used in Jaszkiewicz (2000). As also done in Jaszkiewicz (2000), the number of weight sets used to compute R is equal to 201 for the biobjective instances and to 50 for the three-objective instances.

The bounding point considered to compute the hypervolume is simply the point (0,0), for the biobjective instances. For the three-objective instances, we did not compute the hypervolume since its computation was very time-consuming given the high size of the sets $\widetilde{Z}_N$.

5. Results

A Pentium IV with 3 GHz CPUs and 512 MB of RAM was used for the experiments. Twenty runs of the algorithms are performed each time. The running time of our implementation of the algorithms corresponds to the wall clock time.

5.1. Study of the influence of the quality of the initial population

We have tried different values of S (see Section 3.2.1) for the generation of the initial population. However, given the quality of the neighborhood used in the second phase, the value of S had only a small influence on the quality of the results. We do not report the results here but you can find the results of this study in Lust (2009). In the following, we will use $S = 100$ for the biobjective instances and $S = 150$ for the three-objective instances.

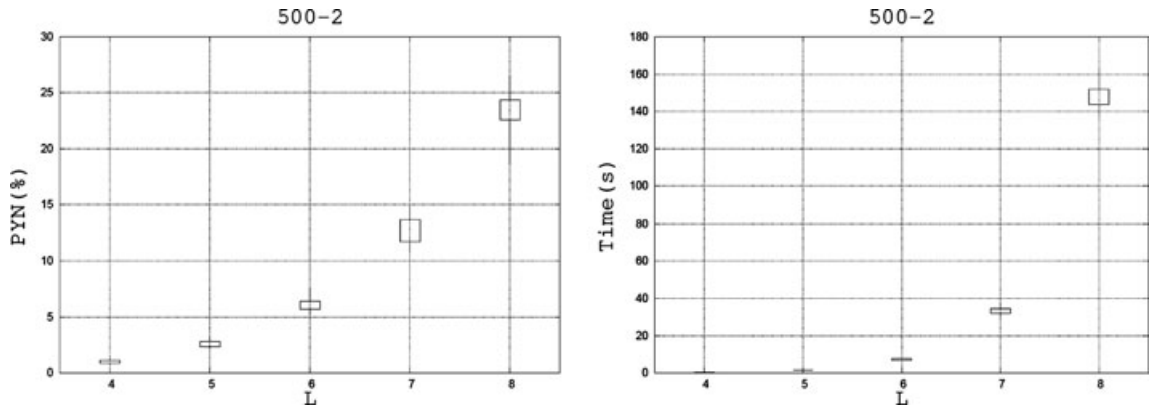

Fig. 1. Influence of L with the EXACT subroutine.

5.2. *Study of the influence of the length of the neighborhood*

It is interesting to measure the improvements of quality when the size of the neighborhood is increased.

We show in Fig. 1 the evolutions of the proportion of nondominated points generated (P_{Z_N}) and the running time (only of the second phase) according to L, when the branch and bound method is used to solve the residual problems (called the EXACT subroutine), for the 500-2 instance. We vary the value of L from 4 to 8. We see that there are strong improvements of P_{Z_N} when L is increased. On the other hand, the running time evolves exponentially according to L. Using a value of L greater than 8 would give unreasonable running times.

In Fig. 2, we show the evolutions of P_{Z_N} and the running time according to L, if MEMOTS is used to solve the residual problems, for the 500-2 instance. We vary the values of L from 4 to 20. We use three different numbers of iterations for MEMOTS: $N = 100$, $N = 200$, and $N = 400$. We see that for small values of L, P_{Z_N} is more or less equal no matter the number of iterations. For L equal to 10, it is clear that we obtain better results if N is higher. On the other hand, the running time is more when N is higher, but still evolves more or less linearly according to L. An interesting behavior is pointed out by the figure showing the evolution of P_{Z_N} according to L. For L equal to about 16 and for a number of iterations N equal to 100 or 200, there is a deterioration of P_{Z_N} while the running time is increasing. It means that the number of iterations performed in MEMOTS is not high enough to solve the residual problems, and that therefore the quality of the approximations obtained for the residual problems is not good enough to improve P_{Z_N}. Fixing good values for L and N seems thus not easy since these two values have to be increased at the same time if we want to improve the quality of the results.

Now, it is interesting to see, for a fixed running time, what is the best option: using the EXACT or MEMOTS subroutine, with which value of L, and for MEMOTS, with how many iterations?

In order to answer this question, we have represented in Fig. 3, the evolution of P_{Z_N} according to the running time (the running time is controlled by the value of L: from 4 to 8 for the EXACT subroutine and from 4 to 20 for the MEMOTS subroutine), for the 250-2 and 750-2 instances. We

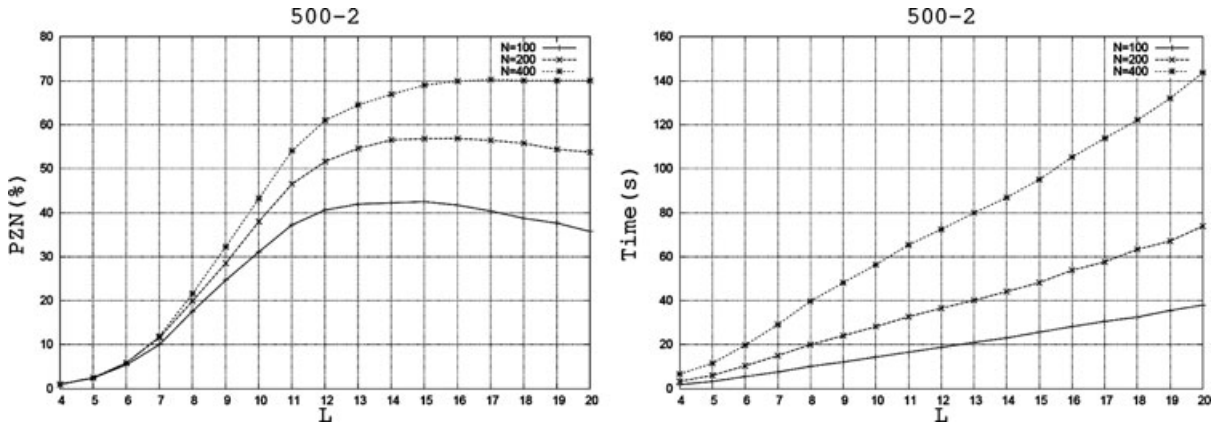

Fig. 2. Influence of L with the MEMOTS subroutine.

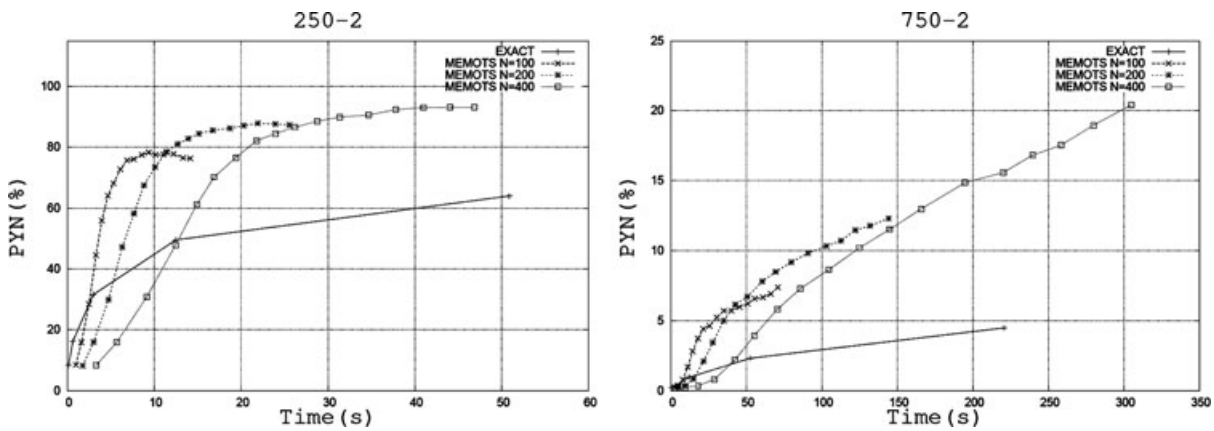

Fig. 3. Comparison between the EXACT and MEMOTS subroutines to solve the residual problems according to the running time.

see that for very small running times, it is better to use the EXACT subroutine. As soon as we have long running times, it is better to use the MEMOTS subroutine. In this case, good combination between L and N has to be determined according to the running time given. The higher the running time allowed, the higher the number of iterations.

5.3. Comparison with other algorithms on biobjective instances

We have obtained the following results for the 250-2, 500-2, 750-2 instances: SPEA (Zitzler and Thiele, 1999) (30 runs); SPEA2 (Zitzler et al., 2001) (30 runs, but only for the 750-2 instance); MOGLS00 (Jaszkiewicz, 2000) (20 runs); MOGLS04 (Jaszkiewicz, 2000) (20 runs, different than MOGLS00 since obtained with the library MOMHLib++ (Jaszkiewicz, 2000)); PMA (Jaszkiewicz, 2004) (20 runs); IMMOGLS (Ishibuchi and Murada, 1998) (20 runs); MOGTS (Barichard and Hao,

Table 1
Comparison between 2PPLS and ALL based on the indicators

Instance	Algorithm	$\mathcal{H}$ (10^7)	$I_{\epsilon 1}$	R	D_1	D_2	$\|PE\|$	P_{Z_N}(%)	Time (s)
250-2	2PPLS	**9.8690**	**1.000508**	**245.740328**	**0.029**	**2.680**	482.10	**68.05**	7.27
	ALL	**9.8690**	1.000839	246.129567	0.069	2.838	376.00	31.87	/
500-2	2PPLS	**40.7873**	**1.000282**	**430.902857**	**0.025**	**1.976**	1,131.00	**42.85**	23.43
	ALL	40.7850	1.000513	431.961766	0.081	2.045	688.00	5.51	/
750-2	2PPLS	**89.3485**	**1.000509**	**743.615748**	**0.076**	**1.494**	1,558.90	**4.15**	17.52
	ALL	89.3449	1.000553	744.089577	0.092	**1.494**	996.00	0.99	/

Values in **bold** indicate the best values obtained among the methods compared.

2003) (1 run); GRASP (Vianna and Arroyo, 2004) (1 run); MOTGA (Alves and Almeida, 2007) (20 runs); PATH-RELINKING (Beausoleil et al., 2008) (30 runs); GPLS (Alsheddy and Tsang, 2009) (30 runs); mGPLS (Alsheddy and Tsang, 2010) (30 runs); iGPLS (Alsheddy and Tsang, 2010) (30 runs). These results have been obtained from web sites or directly from the different authors. We have obtained a large number of results. It is unfortunate that we are not able to obtain the results of Gomes da Silva et al. (2004).

Thanks to these results, we have generated a reference set, called ALL, formed by merging the potentially nondominated points obtained by all the runs of all algorithms, which gives a very high quality set.

However, we show that it is possible to obtain better results than this set, for all the indicators considered, in reasonable times, with 2PPLS and the VLSN coupled with the MEMOTS subroutine. We have carefully selected the parameters such that we obtain better or equal results than the reference set ALL for all the indicators. The parameters are the following:

- 250-2: $L = 9$ and $N = 200$.
- 500-2: $L = 15$ and $N = 100$.
- 750-2: $L = 9$ and $N = 100$.

The results for 2PPLS are given in Table 1. $|PE|$ gives the number of potentially nondominated points generated. We see that we obtain better or equal values for all indicators, in very reasonable running times: 7 seconds for 250-2, 23 seconds for 500-2, and 18 seconds for 750-2. 2PPLS with this configuration thus seems very competitive.

5.4. *Comparison between MEMOTS and 2PPLS on biobjective instances*

We realize in this section a comparison between MEMOTS and 2PPLS, for different running times. We do not show the results of the comparison of 2PPLS with other algorithms, since in Lust and Teghem (2008), we already show that MEMOTS gives better values than MOGLS (Jaszkiewicz, 2000) and PMA (Jaszkiewicz, 2004) for the different indicators used.

For MEMOTS, we use the parameters recommended in Lust and Teghem (2008). The running time of MEMOTS is controlled with the number of iterations, varying between 2,000 and 40,000. For 2PPLS, the running time is controlled by both L and N. We vary L from 4 to 20, and N linearly

evolves according to L in the following way:

$$N = 100 + \frac{75}{4}(L - 4) \tag{3}$$

(in this way, for $L = 4$, $N = 100$ and for $L = 20$, $N = 400$).

The results are presented in Fig. 4 where the evolutions of D_1 and P_{Z_N} according to the running time are shown.

We see that except for small running times, the results obtained with 2PPLS are better than that with MEMOTS. With 2PPLS, we can generate, about 90% of the nondominated points for the 250-2 instance, about 70% for the 500-2 instance, and about 20% for the 750-2 instance, in reasonable running times. The running times are remarkable since, for example, Mavrotas et al. (2009) can generate 81% of Z_N for the 250-2 instance in about 30 minutes, while we can attain this result in about 15 seconds. Also, for this same instance, they need 21 hours to generate 93% of Z_N, while we only need 42 seconds. We are thus 1,800 times faster!

5.5. *Three-objective instances*

We present the results of MEMOTS and 2PPLS in Table 2 for the 250-3 instance. The results of MEMOTS have been obtained with the parameters recommended in Lust and Teghem (2008). For 2PPLS, we have used the following parameters: $L = 12$ and $N = 200$. The results of 2PPLS correspond to only one run (for computational overhead reason).

We see that the results obtained with 2PPLS are of better quality for all the indicators considered. The number of potentially efficient solutions obtained with 2PPLS (68,540 potentially efficient solutions generated!) is five times more important than with MEMOTS. But the running time of 2PPLS is very high: about 8 hours! Indeed, the PLS method is stopped only when it is not possible to find a new nondominated neighbor from one of the potentially efficient solutions. In the case of three-objective instances, there are so many potentially efficient solutions that the stop condition is only met after a long running time.

From these results, we can say that it will be impossible to use 2PPLS to solve the 500-3 and 750-3 instances in a reasonable time, except if the method is stopped before convergence or if the neighborhood is drastically limited.

We have done that for the 250-3 instance, where we limit the running time of 2PPLS to the same running time as in MEMOTS. The results are shown in Table 3, for two different running times.

We see that the results obtained with MEMOTS are better than the results of 2PPLS, except when the running time is higher, but only for the D_1 indicator. Stopping 2PPLS before convergence can indeed yield results of second-rate quality, since in PLS the exploration of the decision space is not managed as in MEMOTS where the search is constantly guided in regions of a low density (in the objective space). We can thus conclude that 2PPLS is more suited to solve biobjective instances.

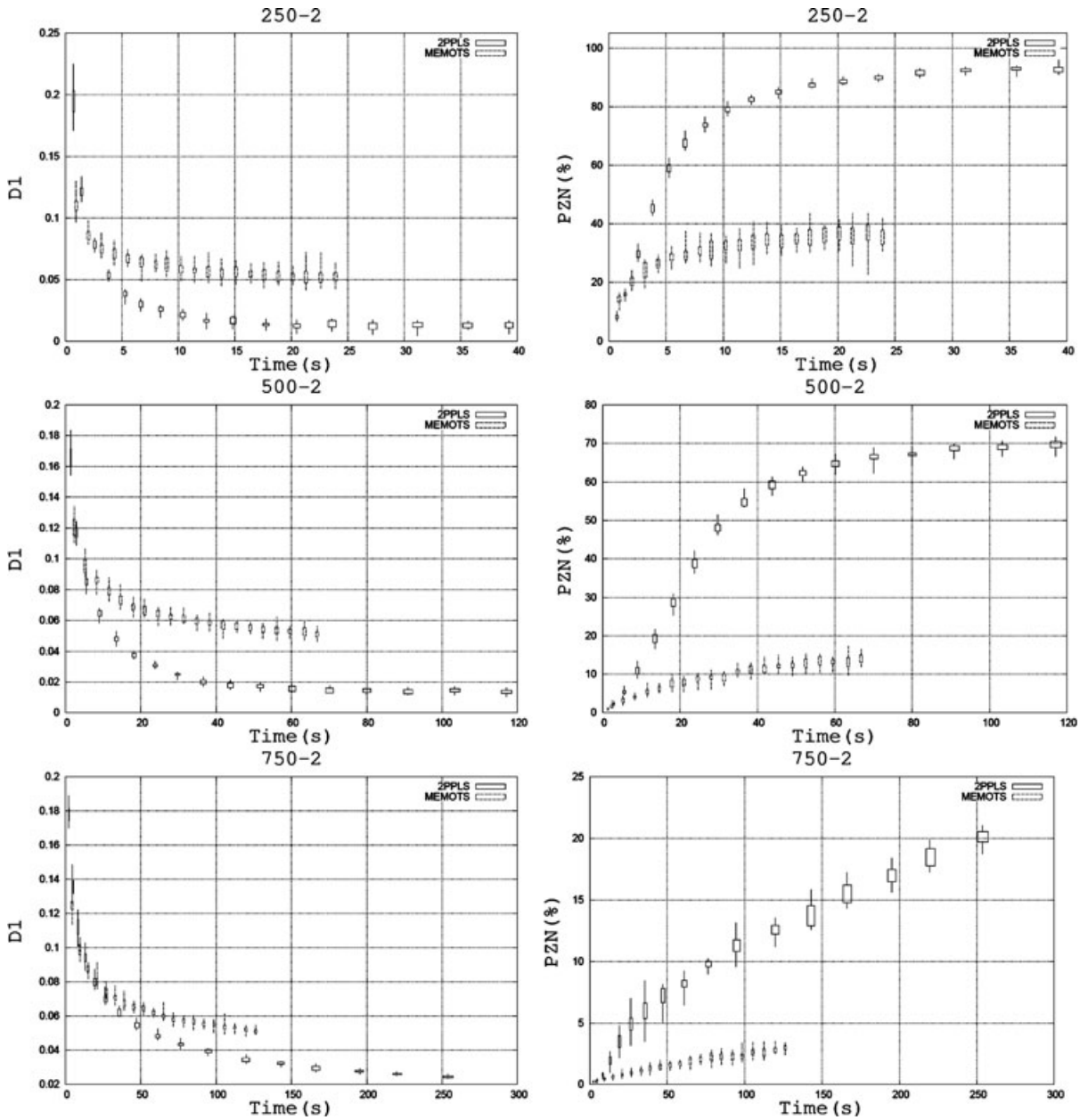

Fig. 4. Comparison of MEMOTS and 2PPLS: evolution of D_1 and P_{Z_N} according to the running time.

5.6. *MOKP*

We have also performed some tests for the MOKP ($m = 1$). In this case, we compare our results with the FPTAS of Bazgan et al. (2009a), with $\epsilon = 0.1$, which means that they guarantee a

Table 2
Comparison between 2PPLS and MEMOTS for the 250-3 instance (1)

Instance	Algorithm	$I_{\epsilon 1}$	R	D_1	D_2	$\|PE\|$	Time (s)
250-3	2PPLS	**1.017902**	**306.469352**	**3.863**	**4.798**	68,540.00	28,626.80
	MEMOTS	1.020264	309.680656	4.074	5.214	12,043.00	**129.46**

Table 3
Comparison between 2PPLS and MEMOTS for the 250-3 instance (2)

Instance	Algorithm	$I_{\epsilon 1}$	R	D_1	D_2	$\|PE\|$	Time (s)
250-3	MEMOTS	**1.025566**	**313.875552**	**4.283**	**5.989**	4,581.50	8.01
	2PPLS	1.029983	317.772372	4.363	6.932	4,872.20	8.25
250-3	MEMOTS	**1.020264**	**309.680656**	4.074	**5.214**	12,043.00	129.46
	2PPLS	1.020712	311.238745	**4.034**	5.510	13,504.35	129.18

Table 4
Comparison between 2PPLS and an FPTAS for MOKP instances

Instance		FPATS		2PPLS	
Type	n	Time (s)	ϵ	Time (s)	ϵ
A	400	6.084	0.0076	**3.55**	**0.00030**
	700	32.275	0.0060	**13.60**	**0.00018**
B	2,000	1.452	0.0028	**1.292**	**0.00055**
	4,000	11.220	0.0023	**9.76**	**0.00026**
C	300	10.099	0.0096	**9.18**	**0.0011**
	500	44.368	0.0064	**23.09**	**0.00076**
D	100	2.356	0.0183	**0.8379**	**0.0081**
	200	36.226	0.0117	**9.9198**	**0.0027**
	250	62.970	0.0098	**18.2884**	**0.0026**

(1, 1)-approximation (the final approximation obtained is however of better quality). We did that for the four types of instances defined by Bazgan et al. The results are provided in Table 4.

We then obtain better values for ϵ in less running times (the computer used by Bazgan et al. is a 3.4 GHz computer with 3,072 MB of RAM). Therefore, we can conclude that compared to a heuristic, the guarantee of performance given by an approximation algorithm has a price.

6. Conclusion and perspectives

In this paper, we have first surveyed the vast literature about the MOMKP. We have then proposed 2PPLS with a VLSN to solve the MOMKP. On biobjective instances, 2PPLS with the VLSN, which uses a simplified version of MEMOTS to solve the residual problems, gives better results than other methods existing in the literature, in particular MEMOTS. Methods based on VLSN for solving MOCO problems are thus a promising stream of research. On the other hand, on three-objective

instances, the convergence time of 2PPLS is very high and MEMOTS turned out to be more efficient.

We think that in the case of randomly generated instances of the MOMKP with more than two objectives, the decision maker should get involved in the process of generating solutions, at the beginning or during the execution of the algorithm, in order to direct the search and to limit the number of solutions generated. An *a posteriori* approach should be avoided as often as possible. Indeed, while 2PPLS was very competitive on biobjective instances, the results with MEMOTS were better on three-objective instances, as the search in 2PPLS is not as well directed as in MEMOTS or as in other memetic algorithms using scalarizing functions to guide the search. It corroborates the need for the intervention of the decision maker during the 2PPLS process and thus to modify this approach in an interactive way.

Acknowledgements

T. Lust acknowledges the "Fonds National de la Recherche Scientifique" for a research fellow grant (Aspirant FNRS). We also thank V. Barichard, D.S. Vianna, M. João Alves, R. Beausoleil, and A. Alsheddy for providing their results.

References

Ahuja, R.K., Ergun, Ö., Orlin, J.B., Punnen, A.P., 2002. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics* 123, 1–3, 75–102.

Alsheddy, A., Tsang, E.P.K., 2009. Guided Pareto local search and its application to the 0/1 multi-objective knapsack problems. In *Proceedings of the Eighth Metaheuristic International Conference (MIC'09)*, Hamburg, Germany, July 13–16, 2009, pp. 1–14.

Alsheddy, A., Tsang, E.P.K., 2010. Guided Pareto local search based frameworks for pareto optimization. In *Proceedings of the WCCCI IEEE World Congress on Computational Intelligence*, Barcelona, July 18–23, 2010, pp. 1–8.

Alves, M.J., Almeida, M., 2007. MOTGA: a multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem. *Computers & Operations Research* 34, 3458–3470.

Angel, E., Bampis, E., Gourvès, L., 2004. A dynasearch neighborhood for the bicriteria traveling salesman problem. In Gandibleux, X., Sevaux, M., Sörensen, K., T'kindt, V. (eds) *Metaheuristics for Multiobjective Optimisation, Vol. 535 of Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin, pp. 153–176.

Balas, E., Zemel, E., 1980. An algorithm for large zero-one knapsack problems. *Operational Research* 28, 1130–1154.

Barichard, V., Hao, J.K., 2002. An empirical study of tabu search for the MOKP. In *Proceedings of the First International Workshop on Heuristics*, Vol. 4, Beijing, July 22–27, 2002, Series of Information & Management Sciences, pp. 47–56.

Barichard, V., Hao, J-K., 2003. Genetic tabu search for the multi-objective knapsack problem. *Journal of Tsinghua Science and Technology* 8, 1, 8–13.

Bazgan, C., Hugot, H., Vanderpooten, D., 2009a. Solving efficiently the 0-1 multi-objective knapsack problem. *Computers & Operations Research* 36, 1, 260–279.

Bazgan, C., Hugot, H., Vanderpooten, D., 2009b. Implementing an efficient FPTAS for the 0-1 multi-objective knapsack problem. *European Journal of Operational Research* 198, 1, 47–56.

Beausoleil, R.P., Baldoquin, G., Montejo, R.A., 2008. Multi-start and path relinking methods to deal with multiobjective knapsack problems. *Annals of Operations Research* 157, 105–133.

Ben Abdelaziz, F., Krichen, S., 1997. A tabu search heuristic for multiobjective knapsack problems. Technical Report RRR 28-97, Rutgers Center for Operations Research, Piscataway, NJ.

Ben Abdelaziz, F., Chaouachi, J., Krichen, S., 1999. A hybrid heuristic for multiobjective knapsack problems. In Voss, S., Martello, S., Osman, I., Roucairol, C. (eds) *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Kluwer Academic Publishers, Dordrecht, pp. 205–212.

Captivo, M.E., Clímaco, J.C.N., Figueira, J.R., Martins, E.Q.V., Santos, J.L., 2003. Solving bicriteria 0-1 knapsack problems using a labeling algorithm. *Computers & Operations Research* 30, 12, 1865–1886.

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

Dantzig, G., 1957. Discrete variable extremum problems. *Operations Research* 5, 226–227.

Deb, K., Goel, T., 2001. Controlled elitist non-dominated sorting genetic algorithms for better convergence. In Zitzler, E., Deb, K., Thiele, L., Coello, C.A., Corne, D. (eds), *Proceedings of First International Conference, EMO2001, Zurich, Switzerland, March 7–9, 2001, Vol. 1993 of Lecture Notes in Computer Science*. Springer, Berlin, pp. 67–81.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation* 6, 2, 182–197.

Delort, C., Spanjaard, O., 2010. Using bound sets in multiobjective optimization: application to the biobjective binary knapsack problem. In *Proceedings of 9th International Symposium on Experimental Algorithms (SEA 2010), Napoli, Italy, May 20–22, 2010, Vol. 6049 of Lecture Notes in Computer Science*, pp. 253–265.

Ehrgott, M., Gandibleux, X., 2002. Multiobjective combinatorial optimization. In Ehrgott, M., Gandibleux, X. (eds) *Multiple Criteria Optimization – State of the Art Annotated Bibliographic Surveys*, Vol. 52. Kluwer Academic, Boston, MA, pp. 369–444.

Ehrgott, M., Gandibleux, X., 2007. Bound sets for biobjective combinatorial optimization problems. *Computers & Operations Research* 34, 2674–2694.

Erlebach, T., Kellerer, H., Pferschy, U., 2002. Approximating multiobjective knapsack problems. *Management Science* 48, 12, 1603–1612.

Figuera, J.R., Wiecek, M., Tavares, G., 2006. Multiple criteria knapsack problems: network models and computational results. In *Proceedings of the Multi-Objective Programming and Goal Programming Conference (MOPGP'06)*, *Tours, France, June 12–14, 2006*.

Florios, K., Mavrotas, G., Diakoulaki, D., 2010. Solving multiobjective, multiconstraint knapsack problems using mathematical programming and evolutionary algorithms. *European Journal of Operational Research* 203, 1, 14–21.

Gandibleux, X., Fréville, A., 2000. Tabu search based procedure for solving the 0-1 multi-objective knapsack problem: the two objectives case. *Journal of Heuristics* 6, 3, 361–383.

Gandibleux, X., Klamroth, K., 2006. Cardinality bounds for multiobjective knapsack problems based on weighted sums scalarizations. In *Proceedings of the Multi-Objective Programming and Goal Programming Conference (MOPGP'06)*, *Tours, France, June 12–14, 2006*.

Gandibleux, X., Morita, H., Katoh, N., 2001. The supported solutions used as a genetic information in a population heuristics. In Zitzler, E., Deb, K., Thiele, L., Coello, C.A., Corne, D. (eds), *Proceedings of First International Conference, EMO2001, Zurich, Switzerland, March 7–9, 2001, Vol. 1993 of Lecture Notes in Computer Science.* Springer, Berlin, pp. 429–442.

Glover, F., 1965. A multiphase dual algorithm for the zero-one integer programming problem. *Operations Research* 13, 6, 879–919.

Glover, F., 1994. Optimization by ghost image processes in neural networks. *Computers & Operations Research* 21, 8, 801–822.

Glover, F., 2000. Multi-start and strategic oscillation methods-principles to exploit adaptive memory. In Laguna, M., Gonzalez-Velarde, J.L. (eds) *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*. Kluwer Academic, Dordrecht, pp. 1–24.

Glover, F., Kochenberger, G., 2003. *Handbook of Metaheuristics*. Kluwer, Boston, MA.

Glover, F., Laguna, M., 1997. *Tabu Search*. Kluwer Academic, Dordrecht.

Gomes da Silva, C., Clímaco, J., Figueira, J.R., 2004. Scatter search method for the bi-criteria multi-dimensional {0,1}-knapsack problem using surrogate relaxation. *Journal of Mathematical Modelling and Algorithms* 3, 3, 183–208.

Gomes da Silva, C., Clímaco, J., Figueira, J.R., 2006. A scatter search method for bi-criteria {0-1}-knapsack problems. *European Journal of Operational Research* 169, 373–391.

Gomes da Silva, C., Clímaco, J., Figueira, J.R., 2008. Core problems in bi-criteria {0,1}-knapsack problems. *Computers & Operations Research* 35, 1, 2292–2306.

Gomes da Silva, C., Figueira, J.R., Clímaco, J., 2007. Integrating partial optimization with scatter search for solving bi-criteria {0,1}-knapsack problems. *European Journal of Operational Research* 177, 1656–1677.

Haimes, Y., Lasdon, L., Wismer, D., 1971. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics* 1, 296–297.

Hansen, P., 1979. Bicriterion path problems. *Lecture Notes in Economics and Mathematical Systems* 177, 109–127.

Ishibuchi, H., Murada, T., 1998. A multi-objective genetic local search algorithm and its application to flow shop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews* 28, 3, 392–403.

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

Jaszkiewicz, A., 2000. Experiments done with the MOMHLIB: http://www-idss.cs.put.poznan.pl/jaszkiewicz/momhlib/. Technical report, Institute of Computing Science, Poznań University of Technology, Poznań, Poland.

Jaszkiewicz, A., 2000. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem—a comparative experiment. Technical Report RA-002/2000, Institute of Computing Science, Poznań University of Technology, Poznań, Poland.

Jaszkiewicz, A., 2001a. A comparative study of multiple-objective metaheuristics on the bi-objective set covering problem and the Pareto memetic algorithm. Technical Report RA-003/01, Institute of Computing Science, Poznań University of Technology, Poznań, Poland.

Jaszkiewicz, A., 2001b. Comparison of local search-based metaheuristics on the multiple-objective knapsack problem. *Foundations of Computing and Decision Sciences* 26, 1, 99–120.

Jaszkiewicz, A., 2002. 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.

Jaszkiewicz, A., 2004. On the computational efficiency of multiple objective metaheuristics. The knapsack problem case study. *European Journal of Operational Research* 158, 2, 418–433.

Jorge, J., Gandibleux, X. 2007. Nouvelles propositions pour la résolution exacte du problème de sac à dos bi-objectif unidimensionnel en variables binaires, février. FRANCORO V – MOSIM'07.

Kellerer, H., Pferschy, U., Pisinger, D., 2004. *Knapsack Problems*. Springer, Berlin.

Klamroth, K., Wiecek, M., 2000. Dynamic programming approaches to the multiple criteria knapsack problem. *Naval Research Logistics* 47, 1, 57–76.

Knowles, J., Corne, D., 1999. The Pareto archived evolution strategy: a new baseline algorithm for multiobjective optimisation. In *Proceedings of 1999 Congress on Evolutionary Computation*, July 6–9, 1999. IEEE Service Center, Washington, DC, pp. 98–105.

Knowles, J., Corne, D., 2000a. M-PAES: A memetic algorithm for multiobjective optimization. In *Proceedings of 2000 Congress on Evolutionary Computation, La Jolle, CA, July 6–9, 2000*, Vol. 1. IEEE Service Center, Piscataway, NJ, pp. 325–332.

Knowles, J., Corne, D., 2000b. A comparison of diverse approaches to memetic multiobjective combinatorial optimization. In *Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program*, Las Vegas, NV, July 8–12, 2000, pp. 103–108.

Kumar, R., Banerjee, N., 2006. Analysis of a multiobjective evolutionary algorithm on the 0-1 knapsack problem. *Theoretical Computer Science* 358, 1, 104–120.

Larranaga, P., Lozano, J.A., 2002. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation (Genetic Algorithms and Evolutionary Computation)*. Kluwer Academic, Boston, MA.

Laumanns, M., Thiele, L., Zitzler, E., 2006. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research* 169, 3, 932–942.

Li, H., Zhang, Q., Tsang, E., Ford, J.A., 2004. Hybrid estimation of distribution algorithm for multiobjective knapsack problem. In *Proceedings of the Fourth European Conference on Evolutionary Computation in Combinatorial Optimization, Coimbre, Portugal, April 5–7, 2004*, pp. 145–154.

Lin, S., Kernighan, B.W., 1973. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research* 21, 498–516.

Lust, T., 2009. New metaheuristics for solving MOCO problems: application to the knapsack problem, the traveling salesman problem and IMRT optimization. PhD thesis, University of Mons, Mons, Belgium.

Lust, T., Teghem, J., 2008. MEMOTS: a memetic algorithm integrating tabu search for combinatorial multiobjective optimization. *RAIRO: Operations Research* 42, 1, 3–33.

Lust, T., Teghem, J., 2010. Two-phase Pareto local search for the biobjective traveling salesman problem. *Journal of Heuristics* 16, 3, 475–510.

Martello, S., Toth, P., 1990. *Knapsack Problems*. Wiley, New York.

Martins, E.Q.V., Dos Santos, J.L.E., 1999. The labeling algorithm for the multiobjective shortest path problem. Technical Report 99/005 CISUC, Departamento de Matemática, Universidade de Coimbra.

Mavrotas, G., Figueira, J.R., Florios, K., 2009. Solving the bi-objective multi-dimensional knapsack problem exploiting the concept of core. *Applied Mathematics and Computation* 215, 7, 2502–2514.

Michalewicz, Z., Arabas, J., 1994. Genetic algorithms for the 0/1 knapsack problem. In Ras, Z.W., Zemankova, M. (eds) *Methodologies for Intelligent Systems Conference (ISMIS)*, Springer, Berlin, pp. 134–143.

Miettinen, K., 1999. *Nonlinear Multiobjective Optimization*. Kluwer Academic, Boston, MA.

Paquete, L., Chiarandini, M., Stützle, T., 2004. Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study. In Gandibleux, X., Sevaux, M., Sörensen, K., T'kindt, V. (eds) *Metaheuristics for Multiobjective Optimisation, Vol. 535 of Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin, pp. 177–199.

Pelikan, M., Sastry, K., Cantu-Paz, E., 2006. *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications (Studies in Computational Intelligence)*. Springer-Verlag New York, Inc., Secaucus, NJ.

Puchinger, J., Raidl, G.R., Pferschy, U., 2006. The core concept for the multidimensional knapsack problem. In Gottlieb, J., Raidl, G.R. (eds) *EvoCOP, Vol. 3906 of Lecture Notes in Computer Science*. Springer, Berlin, pp. 195–208.

Sato, H., Aguirre, H.E., Tanaka, K., 2007. Local dominance and local recombination in MOEAs on 0/1 multiobjective knapsack problems. *European Journal of Operational Research* 181, 3, 1708–1723.

Serafini, P., 1992. Simulated annealing for multi-objective optimization problems. In *Proceedings of the Tenth International Conference on Multiple Criteria Decision Making*, Vol. 1, Taipei, July 19–24, 1992 pp. 87–92.

Steuer, R., 1986. *Multiple Criteria Optimization: Theory, Computation and Applications*. John Wiley & Sons, New York.

Talbi, E-G., 2009. *Metaheuristics: From Design to Implementation*. Wiley-Blackwell, Hoboken, NJ.

Teghem, J., 2009. Multiple objective linear programming. In Bouyssou, D., Dubois, D., Prade, H., Pirlot, M. (eds) *Decision-making Process (Concepts and Methods)*, Wiley-ISTE, Hoboken, NJ, pp. 199–264.

Teghem, J., Kunsch, P., 1986. A survey of techniques for finding efficient solutions to multi-objective integer linear programming. *Asia-Pacific Journal of Operational Research* 3, 2, 95–108.

Tuyttens, D., 2006. Private communication.

Ulungu, E.L., 1993. Optimisation Combinatoire multicritère: Détermination de l'ensemble des solutions efficaces et méthodes interactives. PhD thesis, Université de Mons-Hainaut, Faculté des Sciences, Mons, Belgium.

Ulungu, E.L., Teghem, J., 1994. Multiobjective combinatorial optimization problems: a survey. *Journal of Multi-Criteria Decision Analysis* 3, 83–104.

Ulungu, E.L., Teghem, J., 1995. The two-phases method: an efficient procedure to solve biobjective combinatorial optimization problems. *Foundation of Computing and Decision Science* 20, 149–156.

Ulungu, E.L., Teghem, J., 1997. Solving multiobjective knapsack problems by a branch-and-bound procedure. In Climaco, J.N. (ed.) *Multicriteria Analysis*. Springer, Berlin, pp. 269–278.

Ulungu, E.L., Teghem, J., Fortemps, Ph., Tuyttens, D., 1999. MOSA method: A tool for solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis* 8, 4, 221–236.

Vazirani, V.V., 2001. *Approximation Algorithms*. Springer Verlag, Berlin.

Vianna, D.S., Arroyo, J.E.C., 2004. A GRASP algorithm for the multi-objective knapsack problem. In *XXIV International Conference of the Chilean Computer Science Society (SCC'04), IEEE Computer Society, Arica, Chile, November 2004*, pp. 69–75.

Visée, M., Teghem, J., Pirlot, M., Ulungu, E.L., 1998. Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem. *Journal of Global Optimization* 12, 139–155.

Voudouris, C., Tsang, E.P.K., 1999. Guided local search and its application to the travelling salesman problem. *European Journal of Operational Research* 113, 2, 469–499.

Zhang, C.W., Ong, H.L., 2004. Solving the biobjective zero-one knapsack problem by an efficient LP-based heuristic. *European Journal of Operational Research* 159, 3, 545–557.

Zitzler, E., 1999. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

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

Zitzler, E., Laumanns, M., Thiele, L., 2001. SPEA2: Improving the strength Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland.

Zitzler, E., Laumanns, M., Thiele, L., Fonseca, C.M., Grunert da Fonseca, V., 2002. Why quality assessment of multiobjective optimizers is difficult. In Langdon, W.B., Cantú-Paz, E., Mathias, K., Roy, R., Davis, D., Poli, R., Balakrishnan, K., Honavar, V., Rudolph, G., Wegener, J., Bull, L., Potter, M.A., Schultz, A.C., Miller, J.F., Burke, E., Jonoska, N. (eds) *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002)*, Morgan Kaufmann, San Francisco, CA, pp. 666–673.

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