Ecole Nationale Supérieure des Mines
SAINT-ETIENNE

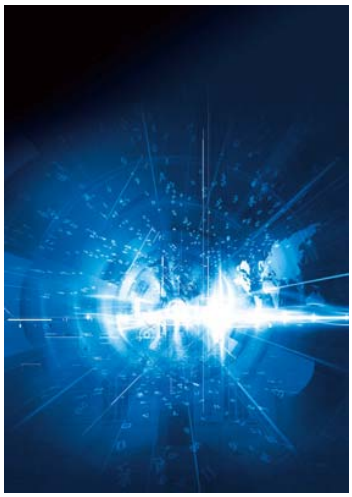
Industrial Engineering and Computer Sciences Division (G2I)

EVOLUTIONARY, CONSTRUCTIVE and HYBRID PROCEDURES FOR THE BIOBJECTIVE SET PACKING PROBLEM

X. DELORME, X. GANDIBLEUX, F. DEGOUTIN

Septembre 2005

RESEARCH REPORT
2005-500-011



MINISTÈRE DE L'ÉCONOMIE
DES FINANCES ET DE L'INDUSTRIE

Evolutionary, constructive and hybrid procedures for the biobjective set packing problem

Xavier DELORME[*], Xavier GANDIBLEUX[†] and Fabien DEGOUTIN[‡§]

Abstract: The bi-objective set packing problem is a multi-objective combinatorial optimization problem similar to the well-known set covering/partitioning problems. To our knowledge, this problem has surprisingly not yet been studied. In order to resolve a practical problems encountered in railway infrastructure capacity planning, procedures for computing a solution to this bi-objective combinatorial problem were investigated. Unfortunately, solving the problem exactly in a reasonable time using a generic solver (Cplex) is only possible for small instances. We designed three alternative procedures approximating solutions to this problem. The first is based on version 1 of the 'Strength Pareto Evolutionary Algorithm', which is a population based-metaheuristic. The second is an adaptation of the 'Greedy Randomized Adaptative Search Procedure', which is a constructive metaheuristic. As underlined in the overview of the literature summarized here, almost all the recent, effective procedures designed for approximating optimal solutions to multi-objective combinatorial optimization problems are based on a blend of techniques, called hybrid metaheuristics. Thus, the third alternative, which is the primary subject of this paper, is an original hybridization of the previous two metaheuristics. The algorithmic aspects, which differ from the original definition of these metaheuristics, are described, so that our results can be reproduced. The performance of our procedures is reported and the computational results for 120 numerical instances are discussed.

1 Introduction

1.1 The bi-objective set packing problem

This study examines approximate solutions to the bi-objective set packing problem (biSPP). The set packing problem (SPP) is a classic optimization problem, close to the set covering and set partioning problems (see, for example, Nemhauser and Wolsey [34]). The SPP can also be viewed as a special case of the multi-dimensional 0-1 knapsack problem, but this property is rarely useful for resolving the problem due to the large number of constraints in SPP instances. Surprisingly, the SPP has not received a lot of attention in the literature, and the biSPP even less. Compared to set covering and set partitioning problems, very few studies have looked at SPP resolution. The initial impetus for our research came from a real railway problem which has been formulated as a multi-objective SPP by Delorme [6]. The principal concerns in Delorme's study are related to evaluating infrastructure capacity in a railway network. However, this paper does not target resolution of this specific railway problem, but rather resolution of any biSPP instance.

The biSPP can be described as follows, using a mathematical model. Given a finite set $I = \{1, \ldots, n\}$ of valuated items and $\{T_j\}, j \in J = \{1, \ldots, m\}$ a collection of subsets of I, a solution is

[*]Centre Génie Industriel et Informatique, École Nationale Supérieure des Mines de Saint-Etienne, 158 cours Fauriel, F42023 Saint-Etienne Cedex 2 - FRANCE — Delorme@emse.fr

[†]LINA, Université de Nantes, 2 rue de la Houssinière BP 92208, F44322 Nantes Cedex 03 - FRANCE — Xavier.Gandibleux@univ-nantes.fr

[‡]INRETS-ESTAS, 20 rue Élisée Reclus, F59650 Villeneuve d'Ascq - FRANCE — Fabien.Degoutin@inrets.fr

[§]LAMIH-ROI - UMR CNRS 8530, Université de Valenciennes, Campus "Le Mont Houy", F59313 Valenciennes Cedex 9 - FRANCE

a subset $R \subseteq I$, such that $|T_j \cap R| \leq 1, \forall j \in J$. The objective of the set packing problem with two objectives $q \in Q = \{1, 2\}$ is to "maximise" the total value of the solution obtained. The biSPP model is summarized as (1):

$$\left[\begin{array}{rrcll} \text{“max”} z^q = \sum_{i \in I} c_i^q x_i & & & & \forall q \in Q \\ \text{sc} \quad \sum_{i \in I} t_{i,j} x_i & \leq & 1 & & \forall j \in J \\ x_i & \in & \{0,1\} & & \forall i \in I \\ t_{i,j} & \in & \{0,1\} & & \forall i \in I, \forall j \in J \end{array}\right] \tag{1}$$

given:

- a vector (x_i) where $x_i = 1$ if $i \in R$, and 0 otherwise,
- a vector (c_i^q) where $c_i^q =$ the value of item i for the objective function q,
- a matrix $(t_{i,j})$ where $t_{i,j} = 1$ if $i \in T_j$, and 0 otherwise.

The biSPP is a multi-objective combinatorial optimization (MOCO) problem. In Ehrgott and Gandibleux's survey [9], no paper considering this problem is mentioned. According to Garey and Johnson [23], the problem is strongly NP-Hard even in the mono-objective case.

Let $X = \{x \mid \sum_{i \in I} t_{i,j} x_i \leq 1 \ \forall j \in J;\ x_i \in \{0,1\} \ \forall i \in I;\ t_{i,j} \in \{0,1\} \ \forall i \in I, \forall j \in J\}$ denote the feasible solutions, or the *decision space*. We consider that solutions for (1) are optimal if they are *efficient*. In other words, a feasible solution $x \in X$ is deemed efficient if there is no $x' \in X$, such that x' dominates x, in the sense of *Pareto dominance* (i.e., $z^q(x') \geq z^q(x)$ for all $q \in Q$ with $z^q(x') > z^q(x)$ for some q; the notation $x' \succ x$ will be considered later). This means that no solution is at least as good as x for all objectives, and none is strictly better for at least one objective. Efficiency refers to solutions x in the *decision space*. We denote the image of the feasible set in the *objective space* as $Z = z(X)$. In the objective space, we consider the notion of *non-dominance*: if x is an efficient solution, then $z(x) = (z^1(x), \ldots, z^{|Q|}(x))$ is a non-dominated vector. The set of efficient solutions is X_E; the set of non-dominated vectors, Z_N. Z_N is also called the *non-dominated frontier*, or the trade-off surface. As in most MOCO problems, X_E is composed of two solutions subsets: X_{SE}, the set of *supported* efficient solutions, and X_{NE}, the set of *non-supported* efficient solutions.

When multiple feasible solutions $x, x' \in X$ map to the same non-dominated point $z(x) = z(x')$, the solutions are said to be *equivalent*. A *complete set* X_E is a set of efficient solutions, such that all $x \in X \setminus X_E$ are either dominated or equivalent to at least one $x \in X_E$. This means that for each nondominated point $z \in Z_N$, there is at least one $x \in X_E$ such that $z(x) = z$. A *minimal complete set* X_{E_m} is a complete set without equivalent solutions (see Hansen [26]). All complete sets contain a minimal complete set. The *maximal complete set* X_{E_M} is a complete set of all solutions, including all the equivalent solutions (i.e., all $x \in X \setminus X_{E_M}$ are dominated, see Przybylski et al. [36]).

In this study, the minimum complete set of efficient solutions was computed with Cplex for all the instances considered (see section 5.1 for details about these 120 instances). But given the difficulty of this problem, its exact resolution was not possible within a reasonable period of time: the solutions were obtained after several days of computation. From the exact solution sets, three main ideas are noteworthy. First, in general, the biSPP has a low number of efficient solutions, compared to other problems, such as the knapsack with two objectives for which Visée et al. [43] report a huge number of solutions. Second, Degoutin [4] and Ehrgott and Gandibleux [11] have shown that the bound sets obtained using linear relaxation and a greedy algorithm are very far from the efficient frontier and do not provide useful information about the frontier (see figure 1). Third, in general, the shape of the efficient frontier observed in this study was not globally convex, which is not true of the knapsack or the assignment problems with two objectives.

The amount of time needed by Cplex to compute solutions for the instances led us to investigate the possibility of solving the biSPP using heuristics. For practical reasons related to our railway planning problem, our goal was to obtain a good approximation of the entire efficient set – denoted X_{PE}, the set of potentially efficient solutions – within a reasonable computing time.

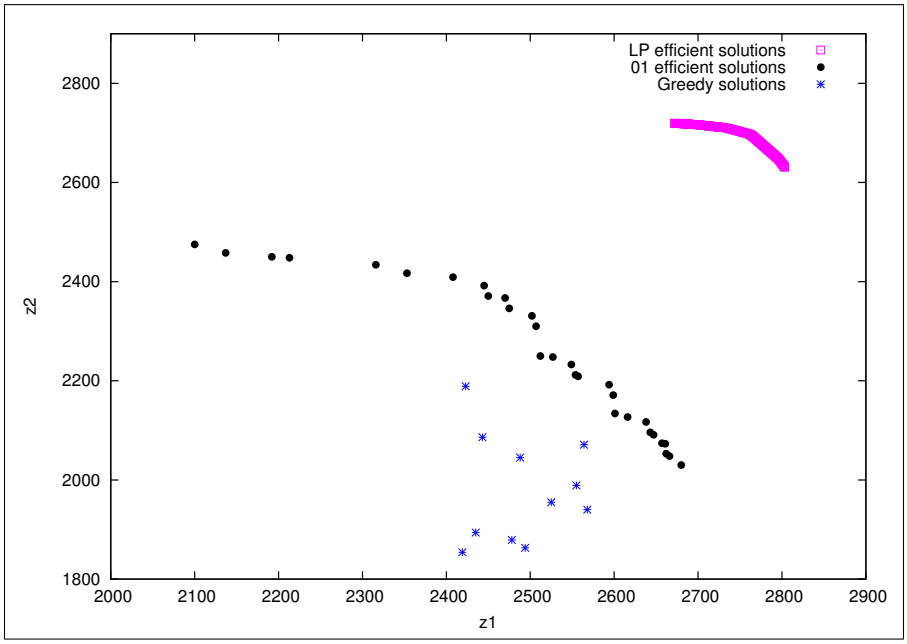


Figure 1: Example of efficient frontier and bound sets

1.2 Multi-Objective metaheuristics (MOMH) and MOCO problems

Although the first adaptation of metaheuristic techniques for solving multi-objective optimization problems was introduced 20 years ago, the MOMH field has clearly mushroomed over the last ten years (see Ehrgott and Gandibleux [10] for a detailled analyze of the literature on that topic). Historically, pioneer approximation methods for multi-objective problems first appeared in 1984. The most important are listed here in order of publication: Genetic Algorithms (GA, Schaffer 1984 [38]), Artificial Neural Networks (ANN, Malakooti et al. 1990 [30]), Simulated Annealing (SA, Serafini 1992 [39]), and Tabu Search (TS, Gandibleux et al. 1997 [18]). These pioneer methods have one common characteristic: they are inspired either by *Evolutionary Algorithms*, such as VEGA (Vector Evaluated Genetic Algorithm, see Schaffer [38]) or MOGA (Multi-Objective Genetic Algorithm, see Fonseca and Fleming [14]), or by *Neighborhood Search Algorithms*, such as MOSA (Multi-Objective Simulated Annealing, see Ulungu [41]) or MOTS (Multi-Objective Tabu Search, see Gandibleux et al. [18]).

Evolutionary Algorithms (EA) manage a solution population $\mathcal{P}$ rather than a single feasible solution. In general, they start with an initial population and then improve approximation quality by combining the principles of self adaptation (i.e. independent evolution, such as the mutation strategy in genetic algorithms), and cooperation (i.e. the exchange of information between individuals, such as the "pheromone" used in ant colony systems). Because the whole population contributes to the evolutionary process, the generation mechanisms run parallel along the frontier, and thus these methods are also called *global convergence-based methods*. This global convergence makes population-based methods very attractive for solving multi-objective problems.

In *Neighborhood Search Algorithms (NSA)* solutions are generated from one individual, who is a current solution x_n, and the neighbors of that solution $\{x\} \subseteq \mathcal{N}(x_n)$. Using a local aggregation mechanism for the objectives (often based on a weighted sum), a weight vector $\lambda \in \Lambda$, and an initial solution x_0, the procedure iteratively projects the neighbors into the objective space in a search direction λ by optimizing the corresponding parametric single objective problem. A local approximation of the non-dominated frontier is obtained using archives of the successive potentially efficient solutions detected. This generation mechanism is sequential along the frontier, producing a local convergence towards the non-dominated frontier, and so such methods are called *local convergence-based methods*. The procedure is repeated in diverse search directions in order to completely approximate the non-dominated frontier. NSAs are well-known for their ability to locate the non-dominated frontier, but they require more diversification than EA to cover the efficient frontier completely.

The first approximation methods proposed for MOCO problems were "pure" NSA strategies and were straightforward extensions of well-known mono-objective metaheuristics for dealing with non-dominated points (e.g. MOSA, MOTS or the Sun method [40]). The methods that followed the pioneers were influenced by two important observations. The first is that NSAs converge rapidly towards some efficient solutions, though they must be guided along the non-dominated frontier, while EAs are quite capable of maintaining a diverse, well-covered solution population along the non-dominated frontier though they often converge too slowly towards that frontier. Naturally, methods have been proposed that combine components of both approaches in order to take advantage of the positive aspects of both EAs and NSAs. These methods are called *hybrid algorithms* for MOPs.

The second observation is that, due to their specific combinatorial structure, MOCO problems contain information that can be advantageously exploited during the approximation process. Single objective combinatorial optimization is a very active field of research, and many combinatorial structures are well understood. Thus combinatorial optimization offers a useful source of knowledge for multi-objective optimization. This knowledge (e.g., cuts for reducing the search space) is increasingly taken into account in the design of efficient approximation methods for particular MOCO (like for the bi-objective knapsack problem in Gandibleux and Fréville [17]). It is not surprising to see an evolutionary algorithm – for global convergence – coupled with a tabu search algorithm – for the exploitation of the combinatorial structure – within one approximation method.

1.3 Three resolution procedures for the biSPP

Despite this recent work on multi-objective metaheuristics, no other metaheuristic other than those mentioned above has, to our knowledge, been proposed for the biSPP. This paper is a first response to this lack. We approached the problem in three steps:

1. An adaptation of a *classic multi-objective evolutionary metaheuristic* was created for solving the biSPP. Such a metaheuristic is considered to be generic solver that can be easily customized to tackle a specific problem. Among the existing metaheuristics, SPEA (Strength Pareto Evolutionary Algorithm, see Zitzler [44]) was chosen because this algorithm produced interesting results on a similar problem, the bi-objective multi-dimensional knapsack problem. In addition to adapting SPEA for the biSPP, the basic SPEA scheme was also enriched to make the algorithm slightly more aggressive. Called Aggressive-SPEA in this paper, this new algorithm was designed to collect all potentially efficient solutions, and to perform a multidirectional repair procedure among other things. A preliminary version of this algorithm has already been presented by Gandibleux et al. [15] and Delorme et al. [7].

2. An *efficient single-objective procedure* for the SPP embedded in a parametric procedure was also used to solve the biSPP (see Delorme et al. [7] for a preliminary version). Such an approach has been mentioned by Jaszkiewicz [29] as a competitive alternative to multi-objective metaheuristics. Delorme et al. [8] proposed an algorithm derived from the GRASP (Greedy Randomized Adaptative Search Procedure, see Féo and Resende [12]) metaheuristic for the SPP. The procedure, named λ-GRASP, was evaluated during this study.

3. A *hybrid procedure*, combining the Aggressive-SPEA with the λ-GRASP procedures according to an original scheme, was also evaluated. The original version 1 of SPEA is reputed to have difficulty approximating the extreme part of the efficient frontier. The λ-GRASP, on the other hand, has difficulty producing a sufficiently dense set of potential efficient solutions without being forced to perform a huge number of λ. Thus, combining the two seems a natural solution to their individual deficiencies. λ-GRASP produces a good cover of the efficient frontier which become the initial population used by Agressive-SPEA to complete (in terms of density and distribution) the approximation along the efficient frontier.

The ability of these three procedures (the Aggressive-SPEA, the λ-GRASP, the hybrid procedure) to solve a wide set of biSPP whose exact solutions are known was compared. This comparison demonstrates the efficiency of the individual procedures to compute good solutions within a reasonable time. Our experimental results confirm the superior efficiency of the hybrid procedure, due to the complementarity of the Aggressive-SPEA and λ-GRASP procedures.

The principles behind the A-SPEA, λ-GRASP and hybrid algorithms, respectively, are presented in detail in sections 2, 3 and 4, along with the algorithmic description of the different operational procedures. Section 5 describes and comments on the experimental instances, and reports and discusses the numerical results. Section 6 concludes the discussion and presents several perspectives for this work.

2 Description of the Aggressive-SPEA for the biSPP

Proposed by Zitzler [44], SPEA (Strength Pareto Evolutionary Algorithm) is a multi-objective evolutionary metaheuristic based on a Pareto dominance-based evaluation. The main caracteristics of the SPEA are: (1) the non-dominated solutions are stored in an external set ($\bar{P}$) at each iteration, (2) the concept of Pareto dominance is used for the scalar fitness assignment, and (3) evolutionary operators (crossover and mutation) are used.

It should be noted that an improved version of this metaheuristic, SPEA2 (see Zitzler et al. [45]), was recently proposed after we had begun our own work. SPEA2 incorporates a new fitness assignment strategy, a density estimation technique, and an enhanced truncation method. It improves the observed performances, notably for problems in higher dimensional objective spaces (i.e. with many objectives). However, some of these improvements, such as the truncation technique, are not realy useful for the biSPP since the instances of this problem contain a low number of efficient solutions.

2.1 General principles of the Aggressive-SPEA

The first procedure for solving biSPP, named Aggressive-SPEA (A-SPEA), is an enhancement of SPEA. This procedure was briefly introduced by Gandibleux et al [15]; it is described in its entirety for the first time in the following paragraphs. This enhancement can be seen as a generic metaheuristic that can be applied to several combinatorial problems. The overall approach is described below (see algorithm 1).

Initial population First, an initial population of size *PopSize* is computed for different weight values of the scalarized function (see algorithm 2). *PopSize* was set to 50 in our experiments. A greedy algorithm is performed on the *PopSize*'s different search directions corresponding to the scalarized functions of the objective functions. A specific *Saturation* function for the problem considered must be defined here (see section 2.2). Then, if the expected population size is not reached, for example because identical solutions are found in several search directions, some randomized saturated solutions are added. A specific *RandomSaturation* function for the problem considered must be defined here (see section 2.2). All these solutions are different in terms of their decision vector (x_i), thus the initial population can not contain any identical solutions.

Fitness value At the beginning of each iteration loop, the set of potentially efficient solutions (denoted by $\bar{P}$ in the A-SPEA, a synonym of X_{PE} for the general case) is updated. Then, individuals from the population of the dominated individuals population P, and the population of potentially efficient solutions $\bar{P}$, are evaluated interdependently and are assigned fitness values. The fitness value is determined by a function (see algorithm 3) based on Pareto dominance and, to preserve diversity, a Pareto niching method:

- the fitness of a non-dominated solution is determined by the number of solutions that it dominates, and
- the fitness of a dominated solution is determined only by the fitness of the non-dominated solution(s) that dominate it.

Selection Two individuals are selected randomly from the population of dominated individuals and the population of potentially efficient solutions if they are not already in the mating pool (P'). These individuals compete in a binary tournament, which determines the best candidate according to its fitness value. This individual is then added to the mating pool (function *PopSelection*, see algorithm 3).

function $A{-}SPEA(PopSize, CrossoverProbability, MutationProbability, stoppingCondition)$

$P \leftarrow PopInitialize(PopSize)$

$\overline{P} \leftarrow \emptyset$

repeat

$\overline{P}' \leftarrow \{s \in \overline{P} \cup P, \nexists s' \in \overline{P} \cup P, s' \succ s\}$

$P \leftarrow P \cup \{s \in \overline{P}, s \notin \overline{P}'\} \setminus \overline{P}'$

$\overline{P} \leftarrow \overline{P}'$

$P' \leftarrow PopSelection(PopSize, P, \overline{P})$

$P'' \leftarrow \emptyset$

while $|P'| > 1$ loop

$(s_1, s_2) \leftarrow RandomSelect(s \in P'), s_1 \neq s_2$

$P' \leftarrow P' \setminus \{s_1, s_2\}$

if $CrossoverProbability$

$(s_3, s_4) \leftarrow Crossover(s_1, s_2)$

$(s_3, s_4) \leftarrow (Repair(s_3), Repair(s_4))$

$P'' \leftarrow P'' \cup \{Saturation(s_3, 0), Saturation(s_3, \frac{z_1(s_3)}{z_1(s_3)+z_2(s_3)}),$
$Saturation(s_3, 1), Saturation(s_4, 0),$
$Saturation(s_4, \frac{z_1(s_4)}{z_1(s_4)+z_2(s_4)}), Saturation(s_4, 1)\}$

else

$P'' \leftarrow P'' \cup \{s_1, s_2\}$

endIf

endWhile

$\overline{P}' \leftarrow \{s \in P'', \nexists s' \in \overline{P} \cup P'', s' \succ s\}$

$P'' \leftarrow P'' \cup P'$

$P \leftarrow \emptyset$

while $P'' \neq \emptyset$ loop

$s_1 \leftarrow RandomSelect(s \in P'')$

$P'' \leftarrow P'' \setminus \{s_1\}$

$s_1 \leftarrow Repair(Mutation(s_1, MutationProbability))$

$P \leftarrow P \cup \{Saturation(s_1, 0), Saturation(s_1, \frac{z_1(s_1)}{z_1(s_1)+z_2(s_1)}), Saturation(s_1, 1)\}$

endWhile

$P \leftarrow P \cup \overline{P}'$

until $stoppingCondition$

$\overline{P} \leftarrow \{s \in \overline{P} \cup P, \nexists s' \in \overline{P} \cup P, s' \succ s\}$

$\overline{P}' \leftarrow \emptyset$

$\overline{P}' \leftarrow \overline{P}' \cup \{LocalSearch(s, \overline{P})\}, \forall s \in \overline{P}$

$\overline{P} \leftarrow \{s \in \overline{P} \cup \overline{P}', \nexists s' \in \overline{P} \cup \overline{P}', s' \succ s\}$

$\overline{P} \leftarrow \overline{P} \cup \{AggressiveLocalSearch(RandomSelect(s \in \overline{P}, z_1(s) = \max_{s' \in \overline{P}}(z_1(s'))), z_1)\}$

$\overline{P} \leftarrow \overline{P} \cup \{AggressiveLocalSearch(RandomSelect(s \in \overline{P}, z_2(s) = \max_{s' \in \overline{P}}(z_2(s'))), z_2)\}$

return $\overline{P}$

end $A{-}SPEA$

Algorithm 1: The Aggressive-SPEA algorithm

function $PopInitialize(PopSize)$
$x_i \leftarrow 0, \forall i \in I$
$P \leftarrow \emptyset; \quad P \leftarrow P \cup \{Saturation(x, \frac{i-1}{PopSize-1})\}, \forall i \in \{1, \ldots, PopSize\}$
while $(|P| < PopSize)$ loop
$\quad P \leftarrow P \cup \{RandomSaturation(x)\}$
endWhile
return P
end $PopInitialize$

Algorithm 2: The initial population generator algorithm

function $PopSelection(PopSize, P, \overline{P})$
$Fitness_s \leftarrow \frac{|\{s' \in P, s \succeq s'\}|}{1+|P|}, \forall s \in \overline{P}$
$Fitness_s \leftarrow 1 + \sum_{s' \in \overline{P}, s' \succeq s} Fitness_s, \forall s \in P$
$P' \leftarrow \emptyset$
while $|P'| < min(PopSize, |P \cup \overline{P}|)$ loop
$\quad (s_1, s_2) \leftarrow RandomSelect(s \in P \cup \overline{P} \setminus P')$
$\quad$ if $Fitness_{s_1} < Fitness_{s_2}$
$\quad\quad P' \leftarrow P' \cup s_1$
$\quad$ else
$\quad\quad P' \leftarrow P' \cup s_2$
$\quad$ endIf
endWhile
return P'
end $PopSelection$

Algorithm 3: The population selection algorithm

Evolutionary operators Two parents are chosen to generate two offspring using a one point crossover operator with a point randomly selected from an ordered set of the biSPP variables (see algorithm 4). This operator was applied with a probability $CrossoverProbability$ equal to 80% in our experiments. Then a mutation operator (see algorithm 5) switches the value of each variable in each solution in the population P' according to the probability $MutationProbability$, which was equal to 4% in our experiments.

function $Crossover(s_1, s_2)$
$i^* \leftarrow RandomSelect(i \in I \setminus \{1, n\})$
$s_3(i) \leftarrow s_1(i), \forall i \in \{1, \ldots, i^*\}; \quad s_3(i) \leftarrow s_2(i), \forall i \in \{i^*, \ldots, n\}$
$s_4(i) \leftarrow s_2(i), \forall i \in \{1, \ldots, i^*\}; \quad s_4(i) \leftarrow s_1(i), \forall i \in \{i^*, \ldots, n\}$
return (s_3, s_4)
end $Crossover$

Algorithm 4: The crossover algorithm

Repair and saturate The solutions generated by the crossover and mutation operators may be non-feasible, in which case, the solution must be repaired so that only feasible solutions are in the population (see section 2.2 for the description of the specific algorithm used for the biSPP). The potentially efficient solutions are always maintained in the population, especially those generated by the crossover operator, which was not true for the original SPEA. After each repair, a *Saturation* function is also applied to the individuals in three search directions: z_1, z_2, and a combination of the two, computed according to the value of the current solution for the two objectives. The saturation on $z1$ and $z2$ permits the algorithm's performance to be

function $Mutation(s, MutationProbability)$
  for $i \in I$ loop
    if $MutationProbability$
      $s(i) \leftarrow 1 - s(i)$
    endIf
  endFor
  return s
end $Mutation$

Algorithm 5: The mutation algorithm

improved for extremal solutions (the best solution for $z1$ and for $z2$). These saturations were not part of the original SPEA algorithm.

Local search Two local search components, designed specifically for the problem under consideration, were also added to the basic SPEA (see Local search components for the biSPP in section 2.2). There were two reasons for this addition: to prevent the holes in the approximation of the efficient frontier that were observed during preliminary experiments, and to improve the ability of the A-SPEA to detect good extremal solutions. According to Moscato [32] A-SPEA falls in the class of memetic algorithms. The first local search is applied to every non-dominated solution, and yields all the non-dominated solutions for the neighborhood of each initial solution. It is applied many times and consumes little CPU time. The second local search is mono-objective. It is only applied to the extremal solutions, with the goal of obtaining more extremal solution by applying an aggressive local search in the direction $z1$ (resp. $z2$) from the current best solution for $z1$ (resp. $z2$).

2.2 Specific components of the A-SPEA for the biSPP

As indicated above, some specific components must be defined in order to obtain an efficient algorithm for a particular problem. The components considered for the biSPP are described below:

Repair component Any non-feasible solution can be repaired (see algorithm 6) using a greedy repair strategy, in which the best variable that can be set to 0 is chosen at each iteration, according to an evaluation based on an objective function (a combination of the two objectives) and the number of constraints upon which this variable acts. This is repeated as long as the solution remains non-feasible.

function $Repair(x)$
  $\lambda_{Rep} \leftarrow z_1(x)/(z_1(x) + z_2(x))$
  $J_{Rep} \leftarrow \{j \in J, \sum_{i \in I} t_{i,j} x_i > 1\}$
  $I_1 \leftarrow \{i \in I, x_i = 1 \wedge (\exists j \in J_{rep}, t_{i,j} = 1)\}$
  $Eval_i \leftarrow (\lambda_{Rep} c_i^1 + (1 - \lambda_{Rep}) c_i^2)/(\sum_{j \in J} t_{i,j}), \forall i \in I_1$
  while $(J_{Rep} \neq \emptyset)$ loop
    $i^* \leftarrow RandomSelect(i \in I_1, Eval_i = \min_{k \in I_1}(Eval_k))$
    $x_{i^*} \leftarrow 0$
    $I_1 \leftarrow I_1 \setminus \{i^*\}$
    $I_1 \leftarrow I_1 \setminus \{i \in I_1, \nexists j \in J_{Rep}, t_{i,j} = 1\}$
    $J_{Rep} \leftarrow J_{Rep} \setminus \{j \in J_{Rep}, \sum_{i \in I} t_{i,j} x_i \leq 1\}$
  endWhile
  return x
end $Repair$

Algorithm 6: The repair algorithm

Saturation components A solution is saturated when no variable can be set to 1 without losing feasibility. For the *Saturation* algorithm (see algorithm 7), the best variable that can be set to 1, of those that were set to 0, is chosen in the same way as the variables in the repair function were chosen. For the *RandomSaturation* algorithm (see algorithm 8), the variables that can be set to 1 are randomly selected .

function $Saturation(x, \lambda)$
  $I_0 \leftarrow \{i \in I, x_i = 0\}$
  $Eval_i \leftarrow (\lambda c_i^1 + (1-\lambda)c_i^2)/(\sum_{j \in J} t_{i,j}), \forall i \in I_0$
  $I_{Sat} \leftarrow \{i \in I_0, \forall j \in J, t_{i,j} = 1 \Rightarrow \sum_{k \in I} t_{k,j} x_k = 0\}$
  while $(I_{Sat} \neq \emptyset)$ loop
    $i^* \leftarrow RandomSelect(i \in I_{Sat}, Eval_i = \max_{k \in I_{Sat}} (Eval_k))$
    $x_{i^*} \leftarrow 1$
    $I_{Sat} \leftarrow I_{Sat} \setminus \{i^*\}$
    $I_{Sat} \leftarrow I_{Sat} \setminus \{i \in I_{Sat}, \exists j \in J, t_{i,j} + t_{i^*,j} > 1\}$
  endWhile
  return x
end *Saturation*

Algorithm 7: The saturation algorithm

function $RandomSaturation(x)$
  $I_0 \leftarrow \{i \in I, x_i = 0\}$
  $I_{Sat} \leftarrow \{i \in I_0, \forall j \in J, t_{i,j} = 1 \Rightarrow \sum_{k \in I} t_{k,j} x_k = 0\}$
  while $(I_{Sat} \neq \emptyset)$ loop
    $i^* \leftarrow RandomSelect(i \in I_{Sat})$
    $x_{i^*} \leftarrow 1$
    $I_{Sat} \leftarrow I_{Sat} \setminus \{i^*\}$
    $I_{Sat} \leftarrow I_{Sat} \setminus \{i \in I_{Sat}, \exists j \in J, t_{i,j} + t_{i^*,j} > 1\}$
  endWhile
  return x
end *RandomSaturation*

Algorithm 8: The random saturation algorithm

Local search components These algorithms are based on a $1-1$ exchange neighbourhood (see algorithm 9) which means that one variable is switched from 1 to 0 when another variable is switched from 0 to 1. The *LocalSearch* function (see algorithm 10) is applied to every non-dominated solution, whereas the *AggressiveLocalSearch* function (see algorithm 11) is applied only to extremal solutions. This second function is a variation in which $1-1$ exchanges are performed iteratively on the current solution and its best neighbour until no better solutions can be found in the neighborhood.

3 Description of the λ-GRASP for the biSPP

The second step in our approach is based on an heuristic algorithm for the mono-objective SPP, derived from the GRASP metaheuristic (Greedy Randomized Adaptative Search Procedure). Proposed by Féo and Resende [12]), GRASP has been applied to a wide range of optimization problems, including academic and industrial problems in scheduling, routing, logic, partitioning, location and layout, graph theory, assignment, manufacturing, transportation, telecommunications, electrical power systems, and VLSI design. An extensive annotated GRASP bibliography has been proposed by Festa and Resende [13].

function $1-1exchanges(x)$
    $I_0 \leftarrow \{i \in I, x_i = 1\}$
    $\mathcal{N} \leftarrow \emptyset$
    for $i_0 \in I_0$ loop
        $s \leftarrow x$
        $s_{i_0} \leftarrow 0$
        $I_1 \leftarrow \{i \in I, \forall j \in J, t_{i,j} = 1 \Rightarrow \sum_{k \in I} t_{k,j} s_k = 0\}$
        for $i_1 \in I_1$ loop
            $s' \leftarrow s$
            $s'_{i_1} \leftarrow 1$
            $\mathcal{N} \leftarrow \mathcal{N} \cup \{s'\}$
        endFor
    endFor
    return $\mathcal{N}$
end $1-1exchanges$

Algorithm 9: The 1-1 exchange algorithm

function $LocalSearch(x, X_{PE})$
    $\mathcal{N} \leftarrow 1-1exchanges(x)$
    $\mathcal{N}_{Sat} \leftarrow \emptyset$
    for $s \in \mathcal{N}$ loop
        if $\nexists s' \in \mathcal{N}_{Sat} \cup X_{PE}, s' \succ s\}$
            $\mathcal{N}_{Sat} \leftarrow \mathcal{N}_{Sat} \cup \{Saturation(s, z_1(s)/(z_1(s) + z_2(s)))\}$
        endIf
    endFor
    return $\{s \in \mathcal{N}_{Sat}, \nexists s' \in \mathcal{N}_{Sat}, s' \succ s\}$
end $LocalSearch$

Algorithm 10: The local search algorithm

function $AggressiveLocalSearch(x, z)$
    $s^* \leftarrow x$
    $\mathcal{N}_{Pop} \leftarrow \emptyset$
    repeat
        $\mathcal{N} \leftarrow 1-1exchanges(s^*)$
        $\mathcal{N}_{Sup} \leftarrow \emptyset$
        for $s \in \mathcal{N}$ loop
            if $z(s) > z(s^*)$
                $\mathcal{N}_{Sup} \leftarrow \mathcal{N}_{Sup} \cup \{Saturation(s, z)\}$
            endIf
        endFor
        $\mathcal{N}_{Pop} \leftarrow \mathcal{N}_{Pop} \cup \mathcal{N}_{Sup}$
        $s^* \leftarrow RandomSelect(s \in \mathcal{N}_{Sup}, z(s) = \max_{s' \in \mathcal{N}_{Sup}} z(s'))$
    until $\mathcal{N}_{Sup} = \emptyset$
    return $\{s \in \mathcal{N}_{Pop}, \nexists s' \in \mathcal{N}_{Pop}, s' \succ s\}$
end $AggressiveLocalSearch$

Algorithm 11: The aggressive local search algorithm

Basically, GRASP is a multistart two-phase metaheuristic for combinatorial optimization problems. The first phase is a construction phase that builds an initial solution using a greedy randomized procedure, whose randomness allows solutions to be obtained in different areas of the solution space. The second phase is a local search phase that improves these solutions. This two-phase process is reiterative. For a general overview of GRASP, see the article by Pitsoulis and Resende [35]. As presented and discussed in Resende and Ribeiro [37], several new components have extended the GRASP scheme (e.g. reactive GRASP, parameter variations, bias functions, memory and learning, improved local search, path relinking, and hybrids).

3.1 General principles of the λ-GRASP

function $\lambda - GRASP(stoppingCondition, SearchDirection)$
  $P \leftarrow \emptyset$
  $graspStopCnd \leftarrow ChoosingCondition(stoppingCondition, SearchDirection)$
  for $\lambda \in \{1, \ldots, SearchDirection\}$ loop
    $P_t \leftarrow \{GRASP(\lambda c^1 + (1-\lambda)c^2, graspStopCnd)\}$
    $P \leftarrow P \cup \{i \in P_t, \nexists j \in P, j \succ i\}$
    $P \leftarrow P \setminus \{i, \exists j \in P, j \succ i\}$
  endFor
  return P
end $\lambda - GRASP$

Algorithm 12: The bi-objective λ-GRASP algorithm

Our idea is to embed the GRASP heuristic in a parametric procedure so that it can be performed successively and independently in a wide set of search directions obtained using a scalarization function. The number of directions is determined by the parameter *SearchDirection*. The value of this parameter must be high enough to avoid poor coverage of the efficient frontier obtained, but low enough to allow the time needed for each resolution and to keep the obtained solutions near the efficient frontier. Preliminary experiments yielded a compromise value equal to 20. This low value is mainly due to the low number of efficient solutions for the biSPP. A *ChoosingCondition* function must be designed to compute a stopping condition for a single mono-objective GRASP resolution based on the stopping condition of the overall process and the number of search directions to be considered. For example, if the stopping condition is a time limit, the following formula is possible:

$$graspStopCnd \leftarrow \frac{stoppingCondition}{SearchDirection}$$

Because the resolution method used is an heuristic method, applying such a scalarization function does not prevent us from obtaining non-supported efficient solutions, whereas using an exact algorithm would do so. All the potentially efficient solutions produced throughout the computation of all λ-GRASP phases are maintained in the solution population, even those that are not the best according to the scalarized objective function under consideration.

3.2 Specific components of the GRASP for the SPP

Our greedy randomized procedure was proposed and fully described by Delorme et al [8]. It builds a solution from the trivial feasible solution, $x_i = 0, \forall i \in I$. Some variable values are set to 1, thus maintaining a feasible solution. Changes are made to only one variable at each iteration. To increase the objective function, the variables that have a minimum number of constraints and a maximum value are prioritized, but the choice among the variables at the top of the priority list is random according to a threshold parameter $\alpha \in [0, 1]$. Changes stop when a variable can not be set to 1 without the solution becoming non-feasible. The local search procedure is based on $0-1$ exchanges, $1-1$ exchanges, $2-1$ exchanges and $1-2$ exchanges. The $k-p$ exchange neighbourhood of a solution x is the set of solutions obtained from x by changing the value of k variables from 1 to 0, and changing p variables from 0 to 1. This procedure was implemented using a first-improving strategy (i.e. the first neighbour whose

value is better than the current solution is selected). Whenever an exchange is accepted, the local search re-starts at this new solution. The local search stops when no further improving exchanges are possible.

Three extensions of GRASP were also integrated into this heuristic. The first extension, called *reactive GRASP*, self-adjusts the parameter α according to the quality of the solutions previously obtained. At the beginning of the process, 5 values – 0.0, 0.5, 0.75, 0.9 and 1.0 – are considered with an equal probability for the parameter α. The second extension adds an intensification phase based on *path relinking* (see Glover and Laguna [24]) at each GRASP iteration. With this procedure, paths from one elite solution to another are generated in the solution space, and explored to obtain better solutions. Finally, the third extension completes the variable evaluation with a learning process that penalizes the variables involved in often saturated constraints, thus acting as a diversification process.

4 Hybrid algorithm for the biSPP

The third step in our approach is based on an original hybridization of the two previous algorithms. Modern approximation methods for MOCO problems appear more and more frequently as problem-oriented techniques (i.e., by selecting components that are advantageously combined to create an algorithm that can tackle the problem in the most efficient way). By nature, such an algorithm is a hybrid, including evolutionary components, neighborhood search components, and problem-dependent components (see Gandibleux and Ehrgott [16]).

4.1 Hybrid approaches reported in the literature

The hybridizations introduced in multi-objective approximation methods are, in chronological order:

1. **EA components integrated into NSA**.
   The use of a population of individuals provides global information about the current approximation and lets that information *drive local search processes* in order to "guarantee" good coverage of the non-dominated frontier. Using, for example, mechanisms based on notions of repulsion between non-dominated points, the search is guided toward sub-areas of the frontier that (i) contain a high density of solutions, or (ii) have not yet been explored. This is the principle behind the PSA method proposed by Czyzak and Jaszkiewicz [3] and the TAMOCO method proposed by Hansen [25].

2. **EA as master strategy, NSA as secondary strategy**.
   Here, EA pilots the search procedure, and activates an NSA. The main idea is *to make the evolutionary algorithm very aggressive* at improving the good solutions resulting from the evolutionary operators for as long as possible. The NSA can, for example, be a depth-first search method, or a basic (or truncated) tabu search. This is the principle behind the memetic version of MOGA proposed by Murata and Ishibuchi [33], MOGLS proposed by Jaszkiewicz [28], MGK proposed by Gandibleux et al. [19], GTS$^{\text{MOKP}}$ proposed by Barichard and Hao [2].

3. **Alternating schemes using EA and NSA as blackboxes**.
   Ben Abdelaziz et al. [1] have proposed a hybrid algorithm that uses both EA and NSA independently. The goal of the EA (a genetic algorithm) is to produce an initial diversified approximation, which is then improved by the NSA (a tabu search algorithm). This algorithm has been applied to the multi-objective knapsack problem.

4. **EA + NSA + problem-dependent components**.
   The most recent hybrid procedures integrate EA and NSA components, as well as problem-dependent components, in order to design a powerful approximation method for MOCO problems. Gandibleux et al. [20, 22] have proposed a population-based method in which a crossover uses a "genetic" map of the population with a path relinking operator that generates new solutions by exploring the trajectories connecting solutions. This procedure has been applied to approximating the non-dominated frontier of assignment and knapsack problems with two objectives.

5. **Approximation and exact procedures in a hybrid method**.
   Gandibleux and Fréville [17] have proposed a procedure for the bi-objective knapsack problem combining an exact procedure for reducing the search space with a tabu search process for identifying the potentially efficient solutions. This reduction principle is based on cuts that eliminate parts of the decision space where (probably) no exact efficient solution exist. The tabu search is triggered on the reduced space, dynamically updating the bounds in order to guarantee the tightest value at any time.

   This type of hybrid can also apply if, in an exact method for generating the non-dominated points, the exact method needs good quality bounds. For example in the *seek and cut method* for solving the assignment problem proposed by Przybylski et al. [36], the "seek" computes a local approximation of the non-dominated frontier (i.e., bounds are computed by a population-based algorithm coupled with path relinking) which is then used to "cut" the search space of an implicit enumeration scheme.

4.2 A hybrid algorithm for the biSPP based on NSA and EA blackboxes

The hybrid algorithm proposed and tested in this paper falls in the third category of the methods described in the section 4.1, in which NSA and EA interact independently. However, this algorithm is based on an original scheme in order to make the most of the main assets of our procedures (see Figure 2).

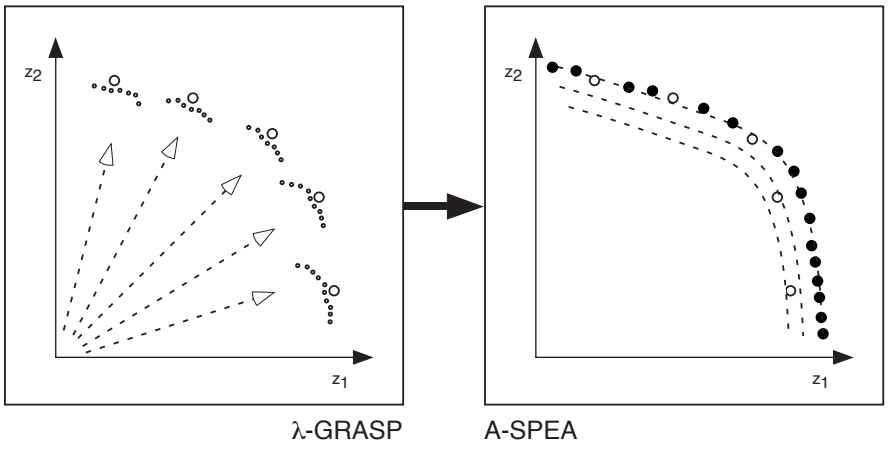


Figure 2: Illustration of the kind of approximation produced by λ-GRASP and A-SPEA

Our hybrid algorithm aims to compute an approximation that will insure good coverage, density and distribution of the approximate solutions along the efficient frontier. It works in two main phases with the available CPU time being split equally between the two:

1. During the phase 1, a known efficient heuristic for the single objective problem embedded in a parametric procedure (λ-GRASP) computes an initial set of very good solutions $\mathcal{P}_0$. This set contains all the potentially efficient produced by the algorithm. It is completed by the five best solutions found for each λ value and each α value (i.e. up to 500 solutions in theory, but less in practice, identical solutions being removed). The approximation obtained is expected to provide good coverage (i.e. an approximation that is well distributed along the efficient frontier).

2. During the phase 2, a multi-objective EA integrating a local search (A-SPEA) is performed on the population generated in the first phase to improve the set X_{PE} of potentially efficient solutions in terms of solution distribution and density.

Practically, the only difference between the A-SPEA described in algorithm 1 and the hybrid algorithm is located at line 2, where the function *PopInitialize* is replaced with the function $\lambda - GRASP$.

Thus, the λ-GRASP algorithm could be seen as a simple initialization phase within the hybrid. However, the CPU time allocated to the first phase and the high quality of the solutions produced

demonstrates its importance in our hybrid algorithm. In fact, even though the time is equally split between the two procedures, the A-SPEA algorithm acts as a post-optimization phase, which intensifies the search over the efficient frontier and densifies the population of potentially efficient solutions obtained. To our knowledge, this type of hybridization using an evolutionary algorithm to intensify the research has not yet been described in the literature.

5 Computational results

5.1 Numerical instances and experimental conditions

This section presents the computational results obtained for all the algorithms presented in this paper. A-SPEA was implemented in C langage (gcc 3.0.4), and λ-GRASP in Ada langage (Gnat 3.14). Both were run on an Pentium III with 800 MHz. We considered 120 numerical instances with the following characteristics:

- 100 or 200 variables,
- from 300 to 1000 constraints,
- a density of non-null elements ranging from 1% to 3% of the constraint matrix T,
- six classes of objective functions:

  A : two randomly generated objectives with
  $c_i^1 = rnd[1..100], c_i^2 = rnd[1..100], \forall i = 1, \ldots, n;$

  B : one randomly generated objective and one symmetrically sound objective
  $c_i^1 = rnd[1..100], \forall i = 1, \ldots, n;\ c_{n-i+1}^2 = c_i^1, \forall i = 1, \ldots, n;$

  C : two randomly generated objectives with patterns
  $l_1 = rnd[1..\frac{n}{10}], l_2 = rnd[1..\frac{n}{10}], \ldots;$
  $c_1^1 = c_2^1 = \ldots = c_{l_1}^1 = rnd[1..100];\ c_{l_1+1}^1 = c_{l_1+2}^1 = \ldots = c_{l_1+l_2}^1 = rnd[1..100]; \ldots$
  (idem for $c_i^2, \forall i = 1, \ldots, n$)

  D : one randomly generated objective with patterns and one symmetrically sound objective
  $l_1 = rnd[1..\frac{n}{10}], l_2 = rnd[1..\frac{n}{10}], \ldots;$
  $c_1^1 = c_2^1 = \ldots = c_{l_1}^1 = rnd[1..100];\ c_{l_1+1}^1 = c_{l_1+2}^1 = \ldots = c_{l_1+l_2}^1 = rnd[1..100]; \ldots$
  $c_{n-i+1}^2 = c_i^1, \forall i = 1, \ldots, n;$

  E : one unicost objective and one randomly generated objective
  $c_i^1 = 1, c_i^2 = rnd[1..100], \forall i = 1, \ldots, n;$

  F : one unicost objective and one randomly generated objective with patterns
  $c_i^1 = 1, \forall i = 1, \ldots, n$ and
  $l_1 = rnd[1..\frac{n}{10}], l_2 = rnd[1..\frac{n}{10}], \ldots;$
  $c_1^2 = c_2^2 = \ldots = c_{l_1}^2 = rnd[1..100];\ c_{l_1+1}^2 = c_{l_1+2}^2 = \ldots = c_{l_1+l_2}^2 = rnd[1..100]; \ldots$

Thus, a wide range of difficult instances were considered. All these instances are available in the MCDMLib, a collection of MOCO problems instances [31].

We obtained the minimal complete set of efficient solution for these instances with a dichotomic procedure (see Degoutin and Gandibleux [5]) using Cplex [27] to solve the following mono-objective parametrical problem:
$\max \left\{ \sum_{q=1}^{2} \lambda_q z^q(x) : x \in X \right\}$ with $0 \leq \lambda_q \leq 1 \forall q \in Q$ and $\sum_{q=1}^{2} \lambda_q = 1$ and two additionnal constraints, $c^1 x > z^1(x^{(B)})$ and $c^2 x > z^2(x^{(A)})$, where $x^{(A)}$ and $x^{(B)}$ are respectively two optimal solutions for z^1 and z^2 of the scalarized problem.

However, even for small sized instances, the CPU time needed to solve this problem exactly can be exorbitant (up to 360 000 seconds on the computer used for this paper).

For each instance, our algorithms performed 16 independant runs with the parameters indicated in section 2, 3 and 4, and a time limit as stopping condition. This time limit was based on a reference time computed according to the following formula: $T_{Ref} = 10^{-3} * n^2$ seconds. Five different time

limits were considered: T_{Ref}, $2 * T_{Ref}$, $3 * T_{Ref}$, $4 * T_{Ref}$ and $5 * T_{Ref}$. This allow us to monitor the evolution of the algorithms performance in consideration with the time consumed. This led us to set time limits from 10s to 50s for instances with 100 variables and from 40s to 200s for instances with 200 variables. All the results presented are average results among the 16 runs, except when it is clearly indicated otherwise.

In order to measure the quality of the approximation produced, and since it is very difficult to compare two potentially efficient solutions sets, three indicators were used to characterize each approximation:

1. the percentage of exact efficient vectors found (M1 proposed by Ulungu et al. [42]),
2. the euclidean distance to the efficient frontier, and
3. the hypervolume defined by the solution set (S-metric proposed by Zitzler and Thiele [46]).

To facilitate the comparisons for the whole set of instances (or for some subset of the instances), the hypervolume values are indicated as percentages of the efficient frontier hypervolume value.

5.2 A-SPEA versus λ-GRASP

The results obtained with our A-SPEA and λ-GRASP algorithms are presented in figures 3, 4 and 5. These results are the average results for all the instances considered. Overall, they seem fairly good, since more than 60% of the efficient solution were found in a very short time, and up to 80% were found with more time. The value of the two other indicators (described in section 5.1) were also good. With enough time, the distance to the efficient frontier had a value less than 5, which is very low with regard to the optimal value of the objective function considered (often more than 2,000 since the value of a single item can be up to 100), and more than 98% of the efficient frontier value for the S-metric indicator. Two main observations can be made:

1. Though A-SPEA seems to be better than λ-GRASP in terms of the distance indicator, the opposite is true for the hypervolume indicator. The two algorithms had nearly equivalent perfomances in terms of the percentage of efficient solutions found. This demonstrates that an analysis should not be based on a single indicator. In fact, A-SPEA produces an approximation that is located near the efficient frontier and that has a good solution density, but appears to have difficulty finding extremal solutions, despite the extensions of the basic SPEA (see section 2). Without these extensions the results obtained for extremal solutions were worse. On the other hand, λ-GRASP was able to find the extremal solutions, but the solution density was lower. Using a higher number of search directions would correct this problem, but this would result in a higher distance from the efficient frontier and less efficient solutions. Figure 6 represents the approximations obtained with the two algorithms on one run of one instance, and illustrates the differences between the approximations quite well.
2. the approximations produced by λ-GRASP improve more quickly than those of A-SPEA when given more time. The learning/diversification process included in λ-GRASP permits more new solutions to be generated, and thus more new good solutions. On the other hand, the A-SPEA algorithm seems to have trouble diversifying its population after a certain time.

In addition, there is an important performance gap for λ-GRASP between the instances of famillies A to D (with two weighted objectives) and those in famillies E and F (with a weighted and an unicost objective). Figure 7 shows the average distance to the efficient frontier obtained with λ-GRASP for these two famillies (the first one is called "weighted" and the second, "unicost"). The gap is important and does not decrease much when more computational time is added. The same is true for the other two indicators. However A-SPEA does not appear to have a performance gap. In fact, it seems that A-SPEA produces slighly better performances on the unicost instances.

We do not have a complete explanation for this phenomenon, but it is most likely due to the fact that we do not normalize the two objective functions when the scalarization function is computed. For the unicost instances, the important range difference between the two objective functions may also be the source of the problem.

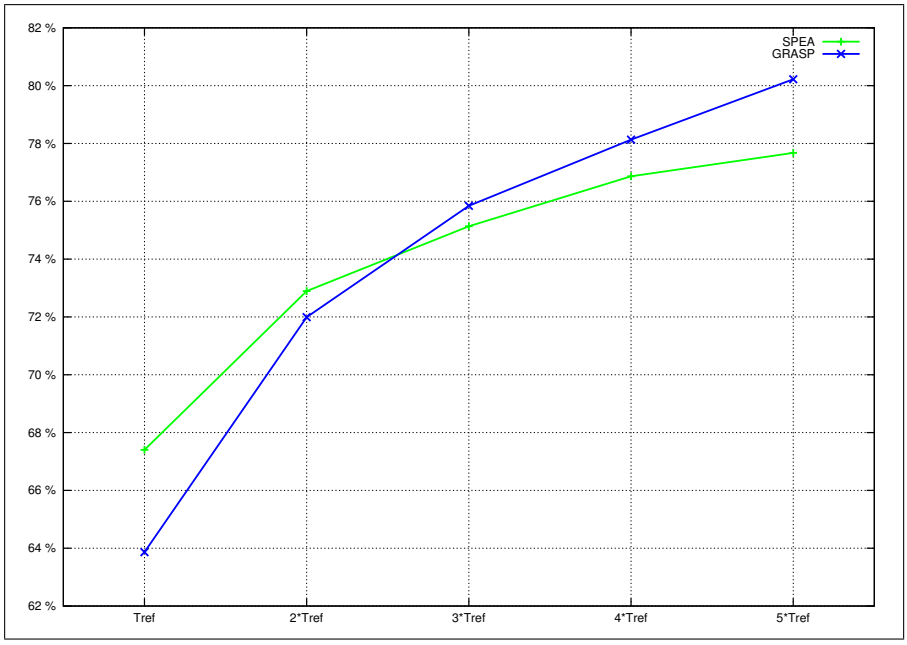


Figure 3: Average percentage of exact efficient solutions found

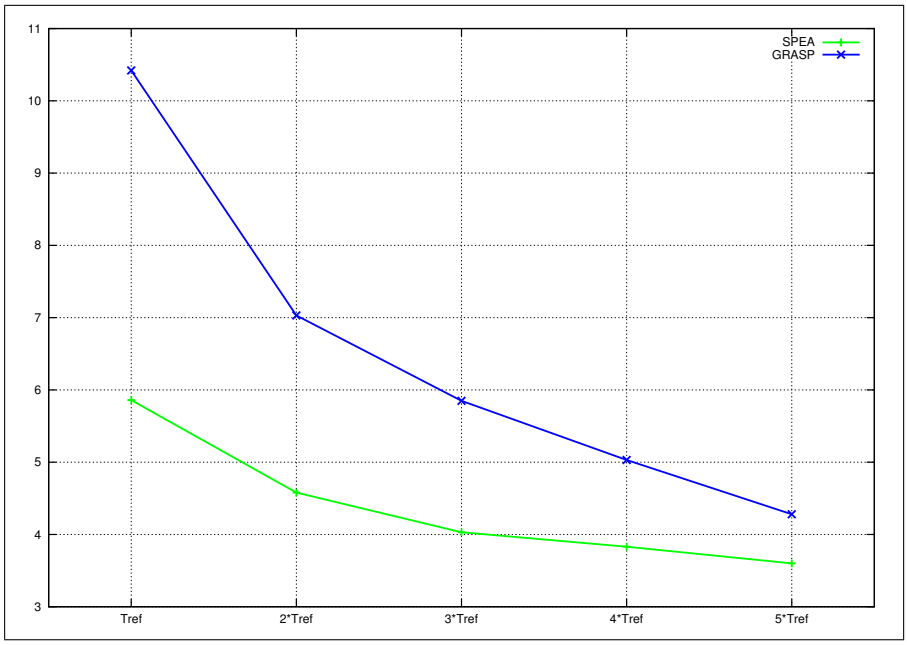


Figure 4: Average distance to the efficient frontier

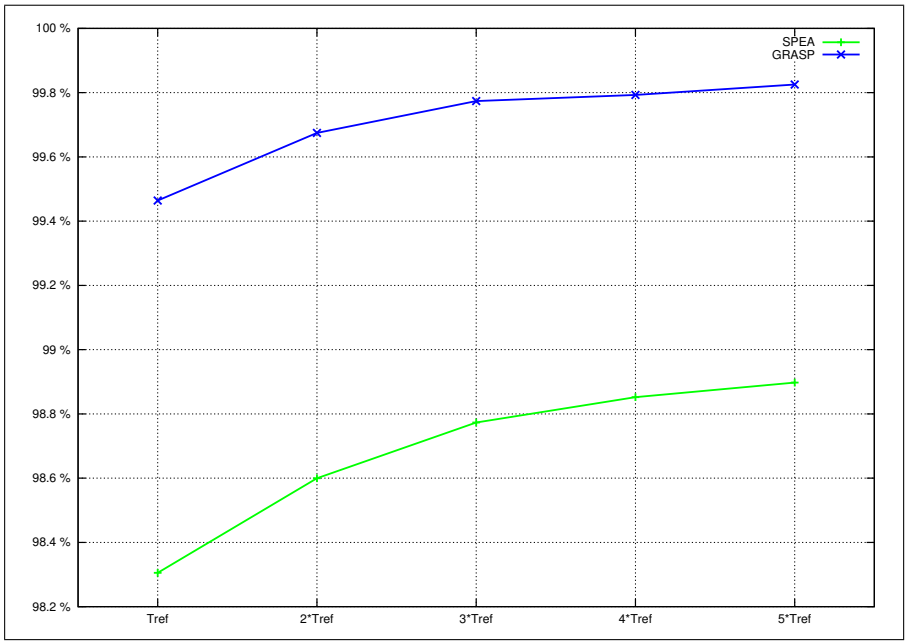


Figure 5: Average percentage of the efficient frontier S-metric value

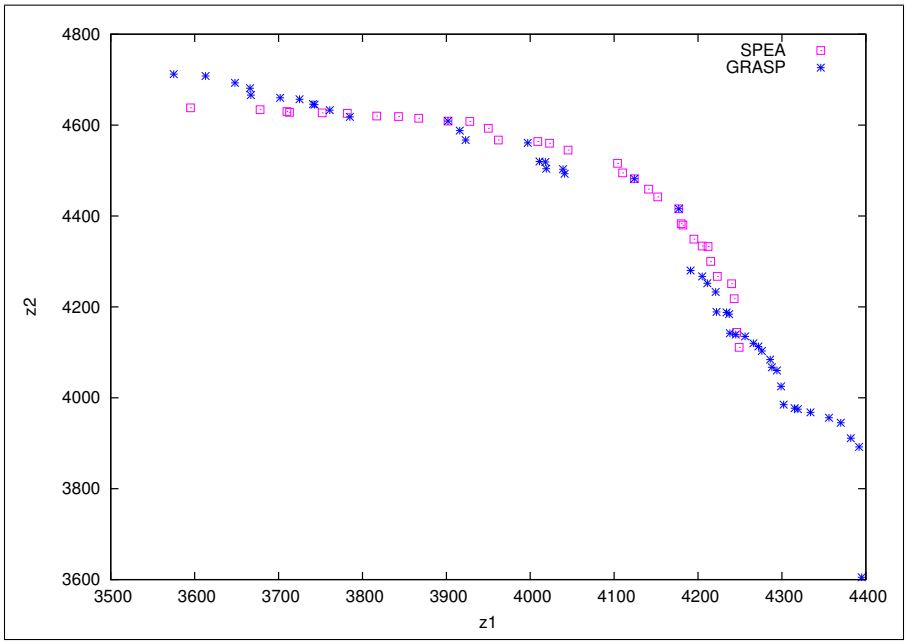


Figure 6: Example of an approximation produced using A-SPEA and λ-GRASP

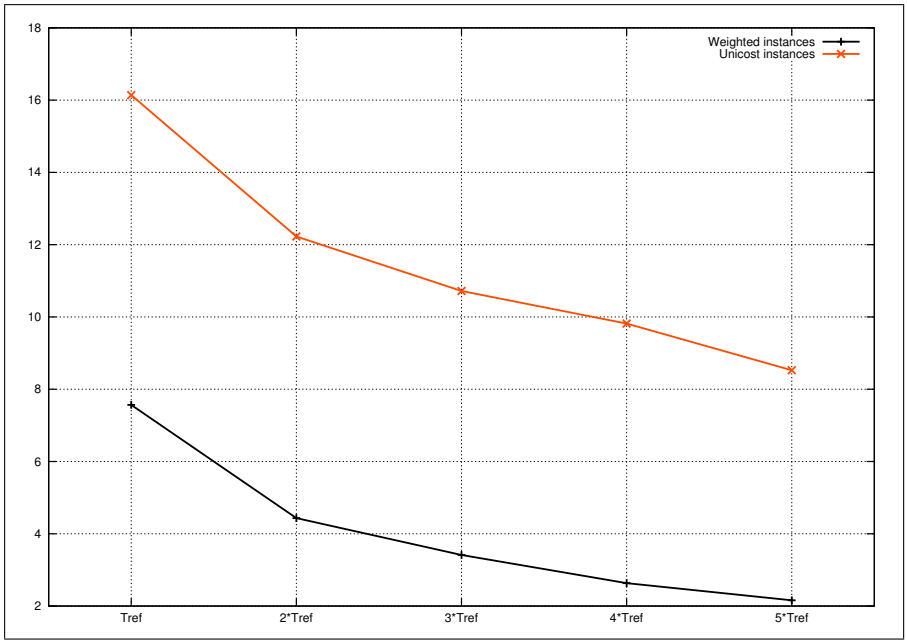


Figure 7: Average distance to the efficient frontier for the GRASP algorithm

The percentage of efficient solutions found by λ-GRASP in the maximal time in function of the ratio of supported efficient solutions ($\frac{SE}{E}$) for the instances performed (see table 1) show no significant performance difference between the instances with a large proportion of non-supported efficient solutions and the others. This confirms that using a scalarization function with an heuristic method allows both non-supported and supported efficient solutions to be found.

$\frac{SE}{E}$	$0-24\%$	$25-49\%$	$50-74\%$	$75-99\%$	100%
M1	62.8%	89.9%	83.4%	75.6%	87.2%

Table 1: Percentage of efficient solutions found by λ-GRASP

5.3 Impact of the hybridization

The average results obtained with our hybrid algorithm, and the A-SPEA and λ-GRASP algorithms, are reported in figures 8, 9 and 10. Overall, the hybrid appears to perform much better than the other two, whatever the indicator or the computational time considered:

- more than 88% of the efficient solution are found in a very short time and up to more than 95% are found when more time is given, which represents a positive gap of more than 15 points compare to the other two algorithms,
- the distance to the efficient frontier is around 2 with the lower time and down to less than 1 with more time, which is more than three times better than A-SPEA and five times better than λ-GRASP,
- more than 99,6% of the efficient frontier value for the S-metric indicator, which represents a slight gap with λ-GRASP and a larger one with A-SPEA (around 1 point).

In fact, the hybrid algorithms performs better than the two others, even when they perform very well (i.e. the distance indicator for A-SPEA and S-metric for λ-GRASP). It seems to be able to effectively compensate for the weaknesses of the algorithms that make it up.

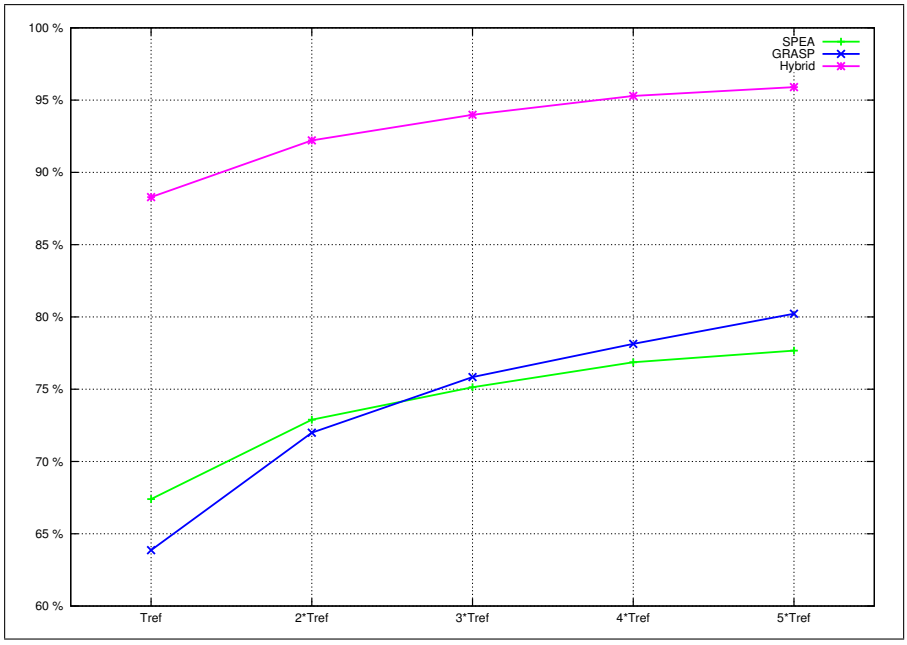


Figure 8: Average percentage of efficient solutions found

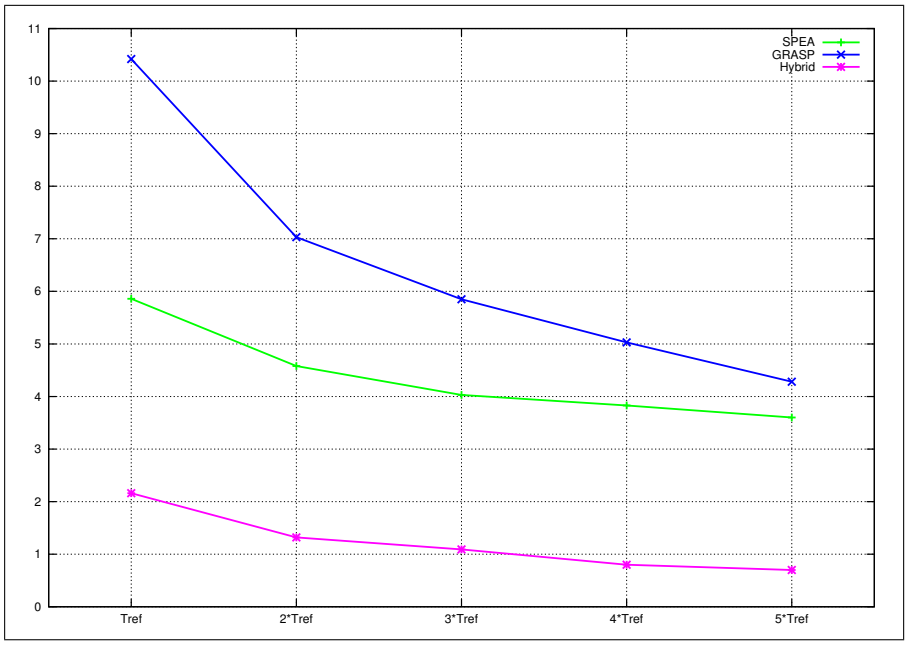


Figure 9: Average distance to the efficient frontier

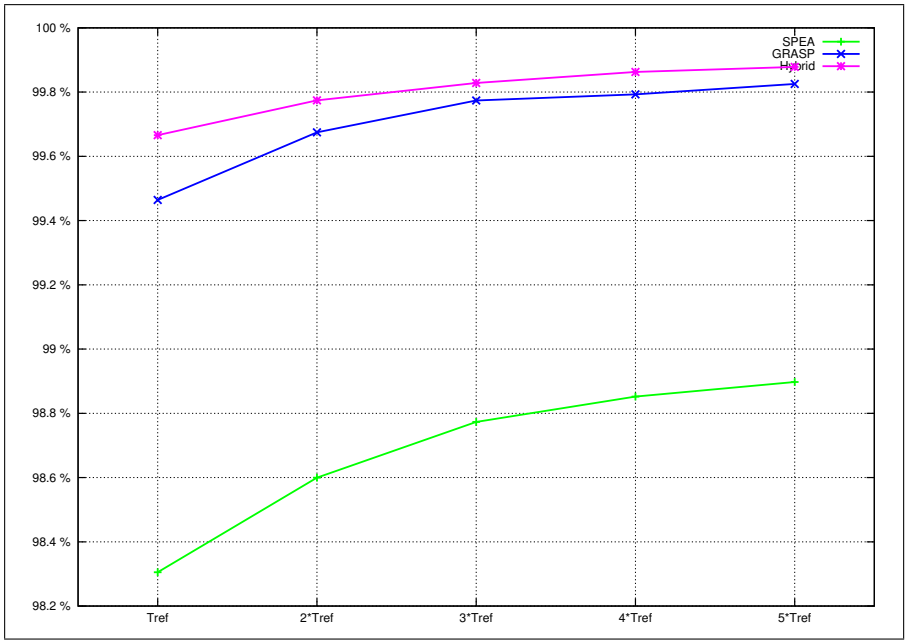


Figure 10: Average percentage of the efficient frontier S-metric value

Furthermore, the poor performance of λ-GRASP on the unicost instances is nearly inexistant with the hybrid algorithm. Figure 11 shows the average distance to the efficient frontier obtained with the hybrid algorithm in the two famillies of instances (see figure 7 for the results with λ-GRASP). Depending of the computational time available, the hybrid is slightly better on the weighted instances, but the difference does not seem significant and is very low in comparison to those observed with λ-GRASP.

Looking at the worst case results instead of average results (i.e. the results obtained on the worst run for the worst instance), the hybrid algorithm still performs better than the other two. Figure 12 shows the percentage of efficient solutions found for this worst case for the three algorithms (NB: this worst case can correspond to different instances for each algorithm). Though this percentage is nearly null for A-SPEA and λ-GRASP, it can be as high as 40% for the hybrid algorithm with enough computational time, which can be considered as a good value. Also, this value increases strongly as the computational time increases. The same observation can be made for the two other indicators:

- The distance to the efficient frontier can be less than 20 with enough computational time. A significant improvement appears when the time available is increased.
- The value for the S-metric indicator is always more than 94 % of the efficient frontier whatever the computational time considered, with a slight increase given more time.

In all the cases (i.e. for each indicator and each time considered), the hybrid algorithm produces better results than the two other algorithms.

6 Conclusion and perspectives

In this paper, we presented two heuristic methods for approximating the efficient frontier of the bi-objective set packing problem, as well as an original hybrid method. These methods are all based on the SPEA and GRASP metaheuristics. The results obtained with the two heuristics seem fairly good despite the specific weakenesses inherent to each one. However, the most important thing is that they are complementary and produce very good results when joined together in a hybrid. This hybrid was quite efficient, thanks to the learning process that is part of λ-GRASP, which produces a population that is diversified enough to succesfully initialize SPEA.

Obviously, the two heuristics algorithms could be improved independently. This is especially true of the λ-GRASP, which could be made more efficient by using a path relinking phase with solutions

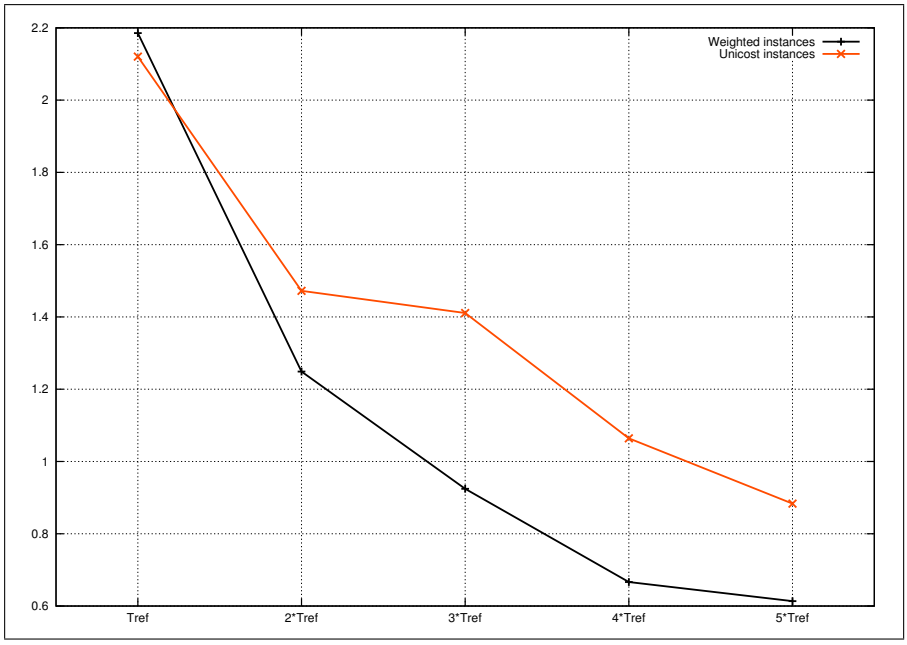


Figure 11: Average distance to the efficient frontier for the hybrid algorithm

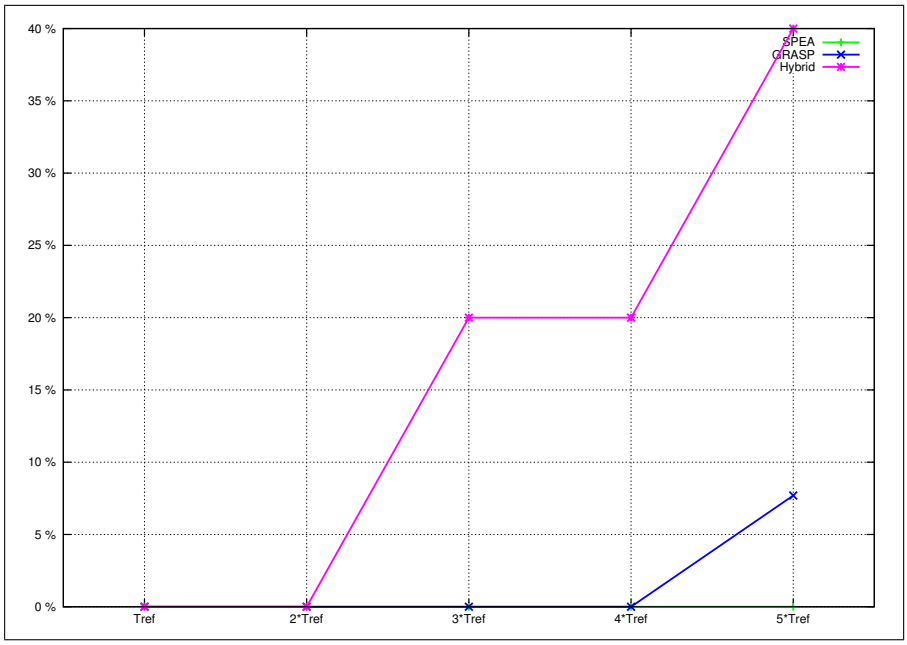


Figure 12: Percentage of efficient solutions found in the worst case

generated in two different search directions (see Gandibleux et al. [21]). It would also be interesting to see the performances obtained with the three algorithms on larger instances, even though the performance would be more difficult to evaluate due to the impossibility of solving the problem exactly, making the efficient frontier hard to obtain. Extending these algorithms to the multi-objective case would also be useful, but would produce the same evaluation difficulties since there is no exact method for this case.

However, the most promising possibilities involve the hybrid method. For example, the time repartition between λ-GRASP and A-SPEA, which is arbitrary set equal in our algorithm, could become a variable parameter in the method. Actually, it seems that, since λ-GRASP makes better use of additional time, giving more time in this phase could improve the performance. Moreover, it would be interesting to see if this new scheme of hybridization can succesfully be extended to other MOCO problems.

This hybrid algorithm acts as a scheme of collaboration between solvers, and combines scalarizing functions and Pareto dominance techniques to move a population towards the efficient frontier. A closer look at this combination could provide interesting information for future use in the design of an even more effective derived algorithm, blending these techniques.

References

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

[2] Vincent Barichard and Jin-Kao Hao. Un algorithme hybride pour le problème de sac à dos multiobjectifs. In *Huitièmes Journées Nationales sur la Résolution Pratique de Problèmes NP-Complets JNPC'2002 Proceedings*, 2002. Nice, France, 27–29 May.

[3] Piotr Czyzak and Andrzej Jaszkiewicz. Pareto simulated annealing. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making. Proceedings of the XIIth International Conference, Hagen (Germany)*, volume 448 of *Lecture Notes in Economics and Mathematical Systems*, pages 297–307, 1997.

[4] Fabien Degoutin. *Problèmes bi-objectifs en optimisation combinatoire et capacité d'infrastructures ferroviaires*. Master thesis, Université de Valenciennes et du Hainaut Cambrésis, Valenciennes, France, 2002. In french.

[5] Fabien Degoutin and Xavier Gandibleux. Un retour d'expériences sur la résolution de problèmes combinatoires bi-objectifs. 5e journée du groupe de travail Programmation Mathématique MultiObjectif (PM20), Angers, France, mai 2002. In french.

[6] Xavier Delorme. *Modélisation et résolution de problèmes liés à l'exploitation d'infrastructures ferroviaires*. PhD thesis, Université de Valenciennes et du Hainaut Cambrésis, Valenciennes, France, 2003. In french.

[7] Xavier Delorme, Xavier Gandibleux, and Fabien Degoutin. Résolution approchée du problème de set packing bi-objectifs. In *Proceedings de l'École d'Automne de Recherche Opérationnelle de Tours (EARO)*, pages 74–80, 2003. In french.

[8] Xavier Delorme, Xavier Gandibleux, and Joaquín Rodriguez. GRASP for set packing problems. *European Journal Of Operational Research*, 153(3):564–580, 2004.

[9] Matthias Ehrgott and Xavier Gandibleux. Multiobjective combinatorial optimization. In M. Ehrgott and X. Gandibleux, editors, *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Survey, Kluwer's International Series in Operations Research and Management Science*, volume 52, pages 369–444. Kluwer Academic Publishers, Boston, 2002.

[10] Matthias Ehrgott and Xavier Gandibleux. Approximative solution methods for multiobjective combinatorial optimization. *TOP (Spanish journal of operations research)*, 12(1):190, 2004.

[11] Matthias Ehrgott and Xavier Gandibleux. Bound sets for biobjective combinatorial optimization problems. *Computers and Operations Research*, page (to appear), 2005.

[12] Thomas A. Féo and Mauricio G.C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8:67–71, 1989.

[13] Paola Festa and Mauricio G.C. Resende. GRASP : an annotated bibliography. In Celso C. Ribeiro and P. Hansen, editors, *Essays and surveys on metaheuristics*, pages 325–367. Kluwer academic publishers, 2002.

[14] C.M. Fonseca and P.J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, California, 1993. University of Illinois at Urbana-Champaign*, pages 416–423. Morgan Kaufman, San Francisco, CA, 1993.

[15] Xavier Gandibleux, Fabien Degoutin, and Xavier Delorme. A first feedback on set packing problems with two objectives. Workshop on Multiple Objective Metaheuristics (MOMH), Carré des Sciences, Paris, France, November 4-5 2002.

[16] Xavier Gandibleux and Matthias Ehrgott. 1984-2004 — 20 years of multiobjective metaheuristics. but what about the solution of combinatorial problems with multiple objectives? In C. A. Coello Coello, A. Hernandez Aguirre, and E. Zitzler, editors, *Evolutionary Multi-Criterion Optimization*, volume 3410 of *Lecture Notes in Computer Science*, pages 33–46. Springer-Verlag, 2005.

[17] Xavier Gandibleux and Arnaud Fréville. Tabu search based procedure for solving the 0/1 multiobjective knapsack problem: The two objective case. *Journal of Heuristics*, 6(3):361–383, 2000.

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

[19] Xavier Gandibleux, Hiroyuki Morita, and Naoki Katoh. A genetic algorithm for 0-1 multiobjective knapsack problem. In *International Conference on Nonlinear Analysis and Convex Analysis (NACA98) Proceedings, July 28-31 1998, Niigata, Japan*, 1998.

[20] Xavier Gandibleux, Hiroyuki Morita, and Naoki Katoh. The Supported Solutions Used as a Genetic Information in a Population Heuristic. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *Evolutionary Multi-Criterion Optimization*, volume 1993 of *Lecture Notes in Computer Science*, pages 429–442. Springer-Verlag, 2001.

[21] Xavier Gandibleux, Hiroyuki Morita, and Naoki Katoh. Evolutionary operators based on elite solutions for biobjective combinatorial optimization. In C. Coello Coello and G. Lamont, editors, *Applications of Multi-Objective Evolutionary Algorithms*, pages 555–579. World Scientific, Singapore, 2004.

[22] Xavier Gandibleux, Hiroyuki Morita, and Naoki Katoh. A population-based metaheuristic for solving assignment problems with two objectives. *Journal of Mathematical Modelling and Algorithms (to appear)*, (to appear), 2004.

[23] Michael R. Garey and David S. Johnson. *Computers and intractability : a guide to the theory of NP-Completeness*. V.H. Freeman and Company, 1979.

[24] Fred Glover and Manuel Laguna. *Tabu Search*. Kluwer academic publishers, 1997.

[25] M.P. Hansen. *Metaheuristics for multiple objective combinatorial optimization*. PhD thesis, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby (Denmark), 1998. Report IMM-PHD-1998-45.

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

[27] ILOG. CPLEX 8.0 (user's manual), 2002.

[28] Andrzej Jaszkiewicz. Multiple objective genetic local search algorithm. In M. Köksalan and S. Zionts, editors, *Multiple Criteria Decision Making in the New Millennium*, volume 507 of *Lecture Notes in Economics and Mathematical Systems*, pages 231–240. Springer-Verlag, 2001.

[29] Andrzej Jaszkiewicz. Evaluation of multiple objective metaheuristics. In X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, pages 65–89. Springer-Verlag, 2004.

[30] B. Malakooti, J. Wang, and E.C. Tandler. A sensor-based accelerated approach for multi-attribute machinability and tool life evaluation. *International Journal of Production Research*, 28:2373, 1990.

[31] MCDMlib. Mcdm numerical instances library. `http://www.terry.uga.edu/mcdm/`.

[32] P. Moscato. Memetic algorithms. In Panos Pardalos and Mauricio Resende, editors, *Handbook of Applied Optimization*. Oxford University Press, 2000.

[33] T. Murata and H. Ishibuchi. MOGA: Multi-objective genetic algorithms. In *Proceedings of the 2nd IEEE International Conference on Evolutionary Computing*, pages 289–294, 1995.

[34] George L. Nemhauser and Laurence A. Wolsey. *Integer and combinatorial optimization*. Willey-Interscience, 1999.

[35] Leonidas S. Pitsoulis and Mauricio G.C. Resende. Greedy randomized adaptive search procedures. In P.M. Pardalos and M.G.C. Resende, editors, *Handbook of Applied Optimization*, pages 168–183. Oxford University Press, 2002.

[36] Anthony Przybylski, Xavier Gandibleux, and Matthias Ehrgott. Seek and cut algorithm computing minimal and maximal complete efficient solution sets for the biobjective assignment problem. In *MOPGP 04: 6th Int. Multi-Objective Programming and Goal Programming conf New Trends and Applications. April 14 - 16, 2004, Hammamet, Tunisia*, 2004.

[37] Mauricio G.C. Resende and Celso C. Ribeiro. Greedy randomized adaptive search procedures. In F. Glover and G. Kochenberger, editors, *Handbook in Metaheuristics*, pages 219–249. Kluwer academic publishers, Boston, 2002.

[38] J.D. Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, Nashville, TN (USA), 1984.

[39] P. Serafini. Simulated annealing for multiobjective optimization problems. In *Proceedings of the 10th International Conference on Multiple Criteria Decision Making, Taipei-Taiwan*, volume I, pages 87–96, 1992.

[40] M. Sun. Applying tabu search to multiple objective combinatorial optimization problems. In *Proceedings of the 1997 DSI Annual Meeting, San Diego, California*, volume 2, pages 945–947. Decision Sciences Institute, Atlanta, GA, 1997.

[41] Ekunda L. Ulungu. *Optimisation Combinatoire multicritère: Détermination de l'ensemble des solutions efficaces et méthodes interactives*. PhD thesis, Faculté des Sciences, Université de Mons-Hainaut, Mons, Belgique, 1993. In french.

[42] Ekunda L. Ulungu, Jacques Teghem, and Philippe Fortemps. Heuristics for multi-objective combinatorial optimisation problem by simulated annealing. In J. Gu, G. Chen, Q. Wei, and S. Wang, editors, *MCDM: Theory and Applications*, pages 228–238. SCI-TECH Information Services, Windsor, UK, 1995.

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

[44] Eckart Zitzler. *Evolutionnary Algorithms for Multiobjective Optimization : Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Suisse, 1999.

[45] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2 : Improving the Strenght Pareto Evolutionary Algorithm for Multiobjective Optimization. In *Evolutionary Methods for Design, Optimisation, and Control*, pages 95–100. CIMNE, Barcelona, Spain, November 2002.

[46] Eckart Zitzler and Lothar Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.

Les rapports de recherche
du Centre G2I de l'ENSM-SE
sont disponibles en format PDF
sur le site Web de l'Ecole

G2I research reports
are available in PDF format
on the site Web of ENSM-SE

www.emse.fr

Centre G2I
Génie Industriel et Informatique

Division for
Industrial Engineering and Computer Sciences
(G2I)

Par courier :

By mail:

Ecole Nationale Supérieure des Mines de Saint-Etienne
Centre G2I
158, Cours Fauriel
42023 SAINT-ETIENNE CEDEX 2
France

Ecole Nationale Supérieure des Mines de Saint-Etienne
Centre G2I
158, Cours Fauriel
42023 SAINT-ETIENNE CEDEX 2

www.emse.fr