

Integrating partial optimization with scatter search for solving bi-criteria $\{0,1\}$ -knapsack problems

Carlos Gomes da Silva^(2,3,*), José Figueira^(1,3,†) and João Clímaco^(1,3)

(1) Faculdade de Economia da Universidade de Coimbra
Av. Dias da Silva, 165, 3004-512 Coimbra, Portugal
Phone: +351 239 760 500, Fax: +351 239 403 511
E-mail: figueira@fe.uc.pt

(2) Escola Superior de Tecnologia e Gestão de Leiria
Morro do Lena, Alto Vieiro, 2401-951 Leiria, Portugal
Phone: +351 244 820 300, Fax: +351 244 820 301
E-mail: cgsilva@estg.iplei.pt

(3) INESC-Coimbra
Rua Antero de Quental, 199
3000-033 Coimbra, Portugal
Phone: +351 239 851 040, Fax: +351 239 824 692
E-mail: jclimaco@inescc.pt

March 28, 2004

Abstract

This paper deals with the problem of inaccuracy of the solutions generated by metaheuristic approaches for bi-criteria $\{0,1\}$ -knapsack problems. A hybrid approach which combines systematic and heuristic search is proposed to reduce that inaccuracy in the context of a scatter search method. The components of the method are used to determine regions in the decision space to be systematically searched.

Comparisons with small and medium size instances exactly solved are presented. Large size instances are also considered and the quality of the approximation is made by taking into account the proximity to the upper frontier, devised by the linear relaxation, and the diversity of the solutions.

The results show the effectiveness of this approach both in small, medium and large size instances.

Key-words: Metaheuristics, exact methods, scatter search, bi-criteria knapsack problems, combinatorial optimization, intensification strategies

*Corresponding author

†Visiting Researcher (March, 2004) at Department of Engineering Science, The University of Auckland, Auckland, NZ

1 Introduction

The multiple criteria $\{0,1\}$ - knapsack problem is about selecting a set of items without exceeding a given capacity of the knapsack and taking into account several conflicting criteria. Generating the exact set of *efficient/non-dominated* solutions is a hard task, even for the easiest case, *i.e.*, for bi-criteria problems. Exact methods can only deal with small and medium size instances (Visée *et al.*, 1998 and Captivo *et al.*, 2003). Large size instances require the use of approximate methods, which lead to approximate *efficient/non-dominated* solutions, called *potentially efficient/non-dominated* solutions.

Several approximate methods, in particular metaheuristics, have recently been proposed, or applied, to approximate the entire set of non-dominated solutions of the multiple criteria $\{0,1\}$ -knapsack problem (Ben Abdelaziz and Krichen, 1997; Czyzak and Jaskiewicz, 1998; Gandibleux and Fréville, 2000; Gandibleux *et al.*, 2001; Gomes da Silva *et al.*, 2003b). They are genetic algorithms, tabu-search, simulated annealing and scatter search metaheuristics. However, when compared with the exact set of efficient/non-dominated solutions the approximation revealed to be poor, particularly in large size instances. The main difficulties faced when using an approximate method are the following:

- 1) the approximate set is not well dispersed along the non-dominated region;
- 2) only a few number of exact efficient/non-dominated solutions is obtained;
- 3) the number of potentially efficient/non-dominated solutions is quite lower than the number of exact efficient/non-dominated solutions;
- 4) a lot of computational time is required to reduce the difficulties 1), 2) and 3).

All the above referred methods suffer, with different degrees of intensity, from the above difficulties. The first difficulty is due to the lack of diversification of the solutions along the iterative process of solving the problem, while the remaining are mainly due to the presence of weak combination diversification/intensification strategies for searching the decision space.

Intensification strategies are responsible for guiding the search towards promising regions and exploiting their solutions space (Glover, 1999; Laguna and Martí, 2003). In 0-1 problems, their application usually consists of evaluating a neighborhood of a given solution by commuting the value of a set of variables. Identifying such a set of variables is particularly time consuming. Hence its general use would lead to an inefficient method. For this reason, it is usual to consider only a small neighborhood. The price to pay for this "low visibility of the search space" (Mautor, 2001) is the poor performance of the intensification procedure. Overcoming difficulties 2) and 3) requires more sophisticated and stronger intensification procedures guided by efficient strategies.

This paper considers the development of such procedures for bi-criteria $\{0,1\}$ -knapsack problems and in the context of the Scatter Search (SS) method by Gomes da Silva *et al.* (2003b). The method by Gomes da Silva *et al.* (2003b) is applied to large size instances and it is structured according to the usual components of SS metaheuristic: diversification method, improvement method, subset generation method and combination method. The computational experiments show that the quality of the approximation concerning the distribution along the non-dominated frontier and the proximity to the upper frontier (derived from the linear relaxation of the problem) is quite good and the CPU time for obtaining such an approximation is reasonable (about one hour of CPU time for instances with a number of items up to 6,000). Nevertheless, it was detected that the number of the solutions obtained increases linearly with the number of items, which contradicts the exponential increase evidenced by exact methods for small and medium

size instances. It was also verified that the percentage of exact non-dominated solutions was not very high and decreases substantially with the size of the problem. The SS method does not suffer from difficulty 1) but difficulties 2) and 3) appear to be present.

Here, the method is modified in order to overcome those drawbacks. The major modifications are made in the combination method by incorporating a mixture of exact and heuristic search strategies. Heuristic search strategies are used to detect interesting regions in the decision space which are then exploited by a systematic search. Indeed, when such regions are found they are completely exploited due to the exact nature of the underlying method. The approach consists of a partial optimization scope hence the original bi-criteria problem is divided into smaller ones, which are then optimized, *i.e.*, all the efficient solutions of it are computed.

The combination of these two types of search has been successfully applied to several single criterion combinatorial optimization problems such as: time tabling, quadratic assignment, vehicle routing, scheduling and network design (Mautor, 2001). As far as we know, the proposed combination method was not already applied to multiple criteria problems with the purpose of exactly exploiting regions detected by heuristic strategies.

The approach here presented requires the existence of an exact method that can work efficiently, at least, for small size instances. In our problem, the systematic search is performed by the exact method of Visée *et al.* (1998). This method is applied to several subproblems of convenient size, *i.e.*, a number of variables that a subproblem should not exceed in order to be quickly solved. Several properties for the combination can be derived helping to accelerate the method and giving insights on the quality of the obtained solutions. The requirements and the characteristics of the combination of solutions impose modifications in the structure of the *reference set update method* and in the *subset generation method*. We begin with the presentation of the changes in the combination method in order to justify the changes in the design of the reference set update method and subset generation method.

The paper is organized as follows. Section 2 presents a brief overview of the SS based method by Gomes da Silva *et al.* (2003b). Section 3 is about the solution combination method, reference set update method and subset generation method. Section 4 is devoted to the computational results. Section 5 presents the main conclusions of this work.

2 A brief overview of the Scatter Search method

This section contains a brief overview of the SS based method proposed in Gomes da Silva *et al.* (2003b) for the bi-criteria $\{0,1\}$ -knapsack problem, which can be formulated as follows:

$$\begin{aligned}
 \max z_1(x_1, \dots, x_j, \dots, x_n) &= \sum_{j=1}^n c_j^1 x_j \\
 \max z_2(x_1, \dots, x_j, \dots, x_n) &= \sum_{j=1}^n c_j^2 x_j \\
 \text{s.t. :} & \\
 \sum_{j=1}^n w_j x_j &\leq W \\
 x_j &\in \{0, 1\}, j = 1, \dots, n
 \end{aligned} \tag{1}$$

where c_j^i represents the value of item j on criterion i , $i = 1, 2$, $x_j = 1$ if item j ($j = 1, \dots, n$) is included in the knapsack and $x_j = 0$ otherwise, w_j is the weight of item j and W is the overall

knapsack capacity. We assume that c_j^1, c_j^2, W and w_j are positive integers and that $w_j \leq W$ with $\sum_{j=1}^n w_j > W$. Constraints $\sum_{j=1}^n w_j x_j \leq W$ and $x_j \in \{0, 1\}, j = 1, \dots, n$, define the feasible region in the decision space, and their image when using the criteria functions z_1 and z_2 define the feasible region in the criteria space.

A feasible solution, x , is said to be *efficient* if and only if there is no a feasible solution, y , such that $z_i(x) \leq z_i(y), i = 1, 2$ and $z_i(x) < z_i(y)$ for at least one i . The image of an efficient solution in the criteria space is called a *non-dominated* solution.

The method is structured according to the basic definition of SS metaheuristic: diversification method, improvement method, reference set update method, subset generation method, and combination method.

The method starts with the *Diversification method* which consists of a non-stochastic procedure based on the continuous bi-criteria $\{0,1\}$ -knapsack. In this procedure all the extreme solutions are found through efficient pivotings using the bi-criteria simplex method with bounded variables (Gomes da Silva *et al.*, 2003a and 2003b). Each of the extreme solutions has only one current basic variable, which is frequently a fractional one. The initial set of integer solutions contains thus the solutions obtained from the extreme solutions, including and excluding the item associated with the basic variable. In the first case, the solution obtained is clearly feasible, but the remaining capacity might be large enough to include additional items. Two new solutions are created using two heuristic procedures. The first (second) one fills the knapsack, as much as possible, with the items which have the highest profit-to-weight ratio, according to criterion z_1 (z_2). In the second case, the inclusion of the fractional variable is only possible if at least one of the included variables is removed. Two new solutions are created using two heuristic procedures. The first (second) corresponds to the deletion of the variable which makes the solution feasible and has the lowest profit-to-weight ratio, according to criterion z_1 (z_2). This is done by the *Improvement method*. Then, the *Reference set update method* builds, in each iteration, a set of solutions selected from the available potentially efficient solutions. Those are first ordered according to non-increasing values of criterion z_2 and divided into groups with approximately the same number of solutions. From each group the mean solution is selected and then added to the reference set. The *Subset generation method* uses the strategy of considering consecutive solutions from the reference set. The *Combination method* explores paths between solutions of each subset. The variables whose values should change in order to achieve one solution from the other are firstly identified. The value of these variables is changed one by one and the improvement method is then applied. The set of the potentially efficient solutions is updated and the process is repeated with the reference set update method. The procedure stops when a maximum number of iterations is achieved.

3 Modifications in the scatter search method

The changes of the SS method proposed by Gomes da Silva *et al.* (2003b) are presented in the next sections. As referred above, we start by presenting the changes in the combination methods, which induces the changes in the reference set method and subset generation method.

3.1 Combination method

The combination method uses the solutions to be combined to define a subproblem, called *residual problem*, desirably a small one, which *a priori* can be solved exactly and rapidly. This will perform an exact search in a subspace of the decision space.

The point consists of finding such a residual problem, *i.e.*, finding a procedure that largely and adequately reduces the original problem.

On the one hand, if the residual problem has few variables then it is possible that some of them are fixed in the contrary values, *i.e.*, lead to dominated solutions. In this case, the quality of the solution of residual problems is penalized. On the other hand, if the residual problem has a large number of variables then the problem could be too difficult to solve it exactly. A balance between these two features must be achieved.

3.1.1 Defining residual problems

To identify the residual problems to solve, the definition of *consistent variable* (Glover, 1977) is used. In our context, consistent variables are the ones that have the same value in all the solutions to be combined.

Let $S = \{x^1, \dots, x^k, \dots, x^q\}$ be a set of solutions to combine, and $\Psi(S) = (\beta_1 \dots \beta_j \dots \beta_n) \in \{0, 1, *\}^n$, where

$$\beta_j = \begin{cases} 1 & \text{if } \sum_{x^k \in S} x_j^k = |S| \\ 0 & \text{if } \sum_{x^k \in S} x_j^k = 0 \\ * & \text{otherwise} \end{cases}, j = 1, \dots, n \quad (2)$$

We call $\Psi(S)$ the *scheme solution* (the term comes from genetic algorithms, Goldberg, 1989). The consistent variables are thus the ones which correspond to values 1 or 0 in $\Psi(S)$.

From the scheme $\Psi(S)$ it is defined $\Omega(S) = \{j : \beta_j = *, j = 1, \dots, n\}$, *i.e.*, the set of the indices of the free variables, the ones whose values must be defined.

Let $R(S) = (\alpha_1 \dots \alpha_j \dots \alpha_n) \in \{0, 1\}^n$ be the *root solution* associated with S , where $\alpha_j = \beta_j$ if $\beta_j \neq *$ and $\alpha_j = 0$ if $\beta_j = *$.

The combination method will produce solutions preserving the values of the variables with value 1 and 0 in the scheme solution.

The variables associated with $\Omega(S)$ are then used to define the *residual problem*, $P(\Omega(S))$:

$$\begin{aligned} \max \tilde{z}_1(x) &= \sum_{j \in \Omega(S)} c_j^1 x_j \\ \max \tilde{z}_2(x) &= \sum_{j \in \Omega(S)} c_j^2 x_j \\ \text{s.t. :} & \\ & \sum_{j \in \Omega(S)} w_j x_j \leq \widetilde{W} \\ & x_j \in \{0, 1\}, j \in \Omega(S) \end{aligned} \quad (3)$$

where $\widetilde{W} = W - \sum_{j \in \{1, \dots, n\} \setminus \Omega(S)} w_j x_j$.

Solutions to the original problem (1) are obtained by replacing the positions corresponding to ”*” in $\Psi(S)$ with the correspondent values of the variables regarding the solutions of $P(\Omega(S))$. We call this way of obtaining solutions to problem (1) from solutions of $P(\Omega(S))$ *extending solutions of $P(\Omega(S))$* . Let $E(S)$ denote such a set of extended solutions.

3.1.2 Some properties of the residual problems

Regarding the residual problem $P(\Omega(S))$ several propositions can be established:

Proposition 1 $\forall x \in S, \exists \bar{x} \in E(S) : x = \bar{x}$ or \bar{x} dominates x .

Proof. Let $x_{\Omega(S)}$ represent the variables corresponding to $\Omega(S)$ and suppose that $\bar{x} \in E(S)$ is dominated by a solution of S . Because $x_j = \bar{x}_j, j \in \{1, \dots, n\} \setminus \Omega(S)$, if x dominates \bar{x} then $x_{\Omega(S)}$ dominates $\bar{x}_{\Omega(S)}$. But, once $x_{\Omega(S)}$ is a feasible solution of $P(\Omega(S))$ that would mean that $\bar{x} \notin E(S)$, which is false. ■

This proposition assures that the quality of extended efficient solutions of $P(\Omega(S))$ can not degrade concerning S .

Definition 1 A scheme, $\Psi(S) = (\beta_1 \dots \beta_j \dots \beta_n)$ is said to be compatible with a scheme $\Psi(S') = (\beta'_1 \dots \beta'_j \dots \beta'_n)$ iff $\beta_j = \beta'_j, j \notin \Omega(S')$, where $\beta'_j, \beta_j, j = 1, \dots, n$ are defined as (2).

Another interesting result, that is important to disregard some schemes, comes from the proposition below.

Proposition 2 If $\Psi(S')$ is compatible with $\Psi(S)$ then extended efficient solutions of $P(\Omega(S'))$ are equal to or are dominated by extended efficient solutions of $P(\Omega(S))$.

Proof. It suffices to note that if $\Psi(S')$ is compatible with $\Psi(S)$ then $\beta'_j = \beta_j, j \notin \Omega(S)$, and $\Omega(S') \subseteq \Omega(S)$. Hence, $P(\Omega(S))$ is a relaxation of $P(\Omega(S'))$ and then solutions of $P(\Omega(S'))$ are equal or are dominated by solutions of $P(\Omega(S))$. ■

Proposition 2 states that if a problem $P(\Omega(S))$ is solved then it is worthless to combine solutions that lead to a scheme compatible with $\Psi(S)$. Propositions 1 and 2 will be useful to define the reference set in the Reference set update method.

Example 1

Consider the following instance of the bi-criteria $\{0,1\}$ -knapsack problem with 20 items:

$$c_j^1 : 1, 21, 32, 43, 8, 30, 78, 85, 17, 25, 49, 29, 50, 3, 51, 1, 78, 21, 96, 25$$

$$c_j^2 : 4, 28, 17, 9, 85, 92, 33, 72, 33, 83, 15, 78, 89, 15, 3, 78, 71, 69, 65, 68$$

$$w_j : 87, 68, 38, 48, 6, 37, 70, 31, 47, 28, 88, 98, 83, 15, 60, 66, 56, 60, 100, 30$$

$$W = 558$$

Suppose that the following six solutions are already available:

$$\begin{aligned}
x^1 &: (00100001110111110101) \Rightarrow z(x^1) = (339; 605) \\
x^2 &: (00110001010111110101) \Rightarrow z(x^2) = (365; 581) \\
x^3 &: (01111101100010111000) \Rightarrow z(x^3) = (416; 577) \\
x^4 &: (00111001111001110011) \Rightarrow z(x^4) = (435; 543) \\
x^5 &: (10100011110000011011) \Rightarrow z(x^5) = (438; 524) \\
x^6 &: (10111011011011001000) \Rightarrow z(x^6) = (452; 493)
\end{aligned}$$

Consider the combination of the solutions in the subset $S = \{x^4, x^5\}$.

The scheme is: $(*01*0*111*00*011)$, where values in positions marked with "*" are determined by the resolution of the residual problem presented below. The root solution is thus

(00100001110000010011) which corresponds to point $z = (281; 416)$ in the criteria space. The residual capacity is $558-340=218$ and $\Omega(S) = \{1, 4, 5, 7, 11, 14, 15, 17\}$.

The residual problem is thus:

$$\begin{aligned}
\max \tilde{z}_1 &= 1x_1 + 43x_4 + 8x_5 + 78x_7 + 49x_{11} + 3x_{14} + 51x_{15} + 78x_{17} \\
\max \tilde{z}_2 &= 4x_1 + 9x_4 + 85x_5 + 33x_7 + 15x_{11} + 15x_{14} + 3x_{15} + 71x_{17}
\end{aligned}$$

s.t. :

$$87x_1 + 48x_4 + 6x_5 + 70x_7 + 88x_{11} + 15x_{14} + 60x_{15} + 56x_{17} \leq 218$$

$$x_1, x_4, x_5, x_7, x_{11}, x_{14}, x_{15}, x_{17} \in \{0, 1\}.$$

When solving the residual problem with an exact method, two non-dominated solutions are obtained: $\tilde{z}^1 = (210, 213)$ and $\tilde{z}^2 = (218, 207)$. By joining these solutions to the image of the root solution, we obtain: $z^1 = (210, 213) + (281, 416) = (491, 629)$; $z^2 = (218, 207) + (281, 416) = (499, 623)$. As it can be seen in Figure 1 these two solutions dominate the initial solutions and, in fact, all the initial set of solutions.

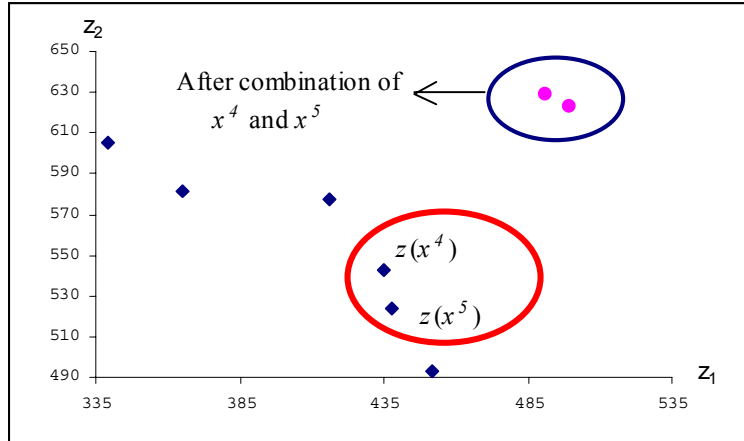


Figure 1: Combining solutions

■

Definition 2 A solution $x^+ = (x_1^+ \dots x_j^+ \dots x_n^+)$ is said to be compatible with the scheme $\Psi(S)$ iff $x_j^+ = \beta_j, j \notin \Omega(S)$.

Efficient solutions of problem (1) are related to extended efficient solutions of residual problems. This is the result of the proposition that follows.

Proposition 3 All the efficient solutions of problem (1) compatible with $\Psi(S)$ are extended efficient solutions of $P(\Omega(S))$, i.e., belong to $E(S)$.

Proof. Let S' be the set of efficient solutions of problem (1) compatible with $\Psi(S)$. Thus $\Psi(S')$ is compatible with $\Psi(S)$. According to Proposition 2 and to the fact that S' contains efficient solutions of problem (1) all the solutions of S' are equal to the extended efficient solutions of $P(\Omega(S))$. ■

In the special case when all the efficient solutions of (1) define a scheme compatible with S , extended efficient solutions of $P(\Omega(S))$ solve problem (1):

Corollary 1 If all the efficient solutions of problem (1) are compatible with $\Psi(S)$ then the extended efficient solutions of $P(\Omega(S))$ solve problem (1).

Proof. Let S' be the set of all the efficient solutions of (1). So, $\Psi(S')$ is compatible with $\Psi(S)$. Proposition 3 proves that all the solutions in S' are extended efficient solutions of $\Psi(S')$.

■

Proposition 3 and Corollary 1 can be extended to describe a more general situation where the set of efficient solutions of (1) can be obtained by solving several smaller problems.

Proposition 4 Let S be the set of efficient solutions of problem (1) and $S^1, S^2, \dots, S^\delta$ be a partition of S . Then S is equal to the union of the extended efficient solutions of $P(\Omega(S^1)), P(\Omega(S^2)), \dots, P(\Omega(S^\delta))$.

Proof. From Proposition 3 and Corollary 1 S^1, \dots, S^δ corresponds to extended efficient solutions of $P(\Omega(S^1)), P(\Omega(S^2)), \dots, P(\Omega(S^\delta))$, i.e., $S^l = E(S^l), l = 1, \dots, \delta$. This concludes the proof once $S^1 \cup S^2 \cup \dots \cup S^\delta = S$ and $S^l \cap S^t = \emptyset, l \neq t, l, t = 1, \dots, \delta$. ■

3.1.3 Structure of the combination method

The solution combination method combines a set of solutions to define a residual problem by identifying consistent variables in that set. The solutions are put in the set S^{view} , which contains the already examined solutions. This is done in order to increase the diversity in the combination process. If the scheme associated with the combined solutions is not compatible with any scheme already tested then the scheme is added to the set of already examined schemes, the residual problem is solved and the set of potentially efficient solutions, \tilde{X} , is updated with the set of the extended efficient solutions obtained.

The new combination method can be outlined as follows.

Procedure Combination_method $(S, T^{sch}, \tilde{X}, S^{view})$

INPUT

S , a subset of solutions
 T^{sch} , set of tabu_schemes
 \tilde{X} , set of potentially efficient solutions
 S^{view} , set of already examined solutions

OUTPUT

\tilde{X} , set of potentially efficient solutions updated

BEGIN

$S^{view} \leftarrow S^{view} \cup S$

IF ($\Psi(S)$ is not compatible with any scheme of T^{sch}) THEN

BEGIN

$T^{sch} \leftarrow T^{sch} \cup \{\Psi(S)\}$

Solve $P(\Omega(S))$

Update \tilde{X} with $E(S)$

END

END

3.2 Reference set update method

The reference set usually combines the quality and diversity features of the solutions (Glover, 1999). In multiple criteria problems there is not a single function to evaluate the solutions, but even though two groups of solutions can be identified: efficient and non-efficient solutions. The concept of efficient solution is used to satisfy the quality requirement of the reference set, R . Let \tilde{Q} be the set of potentially efficient solutions that can be elected to belong to the reference set (at the beginning of the procedure $\tilde{Q} = \tilde{X}$).

Instead of using simple solutions, the reference set works with *pack-solutions*, that is sets of solutions. The packing operation is based on the size of the problem that can be defined upon that set, *i.e.*, on the cardinality of $\Omega(\text{pack-solution})$. Supported by empirical experiments we define the maximum size of the problem, c^{\max} , as follows (when more than one condition is satisfied, the maximum value for the size of the problem is considered):

$$c^{\max} = \begin{cases} 100 & \text{if } |\Omega(\tilde{Q}^{ini})|/3 \leq 100 \\ 80 & \text{if } |\Omega(\tilde{Q}^{ini})|/6 \leq 80 \\ 70 & \text{if } |\Omega(\tilde{Q}^{ini})|/11 \leq 70 \\ 60 & \text{if } |\Omega(\tilde{Q}^{ini})|/17 \leq 60 \\ 50 & \text{otherwise} \end{cases}$$

where \tilde{Q}^{ini} is the first \tilde{Q} set.

The packing operation consists thus of grouping solutions from \tilde{Q} (which is ordered according to increasing values of z_1), in a consecutive manner, while $|\Omega(\text{pack-solution})|$ is not greater than c^{\max} . Let Y^1, Y^2, \dots, Y^q represent the pack-solutions.

In large size instances, q is expected to have a high value. Keeping the spirit of SS, concerning the small size of the population, the maximum size of the reference set is fixed at 30 individuals

(pack-solutions). When q is higher than 30 then it is necessary to ignore some individuals. This is done by trying to keep high the diversity of the reference set.

The problem of selecting individuals in order to obtain a set with the maximal diversity is considered in the works by Kuo *et al.* (1993), Glover *et al.* (1998) and Kochenberger and Glover (1999). Due to the complexity of the maximum diversity problem, heuristic approaches have been proposed. In this paper, the problem is solved by using an adaptation of the constructive heuristic presented in Glover *et al.* (1998), which consists of selecting individuals that maximize the sum of the distances to all the individuals already selected.

In this sense, we initialize the reference set with Y^1 and Y^q . The next member of the reference set is the pack-solution which maximizes the minimum Hamming distance to the members of the reference set ($\Psi(Y^1)$ and $\Psi(Y^q)$). The operation is repeated by computing the Hamming distance to all the members of the reference set until the reference set achieves its maximum cardinality, *i.e.*, 30 individuals.

The Reference set update method is as follows.

Procedure Reference set update method($\gamma, \tilde{Q}, c^{max}$)

INPUT

γ , maximum number of solutions in the reference set
 \tilde{Q} , solutions that can be selected to belong to the reference set
 c^{max} , maximum cardinality for residual problems

OUTPUT

R , reference set

BEGIN

Create the pack-solutions Y^1, \dots, Y^q , from \tilde{Q} , such that $\Omega(Y^u) \leq c^{max}, u = 1, \dots, q$

IF $q \leq \gamma$ THEN $R \leftarrow \{Y^1, \dots, Y^q\}$

ELSE

BEGIN

$R \leftarrow \{Y^1, Y^q\}$

WHILE $|R| < \gamma$ DO

BEGIN

Find $Y^* = \arg \max_{Y^u \notin R} \min_{Y^v \in R} \{H(\Psi(Y^u), \Psi(Y^v))\}$

$R \leftarrow R \cup Y^*$

END

END

END

In the above procedure $H(a, b)$ is the Hamming distance between a and b .

During the iteration, and taking into account the results of the combination method, set \tilde{Q} is only composed of the potentially efficient solutions that were not yet combined in the previous iterations. This will avoid the analysis of very similar problems that could emerge if all the available potentially efficient solutions were considered, which is not useful for generate new efficient solutions, as results from Proposition 2.

3.3 Subset generation method

This method generates subsets of pack-solutions from the reference set, R . Each pack-solution is associated to a residual problem defined as in Section 3.1.1. Taking into account the structure of the combination method, the subsets can only contain pack-solutions that lead to small residual problems. This was observed in the construction of the Reference set update method. To define the subsets we proceed as follows. If the reference set, R , is composed of at most three pack-solutions then each of them is considered a subset. This is the general case for small/medium size instances. If the number of pack-solutions is higher than three, then the difficulty of the corresponding residual problem will be assessed (Section 3.3.2), maybe requiring the decomposition of the pack-solution.

3.3.1 Difficulty of residual problems

There are several factors which influence the difficulty of residual problems:

- 1) number of items;
- 2) relation between the capacity of the knapsack and the sum of the weights of the items;
- 3) linear correlation between one criterion and the knapsack constraint;
- 4) linear (negative) correlation between criteria.

The importance of the first three is well-known for single criterion problems (Martello and Toth, 1990; Martello and Pisinger, 1998). The influence of the linear correlation between criteria is shown in Example 2.

Example 2

Consider the data from Example 1 and let the new coefficients of criterion z_1 be:

$\bar{c}_j^1 : 108, 81, 92, 103, 18, 12, 75, 35, 73, 27, 97, 28, 16, 97, 103, 34, 34, 39, 40, 42.$

The coefficient of linear correlation between criteria z_1 and z_2 changes thus from 0.0876 to -0.99721 . The original problem has 15 efficient solutions while the modified problem comprises 54.

The influence of the linear correlation between criteria concerns the number of solutions. As shown, the increase in the linear correlation (negative) between criteria leads to an increase in the number of efficient solutions, and consequently to a more difficult problem. ■

These factors must be taken into account in the design of the algorithm. Despite the original distribution of the data, residual problems "well-arranged" can appear. The figures below show the original data of a problem with 2,000 items and the data corresponding to a residual problem. As it can be seen, despite the original distribution, strong correlation between criteria and the knapsack constraint was created (Figure 6).

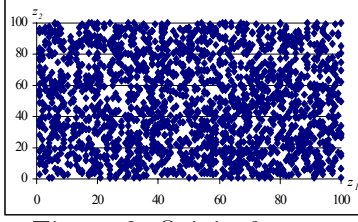


Figure 2: Original z_1z_2

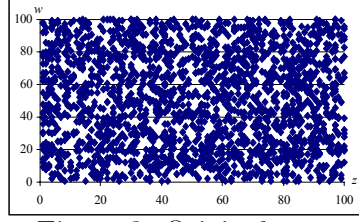


Figure 3: Original z_1w

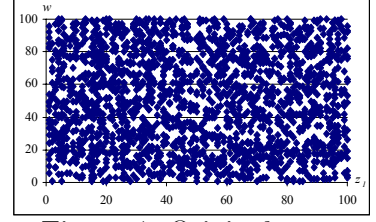


Figure 4: Original z_2w

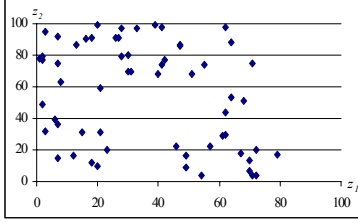


Figure 5: Subproblem z_1z_2

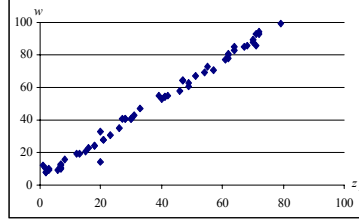


Figure 6: Subproblem z_1w

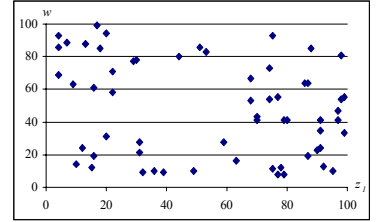


Figure 7: Subproblem z_2w

The emergence of these small but difficult problems is very interesting in the present framework. Note that when using partial optimization strategies the aim is to obtain the entire set of the solutions through a process of decomposition of that problem. But if the residual problems produce the average number of solutions according to their size, then a very small number of solutions to the original problem would be obtained. The difficulty of the sub-problems, giving rise to a much higher number of solutions, seems to be the way to ensure the quality of the approximation.

3.3.2 Structure of the subset generation method

The difficulty associated with a pack-solution is evaluated by the linear correlation coefficient between the two criteria, and one criterion and the knapsack constraint of the subproblem composed of variables corresponding to " * " in the scheme of the pack-solution. The limit for the coefficient of linear correlation between one criterion and the knapsack constraint was set to 0.5, and -0.5 for the coefficient of linear correlation between criteria. Once these values are achieved the residual problem is reduced by dividing each pack-solution into five subpacks of approximately the same size. The construction of those is based on the selection of consecutive solutions.

For each residual problem corresponding to a subpack solution, a maximum size, Θ , is defined. The rule is:

$$\Theta = \begin{cases} 0.3 \times c^{\max} & \text{if } 70 < \delta \leq 80 \\ 0.35 \times c^{\max} & \text{if } 60 < \delta \leq 70 \\ 0.4 \times c^{\max} & \text{if } 50 < \delta \leq 60 \\ 0.5 \times c^{\max} & \text{if } 25 < \delta \leq 50 \\ 25 & \text{if } \delta \leq 25 \end{cases}$$

where δ is the size of the residual problem.

If the residual problem corresponding to a subpack solution has a larger size then some variables have to be fixed in order to obtain an appropriate sized residual problem. The procedure

of fixing additional variables is based on the profit-to-weight ratio heuristic. In the presence of more than one criterion the ratio is not well defined, since it depends on the constructed weighted sum function. Nevertheless, the ratio is bounded from above and below by

$$\min \left\{ \frac{c_j^1}{w_j}, \frac{c_j^2}{w_j} \right\} \leq \frac{\lambda c_j^1 + (1 - \lambda) c_j^2}{w_j} \leq \max \left\{ \frac{c_j^1}{w_j}, \frac{c_j^2}{w_j} \right\}.$$

Thus, to fix variables to zero we use the following rule: $\min_j \left\{ \max \left\{ \frac{c_j^1}{w_j}, \frac{c_j^2}{w_j} \right\} \right\}$.

Due to the division of the pack-solutions and to the size of the residual problem defined by all the initial solutions (Section 3.2), the number of subsets is dynamic and can reach a maximum of 5 times the number of pack-solutions in the reference set, R .

The subset generation method is summarized below.

Procedure Subset Generation Method($R, \alpha, \rho_{z_1 z_2}^{\max}, \rho_{z_1 w}^{\max}, \rho_{z_2 w}^{\max}, \Theta$)

INPUT

$R, \{Y^1, Y^2, \dots, Y^q\}$ sets of pack-solutions

α , number of sub-packs

$\rho_{z_1 z_2}^{\max}$, maximum coefficient of linear correlation between z_1 and z_2

$\rho_{z_1 w}^{\max}$, maximum coefficient of linear correlation between z_1 and knapsack constraint

$\rho_{z_2 w}^{\max}$, maximum coefficient of linear correlation between z_2 and knapsack constraint

Θ , size of the residual problem

OUTPUT

$S^{11}, \dots, S^{1\alpha}, \dots, S^{q1}, \dots, S^{q\alpha}$, subsets of solutions

BEGIN

IF $|R| \leq 3$ THEN $S^t \leftarrow \{Y^t\}, t = 1, \dots, q$

ELSE

FOR $t=1$ TO q DO

BEGIN

$\rho_{z_1 z_2} \leftarrow$ linear correlation between z_1 and z_2 in $\Omega(Y^t)$

$\rho_{z_1 w} \leftarrow$ linear correlation between z_1 and knapsack constraint in $\Omega(Y^t)$

$\rho_{z_2 w} \leftarrow$ linear correlation between z_2 and knapsack constraint in $\Omega(Y^t)$

IF $(-\rho_{z_1 z_2} > \rho_{z_1 z_2}^{\max})$ OR $(\rho_{z_1 w} > \rho_{z_1 w}^{\max})$ OR $(\rho_{z_2 w} > \rho_{z_2 w}^{\max})$ THEN

BEGIN

Divide Y^t into approximately equal sized sets: $Y^{tv}, v = 1, \dots, \alpha$

FOR $v = 1$ TO α DO

BEGIN

WHILE $\Omega(Y^{tv}) > \Theta$ DO

BEGIN

$$\xi \leftarrow \arg \min_{j \in \Omega(Y^{tv})} \left\{ \max \left\{ \frac{c_j^1}{w_j}, \frac{c_j^2}{w_j} \right\} \right\}$$

$$x_\xi^k \leftarrow 0, \forall x^k \in Y^{ij}$$

END

$$S^{tv} \leftarrow \{Y^{tv}\}$$

END

END

END

END

For combination purposes, the pack-solutions in the subsets have to be unpacked, *i.e.*, the corresponding x solutions must be considered. With the unpack operation, the subsets are composed of x solutions. In the above description of the structure of the subset generation method it was assumed that α is equal to 5; $\rho_{z_1 z_2}^{\max} = -0.5$ and that $\rho_{z_1 w}^{\max} = \rho_{z_2 w}^{\max} = 0.5$.

3.4 Overall description of the scatter search method

The new SS method for the bi-criteria $\{0,1\}$ -knapsack problem starts by applying the diversification method, followed by the improvement of the solutions (both presented in Gomes da Silva *et al.*, 2003a). After the improvement method, all the solutions to the original problem are feasible. The efficient solutions are selected and put in sets \tilde{X} (the set of potentially efficient solutions) and \tilde{Q} (the set of solutions to define the reference set at each iteration). The method repeats iteratively the reference set update method (Section 1.3), subset generation method (Section 1.4) and the combination method (Section 1.2) until the defined stopping condition is achieved. Set \tilde{X} is updated with the extended solutions from the combination method. This corresponds to a progressive improvement of the approximation, as results from Proposition 1.

The SS method has thus the following structure:

SS – Method ($\gamma, c^{\max}, \Theta, \alpha, \rho_{z_1 z_2}^{\max}, \rho_{z_1 w}^{\max}, \rho_{z_2 w}^{\max}$)

INPUT

γ , maximum number of solutions in the reference set

c^{\max} , maximum cardinality for residual problems

Θ , size of the residual problem when a problem is decomposed

α , number of sub-pack solutions

$\rho_{z_1 z_2}^{\max}$, max. coefficient of linear correlation between z_1 and z_2

$\rho_{z_1 w}^{\max}$, max. coefficient of linear correlation between z_1 and knapsack constraint

$\rho_{z_2 w}^{\max}$, max. coefficient of linear correlation between z_2 and knapsack constraint

OUTPUT

\tilde{X} , set of potentially efficient solutions

BEGIN

$\tilde{X} \leftarrow \emptyset; \tilde{Q} \leftarrow \emptyset; T^{sch} \leftarrow \emptyset; S^{view} \leftarrow \emptyset$

Diversification method

Improvement method

$\tilde{X} \leftarrow \{\text{potentially efficient solutions after **Improvement method**}\}$

$\tilde{Q} \leftarrow \tilde{X}$

REPEAT

BEGIN

Reference set update method ($\gamma, c^{\max}, \tilde{Q}$)

Let R be the reference set from the reference set update method

Subset Generation Method ($\alpha, \Theta, \rho_{z_1 z_2}^{\max}, \rho_{z_1 w}^{\max}, \rho_{z_2 w}^{\max}, R$)

Let $S^{11}, \dots, S^{1\alpha}, \dots, S^{q1}, \dots, S^{q\alpha}$ be the subsets from the subset generation method

Combination Method ($S^{tv}, T^{sch}, \tilde{X}, S^{view}$), $t = 1, \dots, q; v = 1, \dots, \alpha$

$\tilde{Q} \leftarrow \tilde{X} \setminus S^{view}$

END

UNTIL Stopping-condition

END

Three natural stopping condition are: $|\tilde{Q}| \leq 1; |R| = 1$; or a given number of iterations. The $|\tilde{Q}| \leq 1$ condition holds because it means that, at most, only one solution is available to generate others, but no additional schemes can be defined, so, no new solutions can be found. The $|R| = 1$ condition also holds once all the solutions are considered, defining a scheme with which all the other schemes derived from the obtained extended solutions are compatible with.

4 Computational experiments and results

This section deals with the computational experiments and the results obtained with the proposed new SS based method. Small, medium and large size instances are considered, with a number of items from 100 to 6,000. The computational experiments were performed on a Pentium 4 processor with 256 MB RAM and 40 GB hard disk at 1,495 Mhz. The method was implemented in Borland Delphi 4.

The coefficients were randomly and uniformly generated within the range $[1,100]$ and the capacity of the knapsack was set to 50% of the sum of the items. The stopping condition of the new SS based method was set as the first verified condition: 4 iterations; $|\tilde{Q}| \leq 1; |R| = 1$.

4.1 Small and medium size instances

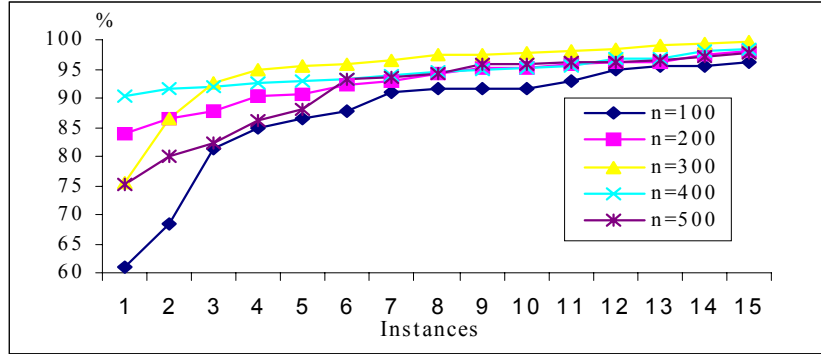
In this section, the set of potentially non-dominated solutions, *PNDS*, obtained with the new SS based method is compared with the exact set of non-dominated solutions, *NDS*. The percentage of the exact non-dominated solutions obtained (m_1) is computed as a measure of the effectiveness of the method. The percentage of dominated solutions among *PNDS* which are dominated by solutions in *NDS* (m_2) is also computed. To obtain the exact set of non-dominated solutions the method by Visée *et al.*, 1998 was implemented.

Instances with $n = 100, 200, 300, 400$ and 500 were considered and for each problem size 15 instances were built. The overall results are presented in Table 1.

As it can be observed the new SS based method was able to find, on average, between 87% to 95% of the exact non-dominated solutions (column m_1) within a low computational time, in seconds, (columns $CPU_ (1)$), which is less than the time involved in solving exactly the problem ($CPU_ (2)$). This is indeed a very important feature. Figure 8 shows the percentage of exact non-dominated solutions for all the instances. For a better visualization, the values are shown in an increasing manner. As it can be observed, the worst percentages are very few. The majority of the percentages are very high.

n	<i>Statistic</i>	<i>NDS</i>	<i>CPU_(1)</i>	$\Omega(\widetilde{Q}_1)$	<i>PNDS</i>	m_1	m_2	<i>CPU_(2)</i>
100	<i>Average</i>	127.3	4.2	39.3	122.9	87.3	7.8	1.3
	<i>Max.</i>	176	7.1	44.0	173	96.1	32.4	2.9
	<i>Min.</i>	73	2.2	32.0	66	61.0	0.8	0.3
	<i>STD</i>	25.8	1.2	3.6	26.5	9.8	8.9	0.7
200	<i>Average</i>	374.5	68.3	78.7	363.3	92.8	4.2	17.2
	<i>Max.</i>	539	90.7	92.0	524	98.3	10.5	31.7
	<i>Min.</i>	241	40.9	64.0	232	83.7	0.3	7.7
	<i>STD</i>	68.9	14.7	7.2	69.5	4.1	3.0	6.2
300	<i>Average</i>	792.9	507.3	120.5	781.4	95.0	3.8	340.3
	<i>Max.</i>	917	807.9	133.0	905	99.7	20.6	552.5
	<i>Min.</i>	612	306.3	106.0	605	75.6	0.1	149.7
	<i>STD</i>	86.9	152.7	7.3	85.5	6.1	5.3	110.1
400	<i>Average</i>	1149	1681.8	156.3	1133.1	94.5	4.2	540.0
	<i>Max</i>	1439	2891.3	186.0	1413	98.3	8.9	1214.9
	<i>Min</i>	936	1155.7	140.0	916	90.3	1.0	383.0
	<i>STD</i>	118.4	445.9	10.4	115.9	2.3	2.2	203.6
500	<i>Average</i>	1583.4	5171.4	193.1	1547.3	91.2	6.8	1045.4
	<i>Max.</i>	1830	7706.5	210.0	1802	97.9	20.0	1870.1
	<i>Min.</i>	1348	3370.6	176.0	1330	75.0	1.4	539.6
	<i>STD</i>	144.1	1292.7	10.2	147.7	6.9	5.5	384.8

Table 1: Overall results on the set of PNDS



Percentage of exact non-dominated solutions

Due to the high percentage of exact non-dominated solutions the number of PNDS (column PNDS) is very similar to the number of exact solutions (column NDS) and consequently the average value of the dominated solutions found is low (column m_2).

Concerning the size of the scheme constructed with the initial set of solutions (column $|\Omega(\tilde{Q}_1)|$) one can see that small problems were solved (residual problems with a small number of items) when the number of items was less than 200. When the number of items was greater or equal to 200, the division of the set $\Omega(\tilde{Q}_1)$ (Section 3.2) occurred. It can be seen (Figure 9) that the size of the initial scheme increases linearly with the size of the problem.

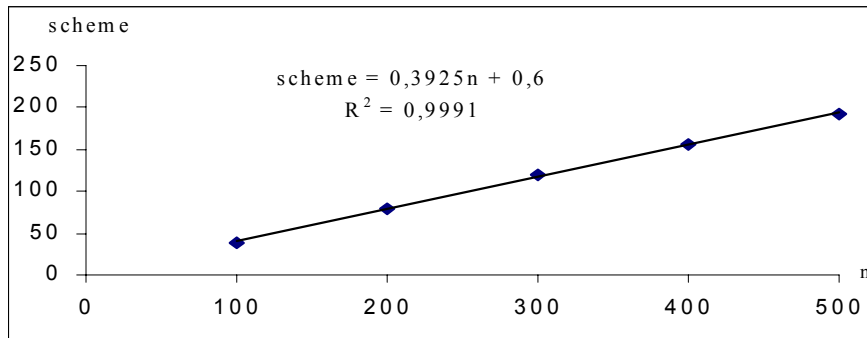


Figure 9: Relation between the size of the original scheme and the size of the problem

4.2 Large size instances

Instances with a number of items from 1,000 to 6,000 were considered. Table 2 presents the size of the scheme constructed with the initial set of solutions. This size also seems to grow linearly with the problem size and, on average, the associated residual problem is composed of approximately 37% of the variables of the original problem, and, as a result, the existent exact methods can not solve efficiently the residual problem when n is greater than 1,000.

In order to incorporate diversity in the process of search for efficient solutions, once a solution is used in the combination process, it is discarded (Section 3.2). Table 3 presents the number of available solutions to define the reference set along the iterations. It is common to all the

n	1000	2000	3000	4000	5000	6000
$\Omega(\tilde{Q}_1)$	357	746	1089	1514	1858	2237

Table 2: Size of the initial scheme

n	\tilde{Q}_1	\tilde{Q}_2	\tilde{Q}_3	\tilde{Q}_4
1000	536	2627	438	362
2000	1228	6035	2402	813
3000	1757	6840	7217	7417
4000	2618	8132	9401	7352
5000	3232	8778	11377	10457
6000	4119	9801	11150	12531

Table 3: Number of PNDS to build the reference set, per iteration

instances a great increase in the number of solutions after the combination method had been applied (\tilde{Q}_2). This means that a big number of new and better solutions are found just with the first iteration of that method. Due to the structure of the new SS method it is expected a decrease of set \tilde{Q} , and when no new solutions are found then \tilde{Q} becomes empty. Observing Table 3, the continuous decrease in set \tilde{Q} was not verified in the 4 iterations limit, particularly when $n \geq 300$ (to distinguish the set \tilde{Q} along the iterations we use \tilde{Q}_t representing the set \tilde{Q} at beginning of iteration t).

At each iteration new efficient solutions are found, as can be observed in Table 4, by the increasing number of solutions in \tilde{X} during the iterations (to distinguish the set \tilde{X} along the iterations we use \tilde{X}_t representing the set \tilde{X} after iteration t). The number of solutions in the terminal set is already high just after 4 iterations. On average it is about 5 times the initial set of solutions.

Due to the structure of the combination method and to the results for the small, medium size instances, it is expected that a significant part of \tilde{X} corresponds indeed to exact efficient solutions.

Concerning the computational time, all the efficient solutions were obtained within one hour (see Table 5). This is an interesting running time given the fact that a) an exact method is

n	\tilde{X}_1	\tilde{X}_2	\tilde{X}_3	\tilde{X}_4
1000	2702	2819	2898	2920
2000	6249	6984	7146	7201
3000	7079	9432	10493	10849
4000	8433	11726	13149	14289
5000	9033	12999	15082	16683
6000	10118	13265	16113	17733

Table 4: Number of potentially efficient solutions, per iteration

n	t_0	t_1	t_2	t_3	t_4	$Total$
1000	2.360	51.580	54.210	46.410	25.710	180.270
2000	11.970	212.170	355.650	248.420	108.920	937.130
3000	27.850	201.140	507.510	658.390	537.340	1932.230
4000	55.200	279.730	632.970	800.370	697.000	2465.270
5000	87.390	286.050	656.520	719.800	904.850	2654.610
6000	135.720	368.330	668.000	925.330	1336.830	3434.210

Table 5: CPU (s), per iteration and total

incorporated; b) the number of obtained solutions is high; and c) the number of items is also high.

The second column in Table 5, t^0 , concerns the diversification method, followed by the improvement method. Columns 3-6 (t^1, t^2, t^3, t^4) concern the computational time involved in each complete iteration of the method. As it can be observed, the CPU time per iteration is not stable. This is due to the variability of \tilde{Q} : as shows Figure 10, the computational time per iteration is highly related (linearly) with the number of solutions in \tilde{Q} .

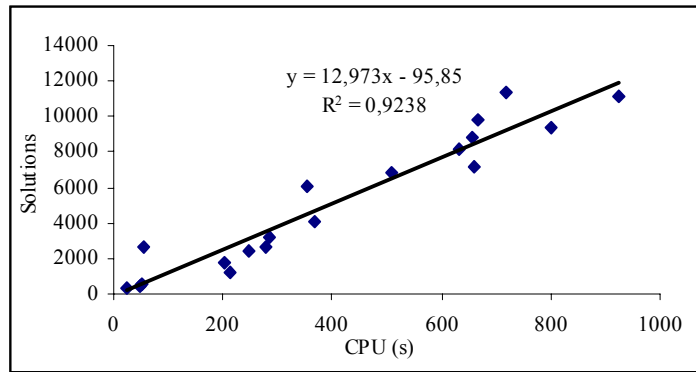


Figure 10: Linear relation between CPU time and the size of \tilde{Q}

The quality of solutions is measured by taking into account the proximity to the upper frontier, derived from the linear relaxation, and the diversity of solutions (Zitzler, 1999; Deb, 2001; and Coello *et al.*, 2002). Considering the first feature, the L_∞ metric is used to determine the nearest point, (z_1^*, z_2^*) , in the surrogate upper frontier of each PNDS, (z_1^+, z_2^+) . These two points are then used to derive the gradient of a weighted sum function: $f(z) = \pi_1 z_1 + \pi_2 z_2$, with $z = (z_1, z_2)$, $\pi_1 = z_1^* - z_1^+$ and $\pi_2 = z_2^* - z_2^+$. The percentage gap between points z^+ and z^* is calculated using $f(z) : \frac{f(z^*) - f(z^+)}{f(z^*)} \times 100$.

Diversity is evaluated based on two measures: 1) the standard deviation of the Euclidean distances between consecutive PNDS in the criteria space; and 2) the standard deviation of the number of solutions *per* region: the upper frontier is used to divide the criteria space into regions. In order to make this, the range of criterion z_1 was divided in 20 equal size intervals, that projected into the space $z_1 z_2$ gave rise to 20 regions. The expected number of solutions *per* region was computed and the standard deviation of the number of solutions *per* region was considered. The lower this value the greater the dispersion of the obtained solutions along the upper frontier.

n	L_∞ distance (%)				Diversity	
	Average	Max	Min	STD	STD_g	STD_d
1000	0.0062	0.0197	0.0000	0.0000	28.0488	5.5993
2000	0.0017	0.0074	0.0000	0.0000	42.5117	5.3429
3000	0.0009	0.0118	0.0000	0.0000	52.5817	4.8959
4000	0.0007	0.0112	0.0000	0.0000	60.4638	6.1093
5000	0.0007	0.0067	0.0000	0.0000	65.8008	5.9412
6000	0.0005	0.0061	0.0000	0.0000	65.9858	7.2372

Table 6: Quality of solutions

For the obtained set of PNDS, the quality is quantified in Table 3. The average of the percentual L_∞ distance is quite low, even for the maximum observed value, revealing a very good proximity to the upper frontier. The standard deviation of the distances is also very low. Concerning the diversity among the solutions along the upper frontier a relative low value for the corresponding standard deviation (STD_g) can be observed. The standard deviation of the distance between consecutive potentially non-dominated solutions (STD_d) is quite low, which is good, in conjunction with the values of $STD - g$, for the coverage of the upper frontier.

5 Conclusions

In this paper a combination of systematic and heuristic search strategies were put to work together in the context of a SS metaheuristic. The systematic search was performed by considering partial optimization in restricted regions of the decision space. The concept of consistent variable was used to define interesting residual problems. Several properties were derived for the residual problem, which were used to evaluate the quality of the solutions and to reduce the number of problems to be solved exactly.

Results on small and medium size instances show that the integration of partial optimization with SS enables the determination of a significant part of the exact set of efficient solutions. Concerning large size instances, the approach revealed to produce a considerable amount of additional solutions compared with the previous version of SS (Gomes da Silva *et al.*, 2003b).

The approach presented here requires the existence of specialized exact methods which work efficiently for, at least, small/medium size instances of the problem. This is the case of the bi-criteria $\{0,1\}$ -knapsack, but also of many other multiple criteria problems, including the shortest path problem (Skriver and Andersen, 2000; Clímaco and Martins, 1982), the spanning tree problem (Ramos *et al.*, 1998) and the assignment problem (Malhotra *et al.*, 1982). Thus, this research can be also useful in those problems, enlarging the size of the instances that can be considered.

Acknowledgements: The authors would like to acknowledge the support from MONET research project (POCTI/GES/37707/2001) and the second author also acknowledges the support of the grant SFRH/BDP/6800/2001 (Fundação para a Ciência e Tecnologia, Portugal).

References

- [1] Ben Abdelaziz, F., Krichen, S. (1997) "A tabu search heuristic for multiobjective knapsack problem", RUTCOR, *Research Report*, Rutgers University, New Jersey
- [2] Captivo, M., Clímaco, J., Figueira, J., Martins, E., Santos, J.L. (2003) "Solving multiple criteria 0-1 knapsack problems using a labeling algorithm", *Computers & Operations Research*, 30. pp. 1865-1886
- [3] Clímaco, J., Martins, E. (1982) "A bicriterion shortest path algorithm", *European Journal of Operations Research*, 11, pp. 399-404
- [4] Coello, C., Veldhuizen, D., Lamont, G. (2002) *Evolutionary Algorithms for Solving Multi-objective problems*, Kluwer Academic Publishers, Dordrecht
- [5] Czyzak, P, Jaskiewicz, A. (1998) "Pareto simulated annealing - a metaheuristic technique for multiple objective combinatorial optimization", *Journal of Multicriteria Decision Analysis*, 7, pp. 34-47
- [6] Deb, K. (2001) *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, New York
- [7] 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, pp. 361-383.
- [8] Gandibleux, X., Morita, H., Katoh, N. (2001) "The supported solutions used as a genetic information in a population heuristic" in E. Zitzler, K. Deb., L. Thiele, C.A. Coello Coello, D. Corne (Eds), *Proceedings of the first international conference on evolutionary multi-criterion optimization*, Lecture Notes in Computer Science 1993, Springer, Berlin, pp. 429-442
- [9] Glover, F., Kuo, C.C., Dhir, K.S. (1998) "Heuristic algorithms for the maximum diversity problem", *Journal of Information & Optimization Sciences*, Vol. 19, pp. 109-132
- [10] Glover, F. (1999) "Scatter search and path relinking" in D. Corne, M. Dorigo and F. Glover (Eds), *New Ideas in Optimization*, McGraw Hill, pp. 297-316
- [11] Gomes da Silva, C., Figueira, J., Clímaco, J. (2003a) "An interactive procedure dedicated to the bi-criteria knapsack problem" (in Portuguese), *Research Report*, N°4, INESC-Coimbra, Portugal . http://www.inescc.pt/download/RR2003_04.pdf
- [12] Gomes da Silva, C., Clímaco, J., Figueira, J. (2003b) "A scatter search method dedicated to the bi-criteria knapsack problems", *Research Report*, N°7, INESC-Coimbra, Portugal. http://www.inescc.pt/download/RR2003_07.pdf
- [13] Kochenberger, G., Glover, F. (1999) "Diversity Data Mining", *Working Paper*, Hearin Center for Enterprise Science, University of Mississippi
- [14] Kuo, C.C., Glover, F., Dhir, K.S. (1993) "Analyzing and modeling the maximum diversity problem by zero-one programming", *Decision Sciences*, Vol. 24, pp. 1171-1185

- [15] Laguna, M. (2002) "Scatter Search" in Pardalos, P., Resende, M. (Eds.), *Handbook of Applied Optimization*, Oxford University Press, Oxford, pp. 51-234
- [16] Laguna, M., Martí, R. (2003) *Scatter Search - Methodology and Implementation in C*, Kluwer Academic Publishers, London
- [17] Malhotra, R., Bhatia, H.L., Puri, M.C. (1982) "Bi-criteria assignment problem", *RAIRO, Recherche Opérationnelle/Operations Research*, 19, pp. 84-96
- [18] Mautor, T. (2001) "Intensification neighborhoods for local search method" in Ribeiro, C., Hansen, P. (Eds), *Essays and Surveys in Metaheuristics*, Kluwer Academic Publishers, Boston
- [19] Ramos, R.M., Alonso, S., Sicilia, J., González, C. (1998) "The problem of the optimal biobjective spanning tree", *European Journal of Operational Research*, 111, pp. 617-628
- [20] Skriver, A.J.V., Andersen, K.A. (2000) "A label correcting approach for solving the bicriterion shortest-path problems", *Computers & Operations Research*, 27, pp. 507-524
- [21] Visée, M., Teghem, J., Ulungu, E.L. (1998) "Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem", *Journal of Global Optimization*, 12, pp. 139-155.