Core problems in the bi-criteria {0,1}-knapsack: new developments*

Carlos Gomes da Silva(2,3,†), João Clímaco(1,3) and José Figueira(3,4)

(1) Faculdade de Economia da Universidade de Coimbra
Av. Dias da Silva, 165, 3004-512 Coimbra, Portugal
Phone: +351 239 790 500, Fax: +351 239 790 514

(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

(4) CEG-IST, Center for Management Studies, Departamento de Engenharia e Gestão
Instituto Superior Técnico
Tagus Park, Av. Prof. Cavaco Silva
2780-990 Porto Salvo, Portugal
Phone: +351 21 423 32 99, Fax: +351 21 423 35 68
Email: figueira@ist.utl.pt

October 15, 2005

Abstract

The most efficient algorithms for solving the single-criterion {0,1}-knapsack problem are based on the concept of core, *i.e.*, a small number of relevant variables. But this concept goes unnoticed when more than one criterion is taken into account. The main purpose of the paper is to check whether or not such a set of variables is present in bi-criteria {0-1}-knapsack instances. Extensive numerical experiments have been performed considering five types of {0,1}-knapsack instances. The results are presented for supported, non-supported and for the entire set of efficient solutions.

Key-words: Bi-criteria knapsack problem, Core problem, Combinatorial optimization

*Also published as a research report of CEG-IST

†Corresponding author

1 Introduction

The {0,1}-knapsack problem is about selecting a set of items such that the sum of their values is maximized and the sum of their weights does not exceed the capacity of the knapsack. The {0,1}-*knapack problem* can be mathematically formulated as follows:

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

where, n is the set of available items, p_j is the value of item j ($j = 1, ..., n$), w_j is the weight of item j, W is the knapsack capacity, $x_j = 1$ if item j is selected and $x_j = 0$, otherwise.

Dantzig (1957) showed that an optimal solution for the continuous {0,1}-knapsack problem can be obtained by sorting the items according to non-increasing *profit-to-weight ratios* (also called *efficiencies)*, and including them until the knapsack capacity is full. At the end there is just one item which cannot be wholly included. This item, b, is called the *break* or *critical item* and it is such that $\sum_{j=1}^{b-1} w_j \leq W < \sum_{j=1}^{b} w_j$.

With the items ordered such that

$$\frac{p_1}{w_1} \geq ... \geq \frac{p_j}{w_j} \geq ... \geq \frac{p_n}{w_n} \tag{2}$$

the optimal solution of the continuous {0,1}-knapsack, $\overline{x}$, also called in this paper *Dantzig solution*, is thus:

$$\overline{x}_j = \begin{cases} 1 & j < b \\ \dfrac{W - \sum_{t=1}^{b-1} w_t}{w_b} & j = b \\ 0 & j > b \end{cases}, j = 1, ..., n \tag{3}$$

Balas and Zemel (1980) observed that for randomly generated instances the optimal solution for (1) is very similar to the Dantzig solution. This similarity lead us to introduce the concept of core. Assuming that x^* is an optimal solution of problem (1), the *core* is $C = \{j_1, ..., j_2\}$, where $j_1 = \min\left\{j : x_j^* = 0, j = 1, ..., n\right\}$ and $j_2 = \max\left\{j : x_j^* = 1, j = 1, ..., n\right\}$.

The *core* is thus a subset of items, with efficiencies similar to the efficiency of the break item, that must be considered to determine the exact solution, thereby defining the so-called *core problem.* Results for large size instances showed that the size of the core is a very small proportion of the total number of items, and it increases very slowly with the latter (Balas and Zemel, 1980), which supports the existence of a small, but relevant problem. The concept of core was the foundation for the development of the most efficient known algorithms for the {0,1}-knapsack: Fayard and Plateau (1982), Martello and Toth (1988), and Pisinger (1995). The first two approaches used an approximation of the core and set the values of all the variables outside the core to 1 and 0. The original problem was thus reduced to comprise only the items that pertained to the core. The use of the concept of core evolved (see Martello *et al.*, 1999 for a description of the use of the core in the construction of knapsack algorithms) and Pisinger (1997) showed that the core could be determined when running the algorithm, during the determination of the optimal solution, thus avoiding guessing the core.

In the single criterion {0,1}-knapsack problems the concept of core is quite important because the complete sorting of the items required for deriving better upper and lower bounds is not necessary. As Balas and Zemel (1980) note, it absorbs a very significant part of the total computational time. It is also important because the solution of the core problem can further be used to provide improved lower bounds for the optimal solution of the original problem, thus making it possible to fix the value of a significant number of variables at their optimal value.

Despite the importance of the concept of core it passes unnoticed in the study of multiple criteria {0,1}-knapsack problems, *i.e.*, when several conflicting criteria are considered.

The paper sets out to investigate the presence of the features pointed out by Balas and Zemel (1980), *i.e.*, the structure of the core, in bi-criteria problem solutions:

$$\begin{array}{l} \max z_1(x) = \sum_{j=1}^{n} c_j^1 x_j \\ \max z_2(x) = \sum_{j=1}^{n} c_j^2 x_j \\ \text{s.t.}: \\ \sum_{j=1}^{n} w_j x_j \leq W \\ x_j \in \{0,1\}, j = 1, ..., n \end{array} \quad (4)$$

where, c_j^i represents the value of item j on criterion $i, i = 1, 2$. We assume that c_j^1, c_j^2, W and w_j are positive integers and that $w_j \leq W$ $j = 1, ..., n$ with $\sum_{j=1}^{n} w_j > W$. Constraints $\sum_{j=1}^{n} w_j x_j \leq W$ and $x_j \in \{0, 1\}, j = 1, ..., n$, define the feasible region in the *decision space*, and their image when using the criteria functions $z_1(x)$ and $z_2(x)$ define the feasible region in the *criterion space* - the spaces in which the solutions and their images under the criteria functions $z_1(x)$ and $z_2(x)$ are contained. A feasible *solution*, x, is said to be *efficient* if and only if there is no feasible solution, y, such that $z_i(y) \geq z_i(x), i = 1, 2$ and $z_i(y) > z_i(x)$ for at least one i. The image of an efficient solution in the criterion space is called a *non-dominated solution.*

In this paper, solving problem (4) consists of determining the set of all the efficient/non-dominated solutions. Figure 1 shows the set of non-dominated solutions of an instance with 100 items, and with the coefficients randomly generated from an uniform distribution.

Certain efficient/non-dominated solutions can be obtained by maximizing weighted-sums of the criteria, called supported efficient/non-dominated solutions, but there is a set of solutions, called *non-supported efficient/non-dominated* solutions, that cannot be obtained in this way, because despite being efficient/non-dominated, they are convex dominated by weighted-sums

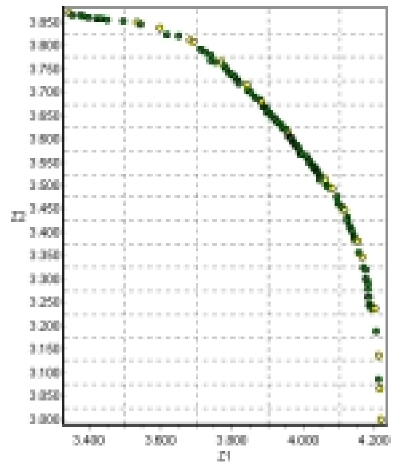

Figure 1: Non-dominated solutions of a random instance

of the criteria. The non-supported non-dominated solutions are located in the dual gaps of consecutive supported non-dominated solutions (Steuer, 1986).

The process of solving the bi-criteria problem can benefit considerably from the developments proposed for solving the single criterion one. In fact, solving (4) can be summarized as the computation of solutions which maximize weighted-sum functions (supported efficient solutions), *i.e.*, single criterion problems, and the computation of solutions that are in the way of those maximizations, just before reaching their optima, *i.e.*, approximate solutions of single criterion optimizations. Consequently, for the supported efficient solutions the available results for the single criterion {0,1}-knapsack problem are valid. As far as we are aware there has been no study on the non-supported efficient solutions.

Despite the similarities between problems (1) and (4) the known algorithms for solving (4) (Ulungu *et al.*, 1997; Visée *et al.*, 1998; Captivo *et al.*, 2003) are limited in comparison with those proposed for solving the single criterion {0,1}-knapsack, in terms of both computational time and the number of instances which can be solved. Even the approximate methods raise too many problems relative to the quality of the approximation (Gandibleux *et al.*, 2001; Gomes da Silva *et al.*, 2006, 2004a).

The presence of similar features in the set of efficient solutions of (4), as reported by Balas and Zemel (1980), could pave the way for the development of better approximate and exact algorithms concerning computational time and the quality of the approximation. This is the main focus of research of the present paper.

The rest of the paper is organized as follows: Section 2 presents the concept of bi-criteria core. Section 3 describes the computational experiments on the size of the bi-criteria core. Finally, Section 4 points out the main conclusions and lines for future research.

2 Bi-criteria cores

In multiple criteria problems there is no a single function and the objective functions of the problems can be aggregated into just one in several ways. In the bi-criteria case an aggregation function may be expressed by $p(x,\lambda) = \lambda z_1(x) + (1-\lambda) z_2(x)$ with $0 \leq \lambda \leq 1$. Let $\Im$ be a family of weighted-sum functions $p(x,\lambda)$. We propose the following definition of core given an efficient solution x :

Definition 1 *Given the family of weighted-sum functions, $\Im$, the bi-criteria core of an efficient solution, x, of (4) is the smallest core, when each function of $\Im$ is considered individually.*

Thus, considering the existence of p efficient solutions and q functions, the core associated with an efficient solution x^t, taking into account the function $p(x,\lambda^k)$, is $C^{k,t} = \left\{ j_1^{k,t}, ..., j_2^{k,t} \right\}$, where $j_1^{k,t} = \min\left\{ j : x_j^t = 0, j = 1,...,n \right\}$ and $j_2^{k,t} = \max\left\{ j : x_j^t = 1, j = 1,...,n \right\}$ $\left(\text{if } j_1^t > j_2^t \text{ we assume that } C^t = \emptyset\right)$, where the items are ordered by non-increasing values of the ratio $\dfrac{\lambda^k c_j^1 + \left(1-\lambda^k\right) c_j^2}{w_j}, j = 1,...,n$. The *bi-criteria core* of x^t is $C^{k^*,t} = \underset{k=1,...,q}{\arg\min} \left\{ \left| C^{k,t} \right| \right\}$.

According to Definition 1, determining the bi-criteria core of an efficient solution requires the analyzis of all the functions $p(x,\lambda)$ of $\Im$. In order to obtain the smallest cores it is best to identify the most favourable function for determining the core of a given efficient solution. This means that we must answer the question: what is the value of λ that produces the smallest core?

When determining the core of an efficient solution, the items must be sorted by non-increasing values of the efficiency ratios:

$$e_j(\lambda) = \frac{\lambda c_j^1 + (1-\lambda) c_j^2}{w_j} = \frac{c_j^2}{w_j} + \frac{c_j^1 - c_j^2}{w_j}\lambda, 0 \leq \lambda \leq 1 \tag{5}$$

The efficiency ratios $e_j(\lambda)$ are functions of λ. Due to the dependence of the value of λ, it is said that the efficiency ratios are not well defined. The ratios are, however, bounded from below and above: $\min\left\{ \dfrac{c_j^1}{w_j}, \dfrac{c_j^2}{w_j} \right\} \leq e_j(\lambda) \leq \max\left\{ \dfrac{c_j^1}{w_j}, \dfrac{c_j^2}{w_j} \right\}$.

For a given λ' the items can be ordered such that:

$$e_{l_1}\left(\lambda'\right) \geq e_{l_2}\left(\lambda'\right) \geq .... \geq e_{l_n}\left(\lambda'\right) \tag{6}$$

where, $\{l_1, l_2, ..., l_n\} = \{1, 2, ..., n\}$.

As λ changes within $[0,1]$ the ordering (6) is not stable. Suppose that we are given the order $e_{l_1}\left(\lambda'\right) \geq ... \geq e_{l_j}\left(\lambda'\right) \geq e_{l_{j+1}}\left(\lambda'\right) \geq ... \geq e_{l_n}\left(\lambda'\right)$. It is kept constant for $\lambda^{min} \leq \lambda \leq \lambda^{max}$, where λ^{max} is given by the optimal solution of the following linear problem,

$$
\begin{array}{l}
Max\ \lambda \\
s.t.: \\
e_{l_j}(\lambda) \geq e_{l_{j+1}}(\lambda), j = 1, ..., n-1 \\
\lambda \leq 1, \lambda \geq 0
\end{array} \qquad (7)
$$

Considering the expressions for $e_{l_j}(\lambda), j = 1, ..., n$, the optimal solution of (7) is given by

$$
\lambda^* = \min \left\{ \frac{\dfrac{c^2_{l_{j+1}}}{w_{l_{j+1}}} - \dfrac{c^2_{l_j}}{w_{l_j}}}{\dfrac{c^1_{l_j} - c^2_{l_j}}{w_{l_j}} - \dfrac{c^1_{l_{j+1}} - c^2_{l_{j+1}}}{w_{l_{j+1}}}} : \frac{c^1_{l_j} - c^2_{l_j}}{w_{l_j}} - \frac{c^1_{l_{j+1}} - c^2_{l_{j+1}}}{w_{l_{j+1}}} < 0, j = 1, ..., n-1 \right\} \qquad (8)
$$

For a $\lambda > \lambda^*$ a different ordering is defined. This new ordering can be easily obtained from the previous one simply by swaping the positions of the items pertaining to the set $B(\lambda^*) = \left\{(l_j, l_{j+1}) : e_{l_j}(\lambda^*) = e_{l_{j+1}}(\lambda^*)\right\}$.

Starting from the ordering associated with $\lambda = 0$, and systematically determining the maximum value of λ according to (8), which preserves the different ordering until $\lambda^* \geq 1$, the range $[0, 1]$ is partitioned into sub-ranges, each of them corresponding to a different ordering. Consequently, we have established the following:

Proposition 1 *The number of different orderings (6) is equal to the number of sub-ranges for* $\lambda \in [0, 1]$ *obtained by solving sequences of problem (7) until* $\lambda \geq 1$*, starting from the ordering corresponding to* $\lambda = 0$.

Once the possible orders of the items are identified, the bi-criteria core can finally be computed.

Example 1 *Let us consider the following instance of the bi-criteria {0,1}-knapsack problem.*

$$
\begin{array}{l}
\max z_1(x) = 85x_1 + 31x_2 + 33x_3 + 25x_4 + 28x_5 + 15x_6 + 29x_7 \\
\max z_2(x) = 72x_1 + 17x_2 + 47x_3 + 83x_4 + 49x_5 + 88x_6 + 78x_7 \\
s.t.: \\
98x_1 + 74x_2 + 94x_3 + 91x_4 + 51x_5 + 57x_6 + 57x_7 \leq 261 \\
x_j \in \{0, 1\}, j = 1, ..., 7
\end{array}
$$

Figure 2 gives the efficiency of the items in accordance with (5), and the points where the efficiency is equal. The vertical lines separate the sub-regions where the order of the items changed. The bold line is explained in Example 3. As may be seen in this instance, 14 possible orders like (6) can be defined. The sub-regions correspond to the following ranges of λ :

[0; 0.230379] , [0.230379; 0.415288] ; [0.415288; 0.416667] ; [0.416667; 0.572514] ;
[0.572514; 0.638643] ; [0.638643; 0.671021] ; [0.671021; 0.799320] ; [0.799320; 0.825548] ;
[0.825548; 0.843705] ; [0.843705; 0.894032] ; [0.894032; 0.910138] ; [0.910138; 0.922328] ;
[0.922328; 0.982020] ; [0.982020; 1] .

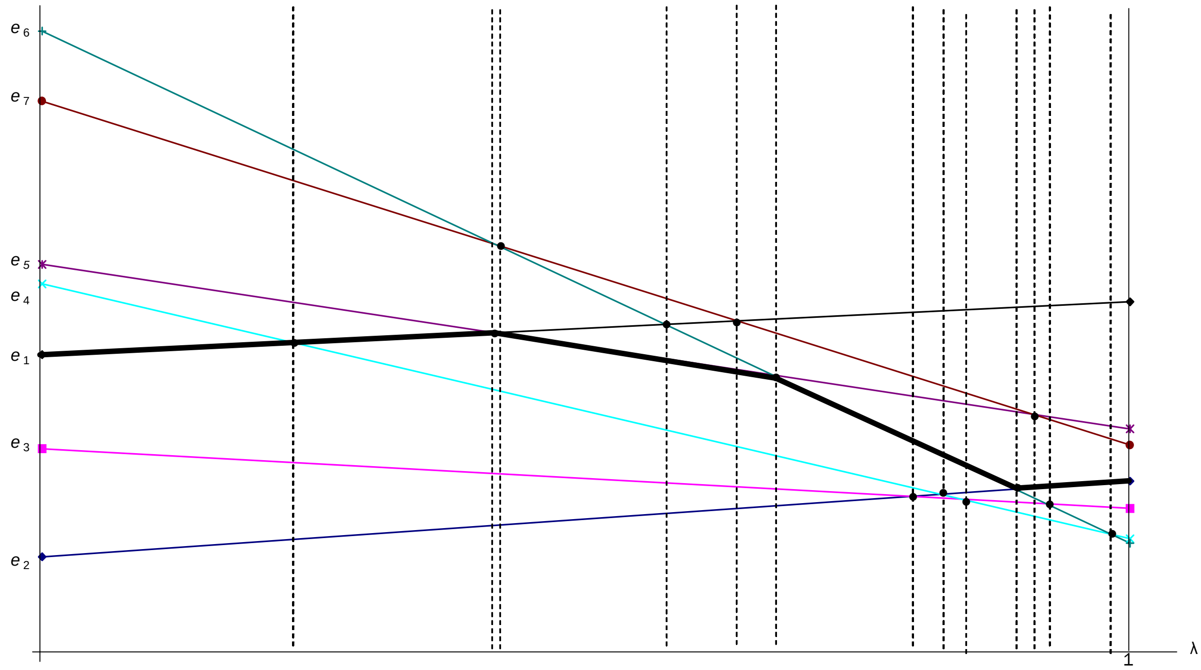

Figure 2: Efficiency functions of the items

When solving the bi-criteria problem by applying an exact method, 7 efficient solutions were found: $x^1 = (0001111)$, $x^2 = (1001001)$, $x^3 = (1010001)$, $x^4 = (0010111)$, $x^5 = (1001010)$, $x^6 = (1000011)$, $x^7 = (1000101)$. *Solutions* x^1, x^2 *and* x^3 *are supported efficient solutions while* x^4, x^5, x^6 *and* x^7 *are non-supported efficient solutions. In the criteria space, the images are* $z^1 = (97, 298)$, $z^2 = (139, 233)$, $z^3 = (147, 197)$, $z^4 = (105, 262)$, $z^5 = (125, 243)$, $z^6 = (129, 238)$ *and* $z^7 = (142, 199)$.

Comparing the composition of each efficient solution with each order, $O_j, j = 1, ..., 14$, *and computing the size of the corresponding core we obtain the results presented in Table 1. For example, with the items ordered according to* O_4*, solution* x^1 *has a core with size 3.*

The bi-criteria cores of efficient solutions, according to Definition 1, are 0, 3, 4, 3, 4, 0 and 0, obtained with λ *belonging to* $[0; 0.230379]$, $[0.572514; 0.8255483]$, $[0.572514; 0.7993203] \cup$

$C^{k,t}$	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}	O_{13}	O_{14}
x^1	0	2	3	3	4	5	5	5	6	7	7	7	7	7
x^2	5	5	5	4	3	3	3	3	4	5	5	6	6	5
x^3	6	6	6	5	4	4	4	5	5	4	4	5	4	4
x^4	3	3	4	4	5	6	6	7	7	6	6	6	6	7
x^5	4	4	4	5	5	4	4	4	5	6	6	6	6	6
x^6	3	2	0	0	0	0	2	2	2	2	3	4	5	6
x^7	5	4	4	3	2	2	0	0	0	0	0	0	0	0

Table 1: Size of the cores for efficient solutions in the different orders of the items

$[0.8437054; 0.9101383] \cup [0.9223284; 1]$, $[0; 0.4152883]$, $[0; 0.416663] \cup [0.6386434; 0.799320]$, $[0.416667; 0.825548]$ *and* $[0.671021; 1]$, *respectively, once they correspond to the smallest possible core.*

If for an efficient solution x^t the ordering (6) corresponds to a sequence of 1's followed by a sequence of 0's this means that x^t can be obtained using the greedy procedure proposed by Dantzig (1957), which leads to the minimum core (the one with cardinality 0). This ordering given by the optimal solution of the following linear program.

$$\begin{aligned}
&Max\ \alpha_1 - \alpha_2 \\
&s.t.: \\
&\alpha_1 \leq \frac{\lambda c_j^1 + (1-\lambda) c_j^2}{w_j}, j \in N_1^t \\
&\alpha_2 \geq \frac{\lambda c_j^1 + (1-\lambda) c_j^2}{w_j}, j \in N_0^t \\
&\lambda \leq 1 \\
&\alpha_1, \alpha_2, \lambda \geq 0
\end{aligned} \tag{9}$$

where $N_1^t = \left\{ j : x_j^t = 1 \right\}$ and $N_0^t = \left\{ j : x_j^t = 0 \right\}$.

Let α_1^*, α_2^*, λ^* be the optimal solution of problem (9), then the following results hold.

Proposition 2 *If* $\alpha_1^* - \alpha_2^* \geq 0$ *then* $x^t = \lfloor \overline{x} \rfloor$ *with* $\overline{x}$ *an optimal solution of* $\max \left\{ p(x, \lambda^*) : \sum_{j=1}^{n} w_j x_j \leq W, x_j \in [0,1], j = 1, ..., n \right\}$, *where* $p(x, \lambda^*) = \lambda^* z_1(x) + (1-\lambda^*) z_2(x)$. *In this case, the cardinality of the corresponding core is 0.*

As Example 1 shows, this can happen either with a supported or non-supported efficient solution.

Corollary 1 *If* $\alpha_1^* - \alpha_2^* < 0$ *then it is not possible to define a function of the type* $p(x, \lambda) = \lambda z_1(x) + (1-\lambda) z_2(x)$ *such that* $x^t = \lfloor \overline{x} \rfloor$. *In this case the core has at least two items.*

Example 2 *Consider the efficient solution* $x^1 = (0001111)$ *from Example 1. The optimal solution for problem (9) is* $\alpha_1^* - \alpha_2^* = 0.17739409$ *and* $\lambda^* = 0$. *Ordering the items according to their efficiency using* $p(x, \lambda^*)$ *the sequence of items 6,7,5,4,1,3,2 is obtained which corresponds to the sequence of variables values* 1111000. *Thus,* x^1 *can be obtained by applying the Dantzig (1957) rule to the function* $p(x, \lambda^*)$, *and rounding down the correspondent solution.*

If $x^2 = (1001001)$ *is considered, the optimal solution for problem (9) is* $\alpha_1^* - \alpha_2^* = -0.2000774$ *and* $\lambda^* = 0.67102149$. *As* $\alpha_1^* - \alpha_2^* < 0$, x^2 *cannot be obtained by rounding down the solution using the Dantzig rule with any function* $p(x, \lambda)$ *since there is no* λ *that can define efficiency ratios for all variables with value 1 greater than those from the variables with value 0. Ordering the items according to their efficiency ratios using* $p(x, \lambda^*)$ *the sequence of items 1,7,5,6,4,3,2 is obtained which corresponds to the following sequence of variables' values:* 1100100.

In bi-criteria {0,1}-knapsack problems there are several Dantzig solutions. The number of such solutions is however bounded from above by the value mentioned in Proposition 1, and the folowing result also holds:

Proposition 3 *The number of Dantzig solutions of a bi-criteria {0,1}-knapsack problem is equal to the number of extreme efficient solutions of the linear relaxation of that problem.*

The Dantzig solutions can be determined by using the bi-criteria simplex method with bounded variables, as explained in Gomes da Silva *et al.* (2003). A step-by-step graphical explanation of this method is given below (Figure 1 illustrates what follows):

Step 1 Consider the graphical representation of the efficiency ratios functions (5) and start with $\lambda = 0$;

Step 2 Apply the Dantzig rule with $p(x, \lambda)$, obtain the Dantzig solution and identify the break item (graphically this solution is kept the same until the point where the efficiency line of the break item intercept another line);

Step 3 Identify the item and the value of λ, which efficiency line intercepts the one of the break item;

Step 4 If $\lambda \geq 1$ stop, all the Dantzig solutions have been determined;

Step 5 The following situations can occur: a) the break item is kept the same, but the identified item is removed from the knapsack; b) the identified item becomes the new break item, and the break item is inserted in the knapsack; c) the identified item becomes the new break item and the previous break item is removed from the knapsack;

Step 6 Return to step 3.

Example 3 *When applying the above procedure to the bi-criteria instance of Example 1, five Dantzig solutions can be seen in Figure 1. They are associated with the intervals of* λ $[0; 0.230379]$, $[0.23037; 0.415288]$, $[0.415288, 0.671021]$, $[0.671021, 0.894032]$, $[0.894032, 1]$, *respectively. In Figure 1, the bold line represents the efficiency of the break item of each Dantzig solution. These break items are 1,1,5,6 and 2, respectively.*

Computing the bi-criteria core requires more than finding the Dantzig solutions since the items can have many other orders which may be associated with a smaller core size. Using only the orders corresponding to the Dantzig solutions, therefore, means that the results for the core size may be overestimated. This is a specificity when more than one criterion is considered.

3 Numerical experiments on the size of the bi-criteria core

This section is devoted to the computational experiments on the size of the bi-criteria core. Five types of instances are considered:

Type 1: $c_j^1, c_j^2, w_j \sim U(1, 100), j = 1, ..., n$ (uncorrelated instances, with small coefficients);

Type 2: $c_j^1, c_j^2, w_j \sim U(1, 10000), j = 1, ..., n$ (uncorrelated instances, with large coefficients);

Type 3: $c_j^1, c_j^2 \sim U(1, 100), w_j = 100$, $j = 1, ..., n$ (uncorrelated criteria functions, with small coefficients and constant weight);

Type 4: $c_j^1, w_j \sim U(1, 100), c_j^2 = w_j + 10, j = 1, ..., n$ (uncorrelated and strongly correlated criterion and weight-sum functions, with small coefficients);

Type 5: $c_j^1, w_j \sim U(1, 100), c_j^2 = 101 - c_j^1$, $j = 1, ..., n$ (uncorrelated criterion and weight functions and strongly correlated criteria functions, with small coefficients).

where, $U(1, a)$ signifies an integer value not greater than a, randomly generated from an uniform distribution.

These instances differ in the way the coefficients are generated and in the range of the coefficients. They are inspired by the types considered by Martello and Toth (1990), Kellerer *et al.* (2004): uncorrelated instances with small and large coefficients and strongly correlated instances.

In all the instances the knapsack capacity remains constant and equal to 50% of the sum of the weights, which generally leads to the highest number of efficient solutions (Visée *et al.*, 1998).

In order to evaluate the size of the bi-criteria core of exact efficient solutions in the bi-criteria {0,1}-knapsack problem, we proceed as follows: 1) generate the entire set of efficient solutions (an implementation of the exact method proposed by Visée *et al.*, 1998 was used); 2) all the possible orders like (2) are generated; 3) the bi-criteria core is computed for each efficient solution.

In the experiments, the number of variables changes depending on the instances types due to the different difficulty of solving them. Type 4 and 5 instances are extremely difficult for the branch-and-bound method by Visée *et al.* (1998). For this reason only small instances are considered.

Tables 3-6 summarize the efficient solutions sets: the average number of extreme efficient solutions ($\overline{T}$), the average percentage of supported solutions (SS) and non-supported solutions (NSS), and the average number of SS and NSS that are equal to the rounded Dantzig solutions (DSS and DNSS). The average number of efficient solutions varies significantly: instances type 4 and 5 have a very low number of efficient solutions and a huge number of efficient solutions, respectively. The performance of types 1-2 instances are in the middle. For types 1-3 instances, the percentage of supported efficient solutions is considerably greater than the percentage of non-supported efficient solutions. This gap increases as the number of items (n) increases. Type 4-5 instances have a balanced number of supported and nonsupported efficient solutions. The average number of Dantzig solutions is very low, but it is slightly greater for supported solutions. In type 3 instances all the supported solutions are Dantzig solutions, because of the mathematical characteristics of these instances (Gomes da Silva *et al.*, 2004b). None of non-supported solutions is a Dantzig one. These instances have the highest number of Dantzig solutions.

			Type of Solutions		Rounded Dantzig solutions	
n	# instances	$\overline{T}$	SS (%)	NSS (%)	DSS	DNSS
100	30	124.9	15.2	84.8	2.8	1.7
300	30	769.5	7.1	92.9	4	2.4
500	10	1754.6	4.9	95.1	4.3	4.1

Table 2: Characterization of efficient solutions: Type 1 instances

			Type of Solutions		Rounded Dantzig solutions	
n	# instances	$\overline{T}$	SS (%)	NSS (%)	DSS	DNSS
100	30	148.7	13.5	86.5	3.2	1.5
300	30	1100	4.8	95.2	3.4	3.6
500	10	2698.1	3.3	96.7	4.2	4.0

Table 3: Characterization of efficient solutions: Type 2 instances

			Type of Solutions		Rounded Dantzig solutions	
n	# instances	$\overline{T}$	SS (%)	NSS (%)	DSS	DNSS
100	30	326.5	10.6	89.4	34.0	0
300	30	2213.2	5.5	94.5	98.2	0
500	10	5894.4	3.3	96.7	189.3	0

Table 4: Characterization of efficient solutions: Type 3 instances

			Type of Solutions		Rounded Dantzig solutions	
n	# instances	$\overline{T}$	SS (%)	NSS (%)	DSS	DNSS
60	30	10.2	59.1	40.9	0.8	0.4
70	30	12.2	58.1	41.9	0.6	0.4
80	30	12	59.2	40.8	0.6	0.7

Table 5: Characterization of efficient solutions: Type 4 instances

			Type of Solutions		Rounded Dantzig solutions	
n	# instances	$\overline{T}$	SS (%)	NSS (%)	DSS	DNSS
40	15	3183.7	49.1	50.9	5.5	1.9
50	15	5102.2	35.3	64.7	6.1	2.1
60	10	16163.6	33.4	66.6	7.3	4.1

Table 6: Characterization of efficient solutions: Type 5 instances

		Type of Solutions											
		Supported				Non-Supported				Overall			
n	T	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range
100	3748	4	7	5.2	0-38	8	11	9.6	0-56	8	11	8.9	0-56
300	23084	2	2.7	2.3	0-32	3.7	5	4.5	0-49.7	3.7	4.7	4.4	0-49.7
500	17546	1.4	2	2.1	0-26.2	2.6	3.6	3.5	0-39.6	2.6	3.6	3.4	0-39.6

Table 7: Bi-criteria core results: Type 1 instances

		Type of Solutions											
		Supported				Non-Supported				Overall			
n	T	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range
100	4460	4	6	5.3	0-42	9	12	10.1	0-65	8	11	9.5	0-65
300	33001	2	3	2.5	0-27.3	4	5.3	4.8	0-59.7	4	5	4.7	0-59.7
500	26981	1.4	2	1.7	0-28.8	2.8	3.6	3.6	0-64.4	2.6	3.6	3.5	0-64.4

Table 8: Bi-criteria core results: Type 2 instances

The results for the size of the bi-criteria core by instance type are presented in Tables 7-11. The tables give the total number of efficient solutions (T), the percentage size of the core for the supported solutions, non-supported solutions and for the entire set of efficient solutions. Columns $\frac{1}{2}T$, $\frac{3}{4}T$, $\overline{C}$ and Range, mean the maximum percentage core of 50% of the total efficient solutions, the maximum percentage core of 75% of the total efficient solutions, the average percentage core, and the range of the percentage core, respectively.

The findings show that, on average, the bi-criteria core is a very small percentage of the total number of items, with type 5 instances being an exception to this issue. Observing columns $\frac{1}{2}T$ and $\frac{3}{4}T$ it can be said that 50% and 75% of solutions can be found by exploring small neighborhoods around the break items of weighted-sum functions. Supported solutions are easier to find in this exploration, as revealed by the smaller bi-criteria cores. This feature is observed in all instances types. It is interesting to note that the average size of the bi-criteria core falls in terms of relative size as the problem size increases. These results are very similar to those obtained in single criterion problems. Inversely correlated instances, concerning criteria functions (type 5 instances), are associated with the highest core size, while solutions for type 3 instances have the smallest bi-criteria core size. These are the most favourable instances for the application of the core concept.

An observation should be made regarding the range of bi-criteria core sizes: the maximum observed size is considerably greater than the average size. However, a detailed analyzis reveals

		Type of Solutions											
		Supported				Non-Supported				Overall			
n	T	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range
100	9794	0	0	0	0-0	5	6	5.1	0-16	5	6	4.6	0-16
300	66395	0	0	0	0-0	2	2.7	2.1	0-7	3	2.7	2.0	0-7
500	58944	0	0	0	0-0	1.4	1.8	1.5	0-4.2	1.4	1.5	1.4	0-4.2

Table 9: Bi-criteria core results: Type 3 instances

		Type of Solutions											
		Supported				Non-Supported				Overall			
n	T	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range
60	299	8.3	11.2	8.1	0-35	11.2	15	11.5	0-30	10	13.3	9.5	0-35
70	366	7.1	10	7.3	0-21.4	10	14.3	10	0-30	8.6	11.4	8.5	0-30
80	371	7.5	8.8	6.7	0-16.3	7.5	11.3	8.1	0-26.3	7.5	8.8	7.3	0-26.3

Table 10: Bi-criteria core results: Type 4 instances

		Type of Solutions											
		Supported				Non-Supported				Overall			
n	T	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range	$\frac{1}{2}T$	$\frac{3}{4}T$	$\overline{C}$	Range
40	47755	30	35	30.2	0-67.5	32.5	37.5	31.5	0-75	30	35	30.6	0-75
50	76533	28	32	23.3	0-54	28	32	28	0-76	26	30	26.4	0-76
60	161636	21.7	26.7	22.7	0-61.7	26.7	31.7	26.9	0-73.3	25	30	25.8	0-73.3

Table 11: Bi-criteria core results: Type 5 instances

that, despite the fact that the size of the bi-criteria core is large, the number of variables belonging to it which assume a value different from the corresponding continuous solution may be very small. To illustrate these results we have taken into account the instances with the highest core from type 1 instances. In Table 12 the size of the core, $|C|$, is presented as well as the number of variables with a different value from the continuous solution, $|C^-|$, and the percentage of items changed with respect to the size of the bi-criteria core.

The sharpest conclusion from the above experiments concerns the consequences of the "compactness" of the bi-criteria core size which is a most promising path for the development of an effective exact or approximate method for solving the {0,1}-knapsack problem more efficiently. The "compactness" reveals the existence of a privileged region in the decision space for a prioritary search for efficient solutions. The percentages found mean that a significant number of efficient solutions are found in those small regions.

Figures 3-6 show the distribution of the percentage size of the bi-criteria core corresponding to supported, non-supported and to all the efficient solutions of instances types 1, 3, 4 and 5, with the highest number of items. The horizontal axes of the figures have the same scale in order to make comparision of the distributions easier. As can be seen, the distributions are biased, and very compact, especially for supported efficient solutions. The distribution pattern is similar both for supported and non-supported solutions.

n	$\|C\|$	$\|C^-\|$	$\|C^-\| / \|C\| \times 100$
100	56	2	3.57%
300	149	40	26.85%
500	198	7	3.54%

Table 12: Items changed in the bi-criteria core

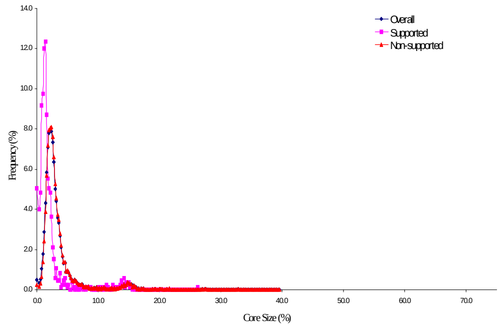

Figure 3: Type 1 instances

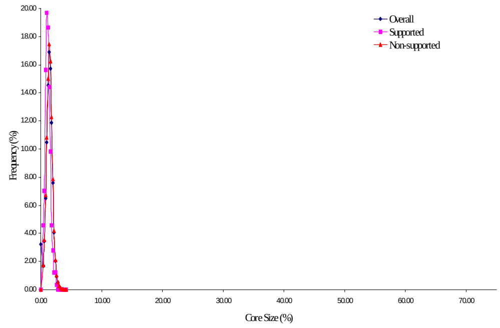

Figure 4: Type 3 instances

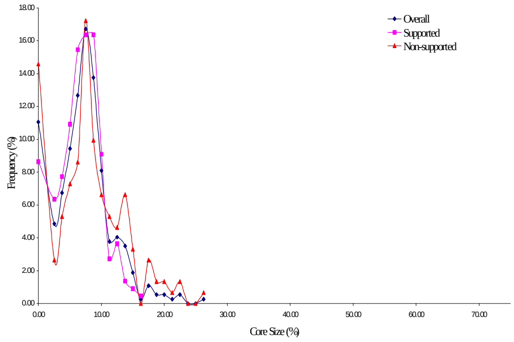

Figure 5: Type 4 instances

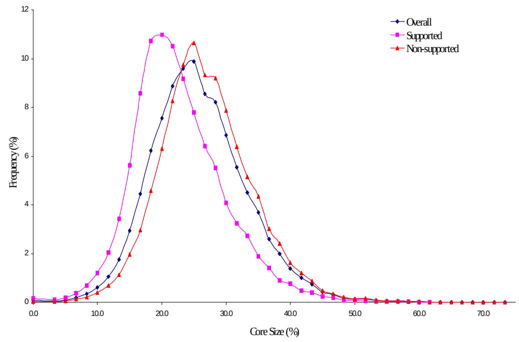

Figure 6: Type 5 instances

4 Conclusions

In this paper, the concept of core was extended to the bi-criteria {0,1}-knapsack domain. This extension was, however, not trivial. The computational experiments conducted with the five types of instances revealed that the characteristics found in the single criterion case were also found in the bi-criteria instances, *i.e.*, small sized cores, with a small increase, according to the dimension of the problem. This is due to the hidden similarities when solving problems (1) and (4). It was also noticed that for the worst cases of bi-criteria core size, very few variables of the continuous solution were changed. Supported on these results, the construction of an exact or approximate methods based on the core itself is a promising line of research for solving efficiently the bi-criteria {0,1}-knapsack problem.

Acknowledgements: The authors would like to acknowledge support from the MONET research project (POCTI/GES/37707/2001).

References

[1] Balas, E., Zemel, E. (1980) "An algorithm for large zero-one knapsack problems", *Operational Research*, 28, pp. 1130-1154

[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] Dantzig, G. (1957) "Discrete variable extremum problems", *Operations Research*, 5, pp. 226-277

[4] Fayard, D., Plateau, G. (1982) "An algorithm for the solution of the 0-1 knapsack problem", *Computing*, 28, pp. 269-287

[5] 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, L., C.A. Coello Coello, (Eds), *Proceedings of the First International Conference on Evolutionary Multi-criterion Optimization*, Lecture Notes in Computer Science 1993, Springer, Berlin, pp. 429-442

[6] Gomes da Silva, C., Figueira, J., Clímaco, J. (2003) "An interactive procedure for the bi-criteria knapsack problems", *Research Report*, N^{o}4, INESC-Coimbra, Portugal (in Portuguese) http://www.inescc.pt/download/RR2003_04.pdf

[7] Gomes da Silva, C., Figueira, J., Clímaco, J. (2004a) "Integrating partial optimization with scatter search for solving the bi-criteria {0, 1}-knapsack problem", *European Journal of Operational Research* (accepted for publication, 2004)

[8] Gomes da Silva, C., Clímaco, J., Figueira, J. (2004b) "Geometrical configuration of Pareto frontier of bi-criteria {0,1}-knapsack", *Research Report*, N^{o}8, INESC-Coimbra, Portugal

[9] Gomes da Silva, C., Clímaco, J., Figueira, J. (2006) "A scatter search method for the bi-criteria knapsack problems", *European Journal of Operational Research*, Vol. 169, Issue 1, pp. 73-391

[10] Kellerer, H., Pferchy, C., Pisinger, D. (2004) *Knapsack problems*, Springer, Berlin

[11] Martello, S., Toth, P. (1990) "A new algorithm for the 0-1 knapsack Problems", *Management Science*, 34, pp. 633-645

[12] Martello, S., Pisinger, D., Toth, P. (1999) "Dynamic programming and strong bounds for the {0,1}-knapsack problems", *Management Science*, 45, pp. 414-424

[13] Martello, S., Toth, P. (1990) *Knapsack Problems - Algorithms and Computer Implementations*, John Wiley & Sons, New York

[14] Pisinger, D. (1995) "An expanding-core algorithm for the exact 0-1 knapsack problem", *European Journal of Operational Research*, 87, pp. 175-187

[15] Pisinger, D. (1997) "A minimal algorithm for the {0,1}-knapsack problem", *Operations Research*, 45, pp. 758-767

[16] Pisinger, D., Toth, P. (1998) “Knapsack problems” *in* Du, D-Z., Pardalos, P. (Eds.), Vol. 1, *Handbook of Combinatorial Optimization*, Kluwer Academic Publishers, Dordrecht.

[17] Pisinger, D. (1999) “Core Problems in knapsack algorithms”, *Operations Research*, 47 (4), pp. 570-575

[18] 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

[19] Steuer, R. (1986) *Multiple Criteria Optimization, Theory, Computation and Application*, John Wiley & Sons, New York

[20] Ulungu, E.L., Teghem, J.(1997) “Solving multi-objective knapsack problem by branch-and-bound procdure” *in* J. Climaco (Ed.), *Multicriteria Analysis*, Springer Verlag, Berlin, pp. 269-278