

On the calculation of stability radius for multi-objective combinatorial optimization problems by inverse optimization

Julien Roland · Yves De Smet · José Rui Figueira

Received: 7 February 2012 / Revised: 24 September 2012 / Published online: 12 October 2012
© Springer-Verlag Berlin Heidelberg 2012

Abstract This paper deals with stability analysis in multi-objective combinatorial optimization problems. The stability radius of an efficient solution is defined as the maximal adjustment of the problem parameters such that this solution remains efficient. An algorithm based on inverse optimization is proposed to compute it. The adjustment is limited to the coefficients of the objective functions and measured by the Chebyshev norm. This approach is applied to randomly generated instances of the bi-objective knapsack problem and computational results are reported. Several illustrative examples are analyzed.

Keywords Multi-objective optimization · Combinatorial optimization · Stability analysis · Inverse optimization

Mathematics Subject Classification (2000) 90C27 · 90C29 · 90C31

J. Roland (✉) · Y. De Smet
CoDE-SMG, Service de Mathématiques de la Gestion, Ecole Polytechnique de Bruxelles,
Université Libre de Bruxelles, Boulevard du Triomphe CP 210-01, 1050 Brussels, Belgium
e-mail: julien.roland@ulb.ac.be

Y. De Smet
e-mail: yves.de.smet@ulb.ac.be

J. R. Figueira
CEG-IST, Instituto Superior Técnico, Technical University of Lisbon, Av. Rovisco Pais,
1049-001 Lisboa, Portugal

J. R. Figueira
LORIA Laboratory, Nancy, France
e-mail: figueira@ist.utl.pt

1 Introduction

Multi-objective optimization consists in finding solutions that simultaneously “maximize” several objective functions over a set of feasible solutions. Since there is generally no ideal feasible solution that is simultaneously optimal for all objective functions, the resolution of such a problem usually leads to finding the so-called efficient solutions. An efficient solution is characterized by the fact that it is not possible to find another solution that leads to an improvement of the outcomes of all objective functions, without a degradation of the outcome of at least one objective function (see [Steuer 1986](#); [Ehrgott 2005](#)).

Modeling a multi-objective optimization problem requires to fix the values of the parameters in order to define the objective functions and the feasible set. Those values rely on some hypothesis as well as on the accuracy of the evaluations. These sources of uncertainty must be taken into account, because a small perturbation on the model can transform an efficient solution into a non-efficient one. This reflects an instability of the model. A way of assessing such an instability is to compute a stability radius for each efficient solution. This stability radius is defined as the maximal variation of the problem parameters that allows the solution to remain an efficient one (see [Emelichev et al. 2004](#); [Emelichev and Kuzmin 2006](#); [Emelichev and Podkopaev 2010](#)).

The purpose of this paper is to study the calculation of the stability radius in the context of multi-objective combinatorial optimization. In single-objective combinatorial optimization, the stability radius can be computed in polynomial time if the problem is polynomially solvable ([Chakravartia and Wagelmansb 1998](#)). However, to the best of our knowledge, no algorithm can compute the stability radius for multi-objective combinatorial problems except by explicit enumeration ([Emelichev and Podkopaev 2010](#)). Theoretical results are available for measuring the stability radius, but the proposed formula requires the complete enumeration of subsets of the feasible set ([Emelichev et al. 2004](#); [Emelichev and Kuzmin 2006](#); [Emelichev and Podkopaev 2010](#)).

The calculation of the stability radius is closely related to inverse multi-objective optimization ([Roland et al. 2011](#)). It is defined as the minimal adjustment of the problem parameters inducing a change (addition or removal of solutions) in the efficient set. Indeed, it is easy to see that the stability radius of an efficient solution can be obtained through the minimal adjustment of the parameters, in such a way that a given solution becomes non-efficient. However, this precise question has not been covered yet in inverse multi-objective optimization, and it constitutes the purpose of the paper.

This paper deals with a particular class of multi-objective combinatorial problems, where each objective function is a maximum sum and the coefficients are restricted to natural numbers. In a first attempt to solve the associated inverse problem, only the objective functions coefficients are subject to an adjustment, which is measured by the Chebyshev distance. Some theoretical results on the nature of the optimal solutions are provided, and an algorithm is proposed. This algorithm performs a particular search on the set of profit matrices (objective functions coefficients) in order to find the minimal adjustment, which transforms a given efficient solution into a non-efficient one. To achieve this purpose, the algorithm makes use of a linear integer program in order to test the efficiency of the feasible solution for a given profit matrix. This algorithm has been implemented and computational experiments were performed on

randomly generated instances of the bi-objective knapsack problem. Finally, several illustrative examples are analyzed in order to have a better understanding of the stability radius.

This paper is organized as follows. In Sect. 2, concepts, definitions, and notation are introduced. In Sect. 3, the inverse problem is defined, theoretical results are provided, and an algorithm is proposed. The design of the experiments and computational results are presented in Sect. 4. Several illustrative examples are analyzed in Sect. 5. We conclude with remarks and directions for future research.

2 Concepts, definitions, and notation

Multi-objective optimization consists of maximizing “simultaneously” several objective functions over a set of feasible solutions. The *feasible set* is denoted by $X \subseteq \mathbb{R}^n$. The outcome of each feasible solution $x \in X$ is denoted by a vector $F(x) = (f^1(x), f^2(x), \dots, f^i(x), \dots, f^q(x))$ composed by the outcomes of q objective functions $f^i : X \rightarrow \mathbb{R}$, with $i \in I = \{1, 2, \dots, q\}$.

A particular class of Multi-Objective Combinatorial Optimization problems (MOCO) is considered. Each instance is defined by a pair (X, C) where $X \subseteq \{x : x \in \{0, 1\}^n\}$ is the feasible set and $C \in \mathbb{N}^{q \times n}$ is the so-called *profit matrix* (composed of non-negative integers). Each objective function $f^i : X \rightarrow \mathbb{N}$, with $i \in I$, is defined by a row of the profit matrix with $f^i(x) = \sum_{j \in J} C_{ij}x_j$, where $J = \{1, 2, \dots, n\}$.

Let $x, y \in \mathbb{R}^n$ be two vectors. The following notation will be used hereafter: $x > y$ iff $\forall j \in J : x_j > y_j$; $x \geq y$ iff $\forall j \in J : x_j \geq y_j$; $x \neq y$ iff $\exists j \in J : x_j \neq y_j$; $x \geq y$ iff $x \geq y$ and $x \neq y$. The binary relations \leq, \leq , and $<$ are defined in a similar way.

In multi-objective optimization, two spaces should be distinguished. The *decision space*, i.e. the space in which the feasible solutions are defined, and the *objective space*, i.e. the space in which the *outcome vectors* are defined. The image of the feasible set in the objective space is denoted by $Y = \{y \in \mathbb{R}^q : y = Cx, x \in X\}$.

An outcome vector $y^* \in Y$ is said to be *ideal* if and only if, for all $y \in Y, y^* \geq y$. An ideal outcome vector does not always exist and there is not a unique total order on \mathbb{R}^q , as soon as $q \geq 2$. Consequently, it is widely accepted to build the dominance relation on the set Y of the outcome vectors. Let $y, y' \in Y$ denote two outcome vectors such that $y \neq y'$. If $y \geq y'$, then y dominates y' . *Dominance* is a binary relation that is irreflexive, asymmetric, and transitive. This relation induces a partition of Y into two subsets: the set of dominated outcome vectors and the set of *non-dominated* outcome vectors. The set of non-dominated outcomes corresponding to an instance (X, C) is denoted by $ND(X, C)$. Similarly, in the decision space the concepts of efficient and non-efficient solutions can be defined. A solution $x^* \in X$ is *efficient* if and only if there is no $x \in X$ such that $F(x) \geq F(x^*)$. The set of efficient solutions corresponding to an instance (X, C) is denoted by $E(X, C)$.

Let us consider the L_∞ distance (Chebyshev distance) between two matrices C and D , i.e. $\|C - D\|_\infty = \max_{i \in I, j \in J} |C_{ij} - D_{ij}|$. The set of all matrices D with distance $\epsilon \in \mathbb{N}$ from C is defined by $\Gamma(\epsilon) = \{D \in \mathbb{N}^{q \times n} : \|C - D\|_\infty = \epsilon\}$. The *stability radius* of an efficient solution $x \in E(X, C)$ is the optimal solution of $\max\{\epsilon \in \mathbb{N} : \forall D \in \Gamma(\epsilon), x \in E(X, D)\}$.

3 Theoretical developments

The stability radius of an efficient solution can be computed by solving a particular inverse multi-objective optimization problem. The considered inverse problem consists of finding a minimal adjustment of the profit matrix such that a given efficient solution becomes non-efficient. This question is formalized as follows. Let (X, C) denote an instance of a MOCO and $x^0 \in X$ an efficient solution. The L_∞ inverse efficiency multi-objective combinatorial optimization problem (IEMOCO- ∞) can be stated as follows:

$$\begin{aligned} & \min \|C - D\|_\infty \\ \text{subject to: } & x^0 \notin E(X, D) \\ & D \in \mathbb{N}^{q \times n}. \end{aligned} \tag{IEMOCO- ∞ }$$

It is easy to understand that the optimal solution value of IEMOCO- ∞ is equal to the stability radius increased by one unit. This problem is not feasible whenever every solution $x \in X \setminus \{x^0\}$ is a subset of x^0 , i.e. when there does not exist a $j \in J$ such that $x_j = 1$ and $x_j^0 = 0$. This situation occurs in the following instance:

$$\begin{aligned} & \max f^1(x) = 5x_1 + 5x_2 \\ & \max f^2(x) = 5x_1 + 5x_2 \\ \text{subject to: } & x_1 + x_2 \leq 2 \\ & x_1, x_2 \in \{0, 1\}, \end{aligned} \tag{1}$$

where $\{v^1, v^2, v^3, v^4\} = \{(1, 1); (1, 0); (0, 1); (0, 0)\}$ is the feasible set and $\{v^1\}$ is the efficient set. The feasible solutions v^2, v^3 and v^4 are subsets of v^1 , which implies that for all profit matrices $D \in \mathbb{N}^{q \times n} : Dv^1 \geq Dv^2, Dv^1 \geq Dv^3, Dv^1 \geq Dv^4$.

Let us analyze the nature of some optimal solutions of IEMOCO- ∞ . Based on a partition of J defined by $J^0 = \{j \in J : x_j^0 = 0\}$ and $J^1 = \{j \in J : x_j^0 = 1\}$, the first theorem establishes that an optimal solution D^* of IEMOCO- ∞ can be built by decreasing, or keeping equal, C_{ij} , for all $j \in J^1$ and by increasing, or keeping equal, C_{ij} , for all $j \in J^0$, for all $i \in I$.

Theorem 1 *For every feasible instance of IEMOCO- ∞ with profit matrix C , there exists an optimal solution $D^* \in \mathbb{N}^{q \times n}$ of IEMOCO- ∞ such that $\forall j \in J^1 : D_{ij}^* \leq C_{ij}$ and $\forall j \in J^0 : D_{ij}^* \geq C_{ij}$, with $i \in I$.*

Proof Let $D \in \mathbb{N}^{q \times n}$ denote any optimal solution of IEMOCO- ∞ . Define the following sets for all $i \in I : J_i^{0<} = \{j \in J^0 : D_{ij} < C_{ij}\}, J_i^{1>} = \{j \in J^1 : D_{ij} > C_{ij}\}$. By definition of D , there is a feasible solution $x \in X$ with $Dx \geq Dx^0$. Consider a matrix $D^* \in \mathbb{N}^{q \times n}$ defined as follows, for all $i \in I, j \in J$:

$$D_{ij}^* := \begin{cases} C_{ij}, & \text{if } j \in \{J_i^{1>} \cup J_i^{0<}\} \\ D_{ij}, & \text{otherwise.} \end{cases} \tag{2}$$

Let us show that if $Dx \geq Dx^0$, then $D^*x \geq D^*x^0$. This is equivalent to show that the following condition holds: if $D(x - x^0) \geq 0$, then $D^*(x - x^0) \geq 0$. The profit matrix D^* is introduced into the first inequality as follows, $(D + D^* - D^*)(x - x^0) \geq 0$, which leads to write: $D^*(x - x^0) \geq (D^* - D)(x - x^0)$. From Eq. 2 and the definition of J^1 , one may deduce that, for all $j \in J^1$, $(x - x^0)_j \leq 0$ and $(D^*_{ij} - D_{ij}) \leq 0$. Similarly, for all $j \in J^0$, $(x - x^0)_j \geq 0$ and $(D^*_{ij} - D_{ij}) \geq 0$. Therefore, $(D^* - D)(x - x^0) \geq 0$ and consequently $D^*(x - x^0) \geq 0$. This implies that D^* is a feasible solution of IEMOCO- ∞ and therefore an optimal one, because the inequality $\|D^* - C\|_\infty \leq \|D - C\|_\infty$ is directly deduced from Eq. 2. \square

Based on the same principle of increasing and decreasing some specific parts of the profit matrices, let us define a particular operator \ominus between pairs of matrices with respect to x^0 .

Definition 1 ($D \ominus E$) Let D and E be two matrices of size $q \times n$. For all $i \in I$ and $j \in J$,

$$(D \ominus E)_{ij} := \begin{cases} \max\{0, D_{ij} - E_{ij}\}, & \text{if } j \in J^1, \\ D_{ij} + E_{ij}, & \text{otherwise.} \end{cases}$$

This particular operation is crucial, because when it is applied on a feasible solution of IEMOCO- ∞ , the resulting solution is also feasible. This is established in the following theorem.

Theorem 2 For all $E \in \mathbb{N}^{q \times n}$, if D is a feasible solution of IEMOCO- ∞ , then $D \ominus E$ is also feasible.

Proof Let us show that if $Dx \geq Dx^0$, then $(D \ominus E)x \geq (D \ominus E)x^0$. Similarly to the proof of Theorem 1, it is equivalent to show that the following condition holds: if $D(x - x^0) \geq 0$, then $(D \ominus E)(x - x^0) \geq 0$. Consequently, one can write: $(D \ominus E)(x - x^0) \geq [(D \ominus E) - D](x - x^0) \geq 0$.

From Definition 1, one may deduce that, for all $j \in J^1$, $(x - x^0)_j \leq 0$ and $[(D \ominus E) - D] \leq 0$. Similarly, for all $j \in J^0$, $(x - x^0)_j \geq 0$ and $[(D \ominus E) - D] \geq 0$. Therefore, $[(D \ominus E) - D](x - x^0) \geq 0$ and consequently $(D \ominus E)(x - x^0) \geq 0$. This implies the proof of this theorem. \square

Let us define a matrix $D^k \in \mathbb{N}^{q \times n}$ of distance at most k from matrix C with respect to the L_∞ distance.

Definition 2 [D^k] Let $k \geq 0$ be a natural number. Then, D^k is a matrix of size $q \times n$, where for all $i \in I$ and $j \in J$,

$$D^k_{ij} := \begin{cases} \max\{0, C_{ij} - k\}, & \text{if } j \in J^1, \\ C_{ij} + k, & \text{otherwise.} \end{cases}$$

The following theorem provides an optimality condition for D^k based on the value of k .

Theorem 3 If there exists an optimal solution D of IEMOCO- ∞ with $\|C - D\|_\infty = k$, then D^k is also an optimal solution of IEMOCO- ∞ .

Proof From Theorem 1, it can be assumed that $\forall i \in I, \forall j \in J^1 : D_{ij} \leq C_{ij}$ and $\forall j \in J^0 : D_{ij} \geq C_{ij}$. It is easy to build a matrix $E \in \mathbb{N}^{q \times n}$ such that $D \ominus E = D^k$. From Theorem 2, D^k is a feasible solution of IEMOCO- ∞ and therefore the theorem is proved. \square

As a corollary of this theorem, an optimal solution of IEMOCO- ∞ can be built on the basis of the optimal solution value. Indeed, if the optimal solution value of IEMOCO- ∞ is equal to k , then an optimal solution of this problem is given by the matrix D^k . Therefore, IEMOCO- ∞ can be reduced to finding the optimal solution value, which is given by the minimal value of k where x^0 is a non-efficient solution with respect to D^k . In order to reduce the search domain, an upper bound on this value is provided in the following lemma.

Lemma 1 *If $\delta_\infty \in \mathbb{N}$ is the optimal solution value of IEMOCO- ∞ , then $\delta_\infty \leq \Delta = \max_{i \in I, j \in J} \{x_j^0 C_{ij}\}$*

Proof For a feasible instance of IEMOCO- ∞ , it is always possible to build a matrix $D \in \mathbb{N}^{q \times n}$ such that $\|C - D\|_\infty = \max_{i \in I, j \in J} \{x_j^0 C_{ij}\}$ and Dx^0 is a dominated solution of (X, D) . The matrix is defined as follows, $\forall i \in I, \forall j \in J^0 : D_{ij} = C_{ij}$ and $\forall i \in I, \forall j \in J^1 : D_{ij} = 0$. It is easy to see that there exists another feasible solution $x \in X$ such that $Dx \geq Dx^0$ and $\|C - D\|_\infty = \max_{i \in I, j \in J} \{x_j^0 C_{ij}\}$. This solution satisfies the condition that there exists a $j \in J$, such that $x_j = 1$ and $x_j^0 = 0$; otherwise the problem is not feasible. This concludes the proof. \square

Based on the previous results, an algorithm for computing an optimal solution of IEMOCO- ∞ is devised. Thanks to Theorem 3, an optimal solution of IEMOCO- ∞ can be built based on the distance between matrices D^* and C . Since this distance is bounded from above by $\Delta = \max_{i \in I, j \in J} \{x_j^0 C_{ij}\}$ (see Lemma 1), the algorithm consists of finding the minimal value $k \in \{1, 2, \dots, \Delta\}$ such that $D^k x^0$ is a dominated vector of the multi-objective instance (X, D^k) . This condition is naturally checked by solving Problem 3.

$$\begin{aligned} & \max \sum_{i \in I} \sum_{j \in J} D_{ij}^k x_j \\ & \text{subject to: } D^k x \geq D^k x^0 \\ & \quad x \in X. \end{aligned} \tag{3}$$

Let x^* be an optimal solution of Problem 3. Wendell and Lee (1977) have shown that $D^k x^0$ is a dominated solution of (X, D^k) if and only if $D^k x^0 \neq D^k x^*$. This problem can be solved as a linear integer program if the feasible set X is defined by a set of linear constraints.

The minimal value of k can be found by performing a binary search on the set $\{1, 2, \dots, \Delta\}$, because the value of k can be increased without altering the non-efficiency of x^0 with respect to matrix D^k , as it can be directly deduced from Theorem 2. Therefore, Algorithm 1 requires to solve $\log_2 \Delta$ times Problem 3. For more details on the procedure, see the pseudo-code of Algorithm 1.

Algorithm 1 Compute an optimal solution D^* of IEMOCO- ∞ .

```

1:  $a \leftarrow 0$ ;
2:  $b \leftarrow \Delta$ ;
3: while ( $a \neq b$ ) do
4:    $k \leftarrow a + \lfloor (b - a)/2 \rfloor$ ;
5:   Build matrix  $D^k$ ;
6:    $x^* \leftarrow$  Solve Problem 3;
7:   if ( $D^k x^0 \neq D^k x^*$ ) then
8:      $b \leftarrow k$ ;
9:   else
10:     $a \leftarrow k + 1$ ;
11:   end if
12: end while
13:  $D^* \leftarrow D^a$ ;

```

4 Computational experiments

The purpose of this section is to report the performance of Algorithm 1 (in terms of CPU time) on several sets of randomly generated instances of the bi-objective $\{0,1\}$ -knapsack problem (BKP). This is a well-known classical multi-objective combinatorial optimization problem (Kellerer et al. 1994), which is formalized as follows:

$$\begin{aligned}
 & \max F(x) = (f^1(x), f^2(x)) \\
 \text{subject to: } & \sum_{j \in J} w_j x_j \leq W \\
 & x_j \in \{0, 1\}, \quad j \in J.
 \end{aligned}
 \tag{BKP}$$

The design of the experiments is inspired by the frameworks used in Martello and Toth (1990) and in Pisinger (1995). For a given number of variables n and data range R , a set of instances was randomly generated in the following way. Each instance $s \in \{1, 2, \dots, S\}$ is generated with the seed number s . The values of C_{1j} , C_{2j} and w_j are uniformly distributed within the range $[1, R]$, and W is computed as the maximum between R and $\lfloor P \sum_{j \in J} w_j \rfloor$, where $P \in [0, 1]$. Groups were composed of $S = 30$ instances. The choice of 30 is based on the rule of thumb in statistics to produce good estimates (Coffin and Saltzman 2000). Other types of randomly generated instances, for which the profit matrices are correlated with the weights, were also considered. Since these instances have led to the same kind of results, we will not detail them hereafter.

For each set of instances, Algorithm 1 was run on each efficient solution. The performance was measured through the average (Avg.), standard deviation (SD), minimum (Min.) and maximum (Max.) of the CPU time in seconds. Algorithm 1 was implemented in the C# programming language and Problem 3 was solved by using the CPLEX solver. All the experiments were performed on a 3.0GHz dual-core processor with 4 GB of RAM.

Results show that Algorithm 1 is very efficient for large scale instances. For example, with $n = 500$, $R = 1,000$ and $P = 0.5$, the average CPU time is 4.91 s with a standard deviation of 2.14. Let us point out that the performances of this algorithm are strongly linked to Problem 3. Therefore, the use of another way to test the efficiency of a feasible solution could either increase or decrease significantly the CPU

Table 1 Impact of varying the number of variables n and data range R on the performance of Algorithm 1 with a group of instances with $P = 0.5$

n	CPU time (s)							
	R = 100				R = 1,000			
	Avg.	SD	Min.	Max.	Avg.	SD	Min.	Max.
10	0.41	0.10	0.30	0.74	0.58	0.09	0.44	0.81
50	0.51	0.10	0.42	0.96	0.77	0.13	0.62	1.30
100	0.68	0.17	0.51	1.14	1.13	0.29	0.80	2.25
500	1.31	0.81	0.28	3.58	4.91	2.14	0.96	8.32

time. For more details about the results of the computational experiments, the reader may consult Table 1.

5 Illustrative examples

The purpose of this section is to present and analyze the stability radius for several illustrative instances of the bi-objective $\{0,1\}$ -knapsack problem. An extended version of the stability radius is considered, where we take into account stable and unstable components in the profit matrix, i.e. some values in the profit matrix may have a different value while others are fixed. This helps to get a better understanding of the stability radius. Let us note that one could even consider intervals on the profits to express more precisely the uncertainty on their adequate values. This would enable a more sophisticated analysis for a real-world application.

The set of stable components in the profit matrix is denoted by $S \subseteq I \times J$. The set of unstable components is denoted by \bar{S} , with $\bar{S} \cup S = I \times J$. This leads to modify IEMOCO- ∞ as follows:

$$\begin{aligned}
 & \min \|C - D\|_\infty \\
 \text{subject to: } & x^0 \notin E(X, D) \\
 & D \in \mathbb{N}^{q \times n} \\
 & C_{ij} = D_{ij}, \text{ for all } (i, j) \in S.
 \end{aligned}
 \tag{IEMOCO- ∞ (S)}$$

Theorems 1,2, and 3 can be extended to IEMOCO- ∞ (S). It requires to modify the definition of D^k as follows, for all $i \in I, j \in J$,

$$D^k_{ij} := \begin{cases} \max\{0, C_{ij} - k\}, & \text{if } j \in J^1 \text{ and } (i, j) \in \bar{S}, \\ C_{ij} + k, & \text{if } j \in J^0 \text{ and } (i, j) \in \bar{S}, \\ C_{ij}, & \text{otherwise,} \end{cases}$$

as well as the definition of $D \ominus E$, for all $i \in I, j \in J$,

$$(D \ominus E)_{ij} := \begin{cases} \max\{0, D_{ij} - E_{ij}\}, & \text{if } j \in J^1 \text{ and } (i, j) \in \bar{S}, \\ D_{ij} + E_{ij}, & \text{if } j \in J^0 \text{ and } (i, j) \in \bar{S}, \\ D_{ij}, & \text{otherwise.} \end{cases}$$

It is obvious to show that Theorems 1,2, and 3 remain valid with these modifications. However, Lemma 1 cannot be applied in this situation as illustrated in the following instance:

$$\begin{aligned}
 \max f^1(x) &= 1x_1 + 2x_2 \\
 \max f^2(x) &= 4x_1 + 1x_2 \\
 \text{subject to: } &x_1 + x_2 \leq 1 \\
 &x_1, x_2 \in \{0, 1\},
 \end{aligned}
 \tag{4}$$

where $\{v^1, v^2, v^3\} = \{(1, 0); (0, 1); (0, 0)\}$ is the feasible set, $\{v^1, v^2\}$ is the efficient set, and $S = \{(1, 1); (1, 2); (2, 1)\}$ is the set of stable components. The minimal adjustment of the profit matrix that transforms v^2 into a non-efficient solution is equal to 3. This provides a counterexample to Lemma 1. However, in practice, one would consider the optimal value of $\max_{i \in I, j \in J} \{C_{ij}\}$ as being large enough to represent a maximal stability radius, because this modification would completely change the initial profit matrix.

Consider, as a first illustrative example, the following knapsack instance:

$$\begin{aligned}
 \max f^1(x) &= 10x_1 + x_2 + 2x_3 \\
 \max f^2(x) &= 2x_1 + 8x_2 + 10x_3 \\
 \text{subject to: } &x_1 + x_2 + x_3 \leq 1 \\
 &x_1, x_2, x_3 \in \{0, 1\},
 \end{aligned}
 \tag{5}$$

where $\{v^1, v^2, v^3, v^4\} = \{(1, 0, 0); (0, 1, 0); (0, 0, 1); (0, 0, 0)\}$ is the feasible set, $\{v^1, v^3\}$ is the efficient set, and all components are unstable. When applying Algorithm 1 on each efficient solution, the stability radii of v^1 and v^3 are 3 and 0, respectively. It means that v^1 remains efficient even if one increases or decreases by 3 the profit of each item (keeping such profits as non-negative), whereas there exists a profit matrix with distance 1 that leads to transform v^3 into a non-efficient solution. This strong difference does not appear when looking only at the non-dominated set in the objective space given by $\{(10, 2); (2, 10)\}$. It is due to the fact that all feasible solutions in Problem 5 are independent (the intersections between the feasible sets of items are empty) and to the existence of the non-efficient solution v^2 that is very close to v^3 in the objective space. Indeed, the independence implies that the outcome of a solution can be improved or deteriorated without modifying the outcome of another solution. Therefore, the outcome of v^3 can be deteriorated and the outcome of v^2 can be improved without modifying the outcome of v^1 . This explains the difference between v^1 and v^2 in terms of stability.

Let us consider the influence of stable components on this instance. If the profit of the second item is stable, then the stability radius of v^3 is increased to 1, because all the modifications to have v^2 dominate v^3 must be applied on the third item.

Consider, as a second example, the following instance:

$$\begin{aligned}
 \max f^1(x) &= 10x_1 + x_2 + x_3 \\
 f^2(x) &= 2x_1 + 8x_2 + 2x_3 \\
 \text{subject to: } &2x_1 + x_2 + x_3 \leq 2 \\
 &x_1, x_2, x_3 \in \{0, 1\},
 \end{aligned} \tag{6}$$

where $\{v^1, v^2, \dots, v^5\} = \{(1, 0, 0); (0, 1, 0); (0, 1, 1); (0, 0, 1); (0, 0, 0)\}$ is the feasible set, $\{v^1, v^3\}$ is the efficient set, and all components are unstable. Even though the image of the feasible set in the objective space is the same for this second instance, the stability radii are different. Their values are both equal to 3. This is because, in this case, v^2 is a subset of v^3 , which implies that any modification to the outcome of v^2 leads to a modification to the one of v^3 . In other words, for all profit matrices $D \in \mathbb{N}^{q \times n} : Dv^3 \geq Dv^2$. This explains why v^3 is more stable in this second example.

Consider, as a third example, the following instance:

$$\begin{aligned}
 \max f^1(x) &= 2x_1 + 4x_2 \\
 \max f^2(x) &= 4x_1 + 2x_2 \\
 \text{subject to: } &x_1 + x_2 \leq 1 \\
 &x_1, x_2 \in \{0, 1\},
 \end{aligned} \tag{7}$$

where $\{v^1, v^2, v^3\} = \{(1, 0); (0, 1); (0, 0)\}$ is the feasible set, $\{v^1, v^2\}$ is the efficient set, and the set of stable components is $S = \{(1, 1); (2, 1)\}$ (the first item's profits are stable). The stability radii of v^1 and v^2 are both equal to 1. This shows that even though v^1 is only composed of a single stable item and v^2 of a single unstable item, both have the same stability radius. If all components are unstable, then the stability radii of v^1 and v^2 are both equal to 0. This shows the influence of stable components on the stability radius.

Even though the situations presented in this section can be easily tackled, one should remember that such an analysis would become fastidious in large scale instances with complex combinatorial structures. This also shows the usefulness of computing the stability radius for providing information on the underlying structure of the feasible set.

6 Conclusion

In this paper, we have addressed the problem of computing the stability radius of an efficient solution in the context of MOCO. More precisely, we focused on the maximal perturbation of the objective functions coefficients such that a given solution remains efficient. This is measured under the Chebyshev norm and the coefficients are restricted to natural numbers. The stability radius of an efficient solution is modeled as a particular inverse optimization problem. The model is solved by an algorithm which requires only to solve a logarithmic number of mixed integer programs. It contains a linear number of constraints and variables compared with the instance of the combinatorial optimization problem if its feasible set can be defined by linear

constraints. To the best of our knowledge, this algorithm is the first one that allows to compute the stability radius in a reasonable amount of time. Further research questions will encompass the extension to the adjustment of all the problem parameters (and not only the objective functions coefficients) and to other norms (such as L_1).

References

- Chakravartia N, Wagelmansb APM (1998) Calculation of stability radius for combinatorial optimization. *Oper Res Lett* 23(1):1–7
- Coffin M, Saltzman MJ (2000) Statistical analysis of computational tests of algorithms and heuristics. *INFORMS J Comput* 12(1):24–44
- Ehrgott M (2005) *Multicriteria optimization*, 2nd edn. Springer, Berlin
- Emelichev VA, Kuzmin KG (2006) Stability radius of an efficient solution of a vector problem of integer linear programming in the gölder metric. *Cybern Syst Anal* 42(4):609–614
- Emelichev VA, Podkopaev D (2010) Quantitative stability analysis for vector problems of 0–1 programming. *Discret Optim* 7(1–2):48–63
- Emelichev VA, Kuz'min KG, Leonovich AM (2004) Stability in the combinatorial vector optimization problems. *Autom Remote Control* 65(2):227–240
- Kellerer H, Pferschy U, Pisinger D (1994) *Knapsack problems*. Springer, Berlin
- Martello S, Toth P (1990) *Knapsack problems: algorithms and computer implementations*. Wiley, Chichester
- Pisinger D (1995) *Algorithms for knapsack problems*. PhD thesis, Department of Computer Science, University of Copenhagen, Denmark
- Roland J, De Smet Y, Rui Figueira J (2011) The inverse multi-objective 0,1-knapsack problem under the chebyshev distance. Technical report TR/SMG/2011-002, SMG, CoDE, Université Libre de Bruxelles, Brussels
- Steuer RE (1986) *Multiple criteria optimization: theory, computation and application*. Wiley, New York
- Wendell RE, Lee DN (1977) Efficiency in multiple objective optimization problems. *Math Program* 12(1):406–414