



HAL
open science

Min Cost Improvement and Max Gain Stability in Multicriteria Sorting Methods on Combinatorial Domains

Nawal Benabbou, Hugo Martin, Patrice Perny

► **To cite this version:**

Nawal Benabbou, Hugo Martin, Patrice Perny. Min Cost Improvement and Max Gain Stability in Multicriteria Sorting Methods on Combinatorial Domains. *Journal of Multi-Criteria Decision Analysis*, 2021, 10.1002/mcda.1743 . hal-03271766

HAL Id: hal-03271766

<https://hal.sorbonne-universite.fr/hal-03271766v1>

Submitted on 27 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Min Cost Improvement and Max Gain Stability in Multicriteria Sorting Methods on Combinatorial Domains

Nawal Benabbou · Hugo Martin ·
Patrice Perny

Received: date / Accepted: date

Abstract Various multicriteria sorting methods have been proposed in the literature to assign the feasible alternatives into predefined categories. We consider here problems involving a set of totally ordered categories representing different achievement levels in the satisfaction of criteria. As in many existing methods, the assignment rule of an alternative to a category is based on the comparison of its performance vector to reference profiles defining lower bounds of the categories. Within this standard setting we address a new problem that consists in finding how to modify a given solution, within a combinatorial set of alternatives, to upgrade it in the upper category (or higher) at minimum cost. We also consider the problem of identifying the sequence of solutions that minimize the total cost while satisfying some budget constraint at every step, and the problem of determining how to modify the current solution to save money while staying in the same category. We first propose a general approach based on mixed integer (linear or quadratic) programming to solve these problems. Then, we implement this approach on various multiobjective combinatorial problems, such as multi-agent assignment problems and multiobjective knapsack problems. Numerical tests are provided to establish the feasibility of the approach on instances of different sizes.

Keywords Multicriteria Sorting Method · Multi-Objective Combinatorial Optimization · Min Cost Improvement · Max Gain Stability

Nawal Benabbou, Hugo Martin and Patrice Perny
Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6 F-75005 Paris,
France
E-mail: nawal.benabbou@lip6.fr, hugo.martin@lip6.fr, patrice.perny@lip6.fr

1 Introduction

In the field of multicriteria decision making, it is usual to distinguish different types of decision problems according to the nature of the recommendation sought, see e.g., [39]. Besides choice and ranking problems that are very frequent in practice and that motivate a large number of contributions in the field of multicriteria decision aid, there are other problems known as *sorting problems* where the goal is to assign the alternatives to predefined categories (ordered or not). One major difference with choice and ranking problems lies in the aim of defining an absolute evaluation system. In sorting problems, we are not merely interested in the relative value of the alternatives but also in their intrinsic qualities. What really matters is how alternatives compare to norms defining the categories. Very often, the categories are totally ordered and in this case the norms can be represented by different reference levels on an overall utility scale (see the UTADIS method [25]) or by different reference vectors in the space of criteria (different achievement levels in the satisfaction of criteria, as in the ELECTRE TRI method [32, 42]). In other cases, categories are not ordered (or only partially ordered) and represent different types of alternatives characterized by specific combinations of features [16, 24, 27].

Multicriteria sorting methods are useful for decision support because they provide intrinsic evaluations of the alternatives (objects, projects, candidates). They allow to partition the set of alternatives into specific classes corresponding to different types of values and norms, that may receive different treatments or lead to different decisions. Moreover, the assignment of an alternative to a category is easily explainable because it is based on the comparison of the alternative to reference norms. These norms can be explicitated as well as the criteria and the rule used to make the comparisons, and can be used as a support to explain the final decisions.

The multicriteria sorting problem has motivated various contributions in different directions. Different versions of the initial ELECTRE TRI method have been proposed and studied, see e.g., [1, 6, 10, 15, 28]. In addition, other multicriteria sorting procedures relying on similar principles have been proposed, offering new variants and some sophistications [11, 22, 30, 31, 44]. Moreover, the sorting problem has been extended to the case of fuzzy categories and some multicriteria filtering procedures based on the comparison of alternatives to reference points have been proposed to define the membership of any alternative to any category [2, 16, 21, 36]. Then the axiomatic foundations of sorting methods have been studied, especially in the case of non-compensatory comparison methods based on a weighted majority principle [4, 5, 40]. Another stream of research on sorting methods concerns the supervised learning of preference parameters used in such methods, typically the weights of criteria, the preference thresholds and/or the reference levels or reference points used to define the categories. Various methods have been proposed for the automatic

learning or the interactive elicitation of these parameters from assignment examples, e.g., [3, 23, 28, 41, 43].

Beyond these works that are still very active, other questions of interest concern post-assignment analysis aiming at investigating the possibility of preserving or improving the level of performance of a given solution or to find the best possible modifications leading to achieve a target level. These questions that have much importance in practice have recently led to various interesting proposals. Let us mention the *post-factum analysis* [9, 26] aiming to highlight the minimal improvements that would warrant the feasibility of a currently impossible outcome or the maximal deteriorations that alternatives can afford to maintain a target result. A similar concern is present in the literature on *stepwise benchmarking* [14, 17, 37, 38] that deals with finding an optimal improvement path for objects, units or entities and to provide recommendations for stepwise improvement.

In this paper, we focus on similar issues for multi-criteria evaluation problems in which the overall value of a solution is defined by its assignment to a category using a sorting procedure (each category representing a clearly identified level of performance). More precisely, we consider the following questions:

1. The Min Cost Improvement problem: given a feasible solution of a multi-objective optimization problem, assigned to a given category by a sorting procedure, find the min cost modifications of this solution that will lead to a new feasible solution belonging to an upper category targeted by the decision maker.
2. The Max Gain Stability problem: given a feasible solution of a multiobjective optimization problem, assigned to a given category by a sorting procedure, find a new solution in the same category allowing the maximal gain (or cost saving).
3. The Min Cost Improving Sequence problem: given a feasible solution x^0 of a multiobjective optimization problem, and a fixed number of steps h , find the sequence of solutions (x^0, \dots, x^h) that minimizes the total cost among those with x^h belonging to the category targeted by the decision maker.

These questions which correspond to very natural practical concerns lead to new optimization problems that must be formulated and solved. The closest work we know on such questions is a recent work on the “Inverse Multicriteria Sorting Problem (IMSP)” [33] but the framework is a bit different. Starting from a given alternative, it is assumed in the IMSP that various possible actions can modify the performance of this alternative and the aim is to find the combination of actions that will allow a min cost improvement, or a max improvement for a given cost. Hence, the admissible modifications of solutions are not imposed by the structure of the set of alternatives but by external actions that could directly modify the criterion values of the alternative under consideration.

As far as we know, very few contributions on this subject address these questions on combinatorial domains. However such questions may also appear in multiobjective combinatorial optimization problems [12, 13], for example to improve the overall quality of a solution to a multiobjective transportation problem aimed at supplying stores from factory warehouses, to increase the efficiency or satisfaction of a team by modifying the assignment of tasks incumbent on each individual, or to improve the value of a group of objects or individuals selected from a set of potential candidates under a budget constraint (multiobjective knapsack problem). When the list of feasible solutions is not explicit, it is much harder to explore the possible solutions that belong to the target category, and this makes the search of the most adequate improvements much more complex. In particular, when the set of feasible alternatives has a combinatorial structure, finding optimal improvements from a given solution leads to combinatorial optimization problems that must be formulated and solved. This paper aims to address these problems.

This is why, in this paper, we consider the case of a multiobjective combinatorial optimization problem where the value of every feasible solution is defined by a multicriteria sorting procedure involving a list of pre-defined ordered categories. Any modification of a solution may generate losses or gains and is limited by feasibility constraints defining the set of admissible solutions. Under these assumptions, we want to propose computational models to answer the two above mentioned questions.

The paper is organized as follows: first, we present the multicriteria sorting method studied in the paper (Section 2). Then, we introduce the min cost improvement, max gain stability and the min cost improving sequence problems, and provide mixed integer (linear/quadratic) formulations to efficiently solve these problems (Section 3). Finally, we provide numerical results for three multicriteria sorting problems on combinatorial domains (transportation, assignment, and knapsack problems) showing the practical efficiency of our approach (Section 4).

2 The Multicriteria Sorting Method

In this paper, we consider sorting problems where alternatives/solutions are evaluated with respect to n criteria to be maximized. The set of alternatives can be implicitly defined as the set of all feasible solutions of a multi-objective combinatorial optimization problem, as defined here below:

$$\begin{aligned} \max \quad & f(x) = Wx \\ \text{s.t.} \quad & \begin{cases} Ax \leq b \\ x \in \mathbb{N}^m \end{cases} \end{aligned}$$

where $x^T = (x_1, \dots, x_m) \in \mathbb{N}^m$ is a solution vector, $W \in \mathbb{Z}^{n \times m}$ is a matrix allowing the definition of performance vectors from solution vectors, $f(x)^T =$

$(f_1(x), \dots, f_n(x)) \in \mathbb{Z}^n$ is the performance vector giving the evaluation of x with respect to every criterion $i \in N = \{1, \dots, n\}$, and $A \in \mathbb{Z}^{d \times m}$ and $b \in \mathbb{Z}^d$ are matrices that jointly impose $d \in \mathbb{N}$ feasibility constraints on solutions. The set of all feasible solutions will be denoted by \mathcal{X} in the sequel.

In sorting problems, alternatives are to be assigned to categories based on their performance vector. Here we consider a set of q categories denoted by $\mathcal{K} = \{K_1, \dots, K_q\}$, where K_1 is the best category and K_q is the worst one. The bounds of categories are defined using a set \mathcal{R} of multicriteria reference profiles $\mathcal{R} = \{r^0, r^1, \dots, r^q\}$, where $r^\ell = (r_1^\ell, \dots, r_n^\ell) \in \mathbb{R}^n$ defines the lower boundary of category K_ℓ , $\ell \in \{1, \dots, q\}$, and $r^0 \in \mathbb{R}^n$ represents a top performance profile chosen to bound above all criterion values. These reference profiles are defined in such a way that $r^\ell \gg r^{\ell+1}$ for all $\ell \in \{0, \dots, q-1\}$, where \gg is the strict Pareto dominance relation defined as follows: $a \gg b$ if and only if $a_i > b_i$ for all $i \in N$. At some point, we will also consider the weak Pareto dominance (i.e. $a \succeq b$ if and only if $a_i \geq b_i$ for all $i \in N$) and the Pareto dominance (i.e. $a \succ b$ if and only if $a_i \geq b_i$ for all $i \in N$ and $a_i > b_i$ for some $i \in N$).

In this context, the general principle of preference-based assignment methods is to assign alternatives to categories according to the following rule:

Assignment Rule (*): Alternative $x \in \mathcal{X}$ is assigned to category K_ℓ if and only if x is preferred to profile r^ℓ and x is not preferred to profile $r^{\ell-1}$.

We focus here on the assignment method introduced in [36] which is a soft implementation of this principle using valued preference relations and fuzzy membership to categories. This sorting method is based on the following preference indices:

Monocriterion preference index: For any criterion $i \in N$, the preference index $P_i(x, r^\ell)$ of any alternative $x \in \mathcal{X}$ with respect to any reference profile $r^\ell \in \mathcal{R}$ is defined by:

$$P_i(x, r^\ell) = \begin{cases} 1 & \text{if } f_i(x) - r_i^\ell > \gamma_i^+ \\ \frac{f_i(x) - r_i^\ell - \gamma_i^-}{\gamma_i^+ - \gamma_i^-} & \text{if } \gamma_i^- < f_i(x) - r_i^\ell \leq \gamma_i^+ \\ 0 & \text{if } f_i(x) - r_i^\ell \leq \gamma_i^- \end{cases} \quad (2)$$

This index measures the credibility of the statement “alternative x is better than profile r^ℓ with respect to criterion i ”. The variation of $P_i(x, r^\ell)$ as the performance difference $f_i(x) - r_i^\ell$ increases is represented in Figure 1. This preference index is maximal (i.e. equals 1) when the difference $f_i(x) - r_i^\ell$ is larger than γ_i^+ the *preference threshold* representing the minimal performance difference compatible with a strict preference. This index is minimal (i.e. equals 0) when the difference $f_i(x) - r_i^\ell$ is below γ_i^- the *indifference threshold* representing the maximal performance difference compatible with an indifference (absence of preference). Between these two thresholds, there is an area where the decision maker may hesitate between strict preference and indifference; in

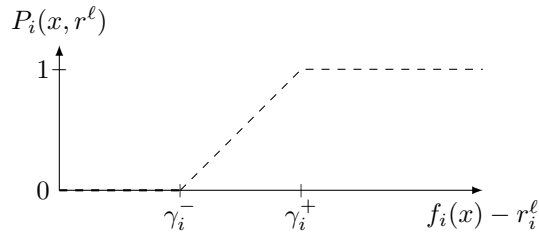


Fig. 1: Preference index $P_i(x, r^\ell)$ as a function of $f_i(x) - r_i^\ell$.

this area, the monocriterion preference index grows linearly with the difference $f_i(x) - r_i^\ell$ (see Figure 1). We assume here that preference and indifference thresholds are part of the definition of the criterion scale and they are defined in such a way that $\gamma_i^+ > \gamma_i^-$ holds.

These monocriterion indices are then aggregated into an overall preference index $P(x, r^\ell)$ using an aggregation function ψ_θ parameterized by θ :

Overall preference index: The overall preference index $P(x, r^\ell)$ of any solution $x \in \mathcal{X}$ with respect to any reference profile $r^\ell \in \mathcal{K}$ is defined by:

$$P(x, r^\ell) = \psi_\theta(P_1(x, r^\ell), \dots, P_n(x, r^\ell)) \quad (3)$$

This overall index represents the strength of arguments supporting the statement “alternative x is better than profile r^ℓ ”. A standard choice for ψ_θ is a compromise operator, i.e. an aggregation function that verifies $\min_{i \in N} \{a_i\} \leq \psi_\theta(a) \leq \max_{i \in N} \{a_i\}$ for all $a = (a_1, \dots, a_n) \in \mathbb{R}^n$. When considering such an operator, $P(x, r^\ell)$ necessarily belongs to $[0, 1]$. It is equal to 1 when the criteria are unanimously in favor of strict preference, and equals 0 when they are unanimously against preference. Moreover, ψ_θ is assumed to be compatible with weak Pareto-dominance, i.e. $\psi_\theta(a) \geq \psi_\theta(b)$ for all $a, b \in \mathbb{R}^n$ such that $a \succeq b$. In this paper, a special focus will be placed on the discrete Choquet integral, which is a family of (non-linear) aggregation functions that is very popular in multicriteria decision making as it enables to model possible synergies between criteria [18, 20]. These aggregation functions are specified by a capacity function $\theta : 2^N \rightarrow [0, 1]$ such that $\theta(\emptyset) = 0$, $\theta(N) = 1$ and $\theta(A) \leq \theta(B)$ for all $A \subset B \subseteq N$, which defines the weight attached to any coalition of criteria [19]. The discrete Choquet integral assigns to any performance vector $a = (a_1, \dots, a_n) \in \mathbb{R}^n$ the following value:

$$\psi_\theta(a) = \sum_{i=1}^n (a_{(i)} - a_{(i-1)}) \theta(A_{(i)}) \text{ with } a_{(0)} = 0,$$

where (\cdot) is a permutation such that $a_{(i-1)} \leq a_{(i)}$ for all $i \in N$ (sorting the components of vector a in increasing order) and $A_{(i)} = \{(i), \dots, (n)\}$ is the set of criteria with respect to which alternative a has a performance greater or equal to $a_{(i)}$. For illustration purposes, let us consider the capacity function

	\emptyset	$\{1\}$	$\{2\}$	$\{3\}$	$\{1, 2\}$	$\{1, 3\}$	$\{2, 3\}$	$\{1, 2, 3\}$
θ	0	0.2	0.1	0.3	0.4	0.7	0.6	1

Table 1: An example of capacity function.

given in Table 1. For $a = (3, 2, 5)$, we have $\psi_\theta(a) = (2 - 0) \times \theta(\{1, 2, 3\}) + (3 - 2) \times \theta(\{1, 3\}) + (5 - 3) \times \theta(\{3\}) = 2 \times 1 + 1 \times 0.7 + 2 \times 0.3 = 3.3$.

The family of Choquet integrals includes the weighted sums (when θ is additive, i.e. $\theta(A \cup B) = \theta(A) + \theta(B)$) and OWA aggregators (when θ is symmetric, i.e. $\theta(A) = \theta(B)$ for all A, B s.t. $|A| = |B|$) as a special case. Moreover, using a convex capacity enables to model preferences for well-balanced preference profiles (see e.g., [8]). Moreover, these subclasses are very interesting from a computational perspective since efficient linearization techniques have been proposed for these functions [7, 29, 34, 35].

Finally, the membership indices and the assignment rule are defined as follows:

Membership index: The membership index $c_\ell(x)$ of any alternative $x \in \mathcal{X}$ to any category $K_\ell \in \mathcal{K}$ is defined by:

$$c_\ell(x) = \min \left\{ P(x, r^\ell), 1 - P(x, r^{\ell-1}) \right\} \quad (4)$$

This index measures the membership of any alternative x to any category K_ℓ on the $[0, 1]$ scale. Equation (4) is a translation in multi-valued logic of the assignment rule (*): in this equation, $1 - P(x, r^{\ell-1})$ is used for the negation and the min t-norm for the conjunction. For any fixed alternative x , $c_\ell(x)$ seen as a function of ℓ is unimodal, with a maximum at least equal to $1/2$ (for more details see [36]).

Assignment rule ():** Alternative $x \in \mathcal{X}$ is assigned to category K_ℓ if and only if ℓ is the smallest index such that $c_\ell(x) = \max_{\ell' \in \{1, \dots, q\}} c_{\ell'}(x)$.

In the sequel, $c(x)$ will denote the index of the category in which x is assigned.

This preference-based sorting method is typical of the “*compare then aggregate*” approach: to determine the category of any alternative $x \in \mathcal{X}$, we first compare x to all reference profiles $r^\ell \in \mathcal{R}$ by considering each criterion separately, leading to monocriterion indices $P_i(x, r^\ell), i \in N$. These indices are then aggregated using (3) to define $P(x, r^\ell)$ and the membership of x to any category K_ℓ by (4). The advantage of this approach lies in the use of reference vectors r^ℓ instead of scalar thresholds on aggregated values, which allows a finer control in the definition of category boundaries: two alternatives having the same “average” performance but different performance vectors may be assigned to different categories, as shown in the following example:

Example 1 Consider a sorting problem with 3 criteria f_1, f_2, f_3 and 3 reference profiles r^0, r^1, r^2 (given in Table 2) allowing the definition of two categories K_1 and K_2 . We use the same preference threshold $\gamma_i^+ = 1$ and indif-

	f_1	f_2	f_3
r_0	30	30	30
r_1	11	10	12
r_2	0	0	0

Table 2: Multicriteria reference profiles defining categories in Example 1.

ference threshold $\gamma_i^- = 0$ for all criteria $i \in N = \{1, 2, 3\}$. We aggregate the preference indices using a Choquet Integral ψ_θ with the capacity θ defined as follows (Table 3):

	\emptyset	$\{1\}$	$\{2\}$	$\{3\}$	$\{1, 2\}$	$\{1, 3\}$	$\{2, 3\}$	$\{1, 2, 3\}$
θ	0	0.2	0.1	0.3	0.4	0.7	0.6	1

Table 3: Capacity θ used in Example 1.

Consider 2 alternatives x, y with the following performance vectors: $f(x) = (6, 11, 16)$ and $f(y) = (7, 6, 20)$. Note that alternatives x and y would be indiscernible after using ψ_θ as a direct scalarization function since they have the same aggregated value: $\psi_\theta(f(x)) = 6 + (11 - 6) \times 0.6 + (16 - 11) \times 0.3 = 10.5$ and $\psi_\theta(f(y)) = 6 + (7 - 6) \times 0.6 + (20 - 7) \times 0.3 = 10.5$. Let us compare x and y indirectly using reference profiles r^1, r^2 and r^3 . Using Equations (3) and (4), we obtain the following overall preference indices and membership values:

	$P(\cdot, r^0)$	$P(\cdot, r^1)$	$P(\cdot, r^2)$	$c_1(\cdot)$	$c_2(\cdot)$
x	0	0.6	1	0.6	0.4
y	0	0.3	1	0.3	0.7

Table 4: Overall preference indices and membership values in Example 1.

For example, using Equation (3), $P(x, r^1)$ is obtained as follows: $P_1(x, r^1) = 0$ (since $f_1(x) - r_1^1 = 6 - 11 \leq 0 = \gamma_1^-$), $P_2(x, r^1) = 1$ and $P_3(x, r^1) = 1$, and $P(x, r^1) = \psi_\theta(0, 1, 1) = 0 + (1 - 0) \times \theta(\{2, 3\}) + (1 - 1) \times \theta(\{3\}) = 0.6$. Furthermore, using Equation (4), we have $c_1(x) = \min\{P(x, r^1), 1 - P(x, r^0)\} = \min\{0.6, 1 - 0\} = 0.6$. Thus x is assigned to category K_1 (i.e. $c(x) = 1$) whereas y is assigned to category K_2 (i.e. $c(y) = 2$); thus x obtains a better position than y in this example. Moreover, it can easily be checked that y remains in K_2 when y_1 increases from 20 to 30, although its aggregated value $\psi_\theta(y)$ increases: there is no point in improving y_1 since it already exceeds r_1^1 while the weaknesses of y on criteria 2 and 3 remain. This illustrates the non-compensatory nature of this sorting procedure, where the difference of performances does not play any role beyond a given threshold.

3 Optimization Problems and Integer Linear Formulations

In this section, we study three optimization problems in which the decision maker wants to move from a given solution to another feasible solution of a given multi-objective optimization problem in order to achieve a target level efficiently, namely the Min Cost Improvement, Max Gain Stability and the Min Cost Improving Sequence problems.

3.1 The Min Cost Improvement Problem

Given a solution $x \in \mathcal{X}$, we consider the problem of moving from x to another solution $y \in \mathcal{X}$ in order to enter into a higher category at a minimal cost. This optimization problem, called Min Cost Improvement (MCI) problem, can be formulated as follows:

$$\begin{aligned} \min g(x, y) & \tag{5a} \\ (\mathcal{P}_{MCI}) : \quad \text{s.t.} \quad & \begin{cases} c(y) \leq c(x) - 1 & \tag{5a} \\ Ay \leq b & \tag{5b} \\ y \in \mathbb{N}^m & \tag{5c} \end{cases} \end{aligned}$$

where $g : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a real-valued function giving the cost of moving from a given solution $x \in \mathcal{X}$ to any another solution $y \in \mathcal{X}$, Equation (5a) ensures that y is assigned to a strictly higher category than x , and Equations (5b-5c) are the feasibility constraints. Given any solution $x \in \mathcal{X}$, the optimal solution y^* of problem \mathcal{P}_{MCI} is the solution that improves x at a minimal cost. When the decision maker does not only wish to enter into a higher category but targets a given category K_ℓ with $\ell < c(x) - 1$, Equation (5a) must be replaced by the constraint $c(y) \leq \ell$. For illustration purposes, an instance of the MCI problem is given below.

Example 2 Consider a knapsack problem where the set $\mathcal{O} = \{o_1, o_2, o_3\}$ of objects/items is evaluated with respect to 2 additive criteria f_1, f_2 as follows¹:

	f_1	f_2
o_1	2	2
o_2	4	1
o_3	1	3

Table 5: The performance vectors attached to objects in Example 2.

Any solution can be represented by a binary vector $x = (x_1, x_2, x_3)$ where $x_i = 1$ if and only if object o_i is selected, and the knapsack constraint only imposes

¹ Additive means that the value of a set of objects is the sum of the values of its elements.

that no more than 2 objects are selected. Moreover, for any two solutions x, y , let us assume that the cost $g(x, y)$ of moving from x to y is defined by:

$$g(x, y) = |x_1 - y_1| + |x_2 - y_2| + |x_3 - y_3|$$

penalizing solutions that are far from x in the decision space. Now consider the multicriteria reference profiles given in Table 2. These reference profiles

	f_1	f_2
r_0	6	6
r_1	3	2
r_2	-1	-1

Table 6: The multicriteria reference profiles used in Example 2.

allow the definition of two categories for this problem. Finally, assume that the preference and indifference thresholds are respectively given by $\gamma_1^+ = \gamma_2^+ = 1$ and $\gamma_1^- = \gamma_2^- = 0$, and that the overall indices are obtained by aggregating monocriterion preference indices using the weighted sum ψ_θ defined by $\psi_\theta = \theta f_1(x) + (1-\theta)f_2(x)$ with $\theta = 2/3$. In this problem, there are only seven feasible solutions, denoted by x^j , $j \in \{0, \dots, 6\}$, with the following overall preference indices and membership values:

	f_1	f_2	$P(\cdot, r^0)$	$P(\cdot, r^1)$	$P(\cdot, r^2)$	$c_1(\cdot)$	$c_2(\cdot)$	$c(\cdot)$
$x^0 = (0, 0, 0)$	0	0	0	0	1	0	1	2
$x^1 = (1, 0, 0)$	2	2	0	0	1	0	1	2
$x^2 = (0, 1, 0)$	4	1	0	2/3	1	2/3	1/3	1
$x^3 = (0, 0, 1)$	1	3	0	1/3	1	1/3	2/3	2
$x^4 = (1, 1, 0)$	6	3	0	1	1	1	0	1
$x^5 = (1, 0, 1)$	3	5	0	1/3	1	1/3	2/3	2
$x^6 = (0, 1, 1)$	5	4	0	1	1	1	0	1

Table 7: Overall preference indices and membership values attached to the feasible solutions in Example 2.

Thus x^0 , x^1 , x^3 and x^5 belong to K_2 (the worst category) whereas x^2 , x^4 and x^6 belong to K_1 (the best category). If x^1 is the current solution, then x^4 is the unique optimal solution of the MCI problem (since we have $g(x^1, x^4) = 1 < g(x^1, x^2) = 2 < g(x^1, x^6) = 3$). If instead x^3 is the current solution, then x^6 is the unique optimal solution (as $g(x^3, x^6) = 1 < g(x^3, x^2) = 2 < g(x^3, x^4) = 3$).

3.2 The Max Gain Stability Problem

Similarly, in a context where any solution has a cost, we consider the problem of reducing the cost while staying in the same category. This amounts to maximizing the gain or cost saving to pass from the current solution x to another solution $y \in \mathcal{X}$. This problem named Max Gain Stability (MGS) can be formulated as follows for any $x \in \mathcal{X}$:

$$\begin{aligned} & \max g(x, y) \\ (\mathcal{P}_{MGS}) : & \text{ s.t. } \begin{cases} c(y) \leq c(x) & (6a) \\ Ay \leq b & (6b) \\ y \in \mathbb{N}^m & (6c) \end{cases} \end{aligned}$$

where $g : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a real-valued function defining the gains and Equation (6a) ensures that y is at least as good as x . When the decision maker is not against reaching a given lower K_ℓ (with $\ell > c(x)$) to maximize her gain, then Equation (6a) must be replaced by the constraint $c(y) \geq \ell$. Note that the formulation of the MGS problem is very similar to that of the MCI problem and therefore can be solved using the same tools. An instance of the MGS problem is given below.

Example 3 *Let us come back to Example 2 and assume that every object $o_k \in \mathcal{O}$ has a market price $\pi_k \in \mathbb{R}_+$ so that the gain $g(x, y)$ obtained when moving from a given solution x to another solution y is simply defined by:*

$$g(x, y) = \pi_1(x_1 - y_1) + \pi_2(x_2 - y_2) + \pi_3(x_3 - y_3)$$

In that case, for any current solution $x \in K_1$, the unique optimal solution of the MGS problem is $x^2 = (0, 1, 0)$, and for any current solution $x \in K_2$, the unique optimal solution is trivially $x^0 = (0, 0, 0)$.

3.3 The Min Cost Improving Sequence Problem

In some situations, a large improvement of the current solution may be unfeasible in one step due to various reasons, for instance because it requires a prohibitive amount of resources in a short period of time or because it induces too important changes that may have significant reconfiguration costs. In order to cope with such situations, we now consider problems where the decision maker wishes to reach a given higher category in possibly several steps at a minimal cost. The aim is therefore to determine which are the proper steps to reach the targeted category optimally. More formally, we are given:

- x^0 : solution at time 0 (current solution)
- ℓ^* : index of the target category

- h : number of periods allowed to achieve the goal (horizon)
- β_1, \dots, β_h : available budget for each period

A feasible solution to the Min Cost Improving Sequence Problem (MCIS) is a sequence (x^1, \dots, x^h) that satisfies the following constraints:

- $Ax^t \leq b$ for all $t \in \{1, \dots, h\}$ (i.e. $x^t \in \mathcal{X}$)
- $c(x^h) \leq \ell^*$ (the last element in the sequence belongs to the targeted category or some higher category)
- $g(x^{t-1}, x^t) \leq \beta_t$ for all $t \in \{1, \dots, h\}$ where $g : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is the cost function of the problem (modifications in the sequence are under a budgetary constraint which depends on the period of time)

The goal is to identify a feasible solution (x^1, \dots, x^h) that minimizes the total cost which is simply defined by:

$$\sum_{t=1}^h g(x^{t-1}, x^t)$$

The MCIS problem can therefore be formulated as follows:

$$\begin{aligned}
 & \min \sum_{t=1}^h g(x^{t-1}, x^t) \\
 (\mathcal{P}_{MCIS}) : & \quad \text{s.t.} \begin{cases} c(x^h) \leq \ell^* \\ g(x^{t-1}, x^t) \leq \beta_t, \forall t \in \{1, \dots, h\} \\ Ax^t \leq b, \forall t \in \{1, \dots, h\} \\ x^t \in \mathbb{N}^m, \forall t \in \{1, \dots, h\} \end{cases}
 \end{aligned}$$

Note that we deliberately omitted monotonicity constraints of type $c(x^t) \leq c(x^{t-1})$ for $t \in \{1, \dots, h\}$ because they are not necessary as soon as the following triangular inequality holds: $g(x, y) \leq g(x, z) + g(z, y)$. This is the case for all functions g considered here (e.g., when $g(x, y)$ is a linear function of $x - y$, or when g is a distance defined from a norm by $g(x, y) = \|x - y\|$). This makes impossible to find 3 solutions x, y, z with $c(y) < c(x)$ (y improves x) and such that the cost of a two step path $x \rightarrow z \rightarrow y$ is lower than the cost of the direct path $x \rightarrow y$. Hence, whenever z belongs to a category worse than that of x (i.e., $c(z) > c(x)$), there is no benefit in moving first from x to z and then from z to y compared to directly moving from x to y since $g(x, z) + g(z, y) \geq g(x, y)$. Moreover, if reaching level ℓ or better must be achieved in the sequence of moves at some intermediary step t , one can easily insert the following constraint $c(x^t) \leq \ell$ in program \mathcal{P}_{MCIS} .

We now give an instance of the MCIS problem below.

Example 4 Let us come back to Example 2. Consider the multicriteria reference profiles (given in Table 8) which allow the definition of three categories for this problem.

	f_1	f_2
r_0	6	6
r_1	4	4
r_2	2	2
r_3	-1	-1

Table 8: The multicriteria reference profiles in Example 2.

Here there are only seven feasible solutions, denoted by x^j , $j \in \{0, \dots, 6\}$, and their overall preference indices and membership values are given in Table 9. From this table, we conclude that solutions x^4 and x^6 enter into category K_1

	f_1	f_2	$P(\cdot, r^0)$	$P(\cdot, r^1)$	$P(\cdot, r^2)$	$P(\cdot, r^3)$	$c_1(\cdot)$	$c_2(\cdot)$	$c_3(\cdot)$	$c(\cdot)$
$x^0 = (0, 0, 0)$	0	0	0	0	0	1	0	0	1	3
$x^1 = (1, 0, 0)$	2	2	0	0	0	1	0	0	1	3
$x^2 = (0, 1, 0)$	4	1	0	0	2/3	1	0	2/3	1/3	2
$x^3 = (0, 0, 1)$	1	3	0	0	1/3	1	0	1/3	2/3	3
$x^4 = (1, 1, 0)$	6	3	0	2/3	1	1	2/3	1/3	0	1
$x^5 = (1, 0, 1)$	3	5	0	1/3	1	1	1/3	2/3	0	2
$x^6 = (0, 1, 1)$	5	4	0	2/3	1	1	2/3	1/3	0	1

Table 9: Overall preference indices and membership values attached to the feasible solutions in Example 4.

(the best category), solutions x_2 and x_5 enter into category K_2 (the second best category) while x^0 , x^1 and x^3 enter to category K_3 (the worst category).

Assume that the current solution is x^0 (belonging to K_3), and that we want to reach category K_1 in at most two steps. The graph depicted in Figure 2 represents all feasible sequences of solutions, i.e. the sequences that start from x_0 and end with a solution in K_1 in at most two steps.

In Figure 2, any arrow of type (x^i, x^j) is labelled by $g(x^i, x^j)$ the cost of moving from solution x^i to solution x^j . Without any budgetary constraint, we know that we can focus the search on paths of size 1 as g satisfies the triangular inequality. In particular, sequence (x^0, x^4) , which consists in moving from x^0 to x^4 in one step, is an optimal solution of the MCIS problem, with a total cost of 2.

Now assume that the cost of each step must not exceed 1 (i.e. we set $\beta_1 = \beta_2 = 1$). In that case, sequence (x^0, x^4) is not feasible anymore, and now sequences (x^0, x^2, x^6) and (x^0, x^2, x^4) are the unique optimal solutions of the MCIS problem, with a total cost of $1 + 1 = 2$.

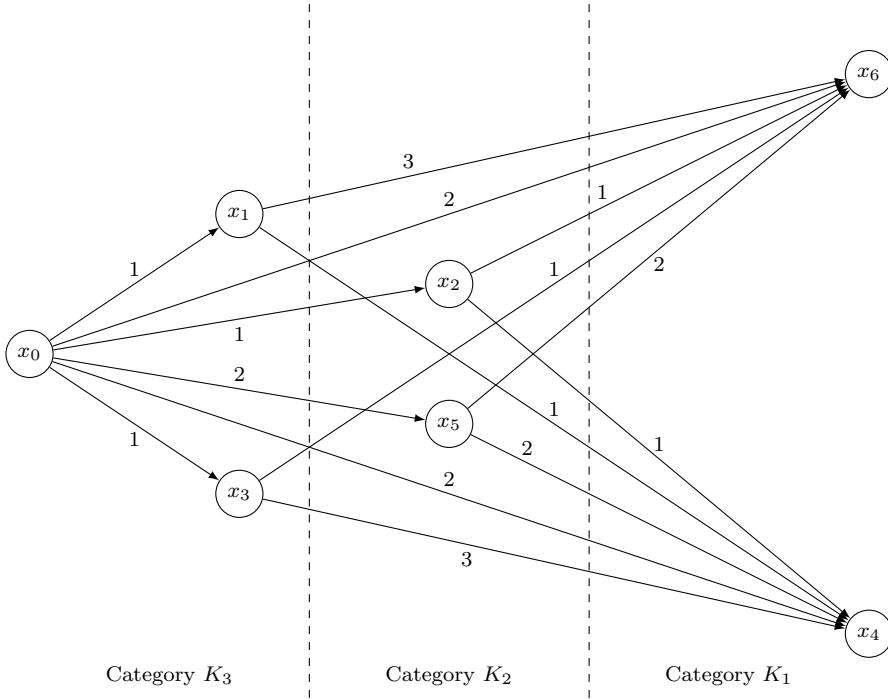


Fig. 2: Feasible sequences of solutions in Example 4.

3.4 Integer Linear Formulations

In the previous subsections, we have introduced three optimization problems, namely the MCI, MGS and MCIS problems, and they have been solved on very small instances of knapsack problems by enumerating all the feasible solutions (see Examples 2, 3 and 4). For bigger instances, it is not possible to enumerate all the feasible solutions as they are too numerous, and therefore there is a need for tools to solve these optimization problems efficiently.

In this subsection, we propose solving methods based on mixed-integer (linear or quadratic) programming. For simplicity of presentation, we now focus on the MCI problem as these three optimization problems are very similar.

In order to obtain a general mixed-integer formulation of problem \mathcal{P}_{MCI} , we now introduce two technical propositions, the proofs of which are given in the Appendix (see Section 6).

Proposition 1 *For any solution $y \in \mathcal{X}$ and any category $K_\ell \in \mathcal{K}$:*

$$c(y) \leq \ell \iff P(y, r^\ell) \geq 1/2.$$

When setting $\ell = c(x) - 1$, we see that constraint $c(y) \leq c(x) - 1$, which corresponds to Equation (5a) in \mathcal{P}_{MCI} , holds if and only if $P(y, r^{c(x)-1}) \geq 1/2$. As a consequence solving \mathcal{P}_{MCI} amounts to solving the following optimization problem, denoted by \mathcal{P}'_{MCI} :

$$\begin{aligned} & \min g(x, y) \\ (\mathcal{P}'_{MCI}) : & \text{ s.t. } \begin{cases} P(y, r^{c(x)-1}) \geq 1/2 & (8a) \\ Ay \leq b & (8b) \\ y \in \mathbb{N}^m & (8c) \end{cases} \end{aligned}$$

Now in order to solve \mathcal{P}'_{MCI} , we propose to use the following program:

$$\begin{aligned} & \min g(x, y) \\ (\mathcal{P}''_{MCI}) : & \text{ s.t. } \begin{cases} \psi_\theta(p_1, \dots, p_n) \geq 1/2 & (9a) \\ p_i \leq \frac{f_i(y) - r_i^{c(x)-1} - \gamma_i^-}{\gamma_i^+ - \gamma_i^-} + (1 - v_i)M, \forall i \in N & (9b) \\ p_i \leq Mv_i, \forall i \in N & (9c) \\ Ay \leq b & (9d) \\ y \in \mathbb{N}^m & \\ p = (p_1, \dots, p_n) \in [0, 1]^n, v = (v_1, \dots, v_n) \in \{0, 1\}^n & (9e) \end{cases} \end{aligned}$$

where M is a constant greater than one allowing the introduction of boolean variables v_i , $i \in N$, that ensure that all variables p_i satisfy $p_i \leq P_i(y, r^{c(x)-1})$ (see Lemma 1 in the Appendix).

Interestingly, program \mathcal{P}''_{MCI} can be used to solve \mathcal{P}'_{MCI} due to the following proposition:

Proposition 2 *For any feasible solution $y \in \mathbb{N}^m$ of problem \mathcal{P}'_{MCI} , the following two properties are equivalent:*

1. *Solution y is optimal for problem \mathcal{P}'_{MCI} .*
2. *There exist $p \in [0, 1]^n$ and $v \in \{0, 1\}^n$ such that (y, p, v) is a feasible optimal solution of problem \mathcal{P}''_{MCI} .*

Note that \mathcal{P}''_{MCI} is a mixed-integer linear program when the cost function is linear. When g is not linear but linearizable or when g is a quadratic function, problem \mathcal{P}''_{MCI} can still be solved efficiently using a state-of-the-art solver, as shown in Section 4.

Using similar arguments, one can easily prove that the MGS problem can be solved using the following program:

$$\begin{aligned}
& \max g(x, y) \\
(\mathcal{P}''_{MGS}) : & \text{ s.t. } \begin{cases} \psi_\theta(p_1, \dots, p_n) \geq 1/2 \\ p_i \leq \frac{f_i(y) - r_i^{c(x)} - \gamma_i^-}{\gamma_i^+ - \gamma_i^-} + (1 - v_i)M, \forall i \in N \\ p_i \leq Mv_i, \forall i \in N \\ Ay \leq b \end{cases} \\
& y \in \mathbb{N}^m \\
& p = (p_1, \dots, p_n) \in [0, 1]^n, v = (v_1, \dots, v_n) \in \{0, 1\}^n
\end{aligned}$$

and the same applies to the MCIS problem:

$$\begin{aligned}
& \min \sum_{t=1}^h g(x^{t-1}, x^t) \\
(\mathcal{P}''_{MCIS}) : & \text{ s.t. } \begin{cases} \psi_\theta(p_1, \dots, p_n) \geq 1/2 \\ p_i \leq \frac{f_i(x^h) - r_i^{\ell^*} - \gamma_i^-}{\gamma_i^+ - \gamma_i^-} + (1 - v_i)M, \forall i \in N \\ p_i \leq Mv_i, \forall i \in N \\ g(x^{t-1}, x^t) \leq \beta_t, \forall t \in \{1, \dots, h\} \\ Ax^t \leq b, \forall t \in \{1, \dots, h\} \end{cases} \quad (11a) \\
& x^t \in \mathbb{N}^m, \forall t \in \{1, \dots, h\} \\
& p = (p_1, \dots, p_n) \in [0, 1]^n, v = (v_1, \dots, v_n) \in \{0, 1\}^n
\end{aligned}$$

4 Some Applications

Depending on the decision problem, function g giving the cost/gain of moving from a given solution to another solution can take different forms. For illustrative purposes, we now study three realistic instances of the MCI problem on combinatorial domains with different cost functions to be minimized².

² Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

4.1 A Transportation Problem

Consider a transportation problem dealing with the distribution of products from a set of m factories $\mathcal{F} = \{F_1, \dots, F_m\}$ to a set of n clients $\mathcal{C} = \{C_1, \dots, C_n\}$, each factory F_j having a limited number of units to supply. We are given a cost matrix $g = (g_{ij})_{n \times m}$ so that $g_{ij} \in \mathbb{R}^+$ is the cost of transporting one unit of products from factory F_j to client C_i for all $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$. In this problem, a feasible solution can be represented by a matrix $X = (x_{ij})_{n \times m}$ where $x_{ij} \in \mathbb{R}^+$ is the number of units shipping from F_j to C_i for all $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$, and the corresponding cost is given by $T(x) = \sum_{i=1}^n \sum_{j=1}^m x_{ij} g_{ij}$. Thus, the cost $g(x, y)$ of moving from a given solution x to another solution y is simply given by $g(x, y) = T(y) - T(x) = \sum_{i=1}^n \sum_{j=1}^m (y_{ij} g_{ij} - x_{ij} g_{ij})$ in this problem.

Now suppose that every client C_i , $i \in \{1, \dots, n\}$, provides 3 values $\alpha_i^1, \alpha_i^2, \alpha_i^3$ representing the minimum number of units required to be very satisfied, satisfied, and slightly satisfied respectively. These thresholds lead to 4 satisfaction categories K_1 (very satisfied), K_2 (satisfied), K_3 (slightly satisfied) and K_4 (not satisfied), which are characterized by the set $\mathcal{R} = \{r^0, r^1, r^2, r^3, r^4\}$ of multicriteria reference profiles defined as follows:

- $r^\ell = (\alpha_1^\ell, \dots, \alpha_n^\ell)$ for all $\ell \in \{1, 2, 3\}$,
- $r^4 = (-1, \dots, -1)$,
- and $r^0 = (M, \dots, M)$ with M large enough to bound above all admissible values (e.g., the total number of products).

By definition, the quality of a given solution x only depends on the number of units shipped to the different clients. Hence the performance vector attached to solution x is simply defined by $f(x) = (f_1(x), \dots, f_n(x))$ where $f_i(x) = \sum_{j=1}^m x_{ij}$ for all $i \in \{1, \dots, n\}$.

Monocriterion preference indices are here aggregated using a Choquet integral ψ_θ with a 2-additive convex capacity θ so as to model possible interactions between any two criteria (e.g., clients who have the same parent company, business partners, or clients located in the same country/city).

Using Proposition 2, we know that solving this MCI problem can be done by solving the following optimization problem:

$$\begin{aligned} \min \quad & g(x, y) = \sum_{i=1}^n \sum_{j=1}^m (y_{ij} g_{ij} - x_{ij} g_{ij}) \\ \text{s.t.} \quad & \left\{ \begin{array}{l} \text{Equations (9a), (9b), (9c) and (9e)} \\ \sum_{i=1}^n y_{ij} \leq b_j, \forall j \in \{1, \dots, m\} \end{array} \right. \quad (12a) \\ & y_{ij} \in \mathbb{N}, \forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, m\} \quad (12b) \end{aligned}$$

where b_j is the capacity of factory F_j so that Equation (12a) corresponds to Equation (5b) the feasibility constraints of the problem.

We now present an explicit example of this transportation problem.

Example 5 Consider an instance of this transportation problem with $n = 3$ three clients (i.e. $\mathcal{C} = \{C_1, C_2, C_3\}$) and $m = 3$ factories (i.e. $\mathcal{F} = \{F_1, F_2, F_3\}$) with the following capacities: $b_1 = 20$, $b_2 = 18$ and $b_3 = 10$. For all $i, j \in \{1, 2, 3\}$, the cost g_{ij} of transporting one unit of product from factory F_j to client C_i is given below:

g_{ij}	F_1	F_2	F_3
C_1	6	7	7
C_2	5	1	3
C_3	5	3	1

Table 10: Delivery costs (given in k€).

	C_1	C_2	C_3
r^0	50	50	50
r^1	17	22	19
r^2	16	14	12
r^3	5	6	6
r^4	-1	-1	-1

Table 11: Multicriteria reference profiles used in the sorting method.

Here the satisfaction of clients only depends on the quantity they receive, and is represented by the reference profiles given in Table 11. These reference profiles define 4 categories: “Very satisfied” (K_1), “Satisfied” (K_2), “Slightly satisfied” (K_3) and “Not satisfied” (K_4). To aggregate monocriterion preference indices, we use the Choquet integral ψ_θ with capacity θ defined as follows:

	\emptyset	$\{1\}$	$\{2\}$	$\{3\}$	$\{1, 2\}$	$\{1, 3\}$	$\{2, 3\}$	$\{1, 2, 3\}$
θ	0	0.2	0.1	0.3	0.4	0.7	0.6	1

Table 12: Capacity θ used in the sorting method.

Let us assume that the current solution x^0 implemented by the decision maker (the delivery company) consists in providing 4 units to C_1 from factory F_1 , 7 units to C_2 from factory F_2 and 7 units to C_3 from factory F_3 (as shown in Figure 3(a)). Its performance vector is $f(x^0) = (4, 7, 7)$, and therefore the resulting preference indices are: $P(x^0, r^0) = P(x^0, r^1) = P(x^0, r^2) = 0$, $P(x^0, r^3) =$

0.6, and $P(x^0, r^4) = 1$. Hence the membership values are: $c_1(x^0) = c_2(x^0) = 0$, $c_3(x^0) = 0.6$ and $c_4(x^0) = 0.4$. As a consequence, solution x^0 enters into category K_3 (“Slightly satisfied”).

Assume first that the decision maker wants to know how much savings could be realized while remaining in the same category. To answer this question, we solve the MGS problem with solution $x = x^0$. We obtain the solution that only differs from x^0 by transporting 3 units from F_1 to C_1 (instead of 4), allowing to save 6 k€.

Assume now that the decision maker wants to know what is the price to pay to improve clients’ satisfaction. To answer this question, we solve the MCI problem with solution $x = x^0$ and we obtain solution x^1 depicted in Figure 3(b), whose performance vector is $f(x^1) = (3, 15, 13)$. This solution belongs to category K_2 (“Satisfied”), as can be derived from its preference indices and membership values: $P(x^1, r^0) = P(x^1, r^1)$, $P(x^1, r^2) = 0.6$, $P(x^1, r^3) = 0.6$, $P(x^1, r^4) = 1$, $c_1(x^1) = 0$, $c_2(x^1) = 0.6$, $c_3(x^1) = 0.4$ and $c_4(x^1) = 0.4$. The cost of moving from x^0 to x^1 is equal to $g(x^0, x^1) = 52 - 38 = 14$ k€.

The decision maker realizes that moving up one level (i.e. to the “Satisfied” category) is not very expensive. Therefore she now wants to know what is the price to pay to move straight to category K_1 (“Very satisfied”). When solving the MCI problem with $x = x^0$ and constraint $c(y) \leq 1$ (instead of Equation (5a)), we obtain solution x^2 depicted in Figure 3(c) whose performance vector is $f(x^2) = (3, 23, 20)$. The cost of moving from x^0 to x^2 is equal to $g(x^0, x^2) = 121 - 38 = 83$ k€.

For the decision maker, paying 83 k€ in one shot is not possible but is willing to delay the objective to the next period. More precisely, she is willing to accept a solution in two steps (one per year), provided that the cost of each step does not exceed 50 k€. Remark that it is not possible for her to first move from solution x^0 to x^1 and then move from solution x^1 to x^2 as the cost $g(x^1, x^2) = 121 - 52 = 69 > 50$ k€. Therefore we want to know whether it is

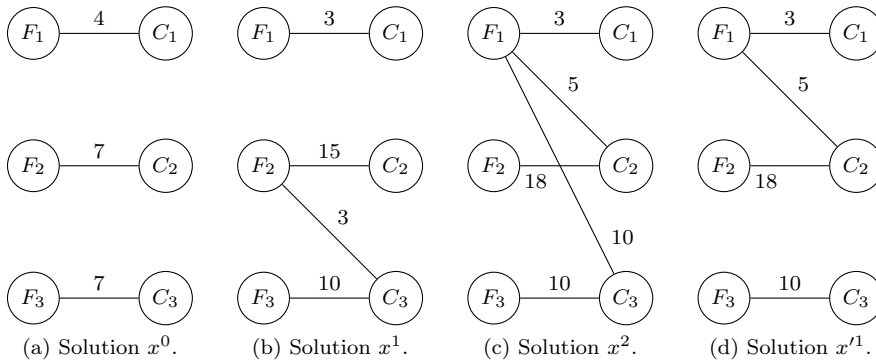


Fig. 3: Solutions found during the interactions with the decision maker.

possible to reach K_1 in two steps, taking into account her budgetary constraints. To answer this question, we solve the MCIS problem with $\ell^* = 1$ (the target category), $h = 2$ (at most 2 steps), and $\beta_1 = \beta_2 = 5$ (the available budgets). We obtain the sequence (x^0, x^1, x^2) where x^1 is the solution belonging to category K_2 (“Satisfied”) which is depicted in Figure 3(d). The total cost is $g(x^0, x^1) + g(x^1, x^2) = 33 + 50 = 83$ k€.

Numerical results. We implemented the presented model using the Gurobi 8.1.1 solver on a computer with 12GB of RAM and an Intel(R) Core(TM) i7 CPU 950 @ 3.07GHz processor. Instances, mathematical programs and initialization algorithms were implemented using python 3.7.5. This model was tested on randomly generated transportation instances where the cost g_{ij} is uniformly drawn in $[0, 1]$ for every client-factory pair (C_i, F_j) , and the capacity b_j is uniformly drawn in $\llbracket 1, 100 \rrbracket$ for every factory F_j . The preference and indifference thresholds are respectively set to 1 and 0 for every criterion, and monocriterion preferences indices are aggregated using randomly generated Choquet integrals with a 2-additive convex capacity and rely on the computational model presented in [29]. Multicriteria reference profiles were chosen to evenly divide the objective space.

In our experiments, we vary n the number of clients (i.e. the number of criteria), m the number of factories and q the number of categories, and the current solution x is generated by solving the transportation problem with smaller capacities. Computation times (in seconds) obtained by averaging over 20 runs are given in Table 13. The results show that the proposed linear formulation enables to efficiently solve instances with a large number of clients and factories.

$n \backslash m$	50	75	100	$n \backslash m$	50	75	100
50	0.95	1.05	1.27	50	0.55	1.02	1.47
75	2.30	5.63	7.12	75	1.27	2.68	5.57
100	4.71	22.11	23.53	100	0.60	4.82	14.13

(a) 4 categories

(b) 9 categories

Table 13: Computation times in the transportation problem.

4.2 An Assignment Problem

Consider an assignment problem where a set of m tasks $T = \{t_1, \dots, t_m\}$ has to be performed by a set of m agents $A = \{a_1, \dots, a_m\}$ in such a way that each agent is assigned to a different task. A feasible solution can be represented by a binary matrix $x = (x_{jk})_{m \times m}$ where $x_{jk} = 1$ if and only if agent a_j is assigned

to task t_k for all $j, k \in \{1, \dots, m\}$. In this problem, the agents' preferences over tasks are represented by utility vectors of type $u^{jk} = (u_1^{jk}, \dots, u_n^{jk}) \in \mathbb{R}_+^n$ where u_i^{jk} is the utility of agent $a_j \in A$ for task $t_k \in T$ with respect to criterion $i \in \{1, \dots, n\}$ (e.g., duration, arduousness, repetitiveness, intellectual stimulation), and the overall performance of a given solution x is simply given by $f(x) = (f_1(x), \dots, f_n(x))$ with $f_i(x) = \sum_{j=1}^m \sum_{k=1}^n x_{jk} u_i^{jk}$ for all criteria $i \in \{1, \dots, n\}$.

Assume that the team leader cares about how happy the agents are with respect to the different criteria, and therefore defines two positive values α_i^1 and α_i^2 for each criterion $i \in \{1, \dots, n\}$ representing the minimum average utility required to be satisfied and slightly satisfied respectively. These values define three categories K_1 (satisfied), K_2 (slightly satisfied), and K_3 (not satisfied), which are characterized by the set $\mathcal{R} = \{r^0, r^1, r^2, r^3\}$ of reference profiles defined as follows:

- $r^\ell = (\alpha_1^\ell, \dots, \alpha_n^\ell)$ for $\ell \in \{1, 2\}$,
- $r^3 = (-1, \dots, -1)$,
- and $r^0 = (M_1, \dots, M_n)$ with M_i large enough to bound above all admissible utility values on criterion $i \in \{1, \dots, n\}$ (e.g., $M_i = \max_{j,k \in \{1, \dots, m\}} u_i^{jk}$).

Then overall preference indices are defined using a Choquet integral ψ_θ with a 2-additive convex capacity to model interactions between some criteria such as duration and arduousness.

Now assume that the team leader wishes to change the current solution in order to obtain a better solution at a minimal cost. Here the cost is defined as follows: for any agent $a_j \in A$, we are given a cost vector $g^j = (g_1^j, \dots, g_m^j) \in \mathbb{R}_+^m$ where g_k^j is the cost incurred when agent a_j is asked to perform task t_k instead of its current assigned task. Then, for any agent $a_j \in A$, the cost $g_j(x, y)$ of moving from a given solution $x = (x_{jk})_{m \times m}$ to another solution $y = (y_{jk})_{m \times m}$ can be formally defined by:

$$g_j(x, y) = \sum_{k=1}^m g_k^j \times \max\{y_{jk} - x_{jk}, 0\}$$

since $\max\{y_{jk} - x_{jk}, 0\}$ is equal to 1 when $y_{jk} = 1$ and $x_{jk} = 0$, and is equal to 0 otherwise. Finally the total cost $g(x, y)$ is given by:

$$\begin{aligned} g(x, y) &= \sum_{j=1}^m g_j(x, y) \\ &= \sum_{j=1}^m \sum_{k=1}^m g_k^j \times \max\{y_{jk} - x_{jk}, 0\} \end{aligned}$$

Using Proposition 2 and a standard linearization of the max aggregator, we obtain the following mixed-integer linear formulation:

$$\begin{aligned}
\min \quad & g(x, y) = \sum_{j=1}^m g_k^j z_{jk} \\
\text{s.t.} \quad & \left\{ \begin{array}{l} \text{Equations (9a), (9b), (9c) and (9e)} \\ \sum_{j=1}^m y_{jk} = 1, \forall k \in \{1, \dots, m\} \quad (13a) \\ \sum_{k=1}^m y_{jk} = 1, \forall j \in \{1, \dots, m\} \quad (13b) \\ z_{jk} \geq y_{jk} - x_{jk}, \forall j \in \{1, \dots, m\}, \forall k \in \{1, \dots, m\} \quad (13c) \\ z_{jk} \geq 0, \forall j \in \{1, \dots, m\}, \forall k \in \{1, \dots, m\} \quad (13d) \\ y_{jk} \in \{0, 1\}, \forall j \in \{1, \dots, m\}, \forall k \in \{1, \dots, m\} \quad (13e) \end{array} \right.
\end{aligned}$$

where Equations (13a) and (13b) correspond to Equation (5b) (the feasibility constraints) while Equations (13c) and (13d) are added so that $z_{jk} = \max\{y_{jk} - x_{jk}, 0\}$ (linearization).

Numerical results. We now present some experimental results obtained for the assignment problem presented in this subsection; the experimental environment is the same as that of the transportation problem, so as the generation methods for thresholds, reference profiles and aggregation functions. The tests were performed on randomly generated instances, where utility vector u^{jk} was uniformly drawn in $\llbracket 1, 100 \rrbracket^n$ for every agent-task pair (a_j, t_k) , and cost g_k^j was uniformly drawn in $[0, 1]$. Then, following a previously fixed permutation of agents, each one has its worst object assigned among the ones remaining. We vary n the number of criteria, m the number of agents (or tasks), and q the number of categories. The results obtained by averaging over 20 runs are given in Table 14 (in seconds); for every run, x is generated by solving the assignment problem obtained by simply aggregating the multi-objective valuations into a single value using a randomly generated weighted sum. We observe that our model is able to solve instances with a large number of tasks, agents and criteria.

$n \backslash m$	75	100	150
3	0.80	1.85	6.74
5	0.91	2.20	8.14
7	3.96	3.99	6.82
10	22.67	70.05	277.35

(a) 4 categories

$n \backslash m$	75	100	150
3	0.79	1.76	6.47
5	1.14	2.10	7.03
7	1.70	3.61	10.73
10	1.79	4.40	13.17

(b) 9 categories

Table 14: Computation times in the assignment problem.

4.3 A Knapsack Problem

Consider a knapsack problem consisting in the selection of W team members from a pool $\mathcal{A} = \{a_1, \dots, a_m\}$ of candidates (e.g., football teams). Any feasible solution can be represented by a binary vector $x = (x_1, \dots, x_m)$ such that $\sum_{j=1}^m x_j = W$, where $x_j = 1$ if and only if candidate a_j is selected. In this selection problem, every candidate $a_j \in \mathcal{A}$ is valued by a vector $u^j = (u_1^j, \dots, u_n^j) \in \mathbb{R}_+^n$ giving the performance of a_j with respect to different criteria (e.g., physical fitness, technique, proper mindset, game intelligence), which are assumed to be additive over the candidates. Hence the performance vector attached to a given team x is simply defined by $f(x) = \sum_{j=1}^m x_j u^j$.

Assume that the recruiter defines two thresholds α_i^1 and α_i^2 for every criterion $i \in \{1, \dots, n\}$ representing the minimum level of performance required to be considered as a high-level and middle-level team respectively. These thresholds define three categories K_1 (high-level), K_2 (middle-level) and K_3 (low-level) which are characterized by the set $\mathcal{R} = \{r^0, r^1, r^2, r^3\}$ of reference profiles defined as follows:

- $r^\ell = (\alpha_1^\ell, \dots, \alpha_n^\ell)$ for $\ell \in \{1, 2\}$,
- $r^3 = (-1, \dots, -1)$,
- and $r^0 = (M_1, \dots, M_n)$ with M_i large enough to bound above all admissible performances on criterion $i \in \{1, \dots, n\}$ (e.g., $M_i = \sum_{j \in \{1, \dots, m\}} u_i^j$).

Then overall preference indices are defined using a Choquet integral ψ_θ with a 2-additive convex capacity to model interactions between pairs of criteria (e.g., technique and game intelligence have a positive interaction). Finally, every candidate $a_j \in \mathcal{A}$ is assumed to have a market price $g_j \in \mathbb{R}^+$ so that the cost $g(x, y)$ of moving from a given team x to another team y can be defined by $g(x, y) = \sum_{j=1}^m g_j(y_j - x_j)$. Thus, the MCI problem can be solved using the following mixed-integer linear program:

$$\begin{aligned} \min_y \quad & g(x, y) = \sum_{j=1}^m g_j(y_j - x_j) \\ \text{s.t.} \quad & \begin{cases} \text{Equations (9a), (9b), (9c) and (9e)} \\ \sum_{j=1}^m y_j = K, \forall j \in \{1, \dots, m\} \end{cases} \quad (14a) \\ & y_j \in \{0, 1\}, \forall j \in \{1, \dots, m\} \end{aligned}$$

where Equation (14a) correspond to Equation (5b) (the feasibility constraints).

Numerical results. We implemented the presented model in the same experimental environment as before, with the same generation methods for thresholds, reference profiles and aggregation functions. The tests were performed on randomly generated knapsack instances with performance vectors u_j uniformly drawn in $\llbracket 1, 100 \rrbracket^n$ and the knapsack capacity W was set to $m/2$. Solution x is generated by picking objects a_j in increasing order according to the overall utility $\sum_{i=1}^n u_{ji}$, until the knapsack capacity is reached. We vary n the number of criteria, m the number of objects, and q the number of categories. We report the computation times (in seconds) obtained by averaging over 20 runs in Table 15. The results show that the proposed mixed-integer linear formulation enables to solve very large instances of the MCI problem specialized for the multicriteria knapsack problem.

$n \backslash m$	100	250	500
3	0.01	0.01	0.03
5	0.02	0.04	0.06
7	0.03	0.04	0.12
10	0.03	0.10	0.17

(a) 4 categories

$n \backslash m$	100	250	500
3	0.004	0.01	0.01
5	0.005	0.01	0.02
7	0.001	0.01	0.02
10	0.007	0.01	0.03

(b) 9 categories

Table 15: Computation times in the knapsack problem.

We conclude this section by considering a particular instance of the multiobjective Knapsack problem in order to illustrate the potential of model \mathcal{P}''_{MCIS} , see Equation (11a), to generate recommendations in terms of min-cost improving sequences reaching the target category.

Example 6 We consider a multicriteria knapsack problem involving a set of 10 objects $\mathcal{O} = \{o_1, \dots, o_{10}\}$ evaluated according to 4 criteria assumed to be additive (i.e., the utility of a subset is defined as the sum of the utilities of its elements). The maximal weight for an admissible solution to the knapsack problem is set to 221. The utilities $f_1(o_k), f_2(o_k), f_3(o_k), f_4(o_k)$ and the weight w_k of object o_k , with $k \in \{1, \dots, 10\}$, are given in the following table:

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
f_1	75	56	36	68	15	79	74	65	86	31
f_2	41	60	56	96	13	2	78	22	1	92
f_3	89	4	11	38	90	28	71	15	9	55
f_4	26	20	54	28	90	58	77	98	93	37
w_k	14	90	56	44	67	23	5	13	61	69

In order to sort the solutions, we consider 6 categories $\mathcal{K} = \{K_1, \dots, K_6\}$ delimited by constant reference profiles $\mathcal{R} = \{r_0, \dots, r_6\}$ such that $f_i(r^0) =$

$375, f_i(r^1) = 313, f_i(r^2) = 250, f_i(r^3) = 187, f_i(r^4) = 124, f_i(r^5) = 62, f_i(r^6) = -1$ for all $i \in \{1, \dots, 4\}$.

The aggregation function used to aggregate preference indices is a Choquet integral ψ_θ where capacity θ is compactly defined by $\theta(A) = \sum_{B \subseteq A} m(B)$ with $m(B) = 0$ for all B except the following focal sets:

B	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1, 2\}$	$\{1, 3\}$	$\{1, 4\}$	$\{2, 3\}$	$\{2, 4\}$	$\{3, 4\}$
$m(B)$	0.174	0.002	0.035	0.159	0.174	0.066	0.098	0.014	0.124	0.154

Assume that the current solution is subset $X_0 = \{o_2, o_3, o_{10}\}$ of utility vector $f(X_0) = (123, 208, 70, 111)$. Given the profiles and the assignment rule, one can easily check that X_0 belongs to category K_5 . In this problem, the decision maker wants to reach category K_1 in at most 3 years under some budget constraints which limit the number of changes (insertion or deletion) per year.

The above example can easily be solved by mathematical programming using \mathcal{P}''_{MCIS} . For illustration purposes, we now give three optimal sequences for three scenarios of constraints limiting the number of changes per year.

- *Scenario 1:* at most 5 changes are allowed per year. The optimal solution is a direct move from X_0 to X_1 where $X_1 = \{o_3, o_4, o_6, o_7, o_8, o_{10}\}$ and $f(X_1) = (353, 346, 218, 352)$, a utility vector belonging to K_1 .
- *Scenario 2:* at most 3 changes are allowed per year. The optimal solution is the sequence (X_0, X'_1, X'_2) where $X'_1 = \{o_3, o_7, o_{10}\}$ and $X'_2 = \{o_1, o_3, o_4, o_7, o_8, o_{10}\}$ whose utility vectors are $f(X'_1) = (141, 226, 137, 168)$ and $f(X'_2) = (349, 286, 279, 320)$ respectively. This sequence allows the decision maker to move first from K_5 to K_4 and then from K_4 to K_1 while satisfying the budget constraints and minimizing the total cost.
- *Scenario 3:* at most 2 changes are allowed per year. The optimal solution is a sequence of solutions $(X_0, X''_1, X''_2, X''_3)$ leading to three moves. The first step is a move from K_5 to K_3 with solution $X''_1 = \{o_2, o_3, o_7, o_{10}\}$ of utility vector $f(X''_1) = (197, 286, 141, 188)$. Then the second step is a move within category K_3 (substitution of o_2 by o_1) leading to solution $X''_2 = \{o_1, o_3, o_7, o_{10}\}$ of utility vector $f(X''_2) = (216, 267, 226, 194)$. Finally the third step is a move from K_3 to K_1 with solution $X''_3 = \{o_1, o_3, o_4, o_7, o_8, o_{10}\}$ of utility vector $f(X''_3) = (349, 385, 279, 320)$.

In all scenarios, solving the MCIS problem yields a sequence reaching the target category at minimal cost under the given constraints. This example provides a good illustration of the possibilities offered by the approach developed in the paper to plan the improvement of a current solution in a multiobjective combinatorial optimization setting.

5 Conclusion

In this paper, we have studied three optimization problems in multicriteria sorting methods on combinatorial domains, namely the min cost improvement, max gain stability and min cost improving sequence problems. We have proposed general mixed-integer (linear or quadratic) formulations, illustrated on the transportation, assignment and knapsack problems. The tests have shown that our approach is very effective in practice in terms of computation times. Note that the proposed approach also applies to continuous sets of alternatives defined by linear constraints leading to simpler problems with even better computation times.

A first direct continuation of this work is to extend the approach to more complex multicriteria sorting methods (e.g., based on assignment rules including veto thresholds, or multiple reference profiles to bound a category). We could also address other combinatorial domains by considering other problems (e.g., the traveling salesman problem). A second direction concerns the development of solution methods when multicriteria reference profiles and/or the aggregation function are imprecisely known and defined by linear constraints.

6 Appendix

Proof of Proposition 1

(\implies) Let $y \in \mathcal{X}$ be such that $c(y) \leq \ell$. Since $c_\ell(y)$ is unimodal when y is fixed, with a maximum at least equal to $1/2$ (see [36]), we know that $c(y) \leq \ell$ implies that there exists $\ell' \leq \ell$ such that $c_{\ell'}(y) \geq 1/2$. Assume that $P(y, r^\ell) < 1/2$. We now prove that $c_{\ell'}(y) < 1/2$ for all $\ell' \leq \ell$ yielding a contradiction.

Consider an index ℓ' such that $\ell' < \ell$. Since $r^{\ell'} \gg r^\ell$ by definition, we have $P_i(y, r^{\ell'}) \leq P_i(y, r^\ell)$ for all criteria $i \in N$. Hence $(P_1(y, r^\ell), \dots, P_n(y, r^\ell)) \succeq (P_1(y, r^{\ell'}), \dots, P_n(y, r^{\ell'}))$. Then, since ψ_θ is compatible with Pareto-dominance, we have $P(y, r^\ell) = \psi_\theta(P_1(y, r^\ell), \dots, P_n(y, r^\ell)) \geq \psi_\theta(P_1(y, r^{\ell'}), \dots, P_n(y, r^{\ell'})) = P(y, r^{\ell'})$. Thus, for all $\ell' \leq \ell$, we have:

$$\begin{aligned} c_{\ell'}(y) &= \min\{P(y, r^{\ell'}), 1 - P(y, r^{\ell'-1})\} && \text{by definition of membership indices} \\ &\leq \min\{P(y, r^\ell), 1 - P(y, r^{\ell'-1})\} && \text{since we have } P(y, r^\ell) \geq P(y, r^{\ell'}) \\ &< \min\{1/2, 1 - P(y, r^{\ell'-1})\} && \text{since } P(y, r^\ell) < 1/2 \text{ by hypothesis} \\ &\leq 1/2 \end{aligned}$$

(\impliedby) Let $y \in \mathcal{X}$ be such that $P(y, r^\ell) \geq 1/2$. Assume that $\ell^* = c(y) > \ell$. Since $c_\ell(y)$ is unimodal with a maximum at least equal to $1/2$, two cases may occur:

- $c_{\ell^*}(y) > 1/2$. In that case, we necessarily have:

$$\begin{aligned} 1/2 &> P(y, r^{\ell^*-1}) && \text{since } c_{\ell^*}(y) = \min\{P(y, r^{\ell^*}), 1 - P(y, r^{\ell^*-1})\} \\ &\geq P(y, r^\ell) && \text{since } \ell^* - 1 \geq \ell \text{ (and therefore } r^{\ell^*-1} \gg r^\ell) \\ &\geq 1/2 && \text{since } P(y, r^\ell) \geq 1/2 \text{ by hypothesis} \end{aligned}$$

yielding a contradiction.

- $c_{\ell^*}(y) = 1/2$. Here we necessarily have $P(y, r^{\ell^*-1}) \leq 1/2$ since $c_{\ell^*}(y) = \min\{P(y, r^{\ell^*}), 1 - P(y, r^{\ell^*-1})\}$ by definition. Thus we obtain:

$$\begin{aligned} c_\ell(y) &= \min\{P(y, r^\ell), 1 - P(y, r^{\ell-1})\} && \text{by definition} \\ &\geq \min\{1/2, 1 - P(y, r^{\ell-1})\} && \text{since } P(y, r^\ell) \geq 1/2 \text{ by hypothesis} \\ &\geq \min\{1/2, 1 - P(y, r^{\ell^*-1})\} && \text{since } \ell^* - 1 \geq \ell - 1 \\ &\geq \min\{1/2, 1/2\} && \text{since } P(y, r^{\ell^*-1}) \leq 1/2 \\ &= 1/2 \\ &= c_{\ell^*}(y) \end{aligned}$$

Hence we have $c_{\ell^*}(y) = c_\ell(y)$ and $\ell^* > \ell$ which contradicts our assignment rule (**). \square

Proof of Proposition 2

In order to establish Proposition 2, we first establish the following Lemma:

Lemma 1 *For any feasible solution of problem (\mathcal{P}''_{MCI}) , we have:*

$$\forall i \in N, p_i \leq P_i(y, r^{c(x)-1})$$

Proof of Lemma 1. Let (y, p, v) be a feasible solution of problem (\mathcal{P}''_{MCI}) . For all $i \in N$, only three cases may occur:

- $f_i(y) - r_i^{c(x)-1} > \gamma_i^+$: In that case, we trivially have $p_i \leq P_i(y, r^{c(x)-1})$ since $P_i(y, r^{c(x)-1}) = 1$ (see Equation (2)) and $p_i \leq 1$ (see Equation (9e)).
- $\gamma_i^- \leq f_i(y) - r_i^{c(x)-1} \leq \gamma_i^+$: Here we have $P_i(y, r^{c(x)-1}) = \frac{f_i(y) - r_i^{c(x)-1} - \gamma_i^-}{\gamma_i^+ - \gamma_i^-}$ by definition (see Equation (2)). If $v_i = 1$, then we derive $p_i \leq P_i(y, r^{c(x)-1})$ from constraint (9b). Otherwise, p_i is necessarily equal to 0 due to constraint (9c) and therefore $p_i = 0 \leq P_i(y, r^{c(x)-1})$ trivially holds.
- $f_i(y) - r_i^{c(x)-1} < \gamma_i^-$: In that case, $P_i(y, r^{c(x)-1}) = 0$ holds by definition (see Equation (2)). Note that variable v_i must be set to 0 to satisfy constraint (9b) since $f_i(y) - r_i^{c(x)-1} - \gamma_i^- < 0$. As a consequence, we necessarily have $p_i = 0$ due to constraint (9c), and therefore $p_i \leq P_i(y, r^{c(x)-1})$ also holds. \square

Using this lemma, we can now establish Proposition 2.

Proof of Proposition 2. (\Leftarrow) Let (y, p, v) be a feasible optimal solution of problem (\mathcal{P}''_{MCI}) . Assume that y is not optimal for (\mathcal{P}'_{MCI}) . In that case, there exists $y' \in \{0, 1\}^m$ such that $Ay' = b$ and $g(x, y') < g(x, y)$. Consider $T(y') = (y', p', v')$ defined as follows:

- $v'_i = 0$ if $P_i(y', r^{c(x)-1}) = 0$ and $v'_1 = 1$ otherwise.
- $p'_i = P_i(y', r^{c(x)-1})$.

It is easy to check that solution $T(y')$ satisfies constraints (9b-9e). Moreover, since y' is a feasible solution of (\mathcal{P}'_{MCI}) , we know that $P(y', r^{c(x)-1}) = \psi_\theta(P_1(y', r^{c(x)-1}), \dots, P_n(y', r^{c(x)-1})) = \psi_\theta(p'_1, \dots, p'_n) \geq 1/2$ due to constraint (8a). Hence $T(y')$ is a feasible solution of (\mathcal{P}''_{MCI}) with $g(x, y') < g(x, y)$, which contradicts the fact that (y, p, v) is optimal for (\mathcal{P}''_{MCI}) .

(\Rightarrow) Let $y \in \{0, 1\}^m$ be a feasible optimal solution of problem (\mathcal{P}'_{MCI}) . Assume that no solution of type (y, p, v) with $p \in [0, 1]^n$ and $v \in \{0, 1\}^n$ is optimal for (\mathcal{P}''_{MCI}) . Consider solution $T(y) = (y, p, v)$ as defined above. Since $T(y)$ is not optimal (by hypothesis), there exists a feasible solution (y', p', v') such that $g(x, y') < g(x, y)$. Moreover, we know that $p'_i \leq P_i(y', r^{c(x)-1})$ for all $i \in N$ (see Lemma 1). As a consequence, constraint (9a) implies constraint (8a), and therefore y' is a feasible solution of (\mathcal{P}'_{MCI}) with $g(x, y') < g(x, y)$, which yields a contradiction since $y \in \mathcal{X}$ is optimal for (\mathcal{P}'_{MCI}) . \square

References

1. Almeida-Dias, J., Figueira, J.R., Roy, B.: A multiple criteria sorting method where each category is characterized by several reference actions: The ELECTRE TRI-nC method. *European Journal of Operational Research* **217**(3), 567–579 (2012)
2. Belacel, N.: Multicriteria assignment method PROAFTN: Methodology and medical application. *European Journal of Operational Research* **125**(1), 175–183 (2000)
3. Benabbou, N., Perny, P., Viappiani, P.: Incremental elicitation of choquet capacities for multicriteria choice, ranking and sorting problems. *Artificial Intelligence* **246**, 152–180 (2017)
4. Bouyssou, D., Marchant, T.: An axiomatic approach to noncompensatory sorting methods in mcdm, i: The case of two categories. *European Journal of Operational Research* **178**(1), 217–245 (2007)
5. Bouyssou, D., Marchant, T.: An axiomatic approach to noncompensatory sorting methods in mcdm, ii: More than two categories. *European Journal of Operational Research* **178**(1), 246–276 (2007)
6. Bouyssou, D., Marchant, T.: On the relations between ELECTRE TRI-B and ELECTRE TRI-C and on a new variant of ELECTRE TRI-B. *European Journal of Operational Research* **242**(1), 201–211 (2015)
7. Chassein, A.B., Goerigk, M.: Alternative formulations for the ordered weighted averaging objective. *Inf. Process. Lett.* **115**(6-8), 604–608 (2015). DOI 10.1016/j.ipl.2015.02.008. URL <https://doi.org/10.1016/j.ipl.2015.02.008>
8. Chateauneuf, A., Dana, R., Tallon, J.M.: Diversification, convex preferences and non-empty core in the Choquet expected utility model. *Economic Theory* **19**(3), 509–523 (1999)
9. Ciomek, K., Ferretti, V., Kadziński, M.: Predictive analytics and disused railways requalification: Insights from a post factum analysis perspective. *Decision Support Systems* **105**, 34–51 (2018)
10. Dias, J.A., Figueira, J.R., Roy, B.: ELECTRE TRI-C : A multiple criteria sorting method based on characteristic reference actions. *European Journal of Operational Research* **204**(3) (2010)
11. Doumpos, M., Zopounidis, C.: *Multicriteria decision aid classification methods*, vol. 73. Springer Science & Business Media (2002)
12. Ehrgott, M.: *Multicriteria optimization*, vol. 491. Springer Science & Business Media (2005)
13. Ehrgott, M., Gandibleux, X.: A survey and annotated bibliography of multiobjective combinatorial optimization. *OR-Spektrum* **22**(4), 425–460 (2000)
14. Fang, L.: Centralized resource allocation based on efficiency analysis for step-by-step improvement paths. *Omega* **51**, 24–28 (2015)
15. Fernández, E., Figueira, J.R., Navarro, J., Roy, B.: ELECTRE TRI-nB: A new multiple criteria ordinal classification method. *European Journal of Operational Research* **263**(1), 214–224 (2017)
16. Fodor, J., Orlovski, S., Perny, P., Roubens, M.: The use of fuzzy preference models in multiple criteria choice, ranking and sorting. In: *Fuzzy sets in decision analysis, operations research and statistics*, pp. 69–101. Springer (1998)
17. Ghahraman, A., Prior, D.: A learning ladder toward efficiency: Proposing network-based stepwise benchmark selection. *Omega* **63**, 83–93 (2016)
18. Grabisch, M.: The application of fuzzy integrals in multicriteria decision making. *European Journal of Operational Research* **89**(3), 445–456 (1996)
19. Grabisch, M.: *Set functions, games and capacities in decision making*. Springer (2016)
20. Grabisch, M., Marichal, J.L., Mesiar, R., Pap, E.: *Aggregation functions*, vol. 127. Cambridge University Press (2009)
21. Greco, S., Matarazzo, B., Slowinski, R.: Fuzzy extension of the rough set approach to multicriteria and multiattribute sorting. In: *Preferences and decisions under incomplete knowledge*, pp. 131–151. Springer (2000)
22. Greco, S., Matarazzo, B., Slowinski, R.: Rough sets methodology for sorting problems in presence of multiple attributes and criteria. *European journal of operational research* **138**(2), 247–259 (2002)

23. Greco, S., Mousseau, V., Słowiński, R.: Multiple criteria sorting with a set of additive value functions. *European Journal of Operational Research* **207**(3), 1455–1470 (2010)
24. Henriët, L.: Systèmes d'évaluation et de classification multicritères pour l'aide à la décision: Construction de modèles et procédures d'affectation. Ph.D. thesis (2000)
25. Jacquet-Lagrez, E.: An application of the UTA discriminant model for the evaluation of r & d projects. In: *Advances in multicriteria analysis*, pp. 203–211. Springer (1995)
26. Kadziński, M., Ciomek, K., Rychły, P., Słowiński, R.: Post factum analysis for robust multiple criteria ranking and sorting. *Journal of Global Optimization* **65**(3), 531–562 (2016)
27. Léger, J., Martel, J.M.: A multicriteria assignment procedure for a nominal sorting problematic. *European Journal of Operational Research* **138**(2), 349–364 (2002)
28. Leroy, A., Mousseau, V., Pirlot, M.: Learning the parameters of a multiple criteria sorting method. In: *International Conference on Algorithmic Decision Theory*, pp. 219–233. Springer (2011)
29. Lesca, J., Perny, P.: LP solvable models for multiagent fair allocation problems. In: *ECAI 2010 - 19th European Conference on Artificial Intelligence*, Lisbon, Portugal, August 16-20, 2010, Proceedings, pp. 393–398 (2010)
30. Marichal, J.L., Meyer, P., Roubens, M.: Sorting multi-attribute alternatives: The tomaso method. *Computers & Operations Research* **32**(4), 861–877 (2005)
31. Massaglia, R., Ostanello, A.: N-tomic: a support system for multicriteria segmentation problems. *Multiple Criteria Decision Support* **356**, 167–174 (1991)
32. Moscarola, J., Roy, B.: Procédure automatique d'examen de dossiers fondée sur une segmentation trichotomique en présence de critères multiples. *RAIRO-Operations Research* **11**(2), 145–173 (1977)
33. Mousseau, V., Özpeynirci, Ö., Özpeynirci, S.: Inverse multiple criteria sorting problem. *Annals of Operations Research* **267**(1-2), 379–412 (2018)
34. Ogryczak, W., Słowiński, T.: On solving linear programs with the ordered weighted averaging objective. *European Journal of Operational Research* **148**(1), 80–91 (2003)
35. Ogryczak, W., Słowiński, T.: On efficient WOWA optimization for decision support under risk. *Int. J. Approx. Reasoning* **50**(6), 915–928 (2009)
36. Perny, P.: Multicriteria filtering methods based on concordance and non-discordance principles. *Annals of operations Research* **80**, 137–165 (1998)
37. Petrović, M., Bojković, N., Anić, I., Stamenković, M., Tarle, S.P.: An ELECTRE-based decision aid tool for stepwise benchmarking: An application over eu digital agenda targets. *Decision Support Systems* **59**, 230–241 (2014)
38. Petrović, M., Bojković, N., Stamenković, M., Anić, I.: Supporting performance appraisal in ELECTRE based stepwise benchmarking model. *Omega* **78**, 237–251 (2018)
39. Roy, B.: *Multicriteria methodology for decision aiding*, vol. 12. Springer Science & Business Media (1996)
40. Słowiński, R., Greco, S., Matarazzo, B.: Axiomatization of utility, outranking and decision rule preference models for multiple-criteria classification problems under partial inconsistency with the dominance principle. *Control and Cybernetics* **31**(4), 1005–1035 (2002)
41. Sobrie, O., Mousseau, V., Pirlot, M.: Learning a majority rule model from large sets of assignment examples. In: *International Conference on Algorithmic Decision Theory*, pp. 336–350. Springer (2013)
42. Yu, W.: Aide multicritère à la décision dans le cadre de la problématique du tri: concepts, méthodes et applications. Ph.D. thesis, Université Paris IX-Dauphine (1992)
43. Zheng, J., Takougang, S.A.M., Mousseau, V., Pirlot, M.: Learning criteria weights of an optimistic ELECTRE TRI sorting rule. *Computers & Operations Research* **49**, 28–40 (2014)
44. Zopounidis, C., Doumpos, M.: Multicriteria classification and sorting methods: A literature review. *European Journal of Operational Research* **138**(2), 229–246 (2002)