Information Sciences 580 (2021) 734–755

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Online multi-criteria portfolio analysis through compromise programming models built on the underlying principles of fuzzy outranking

Gilberto Rivera [a], Rogelio Florencia [a,*], Mario Guerrero [a], Raúl Porras [a], J. Patricia Sánchez-Solís [a]

[a] Universidad Autónoma de Ciudad Juárez, División Multidisciplinaria de Ciudad Universitaria, Av. José de Jesús Macías Delgado #18100, Cd. Juárez, Chihuahua, 32000, Mexico

ARTICLE INFO

Article history:
Received 2 July 2020
Received in revised form 21 August 2021
Accepted 24 August 2021
Available online 27 August 2021

Keywords:
Compromise programming
Multi-criteria project portfolio analysis
Many-objective optimisation
Fuzzy outranking methods
Real-world case study

ABSTRACT

This paper introduces an interactive approach to support multi-criteria decision analysis of project portfolios. In high-scale strategic decision domains, scientific studies suggest that the Decision Maker (DM) can find help by using many-objective optimisation methods, which are supposed to provide values in the decision variables that generate high-quality solutions. Even so, DMs usually wish to explore the possibility of reaching some levels of benefits in some objectives. Consequently, they should repeatedly run the optimisation method. However, this approach cannot perform well – in an interactive way – for large instances under the presence of many objective functions. We present a mathematical model that is based on compromise programming and fuzzy outranking to aid DMs in analysing multi-criteria project portfolios on the fly. This approach allows relaxing the problem of rapidly optimising portfolios while preserving the beneficial properties of the DM's preferences expressed by outranking relations. Our model supports the decision analysis on two instance benchmarks: for the first one, a better compromise solution was generated 84% of the runs; for the second one, this ranged from 93% to 97%. Our model was also applied to a real-world problem involving social projects, obtaining satisfactory results.

1. Introduction

Private and public organisations systematically seek to generate wealth via supporting projects at a strategic level. Unfortunately, resources are scarce and they must often make 'go/no-go' decisions on financing projects. In this context, a portfolio is a set of projects receiving support, which are picked from a larger pool of candidate project proposals. Such a portfolio must be in line with the goals pursued by the organisation as well as consider constraints related to each specific application domain (e.g. budgetary, political, technical, and equity constraints).

In this connection, the Decision Maker (DM) is the entity – composed of a single person or a group of people – whose responsibility is to identify the best portfolio. However, this decision is difficult due to the presence of a large number of solutions because of the combinatorial nature of the problem [cf. 1]; thus, approximate algorithms are quite popular to solve portfolio problems. Additionally, having many objectives to optimise implies that these algorithms should take into

* Corresponding author.
E-mail address: rogelio.florencia@uacj.mx (R. Florencia).

https://doi.org/10.1016/j.ins.2021.08.087
0020-0255/

consideration the preferences of the DM so as to have the ability to identify the best compromise between multiple criteria (selective pressure) [cf. 2]. This has fostered the development of multi-criteria decision-aiding models to find the best compromise solution (i.e. the portfolio whose return is optimal in terms of the organisation's objectives according to the system of the preferences of the DM). Other challenges to consider are interdependencies between the projects (e.g. synergy, redundancy, and time-based dependencies) [cf. 3,4] and vagueness and uncertainty in resource requirements, benefit return, and time to complete the projects [cf. 5,6], as well as incomplete information about the preferences [e.g. 7].

In the specialised literature, a large number of recent scientific studies have focused on solving multi-criteria portfolio problems, not only due to the importance of this problem but also because it is still an open field of research [cf. 8–12]. This growing interest is a consequence of the fact that finding an efficient way to treat project portfolio problems is a matter of concern common to industries, agencies and institutions in many different domains and competencies because there is a wide range of application cases [e.g. 13–18].

Dedicated solution frameworks for this kind of problem need to jointly apply linear and non-linear mathematical modelling, preference elicitation, multi-criteria decision theory, (exact or approximate) vector optimisation, and interactive recommendation software design [cf. 19].

In this paper, we propose a framework that addresses interactive decision analysis on multi-criteria project portfolios. The main contributions of this proposal are the following:

(a) it is a technique based on mathematical programming that is enriched with preferential information, which is taken from the parameters of ELECTRE III;
(b) it is especially suitable for large-scale instances under the presence of many objective functions; and
(c) compared to existing research in the scientific literature, it shows advantages in terms of both quality and runtime.

Our experimental results provide evidence that this proposal can be considered as an effective software tool for interactively analysing multi-criteria decisions on portfolios.

The structure of this paper is as follows. In Section 2, we discuss the background, the key characteristics of the problem, and the related studies. In Section 3, we present the mathematical model based on fuzzy preference relations. Section 4 documents our computational experiments, including both synthetic instances and a real-world case study. Lastly, in Section 5, we discuss some insights and make some recommendations for future research.

2. Background

Portfolio consequences are usually described by multiple attributes that are related to the organisational strategy. Therefore, a vector $z(x) = \langle z_1(x),\ z_2(x),\ z_3(x), \ldots,\ z_p(x)\rangle$ is associated with the consequences of a portfolio x considering p criteria. In the simplest case, $z(x)$ is the cumulative sum of the benefits of the selected projects. However, in the presence of synergy, it is necessary to consider the contributions of interdependent project groups. Without loss of generality, we can assume that larger values of the objective functions are preferred to smaller ones. The following expression represents the generic solution to this problem:

$$\max_{x \in R_F} \left\{ \left\langle z_1(x),\ z_2(x), \ldots,\ z_p(x) \right\rangle \right\}, \tag{1}$$

where R_F is the space of feasible portfolios, which is usually determined by the available budget. Solving Problem 1 means finding the best compromise solution according to the system of preferences and the values of the DM.

The idea of incorporating the fuzzy outranking relations of ELECTRE into metaheuristics for many-objective portfolio optimisation to solve Problem 1 has been broadly studied. The pioneers of this strategy were Fernandez et al. [20], who subsequently inspired a wide range of studies in the last decade exploiting the properties of outranking relations [e.g. 6,21–24,5,4,25]. These studies provide evidence that metaheuristics increase their selective pressure when they are enriched with the DM's preferences as articulated through ELECTRE III. Consequently, they perform better than Pareto-based metaheuristics when many-objective problems are treated.

The basis of the original idea is the relational system of preferences described by Roy in [26]. A crucial model is $\sigma(x, y)$, which is the fuzzy value of the proposition 'x is at least as good as y', and is calculated by methods from the literature [e.g. 27–29]. ELECTRE defines $\sigma(x, y)$ considering.

- the concordance index, $c(x, y)$, which measures the strength of the coalition of criteria in favour of 'x is at least as good as y'; and
- the discordance index, $d(x, y)$, which measures the strength of the criteria invalidating the statement 'x is at least as good as y'.

To estimate the concordance index, it is necessary to know how the DM perceives the criteria and their values. This calculation requires the following parameters:

Weight vector: This represents how important each of the objectives is to the DM and is denoted by the vector $W = \langle w_1, w_2, w_3, \ldots, w_p \rangle$, where $w_k > 0 \; \forall k \in \{1, 2, 3, \ldots, p\}$, where p is the number of objectives and $\sum_{k=1}^{p} w_k = 1$. Usually, the DM will not be able to establish the value of each w_k, but they can use methods such as Swing and Smart [30,31] for this task.

Indifference threshold: This indicates how small the differences – in terms of the values of the objectives – should be for the DM to consider them as marginal or not significant on a practical level. Here, $U = \langle u_1, u_2, u_3, \ldots, u_p \rangle$ represents the indifference thresholds, where u_k is the threshold for the kth criterion.

The concordance index is calculated as:

$$c(x,y) = \sum_{k=1}^{p} c_k(x,y), \tag{2}$$

where

$$c_k(x,y) = \begin{cases} w_k & \text{if } x\mathrm{P}_k y \vee x\mathrm{I}_k y, \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

where $x\mathrm{P}_k y$ and $x\mathrm{I}_k y$ are the fuzzy logical relations of preference and indifference, respectively, when evaluating the kth objective. Preference is defined by

$$x\mathrm{P}_k y = z_k(x) > z_k(y) \; \wedge \; \neg x\mathrm{I}_k y, \tag{4}$$

where z_k is the evaluation function for the kth objective. Indifference is defined by

$$x\mathrm{I}_k y = |f_k(x) - f_k(y)| \leq u_k, \tag{5}$$

where u_k is the indifference threshold for the kth objective.

The discordance index is calculated based on two sets of parameters:

Pre-veto threshold: This is denoted by the vector $S = \langle s_1, s_2, s_3, \ldots, s_p \rangle$, indicating how large the differences in the objectives should be for the DM to consider them as relevant. Here, $s_k \geq u_k \; \forall k \in \{1, 2, 3, \ldots, p\}$.

Veto threshold: This is represented by the vector $V = \langle v_1, v_2, v_3, \ldots, v_p \rangle$ and indicates the magnitude of the differences (in the objectives) between two alternatives needed to trigger a veto condition, with $v_k \geq s_k \; \forall k \in \{1, 2, 3, \ldots, p\}$.

Consequently, $d(x,y)$ can be defined as

$$d(x,y) = \min_{k \in \{1, 2, 3, \ldots, p\}} \{1 - d_k(x,y)\} \tag{6}$$

where $d_k(x,y)$ is defined by the equation

$$d_k(x,y) = \begin{cases} 0 & \text{if } \nabla_k(x,y) < s_k, \\ \frac{\nabla_k(x,y) - s_k}{v_i - s_k} & \text{if } s_k \leqslant \nabla_k(x,y) < v_k, \\ 1 & \text{otherwise}, \end{cases} \tag{7}$$

where v_k and s_k are the veto and pre-veto thresholds, respectively, and $\nabla_k(x,y) = f_k(y) - f_k(x)$. The discordance introduces the following effect of rejection: if there is a difference from x (according to the kth criterion) that exceeds v_k, then the predicate 'x is at least as good as y' is denied, regardless of the concordance index. This non-compensatory property is the most distinctive feature of ELECTRE III: high returns in some criteria do not justify overwhelming losses in others. Not all the criteria require a veto threshold; and those criteria that do require it can vary in intensity (this phenomenon is modelled by the values assigned to v_k and s_k).

Finally, $\sigma(x,y)$ is calculated as

$$\sigma(x,y) = c(x,y) \cdot d(x,y). \tag{8}$$

Considering the parameters λ, β, and ϵ ($0 \leq \epsilon \leq \beta \leq \lambda$ and $\lambda > 0.5$), the relational system of preferences [32] identifies one of the following relations for each pair of portfolios (x,y):

Strict preference: This is associated with conditions in which the DM has clear and well-defined reasons justifying the choice of one alternative to another and is denoted by $x\mathrm{P}y$, which represents the situation when the DM significantly prefers x. This is defined in Eq. 9, where dominance refers to the well-known Pareto efficiency.

$$\mathrm{P} = \{(x,y) : x \text{ dominates } y \vee \sigma(x,y) \geqslant \lambda \wedge \sigma(y,x) < 0.5 \vee \sigma(x,y) \geqslant \lambda \wedge 0.5 \leqslant \sigma(y,x) < \lambda \wedge \sigma(x,y) - \sigma(y,x) \geqslant \beta\} \tag{9}$$

Indifference: This corresponds to the existence of clear and positive reasons that justify equivalence between the two options. From the DM's perspective, the alternatives x and y have a high degree of equivalence, so they cannot state that one is preferred to another. This relation, denoted by $x\mathrm{I}y$, is defined in Eq. 10 in terms of $\sigma(x,y)$.

$$\mathrm{I} = \{(x,y)\ :\ \sigma(x,y) \geq \lambda\ \wedge\ \sigma(y,x) \geq \lambda\ \wedge\ |\sigma(x,y) - \sigma(y,x)| \leq \epsilon\} \tag{10}$$

Weak preference: This relation can be considered as the first 'weakened' form of the strict preference. It models a state of doubt between $x\mathrm{P}y$ and $x\mathrm{I}y$. It can be defined as expressed by Eq. 11, where it is represented by $x\mathrm{Q}y$.

$$\mathrm{Q} = \{(x,y)\ :\ \sigma(x,y) \geq \lambda\ \wedge\ \sigma(x,y) \geq \sigma(y,x)\ \wedge\ \neg x\mathrm{P}y\ \wedge\ \neg x\mathrm{I}y\} \tag{11}$$

Incomparability: This represents the situation when the DM cannot (or does not want to) express a preference because, from the point of view of the DM, there is a high heterogeneity between the alternatives x and y. This relation, denoted by $x\mathrm{R}y$, is expressed in terms of $\sigma(x,y)$ in Eq. 12.

$$\mathrm{R} = \{(x,y)\ :\ \sigma(x,y) < 0.5\ \wedge\ \sigma(y,x) < 0.5\} \tag{12}$$

***k*-Preference**: This is the second 'weakened' form of strict preference. k-Preference represents a state of doubt between $x\mathrm{P}y$ and $x\mathrm{R}y$, and is denoted by $x\mathrm{K}y$. Eq. 13 defines this relation in terms of $\sigma(x,y)$.

$$\mathrm{K} = \left\{(x,y)\ :\ 0.5 \leq \sigma(x,y) \leq \lambda\ \wedge\ \sigma(y,x) < 0.5\ \wedge\ \sigma(x,y) - \sigma(y,x) > {}^{\beta}/_{2}\right\} \tag{13}$$

In addition to these relations, the net flow score is also used to identify the DM's preferences for solutions [33]. It can be defined as

$$F_n(x) = \sum_{y \in O \setminus \{x\}} [\sigma(x,y) - \sigma(y,x)], \tag{14}$$

where O is a set of feasible portfolios, and x and y are feasible solutions. Note that $F_n(x) > F_n(y)$ indicates a preference for x to y. From the set O, the preference-based system defines the sets presented in Table 1.

The best portfolio that is compatible with the fuzzy outranking relation σ should be a non-strictly outranked solution that is simultaneously a non-dominated solution to the problem:

$$\min_{x \in O} \{\langle |S(O,x)|,\ |W(O,x)|,\ |F(O,x)| \rangle\}. \tag{15}$$

As a consequence of the last remark [33], the best portfolio can be found through a lexicographic search, with a pre-emptive priority favouring $|S(O,x)|$.

This three-objective problem is a map of any multi-objective problem in terms of preferences. When the DM is confident of the preference model, then they should accept that the best compromise is a non-dominated solution of Problem 15. It is also interesting that the mapped three-objective problem is valid independently of the dimension of the original multi-objective space. There is evidence that this property is very important in solving portfolio problems with many objective functions [25,23,24,6].

However, the parameters of the model need to be adjusted according to the specific features of the decision problem and the DM's preferences to this approach works well. Thus, the DM should assess the parameters included in the calculation of σ (criterion weights and thresholds) and the system of preferences (λ, β and ϵ). This task can be done by an interaction between the DM and a Decision Analyst (DA), using, if necessary, indirect elicitation methods to support them [e.g. 34,35]. In the literature on outranking methods, this task has been recently approached using examples about which the DM has previously expressed their preferences [e.g. 36,24].

Here, one of the most challenging issues is caused by the fact that the DM does not have precise knowledge of the multi-objective return of the optimised portfolios (obtained after some optimisation process) at the beginning of this process.

Table 1
Sets from the relational system of preferences.

Set	Definition
Solutions strictly outranking x	$S(O,x) = \{y \in O\ :\ y\mathrm{P}x\}$
The non-strictly outranked frontier	$NS(O) = \{x \in O\ :\ S(O,x) = \varnothing\}$
Solutions weakly outranking x	$W(O,x) = \{y \in NS(O)\ :\ y\mathrm{Q}x\ \wedge\ y\mathrm{K}x\}$
The non-weakly-outranked frontier	$NW(O) = \{x \in O\ :\ W(O,x) = \varnothing\}$
Solutions that are greater than x in net flow score	$F(O,x) = \{y \in NS(O)\ :\ F_n(y) > F_n(x)\}$
The net-flow non-outranked frontier	$NF(O) = \{x \in NS(O)\ :\ F(O,x) = \varnothing\}$

Therefore, by analysing the ranges of the optimised objective functions, the DM gets a clearer idea of what they could obtain. Consequently, the DM often wants to get a portfolio according to that updated notion of the best compromise. However, most of the existing approaches do not provide means to enrich the optimisation algorithm or incorporate updates of the DM's preference system. Instead, these approaches assume that the DM can run the optimisation model again and again every time they want to explore the possibility of obtaining a project portfolio according to their latest expression of preferences [e.g. 6,22]. However, when faced with many objectives and projects, this process could be inadequate when time is an important concern because:

- each single run of those optimisation algorithms usually takes minutes (even hours) to reach convergence for large instances;
- the parameters of the preference model need to be updated to reflect this new notion of the best compromise. The DM should then be prepared to compare pairs of solutions (a sufficient number) to calculate the new values for the parameters, which is often a time consuming task; and
- there is no clear notion of how many interactions of this kind the DM shall require to feel satisfied with the solution.

It is therefore questionable whether one can perform a portfolio analysis in an online and interactive fashion by only running the algorithms repeatedly. To provide an alternative for the cases in which these issues are a crucial concern, we propose a framework based on the principles of the theory of fuzzy outranking, which are embedded into a compromise programming model.

3. Our proposal

This approach is based on the idea of enriching a classical mathematical technique (e.g. goal programming, ϵ-constraint, or compromise programming) by incorporating some concepts based on fuzzy relations, taken from the European School of Multi-Criteria Decision Analysis (MCDA). This framework consists of the three stages, as explained in the following subsections.

Phase 1. Getting an initial good compromise

The objective here is to find a solution which is a good approximation to the best compromise. Such a solution, hence, belongs to the true Pareto frontier – or, at least, is acceptably close to it – and satisfies, arguably, the latest expression of the DM's preferences among all known solutions.

There are several approaches in the specialised literature that could be useful to accomplish this task, ranging from *a posteriori* approaches [e.g. 37,38] to *a priori* approaches [e.g. 39,40], and including techniques ranging from mathematical programming [e.g. 41,42] to soft computing [cf. 43].

On the one hand, *a priori* techniques aim to provide a reduced set of solutions – called the Region of Interest (RoI) – that match the DM's preferences. On the other hand, the *a posteriori* techniques are intended to get a representative and uniformly distributed sample of the Pareto frontier, as expressed in Problem 1. Regardless of the method employed, a solution x must be identified as the best compromise, whose benefits are characterised by the vector function $z(x) = \{z_1(x),\ z_2(x),\ z_3(x), \dots,\ z_p(x)\}$, where $z_k(x)$ is the return in the kth objective function in a problem with p criteria to be optimised (see Eq. 1).

With this objective in mind, we suggest that an optimisation method based on a relational system of preferences to identify x should be applied during this stage [e.g. 5,4,23,6], as described in Table 1. Under this scheme, the RoI is composed of the non-dominated solutions to Problem 15, and the best compromise is a solution x belonging to the RoI. Fig. 1 depicts our approach, including Phase 1, which can be consulted for a better understanding.

Phase 2. Searching for an advantageous trade-off within the indifference thresholds

In this phase, the framework builds an optimisation model to generate a new solution that offers a better compromise according to the system of the DM's preferences. Here, the DM has to indicate which criteria have the highest priority and to suggest an intended increment for each of them. Let us denote the set of objectives that have been prioritised as $\mathbb{P}$, and the desired enhancement of the kth objective as g_k, subject to $g_k > u_k\ \forall k \in \mathbb{P}$, where u_k is the indifference threshold for the kth criterion. The aim is to improve these objectives by reaching the planned goals but without decreasing the other ones beyond the respective indifference thresholds. Let us suppose given a portfolio x' that meets these conditions. Clearly, x' is at least as good as x according to ELECTRE III because $\sigma(x', x) = 1$ and $\sigma(x', x) \geq \sigma(x, x')$ according to Eq. 2. Compromise programming is intrinsically adequate to generate such a solution x'.

Compromise programming works by defining an aspiration point (known as the 'ideal point' within that method). If p is the number of objective functions, then we can define each component of the aspiration point $\mathbb{A} = \langle a_1,\ a_2,\ a_3, \dots,\ a_p \rangle$ by

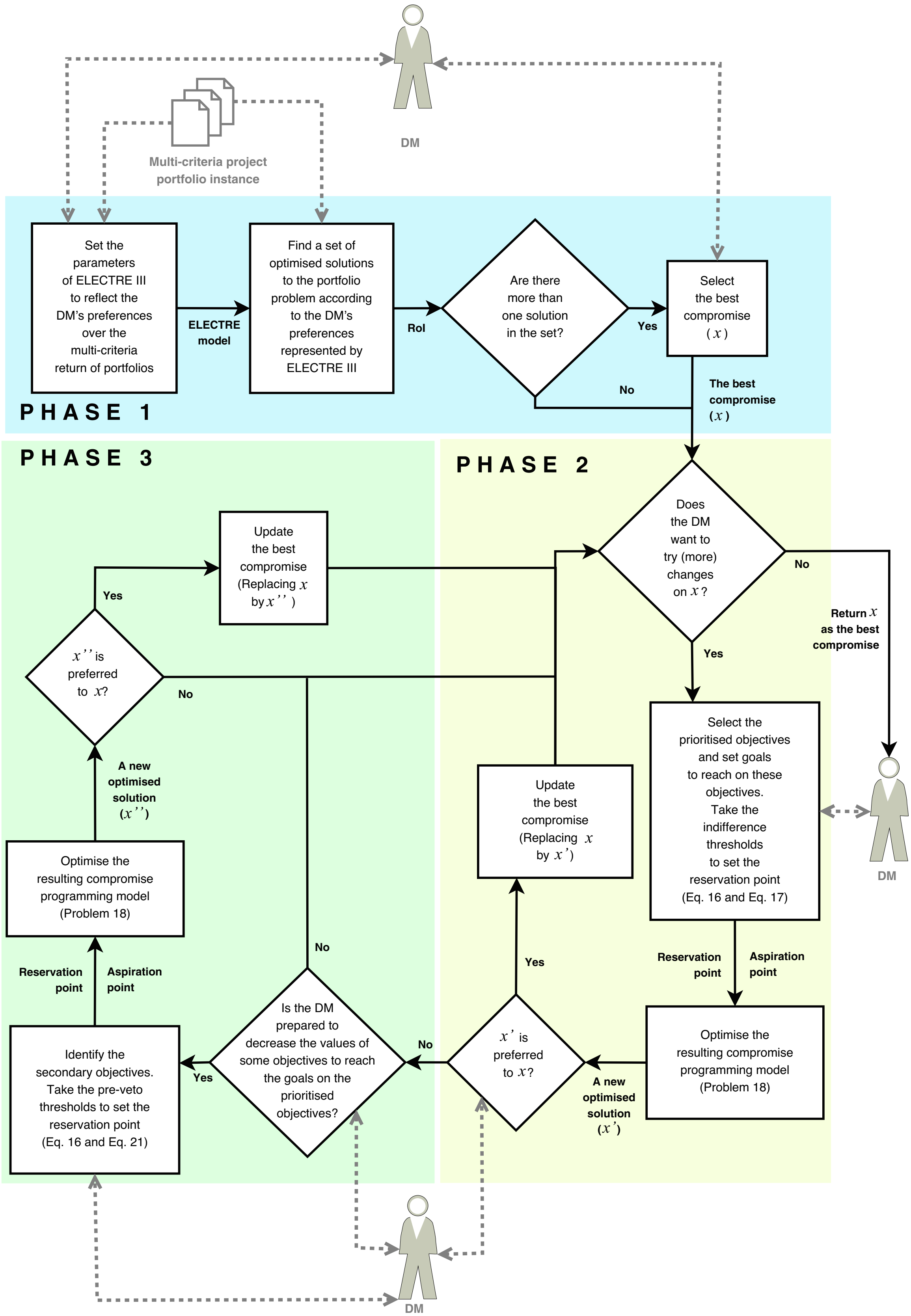

Fig. 1. Three-phased framework for our online multi-criteria portfolio analysis.

$$a_k = \begin{cases} z_k(x) + g_k & \text{if } k \in \mathbb{P}, \\ z_k(x) & \text{otherwise}, \end{cases} \tag{16}$$

where $z_k(x)$ is the value of the kth objective of the portfolio x. Moreover, a reservation point is required, which is represented by $\mathbb{R} = \langle r_1,\ r_2,\ r_3, \ldots,\ r_p \rangle$ and is defined by

$$r_k = \begin{cases} z_k(x) & \text{if } k \in \mathbb{P}, \\ z_k(x) - u_k & \text{otherwise}. \end{cases} \tag{17}$$

Here, $\mathbb{R}$ delimits the search space through the constraints $z_k(x') > r_k\ \forall k \in \{1,\ 2,\ 3, \ldots,\ p\}$. Therefore, a new optimisation problem can be expressed by

$$\min_{x' \in R_F} \delta(x'), \tag{18}$$

where R_F is the region of feasible portfolios, and $\delta(x')$ is the well-known similarity measure based on the distance between two p-dimensional points (in this case the distance to the aspiration point), which is expressed by

$$\delta(x') = \sum_{k=1}^{p} \delta_k(x'), \tag{19}$$

where

$$\delta_k(x') = \left| \frac{a_k - z_k(x')}{a_k - r_k} \right|. \tag{20}$$

By this means, the compromise programming method may reformulate the original p-objective problem as a mono-objective problem which is in line with the DM's preferences in terms of the outranking relations. Here, we employed the Branch & Cut method implemented in CPlex® 12.5 to solve Problem 18. As a matter of fact, any solution x' to Problem 18 is not inferior to x (there is only one of the relations $x'\mathrm{P}x,\ x'\mathrm{Q}x$ and $x'\mathrm{I}x$ that is true).

Then, the DM has to compare the multi-criteria returns of x and x'. This decision should be fast and easy to make because the non-prioritised objectives have differences that are less than the indifference thresholds, and the differences in the objectives with higher priorities are only in favour of x'. Consequently, there is no compensatory effect to analyse. If the DM does not prefer x' to x, our framework suggests a subsequent phase to carry out (See Fig. 1 for a visual aid for Phase 2).

Phase 3. Searching for an advantageous trade-off beyond indifference thresholds

If the DM insists on improving the prioritised objectives, then the framework provides a second optimisation alternative. However, the DM would have to be prepared to accept losses in some objectives. Let us denote the set of the so-called 'secondary' criteria by $\mathbb{S}$. To favour finding a new portfolio that outranks x, consider for this stage the following conditions (see Eq. 8):

(a) tolerable losses on each objective $k \in \mathbb{S}$ have to be less than or equal to the pre-veto threshold s_k;
(b) regarding the increase in the prioritised objectives, consider $g_k > u_k\ \forall k \in \mathbb{P}$; and
(c) about the balance between the prioritised objectives and the secondary ones, considering the weights of the criteria, then $\sum_{k \in \mathbb{P}} w_k > \sum_{k \in \mathbb{S}} w_k$, where w_k is the weight of the kth objective function.

If the outranking model is well tuned to actually reflect the DM's preferences, then these conditions are likely to be desirable and so they are only a guideline to generate non-outranked solutions. To build a compromise model fulfilling such conditions, Eq. 16 is useful to define the aspiration point; conversely, each component of the reservation vector needs to be redefined as

$$r_k = \begin{cases} z_k(x) & \text{if } k \in \mathbb{P}, \\ z_k(x) - s_k & \text{if } k \in \mathbb{S}, \\ z_k(x) - u_k & \text{otherwise}. \end{cases} \tag{21}$$

The next step is to optimise according to Problem 18 and obtain a new solution x'' as the resulting portfolio. Although the model does not guarantee the existence of an ideal x'' with $\delta(x'') = 0$, it does prescribe a solution that is not inferior to x in terms of both Pareto dominance and outranking relations. Additionally, $\sigma(x'', x) \geq \sigma(x, x'')$ is always true.

It is also important to highlight the limits of the human mind when not-inferior alternatives – evaluated in many criteria – are compared. The suggestion is to consider $|\mathbb{P}\ \cup\ \mathbb{S}| \leq 7$ [cf. 44]. Consequently, the DM would have to assess the compensatory effects on up to seven criteria at the same time (because the values of the remaining objective functions only vary within indifference thresholds). In these circumstances, the DM can compare x and x'' and legitimately select the better one. Regardless of their decision, the DM can apply our approach repeatedly until:

(a) they are satisfied with the multi-criteria return reached in each objective; or
(b) although they want greater benefits in the prioritised objectives, such a solution is not reachable by this approach (perhaps, such a portfolio provokes inadvisable losses in other objectives or is even unfeasible).

Fig. 1 shows a diagram of our three-phased proposal.

4. Experimental results

Our proposal was programmed in Java, using the JDK 1.6 and Netbeans 7.4. We used the libraries provided by CPlex® 12.5 and – by this means – we applied the Branch & Cut method to the solution of the programming model that results after each interaction with the user. This procedure was performed on a Mac Pro computer with an Intel Quad processor at 2.8 GHz and 3 GB of RAM. To verify the appropriateness of our proposal, we have conducted a wide series of experiments. The rest of this section is structured as follows. Section 4.1 gives an example of how to apply this approach by addressing a particular synthetic instance reported in the literature. Sections 4.2 and 4.3 address many-objective portfolio problems of different sizes (in terms of the number of projects and objectives) solved in Phase 1 by two leading optimisation *a priori* approaches: an Ant Colony Optimisation (ACO) algorithm [1] and a Genetic Algorithm (GA) [22]. Section 4.4 presents the results when an *a posteriori* approach – a decomposition-based evolutionary algorithm [45] – is used in Phase 1, emphasising the advantages of applying our proposal. Lastly, Section 4.5 presents the application of our framework to a real-world case study.

4.1. A numeric example

To illustrate how to apply our proposal, let us consider one of the multi-criteria portfolio instances solved in [1], whose characteristics are.

(a) Nine objective functions to be maximised ($p = 9$).
(b) One hundred projects ($n = 100$) that were considered as acceptable according to a set of experts.
(c) Six situations of redundant projects (mutually-exclusive projects).
(d) Twenty-four synergetic relations among projects impacting the multi-objective return of the portfolio.

The values of the parameters that define the outranking relations of ELECTRE III were taken from the original source [1] and we shall assume from here on that those values are acceptably adjusted, so that the model actually reflects the preferences of the DM over the expected benefits from the portfolios, which are:

- the weight vector $W = \langle 0.1,\ 0.17,\ 0.06,\ 0.12,\ 0.07,\ 0.13,\ 0.09,\ 0.08,\ 0.18 \rangle$,
- the indifference thresholds $U = \langle 27750,\ 11225,\ 47535,\ 21135,\ 73955,\ 13655,\ 44060,\ 46630,\ 18015 \rangle$,
- the pre-veto thresholds $S = \langle 111015,\ 50510,\ 150535,\ 79255,\ 203385,\ 68295,\ 154220,\ 163215,\ 99080 \rangle$,
- the veto thresholds $V = \langle 166525,\ 78575,\ 205995,\ 116245,\ 258850,\ 109275,\ 220315,\ 233165,\ 162135 \rangle$, and
- the parameters defining the outranking relations, $\lambda = 0.67,\ \epsilon = 0.10$ and $\beta = 0.20$.

In Phase 1 we used an ACO algorithm with *a priori* preference articulation [1] and an initial good compromise x is obtained, whose values on the multi-objective function are

$$z(x) = \langle 1387735,\ 1122535,\ 1584605,\ 1056785,\ 1848960,\ 1365945,\ 2203195,\ 1554445,\ 1801540 \rangle.$$

At this point, we want to explore the possibility of generating a better solution that enhances the values of the objective functions according to the latest representation of the preferences of the DM.

Thus, it is necessary for the DM to indicate the prioritised objectives and their levels of aspiration. Our approach will be applied to improve the values of the prioritised objectives, diminishing those of the non-prioritised objectives up to their indifference thresholds (at most). According to the outranking model, the prioritised objectives (here, we refer to the criteria with the greatest weights) are the second, sixth and ninth. With this idea in mind, consider the aspiration point with the following values (prioritised criteria are overlined)

$$\mathbb{A} = \left\langle 1387735,\ \overline{1139370},\ 1584605,\ 1056785,\ 1848960,\ \overline{1386425},\ 2203195,\ 1554445,\ \overline{1828560} \right\rangle,$$

and the reservation point is

$$\mathbb{R} = \left\langle 1359985,\ \overline{1122535},\ 1537070,\ 1035650,\ 1775005,\ \overline{1365945},\ 2159135,\ 1507815,\ \overline{1801540} \right\rangle,$$

and let us recall the original multi-criteria impact of x is

$$z(x) = \left\langle 1387735,\ \overline{1122535},\ 1584605,\ 1056785,\ 1848960,\ \overline{1365945}\ ,2203195,\ 1554445,\ \overline{1801540} \right\rangle.$$

By optimising the resulting compromise programming model, a solution x' was obtained with the next vector of objective functions

$$z(x') = \left\langle 1363110,\ \overline{1124720},\ 1554150,\ 1049330,\ 1843815,\ \overline{1370995},\ 2178140,\ 1540755,\ \overline{1803470} \right\rangle.$$

The next step is to determine the relation between x' and x. According to the outranking model, $x'\mathrm{I}x$. This is a consequence of $|z_k(x) - z_k(x')| \le u_k\ \forall k \in \{1,\ 2,\ 3, \ldots,\ p\}$ and there is no Pareto dominance between x and x'. According to the preference model, the DM does not perceive any significant advantage of one solution to another; therefore, they are equivalent from that point of view.

If the DM insists on reaching the goals that they set for the prioritised criteria, then they should be prepared to allow the values of some of the objective functions to decrease beyond the indifference threshold (Phase 3). Now, let us suppose that the DM is prepared to decrease the return in the least important criteria in favour of an advantageous trade-off in the prioritised objectives. To be consistent, let us assume that the less important objectives are those with the lowest weights (e.g. the third, fifth and eighth criteria). A major point to make is to determine by how much those objectives should be allowed to decrease. As argued earlier, these losses have been prevented from being greater than the pre-veto thresholds to avoid conditions of incomparability between the two solutions (see Eq. 21). Then, the vector $\mathbb{R}$ is as follows (secondary objectives are underlined)

$$\mathbb{R} = \left\langle 1359985,\ \overline{1122535},\ \underline{1434070},\ 1035650,\ \underline{1645575},\ \overline{1365945},\ 2159135,\ \underline{1391230},\ \overline{1801540} \right\rangle,$$

and after solving the compromise model, the following solution was generated

$$z(x'') = \left\langle 1386135,\ \overline{1139070},\ \underline{1536890},\ 1053580,\ \underline{1740980},\ \overline{1383540},\ 2200685,\ \underline{1510625},\ \overline{1826850} \right\rangle.$$

After getting the result, we have to assess both solutions through the ELECTRE III model. Consequently, a strict preference relation is stated ($x''\mathrm{P}x$). According to the outranking relation, the DM has clear reasons to justify why they prefer x'' to x.

4.2. Using an ACO algorithm in Phase 1

To determine the level of effectiveness of our proposal, we conducted a series of experiments using Non-Outranked ACO (NO-ACO II) [1] in Phase 1. This ACO algorithm is a many-objective portfolio optimisation approach with *a priori* preference articulation. For the purpose of having this article be self-contained, we have added a brief description of NO-ACO II in Appendix A. The parameter values used for the algorithm are $\rho = 0.9,\ \gamma = 25,\ \alpha_1 = 0.65,\ \alpha_2 = 0.85,\ w_0 = 0.6,$ $iter_{\max} = 1000$ and $rep_{\max} = 50$ [1].

We ran this algorithm on two benchmarks of instances [1,22]. The first consists of 10 instances with 100 projects and 9 objectives, conducting 30 runs for each instance. Both the runtime and the results achieved in each instance are presented.

With regard to the runtime, Table 2 presents the average times consumed for each instance using the ACO algorithm (second column), followed by the time consumed by our compromise model in Phase 2 (third column), in Phase 3 (forth column), and the sum of both (fifth column). The ratio of this runtime with that of ACO is next to the raw values. According to Table 2, Phase 2 and Phase 3 are fast enough to perform a multi-criteria analysis on portfolios in an online and interactive way for these instances because the DM can try different scenarios in less than one second (the averages are 155 ms for Phase 2 and 191 ms for Phase 3).

Table 2
Runtimes using NO-ACO II in Phase 1 for instances with 100 projects and 9 objectives.

Instance	Phase 1 Time (ms)	Phase 2 (Ph2) Time (ms)		Phase 3 (Ph3) Time (ms)		Total (Ph2 + Ph3) Time (ms)	
1	6594.83	140.05	2%	163.19	2%	303.24	5%
2	5928.43	138.33	2%	175.43	3%	313.77	5%
3	7554.90	168.70	2%	190.50	3%	359.20	5%
4	8688.83	159.03	2%	271.94	3%	430.97	5%
5	8679.17	121.47	1%	156.41	2%	277.88	3%
6	5930.83	203.10	3%	193.60	3%	396.70	7%
7	6718.30	184.20	3%	225.62	3%	409.82	6%
8	8246.13	131.20	2%	197.95	2%	329.15	4%
9	8074.83	168.23	2%	171.60	2%	339.83	4%
10	7276.47	131.70	2%	162.81	2%	294.51	4%

Table 3
Results using NO-ACO II in Phase 1 for instances with 100 projects and 9 objectives.

Instance	$x'Px$	$x'Qx$	$x''Px'$	$x''Qx'$	$x''Px$	$x''Qx$	$x''Kx$	Effectiveness
1	13	14	13	10	17	9	4	100%
2	8	10	8	9	15	9	0	80%
3	8	10	6	11	11	7	3	70%
4	18	12	15	8	19	6	2	90%
5	11	17	10	13	17	9	2	93%
6	6	10	8	8	11	7	2	67%
7	2	19	6	5	7	19	1	90%
8	11	17	7	19	11	16	0	90%
9	3	17	7	14	10	13	3	87%
10	12	10	8	9	13	12	0	83%

Note: x, x' and x'' are the portfolios obtained after applying Phases 1, 2 and 3, respectively.

This experiment also provides evidence in favour of an improvement in the quality reached after applying both phases. Table 3 summarises these results. Column 1 has the sequential numbers of the instances, columns 2 and 3 present the results (in terms of outranking relations) when x and x' are compared, indicating the quantity of runs for which the (strict/weak) preference relations held. Similarly, columns 4 and 5 compare x' and x''. Columns 6, 7 and 8 compare x and x''. Unlike the previous comparisons, the k-preference relation became active. Column 9 introduces a measure called 'effectiveness', which is the proportion of runs for which x'' outranks x ($x''Px \vee x''Qx \vee x''Kx$). On average, we summarise Table 3 as follows:

(a) after Phase 2, x' strictly outranked x 31% of the runs, and
(b) x' weakly outranked x 45% of the runs,
(c) after Phase 3, x'' strictly outranked x 44% of the runs,
(d) x'' weakly outranked x (we mean $x''Qx \vee x''Kx$) 42% of the runs, and
(e) x'' outranked x 85% of the runs (effectiveness).

We also tested our compromise model on portfolio instances with 500 projects and 16 maximising objectives to get a notion of its effectiveness at a higher scale. Again, we ran our framework 30 times for each instance. Table 4 presents the following averages for each instance: time consumed in Phase 1, in Phase 2, and in Phase 3. The last column has the time the compromise model consumed in both phases, it also includes the proportion with respect to that consumed by the multi-objective metaheuristic approach of Phase 1. In conclusion, this approach is also adequate for performing online multi-criteria analyses for instances with these features (on average, Phase 2 consumes 585 ms, Phase 3 consumes 550 ms, and both phases consume 1135 ms).

In terms of the quality reached by our model, Table 5 summarises the experimental results and its columns should be interpreted, accordingly, as described in Table 3. We want to emphasise the results for Instances 3–6, 8 and 9, for which a better solution (in terms of fuzzy outranking) is always reached after applying the interactive multi-criteria analysis based on compromise programming proposed in this paper. After analysing the averages, we summarise Table 5 as follows:

(a) after Phase 2, x' strictly outranked x 57% of the runs, and
(b) x' weakly outranked x 36% of the runs,
(c) after Phase 3, x'' strictly outranked x 72% of the runs,
(d) x'' weakly outranked x 22% of the runs, and
(e) x'' outranked x 94% of the runs (effectiveness).

Table 4
Runtimes using NO-ACO II in Phase 1 for instances with 500 projects and 16 objectives.

Instance	Phase 1 (Ph1) Time (ms)	Phase 2 (Ph2) Time (ms)	Phase 3 (Ph3) Time (ms)	Total (Ph2 + Ph3) Time (ms)	
1	307773	855.07	789.60	1644.70	0.53%
2	294570	591.00	647.30	1238.30	0.42%
3	295752	623.80	497.10	1120.87	0.38%
4	304850	655.90	560.60	1216.50	0.40%
5	289220	538.87	556.10	1094.97	0.38%
6	304618	536.60	435.30	971.93	0.32%
7	306337	552.00	610.60	1162.60	0.38%
8	301533	578.03	505.30	1083.33	0.36%
9	316402	375.67	389.50	765.20	0.24%
10	343519	545.80	510.50	1056.30	0.31%

Table 5
Results using NO-ACO II in Phase 1 for instances with 500 projects and 16 objectives.

Instance	$x'Px$	$x'Qx$	$x''Px'$	$x''Qx'$	$x''Px$	$x''Qx$	$x''Kx$	Effectiveness
1	8	8	6	15	10	12	0	73%
2	18	11	18	11	20	2	1	77%
3	26	4	20	8	27	0	3	100%
4	21	3	22	8	28	2	0	100%
5	21	7	18	11	23	7	0	100%
6	21	9	23	7	25	3	2	100%
7	16	12	13	15	21	5	1	90%
8	9	16	15	13	21	8	1	100%
9	23	7	23	7	23	7	0	100%
10	14	10	14	14	19	10	0	97%

Note: x, x' and x'' are the portfolios obtained after applying Phases 1, 2 and 3, respectively.

According to Table 3 and Table 5, large instances show the advantages of our framework more tangibly. To provide more experimental evidence, the next section will explore the application of our proposal using other state-of-the-art *a priori* metaheuristic in Phase 1.

4.3. Using H-MCSGA for Phase 1

Our proposal was applied using a GA – called Hybrid Multi-Criteria Sorting Genetic Algorithm (H-MCSGA) [22] – that also solves the portfolio problem for Phase 1. A brief description of H-MCSGA is presented in Appendix B for consultation. The same sets of instances were used: 100 projects and 9 objectives, and 500 projects and 16 objectives. In all, 30 runs were performed with the following parameter settings: $iterations = 500$, $population\_size = 100$, $mutation\_rate = 0.05$, and $assignation\_type = pessimist$ [22].

Table 6 show the average runtime for each instance consumed in Phase 1, Phase 2 and both. The proportions (in relation to the runtime of H-MCSGA) are next to the raw values. According to Table 6, the compromise model is fast enough to be considered 'online' (on average, it required 155 ms in Phase 2 and 191 ms in Phase 3), which is in line with the experiments conducted in Section 4.2.

Regarding the quality of the solutions, Table 7 summarises the experimental results. Its columns should be interpreted as in Table 3. After analysing the averages, we summarise Table 7 as follows:

(a) after Phase 2, x' strictly outranked x 39% of the runs, and
(b) x' weakly outranked x 49% of the runs,
(c) after Phase 3, x'' strictly outranked x 51% of the runs,
(d) x'' weakly outranked x 33% of the runs, and
(e) x'' outranked x 84% of the runs (effectiveness).

We also tested this setup of the framework with 10 instances with 500 projects and 16 objectives. Table 8 presents the average runtimes for each instance, which are consistent with the results of Section 4.2. On average, our approach consumes less than two seconds to perform both phases once (Phase 1 consumes 585 ms and Phase 2 consumes 550 ms).

Table 9 presents the results of the programming model in terms of outranking. There are instances for which our approach reaches 100% effectiveness (Instances 3–5, 7, 9 and 10). After analysing the averages, we summarise Table 9 as follows:

Table 6
Runtimes using H-MCSGA in Phase 1 for instances with 100 projects and 9 objectives.

Instance	Phase 1 (Ph1) Time (ms)	Phase 2 (Ph2) Time (ms)		Phase 3 (Ph3) Time (ms)		Total (Ph2 + Ph3) Time (ms)	
1	1022.53	152.57	15%	150.09	15%	303.47	30%
2	1031.83	121.93	12%	183.91	18%	305.84	30%
3	1043.37	134.23	13%	214.11	21%	348.34	33%
4	1017.83	137.50	14%	166.22	17%	303.72	30%
5	1177.60	124.37	11%	128.43	11%	252.80	21%
6	1366.37	150.13	11%	221.29	16%	371.42	27%
7	1369.73	143.47	10%	172.28	13%	315.75	23%
8	1385.07	106.47	8%	153.70	11%	260.17	19%
9	1360.03	131.87	10%	181.22	13%	313.09	23%
10	1449.57	119.00	8%	186.05	13%	305.05	21%

Table 7
Results using H-MCSGA in Phase 1 for instances with 100 projects and 9 objectives.

Instance	$x'Px$	$x'Qx$	$x''Px'$	$x''Qx'$	$x''Px$	$x''Qx$	$x''Kx$	Effectiveness
1	12	17	14	11	18	10	0	93%
2	11	12	7	10	13	10	4	90%
3	14	15	9	12	15	5	0	67%
4	16	13	17	8	20	5	4	97%
5	14	15	11	16	17	3	3	77%
6	11	12	5	9	19	1	0	67%
7	7	16	5	16	11	14	1	87%
8	7	20	10	11	8	15	3	87%
9	11	15	5	17	16	10	1	90%
10	14	11	8	10	15	11	0	87%

Note: x, x' and x'' are the portfolios obtained after applying Phases 1, 2 and 3, respectively.

Table 8
Runtimes using H-MCSGA in Phase 1 for instances with 500 projects and 16 objectives.

Instance	Phase 1 Time (ms)	Phase 2 (Ph2) Time (ms)		Phase 3 (Ph3) Time (ms)		Total (Ph2 + Ph3) Time (ms)	
1	2932.83	991.87	34%	798.40	27%	1790.27	61%
2	2753.73	562.00	20%	658.20	24%	1220.20	44%
3	3069.77	675.63	22%	600.90	20%	1276.53	42%
4	3730.00	716.83	19%	696.70	19%	1413.53	38%
5	3270.57	644.03	20%	608.30	19%	1252.33	38%
6	3329.40	658.17	20%	611.40	18%	1269.57	38%
7	3367.20	453.23	13%	638.30	19%	1091.53	32%
8	3325.93	709.77	21%	506.60	15%	1216.37	37%
9	3523.07	444.60	13%	369.10	10%	813.70	23%
10	3467.70	779.83	22%	585.90	17%	1365.73	39%

Table 9
Results using H-MCSGA in Phase 1 for instances with 500 projects and 16 objectives.

Instance	$x'Px$	$x'Qx$	$x''Px'$	$x''Qx'$	$x''Px$	$x''Qx$	$x''Kx$	Effectiveness
1	7	10	11	11	11	5	4	67%
2	19	10	14	16	21	4	0	83%
3	21	5	28	2	28	0	2	100%
4	15	9	13	13	18	9	3	100%
5	22	6	24	5	29	1	0	100%
6	17	8	17	10	19	6	0	83%
7	18	12	21	8	22	5	3	100%
8	17	9	8	20	23	6	0	97%
9	24	6	20	10	26	4	0	100%
10	17	9	16	8	24	5	1	100%

Note: x, x' and x'' are the portfolios obtained after applying Phases 1, 2 and 3, respectively.

(a) after Phase 2, x' strictly outranked x 59% of the runs, and
(b) x' weakly outranked x 28% of the runs,
(c) after Phase 3, x'' strictly outranked x 74% of the runs,
(d) x'' weakly outranked x 19% of the runs, and
(e) x'' outranked x 93% of the runs (effectiveness).

Sections 4.2 and 4.3 provide evidence that our approach is compatible with *a priori* many-objective optimisation metaheuristics. However, *a posteriori* algorithms are also popular in the literature. To demonstrate its versatility, in the following experiment, we present the results obtained by instantiating our framework with an *a posteriori* evolutionary algorithm in Phase 1.

4.4. Using MOEA/D for Phase 1

Also, our proposal was applied by taking the MultiObjective Evolutionary Algorithm based on Decomposition (MOEA/D) [45] to solve the portfolio problem in Phase 1 [23]. MOEA/D is standard in the literature for addressing many-objective opti-

Table 10
Runtimes using MOEA/D in Phase 1 for instances with 100 projects and 9 objectives.

Instance	Phase 1 (Ph1) Time (ms)	Phase 2 (Ph2) Time (ms)		Phase 3 (Ph3) Time (ms)		Total (Ph2 + Ph3) Time (ms)	
1	14117.77	171.66	1.22%	175.58	1.24%	347.24	2.46%
2	15249.77	112.95	0.74%	172.96	1.13%	285.91	1.87%
3	14882.23	146.95	0.99%	235.78	1.58%	382.73	2.57%
4	13698.20	147.06	1.07%	176.65	1.29%	323.72	2.36%
5	11684.17	141.69	1.21%	128.83	1.10%	270.52	2.32%
6	11357.63	167.48	1.47%	199.41	1.76%	366.89	3.23%
7	12149.37	157.24	1.29%	179.42	1.48%	336.65	2.77%
8	17126.13	98.27	0.57%	151.59	0.89%	249.87	1.46%
9	12732.87	118.95	0.93%	206.50	1.62%	325.45	2.56%
10	12349.47	113.76	0.92%	204.70	1.66%	318.46	2.58%

Table 11
Results using MOEA/D in Phase 1 for instances with 100 projects and 9 objectives.

Instance	$x'Px$	$x'Qx$	$x''Px'$	$x''Qx'$	$x''Px$	$x''Qx$	$x''Kx$	Effectiveness
1	12	1	11	12	29	1	0	100%
2	14	5	5	16	15	11	1	90%
3	16	4	11	5	24	1	3	93%
4	8	6	9	5	14	10	1	83%
5	15	9	8	1	27	0	3	100%
6	7	11	5	0	14	4	7	83%
7	6	9	7	0	7	3	8	60%
8	6	5	9	2	10	9	4	77%
9	6	12	5	9	6	13	5	80%
10	9	6	9	11	14	5	5	80%

Note: x, x' and x'' are the portfolios obtained after applying Phases 1, 2 and 3, respectively.

misation problems from a perspective based on *a posteriori* decision analysis. This evolutionary algorithm is still widely studied and applied [e.g. 46–49]. We provide a brief description of MOEA/D in Appendix C. In this experiment, MOEA/D ran 30 times on the same benchmarks (100 projects and 9 objectives, and 500 projects and 16 objectives). The parameter setting for MOEA/D was the following: $iterations = 500$, $mutation\_rate = 0.05$, $crossover\_rate = 1$, and $T = 10$ [45]; the population size is 165 for the 9-objective instances, and 136 for the 16-objective instances.

Table 10 presents the following averages for each 9-objective instance: runtime for Phase 1, Phase 2, and Phase 3. The last column has the time the compromise model consumes in both phases; it also includes the proportion with MOEA/D (Phase 1). In conclusion, this approach is adequate for performing an analysis of problems with these features because, on average, Phase 2 consumed 137 ms, Phase 3 consumed 183 ms, and both phases jointly consumed 320 ms; so, our model consumed 2.42% of the time consumed by MOEA/D.

Table 11 summarises the experimental results on the quality of the solutions. Its columns should be interpreted as in Tables 3 and 7. Our approach reached 100% effectiveness in two instances (1 and 5). After analysing the averages, we summarise Table 11 as follows:

(a) after Phase 2, x' strictly outranked x 33% of the runs, and
(b) x' weakly outranked x 23% of the runs,
(c) after Phase 3, x'' strictly outranked x 53% of the runs,
(d) x'' weakly outranked x 32% of the runs, and
(e) x'' outranked x 85% of the runs (effectiveness).

Regarding the instances with 16 objectives, Table 12 presents the average times consumed for each instance using MOEA/D in Phase 1. The columns of Table 12 should be interpreted as in Tables 4 and 8. The averages were 602.46 ms for Phase 2 and 570.29 ms for Phase 3. Phase 2 and Phase 3 jointly consumed 1172.75 ms, which is 1.67% of the time consumed by MOEA/D. As a consequence, Phase 2 and Phase 3 performed the multi-criteria analysis on portfolios in an online and interactive fashion, even on large-scale instances.

Table 13 presents the results of the programming model in terms of outranking. Our approach reached 100% effectiveness in six instances (1, 3–5, 8, and 10). After analysing the averages, we summarise Table 13 as follows:

(a) after Phase 2, x' strictly outranked x 47% of the runs, and
(b) x' weakly outranked x 33% of the runs,
(c) after Phase 3, x'' strictly outranked x 69% of the runs,

Table 12
Runtimes using MOEA/D in Phase 1 for instances with 500 projects and 16 objectives.

Instance	Phase 1 Time (ms)	Phase 2 (Ph2) Time (ms)		Phase 3 (Ph3) Time (ms)		Total (Ph2 + Ph3) Time (ms)	
1	73916.63	931.03	1.26%	726.90	0.98%	1657.93	2.24%
2	63358.90	529.77	0.84%	673.43	1.06%	1203.20	1.90%
3	79033.63	590.59	0.75%	541.97	0.69%	1132.56	1.43%
4	71166.97	624.27	0.88%	633.63	0.89%	1257.90	1.77%
5	65080.47	560.53	0.86%	567.43	0.87%	1127.97	1.73%
6	79338.00	619.63	0.78%	568.92	0.72%	1188.55	1.50%
7	68301.03	429.22	0.63%	597.26	0.87%	1026.48	1.50%
8	67106.20	638.63	0.95%	483.05	0.72%	1121.68	1.67%
9	58542.30	409.36	0.70%	380.66	0.65%	790.02	1.35%
10	73974.27	691.60	0.93%	529.64	0.72%	1221.25	1.65%

(d) x'' weakly outranked x 28% of the runs, and
(e) x'' outranked x 97% of the runs (effectiveness).

In conclusion, the compromise model consistently maintained the runtime and the level of improvement of the solutions for these benchmarks, regardless of the metaheuristic applied in Phase 1. Both test suites, which have been used repeatedly in the literature of multi-criteria portfolio optimisation [e.g. 25,50,22,4], consist of synthetic instances. In the next section, we apply our model to aid the decision analysis to a real-world case study of a social project portfolio.

4.5. Case study: Social welfare project portfolios based on polygons of poverty

The Mexican government programme ‘Prospera’ is an initiative that is implemented in all of Mexico’s cities. Its main objective is to support projects benefiting the most vulnerable strata of the population, who are geographically grouped as ‘polygons of poverty’, because the national government has a particular concern for this sector, which has been marginalised from society. Each local government prioritises and makes specific the goals and the criteria in line with the national level. The decision-making committee (called the DM from here on) faces some challenging issues, for example:

(a) The quality of the portfolio is a holistic measure of the projects (therefore, the projects which are individually evaluated as best do not necessarily make up the best portfolio); so, the DM could hardly estimate the composite impact of each portfolio that they want to try.
(b) The DM has to justify the choice of the selected portfolio according to the national policies for improving the quality of life of these strata.

Ciudad Juárez is the fifth largest city in Mexico (in terms of both population and territory) with 27 polygons of poverty, whose locations are shown in Fig. 2. Each project x_i is described by six attributes:

(a) Number of people benefited – $f_1(i)$: The number of people benefited by the ith project.
(b) Deployment time – $f_2(i)$: The number of months the project needs before benefiting people.
(c) Life span – $f_3(i)$: The number of months the project serves or benefits people (service life).
(d) Region – r_i: The polygon of poverty on which the ith project has an impact.
(e) Area – a_i: Each project chiefly impacts one of the following public areas: health, food, education, and business startup.
(e) Requested budget – b_i: The amount of resources requested by the ith project.

Table 13
Results using MOEA/D in Phase 1 for instances with 500 projects and 16 objectives.

Instance	$x'Px$	$x'Qx$	$x''Px'$	$x''Qx'$	$x''Px$	$x''Qx$	$x''Kx$	Effectiveness
1	16	8	15	12	16	11	3	100%
2	19	4	2	17	25	3	0	93%
3	14	9	0	14	28	2	0	100%
4	10	17	7	14	19	10	1	100%
5	15	8	0	11	23	5	2	100%
6	12	10	14	10	14	12	0	87%
7	19	4	13	14	26	2	0	93%
8	11	13	4	16	13	17	0	100%
9	9	14	8	12	19	8	2	97%
10	15	11	5	17	23	6	1	100%

Note: x, x' and x'' are the portfolios obtained after applying Phases 1, 2 and 3, respectively.

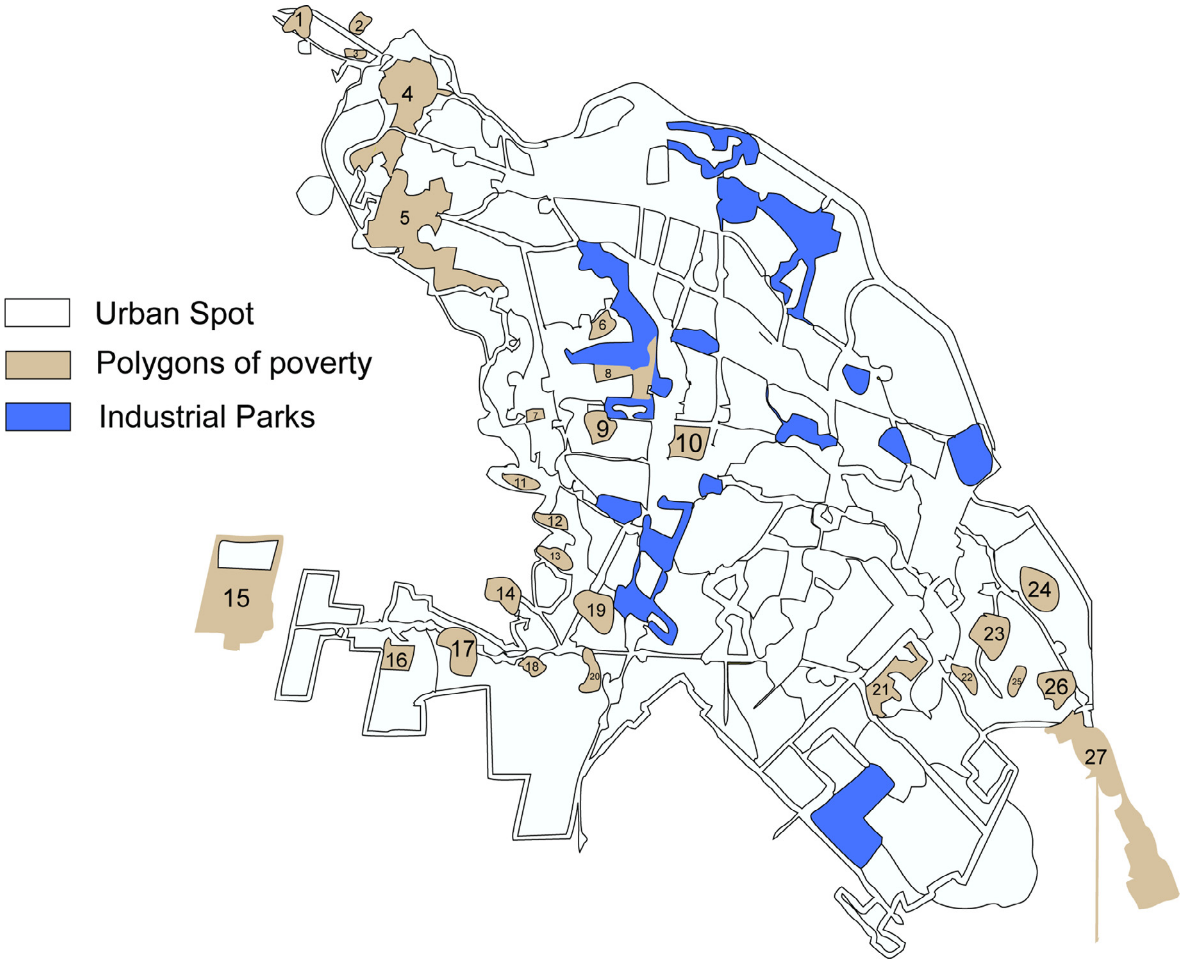

Fig. 2. Polygons of poverty in Ciudad Juárez.

A binary vector $x = \langle x_1, x_2, x_3, \ldots, x_n \rangle$ (where $x_i = 1$ indicates that the ith project receives support, $x_i = 0$ otherwise) represents a portfolio – for n candidate projects – with the multi-criteria benefit return $z(x) = \langle z_1(x), z_2(x), z_3(x), z_4(x), z_5(x) \rangle$, where:

(a) the first objective is to maximise the total number of people benefited, calculated as $z_1(x) = \sum_{i=1}^{n} [x_i \cdot f_1(i)]$;

(b) the second objective is to maximise the quantity of projects supported, $z_2(x) = \sum_{i=1}^{n} x_i$;

(c) the third objective is to minimise the average of the deployment times for the supported projects, $z_3(x) = \sum_{i=1}^{n} [x_i \cdot f_2(i)] \cdot z_2(x)^{-1}$;

(d) the fourth objective is to maximise the average of the time span for the supported projects, $z_4(x) = \sum_{i=1}^{n} [x_i \cdot f_3(i)] \cdot z_2(x)^{-1}$; and

(e) the fifth objective is to maximise the number of polygons supported, $z_5(x) = \sum_{j=1}^{q} \varrho_j(x)$, where q is the number of polygons of poverty (in this case, $q = 27$) and

$$\varrho_j(x) = \begin{cases} 1 & \text{if } \exists i : x_i = 1 \land r_i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

There is also a set of hard constraints regarding the budgetary allocation. As a result of policies in the decision-making process, the candidate projects are grouped by application area, and the organisation sets budgetary limits (or limits that concern the number of supported projects) for each group. Let m be the number of groups limiting the budgetary allocation of the portfolios, and the vectors $\mathscr{L} = \langle \ell_1, \ell_2, \ell_3, \ldots, \ell_m \rangle$ and $\mathscr{U} = \langle u_1, u_2, u_3, \ldots, u_m \rangle$ represent the lower and the upper limits of the budget for each group, respectively. Regarding the application areas, the following constraints are set:

$$\ell_l \leq \sum_{i=1}^{n} [x_i \cdot h_l(i)] \leq u_l \ \forall l \in \{1, 2, 3, \ldots, m\}, \tag{23}$$

where m is the number of areas and

$$h_l(i) = \begin{cases} b_i & \text{if } a_i = l, \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

Given a budget B for the whole social programme, the main budgetary constraint is clearly

$$\sum_{i=1}^{n} [x_i \cdot b_i] \leq B. \tag{25}$$

In this case study, there are 125 candidate projects ($n = 125$) and a total budget of \$1,127,915. To support the DM, we inferred the parameters of the relational system of preferences by applying a method based on preference disaggregation analysis [36]. The preference system has the following values:

- weights, $W = \langle 0.39, 0.10, 0.12, 0.23, 0.16 \rangle$;
- indifference thresholds, $U = \langle 0.04, 0.08, 0.10, 0.10, 0.08 \rangle$;
- pre-veto thresholds, $S = \langle 0.11, 0.23, 0.20, 0.20, 0.11 \rangle$;
- veto thresholds, $V = \langle 0.15, 0.30, 0.25, 0.25, 0.30 \rangle$; and
- the parameters that define the outranking relations, $\lambda = 0.60$, $\epsilon = 0.12$, and $\beta = 0.26$.

Here, we adapted NO-ACO II [1] to get the initial portfolio required in Phase 1. The results after an iteration of our proposal follow.

Phase 1: The initial portfolio x has the following vector of benefits

$$z(x) = \langle 108448, 21, 12.15, 29.52, 19 \rangle.$$

Although the DM thinks that x is a good solution, they wish to increase the number of people benefited (the first criterion) and the average life span of the supported projects (the fourth criterion). Then, Phase 2 and Phase 3 are performed simultaneously.

Phase 2: The model sets the aspiration and reservation points according to Eq. 16 and Eq. 17. Then, the model is described as (prioritised objectives are overlined)

$$\mathbb{A} = \langle \overline{114955}, 21, 12.15, \overline{34.52}, 19 \rangle,$$
$$\mathbb{R} = \langle \overline{108448}, 19, 14.15, \overline{29.52}, 17 \rangle.$$

Branch & Cut solves the compromise programming model (Problem 18) providing an x' with the multi-criteria return

$$z(x') = \langle \overline{111189}, 21, 13.81, \overline{33.73}, 18 \rangle.$$

According to the outranking preferences, $x'\mathrm{Q}x$ because $\sigma(x', x) = 1$ and $\sigma(x, x') = 0.77$. Although x' seems to be slightly better than x, the DM might be likely to feel unsure between $x'\mathrm{P}x$ and $x'\mathrm{I}x$.

Phase 3: Eq. 16 and Eq. 21 define the reservation and aspiration points in this phase as (secondary objectives are underlined)

$$\mathbb{A} = \langle \overline{114955}, \underline{21}, \underline{12.15}, \overline{34.52}, 19 \rangle,$$
$$\mathbb{R} = \langle \overline{108448}, \underline{16}, \underline{15.15}, \overline{29.52}, 17 \rangle,$$

and the resulting solution of this model was

$$z(x'') = \langle \overline{114085}, \underline{19}, \underline{15.11}, \overline{34.46}, 19 \rangle.$$

The relational system of preferences identifies that $x''\mathrm{P}x$ with $\sigma(x'', x) = 0.88$ and $\sigma(x, x'') = 0.38$. Therefore, the DM should be sure that x'' is better than x according to their latest expression of preferences over the criteria and its objective values. In the real-world scenario, x'' was preferred to x (and x' too).

5. Conclusions and directions for future research

In this article, we presented a three-phased framework to aid the multi-criteria decision analysis of portfolios in an online and interactive fashion. The first phase consists of getting a representation of the preferences of the decision maker (DM), based on ELECTRE III, and optimising via a many-objective optimisation method. Afterwards, the best compromise solution is identified. This phase is the most demanding in terms of time and cognitive effort by the DM. At the end of this phase, the DM often has a clearer idea of what values they can obtain in the criteria and a notion of the balance between them. Then, we suggest that the DM go to Phase 2 or Phase 3 if they want to try scenarios with particular goals to reach. Indeed, they should do this to feel confident of the decision to make.

Phase 2 and Phase 3 get the edge by the knowledge gained in Phase 1 about the preferences of the DM and the ranges of the objective functions in the optimised portfolios. If the optimisation method employed in Phase 1 reaches solutions close to the true Pareto frontier, then the DM should be aware that some objectives could suffer as the attainment of the prioritised objectives improves. The main contribution of this paper is in providing a guideline to build compromise programming models that follow the principles of fuzzy outranking theory, based on ELECTRE III, to prevent incomparability during the analysis. In addition, it aids the DM in identifying strictly preferred solutions.

We have tested our framework instantiating Phase 1 with three different many-objective metaheuristic approaches for portfolio optimisation using two benchmarks reported in the literature. Experimental evidence showed that our model computes quickly enough to be interactive and it is even capable of improving high quality solutions; these advantages were especially marked in large-scale problems. For these synthetic instances, a fuzzy system of outranking relations was employed to measure the preferences of the DM over the multi-criteria returns of the portfolios. Also, this three-phased framework showed its versatility by being coupled to different state-of-the-art optimisation methods that use both an *a priori* approach (i.e. H-MCSGA and NO-ACO II) as well as an *a posteriori* one (i.e. MOEA/D).

We also satisfactorily treated a case study with 125 projects and 5 objectives in the context of a social assistance programme in Mexico. Our proposal offered a methodological framework to support a multi-criteria analysis by the decision-making committee. In practical terms, the chief contribution of this approach is the level of confidence that the DM can feel in the final prescription.

There are many different ways to improve and extend this framework. First, after some iterations of applying our approach, several non-ideal phenomena were observed. In particular, incomparability and cyclic preference relations were present between some portfolios. We are currently working on providing an effective resolution of these conditions to make the DM feel confident in the final decision on the portfolios. We will also model outranking preference relations using other mathematical programming techniques, especially ϵ-constraint and goal programming. Finally, the most challenging issue would be to extend this approach to articulate group decisions when the members of the decision entity have heterogeneous systems of preferences.

Credit authorship contribution statement

Gilberto Rivera: Conceptualization, Methodology, Validation, Formal analysis, Data curation, Writing - original draft, Visualization, Supervision. **Rogelio Florencia:** Conceptualization, Methodology, Validation, Writing - original draft, Visualization, Supervision. **Mario Guerrero:** Software, Validation, Data curation, Writing - original draft. **Raúl Porras:** Software, Validation, Data curation. **J. Patricia Sánchez-Solís:** Conceptualization, Validation, Investigation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Non-Outranked Ant Colony Optimisation II

NO-ACO II is an ant colony algorithm that uses a fuzzy outranking preference model and an Integer Linear Programming method during the optimisation process to address portfolio problems with partial-funding features. Fig. 3 shows the NO-ACO II algorithm. The following describes how the algorithm works.

The algorithm receives as input on parameters the information that describes the projects and the decision process. The method starts with constructing an initial population of ants through a *Branch* & *Cut* procedure to optimise a weighted-sum function to have solutions with a certain degree of optimality instead of solutions generated almost at random. Afterwards, the built solutions are evaluated according to a fuzzy outranking model, identifying the non-strictly outranked frontier (NS_{local}) and the best compromise ($\mathscr{F}_1^{local}$). Ants with the best fitness will deposit a pheromone, helping future ants target potentially good areas in the search space.

After the above steps, the main loop of the optimisation process starts. Each ant in the colony constructs a solution by adding projects using a selection rule. After this, a support level is assigned to each ant by applying an assignment rule. The feasible and complete solutions form the set O. Subsequently, a pheromone evaporation process is carried out. Each port-

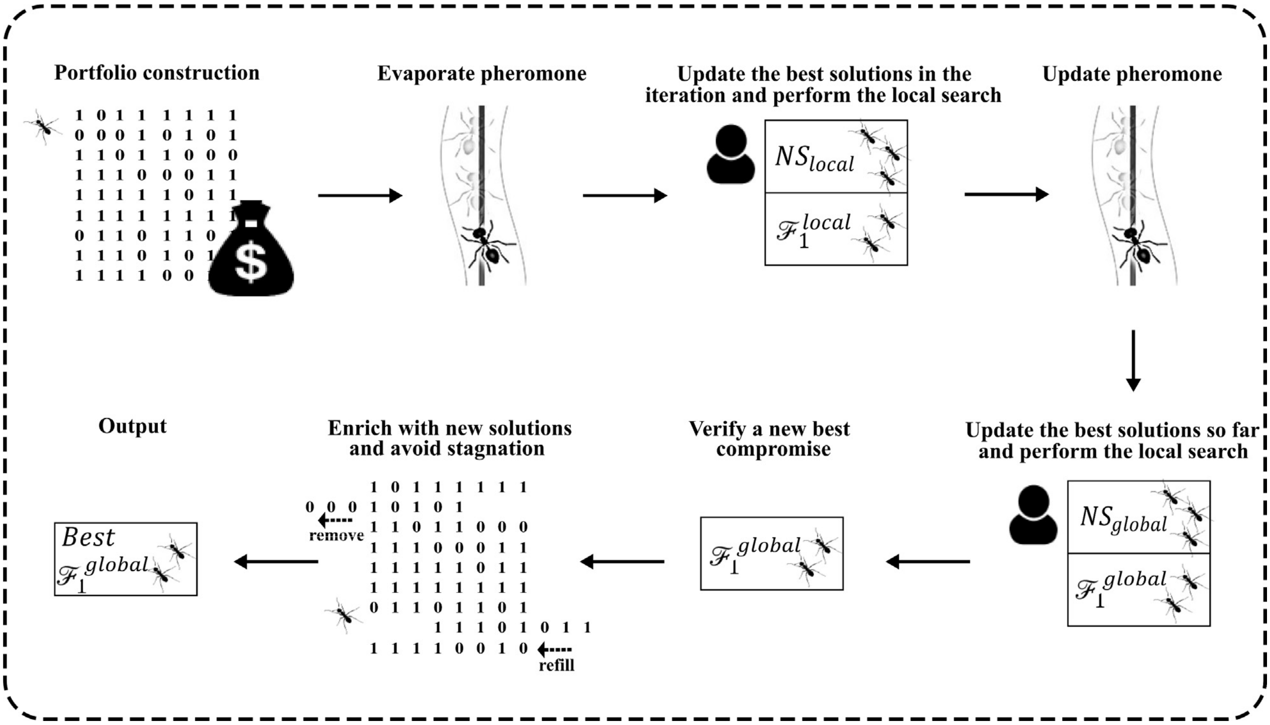

Fig. 3. The NO-ACO II algorithm.

folio in O is evaluated according to the fuzzy outranking model, identifying the non-strictly outranked frontier (NS_{local}) and the best compromise ($\mathscr{F}_1^{local}$), including the previous ones. Then, an updating process of the non-strictly outranked frontier is carried out and the best compromise is determined by applying a local search, exploring the solution space corresponding to the partial-funding model. The next step is the deposition of the pheromone.

Then, the non-strictly outranked frontier (NS_{global}) and the best compromise ($\mathscr{F}_1^{global}$) are updated. The best-known solutions are submitted regularly to the local search to be improved. Subsequently, a process to verify whether the algorithm has found a better solution is performed. For this, a procedure called remove & refill is carried out. This procedure removes from NS_{local} those solutions that have remained in it for a determined number of iterations, replacing them with solutions from (NS_{global}), thus avoiding stagnation.

The algorithm stops if the best solutions have been the same for several iterations or it has reached the maximum number of iterations. The output is an approximation of the best compromise (NS_{global}).

Appendix B. Hybrid Multi-Criteria Sorting Genetic Algorithm

The H-MCSGA is an algorithm that implicitly incorporates in an *a priori* way the preferences of the Decision Maker (DM) in a multi-objective evolutionary optimisation process. Fig. 4 shows the H-MCSGA algorithm.

The algorithm has two phases: (a) generation of a reference set using a metaheuristic approach to reflect the preferences of the DM, and (b) searching for the Region of Interest (RoI) using the evolutionary method proposed by the authors to find solutions according to the preferences of the DM. They define the RoI as those solutions that are non-dominated and considered 'satisfactory' by the DM. The algorithm receives as input the information about the projects and a set of sorted solutions which implicitly reflects the preferences of the DM. The two phases are described below.

Phase 1: Generation of the reference set

The procedure to create this set is to use a multi-objective metaheuristic approach to obtain solutions that represent an approximation to the Pareto frontier. These solutions are shown to the DM to be assigned to the categories 'satisfactory' or 'unsatisfactory', generating the reference set. The authors clarify that they simulated the DM using an outranking method instead. In this way, the preferences of the DM are implicitly reflected in it.

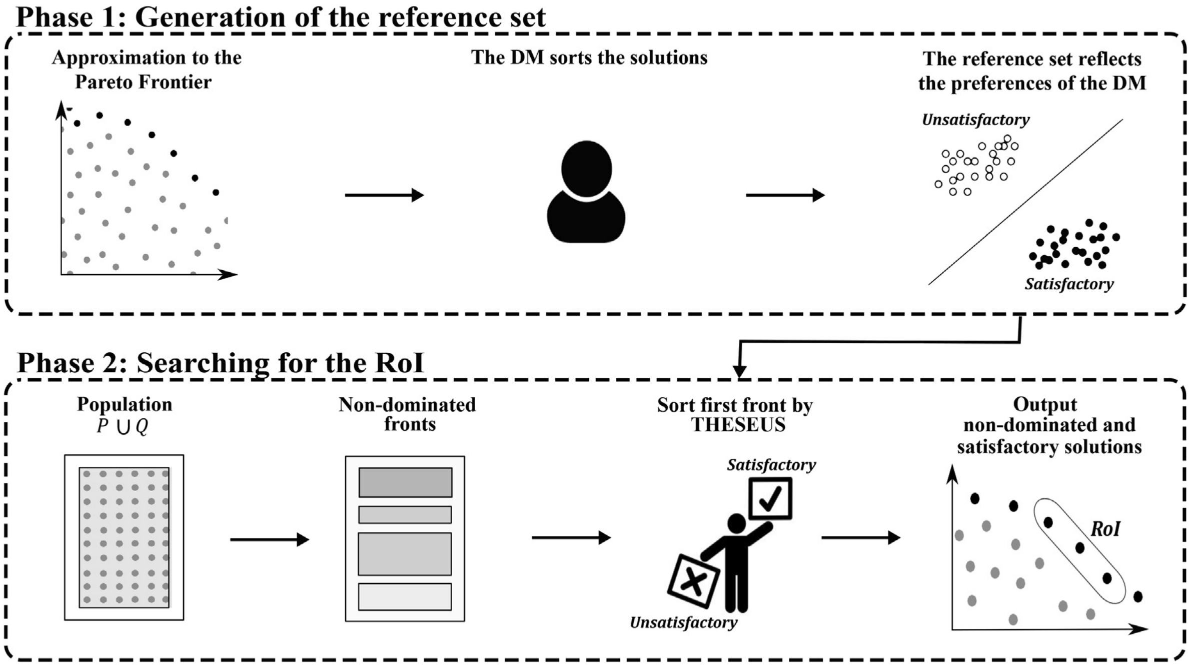

Fig. 4. The H-MCSGA algorithm.

Phase 2: Searching for the RoI

The search phase starts by generating a population of parents P that contains both the solutions from the reference set created in the first phase and random solutions.

Afterwards, non-dominated fronts are generated from P using Pareto Dominance. The solutions in the first front are sorted by a multi-criteria sorting method – THESEUS – to increase the selective pressure towards the RoI. Next, the fronts are enumerated, leaving the ‘satisfactory’ solutions on the first front. Subsequently, a child population Q is generated from P using the genetic operators: binary tournament selection (based on the order of the fronts), recombination, and mutation. Once the initial populations P and Q are available, a pre-determined number of iterations are performed using the following steps: P and Q are joined to generate non-dominated fronts, P'. Again, the solutions of the first front are sorted by THESEUS, and the fronts are enumerated. Afterwards, a new population P is generated according to the non-dominated fronts and a crowding distance. Subsequently, a new child population Q is generated from P using binary tournament selection (based on the order of the non-dominated fronts and the crowding distance), recombination, and mutation. Once the number of iterations is reached, the algorithm output consists of those solutions in the first non-dominated front from the last iteration sorted as ‘satisfactory’ solutions.

As can be seen, H-MCSGA uses a set of solutions sorted by a DM into ordered categories, which implicitly represents their preferences. The solutions in this set, generated during the optimisation process, are assigned to a category using a multi-criteria sorting method. In this way, the search is directed towards solutions that belong to the RoI.

Appendix C. MultiObjective Evolutionary Algorithm based on Decomposition

MultiObjective Evolutionary Algorithm based on Decomposition (MOEA/D) is an evolutionary algorithm that can use any decomposition approach to decompose a multi-objective problem into N subproblems to be optimised at the same time using aggregation functions. Fig. 5 shows the MOEA/D algorithm.

It consists of the following three steps: initialisation, update and stop criteria. The algorithm receives as input the information about the candidate projects, a set of N weight vectors, and a number T indicating the size of the neighbourhood for each vector. The operation of each step is described below.

Step 1: Initialisation

The process starts by creating an empty set called the external population EP, which stores the non-dominated solutions discovered during the optimisation process. The Euclidean distance between each pair of elements of a group of uniformly distributed weight vectors is calculated. Subsequently, a neighbourhood B is created for each vector with the T closest vectors to each of them. Next, an initial population P is randomly generated. Finally, a reference point z is constructed with the best value for each objective of each solution in P.

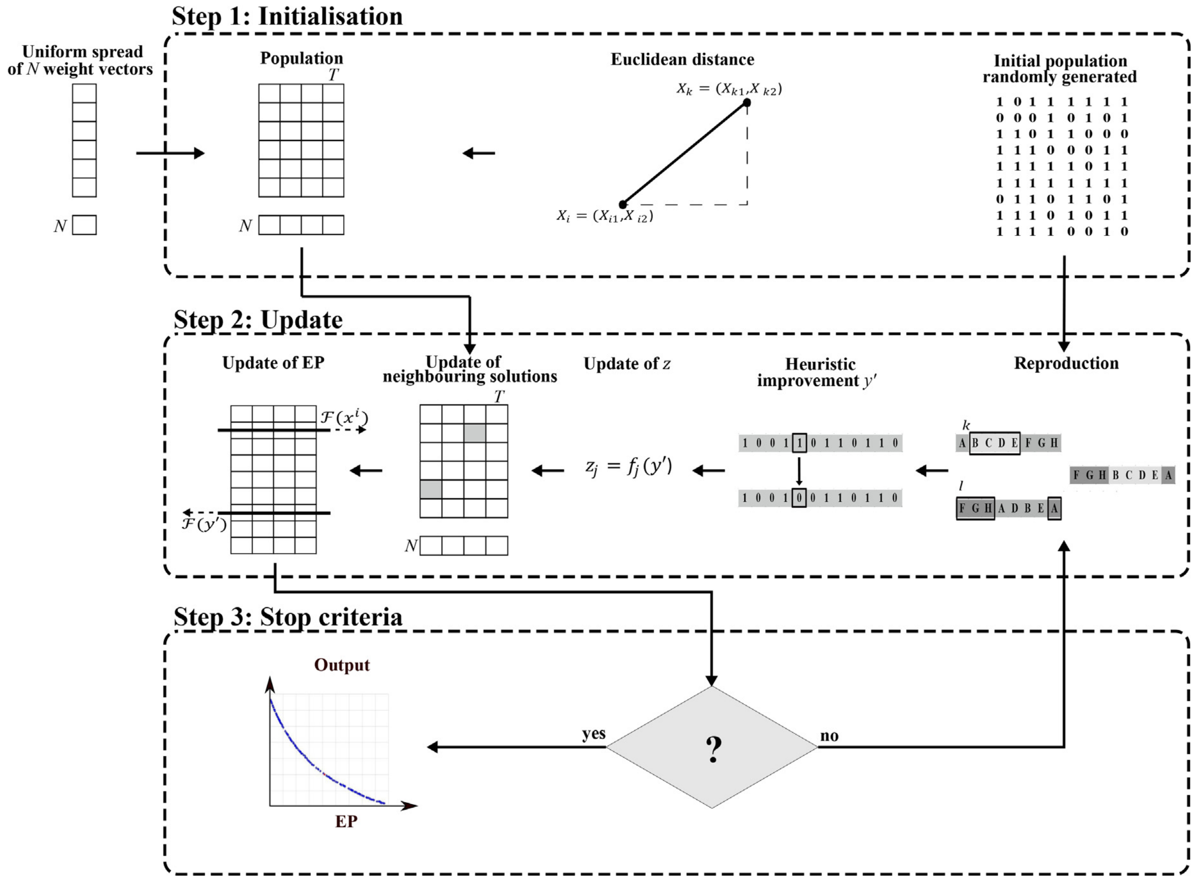

Fig. 5. The MOEA/D algorithm.

Step 2: Update

For each solution in *P*, the following actions are carried out. Two solutions are randomly selected from their neighbourhood *B*. They are used to generate a new solution *y*, applying genetic operators. Afterwards, an improvement or repair process is carried out on *y* to generate *y′*. Subsequently, the vector *z* is updated if any of the values of the objective function applied to *y′* is better than the current value for *z*. If *y′* performs better than any solution in *B*, this solution is replaced by *y′*, thus updating the neighbourhood *B*. Finally, the solutions dominated by *y′* are removed from *EP*, and *y′* is added to *EP* if there is no solution in this set that dominates it.

Step 3: Stop criteria

The algorithm stops and outputs the non-dominated solutions in *EP* when the stop criteria are reached; otherwise, it repeats Step 2.

References

[1] E. Fernandez, C. Gomez, G. Rivera, L. Cruz-Reyes, Hybrid metaheuristic approach for handling many objectives and decisions on partial support in project portfolio optimisation, Inf. Sci. 315 (2015) 102–122, https://doi.org/10.1016/j.ins.2015.03.064.
[2] M. Di Matteo, H.R. Maier, G.C. Dandy, Many-objective portfolio optimization approach for stormwater management project selection encouraging decision maker buy-in, Environ. Modelling Softw. 111 (2019) 340–355, https://doi.org/10.1016/j.envsoft.2018.09.008.
[3] H. Jafarzadeh, P. Akbari, B. Abedin, A methodology for project portfolio selection under criteria prioritisation, uncertainty and projects interdependency–Combination of fuzzy QFD and DEA, Expert Syst. Appl. 110 (2018) 237–249, https://doi.org/10.1016/j.eswa.2018.05.028.
[4] C.G. Gomez, L. Cruz-Reyes, G. Rivera, N. Rangel-Valdez, M.L. Morales-Rodriguez, M. Perez-Villafuerte, Interdependent projects selection with preference incorporation, in: New Perspectives on Applied Industrial Tools and Techniques, Springer-Verlag, 2018, pp. 253–271. doi:10.1007/978-3-319-56871-3_13..
[5] E. Fernandez, J. Navarro, E. Solares, C.A. Coello Coello, A novel approach to select the best portfolio considering the preferences of the decision maker, Swarm Evol. Comput. 46 (2019) 140–153, https://doi.org/10.1016/j.swevo.2019.02.002.
[6] F. Balderas, E. Fernandez, C. Gomez-Santillan, N. Rangel-Valdez, L. Cruz, An interval-based approach for evolutionary multi-objective optimisation of project portfolios, International Journal of Information Technology & Decision Making doi:10.1142/S021962201950024X..

[7] E.A. Frej, P. Ekel, A.T. de Almeida, A benefit-to-cost ratio based approach for portfolio selection under multiple criteria with incomplete preference information, Inf. Sci. 545 (2021) 487–498, https://doi.org/10.1016/j.ins.2020.08.119.
[8] Ö. Şahin Zorluoğlu, Ö. Kabak, A Literature Survey on Project Portfolio Selection Problem, Springer International Publishing, Cham, 2021, pp. 387–411, https://doi.org/10.1007/978-3-030-52406-7_15.
[9] P. Xidonas, R. Steuer, C. Hassapis, Robust portfolio optimization: A categorized bibliographic review, Ann. Oper. Res. 292 (1) (2020) 533–552, https://doi.org/10.1007/s10479-020-03630-8.
[10] M. Saiz, M.A. Lostumbo, A.A. Juan, D. Lopez-Lopez, A clustering-based review on project portfolio optimization methods, International Transactions in Operational Research doi:10.1111/itor.12933..
[11] V. Mohagheghi, S.M. Mousavi, J. Antuchevičienė, M. Mojtahedi, Project portfolio selection problems: a review of models, uncertainty approaches, solution techniques, and case studies, Technol. Econ. Dev. Economy 25 (6) (2019) 1380–1412, https://doi.org/10.3846/tede.2019.11410.
[12] V. Mohagheghi, S. Meysam Mousavi, M. Mojtahedi, Project portfolio selection problems: Two decades review from 1999 to 2019, Journal of Intelligent & Fuzzy Systems 38 (2) (2020) 1675–1689. doi:10.3233/jifs-182847..
[13] I. Yevseyeva, E.B. Lenselink, A. de Vries, A.P. IJzerman, A.H. Deutz, M.T. Emmerich, Application of portfolio optimization to drug discovery, Information Sciences 475 (2019) 29–43. doi:10.1016/j.ins.2018.09.049..
[14] M. Zarjou, M. Khalilzadeh, Optimal project portfolio selection with reinvestment strategy considering sustainability in an uncertain environment: a multi-objective optimization approach, Kybernetes doi:10.1108/K-11-2020-0737..
[15] S. Kudratova, X. Huang, K. Kudratov, S. Qudratov, Corporate sustainability and stakeholder value trade-offs in project selection through optimization modeling: Application of investment banking, Corp. Soc. Responsib. Environ. Manag. 27 (2) (2020) 815–824, https://doi.org/10.1002/csr.1846.
[16] X. Sun, J. Hao, J. Li, Multi-objective optimization of crude oil-supply portfolio based on interval prediction data, Ann. Oper. Res. (2020) 1–29, https://doi.org/10.1007/s10479-020-03701-w.
[17] M.K. Mehlawat, P. Gupta, A.Z. Khan, Portfolio optimization using higher moments in an uncertain random environment, Inf. Sci. 567 (2021) 348–374, https://doi.org/10.1016/j.ins.2021.03.019.
[18] A. Thakkar, K. Chaudhari, A comprehensive survey on portfolio optimization, stock price and trend prediction using particle swarm optimization, Arch. Comput. Methods Eng. 28 (4) (2021) 2133–2164, https://doi.org/10.1007/s11831-020-09448-8.
[19] J.C. Chacon-Hurtado, L. Scholten, Decisi-o-rama: An open-source Python library for multi-attribute value/utility decision analysis, Environ. Modelling Softw. 135 (2021), https://doi.org/10.1016/j.envsoft.2020.104890 104890.
[20] E. Fernandez, E. Lopez, S. Bernal, C.A. Coello Coello, J. Navarro, Evolutionary multiobjective optimization using an outranking-based dominance generalization, Computers & Operations Research 37 (2) (2010) 390–395. doi:10.1016/j.cor.2009.06.004..
[21] S.S. Bastiani, L. Cruz-Reyes, E. Fernandez, C. Gomez, Portfolio optimization from a set of preference ordered projects using an ant colony based multi-objective approach, International Journal of Computational Intelligence Systems 8 (sup2) (2015) 41–53, https://doi.org/10.1080/18756891.2015.1129590.
[22] L. Cruz-Reyes, E. Fernandez, P. Sanchez, C.A. Coello Coello, C. Gomez, Incorporation of implicit decision-maker preferences in multi-objective evolutionary optimization using a multi-criteria classification method, Appl. Soft Comput. 50 (2017) 48–57, https://doi.org/10.1016/j.asoc.2016.10.037.
[23] L. Cruz-Reyes, E. Fernandez, J.P. Sanchez-Solis, C.A. Coello Coello, C. Gomez, Hybrid evolutionary multi-objective optimisation using outranking-based ordinal classification methods, Swarm Evol. Comput. 54 (2020), https://doi.org/10.1016/j.swevo.2020.100652 100652.
[24] E. Fernandez, N. Rangel-Valdez, L. Cruz-Reyes, C. Gomez-Santillan, G. Rivera-Zarate, P. Sanchez-Solis, Inferring parameters of a relational system of preferences from assignment examples using an evolutionary algorithm, Technol. Econ. Dev. Economy 25 (4) (2019) 693–715, https://doi.org/10.3846/tede.2019.9475.
[25] N. Rangel-Valdez, C. Gómez-Santillán, J.C. Hernández-Marín, M.L. Morales-Rodriguez, L. Cruz-Reyes, H.J. Fraire-Huacuja, Parallel designs for metaheuristics that solve portfolio selection problems using fuzzy outranking relations, Int. J. Fuzzy Syst. (2020) 1–13, https://doi.org/10.1007/s40815-019-00794-9.
[26] B. Roy, Multicriteria Methodology for Decision Aiding, Vol. 12 of Nonconvex Optimization and Its Applications, Springer, 1996. doi:10.1007/978-1-4757-2500-1..
[27] G. Miebs, M. Kadziński, Heuristic algorithms for aggregation of incomplete rankings in multiple criteria group decision making, Inf. Sci. 560 (2021) 107–136, https://doi.org/10.1016/j.ins.2021.01.055.
[28] X. Yu, S. Zhang, X. Liao, X. Qi, ELECTRE methods in prioritized MCDM environment, Inf. Sci. 424 (2018) 301–316, https://doi.org/10.1016/j.ins.2017.09.061.
[29] X. Qi, X. Yu, L. Wang, X. Liao, S. Zhang, PROMETHEE for prioritized criteria, Soft. Comput. 23 (22) (2019) 11419–11432, https://doi.org/10.1007/s00500-019-04109-w.
[30] T.J. Lahtinen, R.P. Hämäläinen, C. Jenytin, On preference elicitation processes which mitigate the accumulation of biases in multi-criteria decision analysis, Eur. J. Oper. Res. 282 (1) (2020) 201–210, https://doi.org/10.1016/j.ejor.2019.09.004.
[31] J. Mustajoki, R.P. Hämäläinen, A. Salo, Decision support by interval SMART/SWING—incorporating imprecision in the SMART and SWING methods, Decision Sci. 36 (2) (2005) 317–339, https://doi.org/10.1111/j.1540-5414.2005.00075.x.
[32] E. Fernandez, E. Lopez, F. Lopez, C.A. Coello Coello, Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: the extended NOSGA method, Information Sciences 181 (1) (2011) 44–56. doi:10.1016/j.ins.2010.09.007..
[33] E. Fernandez, E. Lopez, G. Mazcorro, R. Olmedo, C.A. Coello Coello, Application of the non-outranked sorting genetic algorithm to public project portfolio selection, Inform. Sci. 228 (2013) 131–149, https://doi.org/10.1016/j.ins.2012.11.018.
[34] L. Cruz-Reyes, E. Fernandez, N. Rangel-Valdez, A metaheuristic optimization-based indirect elicitation of preference parameters for solving many-objective problems, Int. J. Comput. Intell. Syst. 10 (1) (2017) 56–77, https://doi.org/10.2991/ijcis.2017.10.1.5.
[35] E. Fernández, J.R. Figueira, J. Navarro, An indirect elicitation method for the parameters of the ELECTRE TRI-nB model using genetic algorithms, Appl. Soft Comput. 77 (2019) 723–733, https://doi.org/10.1016/j.asoc.2019.01.050.
[36] N. Rangel-Valdez, E. Fernandez, L. Cruz-Reyes, C. Gomez-Santillan, G. Rivera, R. Florencia, Robustness analysis of an outranking model parameters' elicitation method in the presence of noisy examples, Math. Problems Eng. (2018), https://doi.org/10.1155/2018/2157937.
[37] F. Garcia, J. González-Bueno, F. Guijarro, J. Oliver, R. Tamošiūnienė, Multiobjective approach to portfolio optimization in the light of the credibility theory, Technol. Econ. Dev. Econ. 26 (6) (2020) 1165–1186, https://doi.org/10.3846/tede.2020.13189.
[38] F. Pérez, T. Gómez, R. Caballero, V. Liern, Project portfolio selection and planning with fuzzy constraints, Technol. Forecast. Soc. Chang. 131 (2018) 117–129, https://doi.org/10.1016/j.techfore.2017.07.012.
[39] A.M. Daryani, M.M. Omran, A. Makui, A novel heuristic, based on a new robustness concept, for multi-objective project portfolio optimization, Computers Ind. Eng. 139 (2020), https://doi.org/10.1016/j.cie.2019.106187 106187.
[40] M. Sasaki, A. Laamrani, M. Yamashiro, An interactive genetic algorithm for portfolio optimization considering the decision maker's preference, J. Inform. Optim. Sci. 39 (4) (2018) 989–1008, https://doi.org/10.1080/02522667.2018.1479208.
[41] S. Zolfaghari, S. Mousavi, A novel mathematical programming model for multi-mode project portfolio selection and scheduling with flexible resources and due dates under interval-valued fuzzy random uncertainty, Expert Syst. Appl. (2021), https://doi.org/10.1016/j.eswa.2021.115207 115207.
[42] A. RezaHoseini, S.F. Ghannadpour, M. Hemmati, A comprehensive mathematical model for resource-constrained multi-objective project portfolio selection and scheduling considering sustainability and projects splitting, J. Cleaner Prod. 269 (2020), https://doi.org/10.1016/j.jclepro.2020.122073 122073.
[43] O. Ertenlice, C.B. Kalayci, A survey of swarm intelligence for portfolio optimization: Algorithms and applications, Swarm Evol. Comput. 39 (2018) 36–52, https://doi.org/10.1016/j.swevo.2018.01.009.

[44] G.A. Miller, 11. the magical number seven, plus-or-minus two or some limits on our capacity for processing information, in: Brain Physiology and Psychology, University of California Press, 2020, pp. 175–202, https://doi.org/10.1525/9780520318267-011.
[45] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731, https://doi.org/10.1109/TEVC.2007.892759.
[46] Z. Dong, X. Wang, L. Tang, MOEA/D with a self-adaptive weight vector adjustment strategy based on chain segmentation, Inf. Sci. 521 (2020) 209–230, https://doi.org/10.1016/j.ins.2020.02.056.
[47] Y. Zhang, G.-G. Wang, K. Li, W.-C. Yeh, M. Jian, J. Dong, Enhancing MOEA/D with information feedback models for large-scale many-objective optimization, Inf. Sci. 522 (2020) 1–16, https://doi.org/10.1016/j.ins.2020.02.066.
[48] P. Wang, W. Zhu, H. Liu, B. Liao, L. Cai, X. Wei, S. Ren, J. Yang, A new resource allocation strategy based on the relationship between subproblems for MOEA/D, Inf. Sci. 501 (2019) 337–362, https://doi.org/10.1016/j.ins.2019.06.001.
[49] W.-X. Wang, K.-S. Li, X.-Z. Tao, F.-H. Gu, An improved MOEA/D algorithm with an adaptive evolutionary strategy, Inf. Sci. 539 (2020) 1–15, https://doi.org/10.1016/j.ins.2020.05.082.
[50] L. Cruz-Reyes, C. Medina-Trejo, F. Lopez-Irarragorri, G. Rivera, M. Pérez-Villafuerte, et al., Reduction of decision rules for project explanation on public project portfolio, International Journal of Combinatorial Optimization Problems and Informatics 6 (3) (2015) 5–21. URL:https://ijcopi.org/ojs/article/view/48..