RAIRO-Oper. Res. 55 (2021) S3009–S3034
https://doi.org/10.1051/ro/2020140

RAIRO Operations Research
www.rairo-ro.org

A PYTHON-BASED MULTICRITERIA PORTFOLIO SELECTION DSS

Panos Xidonas[1,*], Haris Doukas[2] and Elissaios Sarmas[2]

Abstract. Our purpose in this article is to develop an integrated portfolio management decision support system, which takes into account the inherent multidimensional nature of the problem, while allowing the DM, *i.e.* investor, to incorporate his/her preferences in the decision process. The proposed DSS has been developed in Python programming language and consists of two components: The first component is associated with the security selection phase, while the second component is associated with the portfolio optimization phase. In the first phase, four discrete multicriteria methods are employed; the PROMETHEE II, the ELECTRE III, the MAUT and the TOPSIS. After the cumulative integration of the results, a series of mathematical programming models are applied in the second phase, that of multicriteria portfolio optimization; a mixed-integer quadratic programming model, a goal programming model, a genetic algorithm model, and a multiobjective PROMETHEE flow model. Finally, the proposed approach is tested through a large-scale illustrative application in several stock markets and various sectors, analyzing simultaneously a very large number of securities.

Mathematics Subject Classification. 90-XX, 90Bxx, 90B50.

Received July 22, 2020. Accepted December 1, 2020.

1. Introduction

The conventional formulation of a typical single-period portfolio selection problem was initially expressed as a non-linear bi-criteria optimization process, where the expected return is maximized and the risk is minimized [12]. One criticism over this model, which has often been addressed both by practitioners and academics, is that it fails to embody the objectives of the decision maker (DM), through the various stages of the decision process [1, 24]. Indeed, the whole portfolio management framework may become very complicated, when multiple investment criteria, such as various components of risk and return, are to be simultaneously taken into consideration [20, 21]. In this case, the portfolio selection process reflects to a multicriteria portfolio selection problem, the computational difficulties and business practice of which, are very complex [8, 17, 18].

Keywords. Operations research (OR), decision support systems (DSS), multicriteria decision making (MCDM), portfolio selection, python.

[1] ESSCA Business School, 55 quai Alphonse Le Gallo, 92513 Paris, France.

[2] NTUA, 9 Heroon Polytechniou Str., 15780 Athens, Greece.

*Corresponding author: panos.xidonas@essca.fr

Article published by EDP Sciences

On this basis, multiple criteria decision making (MCDM), the field of operational research (OR) that deals with problems that involve multiple criteria, can provide the sound methodological basis to resolve the intrinsic multicriteria nature of the portfolio selection problem [14, 22]. Because the classical approach seems to be necessary, but not sufficient to manage portfolio selection efficiently [23], the main contribution of the MCDM framework is associated with two significant issues. Firstly, by exploiting the MCDM benefit there is the potential for more portfolio realistic models to be built, by taking into account, apart from the two basic criteria of return and risk, a number of important other variables, *e.g.* criteria that are based on the theory of fundamental analysis, like for example the security's dividend yield and price to earnings valuation ratio, or criteria related to the market characteristics of the companies, like the alpha and beta coefficients. Second, the classical approach imposes a norm to the investor's behavior that can be restrictive [6], since it cannot incorporate his/her individual goals, personal preferences and attitude towards risk, *i.e.* his/her investment policy statement (IPS). Thus, the MCDM framework has the advantage of taking into account the specific preference system of any particular investor, while allows for synthesizing in a one and single procedure all the theoretical and practical aspects of the portfolio management (PM) theory.

Our purpose in this article is to develop an integrated portfolio management decision support system (DSS), which takes into account the inherent multidimensional nature of the problem, while allowing the DM, *i.e.* investor, to incorporate his/her preferences in the decision process. The proposed DSS has been developed in Python programming language and consists of two components: The first component is associated with the security selection phase, while the second component is associated with the portfolio optimization phase. In the first phase, four discrete multicriteria methods are employed; the PROMETHEE II, the ELECTRE III, the MAUT and the TOPSIS. After the cumulative integration of the results, a series of mathematical programming models are applied in the second phase, that of multicriteria portfolio optimization; a mixed-integer quadratic programming (MIQP) model, a goal programming (GP) model, a genetic algorithm (GA) model, and a multiobjective PROMETHEE flow model. Finally, the proposed approach is tested through a large-scale illustrative application in several stock markets and various sectors, analyzing simultaneously a very large number of securities.

The paper proceeds as follows: In Section 2, we present the proposed methodological framework. In Section 3, we meticulously analyze the corresponding information system. Finally, the empirical results and concluding remarks are given in Sections 4 and 5.

2. Methodological framework

In the existing literature, only few studies have proposed an integrated methodological framework, fully implemented in a decision support system, for modeling the complex problems of security selection and portfolio optimization. In this respect, the proposed model is an innovative methodological framework that fully takes into account the inherent multidimensional nature of the problem, while allowing the decision maker (DM) to express his/her preferences in all the phases of the decision process.

We divide the portfolio management process in two phases: The first phase focuses on portfolio selection, concerning the selection of the securities which prevail as the strongest investment opportunities based on several criteria, while the second phase addresses the portfolio optimization problem, concerning the determination of the most efficient allocation of the available capital to the selected securities in order to satisfy a series of objective functions.

The ultimate goal is the effective management of security portfolios, which undoubtedly constitute one of the most risky market investments. The proposed methodology aspires to combine existing knowledge with a set of theoretical and practical innovations. It is important that the whole methodology should be applied in communication with the DM, as it is necessary that he/she interacts with the model importing his/her preferences during the process. An extensive diagram of the proposed methodological framework is presented in Figure 1.

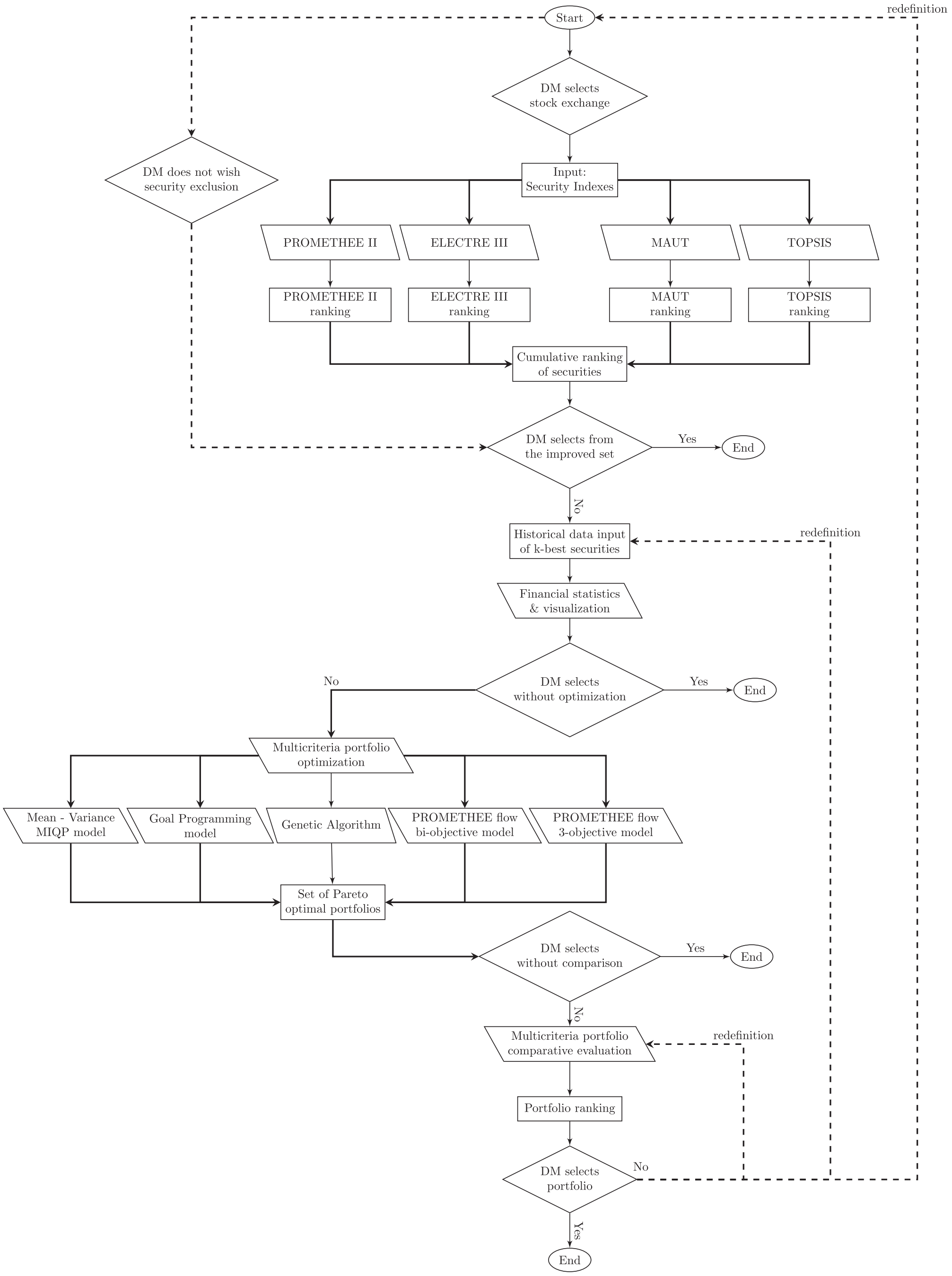
Start
redefinition
DM selects stock exchange
DM does not wish security exclusion
Input: Security Indexes
PROMETHEE II
ELECTRE III
MAUT
TOPSIS
PROMETHEE II ranking
ELECTRE III ranking
MAUT ranking
TOPSIS ranking
Cumulative ranking of securities
DM selects from the improved set
Yes
End
No
Historical data input of k-best securities
redefinition
Financial statistics & visualization
No
DM selects without optimization
Yes
End
Multicriteria portfolio optimization
Mean - Variance MIQP model
Goal Programming model
Genetic Algorithm
PROMETHEE flow bi-objective model
PROMETHEE flow 3-objective model
Set of Pareto optimal portfolios
DM selects without comparison
Yes
End
No
Multicriteria portfolio comparative evaluation
redefinition
Portfolio ranking
DM selects portfolio
No
Yes
End

FIGURE 1. Extensive presentation of the proposed portfolio management methodological framework.

2.1. Phase I: Security selection

The first phase deals with the portfolio selection problem, concerning the construction of a set of securities which are considered as the best investment opportunities. The DM selects the sector and the stock exchange that he/she wishes to get involved with, resulting in a pool of securities which constitute the problem alternatives. The problem of security selection is faced with a series of discrete MCDM methods, under a variety of financial indexes which serve as the criteria of the problem. Each method results in a ranking of the securities and finally the aggregated ranking of the securities can be calculated as the weighted average of the four individual rankings. Consequently, the portfolio should consist of a limited number of securities, excluding securities which have undesirable characteristics.

It is, also, necessary to emphasize the following characteristics of the methodology: (a) The process of security evaluation is based on specific financial indexes (problem criteria), after an extensive study of the existing literature and (b) the companies should be categorized into predefined classes before the evaluation is applied, depending on their activity and the industrial sector they belong to. The necessity of this step derives from the fact that the comparison of financial indexes among companies of different industrial sectors would be a contentious assumption.

Let $A = \{a_1, \ldots, a_n\}$ be a set of n alternatives and let $F = \{f_1, \ldots, f_q\}$ be a consistent family of q criteria. Without loss of generality, we make the assumption that the above criteria should be maximized. The evaluation process of securities is based on a set of suitable financial criteria, which depend on the accounting and economic plans of the companies, as well as on experts' analysis (Tab. 1). The ranking of the securities is based on the following MCDA methods: (a) ELECTRE III, (b) PROMETHEE II, (c) MAUT and (d) TOPSIS.

TABLE 1. Problem criteria.

#	Criteria	Utility	Units
A	Price-to-Earnings Ratio	Minimization	Percentage
B	Earnings per share	Maximization	Percentage
C	Revenue	Maximization	Dollars
D	Beta	Minimization	Fraction
E	Dividend Yield	Maximization	Percentage
F	Monthly technical recommendation	Maximization	Rank
G	Year-to-date performance	Maximization	Percentage
H	1-year performance	Maximization	Percentage

2.1.1. ELECTRE III

The ELECTRE family in MCDA problems is based on the concept of outranking relationship. An alternative a_1 outranks a_2 if and only if there is sufficient evidence to believe that a_1 is better than a_2 or at least a_1 is as good as a_2. More specifically, ELECTRE III method, originally presented by Roy [15], is used for ranking problems, using a structured procedure to calculate the outranking relationship between each pair of alternatives. It includes a preference threshold, an indifference threshold and a veto threshold.

Let $q(f_i)$ and $p(f_i)$ represent the indifference and preference thresholds for each criterion $f_i,\ i = 1, \ldots, q$, respectively, and let P denote a strong preference, Q denote a weak preference and I denote indifference between a_1 and a_2 for criterion k. If $f_k(a_i) \geq f_k(a_j)$, then:

$$\begin{aligned} f_k(a_i) > f_k(a_j) + p(f_k) &\Leftrightarrow a_1 P a_2 \\ f_k(a_j) + q(f_k) < f_k(a_i) < f_k(a_j) + p(f_k) &\Leftrightarrow a_1 Q a_2 \\ f_k(a_j) < f_k(a_i) < f_k(a_j) + q(f_k) &\Leftrightarrow a_1 I a_2. \end{aligned}$$

Algorithm 1: ELECTRE III Algorithm.

```
input: $n$ (alternatives), $q$ (criteria), $f_i(a_j)$ (evaluation table);
for all pairs of alternatives $a_i,\ a_j,\ i,j \in \{1,\dots,n\}$ do
    for all criteria $f_k,\ k \in \{1,\dots,q\}$ do
        compute outranking degree $C_k(a_i,a_j)$
    end
end
for all pairs of alternatives $a_i,\ a_j,\ i,j \in \{1,\dots,n\}$ do
    compute concordance index $C(a_i,a_j)$
end
for all pairs of alternatives $a_i,\ a_j,\ i,j \in \{1,\dots,n\}$ do
    for all criteria $f_k,\ k \in \{1,\dots,q\}$ do
        compute discordance index $D_k(a_i,a_j)$
    end
end
for all pairs of alternatives $a_i,\ a_j,\ i,j \in \{1,\dots,n\}$ do
    compute degree of outranking relationship $S(a_i,a_j)$
end
for all alternatives $a_i,\ i \in \{1,\dots,n\}$ do
    compute concordance credibility index $\phi^+(a_i)$
    compute discordance credibility index $\phi^-(a_i)$
    compute net credibility index $\phi(a_i)$
end
$FinalRanking$ = SortDesc($\phi$)
Result: FinalRanking
```

The outranking degree $C_k(a_i,a_j)$, $(0 \leq C_k(a_i,a_j) \leq 1)$ of the alternative a_i and the alternative a_j for criterion f_k is calculated according to the preference definitions (linear interpolation):

$$C_k(a_i,a_j) = \begin{cases} 0 & \text{if} \quad f_k(a_j) - f_k(a_i) > p(f_k) \\ 1 & \text{if} \quad f_k(a_j) - f_k(a_i) \leq q(f_k) \\ \dfrac{p(f_k) + f_k(a_i) - f_k(a_j)}{p(f_k) - q(f_k)} & \text{otherwise.} \end{cases} \tag{2.1}$$

The concordance index $C(a_i,a_j)$ is computed for each pair of alternatives a_i, a_j, as follows:

$$C(a_i,a_j) = \frac{\sum_{k=1}^{q} w_k C_k(a_i,a_j)}{\sum_{k=1}^{q} w_k}. \tag{2.2}$$

Let $v(f_k)$ represent the veto threshold for criterion f_k. The veto threshold rejects the possibility of $a_i S a_j$ if, for any criterion f_k, the relationship $f_k(a_j) > f_k(a_i) + v(f_k)$ is satisfied. The discordance index $D(a_i,a_j)$, $(0 \leq D_k(a_i,a_j) \leq 1)$ for each criterion is defined as follows:

$$D_k(a_i,a_j) = \begin{cases} 0 & \text{if} \quad f_k(a_j) - f_k(a_i) \leq p(f_k) \\ 1 & \text{if} \quad f_k(a_j) - f_k(a_i) > v(f_k) \\ \dfrac{f_k(a_j) - f_k(a_i) - p(f_k)}{v(f_k) - p(f_k)} & \text{otherwise.} \end{cases} \tag{2.3}$$

Let $J(a_i, a_j)$ represent the set of criteria for which $D_k(a_i, a_j) > C(a_i, a_j)$. The reliability index $S(a_i, a_j)$ is:

$$S(a_i, a_j) = \begin{cases} C(a_i, a_j) & \text{if} \quad D_k(a_i, a_j) \leq C(a_i, a_j) \ \forall k \in J \\ C(a_i, a_j) \times \displaystyle\prod_{k \in J(a_i, a_j)} \frac{1 - D_k(a_i, a_j)}{1 - C(a_i, a_j)} & \text{otherwise.} \end{cases} \tag{2.4}$$

The concordance credibility degree $\phi^+(a_i)$ is an indicator that measures how an alternative a_i dominates all the other alternatives [7, 11]. The definition of concordance credibility degree is:

$$\phi^+(a_i) = \sum_{x \in A} S(a_i, x). \tag{2.5}$$

The discordance credibility degree $\phi^-(a_i)$ is an indicator that measures how an alternative a_i is dominated by all the other alternatives. The definition of discordance credibility degree is:

$$\phi^-(a_i) = \sum_{x \in A} S(x, a_i). \tag{2.6}$$

Finally, the net credibility degree $\phi(a_i)$ is an indicator of the value of the alternative a_i. A higher net credibility degree implies a better alternative. The definition of the net credibility degree for an alternative a_i is:

$$\phi(a_i) = \phi^+(a_i) - \phi^-(a_i). \tag{2.7}$$

The ELECTRE III final ranking is obtained by ordering the alternatives according to the decreasing values of the net flow scores.

2.1.2. PROMETHEE II

PROMETHEE is the product of Brans and Vincke [3] and Brans *et al.* [4]. Insightful applications of PROMETHEE are found in the review of Behzadian *et al.* [2]. One of the creators of PROMETHEE, Bertrand Mareschal, maintains a list of references on his website www.promethee-gaia.net, which as of January 2020 contained over 2200 references.

Once again, consider a problem with m actions or alternatives, $A = \{a_1, \ldots, a_m\}$, which are to be evaluated on a set of k criteria, $F = \{f_1, \ldots, f_k\}$. Suppose, without loss of generality, that all criteria are to be maximized. For each criterion j and for each pair of actions (a, b), assume the DM is able to express his/her degree of preference in the form of $P_j(a, b) \in [0, 1]$, where the order of notation is that action a is preferred to b based upon the difference $d_j(a, b) = f_j(a) - f_j(b)$. The degree of preference is obtained using a preference function chosen by the DM. The preference functions that have been proposed are: (a) the usual criterion, (b) the U-shaped criterion, (c) the V-shaped criterion, (d) the level criterion, (e) the V-shaped criterion with indifference region, and (f) the Gaussian criterion. These six types are easy to define and have a clear intuition for the DM.

Depending on the function chosen, threshold values may be required. For example, if the DM selects a V-shaped criterion with indifference region, the DM is then required to specify the threshold values of p_j (strict preference) and q_j (indifference). If the difference between the evaluation of a and b on the j_{th} criterion is smaller than the indifference threshold q_j, then neither action is preferred. If the difference between the evaluations of a and b is greater than the preference threshold, p_j, $d_j(a, b) > p_j$, then action a is preferred to action b.

In order to evaluate how much action a is preferred to b over all criteria, the preference index $\pi(a, b)$ is calculated using a weighted sum of the degrees of preference $P_j(a, b)$. The weights, $w_j > 0$, are to reflect the importance of each criterion in the decision. That is, the greater the weight, the more important is the criterion. The preference indices are:

$$\pi(a, b) = \sum_{j=1}^{k} w_j \times P_j(a, b) \tag{2.8}$$

Algorithm 2: PROMETHEE II Algorithm.

```
input: $n$ (alternatives), $q$ (criteria), $f_i(a_j)$ (evaluation table), $P_i$ (preference function);
for all criteria $f_k$, $k \in \{1, \dots, q\}$ do
    $d_k(a_i, a_j) = f_k(a_i) - f_k(a_j)$
end
for all pairs of alternatives $a_i$, $a_j$, $i, j \in \{1, \dots, n\}$ do
    $\pi_k(a_i, a_j) = P_k[d_k(a_i, a_j)]$
end
for all criteria $f_k$, $k \in \{1, \dots, q\}$ do
    $\pi(a_i, a_j) = \mathbf{sum}(\pi_k(a_i, a_j))$
end
for all pairs of alternatives $a_i$, $a_j$, $i, j \in \{1, \dots, n\}$ do
    $\phi^+(a_i) = \mathbf{sum}(\pi(a_i, a_j))$
    $\phi^-(a_i) = \mathbf{sum}(\pi(a_j, a_i))$
end
for all alternatives $a_i$, $i \in \{1, \dots, n\}$ do
    $\phi(a_i) = \phi^+(a_i) - \phi^-(a_i)$
end
$FinalRanking = \mathbf{SortDesc}(\phi)$
Result: FinalRanking
```

and

$$\pi(b,a) = \sum_{j=1}^{k} w_j \times P_j(b,a) \tag{2.9}$$

where $\pi(a,b)$ expresses the degree to which a is preferred over b for all criteria, and $\pi(b,a)$ represents how much b is preferred to a. As each action is compared with other $m-1$ actions, positive ϕ^+ and negative ϕ^- outranking flows can be defined as follows:

$$\phi^+(a) = \frac{1}{m-1} \sum_{x \in A} \pi(a,x) \tag{2.10}$$

$$\phi^-(a) = \frac{1}{m-1} \sum_{x \in A} \pi(x,a). \tag{2.11}$$

The positive flow $\phi^+(a)$ expresses how much alternative a outranks all other $m-1$ alternatives, thus it represents the global preference for action a in comparison to all the other actions. The higher the value of $\phi^+(a)$, the better the alternative is. The negative flow, $\phi^-(a)$, expresses how alternative a is outranked by all other $m-1$ alternatives, thus it represents the global weakness of a in comparison to all the other actions. The smaller $\phi^-(a)$, the better the alternative is. Based on the positive and negative outranking flows, the PROMETHEE I partial ranking is defined as follows:

$$a \ P \ b \quad \text{iff} = \begin{cases} \phi^+(a) > \phi^+(b) & \text{and } \phi^-(a) < \phi^-(b) \\ & \text{or} \\ \phi^+(a) = \phi^+(b) & \text{and } \phi^-(a) < \phi^-(b) \\ & \text{or} \\ \phi^+(a) > \phi^+(b) & \text{and } \phi^-(a) = \phi^-(b) \end{cases} \tag{2.12}$$

$$a \ I \ b \quad \text{iff} = \phi^+(a) = \phi^+(b) \text{ and } \phi^-(a) = \phi^-(b) \tag{2.13}$$

$$a \ R \ b \quad \text{iff} = \begin{cases} \phi^+(a) > \phi^+(b) & \text{and } \phi^-(a) > \phi^-(b) \\ & \text{or} \\ \phi^+(a) < \phi^+(b) & \text{and } \phi^-(a) < \phi^-(b). \end{cases} \tag{2.14}$$

The positive and the negative flows can be combined to obtain the net outranking flow, defined as follows:

$$\phi(a) = \phi^{+}(a) - \phi^{-}(a). \tag{2.15}$$

PROMETHEE II exploits the above net flow in order to provide a complete ranking of actions, from best to worst; the higher the value of $\phi(a)$, the better the alternative is:

$$\begin{cases} aPb & \text{iff} \quad \phi(a) > \phi(b) \\ aIb & \text{iff} \quad \phi(a) = \phi(b). \end{cases} \tag{2.16}$$

2.1.3. MAUT

Multi-Attribute Utility Theory (MAUT), developed by Keeney and Raiffa [10], is a structured methodology which was originally designed in order to handle the trade-offs among multiple objective functions. MAUT belongs to the family of multicriteria utility theory, it can be considered as an additive value function and it has the advantage that it is adaptable to the profile of the DM, as it can describe optimistic and pessimistic behaviors.

Algorithm 3: MAUT Algorithm.

```
input: $n$ (alternatives), $q$ (criteria), $w_i$ (weights), $f_i(a_j)$ (evaluation table);
for all alternatives $a_i$, $i \in \{1, \dots, n\}$ do
    for all criteria $f_k$, $k \in \{1, \dots, q\}$ do
        compute normalized decision matrix $x_k(a_i)$
    end
end
for all alternatives $a_i$, $i \in \{1, \dots, n\}$ do
    for all criteria $f_k$, $k \in \{1, \dots, q\}$ do
        compute integrated DM's attitude $u_k(a_i)$
    end
end
for all alternatives $a_i$, $i \in \{1, \dots, n\}$ do
    compute utility $U(i) = \mathbf{sum}(w_k u_k(a_i))$
end
$FinalRanking = \mathbf{SortDesc}(U)$
Result: FinalRanking
```

Let $f_k(a_{\min})$, $f_k(a_{\min})$ represent the minimum and maximum value for criterion k. The evaluation table is normalized, as follows:

For maximization criteria:

$$x_k(a_i) = \frac{f_k(a_i) - f_k(a_{\min})}{f_k(a_{\max}) - f_k(a_{\min})}. \tag{2.17}$$

For minimization criteria:

$$x_k(a_i) = \frac{f_k(a_{\max}) - f_k(a_i)}{f_k(a_{\max}) - f_k(a_{\min})}. \tag{2.18}$$

The attitude of the decision-maker is incorporated into the normalized decision matrix, as follows:

$$u_k(a_i) = \frac{1 - e^{cx_i}}{1 - e^{c}}$$

where c is an index that represents the attitude of the decision maker. The Utility function is computed as follows:

$$U_i = \sum_{k=1}^{q} w_k u_k(a_i). \tag{2.19}$$

The MAUT final ranking is obtained by ordering the alternatives according to the decreasing values of the utility function.

2.1.4. TOPSIS

TOPSIS is the product of Hwang and Yoon [9] and Chen and Hwang [5]. TOPSIS stands for Technique for Order of Preference by Similarity to Ideal Solution. Representative applications in a number of areas can be found in the reviews of Palczewski and Sałabun [13] and Salih *et al.* [16].

Algorithm 4: TOPSIS Algorithm.

```
input: $n$ (alternatives), $q$ (criteria), $w_i$ (weights), $f_i(a_j)$ (evaluation table);
for all alternatives $a_i$, $i \in \{1,\dots,n\}$ do
    for all criteria $f_k$, $k \in \{1,\dots,q\}$ do
        compute normalized score $r_k(a_i)$
    end
end
for all alternatives $a_i$, $i \in \{1,\dots,n\}$ do
    for all criteria $f_k$, $k \in \{1,\dots,q\}$ do
        compute weighted normalized score $t_k(a_i)$
    end
end
compute positive ideal solution $A^+$
compute negative ideal solution $A^-$
for all alternatives $a_i$, $i \in \{1,\dots,n\}$ do
    compute separation distance from positive ideal solution $S^+(i)$
    compute separation distance from negative ideal solution $S^-(i)$
    compute relative closeness to the positive ideal solution $C^-(i)$
end
$FinalRanking = \mathbf{SortDesc}(C)$
Result: FinalRanking
```

Consider a problem with m alternatives numbered 1 to m, and k criteria numbered 1 to k. Let each alternative be evaluated with respect to each criterion. This yields a decision matrix $\mathbf{X} = (x_{ij})_{m\times k}$, where x_{ij} in $\mathbf{X}$ is the value assigned to alternative i by criterion j. According to TOPSIS, the first step is to make the criteria dimensionless. This is done by normalization, which is accomplished by re-scaling the columns of $\mathbf{X}$, that is, by converting each x_{ij} value into an r_{ij} as follows:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{q=1}^{m} x_{qj}^2}}, \quad i = 1,\dots,m, j = 1,\dots,k. \tag{2.20}$$

Then each r_{ij} is converted into a v_{ij} value as follows:

$$v_{ij} = w_j r_{ij}, \quad i = 1,\dots,m, j = 1,\dots,k \tag{2.21}$$

where the w_j are the criterion weights obtained from the weighting system. In this way, the ith row of $\mathbf{V} = (v_{ij})_{m\times k}$ is the weighted normalized criterion vector of the ith alternative.

The next task of TOPSIS is to construct the ideal (zenith) and anti-ideal (nadir) solutions of the problem. The simplest case is that the ideal and anti-ideal points are fixed by the decision maker, but this should be avoided as it would imply that the decision maker can actually make a credible elicitation of the two points and it would add more subjectivity to the procedure. A better approach is to construct the components of the ideal solution $v^+ = (v_1^+, \ldots, v_k^+)$ by means of:

$$v_j^+ = \begin{cases} \max\ \{v_{ij} | i = 1, \ldots, m\} & \text{if } j \text{ is a benefit criterion} \\ \min\ \{v_{ij} | i = 1, \ldots, m\} & \text{if } j \text{ is a cost criterion} \end{cases} \tag{2.22}$$

and the components of the anti-ideal solution $v^- = (v_1^-, \ldots, v_k^-)$ by means of:

$$v_j^- = \begin{cases} \min\ \{v_{ij} | i = 1, \ldots, m\} & \text{if } j \text{ is a benefit criterion} \\ \max\ \{v_{ij} | i = 1, \ldots, m\} & \text{if } j \text{ is a cost criterion.} \end{cases} \tag{2.23}$$

Now it is necessary to calculate how far the weighted normalized criterion vector of each alternative is from the ideal solution. This is done by computing:

$$D_i^+ = \sqrt{\sum_{j=1}^{k} (v_{ij} - v_j^+)^2},\ i = 1, \ldots, m,\ j = 1, \ldots, m. \tag{2.24}$$

Similarly, it is necessary to calculate how far the weighted normalized criterion vector of each alternative is from the anti-ideal solution. This is done by computing:

$$D_i^- = \sqrt{\sum_{j=1}^{k} \left(v_{ij} - v_j^-\right)^2},\ i = 1, \ldots, m,\ j = 1, \ldots, m. \tag{2.25}$$

Using these two distances, we compute each alternative's relative closeness to the ideal solution:

$$C_i^+ = \frac{D_i^+}{D_i^+ - D_i^-}\cdot \tag{2.26}$$

2.1.5. Cumulative ranking

After the application of the four MCDA methods, four ranking lists of the alternatives have been formulated. However, the decision-maker should be provided with a final ranking in order to select the k-best securities among them. The suggested methodology to combine the four rankings is the weighted average measure. More specifically, each ranking method is provided with a weighting factor $w_k,\ k = \{1, 2, 3, 4\}$. The cumulative ranking index CR_i for alternative i is calculated as follows:

$$\mathrm{CR}_i = \sum_{k=1}^{4} w_k r_k \tag{2.27}$$

where r_k represents the ranking of alternative i in method k.

2.2. Phase II: Portfolio optimization

The second phase involves the problem of portfolio optimization. The conventional approach to this problem has been a bi-criteria model where the expected return should be maximized while the portfolio risk should be minimized. The proposed methodological framework sets a series of models to address the problem. Firstly, a bi-objective integer programming model is formulated based on the mean-variance approach, where additional integer constraints are imposed in order to control the weighting factor of each security. Secondly, a goal programming methodology is introduced. Additionally, an implementation of a genetic algorithm for portfolio optimization is presented. Finally, a three-objective optimization approach is introduced, involving the net flow of PROMETHEE method.

2.2.1. MIQP model

The conventional formulation of the portfolio optimization problem was expressed as a nonlinear bi-criteria optimization problem. According to Markowitz, the portfolio expected return should be maximized and the portfolio risk should be minimized. The risk is quantified as the variance of portfolio returns, resulting in a quadratic programming problem.

Let $E(R_i)$ be the expected return and w_i the weighting factor of security i. The first objective concerns the portfolio expected return and is expressed as follows:

$$\max_{w} E(R_p) = \sum_{i=1}^{m} w_i E(R_i) \tag{2.28}$$

where m is the total number of securities. Let σ_{ij} be the covariance between securities i and j. The second objective concerns the portfolio risk which is expressed as follows:

$$\min_{w} \sigma_P^2 = \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} w_i w_j \sigma_{ij}. \tag{2.29}$$

Moving to the model's set of constraints, the corresponding expression for capital completeness is introduced:

$$\sum_{i=1}^{m} w_i = 1 \tag{2.30}$$

while the restriction concerning no short sales allowance is:

$$w_i \geq 0. \tag{2.31}$$

Thus, a bi-objective quadratic optimization problem is formulated, which is presented as follows:

$$\begin{aligned} &\max_{w}\ E(r_P) = \sum_{i=1}^{m} w_i E(r_i) \\ &\min_{w}\ \sigma_P^2 = \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} w_i w_j \sigma_{ij} \\ &\text{s.t.} \sum_{i=1}^{m} w_i = 1 \\ &\qquad w_i \geq 0, \qquad\qquad i = 1, 2, \ldots, m. \end{aligned} \tag{2.32}$$

The problem is solved parametrically for a predefined parameter of the portfolio expected return. Let R be the portfolio expected return. The problem is transformed into a linear programming problem with an additional restriction concerning the expected return, which is presented below:

$$\begin{aligned} &\min_{w}\ \sigma_P^2 = \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} w_i w_j \sigma_{ij} \\ &\text{s.t.} \sum_{i=1}^{m} w_i E(r_i) = R \\ &\qquad \sum_{i=1}^{m} w_i = 1 \\ &\qquad w_i \geq 0, \qquad\qquad i = 1, 2, \ldots, m. \end{aligned} \tag{2.33}$$

In the proposed methodology, a variation of the conventional mean-variance model, originally introduced by Xidonas and Mavrotas [20] is presented. The model is equipped with binary variables b_i, in order to control the existence of each security in the portfolio. More specifically, if $b_i = 1$ the i_{th} security participates in the portfolio, else if $b_i = 0$ it does not. The use of binary variables allows the direct determination of the number of securities in the portfolio, producing the following cardinality constraint equation:

$$S_L \leq \sum_{i=1}^{m} b_i \leq S_U$$

where S_L and S_U are the minimum and maximum number of securities allowed to participate in the portfolio.

Moreover, the diversification of the portfolio can be supported constraining the upper bound of each security weight. In order to determine the lower and upper weighting factor of each security the following restrictions are introduced:

$$\begin{aligned} w_i - W_L \times b_i \geq 0, &\quad i = 1, 2, \ldots, m \\ w_i - W_U \times b_i \leq 0, &\quad i = 1, 2, \ldots, m \end{aligned}$$

where W_L and W_U are the minimum and maximum security weights that are allowed in the portfolio.

Thus, the following multiobjective integer programming (MOIP) problem is formulated:

$$\begin{aligned} &\max_{w} E(r_P) = \sum_{i=1}^{m} w_i E(r_i) \\ &\min_{w} \sigma_P^2 = \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} w_i w_j \sigma_{ij} \\ &\text{s.t.} \qquad \sum_{i=1}^{m} b_i \leq S_U \\ &\qquad\qquad \sum_{i=1}^{m} b_i \geq S_L \\ &\qquad\qquad \sum_{i=1}^{m} w_i = 1 \\ &w_i - W_L \times b_i \geq 0 \qquad i = 1, 2, \ldots, m \\ &w_i - W_U \times b_i \leq 0 \qquad i = 1, 2, \ldots, m. \end{aligned} \tag{2.34}$$

Similarly, the solution is determined parametrically for a predefined parameter of the portfolio expected return. The problem is transformed into a mixed-integer quadratic programming (MIQP) problem with an additional restriction for the expected return, which is presented below:

$$\begin{aligned} &\min_{w} \sigma_P^2 = \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} w_i w_j \sigma_{ij} \\ &\text{s.t.} \quad \sum_{i=1}^{m} w_i E(r_i) = R \\ &\qquad\qquad \sum_{i=1}^{m} b_i \leq S_U \\ &\qquad\qquad \sum_{i=1}^{m} b_i \geq S_L \\ &\qquad\qquad \sum_{i=1}^{m} w_i = 1 \\ &w_i - W_L \times b_i \geq 0 \qquad i = 1, 2, \ldots, m \\ &w_i - W_U \times b_i \leq 0 \qquad i = 1, 2, \ldots, m. \end{aligned} \tag{2.35}$$

The formulated problem can be solved parametrically considering the parameter R, thus producing the efficient frontier of solutions.

2.2.2. Goal programming model

The decision variables of the goal programming problem will be the weighting factor w of each security. Let w_i be the weighting factor of the i_{th} security. The following goals are defined:

(1) The beta index of the portfolio β_P, which is defined as the weighted sum of the individual beta index of each security, is given the target value β_G.

$$\beta_P = \sum_{i=1}^{m} w_i \times \beta_i.$$

(2) The portfolio dividend yield, which is defined as the weighted sum of the individual dividend yield of each security, is given the target value DY_G

$$\mathrm{DY}_P = \sum_{i=1}^{m} w_i \times \mathrm{DY}_i.$$

(3) The portfolio PROMETHEE flow, which is defined as the weighted sum of the individual flow of each security, is given the target value ϕ_G

$$\phi_P = \sum_{i=1}^{m} w_i \times \phi_i.$$

Introducing the deviational (or slack) variables d_i^+, d_i^- the problem is formulated as follows:

$$\begin{aligned}
\min_{d^+,d^-} \quad & \frac{w_1^+ d_1^+ + w_1^- d_1^-}{\beta_G} + \frac{w_2^+ d_2^+ + w_2^- d_2^-}{\mathrm{DY}_G} + \frac{w_3^+ d_3^+ + w_3^- d_3^-}{\phi_G} \\
\text{s.t.} \quad & \sum_{i=1}^{m} w_i \beta_i + d_1^- - d_1^+ = \beta_G \\
& \sum_{i=1}^{m} w_i \mathrm{DY}_i + d_2^- - d_2^+ = \mathrm{DY}_G \\
& \sum_{i=1}^{m} w_i \phi_i + d_3^- - d_3^+ = \phi_G \\
& \sum_{i=1}^{m} b_i \leq S_U \\
& \sum_{i=1}^{m} b_i \geq S_L \\
& \sum_{i=1}^{m} w_i = 1 \\
& w_i - W_L \times b_i \geq 0 \qquad i = 1, 2, \ldots, m \\
& w_i - W_U \times b_i \leq 0 \qquad i = 1, 2, \ldots, m
\end{aligned} \tag{2.36}$$

where w^+, w^- are the weights of the deviational variables. Attention is needed not to confuse the weighting factor w of each security with the overachievement and underachievement weights w^+, w^- of the deviational variables.

2.2.3. Genetic algorithm model

The philosophy of this problem differs from all the other, as the weighting factors are determined with the assistance of a market index. Additionally, another significant difference is that in this case there is a unique solution of the optimization problem, while the other problems result in a set of Pareto efficient solutions.

More specifically, let m be the market index and r_{im} the return of index m in period i. Let us define, also, m securities and r_{ij} the return of security j in period i. The portfolio return in period i is equal to:

$$r_{ip} = \sum_{j=1}^{m} w_j r_{ij} \tag{2.37}$$

where w_j is the proportion of the security j in portfolio p.

We say that *the constructed portfolio beats the market index* in period i if the following inequation applies:

$$r_{ip} \geq r_{im}. \tag{2.38}$$

Therefore, the genetic algorithm takes as input the historical data for T periods and attempts to maximize the percentage of cases that the constructed portfolio beats the market index. This claim is quantified as follows:

$$\begin{aligned} &\max_{w_i} \sum_{i=1}^{T} \frac{b_i}{T} \\ &\text{s.t.} \sum_{i=1}^{N} w_i &= 1 \\ &w_i &\geq 0 \quad \forall i = 1, 2, \ldots N \end{aligned} \tag{2.39}$$

where b_i is a binary variable that takes the value 1 if $r_{ip} \geq r_{im}$ in period i, else it takes the value 0.

The selected genetic algorithm that we use in the proposed methodology is called *differential evolution*. Differential evolution [19] is a stochastic population based method developed by Storn and Price, which is used in order to find the global minimum of a multivariate function. Every time that the algorithm examines the population, it mutates each candidate solution by mixing it with other candidate solutions in order to create a trial candidate. The most common strategy to create a trial candidate is the *best1bin* strategy, where two members of the population are randomly chosen and their difference is used to mutate the best member b_0:

$$b' = b_0 + \text{mutation} * (\text{population}\,[i] - \text{population}\,[j]). \tag{2.40}$$

Then, a trial vector is constructed which is filled with parameters either from b' or the original candidate, generated with a binomial distribution (meaning that we generate a random number between 0 and 1; if the number is less than a predefined constant then the parameter is filled from b', otherwise it is filled from the original candidate. After the construction of the trial candidate, its fitness is assessed. If the trial candidate is better than the original candidate, it replaces the original candidate. If it is also better than the best overall candidate, it also replaces that. By default the best solution vector is updated continuously within a single iteration. Finally, the possibility of finding a global minimum is improved by increasing the population size, which results in widening the search radius, but slowing the convergence of the algorithm.

2.2.4. Multiobjective PROMETHEE flow model

This approach connects the concept of the PROMETHEE method of multicriteria decision analysis with a measure of risk, in this case Beta index. Finally, let us introduce the dividend yield as a third objective function.

Additionally, the problem could be equipped with integer decision variables in order to control the number of securities with non-zero proportion to the portfolio. Based on the above observations the optimization problem is formulated as follows:

$$\begin{aligned}
&\max_{w}\ \phi_P = \sum_{i=1}^{m} w_i\phi_i \\
&\min_{w}\ \beta_P = \sum_{i=1}^{m} w_i\beta_i \\
&\max_{w}\ \mathrm{DY}_P = \sum_{i=1}^{m} w_i\mathrm{DY}_i \\
&\text{s.t.} \qquad \sum_{i=1}^{m} b_i \leq S_U \\
&\qquad\qquad \sum_{i=1}^{m} b_i \geq S_L \\
&\qquad\qquad \sum_{i=1}^{m} w_i = 1 \\
&\qquad w_i - W_L \times b_i \geq 0 \quad i = 1, 2, \ldots, m \\
&\qquad w_i - W_U \times b_i \leq 0 \quad i = 1, 2, \ldots, m.
\end{aligned} \tag{2.41}$$

This is a multiobjective programming problem in 3 dimensions with integer variables. It is obvious that the computational complexity of the above problem becomes huge, especially if the number of securities is significantly large. A variety of methodologies have been proposed for problems like this such as the e-constraint method, which faces the problem as a 1-objective optimization problem, transforming the remaining objectives into constraints. However, in this paragraph a methodology based on goal programming and the MINIMAX objective is proposed to solve this MOLP problem.

The first step of the methodology is to solve the model to find the solution that minimizes each objective function ignoring the other objective function. If we solve the problem for all objective functions, we obtain the optimal value for each objective, respectively.

The next step is to formulate the goal programming problem. The target value for each objective function is set equal to the optimal value calculated in the previous step. The percentage deviation from this target can be computed as follows:

$$t = \frac{\text{actual value} - \text{target value}}{\text{target value}} \tag{2.42}$$

for goals derived from minimization objectives,

$$t = \frac{\text{target value} - \text{actual value}}{\text{target value}} \tag{2.43}$$

for goals derived from maximization objectives.

Therefore, having determined the goals of the GP model, the last step involves the configuration of the objective function. The implementation of the objective function is made with the introduction of a MINIMAX variable Q which should be minimized. If w_i is the offset for the i_{th} objective function, the goal is to minimize the maximum of $w_i t_i$. The above claim is expressed with the following mathematical equation:

$$\begin{aligned}
&\min Q \\
&\quad \text{s.t. } w_1 t_1 \leq Q \\
&\qquad\quad w_2 t_2 \leq Q \\
&\qquad\quad w_3 t_3 \leq Q.
\end{aligned} \tag{2.44}$$

Thus, a set of Pareto optimal solutions derives from the adjustment of the weighting factors w_i.

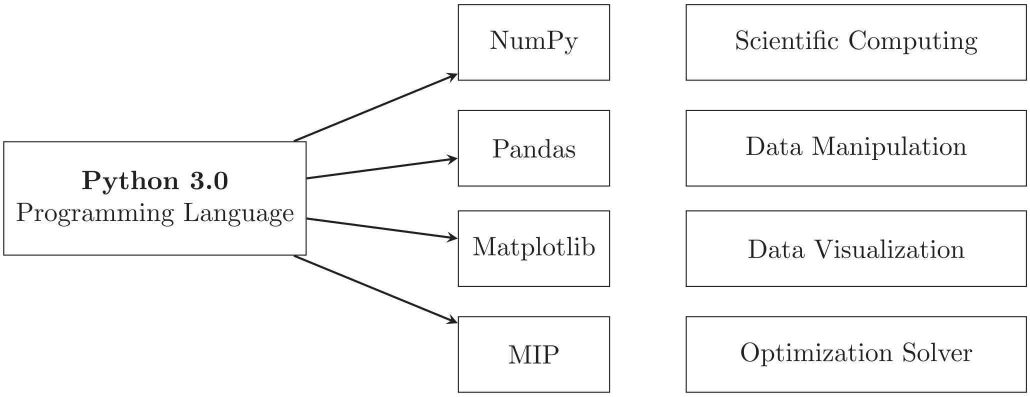

FIGURE 2. A general framework of the programming language and the additional packages used for the development of the information system.

3. Information system

3.1. System architecture

The Portfolio Management process includes the analysis of a large volume of financial information and data. In order to support real-time investment decision it is important to analyse a continuous load of information for all the securities which can be possible investment placements. Therefore, the development of modern and user-friendly software applications is really important in order to support the analysis of the necessary financial information.

In this context, we have developed a Python-based information which implements the proposed methodology in real time. In this system, all the discussed methods of both phases are developed in detail in order to support the application of the discussed portfolio management approach. The information system is implemented in Python 3 programming language which makes it available for Windows, Linux and macOS operating systems. It also incorporates a series of open-source libraries (Fig. 2) such as: (a) *Matplotlib:* a Python plotting library which produces high quality figures in a variety of formats, (b) *Pandas:* an open source library which provides high performance, useful data structures and data analysis tools, (c) *NumPy:* the fundamental library for scientific computing with Python and (d) *MIP:* a library of Python tools for the modeling and solution of Mixed-Integer Linear programming problems.

3.2. Interaction diagrams

The innovative features of the developed information system for portfolio construction and selection system are the following: Firstly, it facilitates high level of interaction between the DM and the proposed methodology, as the model is structured each time according to the DM's preferential system and investment strategy. Thus the role of the investor is improved, opposing to the conventional decision support systems which do not take into consideration the investor's preferences. Secondly, it combines the whole methodological framework from security selection until portfolio optimization. All the provided techniques are based on highly sophisticated models and algorithms, facing the portfolio management process with a well-structured scientific background. Finally, the system is fully configurable, making possible to operate in real-time problems. It can easily be connected with the proper databases and handle a huge volume of financial data.

In Figure 3, the sequential diagram is presented, describing the interactions among each part of the system arranged in time sequence. It depicts the sequence of messages which are exchanged between the different parts of the system (decision-maker, database etc.), in order to achieve the functionality of each scenario.

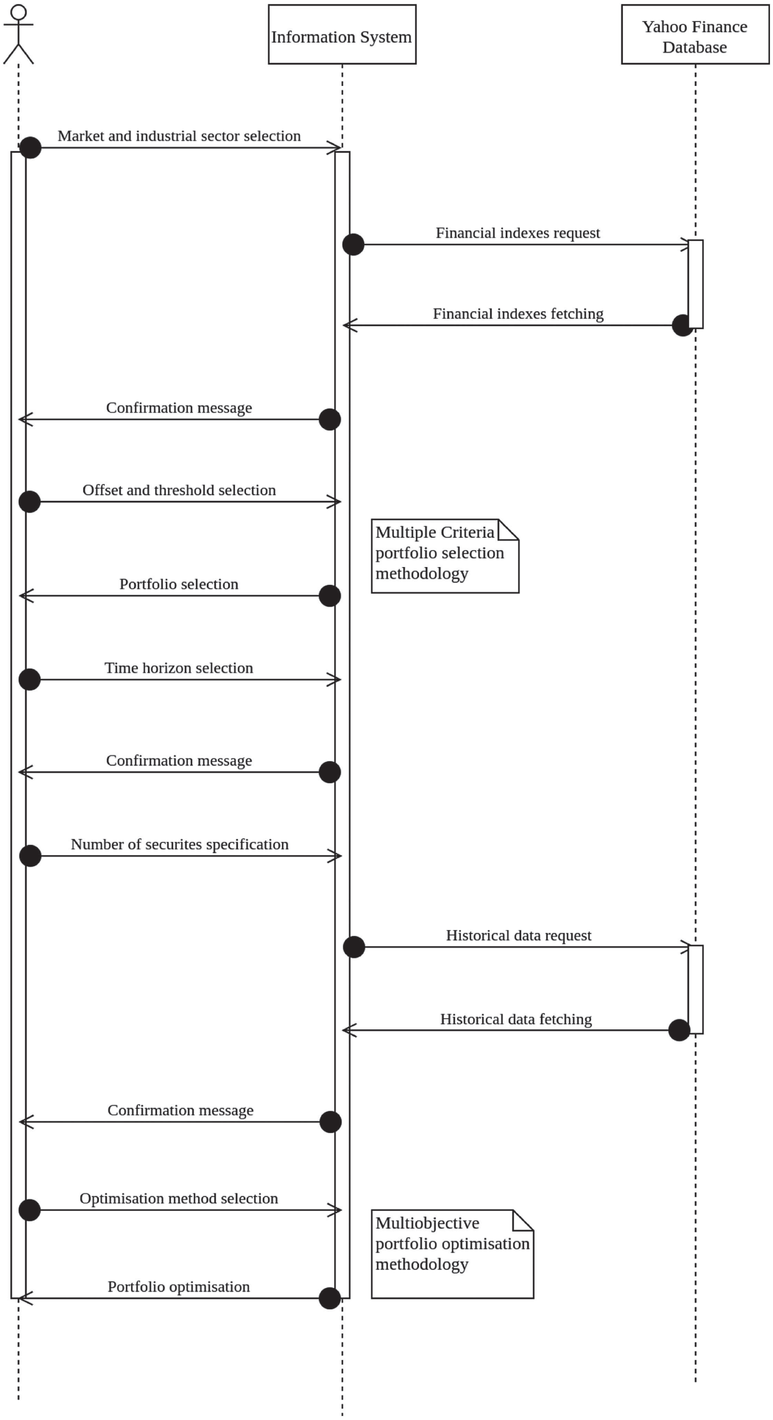
Information System
Yahoo Finance Database
Market and industrial sector selection
Financial indexes request
Financial indexes fetching
Confirmation message
Offset and threshold selection
Multiple Criteria portfolio selection methodology
Portfolio selection
Time horizon selection
Confirmation message
Number of securites specification
Historical data request
Historical data fetching
Confirmation message
Optimisation method selection
Multiobjective portfolio optimisation methodology
Portfolio optimisation

FIGURE 3. The UML Sequential Diagram of the developed information system.

3.3. Source code

As mentioned earlier, the proposed methodology was implemented in Python 3.0. A series of indicative source code examples from both phases, along with the corresponding output are presented in the following paragraph. The first cell of code includes a part of the TOPSIS algorithm, and more specifically the calculation of the positive and negative ideal solutions and the separation distances. The second cell of code shows the connection with the Yahoo API, in order to obtain the necessary input data for the second phase. Finally, the third cell of code demonstrates the visualization of several securities’ returns in a comparative diagram.

[1]:

```
idealSolution = [0 for i in range(criteria)]
nonIdealSolution = [0 for i in range(criteria)]
for j in range(criteria):
    maxValue = float(’-inf’)
    minValue = float(’inf’)
    for i in range(alternatives):
        if weightedDecisionMatrix[i][j] < minValue:
            minValue = weightedDecisionMatrix[i][j]
        if weightedDecisionMatrix[i][j] > maxValue:
            maxValue = weightedDecisionMatrix[i][j]
    if optimizationType[j] == 0:
        idealSolution[j] = maxValue
        nonIdealSolution[j] = minValue
    elif optimizationType[j] == 1:
        idealSolution[j] = minValue
        nonIdealSolution[j] = maxValue

sPlus = [0 for i in range(alternatives)]
sMinus = [0 for i in range(alternatives)]
for i in range(alternatives):
    sumPlusTemp = 0
    sumMinusTemp = 0
    for j in range(criteria):
        sumPlusTemp = sumPlusTemp + math.
 ↪pow(idealSolution[j]-weightedDecisionMatrix[i][j],2)
        sumMinusTemp = sumMinusTemp + math.
 ↪pow(nonIdealSolution[j]-weightedDecisionMatrix[i][j],2)
    sPlus[i] = math.sqrt(sumPlusTemp)
    sMinus[i] = math.sqrt(sumMinusTemp)
```

```
============ Positive/Negative Ideal Solution ============
     Criterion  Positive Score  Negative Score
P/E Ratio                0.025           0.071
EPS                      0.096           0.014
Revenue (B)              0.094           0.002
```

```
Beta                      0.029           0.067
Dividend Yield            0.101           0.027
Monthly                   0.053           0.042
YTD (%)                   0.066           0.029
1 Year                    0.081           0.000

===== Distance from Positive/Negative Ideal Solutions =====
  Company  Dis. from Positive  Dis. from Negative
Accenture                0.119               0.068
Northrop                 0.098               0.114
IBM                      0.104               0.133
Motorola                 0.138               0.091
MSCI                     0.144               0.092
Oracle                   0.132               0.067
```

```
[2]: tickers = [’ACN’, ’NOC’, ’IBM’, ’MSI’, ’MSCI’, ’ORA’ ]
     startDate = ’2016-01-01’
     endDate = ’2018-12-31’

     historicalDataNOC = data.DataReader(’ACN’, ’yahoo’, startDate,␣
      ↪endDate)
     display(historicalDataNOC)

     historicalValues = data.DataReader(tickers, ’yahoo’, startDate,␣
      ↪endDate)
     stockValues = historicalValues[’Open’]
     numOfDates = stockValues.shape[0]
     numOfSecurities = stockValues.shape[1]
     print("Number of securities:", numOfSecurities)
     print("Number of dates:", numOfDates, "\n")
     stockValues = stockValues.fillna(method=’ffill’)
     display(stockValues)
```

```
Number of securities: 6
Number of dates: 754

==============  Stock Values  ==============
Symbols          ACN       NOC       IBM       MSI      MSCI       ORA
Date
2016-01-04  102.6200  185.9799  135.6000   67.6699   71.0999   35.6300
2016-01-05  101.9700  187.8500  136.7599   66.5000   70.6100   35.1500
2016-01-06  100.8099  190.1600  134.3800   65.5700   70.2600   34.7900
2016-01-07   99.7500  187.8999  133.6999   64.6299   69.2399   35.0400
2016-01-08   99.4800  188.7899  133.1799   64.3300   69.5800   35.5200
...              ...       ...       ...       ...       ...       ...
2018-12-31  140.3999  243.3099  113.3300  113.0299  146.5000   53.0099
```

```
[3]: fig, ax = plt.subplots(2, 3, figsize=(16,9))
     for j in range(numOfSecurities):
         colors = np.array([(1,0,0)]*len(returnDates))
         for i in range(numOfDates-1):
             if stockReturnsArray[i][j] > 0:
                 colors[i] = (0,1,0)
         ax[j//3,j%3].bar(returnDates, stockReturnsArray[:,j],␣
      ↪color=colors)
         ax[j//3,j%3].set_title(stockValues.columns[j])
         ax[j//3,j%3].set_xticklabels([])
         ax[j//3,j%3].set_xlabel("Date")
         ax[j//3,j%3].set_ylabel("Return")
         fig.suptitle('Visualization of Security Returns',␣
      ↪fontweight='bold', fontsize=21)
```

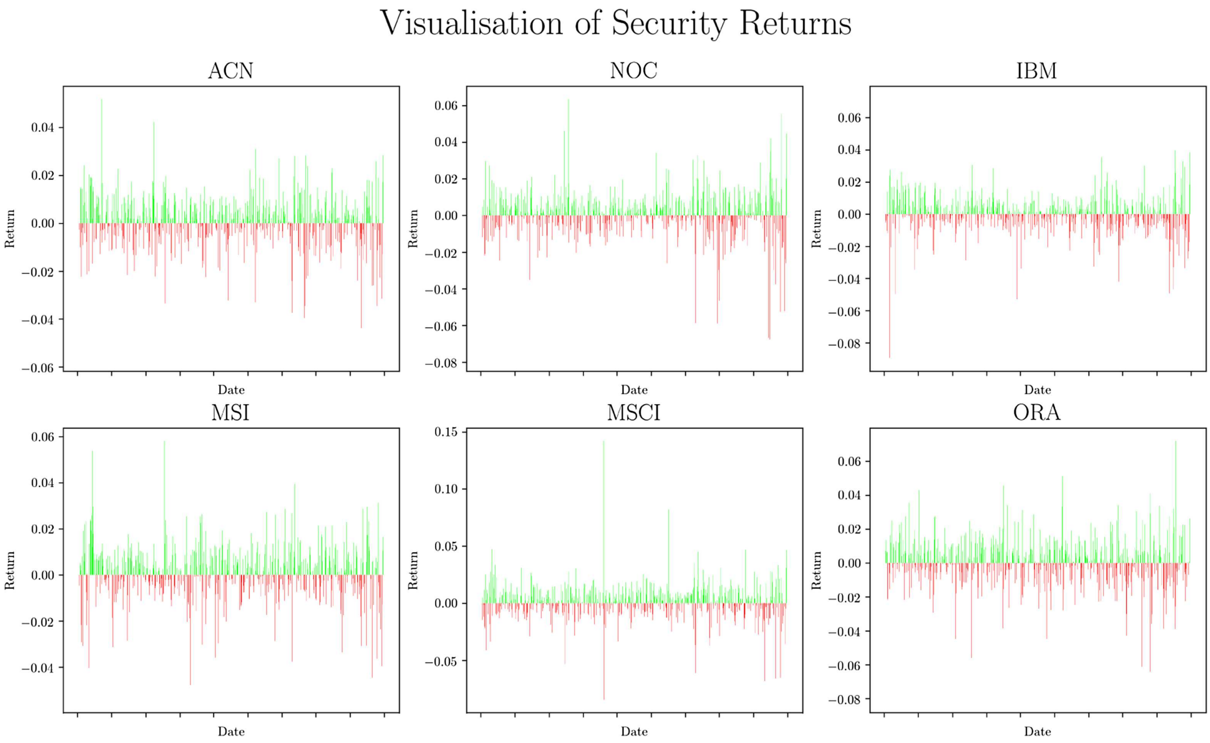

Figure 4. Visualisation of Security Returns in Python 3.0.

4. Empirical testing

The proposed methodological approach has been applied on data from four stock exchanges: (a) NYSE, (b) NASDAQ, (c) Paris and (d) Tokyo. The sample considered in the study consists of securities from three different sectors: (a) technological, (b) energy and (c) financial. The time horizon of the study was set to three calendar years (between 1/1/2016 and 31/12/2018), recording the daily closing prices. In this article we present the application on NYSE stock exchange, including a set of 516 securities (69 securities from the technological sector, 89 securities from the energy sector and 358 securities from the financial sector). Therefore, in Table 2 specific information is provided about the securities which have participated in the empirical testing process.

Table 2. Empirical testing input information.

Stock exchange	Industrial sector	Number of securities used	Number of securities with missing data	Total number of securities
NYSE	Technology	69	177	246
	Energy	89	131	220
	Financial	358	461	819
NASDAQ	Technology	326	213	539
	Energy	6	40	46
	Financial	93	471	564
Paris	Technology	50	91	141
	Energy	7	8	15
	Financial	33	24	57
Tokyo	Technology	485	263	748
	Energy	30	4	34
	Financial	143	51	194

In this paragraph, we present the experimental application on NYSE stock exchange. The selection of the offsets was determined according to three different scenarios, in order to conduct a sensitivity analysis on the results. In this paragraph, the results according to the scenario of equal offsets among the alternatives are presented.

The thresholds configuration differs significantly according to the multicriteria method. For each ranking method the configuration process was based on the partition of the values' range. Firstly, we determine the range by calculating the minimum and maximum value of the alternatives for each criterion and secondly we split this range as follows: (i) For ELECTRE III method which involves three different thresholds (preference p, indifference q and veto v) we split the range in four sections and assign the values respectively: $q(i) < p(i) < v(i)$, (ii) For PROMETHEE II method which involves two different thresholds (preference p and indifference q) we split the range in three sections and assign the values respectively: $q(i) < p(i)$, (iii) MAUT and TOPSIS methods do not involve any thresholds.

In Table 3, the results of the four ranking multicriteria methods for the technological sector are presented indicatively. The same process has been applied to the energy and financial sectors, as well. The securities with highest aggregated score are the ones which finally participate to the portfolio. The number of securities that will be included in the portfolio is selected by the DM. In the context of this experimental application the total number of securities in the portfolio is set to 60. Therefore, we selected the 20-highest ranked securities from each one of the three sectors, which means that the portfolio consists of 20 technological securities, 20 energy securities and 20 financial securities.

Given the results of the multicriteria methods, the final step of the first phase involves the cumulative ranking of the securities. Assuming that we select the 20-highest ranked securities from each industrial sector, the portfolio has been formulated as shown in Table 4.

Table 3. Results presentation for Phase I (security selection) for the Technology Sector of NYSE stock exchange.

Security Name	ELECTRE III	MAUT	PROMETHEE II	TOPSIS
ABB ADR	11.57	26.99	−22.32	30.96
Accenture	51.66	62.42	38.03	43.95
SAP ADR	1.74	40.47	−3.72	33.49
Infosys ADR	18.75	57.52	23.24	38.55
Wipro ADR	−28.89	23.38	−31.61	29.81
BT ADR	17.89	33.19	−19.79	34.52
STMicroelectronics ADR	10.58	54.31	16.04	39.92
Canon ADR	10.05	32.53	−10.75	34.49
Agilent Technologies	6.24	52.21	7.06	36.48
Allegion PLC	13.61	54.72	18.49	38.85
Ametek	12.99	53.93	16.71	38.90
...	...	...	...	...
Xerox	−3.92	48.70	4.03	40.19

Table 4. Selected securities from NYSE stock exchange.

Technology	Energy	Financial
Northrop Grumman	Phillips 66	Renaissancere
GlobalSCAPE	NACCO Industries	White Mountains Insurance
Accenture	Cypress Energy Partners LP	Triplepoint Venture
Synnex	Global Partners	JPMorgan
IBM	Sunoco LP	Cohen Steers TR Realty Closed
Taiwan Semiconductor	TC Energy	Santander Consumer USA Holdings Inc
Motorola	Royal Dutch Shell ADR	BlackRock Taxable Muni Bond Trust
Jabil Circuit	GasLog Partners Pref A	Hartford
Oracle	Phillips 66 Partners LP	Wells Fargo Real Estate Invest Pref
MSCI	Royal Dutch Shell B ADR	Nuveen AMT Free Muni Credit
Roper Technologies	World Fuel Services	MFS California
Danaher	Cosan Ltd	Wells Fargo Pref L
Leidos	Magellan	Saratoga Investment Corp
Benchmark Electronics	CVR Energy	Allstate
Infosys ADR	BP ADR	Blackrock Muni Target Term Closed
Hubbell	Chevron	PennyMac Mortgage
Nelnet	CNOOC ADR	Flaherty and Crumrine Dynamic Pref
CAE Inc.	Exxon Mobil	Metlife Inc Pref
Hexcel	ONEOK	PNC Financial
Broadridge	PetroChina ADR	Reinsurance of America

The model is equipped with both continuous and binary decision variables. Firstly, the continuous variables X_i, where $i = 1, \dots, 60$ represent the percentage of capital to be invested in the i_{th} security of the portfolio. Secondly, the use of the binary variables controls the existence of the i_{th} security in the portfolio (if $b_i = 1$ the i_{th} security participates in the portfolio, else if $b_i = 0$ the i_{th} security does not participate). The necessity for the incorporation of both kinds of variables stems from the fact that the combination of continuous and binary variables results in a more realistic modeling of the portfolio optimisation problem.

The models are equipped with both mandatory constraints and policy constraints for the needs of the application. The mandatory constraints are necessary for the correct formulation of the problem and the avoidance of logical errors, while the policy constraints are very important in order to incorporate the preferences of the DM. More specifically, the imposed constraints are the following:

(1) *Completeness constraint:* the completeness constraint is a mandatory constraint which requires all the available capital to be invested.

$$\sum_{i=1}^{60} X_i = 1. \tag{4.1}$$

(2) *Minimum/Maximum number of securities to participate in the portfolio:* the existence of binary variables has provided the opportunity to control the minimum and maximum number of securities to participate in the portfolio. It is a common strategy to restrict the number of minimum securities, in order to maintain a diversified portfolio. Additionally, the maximum number of securities could also be restricted according to the specific needs of the DM. The constraint is formulated as follows:

$$6 \leq \sum_{i=1}^{m} b_i \leq 40. \tag{4.2}$$

(3) *Minimum/Maximum percentage of capital invested in a security:* we also restrict the minimum and maximum amount of capital that can be invested in a specific security. The maximum amount of capital should be restricted in order to avoid the allocation of big capital percentage to a specific security, while the minimum amount is restricted because there is a lower threshold to the amount one can invest. Therefore this constraint is formulated as follows:

$$w_i - 0.03 \times b_i \geq 0, \quad i = 1, 2, \ldots, 60 \tag{4.3}$$

$$w_i - 20 \times b_i \leq 0, \quad i = 1, 2, \ldots, 60. \tag{4.4}$$

The results of each model are presented in Tables 5–8, respectively. More specifically: In Table 5 we present the Pareto efficient portfolios which were generated from the Mean-Variance MIQP model. In Table 6 we present the solution of the Goal Programming model. In Table 7 we present the Pareto efficient frontier of the MOIP PROMETHEE Flow model. Finally, in Table 8 the Genetic Algorithm portfolio is presented.

5. Conclusions

The criticism addressed, over the traditional bi-criteria optimization Markowitz models, which, among else, fail to assess complex scenarios of multiple investment objectives and constraints, is consistently emphasized by both practitioners and academics. Our aim in this article was to expand the limited Markowitz framework, within which the portfolio selection process is conventionally dealt with. More specifically, we presented a Python-based multicriteria portfolio selection DSS. The contribution of both the proposed methodology and the corresponding DSS is a multi-tiered one. The DM's preference system is effectively incorporated in the decision-making process by fully assessing his/her investment policy objectives and constraints, regarding the portfolio architecture.

The proposed DSS consists of two components: The first is linked with security selection, while the second one is connected with the portfolio optimization phase. In the first phase, four multicriteria methods are utilized, such as the PROMETHEE II, the ELECTRE III, the MAUT and the TOPSIS. After integrating the results, a variety of mathematical programming models are applied, such as a mixed-integer quadratic programming (MIQP), a goal programming (GP), a genetic algorithm (GA), and a multiobjective PROMETHEE flow. The proposed approach is tested through a large-scale illustrative application in several stock markets and various sectors, analyzing simultaneously a very large number of securities.

Table 5. Set of efficient portfolios produced by the Mean-Variance MIQP model for NYSE stock exchange.

Portf	NOC	GSB	ACN	SNX	IBM	TSM	MSI	JBL	ORCL	MSCI	ROP	...	DFP	MET	PNC	RGA
1	0.01	0.00	0.0	0.0	0.0	0.0	0.01	0.0	0.0	0.00	0.0	...	0.08	0.0	0.00	0.00
2	0.01	0.00	0.0	0.0	0.0	0.0	0.01	0.0	0.0	0.00	0.0	...	0.08	0.0	0.01	0.00
3	0.01	0.01	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.00	0.0	...	0.07	0.0	0.01	0.00
4	0.00	0.01	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.00	0.0	...	0.06	0.0	0.00	0.00
5	0.01	0.01	0.0	0.0	0.0	0.0	0.03	0.0	0.0	0.00	0.0	...	0.06	0.0	0.00	0.00
6	0.00	0.01	0.0	0.0	0.0	0.0	0.04	0.0	0.0	0.00	0.0	...	0.05	0.0	0.00	0.01
7	0.00	0.01	0.0	0.0	0.0	0.0	0.04	0.0	0.0	0.00	0.0	...	0.04	0.0	0.00	0.02
8	0.00	0.01	0.0	0.0	0.0	0.0	0.04	0.0	0.0	0.00	0.0	...	0.03	0.0	0.00	0.02
9	0.00	0.01	0.0	0.0	0.0	0.0	0.04	0.0	0.0	0.00	0.0	...	0.00	0.0	0.00	0.03
10	0.00	0.01	0.0	0.0	0.0	0.0	0.04	0.0	0.0	0.00	0.0	...	0.00	0.0	0.00	0.03
11	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.01	0.0	...	0.00	0.0	0.00	0.03
12	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.01	0.0	...	0.00	0.0	0.00	0.03
13	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.02	0.0	...	0.00	0.0	0.00	0.03
14	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.02	0.0	...	0.00	0.0	0.00	0.03
15	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.03	0.0	...	0.00	0.0	0.00	0.03
16	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.03	0.0	...	0.00	0.0	0.00	0.04
17	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.04	0.0	...	0.00	0.0	0.00	0.04
18	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.05	0.0	...	0.00	0.0	0.00	0.04
19	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.05	0.0	...	0.00	0.0	0.00	0.04
20	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.06	0.0	...	0.00	0.0	0.00	0.05
21	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.06	0.0	...	0.00	0.0	0.00	0.05
22	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.06	0.0	...	0.00	0.0	0.00	0.06
23	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.07	0.0	...	0.00	0.0	0.00	0.06
24	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.07	0.0	...	0.00	0.0	0.00	0.07
25	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.08	0.0	...	0.00	0.0	0.00	0.06
26	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.08	0.0	...	0.00	0.0	0.00	0.06
27	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.09	0.0	...	0.00	0.0	0.00	0.07
28	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.10	0.0	...	0.00	0.0	0.00	0.07
29	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.11	0.0	...	0.00	0.0	0.00	0.07
30	0.00	0.01	0.0	0.0	0.0	0.0	0.07	0.0	0.0	0.12	0.0	...	0.00	0.0	0.00	0.07

Table 6. The portfolio produced by the Goal Programming model for NYSE stock exchange.

NOC	GSB	ACN	SNX	IBM	TSM	MSI	JBL	ORCL	MSCI
0.1158	0.0300	0.0	0.0	0.0	0.0	0.0300	0.0	0.2000	0.0
ROP	DHR	LDOS	BHE	INFY	HUBB	NNI	CAE	HXL	BR
0.0300	0.0299	0.0	0.0443	0.0356	0.0	0.0300	0.0300	0.0	0.0305
PSX	NC	CELP	GLP	SUN	TRP	RDS-A	GLOP	PSXP	RDS-B
0.0	0.0300	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
INT	CZZ	MMP	CVI	BP	CVX	CEO	XOM	OKE	PTR
0.0300	0.0300	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
RNR	WTM	TPVG	JPM	RFI	SC	BBN	HIG	NVG	WFC
0.0300	0.0300	0.0	0.0	0.0	0.0	0.0	0.0300	0.0	0.0
SAR	ALL	BTT	PMT	DFP	MET	PNC	RGA		
0.0	0.1541	0.0300	0.0	0.0	0.0	0.0	0.0300		

Table 7. Set of efficient portfolios for NYSE stock exchange with MOIP PROMETHEE methodology.

Portf	NOC	GSB	ACN	SNX	IBM	TSM	MSI	JBL	ORCL	MSCI	ROP	...	DFP	MET	PNC	RGA
1	0.00	0.03	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	...	0.03	0.00	0.00	0.03
2	0.03	0.03	0.00	0.03	0.00	0.00	0.03	0.00	0.00	0.00	0.00	...	0.03	0.00	0.00	0.00
3	0.03	0.03	0.00	0.03	0.00	0.00	0.03	0.00	0.00	0.00	0.00	...	0.03	0.00	0.00	0.00
4	0.03	0.03	0.03	0.03	0.00	0.03	0.03	0.00	0.00	0.00	0.00	...	0.03	0.00	0.00	0.00
5	0.09	0.03	0.03	0.03	0.00	0.03	0.03	0.00	0.00	0.00	0.00	...	0.03	0.00	0.00	0.00
6	0.14	0.03	0.03	0.03	0.00	0.03	0.03	0.00	0.00	0.00	0.00	...	0.03	0.00	0.00	0.00
7	0.18	0.03	0.03	0.03	0.00	0.03	0.03	0.00	0.00	0.00	0.00	...	0.03	0.00	0.00	0.00
8	0.20	0.03	0.03	0.03	0.00	0.03	0.03	0.03	0.00	0.00	0.00	...	0.03	0.00	0.00	0.00
9	0.20	0.03	0.03	0.03	0.00	0.03	0.03	0.03	0.00	0.00	0.00	...	0.03	0.00	0.00	0.00
10	0.20	0.03	0.03	0.08	0.00	0.03	0.03	0.03	0.00	0.00	0.00	...	0.03	0.00	0.00	0.00
11	0.20	0.03	0.03	0.10	0.00	0.03	0.03	0.03	0.00	0.03	0.00	...	0.00	0.00	0.00	0.00
12	0.20	0.03	0.03	0.11	0.00	0.03	0.03	0.03	0.00	0.03	0.00	...	0.00	0.00	0.00	0.00
13	0.20	0.03	0.03	0.16	0.00	0.03	0.03	0.03	0.00	0.03	0.00	...	0.00	0.00	0.00	0.00
14	0.20	0.03	0.03	0.17	0.00	0.03	0.03	0.03	0.00	0.03	0.00	...	0.00	0.00	0.00	0.00
15	0.20	0.03	0.03	0.20	0.00	0.03	0.03	0.03	0.00	0.03	0.00	...	0.00	0.00	0.00	0.00
16	0.20	0.03	0.04	0.20	0.00	0.03	0.03	0.03	0.00	0.03	0.00	...	0.00	0.00	0.00	0.00
17	0.20	0.03	0.04	0.20	0.00	0.03	0.03	0.03	0.00	0.03	0.00	...	0.00	0.00	0.00	0.00
18	0.20	0.03	0.09	0.20	0.00	0.03	0.03	0.03	0.00	0.03	0.00	...	0.00	0.00	0.00	0.00
19	0.20	0.03	0.09	0.20	0.00	0.03	0.03	0.03	0.00	0.03	0.00	...	0.00	0.00	0.00	0.00
20	0.20	0.00	0.08	0.20	0.00	0.03	0.03	0.03	0.00	0.03	0.00	...	0.00	0.00	0.00	0.00
21	0.20	0.00	0.04	0.20	0.03	0.03	0.03	0.03	0.00	0.03	0.00	...	0.00	0.00	0.00	0.00
22	0.20	0.00	0.04	0.20	0.03	0.03	0.03	0.03	0.00	0.03	0.00	...	0.00	0.00	0.00	0.00
23	0.18	0.00	0.03	0.20	0.03	0.03	0.00	0.03	0.03	0.03	0.00	...	0.00	0.00	0.00	0.00
24	0.13	0.00	0.03	0.20	0.03	0.03	0.00	0.03	0.03	0.03	0.00	...	0.00	0.00	0.00	0.00
25	0.09	0.00	0.03	0.20	0.03	0.03	0.00	0.03	0.03	0.03	0.00	...	0.00	0.00	0.00	0.00
26	0.04	0.00	0.03	0.20	0.03	0.03	0.00	0.03	0.03	0.03	0.00	...	0.00	0.00	0.00	0.00
27	0.03	0.00	0.03	0.17	0.03	0.03	0.00	0.03	0.03	0.03	0.03	...	0.00	0.00	0.00	0.00
28	0.03	0.00	0.03	0.10	0.03	0.03	0.00	0.03	0.03	0.03	0.03	...	0.00	0.00	0.00	0.00
29	0.03	0.00	0.03	0.04	0.03	0.03	0.00	0.03	0.03	0.03	0.03	...	0.00	0.00	0.00	0.00
30	0.00	0.00	0.03	0.00	0.03	0.00	0.00	0.00	0.03	0.03	0.03	...	0.00	0.03	0.03	0.00

Table 8. The portfolio produced by the Genetic algorithm model for NYSE stock exchange.

NOC	GSB	ACN	SNX	IBM	TSM	MSI	JBL	ORCL	MSCI
0.1	0.005	0.003	0.209	0.004	0.0	0.003	0.005	0.003	0.004
ROP	DHR	LDOS	BHE	INFY	HUBB	NNI	CAE	HXL	BR
0.322	0.003	0.003	0.005	0.004	0.004	0.003	0.004	0.196	0.003
PSX	NC	CELP	GLP	SUN	TRP	RDS-A	GLOP	PSXP	RDS-B
0.003	0.0	0.0	0.0	0.003	0.004	0.003	0.005	0.005	0.004
INT	CZZ	MMP	CVI	BP	CVX	CEO	XOM	OKE	PTR
0.003	0.004	0.002	0.003	0.003	0.004	0.004	0.003	0.002	0.004
RNR	WTM	TPVG	JPM	RFI	SC	BBN	HIG	NVG	WFC
0.003	0.003	0.004	0.004	0.004	0.004	0.003	0.003	0.005	0.004
SAR	ALL	BTT	PMT	DFP	MET	PNC	RGA		
0.004	0.002	0.004	0.003	0.003	0	0	0.003		

In closing, further research that may be considered for broadening the proposed approach can be summarized as follows: (a) Expansion of the criteria set towards a qualitative direction, by considering for example analysts' estimates, buy-hold-sell signals etc., (b) expansion of the methodology's focus so as to include additional asset classes, such as bonds, ETFs etc., and (c) connection of the DSS with a portfolio reporting and order execution system, for the vertical integration of the whole portfolio management business cycle, as a professional routine.

References

[1] B. Aouni, M. Doumpos, B. Pérez-Gladish and R.E. Steuer, On the increasing importance of multiple criteria decision aid methods for portfolio selection. *J. Oper. Res. Soc.* **69** (2018) 1525–1542.

[2] M. Behzadian, S.K. Otaghsara, M. Yazdani and J. Ignatius, A state-of the-art survey of TOPSIS applications. *Expert Syst. App.* **39** (2012) 13051–13069.

[3] J.-P. Brans and P. Vincke, Note – A preference ranking organisation method: (the promethee method for multiple criteria decision making). *Manage. Sci.* **31** (1985) 647–656.

[4] J.-P. Brans, P. Vincke and B. Mareschal, How to select and how to rank projects: the PROMETHEE method. *Eur. J. Oper. Res.* **24** (1986) 228–238.

[5] S.-J. Chen and C.-L. Hwang, Fuzzy multiple attribute decision making methods. In: Fuzzy Multiple Attribute Decision Making. Springer (1992) 289–486.

[6] M. Doumpos and C. Zopounidis, Multicriteria Analysis in Finance. Springer (2014).

[7] S.S. Hashemi, S.H.R. Hajiagha, E.K. Zavadskas and H.A. Mahdiraji, Multicriteria group decision making with ELECTRE III method based on interval-valued intuitionistic fuzzy information. *Appl. Math. Modell.* **40** (2016) 1554–1564.

[8] M. Hirschberger, R.E. Steuer, S. Utz, M. Wimmer and Y. Qi, Computing the nondominated surface in tri-criterion portfolio selection. *Oper. Res.* **61** (2013) 169–183.

[9] C.-L. Hwang and K. Yoon, Methods for multiple attribute decision making. In: Multiple Attribute Decision Making. Springer (1981) 58–191.

[10] R.L. Keeney, H. Raiffa and R.F. Meyer, Decisions with Multiple Objectives: Preferences and Value Trade-offs. Cambridge University Press (1993).

[11] H.-F. Li and J.-J. Wang, An improved ranking method for ELECTRE III. In: *2007 International Conference on Wireless Communications, Networking and Mobile Computing. IEEE* (2007) 6659–6662.

[12] H. Markowitz, Portfolio selection. *J. Finance* **7** (1952) 77–91.

[13] K. Palczewski and W. Sałabun, The fuzzy TOPSIS applications in the last decade. *Proc. Comput. Sci.* **159** (2019) 2294–2303.

[14] E. Ralph and P. Na, Multiple criteria decision making combined with finance: A categorized bibliographic study. *Eur. J. Oper. Res.* **150** (2003) 496–515.

[15] B. Roy, ELECTRE III: Un algorithme de classements fondé sur une représentation oue des préférences en présence de critères multiples. *Cahiers du CERO* **20** (1978) 3–24.

[16] M.M. Salih, B.B. Zaidan, A.A. Zaidan and M.A. Ahmed, Survey on fuzzy TOPSIS state-of-the-art between 2007 and 2017. *Comput. Oper. Res.* **104** (2019) 207–227.

[17] R.E. Steuer, Y. Qi and M. Hirschberger, Suitable-portfolio investors, nondominated frontier sensitivity, and the effect of multiple objectives on standard portfolio selection. *Ann. Oper. Res.* **152** (2007) 297–317.

[18] R.E. Steuer, Y. Qi and M. Hirschberger, Comparative issues in large-scale mean-variance efficient frontier computation. *Decis. Support Syst.* **51** (2011) 250–255.

[19] R. Storn and K. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11** (1997) 341–359.

[20] P. Xidonas and G. Mavrotas, Multiobjective portfolio optimization with non-convex policy constraints: evidence from the Eurostoxx 50. *Eur. J. Finance* **20** (2014) 957–977.

[21] P. Xidonas, G. Mavrotas, C. Zopounidis and J. Psarras, IPSSIS: An integrated multicriteria decision support system for equity portfolio construction and selection. *Eur. J. Oper. Res.* **210** (2011) 398–409.

[22] P. Xidonas, G. Mavrotas, T. Krintas, J. Psarras and C. Zopounidis, Multicriteria Portfolio Management. Springer, New York (2012).

[23] C. Zopounidis and M. Doumpos, Multicriteria decision systems for financial problems. *Top* **21** (2013) 241–261.

[24] C. Zopounidis, E. Galariotis, M. Doumpos, S. Sarri and K. Andriosopoulos, Multiple criteria decision aiding for finance: an updated bibliographic survey. *Eur. J. Oper. Res.* **247** (2015) 339–348.