"THIS ARTICLE HAS BEEN ACCEPTED FOR PUBLICATION IN THE JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY, PUBLISHED BY TAYLOR & FRANCIS."

A surrogate similarity measure for the mean-variance frontier optimization problem under bound and cardinality constraints

Francisco Guijarro[a] and Prodromos E. Tsinaslanidis[b]

[a] Research Institute for Pure and Applied Mathematics, Universitat Politècnica de València, Camí de Vera s/n, Valencia, Spain; [b] Christ Church Business School, Canterbury Christ Church University, North Holmes Road, Canterbury, Kent, CT1 1QU, UK

ARTICLE HISTORY
Compiled August 27, 2019

CONTACT Francisco Guijarro. Email: fraguima@upvnet.upv.es

ABSTRACT
This paper deals with the mean-variance optimization frontier problem when realistic constraints are considered. Our proposed methodology hybridizes a heuristic algorithm with an exact solution approach. A genetic algorithm is applied for the identification of the assets in the portfolio, whilst the asset weights in the portfolios are obtained by a quadratic programming model. The proposed algorithmic framework produces a constrained frontier that actually fulfils the bound and cardinality constraints, unlike other proposals where the frontier is composed of several sub-frontiers, each one considering the cardinality constraint but with different assets in each sub-frontier, thus violating the cardinality constraint. This brings us to propose a surrogate similarity measure for the optimization of the constrained frontier, which differs from a previous proposal where no bound constraints were considered. Regarding the genetic algorithm, we propose an initial population to boost the convergence of the optimization process, whilst the adopted mutation and crossover genetic operators result in feasible individuals. An illustrative example using components of five major stock market indices is provided to demonstrate the effectiveness of the proposed method.

KEYWORDS
Portfolio optimization; cardinality constraint; bound constraint; genetic algorithm

1. Introduction

The portfolio optimization problem was defined by Markowitz (1952) as investing in some risky assets in an efficient way. Markowitz formulated the problem as a mathematical programming model with the following two competing objectives: (i) maximizing the expected return of the portfolio and (ii) minimizing its risk (variance of the returns). The efficient frontier is the set of non-dominated portfolios in the mean-variance sense that meet the aforementioned objectives.

Some researchers have pointed out several weaknesses in the mean-variance portfolio optimization problem. From a practical point of view, investors usually face typical constraints such as the cardinality constraint, which sets a limit on the number of assets composing a portfolio, and lower and upper bounds for the weights of assets (Chang, Meade, Beasley, & Sharaiha, 2000; Liagkouras & Metaxiotis, 2014; Woodside, Lucas, & Beasley, 2011). Restricting the number of assets by considering cardinality constraints can simplify the management of the portfolio and reduce transaction costs. This is particularly important if some assets have a relatively small weight in the portfolio. Removing these assets from the portfolio can reduce transaction costs without having any practical impact on the portfolio's expected return and risk (Chang et al., 2000; Guijarro, 2018; Ruiz-Torrubiano & Suárez, 2015). Bound constraints enable fund managers to limit the minimum and maximum amount to be invested in the assets; the lower bound reduces the transaction costs, whilst the upper limits the portfolio's exposure to a specific asset or sector. In addition, these constraints can improve both the in-sample and out-of-sample robustness and performance of the portfolios (Ruiz-Torrubiano & Suárez, 2015).

Constraints on cardinality transform the model into a quadratic mixed-integer problem, which has been proved to be NP-hard (Ruiz-Torrubiano & Suárez, 2009, 2015; Shaw, Liu, & Kopman, 2008). This is one of the main reasons why meta-heuristic algorithms have gained attention for the constrained portfolio optimization problem, the genetic algorithm being one of the academics' favourites (see Section 2).

In this paper, we address the portfolio optimization problem taking into account realistic constraints, such as cardinality and bound constraints. In line with Guijarro (2018), we focus on the whole constrained mean-variance frontier, and not on a specific portfolio on the frontier. An alternative consists of building the cardinality constrained frontier from different sub-frontiers, each one considering the cardinality constraint but with different assets in each sub-frontier. However, in practical terms, the "compounded" frontier violates the cardinality constraint. Furthermore, considering the cardinality constraint may result in a discontinuous frontier, where discontinuities imply that there are certain returns which no rational investor would consider, since there exist portfolios with the same risk but with higher returns (Chang et al., 2000). In addition, we also point out that taking the cardinality constraint as an equality entails a suboptimal solution when bound constraints have to be met. However, considering the cardinality constraint as an inequality mitigates this problem. This led us to propose an algorithm for the optimization of the constrained frontier when the cardinality constraint is interpreted as the maximum number of assets in the portfolio, instead of the exact number of assets. Including bound constraints allows to revisit the model proposed in Guijarro (2018) and create a surrogate similarity measure for the mean-variance frontier optimization under bound and cardinality constraints.

Another difference regarding the paper from Guijarro (2018) is the relaxation of the required return constraint for portfolios. Again, we use an inequality instead of an equality constraint, although it is known that in practice it is very difficult to give a specific value of "desired return". This way, the paper considers two constraint relaxations to solve the problem under realistic constraints, i.e. cardinality and return.

Our proposal hybridizes a heuristic algorithm with an exact solution approach. A genetic algorithm (GA) is applied for the identification of the assets in the portfolio, while the assets' weights in the portfolios are obtained by a quadratic programming model. We must emphasise that the GA identifies which assets are used for the frontier construction considering the aforementioned constraints. Thus, all portfolios use the same set of assets and the cardinality constraint is actually observed along the whole frontier, by setting the same assets for any portfolio in the frontier. The quadratic programming model takes those assets to compute their weights, according to the required return of each portfolio. Furthermore, the proposed framework can be considered in a multi-period setting (Li & Ng, 2000). In this situation, the decision maker needs to reinvest his/her wealth at the end of each investment period. This implies to readjust the assets' weights but without changing the assets involved in the portfolio in order to avoid excessive transaction costs.

The GA model does not have any cardinality constraints, which means that it can be solved using standard quadratic programming solvers. This combinatorial encoding offers a significant advantage over mixed encodings, such as those used in Chang et al. (2000), where chromosomes with both discrete and continuous components are used. More precisely, the GA can focus on solving the combinatorial optimization problem of finding the optimal subset of assets without handling the continuous constraints. This separation has been shown to increase the performance of cardinality constrained portfolio selection algorithms (Moscato & Cotta, 2003; Ruiz-Torrubiano & Suárez, 2015). We also propose an initial population that accelerates the convergence of the algorithm, whilst the mutation and crossover genetic operators adopted result in feasible individuals. Thus, neither a penalty function nor repair operator are required to convert non-feasible solutions to feasible. Our experiments were carried out on a database frequently used in similar studies. Our findings suggest that the present proposal for population initialisation enhances the performance of the algorithm and obtains better solutions in a significantly shorter computation time.

The rest of the paper is structured as follows: Section 2 includes a brief survey of the literature on the bound and cardinality constrained mean-variance problem and the heuristics that have been applied to solve it. Section 3 introduces the surrogate similarity measure used for the optimization of the bound and cardinality constrained mean-variance frontier. Section 4 presents and discusses the computational results and the conclusions are offered in Section 5.

2. Related work

This section reviews the literature on the constrained portfolio optimization problem and is mainly concentrated on studies that use a heuristic approach to consider the cardinality constraint.

The comparison between different heuristics in the cardinality constrained portfolio optimization problem was first addressed by Chang et al. (2000), who propose genetic algorithm, tabu search (TS) and simulated annealing (SA) to extend the standard model by including cardinality constraints and imposing limits on the proportion held in a given asset. They use the OR-Library (Beasley, 1990), a public database that has enabled other researchers to come up with alternative models and compare their results from the perspective of efficacy and efficiency. The database includes information on five well-known stock market indices, with cardinality ranging from 31 (Hang Seng Index) to 225 (Nikkei Index).

Chang, Yang, & Chang (2009) introduce several risk measures by employing GA, and compare the results with those obtained by the mean-variance model. The alternative risk measures are the semi-variance, mean absolute deviation and variance with skewness. In this work, computation times are similar to those reported by Ruiz-Torrubiano & Suárez (2009) and proportional to the number of assets in the cardinality constrained portfolio.

Soleimani, Golmakani, & Salimi (2009) consider the aforementioned constraints on cardinality and floor-ceiling limits, and add a new constraint regarding sector capitalization. The GA model is applied to large scale problems with 500 and 2,000 stocks, and acceptable results are obtained in a sensible time (less than 7 minutes).

Cura (2009) uses Particle Swarm Optimization (PSO) in the cardinality constrained mean-variance model. The experimental results show that when compared to GA, TS and SA, none of the four heuristics outperforms the others. In line with this, Deng, Lin, & Lo (2012) propose an improved PSO which increases the exploration in the initial search steps, boosting the convergence process in the last search steps. Their results show that this new version is more robust and effective than existing PSO, GA, SA and TS algorithms.

Sadjadi, Gharakhani, & Safari (2012) propose a new framework to solve the cardinality constrained portfolio problem when input parameters are subject to uncertainty. Since no practical solution for robust optimization exists when the uncertainty level increases, the authors propose a GA model to find near-optimal solutions. The CPU-time for the results reported were below one minute, but the database used was not the OR-Library.

Woodside et al. (2011) extend Chang et al.'s findings (2000). They compare the performance of GA, TS and SA, but solving to optimality a mixed-integer quadratic optimization problem. As stated by the authors, embedding an algorithmic step involving the optimal solution in a heuristic algorithm is relatively uncommon in the literature. They do not exactly define return but allow it to be specified within a certain range. The reported CPU-time for the largest test problems is below fifteen minutes.

Another novel approach entails different trading operations in the portfolio rebalancing, such as sell, hold or buy. Instead of building a static portfolio, Ruiz-Torrubiano & Suárez (2015) design a dynamic model over time and optimize it through GA. An adapted RAR crossover operator produces individuals that satisfy all the constraints, so that no repair mechanisms are needed. The exploration and exploitation capabilities of the GA are enforced using the proposed crossover and mutation operators that take advantage of specific features of the problem.

GA is also used by Guijarro (2018) where the optimization problem is not the search for a specific portfolio, but the whole cardinality constrained frontier. The solution is reached by optimizing the similarity between the unconstrained mean-variance frontier and the cardinality constrained frontier. The author only considers cardinality constraints.

Another example of GA to solve a financial problem is proposed by Ruiz-Torrubiano & Suárez (2009). The authors apply GA to the index tracking problem, which consists of reproducing the performance of a stock-market index by investing in a subset of the stocks included in the index –cardinality constraint–. They handle the problem on two levels: the combinatorial problem of identifying the appropriate assets is solved by GA, while quadratic programming is used to determine the proportion. No bound constraints are considered. The results are reported for the OR-Library stock market indices and optimal solutions for 10-assets cardinality constrained portfolios are obtained in less than a minute's computation time in all cases.

Although GA is among the most popular heuristics in academia, other approaches have also been recently applied to the cardinality constrained portfolio optimization problem, like the artificial bee colony algorithm (ABC) or the electromagnetism-like algorithm. Some recent examples can be found in Kalayci, Ertenlice, Akyer, & Aygoren (2017) and Salehpoor & Molla-Alizadeh-Zavardehi (2019).

A remarkable approach is proposed by Bruni, Cesarone, Scozzari, & Tardella (2015), who implicitly control the cardinality by controlling the return (the extra-return with respect to a given index, which is easier to specify in practical cases), without the need for explicit cardinality constraints.

3. Formulation of the problem

This section describes the proposed framework of the bounded and cardinality constrained mean-variance optimization problem, plus several issues concerning the constraint on asset weights. We first propose a surrogate for the similarity measure given in Guijarro (2018) and then introduce specific operators for the improvement of the genetic algorithm.

3.1. The bounded and cardinality constrained mean-variance optimization problem

Here we introduce different elements related to the constrained mean-variance model we propose. We initially assume the model, Eq. (1), from Chang et al. (2000), which draws up the efficient portfolio with return r^*, constrained to k assets and bounded to limits l_i and u_i, $i = 1, \ldots, N$.

$$\text{minimize} \quad \sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \sigma_{ij} \tag{1a}$$

$$\text{subject to} \quad \sum_{i=1}^{N} x_i r_i = r^* \quad \text{(desired return constraint)} \tag{1b}$$

$$\sum_{i=1}^{N} x_i = 1 \quad \text{(budget constraint)} \tag{1c}$$

$$l_i z_i \leq x_i \leq u_i z_i \quad \text{(bound constraints)} \tag{1d}$$

$$\sum_{i=1}^{N} z_i = k \quad \text{(cardinality constraint)} \tag{1e}$$

$$z_i \in \{0, 1\} \tag{1f}$$

where:

x_i is the proportion held in asset i ($i = 1, \ldots, N$),
σ_{ij} is the covariance between the return of asset i and asset j ($i, j = 1, \ldots, N$),
r_i is the expected return of asset i ($i = 1, \ldots, N$),

r^* is the desired level for the portfolio's return,
N is the number of assets available to invest in,
k is the cardinality of the portfolio,
z_i is a binary variable which is 1 if asset i $(i = 1, \dots, N)$ is held, and 0 otherwise,
l_i (u_i) is the minimum (maximum) proportion to be invested in asset i $(i = 1, \dots, N)$, if any investment is made in i.
Therefore x_i and z_i are the decision variables in model described in Eq. (1).

To illustrate some features of this model we begin with the data of the example provided by Chang et al. (2000) with $N = 4$ (Table 1). The original data consists of returns, standard deviations and correlations. We have substituted the covariance matrix for the correlation matrix, with the aim of drawing a direct parallel with model in Eq. (1).

[Table 1 about here.]

From this data, we have obtained both the unconstrained efficient frontier (UEF) and the cardinality constrained efficient frontier (CCEF) with $k = 3$ in Fig. 1. No bound constraints are considered for the CCEF at this stage, i.e. $l = 0$ and $u = 1$, and thus no short sales are allowed. However, the UEF problem does allow short sales, which facilitate the similarity measure proposed by Guijarro (2018). We can see that the CCEF is actually composed of two different frontiers: one is constrained to assets 1, 2 and 3, while the other is limited to assets 2, 3 and 4. Each frontier therefore complies with the cardinality constraint, but the whole CCEF violates this assumption while using all the assets.

[Figure 1 about here.]

In order to effectively fulfil the cardinality constraint in the whole frontier, Guijarro (2018) proposes a similarity measure for constraining the cardinality in the mean-variance optimization problem. However, in Guijarro (2018) the bounding constraints in Eq.(1d) are excluded. For this he compares the area of the UEF with the area of each candidate CCEF. The CCEF with the largest area is regarded as the most similar to the UEF, and hence is chosen as the optimal CCEF. Fig. 2 compares the area for the 4 different CCEF with the $k = 3$ constraint, but assuming the return constraint as an inequality, as we later discuss. We can see that the frontier with assets 1, 2 and 3 –CCEF$(1, 2, 3)$– is the one with the biggest area, so the investor should choose this frontier as optimal. For additional details on how the area is computed we refer to Guijarro (2018).

[Figure 2 about here.]

The similarity measure is devised for those frontiers that follow a continuous differentiable function, as does variance of returns, although in the case of bound limits the calculation of the area can become more difficult than in the unbounded case. Fig. 3 draws the bounded and cardinality constrained frontier devised with assets 1, 2 and 3 and with weights constrained to $l_i = 0.15$ and $u_i = 0.80$: BCCF$(1, 2, 3)$. We have also included the UEF for comparative purpose.

[Figure 3 about here.]

Chang et al. (2000) formulate the problem with an equality rather than an inequality in the cardinality and return constraints. They argue that if cardinality is expressed as a range $(k_L \leq k \leq k_U)$ then the problem can be faced by examining all values of

k between k_L and k_U. They state that "*the decision as to the number of assets (k) to have in the chosen portfolio is one that can only be decided by the decision-maker in the light of the tradeoff between the three factors (risk, return and k) involved*". Nevertheless, a major point we address in our paper is that in terms of the above reasoning the BCCF is not efficient, as considering the cardinality constraint as an equality makes the frontier inefficient. The same applies for the constraint related with the desired level for the portfolio's return Eq. (1b). In the following we detail our approach, which differentiates from Chang et al. (2000) in that we focus on the cardinality constrained frontier and not on a single portfolio from the frontier.

In Fig. 3 we have also depicted the bounded and cardinality constrained efficient frontier: BCCEF. This is computed with the same assets and bounding limits as BCCF, but considering both the return and cardinality constraints as an inequality, i.e. $\sum_{i=1}^{N} x_i r_i \geq r^*$ and $\sum_{i=1}^{N} z_i \leq k$. Therefore, we require a minimum return r^* and we allow the portfolios to include up to k assets, instead of exactly requiring k assets. In Fig. 3 we can see how both frontiers overlap from the minimum return considered in the plot (0.0020) to the return of portfolio p_3. But the BCCEF dominates the BCCF for higher returns. This happens because BCCF is not allowed to violate the bound limits of the assets, while the BCCEF compares the solution in which the asset weight is in the lower (upper) limit with the one obtained if the asset is excluded from the portfolio ($x_i = 0$). Fig. 3 shows that observing the cardinality through a strict equality also constrains the maximum return achievable. The return for the BCCF is limited to the one obtained by portfolio p_4, while BCCEF achieves a higher return. In some cases the bounds imposed by the investor can become redundant in practice when the problem is solved. From the above example with $l_i = 0.15$ and $u_i = 0.80$ for BCCF$(1, 2, 3)$, the portfolio p_4 in Fig. 3 takes the minimum vales for x_2 and x_3, i.e. $x_2 = x_3 = 0.15$. This constrains the value of x_1 to 0.70, and hence the theoretical upper limit of 0.80 is actually unattainable. The investor should take into account that the actual bounds, l_i^* and u_i^*, for the weights are delimited by Eqs. (2)-(3),

$$l_i^* = \max \left(l_i, 1 - \sum_{j=1, j \neq i}^{k} u_j \right), \tag{2}$$

$$u_i^* = \min \left(u_i, 1 - \sum_{j=1, j \neq i}^{k} l_j \right), \tag{3}$$

where $i = 1, \ldots, k$.

In short, (i) the BCCF assumes an equality constraint both for the return and cardinality and this implies a suboptimal solution, whilst (ii) the inequality constraints in the BCCEF ensure optimality for the investor and a wider range for the return.

Fig. 4 illustrates how the weights of assets change as return evolves. Both models allocate the same weights to assets from the point with minimum return to the return level of portfolio p_3. For higher returns, BCCEF excludes asset 2 from the portfolios ($x_2 = 0$), whilst BCCF keep its weight in the minimum value (0.15). BCCF is not capable of going beyond p_4, in contrast with BCCEF, which gets higher returns by considering assets 1 and 3 in their portfolios.

It is important to note that including limits makes the weights non linear with return. In the classical unconstrained mean-variance model, these weights vary in proportion to return, but Fig. 4 highlights that the weight of each asset changes between marked portfolios. This means that the similarity measure proposed in Guijarro (2018) for the cardinality constrained efficient frontier must be reconsidered to handle bound constraints. Instead of computing the area of the cardinality constrained frontier, as shown in Fig. 2, we propose a surrogate for the similarity measure as the distance between the UEF and the BCCEF.

[Figure 4 about here.]

3.2. A surrogate for the similarity measure

Here we provide an insight into the modified similarity measure for the identification of the bounded and cardinality constrained frontier. We use Fig. 5 to help us to introduce the surrogate similarity measure to the example given above.

Instead of computing the whole area of the frontiers, we propose to compute the similarity measure by measuring the distance from the BCCEF to the UEF. With the aim of reducing computational time, we only consider a limited number of portfolios for the distance calculation.

Fig. 5 gives an example of the selected portfolios, which connect the UEF and the BCCEF through horizontal lines. For the return we chose equidistant portfolios on the basis of the minimum and maximum return considered by Guijarro (2018). The figure suggests that in all four cases the changes in the weight of assets are non linear, as previously indicated in Fig. 4. Furthermore, the maximum return is not attained for BCCEF$(1, 2, 4)$ and BCCEF$(2, 3, 4)$. This is because the bounding constraints make it impossible to reach some return levels; i.e. no feasible portfolio reaches the required returns while satisfying the bounding constraints. Therefore, the optimal BCCEF must meet the following assumptions:

(a) In order to discard non-feasible solutions, the frontiers that do not reach the minimum and maximum return values will not be considered as feasible frontiers. If we did so, we could be favouring any frontiers dissimilar to the UEF but with a low reported distance because of the fewer portfolios considered. In the example in Fig. 5, BCCEF$(1, 2, 4)$ and BCCEF$(2, 3, 4)$ do not fulfil this requirement. We can see that if this condition is not required, BCCEF$(2, 3, 4)$ will be unfairly selected as optimal.
(b) The –feasible– BCCEF with the minimum sum of horizontal distances to the UEF will be chosen as optimal. In the case of Fig. 5, BCCEF$(1, 2, 3)$ would be the optimal frontier chosen by the algorithm.

Alternatively, we could compute the distance between the BCCEF and the BCCF (as in Fig. 3) instead of computing the distance between the BCCEF and the UEF. In the following we explain why we chose the BCCEF-UEF distance for the proposed similarity measure calculation.

First, considering the BCCF could involve limiting the return range (Fig. 3), while the UEF does not have this disadvantage because of the short sales. Secondly, let's suppose we are confronting two alternative BCCEF (A and B) which must be compared in the mean-variance sense. Without loss of generality we assume that A obtains a shorter distance to the UEF than B. If BCCF were used, then A would also get a shorter distance to BCCF than B. It is thus indifferent whether we use UEF or BCCF for the comparison between A and B frontiers. Nevertheless, the computation of the

BCCF requires solving a quadratic programming model with bounded constraints, while the UEF computation can be solved in a shorter time. Let $\mathbf{r}$ be the $k \times 1$ return vector of assets, $\mathbf{V}$ be the $k \times k$ covariance matrix of returns and $\mathbf{1}$ be the $k \times 1$ vector in which every entry equals 1. We only need to compute the variance σ^2 of a limited number of portfolios on the UEF (Fig. 5), which can be easily solved by using Eq. (4) as shown in Guijarro (2018).

$$\sigma^2 = \left(a - 2br^* + c(r^*)^2\right) / \left(ac - b^2\right), \tag{4}$$

where r^* is the desired expected return of the portfolio, $a = \mathbf{r}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{r}$, $b = \mathbf{r}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{1}$ and $c = \mathbf{1}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{1}$. Thus, values for a, b and c are computed only once for the whole frontier, and the variance is computed by varying the desired return.

[Figure 5 about here.]

Algorithm 1 presents the pseudo code for the calculation of the optimal BCCEF for a given subset of assets with cardinality k. This algorithm will ultimately be considered as the fitness function for the genetic algorithm. The inputs are the vector of returns ($\mathbf{r}$) and the variance-covariance matrix ($\mathbf{V}$) for the k assets considered, the upper ($\mathbf{u}$) and lower ($\mathbf{l}$) bounded limits, the range for the portfolios return (r.min and r.max), the number of portfolios to be considered in the similarity distance calculation (n.portfolios), and the variance of the portfolios in the UEF. The output is the optimal BCCEF both considering the bounded and the cardinality constraints, where the BCCEF is reported as a matrix containing the return and variance of the portfolios, and the distance between the aforementioned variance and the one reported by the UEF. As stated above, the return range is divided into n.portfolios at equally spaced intervals. The aim of the algorithm is to search the minimum distance between each of the points in the UEF and the BCCEF, as shown in Eq.(5):

$$\text{distance} = \sum_{p=1}^{\text{n.portfolios}} \left(\sigma^2_{p,\text{BCCEF}} - \sigma^2_{p,\text{UEF}}\right) \tag{5}$$

The following steps are followed for each return in the aforementioned range. First, the quadratic programming model is solved by considering the bound constraints. Since the inputs of Algorithm 1 include the specific assets to be considered in the frontier, the function solve.QP computes the model (1a)-(1d), i.e. the cardinality is implicit. If no feasible solution exists for the return and bound constraints considered, the problem is solved again by removing the bounded constraints. Secondly, we identify whether any asset has a weight on the upper or lower limits. In some cases a better solution can be attained if some of these assets are excluded from the portfolio. If there is a weight lying on the lower limit, the solution can be improved by removing the asset from the portfolio, i.e. $x_i = 0$. If there is a weight lying on the upper bound, removing the asset results in an improvement of the variance only in unusual cases.

As already mentioned, we consider both return and cardinality as inequality constraints. The point here is that the number of possible portfolios to be examined is exponential with the number of assets in the upper or lower limits. If only one asset is in this situation, then the variance of two portfolios should be compared: the portfolio with the asset weight in the lower (upper) limit, and the one with zero asset weight.

If two assets have weights on the limits, then four portfolios should be compared; and so on. Therefore, as n.bounding.weights is the number of assets with weights on either the lower or the upper limit, we need to compare $2^{\text{n.bounding.weights}}$ portfolios to solve for the optimal solution.

Algorithm 1 Pseudo code for the computation of the optimal BCCEF (fitness function for the genetic algorithm). Comments are placed after a hash (#) in italics.

Inputs: $\mathbf{r} = [r_1, \ldots, r_k]^{\mathrm{T}}$, $\mathbf{V} = [v_{i,j}]\ i,j = 1, \ldots, k$, $\mathbf{l} = [l_1, \ldots, l_k]^{\mathrm{T}}$, $\mathbf{u} = [u_1, \ldots, u_k]^{\mathrm{T}}$, n.portfolios, r.min, r.max, $\mathbf{UEF} = [v_1, \ldots, v_{\text{n.portfolios}}]^{\mathrm{T}}$.

Output: $F \in \mathbb{R}^{\text{n.portfolios}\times 3}$. *# An n.portfolios × 3 matrix containing the return and variance for n.portfolios and the distance between the BCCEF and UEF.*

 # $|a|$ stands for the cardinality of set a
 # seq$(i, j, length)$ generate a regular sequence from i to j of length = 'length'
 # solve.QP$(r, V, ret, bound.constr)$ computes the Markowitz quadratic programming model with r being the expected return vector, V the variance matrix, ret the required return, and bound.constr the bound constraints
Begin
 # r.seq determines the return for the n.portfolios different portfolios
r.seq = seq(r.min, r.max, length = n.portfolios)
$p = 1$
 # optimal portfolio estimation for each return value in r.seq
for all (ret in r.seq) **do**
    # p.weights is a vector with the weights of the optimal portfolio constrained to ret and bounded constraints u and l
    p.weights = solve.QP(r, V, ret, bound.constr = (u, l))
    # if no feasible solution is obtained, solve again with no bound constraints
    if (p.weights == ∅) **then**
        p.weights = solve.QP(r, V, ret, bound.constr = ∅)
    end if
    # assets with a weight out of the bound values
    l.weights = which(p.weights ≤ l)
    u.weights = which(p.weights ≥ u)
    n.bound.weights = |l.weights| + |u.weights|
    # if bound constraints are not violated, the p.weights are the weights of the optimal portfolio
    if (n.bound.weights == 0) **then**
        temp.weights = p.weights
        temp.variance = $\text{p.weights}^{\mathrm{T}} \times \mathrm{V} \times \text{p.weights}$
    else
        # if bound constraints are violated, we search for the constrained optimal portfolio
        temp.weights = ∅
        temp.variance = large.number
        # compute the binary combinations for n.bound.weights assets: $2^{n.boung.weights}$
        combinations = $\{0, 1\}^{\text{n.bound.weights}}$
        n.combinations = $2^{\text{n.bound.weights}}$

```
    for all (j in combinations) do
      # Solve the problem by considering the bound constraints for the assets in combinations[j]
      (u, l) = combinations[j]
      p.weights = solve.QP(r, V, ret, bound.constr = (u, l))
      p.variance = p.weights^T × V × p.weights
      if (p.variance < temp.variance) then
        temp.weights = p.weights
        temp.variance = p.variance
      end if
    end for
  end if
  p.weights = temp.weights
  F[p, 1] = ret
  F[p, 2] = p.weights^T × V × p.weights
  F[p, 3] = F[p, 2] − UEF[p]
  p = p + 1
end for
return (F)
End begin
```

3.3. Configuring the genetic algorithm

Once we have devised an algorithm to compute the optimal BCCEF for a subset of assets, we can use it as the fitness function in the genetic algorithm. Given the bound and cardinality constraints, the target of the genetic algorithm is to find which specific assets will participate in the frontier by minimizing the distance reported by the fitness function of Algorithm 1. In the following we propose how to implement the main operators of the genetic algorithm.

3.3.1. Initial population

Instead of randomly select the initial population, we propose to feed the genetic algorithm with the population obtained by solving a shorter version of the model described in Eq. (1). The initial population is created following the proposal of Guijarro (2018), i.e. excluding the bound constraints. This allows individuals to be feasible by considering the cardinality constraint and to accelerate the convergence of the genetic algorithm, as shown in Section 4.

3.3.2. Crossover operator

In order to avoid a repair operator, both the crossover and mutation operators have been devised to ensure feasibility in individuals. Algorithm 2 presents the pseudo code for the crossover operator, where two parents (input) are crossed to produce two children (output). We first identify the assets included in any parent. As the parents meet the cardinality constraint, this ensures that the number of assets identified is between k and $2k$. Secondly, each child is composed by randomly selecting k from the aforementioned subset of assets, so that they comply with the cardinality constraint.

Algorithm 2 Pseudo code for crossover operator. Comments are placed after a hash (#) in italics.

Inputs: parent1, **parent2** $\in \{0,1\}^k$
Outputs: child1, **child2** $\in \{0,1\}^k$

```
# sample(set, size) takes a sample of the specified size from the elements in set without replacement
Begin
which.parent1 = which(parent1 == 1)
which.parent2 = which(parent2 == 1)
which.assets = which.parent1 ∪ which.parent2
child1[sample(which.assets, k)] = 1
child2[sample(which.assets, k)] = 1
return(child1, child2)
End begin
```

3.3.3. Mutation operator

The mutation operator randomly selects an asset in the portfolio and changes it for another asset not included in the portfolio. This maintains the cardinality of the mutated portfolio.

Algorithm 3 Pseudo code for mutation operator. Comments are placed after a hash (#) in italics.

Input: portfolio $\in \{0,1\}^k$
Output: portfolio $\in \{0,1\}^k$

```
# sample(set, size) takes a sample of the specified size from the elements in set without replacement
Begin
which.portfolio.0 = which(portfolio == 0)
which.portfolio.1 = which(portfolio == 1)
i = sample(which.portfolio.0, 1)
j = sample(which.portfolio.1, 1)
portfolio[i] = 1
portfolio[j] = 0
return(portfolio)
End begin
```

4. Computational results

We tested the performance of our proposal for finding the bounded and cardinality constrained efficient frontier using test problems for five major stock market indices available from OR-Library (Beasley, 1990): Hang Seng ($N = 32$), DAX 100 ($N = 85$), FTSE 100 ($N = 89$), S&P 100 ($N = 98$) and Nikkei 225 ($N = 225$). This library has been widely used in previous studies (Cesarone, Scozzari, & Tardella, 2013; Chang et al., 2000; Cura, 2009; Guijarro, 2018; Liagkouras & Metaxiotis, 2014; Lwin & Qu, 2013; Meghwani & Thakur, 2017; Woodside et al., 2011). Bruni, Cesarone, Scozzari, & Tardella (2016) have recently provided these datasets, in a version cleaned from errors and adjusted for dividends and for stock splits.

The experiments were carried out using 50 equally spaced desired return levels in the distance calculation between the UEF and the BCCEF in Algorithm 1. The minimum and maximum value for the return were obtained through the 0.1 and 0.9 percentiles of weekly assets return, respectively. In this way, the tracing frontier is devised within a sensible return range, excluding unrealistic values. We used $l_i = 0.01$ and $u_i = 1$ $(i = 1, \ldots, N)$ and $k \in \{10, 15\}$, following a similar approach to Woodside et al. (2011).

Along with the aforementioned features of the optimization problem, we must also determine other relevant parameters in the genetic algorithm. In our study the population size and the number of generations is set to 20. One could argue that both are small numbers, but there are two reasons why we decided to constrain these parameters: i) the high computational cost of Algorithm 1, which must be applied to every individual in the population, and ii) the quality of the initial population, which obtains such good solutions that 20 individuals was enough in the experiments we performed. Lastly, we set the probability of mutation to 0.2.

The implementation was carried out in R using the package GA (Genetic Algorithm), by adapting the original code for the proposed initial population, crossover and mutation operators. The quadratic programming model was also solved with R and the package quadprog, and no change in the original code was required. The system runs under macOS v.10.13.1, 1.6 GHz Intel Core i5 and 8 GB RAM.

The results are compared for different cardinalities $k \in \{10, 15\}$, different stock market indices, comparing random initial populations vs. proposed initial population, and here we report the performance of the model, both for the distance between the UEF and the BCCEF and the computation time. We argue that the investor can be interested in composing a frontier closed to the UEF as far as possible, but also in a sensible computation time, so both metrics must be kept in mind. We run each experiment 50 times in order to analyze the significance of the results from a statistical point of view. This makes it possible to account for a distribution of distances and computation times, and statistically analyze differences in mean.

Figs. 6 and 7 plot the distance between the UEF and the BCCEF for $k = 10$ and $k = 15$, respectively, and all the stock market indices considered. We represent the index (from ind1 – Hang Seng to ind5 – Nikkei 225) on the x-axis, along with the procedure for the initial population setting (A – random initial population, B – proposed initial population). The y-axis represents the distance between the aforementioned frontiers, i.e. an inverted similarity measure in terms of Guijarro (2018). Boxplots depict the distance distribution for the 50 experiments carried out. We can see noticeable differences for initial populations A and B regarding indices from ind2 to ind5. The proposed initial population, which makes use of the unbounded optimization in Guijarro (2018), improves the solution gathered by the random population. The only exception is index 1, because the low number of assets (32) enables both approaches to quickly converge to the optimal solution. To compare these distributions with a boxplot, we conducted a statistical analysis to confirm the significance of the differences. The nonparametric Mann-Whitney U test was applied to the distributions to find any differences between the initial population performances. Sample differences are statistically significant at a 99% confidence level except for the case of index 1 (Hang Seng).

We can therefore conclude that the initial population taken from the unbounded version of the cardinality constrained optimization model gives superior results in relation to the alternative of the random initial population, for both $k = 10$ and $k = 15$.

[Figure 6 about here.]

[Figure 7 about here.]

Figs. 8 and 9 report the computational times required by the model for $k = 10$ and $k = 15$, respectively. In the first case we can see that all the experiments were conducted in less than 2.1 minutes. When the cardinality is increased to $k = 15$, the computation time of Algorithm 1 rapidly increases; in the worst case, nearly 55 minutes were necessary to process the whole number of generations. Comparing both figures, it can be seen that computational times for indices 2, 3 and 4 remain quite similar and are substantially lower than those for indices 1 and 5. This is because in a relatively high number of instances in indices 1 and 5 the portfolios were compounded with weights in the bound limits, which involves comparing more portfolios with some weight assets in the bounds and others with some asset weights at zero.

Regarding the difference between the two initial population procedures, the Mann-Whitney U test reported no statistical difference with $k = 10$. However, in $k = 15$ we obtained statistically significant differences at a 99% confidence level for all indices. As stated below, the proposed initial population takes as a starting point a solution closer to the optimal solution than in the case of a random initial population. This enables a quicker convergence in such a way that the population is composed of mostly the same optimal solution at that moment, along with other sub-optimal solutions. In a specific generation the genetic algorithm does not obtain the fitness value of a solution if it has been computed in a previous generation. In this case, if the same solution is repeated in the population, the genetic algorithm only computes its fitness value once (Algorithm 1) and the computation time for the whole process is considerably reduced.

Along with the results outlined in Figs. 6–9, we include summary statistics for different performance measures of the reported portfolios in Appendix A. We must point that the proposed framework seeks to minimize the distance between the mean-variance frontier and the constrained frontier. Hence, the reported performance measures were not included in the optimization process.

[Figure 8 about here.]

[Figure 9 about here.]

5. Conclusions

This paper deals with the mean-variance optimization frontier problem when realistic constraints are considered. We focus on bound and cardinality constraints and our proposal hybridizes a genetic algorithm, which is applied to the identification of the assets in the portfolio, with a quadratic programming model which computes the asset weights.

Unlike other approaches, our model actually satisfies the cardinality constraint for the whole devised frontier, while previous models obtain the constrained frontier as the union of constrained sub-frontiers, each one considering the cardinality constraint, but with different assets in each sub-frontier. We give an illustrative example which shows that the compounded frontier does violate the cardinality constraint. Our proposal utilizes the same set of assets for all portfolios along the frontier, hence observing the cardinality constraint for the whole frontier and not only for individual portfolios.

The genetic algorithm focuses on the identification of the assets to be used in the frontier. The quadratic programming model extracts the portfolios solely on those assets. We also show that taking the cardinality constraint as an equality obtains suboptimal solutions when bound constraints are considered. This led us to propose a surrogate similarity measure for the mean-variance optimization problem under bound and cardinality constraints, as a modification of a previously stated similarity measure. Another realistic assumption we have addressed is the consideration of the required return' constraint as an inequality.

We propose specific initial population, crossover and mutation operators for the genetic algorithm in order to preserve the feasibility of the solutions throughout the entire process. By considering a dataset of five major stock market indices broadly used in the literature, we provide empirical evidence suggesting that the proposed initial population outperforms the classical random initial population. Furthermore, the computation burden is significantly reduced for cardinality $k = 15$. The proposed initial population includes the best individuals from the bound unconstrained optimization problem, which runs much faster than the constrained version. In a few generations, the population is mainly composed of the individuals with the highest fitness value, whilst the random initial population gets a more diversified population and needs to explore very poor solutions before achieving a good one.

An important future research area includes the consideration of the proposed framework in a multi-period setting. Under this approach, the analyst could manage the readjustment of assets' weights without changing the assets in his/her investment, and hence avoiding unnecessary transaction costs.

Acknowledgements

We would like to thank two anonymous referees for their constructive comments and suggestions that substantially improved this article.

References

Beasley, J. E. (1990). OR-Library: Distributing test problems by electronic mail. *The Journal of the Operational Research Society*, *41*(11), 1069–1072.

Bruni, R. & Cesarone, F. & Scozzari, A. & Tardella, F. (2014). A linear risk-return model for enhanced indexation in portfolio optimization. *OR Spectrum*, *37*(3), 735–759.

Bruni, R. & Cesarone, F. & Scozzari, A. & Tardella, F. (2016). Real-world datasets for portfolio selection and solutions of some stochastic dominance portfolio models. *Data in Brief*, *8*(), 858–862.

Cesarone, F. & Scozzari, A. & Tardella, F. (2013). A new method for mean-variance portfolio optimization with cardinality constraints. *Annals of Operations Research*, *205*(1), 213–234.

Chang, T. J. & Meade, N. & Beasley, J. E. & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, *27*(13), 1271–1302.

Chang, T. J. & Yang, S. C. & Chang, K. J. (2009). Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications*, *36*(7), 10529–10537.

Cura, T. (2009). Particle swarm optimization approach to portfolio optimization. *Nonlinear Analysis: Real World Applications*, *10*(4), 2396–2406.

Deng, G. F. & Lin, W. T. & Lo, C. C. (2012). Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. *Expert Systems with Applications*, *39*(4), 4558–4566.

Guijarro, F. (2018). A similarity measure for the cardinality constrained frontier in the mean–variance optimization model. *The Journal of the Operational Research Society*, *69*(6), 928–945.

Kalayci, C. B. & Ertenlice, O. & Akyer, H. & Aygoren, H. (2017). An artificial bee colony algorithm with feasibility enforcement and infeasibility toleration procedures for cardinality constrained portfolio optimization. *Expert Systems with Applications*, *85*, 61–75.

Li, D. & Ng, W-L. (2000). Optimal dynamic portfolio selection: Multiperiod mean-variance formulation. *Mathematical Finance*, *10*(3), 387–406.

Liagkouras, K. & Metaxiotis, K. (2014). A new Probe Guided Mutation operator and its application for solving the cardinality constrained portfolio optimization problem. *Expert Systems with Applications*, *41*(14), 6274–6290.

Lwin, K. & Qu, R. (2013). A hybrid algorithm for constrained portfolio selection problems. *Applied Intelligence*, *39*(2), 251–266.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77–91.

Meghwani, S. S. & Thakur, M. (2017). Multi-criteria algorithms for portfolio optimization under practical constraints. *Swarm and Evolutionary Computation*, *37*, 104–125.

Moscato, P. & Cotta, C. (2003). A gentle introduction to memetic algorithms. In F. Glover & G. Kochenberger (Eds.), *Handbook of metaheuristics* (pp. 105–144). Springer.

Peterson, B. & Carl, P. (2018). PerformanceAnalytics: Econometric Tools for Performance and Risk Analysis. *https://CRAN.R-project.org/package=PerformanceAnalytics*.

Ruiz-Torrubiano, R. & Suárez, A. (2009). A hybrid optimization approach to index tracking. *Annals of Operations Research*, *166*(1), 57–71.

Ruiz-Torrubiano, R. & Suárez, A. (2015). A memetic algorithm for cardinality-constrained portfolio optimization with transaction costs. *Applied Soft Computing*, *36*, 125–142.

Sadjadi, S. J. & Gharakhani, M. & Safari, E. (2012). Robust optimization framework for cardinality constrained portfolio problem. *Applied Soft Computing*, *12*(1), 91–99.

Salehpoor, I. B., & Molla-Alizadeh-Zavardehi, S. (2019). A constrained portfolio selection model at considering risk-adjusted measure by using hybrid meta-heuristic algorithms. *Applied Soft Computing*, *75*, 233–253.

Shaw, D. X. & Liu, S. & Kopman, L. (2008). Lagrangian relaxation procedure for cardinality-constrained portfolio optimization. *Optimisation Methods & Software*, *23*(3), 411–420.

Soleimani, H. & Golmakani, H. R. & Salimi, M. H. (2009). Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm. *Expert Systems with Applications*, *36*(3), 5058–5063.

Woodside-Oriakhi, M. & Lucas, C. & Beasley, J. E. (2011). Heuristic algorithms for the cardinality constrained efficient frontier. *European Journal of Operational Research*, *213*(3), 538–550.

List of Figures

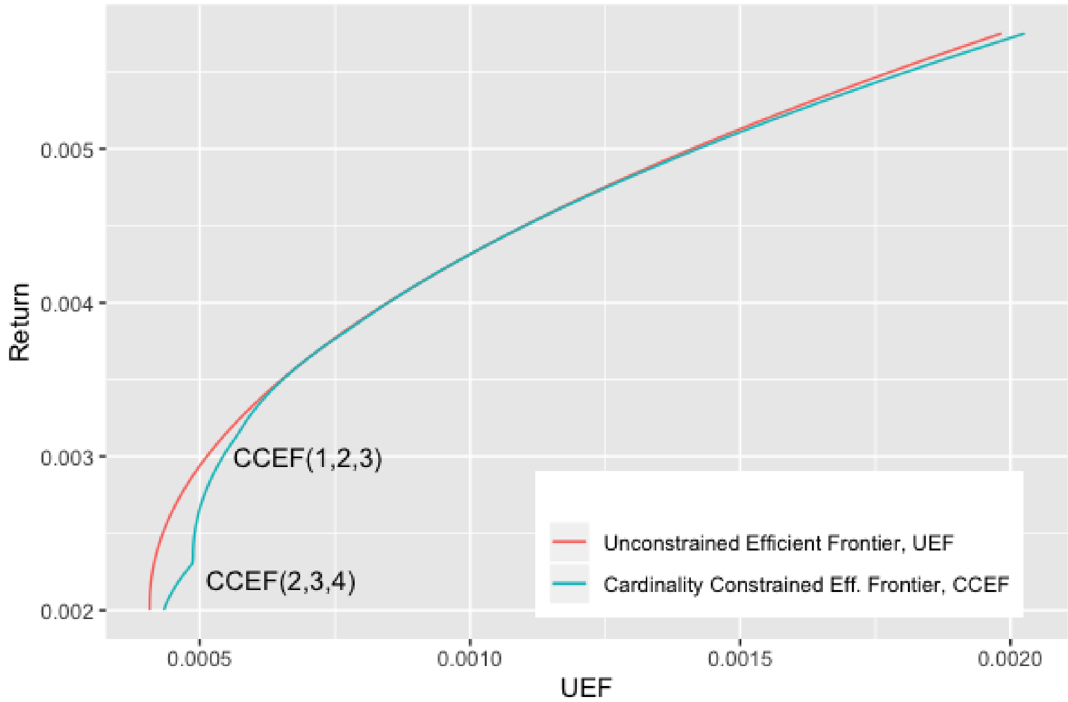

Figure 1. An example on $k = 3$ CCEF which actually uses 4 assets.

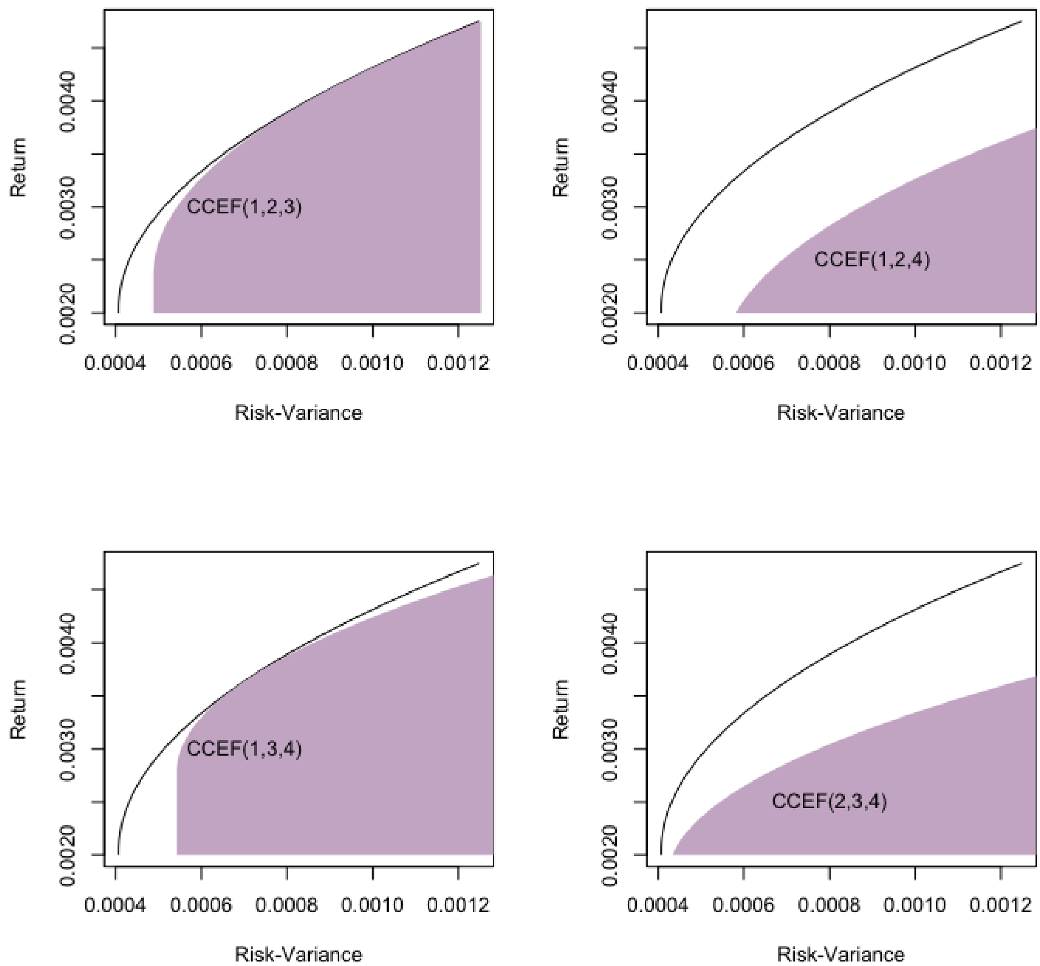

Figure 2. Different cardinality constrained frontiers with $k = 3$.

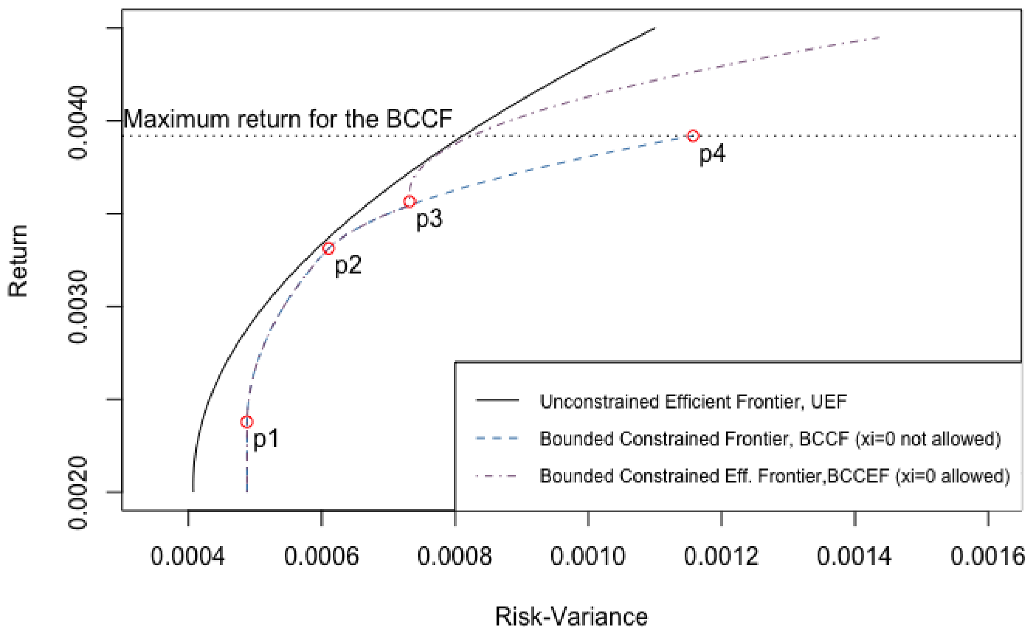

Figure 3. Comparison between UEF, BCCF and BCCEF.

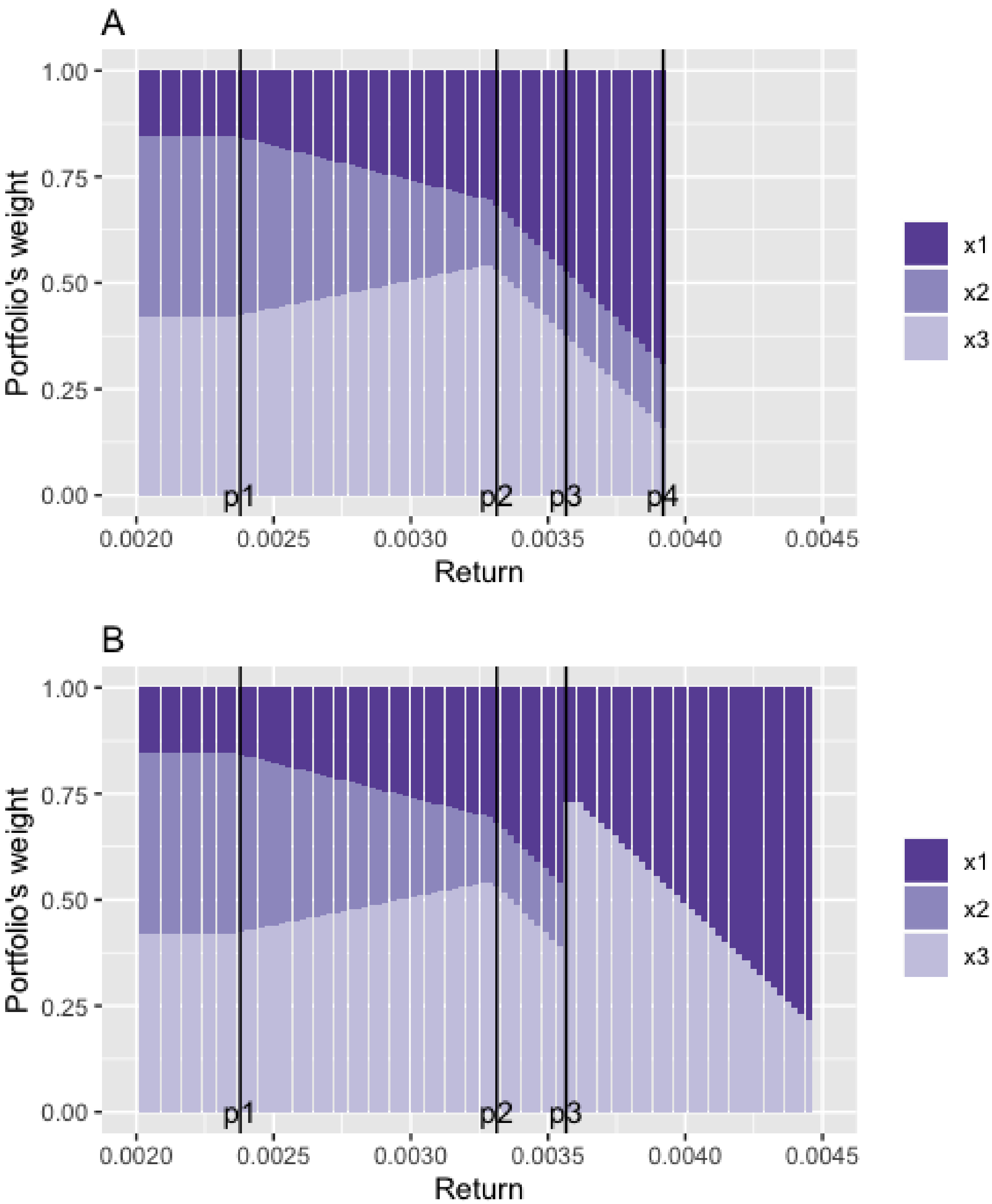

Figure 4. Portfolio's weight for different returns in A) BCCF and B) BCCEF.

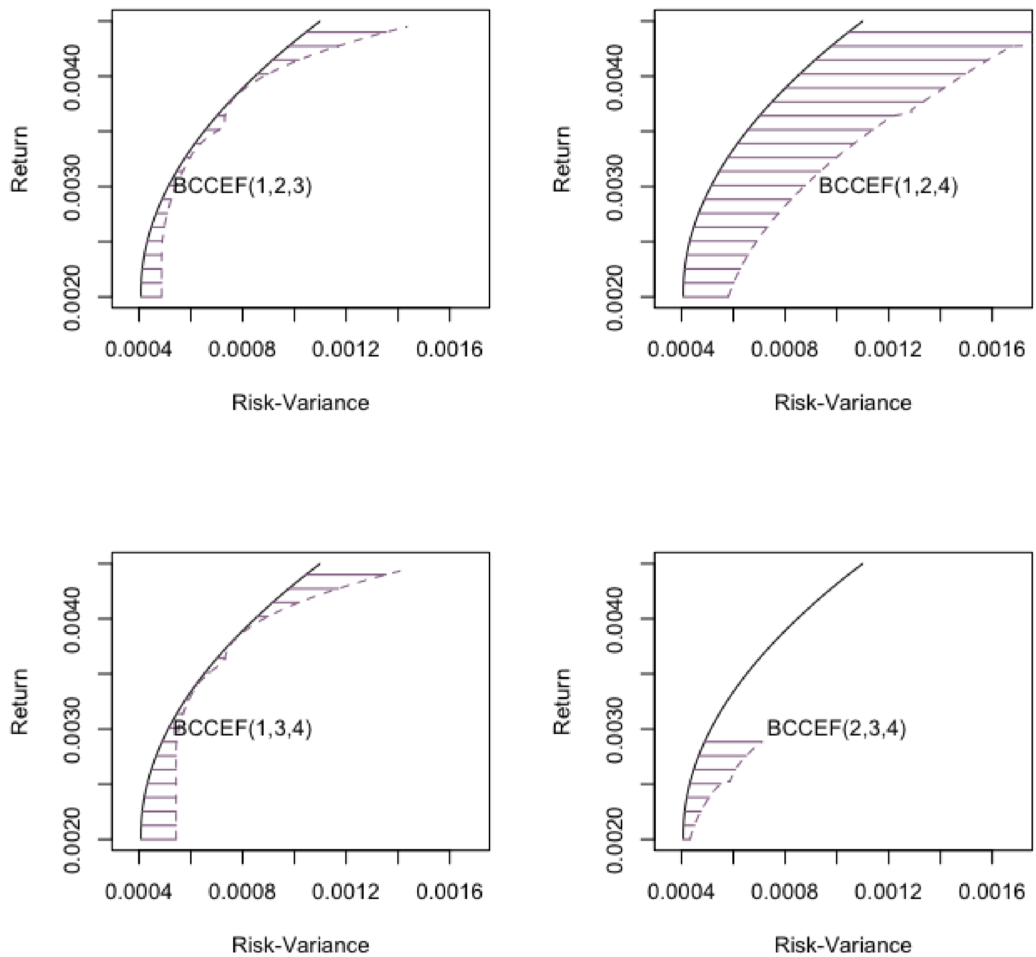

Figure 5. The surrogate similarity measure for different bounded and cardinality constrained frontiers with $k = 3$.

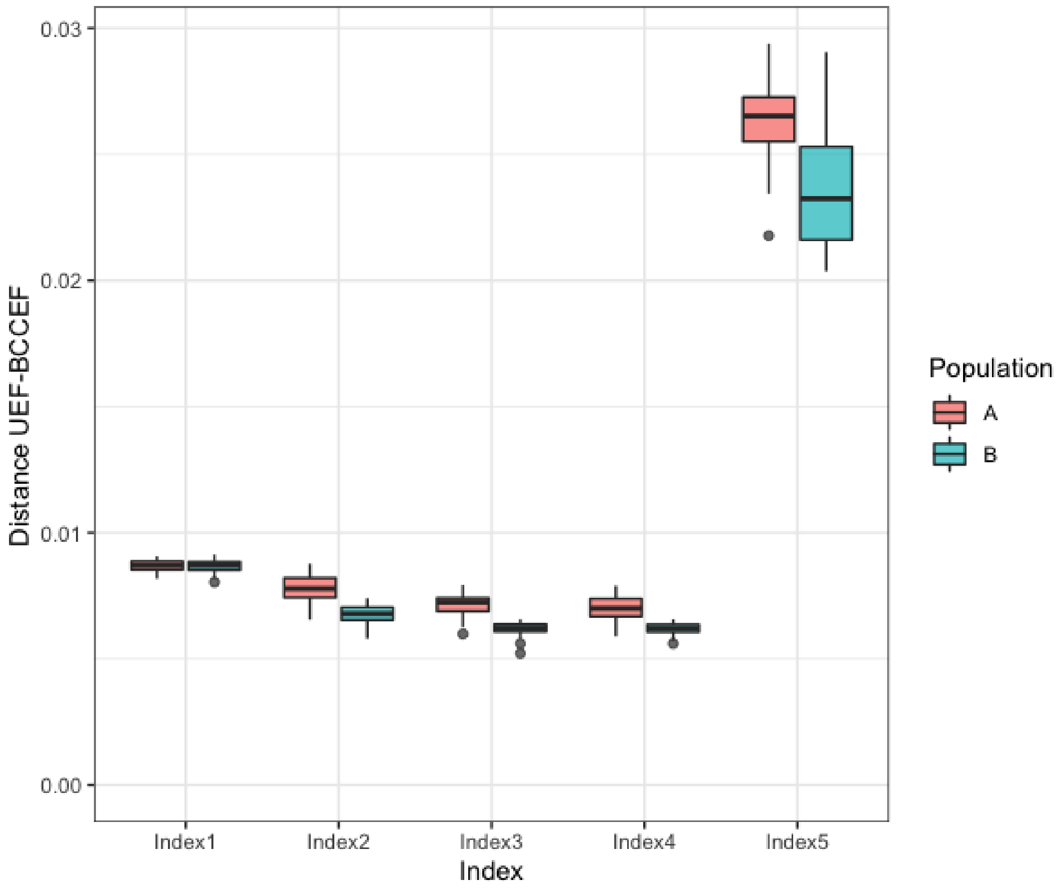

Figure 6. Distances between the UEF and the computed $k = 10$ BCCEF for different indices. Comparison between random population (A) and the proposed initial population (B).

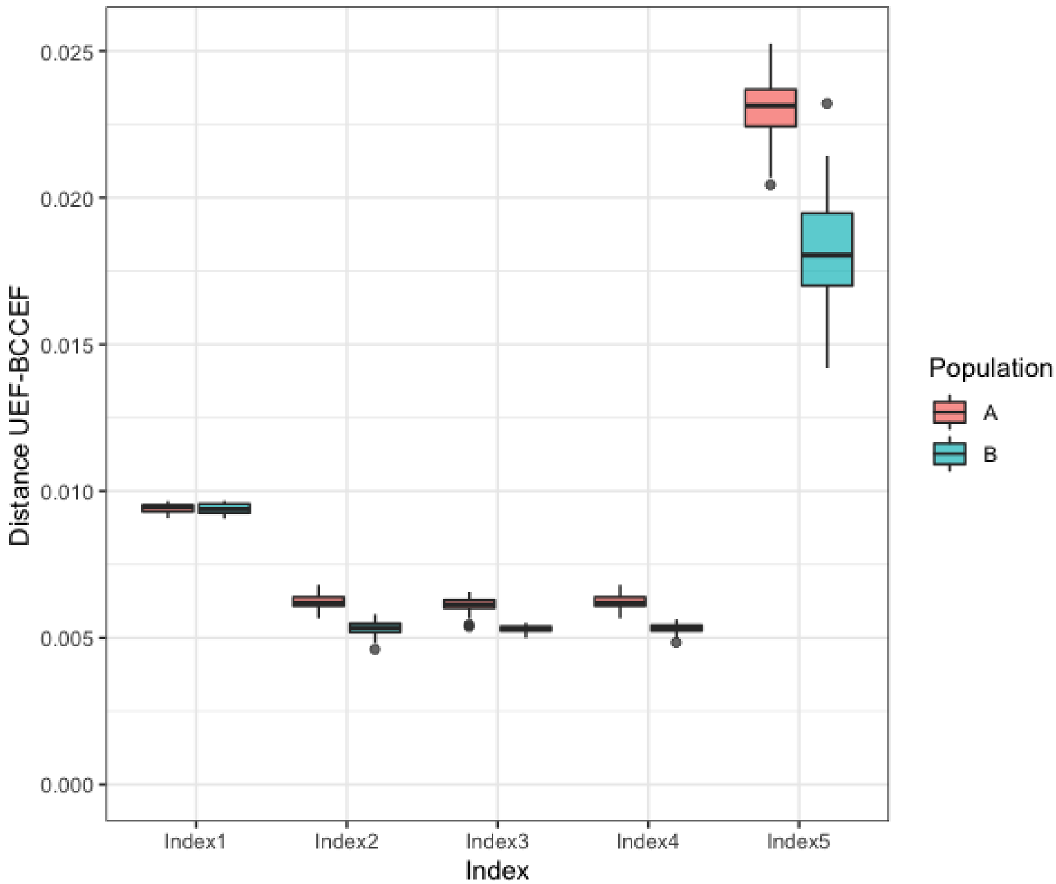

Figure 7. Distances between the UEF and the computed $k = 15$ BCCEF for different indices. Comparison between random population (A) and the proposed initial population (B).

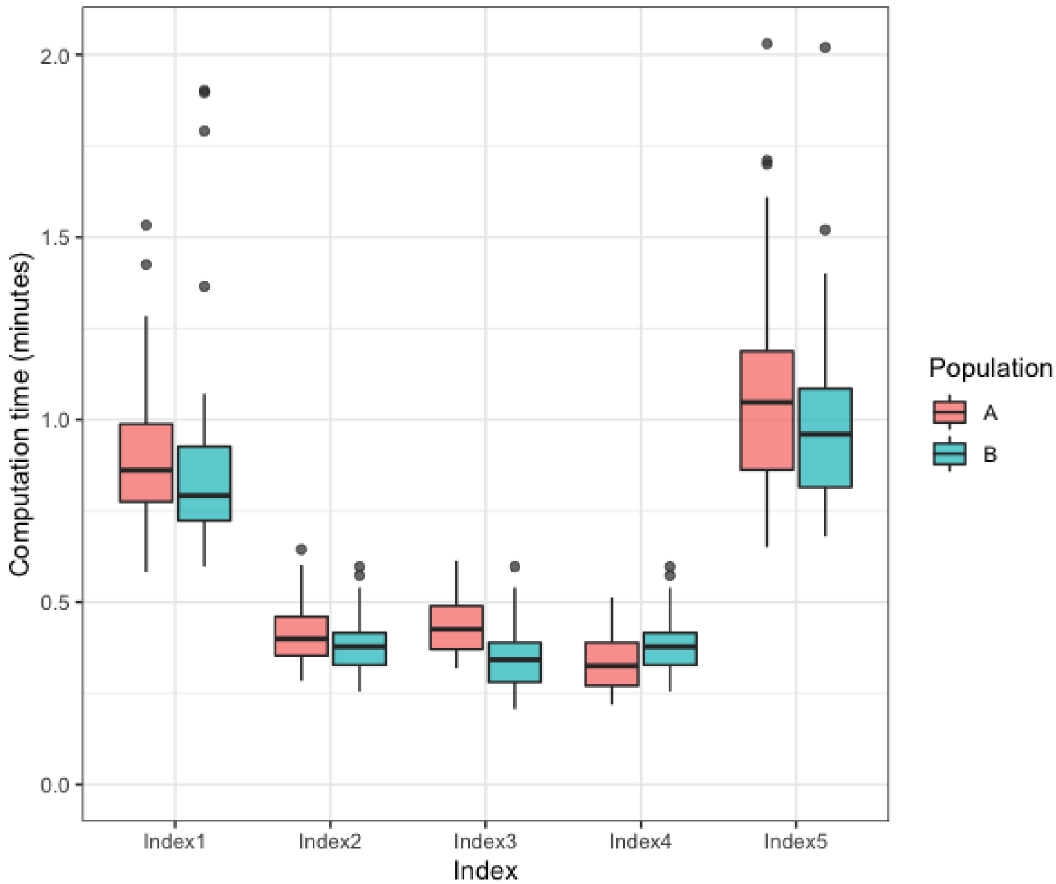

Figure 8. Time computation of the BCCEF for $k = 10$ and different indices. Comparison between random population (A) and the proposed initial population (B).

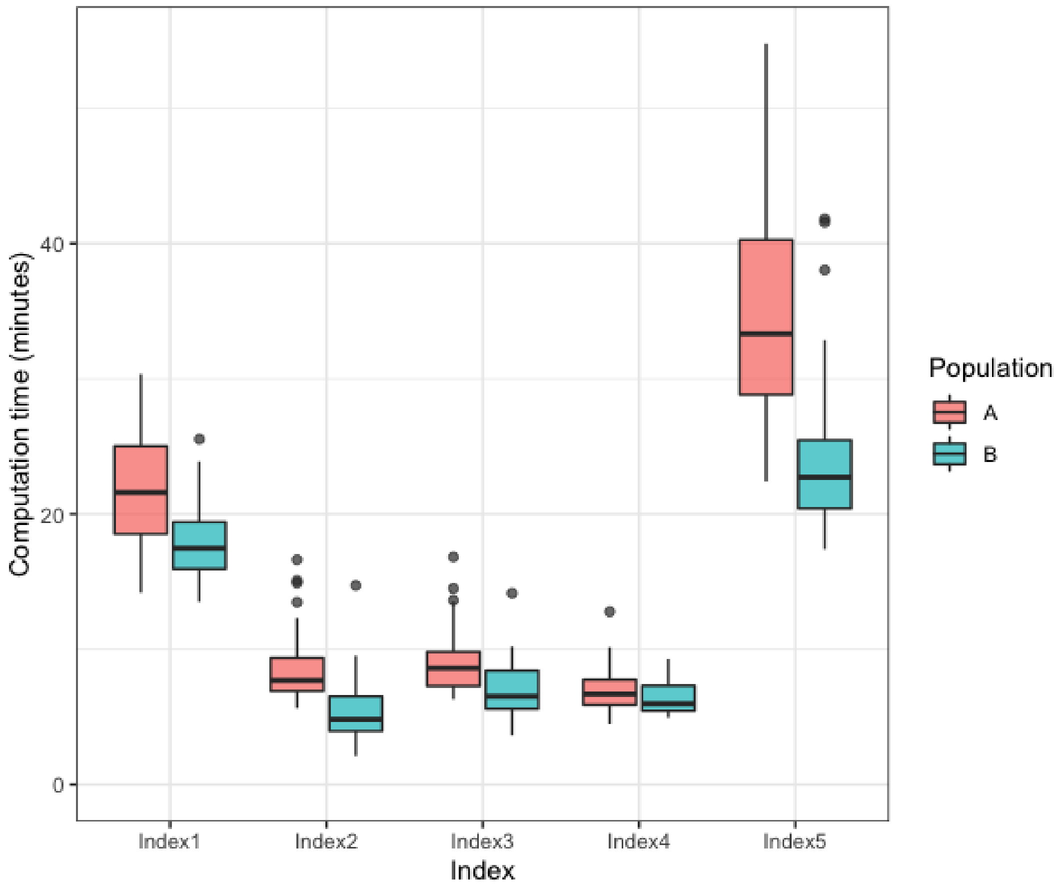

Figure 9. Time computation of the BCCEF for $k = 15$ and different indices. Comparison between random population (A) and the proposed initial population (B).

List of Tables

Table 1. Data example from Chang et al. (2000).

Asset	Return	Standard deviation	Covariance matrix			
			1	2	3	4
1	0.004798	0.046351	0.002148	0.000168	0.000203	0.000418
2	0.000659	0.030586		0.000935	0.000153	0.000109
3	0.003174	0.030474			0.000929	0.000091
4	0.001377	0.035770				0.001279

Appendix A. Performance measures of the reported frontiers

Several measures have been computed to give insight regarding the performance of portfolios in the reported frontiers: alfa and beta (from the CAPM model), tracking error, information ratio, Treynor ratio, Sharpe ratio, semi–variance, downside deviation, Omega and VaR. We have used the PerformanceAnalytics package in R (Peterson & Carl, 2018), a collection of econometric functions for performance and risk analysis. We refer to that package for a precise definition of those performance measures. The following tables give summary statistics of these measures:

[Table 2 about here.]

[Table 3 about here.]

[Table 4 about here.]

[Table 5 about here.]

[Table 6 about here.]

List of Tables

Table A1. Summary statistics of the performance measures of portfolios in the BCCEF. Index1 and $k = \{10, 15\}$. Results have been obtained by using the proposed initial population for the GA.

	Alfa	Beta	TE	IR	Treynor	Sharpe	$\sigma^2_{\texttt{semi}}$	DD	Ω	VaR
Panel A: $k = 10$										
Min.	0.001	0.537	0.114	-0.353	0.266	0.125	0.025	0.015	0.891	-0.069
1st Qu.	0.002	0.606	0.149	0.306	0.396	0.176	0.027	0.017	1.050	-0.049
Median	0.004	0.672	0.167	1.247	0.646	0.242	0.030	0.018	1.290	-0.043
Mean	0.005	0.676	0.193	1.093	0.659	0.223	0.032	0.019	1.281	-0.045
3rd Qu.	0.007	0.737	0.226	1.852	0.901	0.262	0.036	0.021	1.513	-0.040
Max.	0.011	0.905	0.394	2.333	1.354	0.288	0.050	0.029	1.647	-0.035
Panel B: $k = 15$										
Min.	0.001	0.505	0.132	-0.299	0.287	0.134	0.024	0.015	0.893	-0.066
1st Qu.	0.002	0.558	0.161	0.112	0.412	0.174	0.026	0.016	1.003	-0.045
Median	0.005	0.608	0.171	1.225	0.708	0.258	0.029	0.017	1.306	-0.040
Mean	0.005	0.615	0.193	1.105	0.722	0.234	0.031	0.018	1.293	-0.042
3rd Qu.	0.008	0.663	0.220	2.011	0.995	0.285	0.034	0.019	1.555	-0.038
Max.	0.011	0.812	0.334	2.470	1.350	0.303	0.045	0.025	1.709	-0.033

Portfolios' performance measures: Alfa, Beta, *TE* Tracking Error, *IR* Information Ratio, *Treynor* Treynor ratio, *Sharpe* Sharpe ratio, $\sigma^2_{\texttt{semi}}$ Semi–variance, *DD* Downside deviation, Ω Omega and *VaR* Value at Risk.

Table A2. Summary statistics of the performance measures of portfolios in the BCCEF. Index2 and $k = \{10, 15\}$. Results have been obtained by using the proposed initial population for the GA.

	Alfa	Beta	TE	IR	Treynor	Sharpe	$\sigma^2_{\texttt{semi}}$	DD	Ω	VaR
Panel A: $k = 10$										
Min.	0.001	0.162	0.081	-0.031	0.234	0.155	0.013	0.008	0.975	-0.047
1st Qu.	0.002	0.441	0.106	0.532	0.370	0.219	0.015	0.009	1.164	-0.026
Median	0.003	0.495	0.119	1.146	0.468	0.272	0.017	0.010	1.344	-0.023
Mean	0.004	0.497	0.136	1.317	0.731	0.282	0.018	0.010	1.508	-0.025
3rd Qu.	0.007	0.558	0.159	2.129	1.008	0.346	0.020	0.011	1.901	-0.022
Max.	0.011	0.692	0.301	2.949	4.475	0.409	0.035	0.019	2.343	-0.017
Panel B: $k = 15$										
Min.	0.001	0.127	0.094	0.124	0.267	0.182	0.013	0.007	1.022	-0.039
1st Qu.	0.002	0.434	0.107	0.481	0.370	0.233	0.014	0.008	1.151	-0.024
Median	0.003	0.470	0.115	1.034	0.503	0.286	0.016	0.009	1.358	-0.021
Mean	0.004	0.468	0.131	1.367	0.753	0.303	0.017	0.009	1.553	-0.022
3rd Qu.	0.007	0.505	0.150	2.329	1.060	0.384	0.018	0.010	2.011	-0.020
Max.	0.011	0.610	0.255	3.389	5.908	0.438	0.031	0.017	2.512	-0.016

As for Table A1.

Table A3. Summary statistics of the performance measures of portfolios in the BCCEF. Index3 and $k = \{10, 15\}$. Results have been obtained by using the proposed initial population for the GA.

	Alfa	Beta	TE	IR	Treynor	Sharpe	$\sigma^2_{\texttt{semi}}$	DD	Ω	VaR
Panel A: $k = 10$										
Min.	0.000	0.674	0.040	-0.536	0.122	0.123	0.014	0.009	0.917	-0.050
1st Qu.	0.001	0.774	0.075	0.570	0.218	0.188	0.016	0.010	1.104	-0.028
Median	0.002	0.815	0.082	1.389	0.290	0.242	0.017	0.010	1.295	-0.024
Mean	0.003	0.828	0.096	1.359	0.336	0.241	0.018	0.011	1.365	-0.025
3rd Qu.	0.005	0.865	0.117	2.194	0.456	0.297	0.020	0.011	1.644	-0.022
Max.	0.007	1.178	0.227	2.889	0.707	0.337	0.036	0.021	1.872	-0.019
Panel B: $k = 15$										
Min.	0.000	0.536	0.045	-0.344	0.139	0.136	0.014	0.008	0.937	-0.045
1st Qu.	0.001	0.745	0.068	0.475	0.214	0.189	0.015	0.009	1.090	-0.025
Median	0.002	0.777	0.081	1.415	0.305	0.254	0.016	0.009	1.309	-0.022
Mean	0.003	0.779	0.092	1.378	0.357	0.252	0.017	0.010	1.382	-0.023
3rd Qu.	0.005	0.805	0.112	2.331	0.502	0.318	0.019	0.010	1.700	-0.021
Max.	0.008	0.996	0.258	3.105	1.013	0.369	0.033	0.019	2.003	-0.017

As for Table A1.

Table A4. Summary statistics of the performance measures of portfolios in the BCCEF. Index4 and $k = \{10, 15\}$. Results have been obtained by using the proposed initial population for the GA.

	Alfa	Beta	TE	IR	Treynor	Sharpe	$\sigma^2_{\texttt{semi}}$	DD	Ω	VaR
Panel A: $k = 10$										
Min.	0.000	0.279	0.040	.0.324	0.118	0.093	0.009	0.005	0.631	.0.047
1st Qu.	0.001	0.405	0.060	0.367	0.196	0.136	0.010	0.006	0.756	.0.018
Median	0.002	0.437	0.067	0.845	0.253	0.171	0.011	0.007	0.879	.0.016
Mean	0.002	0.453	0.079	0.908	0.360	0.178	0.012	0.007	0.975	.0.017
3rd Qu.	0.004	0.474	0.092	1.441	0.488	0.214	0.013	0.007	1.182	.0.015
Max.	0.008	1.059	0.252	2.739	1.928	0.359	0.036	0.020	1.976	.0.012
Panel B: $k = 15$										
Min.	-0.001	0.563	0.025	-1.553	0.120	0.139	0.012	0.007	0.806	-0.050
1st Qu.	0.001	0.752	0.067	-0.240	0.210	0.205	0.015	0.009	0.950	-0.031
Median	0.002	0.858	0.079	1.662	0.339	0.305	0.017	0.009	1.343	-0.023
Mean	0.003	0.885	0.097	1.207	0.350	0.274	0.020	0.011	1.322	-0.027
3rd Qu.	0.005	1.001	0.125	2.470	0.481	0.333	0.024	0.013	1.677	-0.021
Max.	0.008	1.349	0.230	2.945	0.722	0.365	0.039	0.022	1.871	-0.016

As for Table A1.

Table A5. Summary statistics of the performance measures of portfolios in the BCCEF. Index5 and $k = \{10, 15\}$. Results have been obtained by using the proposed initial population for the GA.

	Alfa	Beta	TE	IR	Treynor	Sharpe	$\sigma^2_{\mathtt{semi}}$	DD	Ω	VaR
Panel A: $k = 10$										
Min.	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.051
1st Qu.	0.001	0.429	0.144	0.572	0.116	0.061	0.018	0.012	1.202	-0.032
Median	0.002	0.470	0.151	0.681	0.149	0.078	0.019	0.013	1.256	-0.030
Mean	0.002	0.454	0.152	0.729	0.200	0.085	0.019	0.013	1.247	-0.029
3rd Qu.	0.002	0.510	0.162	0.923	0.222	0.108	0.021	0.014	1.351	-0.028
Max.	0.005	0.795	0.253	1.642	1.004	0.196	0.034	0.022	1.711	0.000
Panel B: $k = 15$										
Min.	0.000	0.232	0.141	0.199	0.009	0.013	0.015	0.010	1.060	-0.050
1st Qu.	0.001	0.371	0.155	0.487	0.116	0.059	0.017	0.011	1.196	-0.029
Median	0.001	0.404	0.160	0.567	0.143	0.070	0.018	0.012	1.231	-0.027
Mean	0.002	0.401	0.166	0.670	0.225	0.087	0.018	0.012	1.292	-0.028
3rd Qu.	0.002	0.428	0.170	0.772	0.249	0.105	0.019	0.013	1.344	-0.025
Max.	0.005	0.579	0.249	1.570	1.134	0.210	0.032	0.020	1.798	-0.022

As for Table A1.