Article

A Biased-Randomized Iterated Local Search Algorithm for Rich Portfolio Optimization

Renatas Kizys [1], **Angel A. Juan** [2], **Bartosz Sawik**[3,*] **and Laura Calvet** [2]

[1] Subject Group of Economics and Finance, Portsmouth Business School, University of Portsmouth, UK, Tel.: +44 2392 844635; renatas.kizys@port.ac.uk

[2] IN3 – Department of Computer Science, Open University of Catalonia, Spain; {ajuanp, lcalvetl}@uoc.edu

[3] Department of Applied Computer Science, AGH University of Science and Technology, Poland; University of California at Berkeley, USA; b_sawik@cal.berkeley.edu, bsawik@zarz.agh.edu.pl

* Correspondence: b_sawik@cal.berkeley.edu, bsawik@zarz.agh.edu.pl; Tel.: +491-5777-444-221

Received: date; Accepted: date; Published: date

Abstract: This research develops an original algorithm for rich portfolio optimization (ARPO), considering more realistic constraints than those usually analyzed in the literature. Using a matheuristic framework that combines an iterated local search metaheuristic with quadratic programming, ARPO efficiently deals with complex variants of the mean-variance portfolio optimization problem, including the well-known cardinality and quantity constraints. ARPO proceeds in two steps. First, a feasible initial solution is constructed by allocating portfolio weights according to the individual return rate. Secondly, an iterated local search framework, which makes use of quadratic programming, gradually improves the initial solution throughout an iterative combination of a perturbation stage and a local search stage. According to the experimental results obtained, ARPO is very competitive when compared against existing state-of-the-art approaches, both in terms of the quality of the best solution generated as well as in terms of the computational times required to obtain it. Furthermore, we also show that our algorithm can be used to solve variants of the portfolio optimization problem, in which inputs (individual asset returns, variances and covariances) feature a random component. Notably, the results are similar to the benchmark constrained efficient frontier with deterministic inputs, if variances and covariances of individual asset returns comprise a random component. Finally, a sensitivity analysis has been carried out to test the stability of our algorithm against small variations in the input data.

Keywords: Constrained portfolio optimization; metaheuristics; efficiency indices; financial assets; iterated local search; biased randomization

1. Introduction

Since Markowitz [1], mean-variance optimization has become the workhorse model for portfolio selection. As discussed in [2], Markowitz' s theory of portfolio selection has had a major impact on academic research and on the industry of financial services[1]. An important assumption underlying this model is that investors are concerned about both the expected returns from their investment and the risk from that investment, where risk is defined as the variance of future returns. By optimally allocating appropriate weights to imperfectly correlated risky assets, investors can reduce the variance of future portfolio returns and thus diversify their investment.

Because of the non-negativity constraint on the level of investment in each asset, a closed analytical solution is generally not feasible [8], and optimization methods need to be used to determine optimal portfolio weights. Since the mean-variance portfolio formulation, an enormous amount of papers have

[1] Other recent studies on the theory of portfolio selection include [3] and [4], among many others. Mean-variance portfolio optimization has also provided practical solutions to defined contribution pension schemes [5], hedging mechanisms [6], as well as to long-term fixed-income investments [7]

been published extending or modifying the basic model in three directions [9]. The first path goes to simplification of the amount or the type of input data [10]. The second path focuses on the introduction of an alternative measure of risk [11]. The third path involves incorporation of the additional criteria and constraints [12,13]. Traditional optimization methods include, among others, linear programming [14–20], quadratic programming [21–23] and stochastic programming [24]. However, traditional optimization methods (notably, linear and quadratic programming) are plagued by a number of caveats. A notable characteristic of these methods is that they work only for problems that rely upon strict theoretical rules, albeit with a limited practical appeal [2]. Unfortunately, the complexity of the mean-variance model for portfolio optimization increases dramatically upon adding additional constraints that are needed to provide a rich representation of the investor's portfolio choice, which reduces the efficiency of exact methods in other than small-size instances. More recently, metaheuristic approaches have been proposed in the literature as a way to surpass these limitations [8,25–28].

Accordingly, this paper focuses on a single-period version of the so-called constrained mean-variance portfolio optimization problem. We will assume in this work that all assets are risky assets (stocks). However, the solving approach introduced here could be also used for similar optimization problems regarding commodities, futures, options and swaps. Moreover, as in [29,30], our approach can be extended to include risk-free assets. The decision problem involves minimizing the portfolio variance for a given required rate of return. Portfolio weights add up to one and are constrained to take on non-negative values only. The latter assumption rules out short sales and thus places a constraint on excessive risk taking of investors. Under the above assumptions, the unconstrained efficient frontier (UEF) can be determined that gives, for each user-specified expected return, the minimum associated risk of investment. However, more realistic portfolio selection problems may involve additional constraints. First, justified on the grounds of the investor's preference and/or taste, the *pre-assignments* force some specific assets to be included in the portfolio. Second, the *quantity* constraint keeps the quantity of each selected asset within user-specified *floor* and *ceiling* values. The ceiling rules out excessive exposure to a specific asset. The floor is introduced in order to rule out the possibility of tiny (and therefore disproportionately costly) fractions of assets to be included in the portfolio. Third, the *cardinality* constraint, which imposes a floor and a ceiling on the number of assets included in the portfolio, accounts for the fact that diversification benefits decrease when the portfolio features a huge number of assets. When the above additional constraints are incorporated in the portfolio optimization problem, the constrained efficient frontier (CEF) can be obtained. In the presence of these rich constraints, the problem becomes NP-hard [31] and, thus, exact optimization methods quickly lose their efficiency as the number of considered assets grows. The cardinality constraint also implies that the mean-variance frontier can become discontinuous for certain values of expected return [32]. In summary, as pointed out by [33], the cardinality and quantity constraints make large-size instances of the problem to be computationally intractable using traditional optimization approaches. Because our research involves the CEF, we devise a matheuristic algorithm for rich portfolio optimization (ARPO) that is based on the combination of an iterated local search (ILS) metaheuristic [34,35], quadratic programming, and biased randomization strategies [36–38]. The contribution of this study is fourfold. First, we show that using a carefully devised matheuristic solver can significantly reduce the minimum running time necessary to obtain near-optimal solutions. Indeed, as it will be discussed later, the computing-time performance of ARPO is significantly better than those of the solvers proposed by [39–41]. Second, in terms of the minimum average percentage loss, the CEF determined by ARPO is approximately as close (or even closer in most cases) to the unconstrained efficient frontier as in the aforementioned studies. Third, the adoption of an initial solution that is based on a well-chosen criterion makes our solver more flexible when solving tight instances. The rate of return on an individual asset is used as a main criterion to construct the initial solution. It is worth noting that this criterion provides the best possible solution in terms of guarantying feasibility of the required portfolio return –i.e., if this initial solution is not feasible, then the problem has no feasible solution. By contrast, in most of the existing approaches, an initial solution is randomly drawn and, hence, the feasibility of these initial solutions cannot be guaranteed. Fourth and foremost, our paper relaxes the widely held assumption that inputs (individual asset returns, variances and covariances) are accurately measured and deterministic. In this regard, we show that ARPO can be used to solve portfolio optimization problems, in which a number of scenarios comprising uncertain returns, variances and covariances are studied and compared with the CEF-ARPO with accurate inputs. As expected, the CEF-ARPO with uncertain individual asset returns manifests in a higher portfolio variance value for a given required rate of return. By contrast, no added uncertainty to individual asset returns translates into no material change to the CEF-ARPO with accurate inputs when variances and covariances between individual asset returns are subjected to a random disturbance. Finally, we further carry out a stability analysis – involving small variations in cardinality and quantity constraints – that shows no material deviations from the benchmark CEF in terms of the portfolio variance and computational time.

The remainder of this article is organized as follows. Section 2 provides a review of the literature related to the theme, also discussing the limitations of traditional methods and the need for new approaches based on metaheuristics. Section 3 gives a formal description of the optimization problem being considered, while Section 4 provides an overview of the ARPO algorithm, including its pseudo-code for quick implementation. Section 5 introduces the numerical experiments performed to test the algorithm performance, while the results are discussed in Section 6. Finally, Section 7 highlights the main contributions of this work and Section 8 outlines plans for future research.

2. Need for new metaheuristic- and learning-based approaches

An updated literature review on the portfolio optimization problem can be found in [2]. For this reason, our review focuses on analyzing the limitations of traditional approaches and discussing about the need of new algorithms that consider richer constraints and large-scale instances.

Traditional optimization methods feature a number of theoretical and practical limitations. A notable characteristic of these methods is that they work only for problems that typically rely upon strict deterministic rules. First, they can produce wrong solutions when the portfolio selection problem under consideration has one or more local maxima in addition to a global maximum [8]. Second, the optimal solution may be disguised by the presence of estimation errors in expected returns and variances that are used as inputs [2,42]. Third, the optimal solution is notoriously unstable, as small changes in inputs can cause large changes in the optimal portfolio weights [43]. Fourth, in terms of out-of-sample performance, traditional optimization methods are sometimes no better that the naïve portfolio, wherein all assets are allocated the same weight [44]. Along similar lines, [45] argues that traditional optimization models can generate accurate in-sample forecasts, but their out-of-sample performance is generally weak. Fifth, a more realistic portfolio selection problem is typically confined to a subset of assets, which are selected by imposing additional constraints. A number of restrictions are specified in an investment management agreement between a client and a portfolio manager [2]. The client may impose a limit on the number of assets in the portfolio (*cardinality constraints*) [46]. Further, the client may also ask the manager to invest or not to invest in certain industries or companies. Therefore, the so-called *pre-assignment constraints* may be used that pre-assign certain industries or assets in the portfolio. The client may further impose a discretionary limit on exposure to certain industries or assets that aims at keeping the quantity of each asset within a given range (*quantity constraints*). Another battery of constraints is dictated by the presence of *transaction costs* [2,18]. If one or more of these constraints are introduced in portfolio optimization, indeterminacies in the efficient portfolio frontier may arise further limiting the usefulness of traditional optimization methods.

In addition to the above theoretical limitations, traditional optimization methods have received a weak support in practice. Whilst diversification is a dominant portfolio selection strategy that contributes to lowering the risk of portfolio investment, in practice investors invest in fewer assets due to a variety of reasons. First, the administration of large portfolios can be cumbersome [41] or involve transaction costs in the form of bank and broker fees [47]. Second, there is evidence that investors' desire to diversify is limited [48–50]. Third, [8] and [22] argue that diversification can be achieved by investing in a small, yet well-chosen sub-set of assets. Similarly, [51] assert that, in most practical situations, investing in a portfolio comprising between 10 and 20 individual stocks can reasonably reduce the risk of investment. By the same token, [14] suggests including between 15 and 20 assets to diversify away most diversifiable risk. [52] find that that, quite often, the optimal diversification should not use more than 27 assets in real-life investment. Moreover, a smaller number of assets imply less number of parameters to be estimated by using sample information and, therefore, less room for estimation error [44]. [42] provides a detailed account of advantages and disadvantages of traditional optimization methods. Specifically, he argues that these methods do not appeal to practitioners, since they do not make investment sense and do not have investment value. Such financial irrelevance is exacerbated by the fact that traditional optimization methods tend to over-weight assets with large returns and small variances and under-weight assets with small returns and large variances [53]. However, [2] maintain that the presence of theoretical and practical limitations does not invalidate Markowitz's theory of portfolio selection. Rather, traditional optimization methods need to be modified and new methods need to be developed, so that the existing gap between theory and practice can be bridged.

In order to address some of the theoretical and practical limitations of more classical approaches, metaheuristic methods to portfolio optimization have emerged in the literature [8,25–28]. Metaheuristics emerged from simple heuristics that were proposed to address some of the weaknesses inherited by traditional optimization methods. A heuristic is a common-sense and experience-based solution search method. [14] proposes a simple heuristic based on the beta as a risk measure to solve the portfolio selection problem. [54] builds upon the expected return-variance ratio to develop a heuristic that determines an optimal portfolio location on the efficient frontier. More specifically, they construct a decision rule that determines whether the asset will enter the portfolio and calculates its weight in the portfolio. A simplification to the decision rule in [54] is made by [55]. In particular, he derives a heuristic that is based on the expected return-semi-variance ratio that assumes the average correlation between securities to be zero. The use of the semi-variance as a measure of risk owes to the notion that investors are more concerned about downside risk than upside risk. [55] demonstrates that the return-risk heuristic that uses the downside risk measure provides better investment performance than traditional optimization methods for long-term investments, albeit not for short-term ones. A comparison between the return-risk heuristic and two traditional optimization methods is performed by [53]. He finds that the return-risk heuristic produces the highest return along with the highest standard deviation and the lowest semi-deviation among the three portfolios. However, heuristics always return a local optimum, which may or may not be global optimum. The use of these simple return-risk heuristics for the constrained portfolio optimization problem has been discontinued in the last two decades. In fact, an increasing number of studies have tended to focus on metaheuristics. Metaheuristic search methods are less restrictive than traditional optimization methods and thus can be tailored to solve a particular optimization problem that features a number of constraints. Metaheuristics have been employed in several studies on portfolio optimization, such as [32,39–41,56–59] to name just few. However, most of these studies feature the use of random initialization to the portfolio selection problem or simply downplay their strategies of portfolio initialization. [41] are maybe an exception to this rule. In addition to the randomly generated initial solution, these authors propose two simple heuristics. First, they construct the portfolio that produces the maximum possible return, independently of the risk. Second, they use the final solution of the previously computed point on the efficient frontier. The amount of processor time it takes for the local search algorithm to reach the optimal solution on the efficient frontier may be shorter if a sensible simple heuristic is used to construct an initial solution.

Although this paper focuses on the use of metaheuristics to solve rich versions of the portfolio optimization problem, applications of machine learning and deep learning algorithms can be explored as well. These algorithms are becoming increasingly popular in computational finance, particularly for addressing optimization and prediction problems in scenarios characterized either by a high number of features and samples or by a high level of uncertainty and dynamism. Thus, for instance, [60] combines a grid trading strategy with machine learning to optimize trading in the Forex Market. Likewise, [61] proposes a theoretical framework, based on deep reinforcement learning, for finance portfolio management. Similarly, [62] develops and tests a deep learning algorithm to address portfolio management with proportional transactions cost. As discussed in [63], the combination of metaheuristics with machine learning for solving optimization problems with dynamic inputs is also becoming an emergent trend in the optimization literature.

3. Problem definition

In this section, the constrained mean-variance portfolio optimization problem is described. Section 3.1 provides an overview of the parameters, variables and constraints used in the problem. Section 3.2 contains the mathematical model, which is based on the one provided in [41]. For a comprehensive overview of the formulations of the portfolio selection problem, see [64]. Finally, Section 3.3 summarizes the algorithm's inputs and outputs.

3.1. Problem parameters, variables, and constraints

A set of n assets is given by the market, $A = \{a_1, a_2, \ldots, a_n\}$, where: *(a)* $\forall i \in \{1, 2, \ldots, n\}$, a_i has a known expected return, $r_i \geq 0$; *(b)* $\forall i, j \in \{1, 2, \ldots, n\}$, the pair (a_i, a_j) has a known expected covariance, $\sigma_{ij} = \sigma_{ji} \geq 0$; *(c)* a user-provided value, $R > 0$, represents the minimum expected return from the investment (expected return constraint); *(d)* a portfolio is a vector $X = (x_1, x_2, \ldots, x_n)$ such that each x_i represents the fraction of the total wealth invested in asset a_i, i.e., $0 \leq x_i \leq 1$ and $\sum_{i=1}^{n} x_i = 1$; *(e)* $\forall i \in \{1, 2, \ldots, n\}$, $z_i = 1$ if $x_i > 0$ (i.e., a_i in portfolio) and $z_i = 0$ otherwise; *(f)* the number of assets in the portfolio, $\sum_{i=1}^{n} z_i$, is bounded by user-defined values, k_{min} and k_{max} (cardinality constraints); *(g)* the user can pre-select certain assets to be included in the portfolio, i.e.: $\forall i \in \{1, 2, \ldots, n\}$, $p_i = 1$ if a_i is pre-assigned (i.e., $x_i > 0$) and $p_i = 0$ otherwise (pre-assignment constraints); and *(h)* for each asset a_i, its associated quantity in the portfolio, x_i, is bounded by user-defined values, ε_i and δ_i (quantity constraints). The typical objective of this problem is to select the optimal combination of fractions of each asset, x_i, so that the overall variance (risk) is minimized while satisfying all the aforementioned constraints.

3.2. The mathematical model

The mathematical model comprises an objective function and a set of constraints:

$$\min f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j \tag{1}$$

subject to:

$$\sum_{i=1}^{n} r_i x_i \geq R \tag{2}$$

$$\sum_{i=1}^{n} x_i = 1 \tag{3}$$

$$0 \leq x_i \leq 1, \forall i \in \{1, 2, \ldots, n\} \tag{4}$$

$$k_{min} \leq \sum_{i=1}^{n} z_i \leq k_{max} \tag{5}$$

$$\varepsilon_i z_i \leq x_i \leq \delta_i z_i, \forall i \in \{1, 2, ..., n\} \tag{6}$$

$$0 \leq \varepsilon_i \leq \delta_i \leq 1, \forall i \in \{1, 2, ..., n\} \tag{7}$$

$$p_i \leq z_i, \forall i \in \{1, 2, ..., n\} \tag{8}$$

$$z_i \leq M x_i, \forall i \in \{1, 2, ..., n\} \tag{9}$$

$$z_i \in \{0, 1\}, \forall i \in \{1, 2, ..., n\} \tag{10}$$

Equation (1) describes the investor's objective function. The investor's objective is to minimize the portfolio variance. Equations (1) – (4) outline the basic ("unconstrained") optimization problem and determine the UEF. Specifically, Equation (2) provides the lower bound for the investor's required return. Equation (3) ensures that portfolio weights add up to unity. The purpose of Equation (4) is to regulate leveraged positions. This equation is justified on the grounds of the existing short selling regulations in a wide range of countries [65]. A number of realistic situations often require additional constraints that can be summarized by means of Equations (5) – (10). By solving the constrained optimization problem given by Equations (1) – (10) the so-called constrained efficient frontier (CEF) is obtained. Equation (5) formulates cardinality constraints. Equation (6) defines quantity constraints. The quantity of each asset a_i is confined to a given range. A minimum quantity of wealth invested in asset a_i is given by ε_i. A maximum quantity of wealth invested in asset a_i is given by δ_i. Both parameters ε_i and δ_i range from 0 to 1 (Equation (7)). Note that Equation (4) is redundant when Equations (6) and (7) are considered. Equations (8) and (9) imply that certain assets are pre-assigned in the portfolio. In particular, given a vector of n binary decision variables Z (Equation (10)) (where z_i takes on value 1 if included in the portfolio and 0 otherwise), and a binary vector P of pre-assignments (in which p_i takes on value 1 if pre-assigned and 0 otherwise), whenever asset a_i is pre-assigned, it has to be included in the portfolio (Equation (8)). In Equation (9), M is a large positive value such that $Mx_i \geq 1$ for all $x_i > 0$. Thus, if the quantity in the portfolio of asset a_i, x_i, is equal to 0, it means that this asset is not included in the portfolio (i.e., $z_i = 0$). Only a few papers have considered the use of pre-assignment constraints. An example is [41], who found that the CEF obtained by pre-assigning a high-yield asset strictly dominates the CEF obtained by pre-assigning a low-yield asset. In general, they find an inverse relation between the average percentage loss (with regard to the UEF) and the return value. However, they do not report computing times for the CEF with pre-assignment constraints, which makes impossible to complete a fair comparison with their results.

3.3. Algorithm inputs and outputs

According to the problem description, the output of the algorithm will be an assets-investment plan (solution), $X = (x_1, x_2, \ldots, x_n)$, satisfying all the aforementioned constraints and with the lowest possible risk, $f(X)$. Similarly, the inputs of the algorithm are the following ones: *(a)* for each asset $a_i \in A$, the following values: r_i, ε_i, and δ_i; *(b)* the matrix of co-variances: $\{\sigma_{ij} / i \leq j\}$; *(c)* the user-defined minimum expected return: $R > 0$; and *(d)* the boundaries on the number of assets to include: k_{min} and k_{max}. In order to perform a fair comparison with some previous works and existing benchmarks [32,39,40,58,59], in this paper we will not consider pre-selected assets. However, due to its flexibility, the ARPO algorithm could be adapted without too much effort to deal with this constraint too.

4. The ARPO metaheuristic

The ARPO matheuristic combines three main components: *(a)* an ILS framework [34]; *(b)* the use of a biased randomization process [66,67] that guides the generation of new 'promising' solutions (perturbation stage); and *(c)* the use of a quadratic programming solver that, given a current portfolio, optimizes the levels of investment of each asset (local search). ILS is a conceptually simple yet powerful metaheuristic that has proven to be very efficient in solving complex combinatorial optimization problems. The underlying idea behind ILS is to narrow the search for candidate local optimal solutions returned by some embedded algorithm, typically a local search heuristic. [68] show that ILS obtains the best average performance among a set of selected metaheuristic approaches in three classical combinatorial optimization problems: bin packing, permutation flow shop, and personnel scheduling. The authors also emphasize two main factors for its success: *(i)* an excellent balance between exploration and exploitation by "systematically combining a perturbation followed by local search"; and *(ii)* its relative simplicity and the reduced number of parameters required, factors that facilitate its quick implementation in practical applications.

Pseudo-code 1 shows the main procedure of the ARPO algorithm. Apart from the inputs defining the instance, also the maximum computing time allowed, $maxTime$, and an additional parameter, $beta$, are passed to the procedure –the use of this additional parameter will be discussed later.

Algorithm 1 Main procedure of the ARPO algorithm (ILS framework).

```
    procedure ARPO(inputs, minReturn, maxTime, beta)
 1: initSol ← genInitSol(inputs)                         ▷ generate sol with highest possible return rate
 2: if {getReturn(initSol) < minReturn} then
 3:     return unfeasible                                 ▷ unfeasible problem
 4: end if
 5: genFriendshipLists(inputs)                           ▷ generate a sorted list of "friends" for each asset
 6: baseSol ← QPOptimize(initSol, minReturn)             ▷ optimize levels for each asset in portf.
 7: baseSol ← cleanSol(baseSol)                          ▷ delete from portf. assets with level = 0
 8: bestSol ← baseSol                                    ▷ initialize bestSol
 9: elapsedTime ← 0
10: credit ← 0                                           ▷ used in the acceptance criterion
11: while {elapsedTime < maxTime} do                     ▷ iterated local search
12:     newSol ← perturbateSol(baseSol, inputs, beta)    ▷ destruction-construction stages
13:     if {getMaxReturnAsset(newSol) < minReturn} then  ▷ fix solution if unfeasible
14:         newSol ← repairSol(newSol, inputs)
15:     end if
16:     if {newSol is in cache} then                     ▷ already optimized levels
17:         newSol ← loadFromCache(newSol)               ▷ use optimized levels saved in cache
18:     else                  ▷ apply a local search based on quadratic programming optimization
19:         newSol ← QPOptimize(newSol, minReturn)       ▷ optimize levels f.e. asset in portf.
20:         newSol ← cleanSol(newSol)                    ▷ delete from portf. assets with level = 0
21:         saveInCache(newSol)
22:     end if
23:     delta ← getRisk(newSol) - getRisk(baseSol)       ▷ newSol improves baseSol
24:     if {delta < 0} then
25:         credit ← -delta
26:         baseSol ← newSol
27:         if {getRisk(newSol) < getRisk(bestSol)} then ▷ newSol improves bestSol
28:             bestSol ← newSol
29:         end if
30:     else{delta > 0 and delta ≤ credit}               ▷ acceptance criterion
31:         credit ← 0
32:         baseSol ← newSol
33:     end if
34:     update elapsedTime
35: end while
36: return bestSol
    end procedure
```

The ARPO procedure starts by generating a 'dummy' initial solution (line 01). This initial solution is constructed by including the k_{min} assets with the highest return levels. The weights are determined by maximizing the expected return without violating the quantity constraints. This way, if the expected return provided by this solution does not reach the minimum return threshold imposed by the investor, then the problem will be infeasible since no other solution will do it (lines 02-04). Notice, however, that it is also likely to obtain a high risk associated with this initial solution –hence the name dummy.

At this point, a quick local search (lines 06-07) is applied to this initial solution in order to improve it without losing its feasibility. This local search uses quadratic programming in order to optimize the investment level assigned to each asset (i.e., $x_i \ \forall i \in \{1, ..., n\}$) in the current portfolio (solution), given a sub-set of assets ($z_i \ \forall i \in \{1, ..., n\}$). The improved solution will be considered both as the current 'base' solution and the 'best-so-far' solution (line 08). Now, the ARPO procedure resumes by starting an iterative improvement process (lines 11-35). As in most ILS frameworks, this process comprises three stages: *(a)* the *perturbation* stage (lines 12-15), which applies strong changes to the current base solution in order to increase exploration of the space of solutions; *(b)* the *local search* stage (lines 16-22), which tries to perform a quick improvement of the current base solution by applying some operators –in our case, it is based on the combined use of quadratic programming and a cache memory; and *(c)* the *acceptation* stage (lines 23-33), which makes use of a credit-based system in order to allow accepting, under certain restrictive conditions, a new base solution even when it offers a slightly higher risk than the current base solution –this 'degradation' of the base solution is allowed in order to reduce the probabilities of getting trapped in a local minimum during the searching process.

As regards as the *perturbation* stage (Pseudo-code 2), this follows a destruction - reconstruction process. First, this process takes as an input the current base solution. Second, the current base solution is partially destroyed according to some random criterion –in our case, a randomly selected number of assets are deleted from the portfolio– (lines 01-09). Third, the destroyed solution is re-constructed (completed) by adding new assets to the portfolio (lines 10-19).

Algorithm 2 Perturbation procedure to generate new 'promising' solutions.

```
    procedure perturbateSol(baseSol, inputs, beta)
 1: newSol ← copySol(baseSol)
                    ▷ 1. Remove a random number of randomly selected assets (destruction stage)
 2: nAssetsInSol ← getNAssetsInSol(newSol)
 3: if {nAssetsInSol > 1} then
 4:     nAssetsToRemove ← genRandomNumber(1, nAssetsInSol - 1)
 5:     for {i = 1 to nAssetsToRemove} do
 6:         asset ← selectRandomAsset(newSol)
 7:         newSol ← removeAsset(asset, newSol)
 8:     end for
 9: end if
                    ▷ 2. Randomly select one asset in current portf. to add several of its "friends"
10: asset ← getRandomAsset(newSol)
                    ▷ 3. Use biased rand. to add friendly assets until reaching kMax (re-construction stage)
11: while {size(newSol) < getKMax(inputs)} do
12:     listOfFriendlyAssets ← getFriendlyList(asset)          ▷ Sorted list of friendly assets
13:     do                  ▷ Randomly select a position using a Geometric(beta) prob. distribution
14:         position ← biasedRandom(size(listOfFriendlyAssets), beta)
15:         newAsset ← getAsset(listOfFriendlyAssets, position)
16:     while {newAsset in newSol}                ▷ Repeat until newAsset not in current portf.
17:         newSol ← addAsset(newAsset, newSol)
18:         asset ← newAsset
19: end while
20: return newSol
    end procedure
```

During this re-construction process, the selection of each new asset added to the portfolio is done following a 'friendship' criterion, i.e.: although the selection of the new asset is random, this new asset will be most likely selected among those assets that are highly compatible –i.e., showing a low covariance value– with the last asset added to the portfolio. This special behavior is attained throughout the use of a biased randomization selection process (line 14), which makes use of a geometric distribution of parameter beta $(0 < beta < 1)$. More details on biased randomization processes can be found in [66,67].

Finally, there might be times in which the newly generated solution does not fulfill the minimum return requirement. In those cases, a 'repair' stage is used to swap a randomly selected asset in the current portfolio (lines 01 – 04) by a high-return asset not currently in the portfolio (lines 05 – 10) (Pseudo-code 3).

As iterated local search and other similar approaches are stochastic in nature, it is not possible to give a sharp worst case complexity estimate for ARPO. However, experimentally, ILS and other similar approaches are found to perform well [69]. In our case, the time spent is bounded by the predefined time limit *maxTime*. Within the **while** loop, the search examines an ARPO perturbation, which is $\Theta(n^3)$. With an efficient random number generator, the other steps are constant or $O(n)$ time. Thus, the total complexity is $\Theta(n^3)$ per iteration.

Algorithm 3 Repair procedure to make newly generated solutions feasible.

```
    procedure repairSolution(sol, inputs)
 1: unusedAssets ← getAssetsNotInSol(sol, inputs)          ▷ Consider assets not in portf.
 2: unusedAssets ← shuffle(unusedAssets)          ▷ Random sorting of the unused assets list
 3: assetA ← getRandomAsset(sol)                      ▷ Select a random assetA in current portf.
 4: sol ← deleteAsset(assetA, sol)                        ▷ Delete assetA from current portf.
 5: for {each asset assetB in unusedAssets} do          ▷ Search unused assetB with high return
 6:     if {getReturn(assetB) ≥ minReturn} then
 7:         sol ← addAsset(assetB, sol)                       ▷ Add assetB to current port.
 8:         return sol
 9:     end if
10: end for
11: end procedure
```

5. Numerical experiments

The ARPO algorithm has been implemented as a Java application. Being an interpreted language, Java-based programs do not execute as fast as other compiled programs, such as those developed in C or C++. Nevertheless, Java permits a rapid, platform-independent, development of object-oriented prototypes that can be used to test the potential of an algorithm. Also, using Java allowed to integrate our code with ojAlgo (http://ojalgo.org), an open-source quadratic programming solver developed in Java. A standard personal computer, Intel Core i5 CPU at 3.2 GHz and 4 GB RAM with Linux Ubuntu, was used to perform all tests.

In this research, we experiment with stock market data already used in previous studies. It was retrieved from the repository ORlib, and it can be downloaded from the following website: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html. These instances were proposed by [32] and were studied by [39–41,58,59]. The data set comprises constituents of five stock market indices, Hang Seng (Hong Kong), DAX 100 (Germany), FTSE 100 (United Kingdom), S&P 100 (United States) and NIKKEI 225 (Japan). These indices were extracted from DataStream and are measured at weekly frequency spanning the period from March 1992 to September 1997.

Following [41] we divided the portfolio frontier into 100 equidistant points on the vertical axis that represents the user-defined rate of expected portfolio return. Although the algorithm has been designed for the constrained case, it is initially tested on the unconstrained mean-variance optimization problem. The test results show that our solver is able to return solutions that are overlapping with the unconstrained efficient frontier (UEF) published at the OR Library, which contributes to validate the effectiveness of our approach.

Next, we execute the algorithm on a constrained mean-variance frontier (the algorithm is executed 30 times and both the best and average results are recorded). The maximum time of execution for each instance is 20 seconds. The benchmark constraints are those imposed by the previous authors.

Essentially, the constraints involve the following conditions: $\varepsilon_i = 0.01$, $\delta_i = 1$, $k_{min} = 1$, $k_{max} = 10$, $\forall i \in \{1, 2, \ldots, n\}$. As in the aforementioned studies, pre-assignment constraints are not considered in these experiments, i.e., $p_i = 0$, $\forall i \in \{1, 2, \ldots, n\}$. Notice that, despite other authors claim that their approaches can solve the constrained problem with all the aforementioned constraints, this fact is not clearly showed neither in the description of their methods nor in the benchmarks they solve, since the parameter values they use in their benchmarks do not seem to impose a real challenge for their algorithms in terms of tight constraints.

6. Discussion of results

6.1. CEF-ARPO with certain inputs

We evaluate the quality of our ARPO algorithm by means of the so called *average percentage loss* (APL). APL can be defined as follows. Let R_h be the minimum required rate of return, and $V_{CEF}(R_h)$ and $V_{UEF}(R_h)$ ($\forall h \in \{1, 2, \ldots, H\}$, where h denotes a point of the frontier, and H denotes the number of points on the frontier) the values of $f(x)$ evaluated in the solutions to the constrained and unconstrained optimization problem, henceforth CEF and UEF, respectively. The latter is provided in ORlib, whereas the former is returned by ARPO. Then, $APL = \frac{100}{H} \sum_{h=1}^{H} \frac{V_{CEF}(R_h) - V_{UEF}(R_h)}{V_{UEF}(R_h)}$. Table 1 shows the values of APL and associated computational times. Notice that, in terms of the minimum APL, our ARPO algorithm outperforms on Instances 2 – 5 the hybrid solvers proposed by [41], which comprise combinations of first descent and steepest descent with quadratic programming (FD+QP and SD+QP, respectively). With regard to the first instance, our APL is greater, but this result may emanate from rounding errors. In terms of computational time, ARPO shows a superior performance relative to that of the solver's SD+QP and is comparable or better than the solver's FP+QP performance. We next contrast our results with the results reported by [39] and [40]. Although the minimum APL provided by ARPO is slightly superior to the hybrid solver combining a genetic algorithm (GA) and quadratic programming (QP) in [40], on the remaining instances the minimum APL accomplished by ARPO is lower. Furthermore, our computational times are considerably lower than those reported by the tabu search (TS) in [39], and by GA+QP in [40].

The UEF (as provided in the ORlib) and CEF (as provided by ARPO) for the five stock market indices are compared in Panels A – E of Figure 1.

Panel A of Figure 1 depicts the CEF for the Hang Seng (Hong Kong) stock market. A visual inspection suggests that for the Hang Seng stock index the CEF is hardly distinguishable from the UEF. However, as the rate of expected return increases, along with increasing risk of investment, the CEF tends to diverge relatively less from the UEF. In particular, at the higher end of the CEF that features rewarding but risky portfolios, the expected rate of return can be attained with fewer assets.

Panel B of Figure 1 depicts the CEF for the DAX 100 (Germany) stock market. Visual inspection indicates that for the DAX 100 stock index the CEF diverges from the UEF at the lower end of expected return, more specifically, for $R < 0.006$. As the rate of expected return increases, the CEF becomes indistinguishable from the UEF. Notably, portfolios with an expected return at the lower end of the CEF tend to be riskier (i.e., with higher portfolio variance) than portfolios on the UEF.

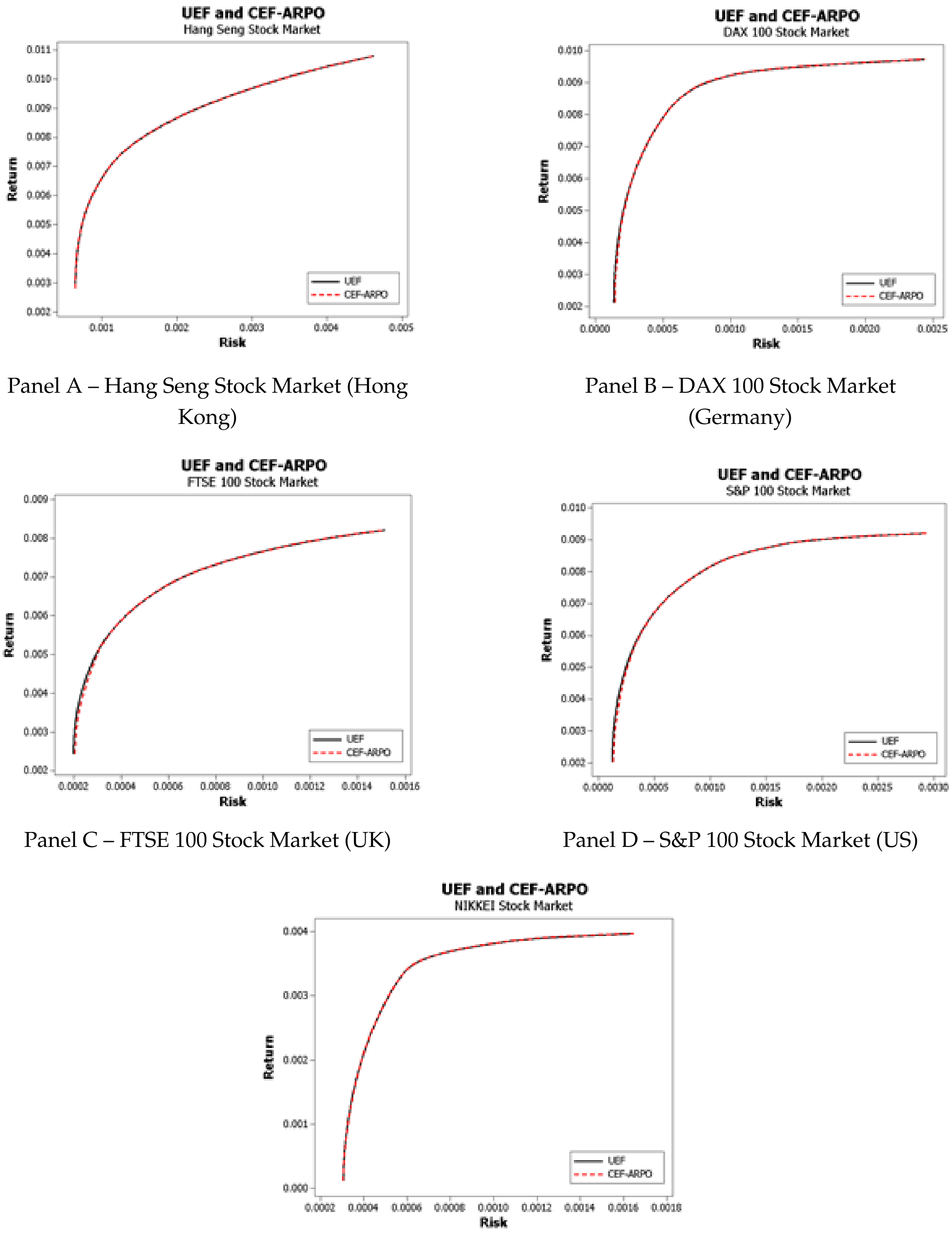
Panel C of Figure 1 depicts the CEF for the FTSE 100 (United Kingdom) stock market. It indicates that for the FTSE 100 stock index –similarly to the DAX 100 stock index– the CEF departs from the UEF at the lower end of expected return. As the rate of expected return increases, the CEF converges to the UEF. Noteworthy, portfolios featuring an expected return at the lower end of the CEF tend to be riskier (i.e., with higher portfolio variance) than portfolios on the UEF. At the higher end of the CEF that features rewarding but risky portfolios, the expected rate of return can be achieved with fewer assets.

Panel D of Figure 1 depicts the CEF for the S&P 100 (United States) stock market. It indicates that for the S&P 100 stock index –as with the DAX 100 and FTSE 100 stock indices– the CEF departs from the UEF at the lower end of expected return. As the rate of expected return increases, the CEF becomes visually indistinguishable from the UEF. Portfolio investments with an expected return at the lower

Table 1. Summary of Results

Instance	Di Gaspero et al. [41]				Moral-Escudero et al. [40]		Schaerf [39]		ARPO	
	FD + QP		SD + QP		GA + QP		TS		LS + QP	
	APL	T(s)	APL	T(s)	APL	T(s)	APL	T(s)	APL	T(Std) (s)
HS	0.00366	1.5	0.00321	3.1	0.00321	415.1	0.00409	251	0.00399	2.0 (0.87)
DAX 100	2.66104	9.6	2.53139	14.1	2.53180	552.7	2.53617	531	2.45403	1.0 (1.0)
FTSE 100	2.00146	10.1	1.92146	16.1	1.92150	886.3	1.92597	583	1.88340	13.7 (9.28)
S&P 100	4.77157	11.2	4.69371	18.8	4.69507	1163.7	4.69507	713	4.65095	15.5 (8.42)
NIKKEI 225	0.24176	25.3	0.20219	45.9	0.20198	1465.8	0.20198	1603	0.20189	5.0 (17.20)

Table 1 summarizes the results of ARPO for the CEF, including the minimum average percentage loss (APL) and computational time in seconds (T(s)). APL is calculated as in [41]. We also report the standard deviation of computational time for ARPO. The performance indicators of ARPO are also compared against its competitors, including [39–41].



Panel A – Hang Seng Stock Market (Hong Kong)

Panel B – DAX 100 Stock Market (Germany)

Panel C – FTSE 100 Stock Market (UK)

Panel D – S&P 100 Stock Market (US)

Panel E – NIKKEI Stock Market (Japan)

Figure 1. UEF and CEF-ARPO

Notes: Figure 1 illustrates the UEF and the CEF-ARPO for the five instances (Panel A – Hang Seng, Panel B – DAX 100, Panel C – FTSE 100, Panel D – S&P 100 and Panel E – NIKKEI 225). The UEF is represented with a black solid line, whereas the CEF-ARPO is represented with a dash red line. Portfolio variance is depicted on the horizontal axis, whereas the required rate of return is depicted on the vertical axis. The minimum average percentage loss is 0.00399% for the Hang Seng portfolio, 2.45403% for the DAX 100 portfolio, 1.88340% for the FTSE100 portfolio, 4.65095% for the S&P 100 portfolio and 0.20189% for the NIKKEI 225 portfolio.

Table 2. Hang Seng Stock Market (Hong Kong).

Curve Position	Required Return	UEF-Variance	ARPO-Variance	APL	Time (s)
20	0.002861137	0.0006424068	0.0006424114	0.0007160572	4.212
40	0.002941981	0.0006428092	0.0006429074	0.0152766949	2.74
60	0.003022827	0.0006434196	0.0006437456	0.0506667811	1.978
80	0.003103671	0.0006442382	0.0006443922	0.0239042019	3.759
100	0.003184516	0.0006452648	0.0006454721	0.0321263456	1.226
120	0.003265361	0.0006464996	0.0006467783	0.0431090754	1.917
140	0.003346206	0.0006479424	0.0006483109	0.0568723393	2.012
160	0.003427051	0.0006495933	0.0006499731	0.0584673518	2.686
180	0.003507896	0.0006514524	0.0006516646	0.0325733699	1.491
200	0.003588740	0.0006535208	0.0006536148	0.0143836279	4.464
1820	0.010137479	0.0035773525	0.0035773526	0.0000027954	0.004
1840	0.010218315	0.0036907539	0.0036907539	0.0000000000	0.009
1860	0.010299151	0.0038090873	0.0038090873	0.0000000000	0.001
1880	0.010379986	0.0039323522	0.0039323522	0.0000000000	0.009
1900	0.010460822	0.0040605480	0.0040605480	0.0000000000	0.001
1920	0.010541657	0.0041936758	0.0041936758	0.0000000000	0.003
1940	0.010622493	0.0043317350	0.0043317350	0.0000000000	0.010
1960	0.010703329	0.0044747255	0.0044747255	0.0000000000	0.001
1980	0.010784164	0.0046226475	0.0046226476	0.0000021633	0.003
2000	0.010865000	0.0047755010	0.0047755010	0.0000000000	0

Notes: Table 2 summarizes the UEF and the CEF-ARPO for the Hang Seng portfolio for 10 initial points and 10 last values of returns. In column 2, values of the required rate of return are provided. In column 3, values of the UEF solution (portfolio variance) are provided. In column 4, values of the CEF-ARPO solution (portfolio variance) are provided. In column 5, values of the minimum average percentage loss are reported. In column 6, computational times are reported.

Table 3. DAX 100 Stock Market (Germany).

Curve Position	Required Return	UEF-Variance	ARPO-Variance	APL	Time (s)
20	0.002175078	0.0001368925	0.0001481318	0.0821031101	1.5320
40	0.002252039	0.0001370119	0.0001483250	0.0825702001	0.2180
60	0.002329001	0.0001372175	0.0001484472	0.0818386868	0.3910
80	0.002405963	0.0001375210	0.0001485474	0.0801797544	0.8330
100	0.002482925	0.0001379123	0.0001486181	0.0776275938	1.8510
120	0.002559887	0.0001383842	0.0001487104	0.0746197904	0.8630
140	0.002636849	0.0001389376	0.0001490367	0.0726880269	0.5330
160	0.00271381	0.0001395866	0.0001495985	0.0717253662	0.5620
180	0.002790771	0.0001403353	0.0001503958	0.0716890191	0.9450
200	0.002867732	0.0001411837	0.0001514287	0.0725650341	1.2600
1820	0.009101412	0.0008965075	0.0008965075	0.0000000000	0.0000
1840	0.009178374	0.0009614987	0.0009614987	0.0000000000	0.0000
1860	0.009255336	0.0010349696	0.0010351065	0.0001322744	0.0000
1880	0.009332288	0.0011354764	0.0011354764	0.0000000000	0.0000
1900	0.009409241	0.0012881113	0.0012881113	0.0000000000	0.0000
1920	0.009486192	0.0014930083	0.0014930083	0.0000000000	0.0000
1940	0.009563145	0.0017501725	0.0017501725	0.0000000000	0.0000
1960	0.009640096	0.0020595971	0.0020595971	0.0000000000	0.0000
1980	0.009717049	0.0024212903	0.0024212904	0.0000000413	0.0000
2000	0.009794000	0.0028352430	0.0028352430	0.0000000000	0.0000

Notes: Table 3 summarizes the UEF and the CEF-ARPO for the DAX 100 portfolio for 10 initial points and 10 last values of returns. In column 2, values of the required rate of return are provided. In column 3, values of the UEF solution (portfolio variance) are provided. In column 4, values of the CEF-ARPO solution (portfolio variance) are provided. In column 5, values of the minimum average percentage loss are reported. In column 6, computational times are reported.

Table 4. FTSE 100 Stock Market (United Kingdom).

Curve Position	Required Return	UEF-Variance	ARPO-Variance	APL	Time (s)
20	0.002420865	0.0001985238	0.0002060320	0.037820	24.1290
40	0.002479328	0.0001986154	0.0002061982	0.038178	4.7360
60	0.002537792	0.0001987642	0.0002065801	0.039322	21.8640
80	0.002596256	0.0001989714	0.0002066068	0.038374	16.4550
100	0.00265472	0.0001992442	0.0002068282	0.038064	25.5930
120	0.002713184	0.0001995842	0.0002073013	0.038666	13.9090
140	0.002771647	0.0001999959	0.0002080261	0.040152	26.5100
160	0.002830111	0.0002004890	0.0002084997	0.039956	18.1160
180	0.002888575	0.0002010665	0.0002089129	0.039024	15.0670
200	0.002947039	0.0002017309	0.0002095566	0.038793	17.4660
1820	0.007682888	0.0010170776	0.0010170776	0.000000	1.3320
1840	0.007741359	0.0010578347	0.0010581058	0.000256	0.5650
1860	0.00779983	0.0011002537	0.0011003206	0.000061	11.5260
1880	0.007858284	0.0011455465	0.0011455466	0.000000	13.3580
1900	0.007916738	0.0011954685	0.0011954685	0.000000	3.9120
1920	0.007975191	0.0012500871	0.0012500871	0.000000	28.7840
1940	0.008033645	0.0013094021	0.0013094021	0.000000	27.6720
1960	0.008092098	0.0013734126	0.0013734126	0.000000	26.4330
1980	0.008150551	0.0014421206	0.0014423115	0.000132	19.7400
2000	0.008209000	0.0015166351	0.0015166351	0.000000	0.0000

Notes: Table 4 summarizes the UEF and the CEF-ARPO for the FTSE 100 portfolio for 10 initial points and 10 last values of returns. In column 2, values of the required rate of return are provided. In column 3, values of the UEF solution (portfolio variance) are provided. In column 4, values of the CEF-ARPO solution (portfolio variance) are provided. In column 5, values of the minimum average percentage loss are reported. In column 6, computational times are reported.

Table 5. S&P 100 Stock Market (United States).

Curve Position	Required Return	UEF-Variance	ARPO-Variance	APL	Time (s)
20	0.002005874	0.0001214699	0.000134226	0.105014	19.1900
40	0.002078497	0.0001216461	0.000134619	0.106648	26.0990
60	0.002151121	0.0001219398	0.000135389	0.110297	20.5970
80	0.002223742	0.0001223689	0.000136242	0.113373	17.2350
100	0.002296365	0.0001229290	0.000137073	0.115057	22.1810
120	0.002368987	0.0001236105	0.000138075	0.117013	23.3780
140	0.00244161	0.0001244126	0.000139442	0.120799	22.3860
160	0.002514232	0.0001253355	0.000140429	0.120421	19.7720
180	0.002586853	0.0001263852	0.000141311	0.118098	26.2040
200	0.002659475	0.0001275649	0.000142711	0.118735	16.2610
1820	0.008541599	0.0012695539	0.001269554	0.000000	0.6500
1840	0.008614195	0.0013438638	0.001343864	0.000000	0.3990
1860	0.008686789	0.0014260901	0.00142609	0.000000	2.2590
1880	0.008759385	0.0015162347	0.001516235	0.000000	0.1940
1900	0.008831981	0.0016142967	0.001614297	0.000000	14.5950
1920	0.008904579	0.0017205530	0.001720553	0.000000	0.1340
1940	0.008977215	0.0018772540	0.001877254	0.000000	19.9680
1960	0.009049852	0.0021147413	0.002114741	0.000000	22.3300
1980	0.009122459	0.0024387539	0.002438754	0.000000	0.0070
2000	0.009195000	0.0029387241	0.002938724	0.000000	0.0000

Notes: Table 5 summarizes the UEF and the CEF-ARPO for the S&P 100 portfolio for 10 initial points and 10 last values of returns. In column 2, values of the required rate of return are provided. In column 3, values of the UEF solution (portfolio variance) are provided. In column 4, values of the CEF-ARPO solution (portfolio variance) are provided. In column 5, values of the minimum average percentage loss are reported. In column 6, computational times are reported.

Table 6. NIKKEI Stock Market (Japan).

Curve Position	Required Return	UEF-Variance	ARPO-Variance	APL	Time (s)
20	0.0001078963	0.0003046821	0.0003048207	0.000455	0.0340
40	0.0001469218	0.0003048095	0.0003049299	0.000395	0.0820
60	0.0001859471	0.0003050116	0.0003051331	0.000398	0.0850
80	0.0002249721	0.0003052881	0.0003054303	0.000466	0.0660
100	0.0002639974	0.0003056390	0.0003058216	0.000597	0.1000
120	0.0003030223	0.0003060641	0.0003063069	0.000793	0.1930
140	0.0003420475	0.0003065635	0.0003068863	0.001053	0.1930
160	0.0003810730	0.0003071384	0.0003075597	0.001372	0.0540
180	0.0004200985	0.0003077930	0.0003083271	0.001735	0.1150
200	0.0004591229	0.0003085244	0.0003091886	0.002153	0.0400
1820	0.0036202364	0.0007178707	0.0007178707	0.000000	0.0320
1840	0.0036592244	0.0007585282	0.0007585282	0.000000	0.0210
1860	0.0036982126	0.0008065920	0.0008065920	0.000000	0.0520
1880	0.0037372032	0.0008621079	0.0008621079	0.000000	60.6030
1900	0.0037762020	0.0009261274	0.0009261274	0.000000	0.0050
1920	0.0038152008	0.0009990847	0.0009990847	0.000000	0.0050
1940	0.0038541986	0.0010809831	0.0010809831	0.000000	85.8630
1960	0.0038931362	0.0011966840	0.0011966841	0.000000	0.0010
1980	0.0039320680	0.0013855561	0.0013855562	0.000000	0.0010
2000	0.0039710000	0.0016485224	0.0016485224	0.000000	0.0000

Notes: Table 6 summarizes the UEF and the CEF-ARPO for the NIKKEI 225 portfolio for 10 initial points and 10 last values of returns. In column 2, values of the required rate of return are provided. In column 3, values of the UEF solution (portfolio variance) are provided. In column 4, values of the CEF-ARPO solution (portfolio variance) are provided. In column 5, values of the minimum average percentage loss are reported. In column 6, computational times are reported.

end of the CEF involve relatively more risk than portfolios with the same expected return located on the UEF.

Finally, Panel E of Figure 1 depicts the CEF for the NIKKEI (United States) stock market. It indicates that for the NIKKEI stock index, the relation between the CEF and the UEF follows a pattern similar to the Hang Seng stock index. Specifically, although the CEF departs from the UEF at the lower end of expected return, the difference is visually very small. As the rate of expected return increases, the CEF gradually approaches the UEF. At the higher end of the CEF that includes portfolios with high expected return and high risk, the APL approaches to zero.

To evaluate differences between the UEF and the CEF-ARPO, we also provide the portfolio weights for Instance 3 (FTSE 100), where the required rate of return is 0.0041572635, which is an approximately central value within the overall of returns. This instance was executed twice with the same seed, where the maximum time of execution was 20 seconds. The UEF considered all assets with weights ranging from 0 to 1 inclusively. The CEF was constrained to the minimum of 1 and the maximum of 10 assets, with portfolio weights ranging from 0.01 to 1. The minimum values of the portfolio variance were 2.3872556507357437E-4 (the UEF) and 2.5098945345432527E-4 (CEF-ARPO). The percentage loss is 5.137%. The UEF selected 26 assets (the remaining assets were allocated zero weight), whereas the CEF portfolio selected 10 assets, the upper bound of the cardinality constraint. The 10 assets are the subset of assets selected in the UEF portfolio, where assets indexed with 61, 36 and 52 carry the largest weight in the CEF portfolio.

6.2. CEF-ARPO with uncertain inputs

Key to our research is a widely held assumption –in line with most of the existing literature –that the returns vector and the variance and covariance matrix are well-know and deterministic; i.e., they are known with certainty. Whilst this assumption is analytically and computationally convenient, its practical appeal may be limited. In order to account for this practical restraint, and to test if our

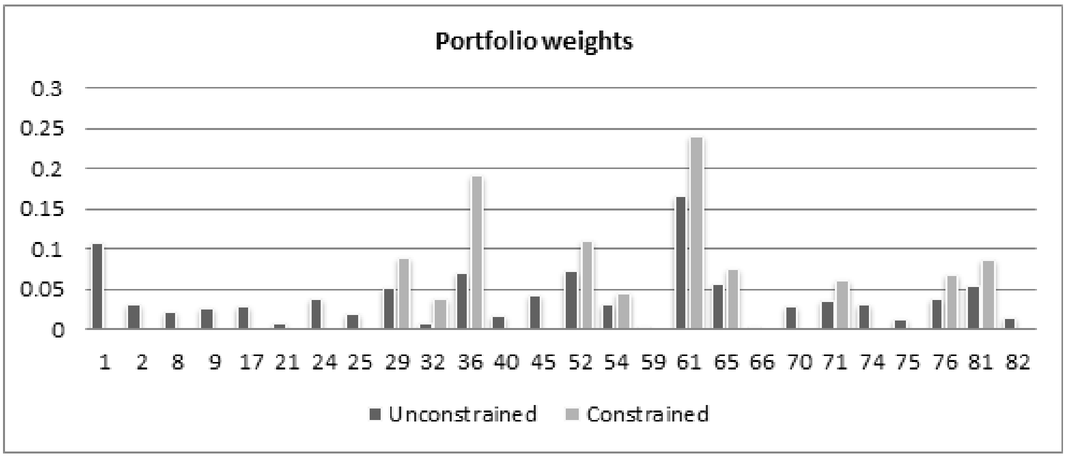


Figure 2. Portfolio Weights

Notes: The horizontal axis indexes assets, whereas the vertical axis measures the weight of an asset.

algorithm is robust to small changes in the returns vector and the variance-covariance matrix, we use the data set of Hang Seng stock market to perform computational experiments that involve some degree of uncertainty in the first and second moments. We consider three different scenarios. In Scenario 1 (S1), a random disturbance term distributed with a normal distribution with mean 0 and standard deviation 0.00015 is added to each return of the original instance. Under this scenario, an original return of 0.00350406 (which is the mean return of the instance) now takes on values within the interval of 0.00350406±2·0.00015=(0.00320406,0.00380406) with the probability of 95.45%, which can be deemed a realistic interval. In Scenario 2 (S2), a random disturbance term distributed with a normal distribution with mean 0 and standard deviation 0.000025 is added to each element of the covariance matrix. Under this scenario, the original covariance of 0.00113094 now takes on values within the interval of (0.00108094,0.00118094) with the probability of 95.45%. In Scenario 3 (S3), Scenarios 1 and 2 are jointly simulated. The results are compared against Scenario 0 (S0), which depicts the CEF–ARPO with $k_{min} = 1$, $k_{max} = 10$, and portfolio weights confined between 0.01 and 1.

Table 7 reports the portfolio variance and the APL loss relative to S0 of the three scenarios, for 10 initial and 10 last portfolio returns on the CEF-ARPO. The average APLs for the three scenarios are 3.324%, -0.109% and 3.274%, respectively (See also Table 10, where the average APL, average computational times and the number of infeasible rates of return are summarized). Thus our algorithm delivers a reasonable-quality solution also in an environment that allows for inaccurate inputs. However, these results should be interpreted with caution. Importantly, it should be recognized that S1 – S3 amount to different optimization problems. A relative larger APL for S1 may imply that it is more costly – in terms of the portfolio variance – to attain a given portfolio return when individual asset returns are randomized than when they are not. The induced random variation in individual asset returns effectively boosts the corresponding elements of the variance and covariance matrix. This leads to a larger overall portfolio variance for a given portfolio return. As a result, portfolios on the higher end of the efficient frontier become more costly and even infeasible (See also Table 10). In S2, where individual variances and covariances are randomized, the resulting CEF-ARPO remains very similar to the CEF-ARPO, as manifested by the average APL. Indeed, in contrast to S1, adding a normally distributed random disturbance term to variances and covariances between individual assets does not appear to alter the optimal solution. For portfolio on the lower end of the efficient frontier, the APL takes on low positive values; however, for more rewarding and riskier portfolios with feasible returns, the solution is better than for the CEF-ARPO, suggesting further diversification opportunities for investors in high-yield portfolios. Finally, S3 shows combined effects of S1 and S2. Next, we carry out a stability analysis of the ARPO algorithm to ratify our main findings. The stability analysis consists of six scenarios (numbered successively) as follows, which are evaluated against S0.

Table 7. Comparison of results for different scenarios based on Instance 1 (Hang Seng) varying the returns and the covariance matrix.

Curve Position	Required Return	ARPO – Variance S0	ARPO – Variance S1	APL S1 – S0	Time (s)	ARPO – Variance S2	APL S2 – S0	Time (s)	ARPO – Variance S3	APL S3 – S0	Time (s)
20	0.002861137	0.000642411	0.000642777	0.057%	0.848	0.000643991	0.246%	0.194	0.000644237	0.284%	1.991
40	0.002941981	0.000642907	0.000643518	0.095%	0.686	0.000644294	0.216%	0.133	0.000644828	0.299%	0.182
60	0.003022827	0.000643746	0.000644209	0.072%	0.227	0.000644951	0.187%	0.061	0.000645441	0.263%	0.105
80	0.003103671	0.000644392	0.000645228	0.130%	0.102	0.000645594	0.186%	0.324	0.000646286	0.294%	0.030
100	0.003184516	0.000645472	0.000646467	0.154%	0.189	0.00064646	0.153%	0.154	0.000647432	0.304%	0.072
120	0.003265361	0.000646778	0.000647925	0.177%	0.026	0.000647665	0.137%	1.397	0.000648805	0.313%	0.035
140	0.003346206	0.000648311	0.000649556	0.192%	0.111	0.000649104	0.122%	0.224	0.000650406	0.323%	0.140
160	0.003427051	0.000649973	0.000651174	0.185%	0.413	0.000650778	0.124%	0.025	0.000652234	0.348%	0.050
180	0.003507896	0.000651665	0.000653049	0.212%	0.071	0.000652687	0.157%	0.249	0.000654075	0.370%	0.143
200	0.00358874	0.000653615	0.000655182	0.240%	0.228	0.000654567	0.146%	0.043	0.000655994	0.364%	0.369
1820	0.010137479	0.003577353	0.003846626	7.527%	0.000	0.003553484	-0.667%	0.003	0.00382672	6.971%	0.006
1840	0.010218315	0.003690754	0.003973414	7.659%	0.003	0.003668539	-0.602%	0.003	0.003955413	7.171%	0.005
1860	0.010299151	0.003809087	0.00410529	7.776%	0.001	0.003788622	-0.537%	0.000	0.004089294	7.356%	0.000
1880	0.010379986	0.003932352	0.004242251	7.881%	0.000	0.003913731	-0.474%	0.002	0.004228359	7.527%	0.000
1900	0.010460822	0.004060548	0.004384302	7.973%	0.019	0.00404387	-0.411%	0.004	0.004372613	7.685%	0.003
1920	0.010541657	0.004193676	0.004531439	8.054%	0.008	0.004179034	-0.349%	0.001	0.004522052	7.830%	0.001
1940	0.010622493	0.004331735	0.004683665	8.124%	0.002	0.004319228	-0.289%	0.003	0.004676680	7.963%	0.000
1960	0.010703329	0.004474726	NF			0.00446445	-0.230%	0.000	NF		
1980	0.010784164	0.004622648	NF			0.004614697	-0.172%	0.004	NF		
2000	0.010865	0.004775501	NF			0.004769974	-0.116%	0.000	NF		

Notes: Table 7 summarizes the results of ARPO for different scenarios generated from Instance 1 (Hang Seng), in which the returns and the variance-covariance matrix are randomized. SX represents the scenario number X, where S0 is the benchmark scenario. NF indicates non-feasible portfolios.

- Scenario 4 (S4): the minimum number of assets in the portfolio is increased from 1 to 2.
- Scenario 5 (S5): the maximum number of assets in the portfolio is decreased from 10 to 9.
- Scenario 6 (S6): Scenarios S4 and S5 are jointly considered.
- Scenario 7 (S7): the minimum quantity for all assets is increased from 0.01 to 0.015.
- Scenario 8 (S8): the maximum quantity for all assets is decreased from 1 to 0.995
- Scenario 9 (S9): Scenarios S7 and S8 are jointly considered.

Observe that in Scenarios S4 – S9 the vector of asset returns and the variance and covariance matrix are calculated from historical data and, in contrast to Scenarios S1 – S3, do not involve any random inputs. Table 9 presents the calculated CEF-ARPO variances and APLs for Scenarios S4 – S6. The calculations show that increasing the lower bound of the cardinality constraint does not have any effect on APLs or computational times. The APL is always 0 irrespectively of the CEF-ARPO position, including the higher end of the CEF. Similarly, decreasing the number of assets from 10 to 9 under S5 poses no significant challenge to portfolio diversification, since the resulting APLs – mainly at the lower end of the CEF – are very low. Scenario S6 that combines the two preceding scenarios shows some insignificant departures from S0, essentially due to the lower maximum number of assets that can be included in a portfolio.

Finally, Scenarios S7 – S9 show that small variations in the minimum and maximum quantities are followed by similarly unimportant changes in the CEF. Indeed, an increase in the minimum weight attributed to each asset leads to the average APL of 0.013% (S7 and S9) relative to the optimal solution under S0 (See also Table 10). However, a decrease in the maximum weight does not appear to alter the optimal solution. All in all, additional computational experiments indicate that the ARPO algorithm shows a reasonable level of stability when dealing with constrained portfolio optimization problems.

7. Conclusions

In this paper we propose the ARPO algorithm to solve the constrained mean-variance optimization problem. The specific constraints we use include the cardinality and quantity constraints. The ensuing complexity of the problem rules out closed-form solutions and conventional optimization methods. Specifically, the cardinality and quantity constraints render the problem computationally expensive to solve large-scale instances. Therefore, the use of metaheuristics –that handle such problems more flexibly and efficiently– is needed. The ARPO algorithm first generates an initial feasible solution. Then, this solution is iteratively improved using an iterated local search process which combines quadratic programming with a cache of previously computed solutions. The quality of the best-found solution is evaluated using the average percentage loss relative to the unconstrained efficient frontier.

According to the computational experiments, our solver outperforms –in terms of average percentage loss– most of the recent state-of-art approaches used in the literature, which tend to be more complex and difficult to implement in practical applications. In addition, the running time for our solver is notably shorter than the solvers used in other recent studies.

8. Future work

In future research, tighter cardinality and quantity constraints should be considered, as well as the pre-assignment constraint. Mean-variance portfolio optimization provides a natural laboratory for testing the performance of ARPO. While this algorithm is tested on the five benchmark indices, an unexhausted list application could potentially include other periods, different countries, regions, sectors and asset classes. Moreover, the practical appeal of ARPO is not limited to portfolio optimization, but rather could be extended to other optimization problem comprising, for instance, asset and liability management of companies. Likewise, it can be conveniently applied to construct a portfolio of international investments and to study whether ARPO can alleviate the widely documented "home bias" phenomenon.

Investment portfolio formulation with variance as a risk measure is just one option. It is possible to extend presented research using multi-criteria investing portfolio models [17,23] with the use of

Table 8. Comparison of results for different scenarios based on Instance 1 (Hang Seng) varying problem cardinality constraints.

Curve Position	Required Return	ARPO – Variance S0	ARPO – Variance S4	APL S4 – S0	Time (s)	ARPO – Variance S5	APL S5 – S0	Time (s)	ARPO – Variance S6	APL S6 – S0	Time (s)
20	0.002861137	0.000642411	0.000642411	0.000%	1.446	0.00064252	0.017%	0.557	0.00064252	0.017%	0.254
40	0.002941981	0.000642907	0.000642907	0.000%	0.118	0.000643026	0.019%	0.368	0.000643026	0.019%	0.391
60	0.003022827	0.000643746	0.000643746	0.000%	0.127	0.000643875	0.020%	0.527	0.000643875	0.020%	0.256
80	0.003103671	0.000644392	0.000644392	0.000%	0.365	0.000645066	0.105%	0.185	0.000645066	0.105%	0.090
100	0.003184516	0.000645472	0.000645472	0.000%	0.338	0.0006466	0.175%	0.191	0.0006466	0.175%	0.285
120	0.003265361	0.000646778	0.000646778	0.000%	0.012	0.000647619	0.130%	0.803	0.000647619	0.130%	0.058
140	0.003346206	0.000648311	0.000648311	0.000%	0.101	0.000648891	0.089%	0.171	0.000648891	0.089%	0.138
160	0.003427051	0.000649973	0.000649973	0.000%	0.168	0.000650437	0.071%	0.123	0.000650437	0.071%	0.020
180	0.003507896	0.000651665	0.000651665	0.000%	0.589	0.000652259	0.091%	0.408	0.000652259	0.091%	0.131
200	0.00358874	0.000653615	0.000653615	0.000%	0.089	0.000654354	0.113%	0.141	0.000654354	0.113%	0.093
1820	0.010137479	0.003577353	0.003577352	0.000%	0.002	0.003577352	0.000%	0.000	0.003577352	0.000%	1.847
1840	0.010218315	0.003690754	0.003690754	0.000%	4.421	0.003690754	0.000%	0.002	0.003690754	0.000%	3.000
1860	0.010299151	0.003809087	0.003809088	0.000%	1.707	0.003809088	0.000%	0.002	0.003809088	0.000%	2.188
1880	0.010379986	0.003932352	0.003932352	0.000%	1.766	0.003932352	0.000%	0.001	0.003932352	0.000%	0.743
1900	0.010460822	0.004060548	0.004060549	0.000%	2.067	0.004060549	0.000%	0.002	0.004060549	0.000%	0.118
1920	0.010541657	0.004193676	0.004193675	0.000%	0.987	0.004193675	0.000%	0.048	0.004193675	0.000%	0.507
1940	0.010622493	0.004331735	0.004331735	0.000%	14.906	0.004331735	0.000%	0.001	0.004331735	0.000%	0.685
1960	0.010703329	0.004474726	0.004474726	0.000%	0.000	0.004474726	0.000%	0.000	0.004474726	0.000%	0.000
1980	0.010784164	0.004622648	0.004622647	0.000%	0.000	0.004622647	0.000%	0.001	0.004622647	0.000%	0.000
2000	0.010865	0.004775501	0.004775501	0.000%	0.000	0.004775501	0.000%	0.000	0.004775501	0.000%	0.000

Notes: Table 8 summarizes the results of ARPO for different scenarios generated from Instance 1 (Hang Seng), in which problem cardinality constrains are modified. SX represents the scenario number X, where S0 is the benchmark scenario.

Table 9. Comparison of results for different scenarios based on Instance 1 (Hang Seng) varying problem quantity constraints.

Curve Position	Required Return	ARPO – Variance S0	ARPO – Variance S7	APL S7 – S0	Time (s)	ARPO – Variance S8	APL S8 – S0	Time (s)	ARPO – Variance S9	APL S9 – S0	Time (s)
20	0.002861137	0.000642411	0.00064252	0.017%	0.953	0.000642411	0.000%	0.701	0.00064252	0.017%	0.174
40	0.002941981	0.000642907	0.000643026	0.019%	0.962	0.000642907	0.000%	0.387	0.000643026	0.019%	0.094
60	0.003022827	0.000643746	0.000643875	0.020%	0.087	0.000643746	0.000%	0.789	0.000643875	0.020%	0.229
80	0.003103671	0.000644392	0.000645066	0.105%	0.073	0.000644392	0.000%	0.144	0.000645066	0.105%	0.424
100	0.003184516	0.000645472	0.000646032	0.087%	1.309	0.000645472	0.000%	0.246	0.000646032	0.087%	0.044
120	0.003265361	0.000646778	0.000646778	0.000%	0.154	0.000646778	0.000%	0.236	0.000646778	0.000%	0.031
140	0.003346206	0.000648311	0.000648311	0.000%	0.083	0.000648311	0.000%	0.005	0.000648311	0.000%	0.104
160	0.003427051	0.000649973	0.00065007	0.015%	0.420	0.000649973	0.000%	0.642	0.00065007	0.015%	0.032
180	0.003507896	0.000651665	0.000651665	0.000%	0.681	0.000651665	0.000%	0.388	0.000651665	0.000%	0.148
200	0.00358874	0.000653615	0.000653615	0.000%	0.372	0.000653615	0.000%	0.282	0.000653615	0.000%	0.109
1820	0.010137479	0.003577353	0.003577352	0.000%	0.006	0.003577352	0.000%	0.000	0.003577352	0.000%	0.002
1840	0.010218315	0.003690754	0.003690754	0.000%	0.006	0.003690754	0.000%	0.002	0.003690754	0.000%	0.001
1860	0.010299151	0.003809087	0.003809088	0.000%	0.000	0.003809088	0.000%	0.001	0.003809088	0.000%	0.000
1880	0.010379986	0.003932352	0.003932352	0.000%	0.001	0.003932352	0.000%	0.000	0.003932352	0.000%	0.002
1900	0.010460822	0.004060548	0.004060549	0.000%	0.001	0.004060549	0.000%	0.011	0.004060549	0.000%	0.000
1920	0.010541657	0.004193676	0.004193675	0.000%	0.010	0.004193675	0.000%	0.002	0.004193675	0.000%	0.000
1940	0.010622493	0.004331735	0.004331735	0.000%	0.001	0.004331735	0.000%	0.000	0.004331735	0.000%	0.003
1960	0.010703329	0.004474726	0.004474726	0.000%	0.000	0.004474726	0.000%	0.000	0.004474726	0.000%	0.003
1980	0.010784164	0.004622648	0.004622647	0.000%	0.000	0.004622647	0.000%	0.001	0.004622647	0.000%	0.004
2000	0.010865	0.004775501	0.004775501	0.000%	0.000	0.004775501	0.000%	0.000	0.004775501	0.000%	0.000

Notes: Table 9 summarizes the results of ARPO for different scenarios generated from Instance 1 (Hang Seng), in which problem quantity constrains are modified. SX represents the scenario number X, where S0 is the benchmark scenario.

Table 10. Different scenarios based on Instance 1 (Hang Seng).

Scenario	Average APL	Non feasible portfolios	Time (s)
Scenario 1	3.324%	3	0.173
Scenario 2	-0.109%	0	0.141
Scenario 3	3.274%	3	0.196
Scenario 4	0.000%	0	1.460
Scenario 5	0.042%	0	0.177
Scenario 6	0.042%	0	0.540
Scenario 7	0.013%	0	0.256
Scenario 8	0.000%	0	0.192
Scenario 9	0.013%	0	0.070

Notes: Table 10 shows the average APL (column 2), the number of non feasible portfolios (column 3), and the average computational times (column 4) obtained for 9 different scenarios on Instance 1 (Hang Seng).

value-at-risk (VaR) and conditional value-at-risk (CVaR) or multi-objective supply portfolio models [70,71] with VaR and CVaR or with differently formulated risk measures [72]. In all these cases a novel contribution to this issue would consist in a comparison between exact methods and ARPO-like metaheuristics.

One important shortcoming in the literature of mean-variance portfolio optimization is that the models typically rely upon expected returns and variance data, instead of considering the return as a real random variable. A novel contribution to this issue would consist of adding stochastic elements to the problem formulation, so that it includes random effects. Related to this, it would be possible to consider hierarchical statistical inference to handle uncertainty and then to employ high-dimensional data mining algorithms such as the one proposed in [73].

Author Contributions: conceptualization, R.K., A.A.J. and B.S.; methodology, A.A.J.; software, A.A.J.; validation, R.K., A.A.J. and B.S.; formal analysis, R.K.; investigation, R.K., A.A.J., B.S. and L.C.; resources, R.K. and B.S.; data curation, A.A.J., R.K. and L.C.; writing–original draft preparation, R.K., A.A.J., B.S. and L.C.; writing–review and editing, R.K., A.A.J. and B.S.; visualization, L.C.; supervision, A.A.J.; project administration, R.K.

Funding: This work has been partly supported by a NCN research grant, #DEC-2013/11/B/ST8/04458.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Markowitz, H. Portfolio selection. *Journal of Finance* **1952**, *7*, 77–91.
2. Kolm, P.N.; Tütüncü, R.; Fabozzi, F.J. 60 Years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research* **2014**, *234*, 356–371.
3. Levy, M.; Ritov, Y. Mean–variance efficient portfolios with many assets: 50% short. *Quantitative Finance* **2011**, *11*, 1461–1471.
4. Leippold, M.; Trojani, F.; Vanini, P. Multiperiod mean-variance efficient portfolios with endogenous liabilities. *Quantitative Finance* **2011**, *11*, 1534–1546.
5. Vigna, E. On efficiency of mean–variance based portfolio selection in defined contribution pension schemes. *Quantitative Finance* **2014**, *14*, 237–258.
6. Fujii, M.; Takahashi, A. Making mean-variance hedging implementable in a partially observable market. *Quantitative Finance* **2014**, *14*, 1709–1724.
7. Zumbach, G. A mean/variance approach to long-term fixed-income portfolio allocation. *Quantitative Finance* **2013**, *13*, 1459–1471.
8. Maringer, D.G. *Portfolio management with heuristic optimization*; Springer, 2005.
9. Sawik, B. Survey of multi-objective portfolio optimization by linear and mixed integer programming. In *Applications of Management Science*; Lawrence, K.D.; Kleinman, G., Eds.; Emerald Group Publishing Limited, 2013; Vol. 16, pp. 55–79.

10. Bertsimas, D.; Pachamanova, D. Robust multiperiod portfolio management in the presence of transaction costs. *Computers and Operations Research* **2008**, *35*, 3–17.
11. Angelelli, E.; Mansini, R.; Speranza, M.G. A comparison of MAD and CVaR models with real features. *Journal of Banking Finance* **2007**, *32*, 1188–1197.
12. Perez, G.B.; Jones, D.F.; Tamiz, M.; Bilbao, T.A. An interactive three-stage model for mutual funds portfolio selection. *Omega. The International Journal of Management Science* **2007**, *35*, 75–88.
13. Sawik, B. A Three Stage Lexicographic Approach for Multi-Criteria Portfolio Optimization by Mixed Integer Programming. *Przeglad Elektrotechniczny* **2008**, *84*, 108–112.
14. Sharpe, W.F. A linear programming algorithm for mutual fund portfolio selection. *Management Science* **1967**, *13*, 499–510.
15. Sharpe, W.F. A linear programming approximation for the general portfolio analysis problem. *Journal of Financial and Quantitative Analysis* **1971**, *6*, 1263–1275.
16. Speranza, M.G. Linear programming models for portfolio optimization. *Finance: Revue de l'Association Française de Finance* **1993**, *14*, 107–123.
17. Sawik, B. Downside risk approach for multi-objective portfolio optimization. In *Operations Research Proceedings*; Klatte, D.; Luthi, H.; Schmedders, K., Eds.; Springer, 2012; pp. 191–196.
18. Mansini, R.; Ogryczak, W.; Speranza, M.G. Twenty years of linear programming based portfolio optimization. *European Journal of Operational Research* **2014**, *234*, 518–535.
19. Pae, Y.; Sabbaghi, N. Log-robust portfolio management after transaction costs. *OR Spectrum* **2014**, *36*, 95–112. doi:10.1007/s00291-013-0322-y.
20. Bruni, R.; Cesarone, F.; Scozzari, A.; Tardella, F. A linear risk-return model for enhanced indexation in portfolio optimization. *OR Spectrum* **2015**, *37*, 735–759. doi:10.1007/s00291-014-0383-6.
21. Wolfe, P. The simplex method for quadratic programming. *Econometrica* **1959**, *27*, 382–398.
22. Pachamanova, D.A.; Fabozzi, F.J. *Simulation and Optimization in Finance*; John Wiley and Sons, 2010.
23. Sawik, B. Bi-criteria portfolio optimization models with percentile and symmetric risk measures by mathematical programming. *Przeglad Elektrotechniczny* **2012**, *88*, 176–180.
24. Pınar, M.Ç. Robust scenario optimization based on downside-risk measure for multi-period portfolio selection. *OR Spectrum* **2007**, *29*, 295–309. doi:10.1007/s00291-005-0023-2.
25. Ólafsson, S. Chapter 21 Metaheuristics. In *Handbooks in Operations Research and Management Science 13*; Elsevier, 2006; pp. 633–654.
26. Jourdan, L.; Basseur, M.; Talbi, E.G. Hybridizing exact methods and metaheuristics: A taxonomy. *European Journal of Operational Research* **2009**, *199*, 620–629.
27. Blum, C.; Puchinger, J.; Raidl, G.R.; Roli, A. Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing* **2011**, *11*, 4135–4151.
28. Boussaïd, I.; Lepagnot, J.; Siarry, P. A survey on optimization metaheuristics. *Information Sciences* **2013**, *237*, 82–117.
29. Tobin, J. Liquidity preference as behavior towards risk. *Review of Economic Studies* **1958**, *25*, 65–86.
30. Tobin, J. The theory of portfolio selection. In *The theory of interest rates*; Hahn, F.; Brechling, F., Eds.; Macmillan & Co. Ltd., London, 1965; pp. 3–51.
31. Bienstock, D. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical programming* **1996**, *74*, 121–140.
32. Chang, T.J.; Meade, N.; Beasley, J.E.; Sharaiha, Y.M. Heuristics for cardinality constrained portfolio optimisation. *Computers and Operations Research* **2000**, *27*, 1271–1302.
33. Jobst, N.J.; Horniman, M.D.; Lucas, C.A.; Mitra, G. Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. *Quantitative finance* **2001**, *1*, 489–501.
34. Lourenço, H.; Martin, O.; Stützle, T. Iterated local search: framework and applications. In *Handbook of Metaheuristics*; Gendreau, M.; Potvin, J.Y., Eds.; Springer US, 2010; Vol. 146, *International Series in Operations Research & Management Science*, pp. 363–397.
35. Ferrer, A.; Guimarans, D.; Ramalhinho, H.; Juan, A.A. A BRILS metaheuristic for non-smooth flow-shop problems with failure-risk costs. *Expert Systems with Applications* **2016**, *44*, 177–186.
36. Grasas, A.; Juan, A.A.; Faulin, J.; de Armas, J.; Ramalhinho, H. Biased randomization of heuristics using skewed probability distributions: a survey and some applications. *Computers & Industrial Engineering* **2017**, *110*, 216–228.

37. Dominguez, O.; Guimarans, D.; Juan, A.A.; de la Nuez, I. A biased-randomised large neighbourhood search for the two-dimensional vehicle routing problem with backhauls. *European Journal of Operational Research* **2016**, *255*, 442–462.
38. Gonzalez-Neira, E.M.; Ferone, D.; Hatami, S.; Juan, A.A. A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times. *Simulation Modelling Practice and Theory* **2017**, *79*, 23–36.
39. Schaerf, A. Local search techniques for constrained portfolio selection problems. *Computational Economics* **2002**, *20*, 177–190.
40. Moral-Escudero, R.; Ruiz-Torrubiano, R.; Suárez, A. Selection of optimal investment portfolios with cardinality constraints. IEEE Congress on Evolutionary Computation. Press IEEE, 2006, pp. 2382–2388.
41. Di Gaspero, L.; Tollo, G.D.; Roli, A.; Schaerf, A. Hybrid metaheuristics for constrained portfolio selection problems. *Quantitative Finance* **2011**, *11*, 1473–1487.
42. Michaud, R.O. The Markowitz optimization enigma: is 'optimized' optimal? *Financial Analysts Journal* **1989**, *45*, 31–42.
43. Kallberg, J.G.; Ziemba, W.T. Mis-specifications in portfolio selection problems. In *Risk and Capital*; Springer-Verlag, New York, 1984; pp. 74–87.
44. DeMiguel, V.; Garlappi, L.; Uppal, R. Optimal versus naïve diversification: How inefficient is the 1/N portfolio strategy? *Review of Financial Studies* **2009**, *22*, 1915–1953.
45. Jorion, P. International portfolio diversification with estimation risk. *Journal of Business* **1985**, *58*, 259–278.
46. Maringer, D.G.; Kellerer, H. Optimization of cardinality constrained portfolios with a hybrid local search algorithm. *OR Spectrum* **2003**, *25*, 481–495.
47. Baule, R. Optimal portfolio selection for the small investor considering risk and transaction costs. *OR Spectrum* **2010**, *32*, 61–76. doi:10.1007/s00291-008-0152-5.
48. Blume, M.E.; Friend, I. The asset structure of individual portfolios and some implications for utility functions. *Journal of Finance* **1975**, *30*, 585–603.
49. Guiso, L.; Jappelli, T.; Terlizzese, D. Income risk, borrowing constraints, and portfolio choice. *The American Economic Review* **1996**, *86*, 158–172.
50. Jansen, R.; van Dijk, R. Optimal benchmark tracking with small portfolios. *Journal of Portfolio Management* **2002**, *28*, 9–22.
51. Evans, J.L.; Archer, S.H. Diversification and the reduction of dispersion: an empirical analysis. *Journal of Finance* **1968**, *23*, 761–767.
52. Lloyd, W.P.; Hand, J.H.; Modani, N.K. The effect of portfolio construction rules on the relationship between portfolio size and effective diversification. *Journal of Financial Research* **1981**, *4*, 183–193.
53. Nawrocki, D.N. Portfolio Optimization, Heuristics and the 'Butterfly Effect'. *Journal of Financial Planning-Denver* **2000**, *13*, 68–75.
54. Elton, E.J.; Gruber, M.J.; Padberg, M.W. Simple criteria for optimal portfolio selection. *Journal of Finance* **1976**, *31*, 1341–1357.
55. Nawrocki, D.N. A comparison of risk measures when used in a simple portfolio selection heuristic. *Journal of Business Finance Accounting* **1983**, *10*, 183–194.
56. Crama, Y.; Schyns, M. Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research* **2003**, *150*, 546–571.
57. Derigs, U.; Nickel, N.H. Meta-heuristic based decision support for portfolio optimization with a case study on tracking error minimization in passive portfolio management. *OR Spectrum* **2003**, *25*, 345–378.
58. Armañanzas, R.; Lozano, J.A. A multiobjective approach to the portfolio optimization problem. IEEE Congress on Evolutionary Computation. IEEE Press, 2005, Vol. 2, pp. 1388–1395.
59. Fernández, A.; Gómez, S. Portfolio selection using neural networks. *Computers and Operations Research* **2007**, *34*, 1177–1191.
60. Rundo, F.; Trenta, F.; di Stallo, A.L.; Battiato, S. Grid trading system robot (GTSbot): A novel mathematical algorithm for trading FX market. *Applied Sciences* **2019**, *9*, 1796.
61. Hu, Y.J.; Lin, S.J. Deep reinforcement learning for optimizing finance portfolio management. 2019 Amity International Conference on Artificial Intelligence (AICAI). IEEE, 2019, pp. 14–20.

62. Zhang, W.; Zhou, C. Deep learning algorithm to solve portfolio management with proportional transaction cost. 2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr). IEEE, 2019, pp. 1–10.
63. Calvet, L.; de Armas, J.; Masip, D.; Juan, A.A. Learnheuristics: Hybridizing metaheuristics with machine learning for optimization with dynamic inputs. *Open Mathematics* **2017**, *15*, 261–280.
64. Di Tollo, G.; Roli, A. Metaheuristics for the portfolio selection problem. *International Journal of Operations Research* **2008**, *5*, 13–35.
65. Jain, A.; Jain, P.K.; McInish, T.H.; McKenzie, M. Worldwide reach of short selling regulations. *Journal of Financial Economics* **2013**, *109*, 177–197.
66. Juan, A.A.; Faulin, J.; Ruiz, R.; Barrios, B.; Caballé, S. The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem. *Applied Soft Computing* **2010**, *10*, 215–224.
67. Juan, A.A.; Faulin, J.; Jorba, J.; Riera, D.; Masip, D.; Barrios, B. On the use of Monte Carlo simulation, cache and splitting techniques to improve the Clarke and Wright savings heuristics. *Journal of the Operational Research Society* **2011**, *62*, 1085–1097.
68. Burke, E.; Curtois, T.; Hyde, M.; Kendall, G.; Ochoa, G.; Petrovic, S.; Vázquez-Rodríguez, J.A.; Gendreau, M. Iterated local search vs. hyper-heuristics: Towards general-purpose search algorithms. IEEE Congress on Evolutionary Computation. IEEE Press, 2010, pp. 3073–3080.
69. Nouraniy, Y.; Andresen, B. A comparison of simulated annealing cooling strategies. *J. Phys. A: Math. Gen.* **1998**, *31*, 8373–8385.
70. Sawik, T. Selection of supply portfolio under disruption risks. *Omega. The International Journal of Management Science* **2011**, *39*, 194–208.
71. Sawik, T. Selection of resilient supply portfolio under disruption risks. *Omega. The International Journal of Management Science* **2013**, *41*, 259–269.
72. Heckmann, I.; Comes, T.; Nickel, S. A critical review on supply chain risk – Definition, measure and modeling. *Omega. The International Journal of Management Science* **2015**, *52*, 119–132.
73. Cai, X.; Rajasekaran, S.; Zhang, F. Efficient Approximate Algorithms for the Closest Pair Problem in High Dimensional Spaces. Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, 2018, pp. 151–163.