



The application of water cycle algorithm to portfolio selection

Mohammad Moradi, Ali Sadollah, Hoda Eskandar & Hadi Eskandar

To cite this article: Mohammad Moradi, Ali Sadollah, Hoda Eskandar & Hadi Eskandar (2017) The application of water cycle algorithm to portfolio selection, Economic Research-Ekonomiska Istraživanja, 30:1, 1277-1299, DOI: [10.1080/1331677X.2017.1355254](https://doi.org/10.1080/1331677X.2017.1355254)

To link to this article: <https://doi.org/10.1080/1331677X.2017.1355254>



© 2017 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 05 Sep 2017.



Submit your article to this journal [↗](#)



Article views: 261



View related articles [↗](#)



View Crossmark data [↗](#)

The application of water cycle algorithm to portfolio selection

Mohammad Moradi^a, Ali Sadollah^b , Hoda Eskandar^c and Hadi Eskandar^d

^aFaculty of Management, University of Tehran, Tehran, Iran; ^bYoung Researchers and Elites Club, Science and Research Branch, Islamic Azad University, Tehran, Iran; ^cDepartment of Accounting, Faculty of Economics and Accounting, Central Tehran Branch, Islamic Azad University, Tehran, Iran; ^dFaculty of Mechanical Engineering, University of Semnan, Semnan, Iran

ABSTRACT

Portfolio selection is one of the most vital financial problems in literature. The studied problem is a nonlinear multi-objective problem which has been solved by a variety of heuristic and metaheuristic techniques. In this article, a metaheuristic optimiser, the multi-objective water cycle algorithm (MOWCA), is represented to find efficient frontiers associated with the standard mean-variance (M-V) portfolio optimisation model. The inspired concept of WCA is based on the simulation of water cycle process in the nature. Computational results are obtained for analyses of daily data for the period January 2012 to December 2014, including S&P100 in the US, Hang Seng in Hong Kong, FTSE100 in the UK, and DAX100 in Germany. The performance of the MOWCA for solving portfolio optimisation problems has been evaluated in comparison with other multi-objective optimisers including the NSGA-II and multi-objective particle swarm optimisation (MOPSO). Four well-known performance metrics are used to compare the reported optimisers. Statistical optimisation results indicate that the applied MOWCA is an efficient and practical optimiser compared with the other methods for handling portfolio optimisation problems.

ARTICLE HISTORY

Received 18 October 2015
Accepted 28 October 2016

KEYWORDS

Portfolio optimisation; mean-variance (M-V) optimisation; multi-objective optimisations; multi-objective water cycle algorithm (MOWCA); non-dominated sorting Genetic Algorithm (GA)

JEL CODES

C61; G10; G11; G14; G61

1. Introduction

Portfolio management is one of the most important financial research areas for many scientists and researchers (Zhu et al., 2011). Lots of efforts has been made by portfolio managers in order to create appropriate portfolios for investors (Woodside-Oriakhi, Lucas, & Beasley, 2011). Making effective portfolio selections in real stock markets is not such an easy task (Corazza, Fasano, & Gusso, 2013). That is why all financial experts have been trying to find a better practical model compared with to the others.

Although, a proportion of portfolio selection decisions are taken on a qualitative basis, quantitative approaches are being used more frequently. Markowitz (1952) set up a quantitative framework for the selection of a portfolio (Chang, Meade, Beasley, & Sharaiha, 2000). He introduced mean-variance (M-V) model. The M-V model has become the foundation of

CONTACT Hoda Eskandar  heskandar@ut.ac.ir

© 2017 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the modern finance theory since 1950s (Chen, 2015). In its basic form, this model requires to determine the composition of a portfolio of assets which minimises risk while achieving a predetermined level of expected return (Crama & Schyns, 2003). His theory has revolutionised the way people think about portfolios of assets, and has gained widespread acceptance as a practical tool for portfolio optimisation. However, in some cases, the characteristics of the problem, such as its size, real-world requirements, very limited computation time, and limited precision in estimating instance parameters, may make analytical methods not particularly suitable for tackling large instances of this M-V model.

Such examples, however, are not necessarily a sign that the theory of risk-return optimisation is flawed. Rather, it means that the classical framework has to be modified when used in practice in order to achieve reliability, stability, and robustness with respect to model and limitations (Kolm, Tütüncü, & Fabozzi, 2013). Therefore, after Markowitz's work, further work has been done to improve and extend his model (Chen, 2015). These attempts, regarding the limitations of a factual market, have tried to make his model more practical.

Over recent years, various algorithms have been developed to solve portfolio optimisation problems. Most of these algorithms are based on numerical linear and nonlinear programming methods that may require substantial gradient information and usually seek to improve the solution in the neighbourhood of a starting point. These numerical optimisation algorithms provide a useful strategy to obtain the global optimum solution for simple and ideal models. The portfolio optimisation problem, however, is very complex in nature and quite difficult to solve. If there is more than one local optimum in the problem, the results may depend on the selection of the starting point for which the obtained optimal solution may not necessarily be the global optimum.

Furthermore, the gradient search method may become unstable when the objective function and constraints have multiple or sharp peaks. The drawbacks (i.e., efficiency and accuracy) of existing methods have encouraged researchers to rely on metaheuristic optimisation algorithms based on simulations and nature-inspired methods to solve optimisation problems. Metaheuristic algorithms commonly operate by combining rules and randomness to imitate natural phenomena (Goldberg, 1989; Woodside-Oriakhi et al., 2011). Genetic algorithms (GAs), one of the most widely used metaheuristic algorithms, is based on the genetic process of biological organisms (Goldberg, 1989). GA is the most popular optimisation method in financial optimisation problems used in the literature (Bermúdez, Segura, & Vercher, 2012; Chang, Sang-Chin, & Chang, 2009; Li & Xu, 2007; Lin & Liu, 2008; Najafi & Mushakhian, 2015; Ruiz-Torrubiano & Suarez, 2010; Soleimani, Golmakani, & Salimi, 2009).

Metaheuristic methods are capable of finding the near-optimum solution to the numerical real-valued test problems. This study introduces and utilises the water cycle algorithm (WCA) as a metaheuristic optimisation method for solving optimising portfolio selection. Hence, the ability and efficiency of the WCA to create high-quality solutions for Markowitz M-V model will be investigated. The idea of WCA was first suggested for solving engineering optimisation problems (Eskandar, Sadollah, Bahreininejad, & Hamdi, 2012).

Recently, the multi-objective water cycle algorithm (MOWCA) has been validated and implemented for solving constrained and unconstrained multi-criteria benchmark problems (Sadollah, Eskandar, & Kim, 2015; Sadollah, Eskandar, Kim, & Bahreininejad, 2014). The main objective of this article is to show the efficiency of WCA for solving portfolio optimisation problems. Statistical optimisation results obtained by the MOWCA are compared

with the other multi-objective optimisers named as non-dominated sorting GA (NSGA-II) (Deb, Pratap, Agarwal, & Meyarivan, 2002) and multi-objective particle swarm optimisation (MOPSO) (Li, 2003).

The rest of this article is organised as follows: Section 2 provides a literature review regarding the applied optimisers for solving portfolio optimisation problem. The portfolio optimisation problem is formulated and represented in Section 3. Multi-objective optimisation problems (MOPs) are explained in brief in Section 4 and then a detailed MOWCA is introduced in Section 5. Computational experiments, their comparisons, and discussions using different optimisers have been provided in Section 6. Finally, conclusions and further studies are drawn in Section 7.

2. Literature review

Despite the simplicity and intuitive appeal of portfolio construction using modern portfolio theory, it took many years until portfolio managers started using portfolio optimisation to manage real money. In real world applications, there are many concerns associated with its use, and portfolio optimisation is still considered by many practitioners to be impractical to apply (Kolm et al., 2013).

Recently, various algorithms have been developed to solve portfolio optimisation problems. Due to the complexity and the instantaneity of these problems, applying metaheuristic optimisation algorithms is a good alternative. The most of these algorithms have some shortcomings in solving the portfolio selection problem. However, as reported by Zhu et al. (2011), GA and particle swarm optimisation (PSO) required more computational time when the size of the problem is too large to find adequate solutions.

In fact, there are many applications using the GA and PSO (i.e., considered as the most commonly used metaheuristic optimisation methods) for solving portfolio optimisation problems (Chang et al., 2009; Li & Xu, 2007; Najafi & Mushakhian, 2015; Soleimani et al., 2009). Therefore, the following sections are designated for some applications of GA, PSO, and proposed WCA.

2.1. Genetic algorithm

Based on the Darwin principle ‘the fittest survive’ in nature, the GA was first introduced by Holland (1975) and has rapidly become the best-known techniques (Chang et al., 2009). The GA is an optimisation method which works by mimicking the evolutionary principles and chromosomal processing in natural genetics. It belongs to the larger class of evolutionary algorithms (EA), which generate solutions to optimisation problems using techniques inspired by natural evolution such as inheritance, mutation, selection, and crossover.

A GA begins its search with a random set of solutions usually coded in binary strings. Every solution is assigned a fitness which is directly related to the objective function of the optimisation problem. Thereafter, the population of solutions is modified to a new by applying three operators similar to natural genetic operators’ reproduction, crossover, and mutation. It works iteratively by successively applying these three operators in each generation till a termination criterion is satisfied (Holland, 1975).

Since the developing of GA, numerous related GA-based portfolio selection approaches have been tackled. For instance, Kyong et al. (2005) used the GA to support portfolio

optimisation for index fund management. Chang et al. (2009), and Lin and Liu (2008) also proposed the GA for portfolio selection problems with minimum transaction lots.

2.2. Particle swarm optimisation

The PSO is originally attributed to Kennedy and Eberhart (1995) and was first introduced for simulating social behaviour, as a stylised representation of the movement of organisms in a bird flock or fish school. It is based on a psychosocial model of social influence and social learning. A PSO swarm resembles a population, and a particle resembles an individual. It is initialised with a particle swarm, and each particle position represents a possible solution. The particles fly through the multidimensional search space by dynamically adjusting velocities according to its own experience and that of its neighbours (Kennedy et al., 2001).

The PSO has proven its efficiency in many empirical studies in the literature (Ali & Kaelo, 2008; Jiang et al., 2007; Tsai et al., 2010; Yang et al., 2007). The PSO have been widely applied for solving portfolio optimisation problems. The portfolio model was tested on various restricted and unrestricted risky investment portfolios and compared with the GA. The PSO demonstrated high computational efficiency in portfolio optimisation reported in the literature (Deng et al., 2012; Liu et al., 2012; Zhu, Wang, Wang, & Chen, 2011).

Recently, Golmakani and Fazel (2011) used the PSO for selecting the constrained portfolio. They showed that the proposed PSO effectively outperforms the GA especially in large-scale problems. Corazza et al. (2013) proposed a non-smooth penalty reformulation PSO for solving a complex portfolio selection problem.

2.3. Water cycle algorithm

The WCA is based on the observation of water cycle process and how rivers and streams flow into downhill towards the sea in nature (Eskandar et al., 2012). It was first introduced by Eskandar et al. (2012) for solving engineering optimisation problems. They showed that the WCA is more able to find a wider range of solutions compared with the GA and PSO. Compared to the GA and PSO, the WCA has demonstrated considerable success in providing good solution qualities to many complex engineering optimisation problems.

Recently, Haddad, Moravej, and Loáiciga (2014) utilised the WCA for finding optimal operation of reservoir systems. Their obtained optimisation results demonstrate the high efficiency and reliability of the WCA in solving reservoir operation problems. Lenin, Reddy, and Kalavathi (2014) used the WCA for detecting optimum reactive power dispatch problems. They claimed that the WCA has been applied on standard IEEE 30 bus test systems and simulation results show clearly about the superior performance of the WCA in decreasing real power loss.

Jabbar and Zainudin (2014) applied the WCA for attribute reduction problems in rough set theory. Based on their findings, it has been shown that the WCA performed equally well or even better than other methods for detecting optimal attribute selection. Guney and Basbug (2014) proposed an improved version of WCA, so called quantised WCA (QWCA) and applied it for solving antenna array pattern synthesis. The internal quantisation mechanism of QWCA is utilised to achieve digital values matching to the discrete values of the phase shifter instead of the simple rounding up/down routines after optimisation.

Therefore, the motivation is created to use the MOWCA (Sadollah et al., 2014, 2015) for solving multi-objective portfolio optimisation problem. Moreover, the portfolio

optimisation problem with multi-objective functions cannot be efficiently solved using traditionally approaches (Zhu et al., 2011). This article represents a multi-objective meta-heuristic approach to portfolio optimisation problem using the WCA.

3. Portfolio problem formulation

In this section, the cardinality constrained M-V portfolio optimisation problem has been formulated. For solving the portfolio selection problem, the notations used in this analysis are based on M-V model (Markowitz, 1952, 1991). Markowitz was the first researcher who applied variance or standard deviation as a measure of risk. He assumed that his classical formation can be formulated given as follows (Markowitz, 1952):

$$\text{Minimise } \sum_{i=1}^N \sum_{j=1}^N W_i W_j \delta_{ij}. \quad (1)$$

Equation (1) minimises the total variance (risk) associated with the portfolio. From a practical perspective, however, the Markowitz model may often be considered too basic. Real world investors have to face a lot of constraints such as trading limitations, size of the portfolio, etc. The portfolio optimisation problem becomes much more difficult if the number of variables is increased or if additional constraints such as cardinality constraints are introduced. Including such constraints in the formulation results in a problem which is considerably more practical than the original model (Maringer & Kellerer, 2003).

Therefore, in order to enrich the model, as can be seen in Equation (1), in this study, the portfolio optimisation problem is investigated as a multi-objective problem, where the expected return is maximised (i.e., first objective function) and the risk is minimised (i.e., second objective function). Therefore, the standard format of our objectives and their constraints are given in the following equations:

$$\text{Maximise Portfolio Return } F_1(w) = \sum_{i=1}^N w_i \bar{r}_i, \quad (2)$$

$$\text{Minimise Portfolio Risk } F_2(w) = \sum_{i=1}^N \sum_{j=1}^N w_i w_j \delta_i \delta_j \rho_{ij}, \quad (3)$$

subject to:

$$\sum_{i=1}^N w_i = 1, \quad (4)$$

$$0 \leq w_i \leq 1, \quad i = 1, 2, 3, \dots, N, \quad (5)$$

where r_i is the expected return of asset i^{th} , N is the number of available assets, w_i is the proportion of the portfolio held in assets i^{th} , ρ_{ij} is the correlation between asset i^{th} and j^{th} , and δ_i and δ_j are the standard deviation of stocks returns i^{th} and j^{th} , respectively. Therefore, Equation (2) maximises the portfolio return, Equation (3) minimises the total variance (risk)

associated with the portfolio. In terms of applied constraints, Equation (4) ensures that the sum of the proportions is equal to one. Also, the proportion held in each asset is between zero (minimum amount) and one (maximum amount) as can be seen in Equation (5).

4. Standard format of multi-objective optimisation problems

MOPs is an area of multiple criteria decision-making involving more than one objective function to be optimised simultaneously as can be seen in the following equation:

$$F(X) = [F_1(X), F_2(X), \dots, F_m(X)]^T \quad (6)$$

where $X = [x_1, x_2, x_3, \dots, x_d]$ is a vector of design variables, and, d and m are the number of design variables and objectives, respectively. A simple approach for solving MOPs is to use a wide variety of weight factors to convert a MOPs into a single-objective optimisation problem. It can be formulated based on the following equation (Haupt & Haupt, 2004):

$$F = \sum_{i=1}^m w_i f_i \quad (7)$$

where m is the number of objective functions and w_i and f_i are weighting factors and objective functions, respectively. However, it is time consuming task and it is considered as a major downside of this method. The most common way to solve MOPs is by keeping a set of best solutions in an archive and updating the archive at each iteration. In this method, the best solutions are defined as non-dominated solutions or Pareto optimal solutions. A solution can be considered as a non-dominated solution if and only if the following conditions become satisfied by the solution as given follows (Coello C.A.C 2000):

- (a) Pareto dominance: $U = (u_1, u_2, u_3, \dots, u_n) < V = (v_1, v_2, v_3, \dots, v_n)$ if and only if U is partially less than V in the objective space which it means:

$$\begin{cases} F_i(U) \leq F_i(V) & \forall i \\ F_i(U) < F_i(V) & \exists i \end{cases}, \quad i = 1, 2, 3, \dots, m \quad (8)$$

- (b) Pareto optimal solution: vector U is a Pareto optimal solution if and only if any other solutions cannot be determined to dominate U . A set of Pareto optimal solutions is called a Pareto optimal front ($PF_{optimal}$).

Figure 1 shows that among three solutions A , B , and C , solution C has the highest values for f_1 and f_2 . It means that this solution is a solution dominated by solutions A and B . In contrast, both solutions A and B can be considered as non-dominated solutions, as neither of them dominates each other (Sadollah et al., 2015).

4.1. Performance metrics for MOPs

Four performance parameters that are used to evaluate the performance of metaheuristic algorithms are investigated in this article. These criteria are listed as follows: generational

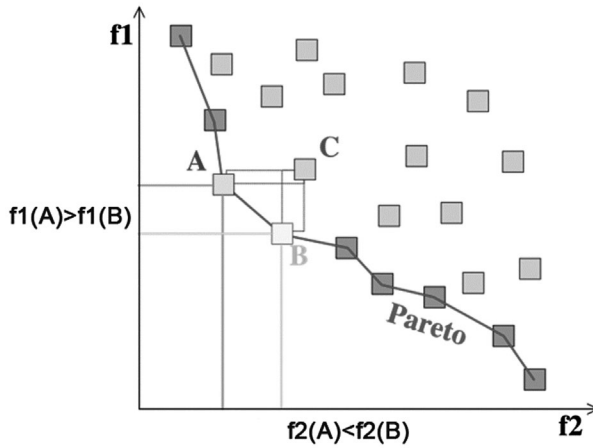


Figure 1. Optimal Pareto solutions (A and B) for the two-dimensional space. Source: Sadollah et al. (2015).

distance (GD) metric, metric of spacing, diversity metric, and metric of maximum spread (MS) widely used in the literature (Sadollah et al., 2015).

Veldhuizen and Lamont (1998) introduced the GD metric. It clarifies the capability of the different algorithms of finding a set of non-dominated solutions having the lowest distance with the Pareto optimal fronts ($PF_{optimal}$). An algorithm with the minimum GD has the best convergence to the $PF_{optimal}$. This evaluator is defined as the following:

$$GD = \left(\frac{1}{NPF} \sum_{i=1}^{NPF} d_i^2 \right)^{1/2}, \tag{9}$$

where NPF is the number of members in the generated Pareto front (PF_g) and d is Euclidean distance between member i^{th} in the PF_g and the nearest member in the $PF_{optimal}$. Figure 2 shows a schematic view of GD performance metric for the 2D space. The best obtained value for the GD metric is equal to zero which means that the PF_g exactly covers the $PF_{optimal}$.

Schott (1995) suggested the metric of spacing (S). It shows the distribution of non-dominated solutions obtained by a specified algorithm. It can show how well the obtained solutions are distributed among each other. It is defined as the following equation:

$$S = \sqrt{\frac{1}{NPF - 1} \sum_{i=1}^{NPF} (\tilde{d}_i - \bar{d})^2}, \tag{10}$$

where $\tilde{d}_i = \min_j (|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)|)$, $i, j = 1, \dots, NPF$ and \bar{d} is average of all d_i . The smallest value of S gives the best uniform distribution in the PF_g . If all non-dominated solutions are uniformly distributed in the PF_g , then, the values of d_i and \bar{d} are the same, therefore, the value of S metric is zero. Figure 3 displays a schematic view of the spacing metric used in this article.

Deb (2001) proposed the metric of diversity (Δ). It determines the extent of spread attained by the non-dominated solutions obtained from a specified algorithm. To be more

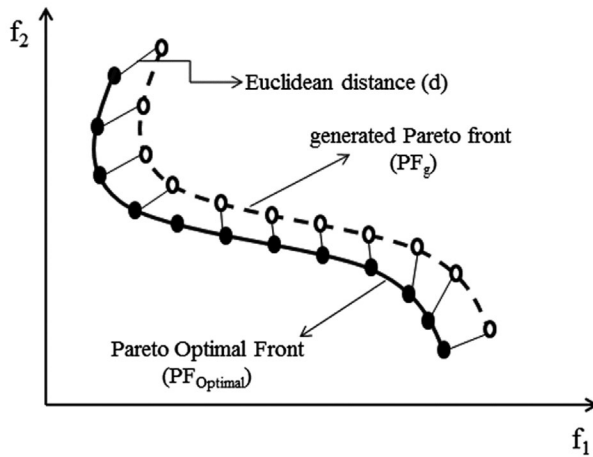


Figure 2. Schematic view of GD criterion for the MOPs. Source: Veldhuizen and Lamont (1998).

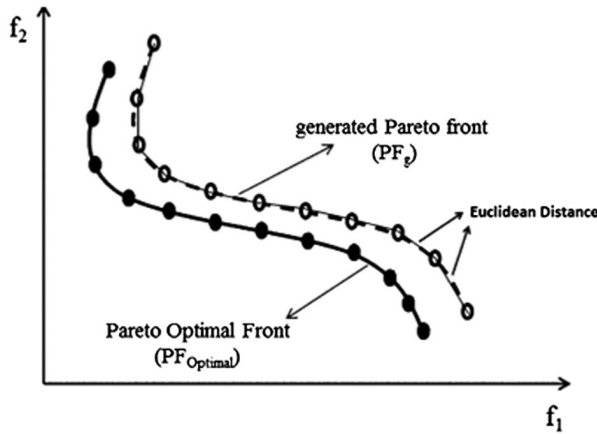


Figure 3. Schematic view of S evaluator for the MOPs. Source: Scott (1995).

precise, it can analyse how the solutions are extended across the $PF_{optimal}$. It is defined as follows:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{NPF} |d_i - \bar{d}|}{d_f + d_l + (NPF - 1)\bar{d}} \tag{11}$$

where d_f and d_l are Euclidean distance between the extreme solutions in the $PF_{optimal}$ and PF_g , respectively. Also, d_i is Euclidean distance between each point in the PF_g and the closest point in the $PF_{optimal}$.

Value of the Δ metric is always greater than zero, and a small value of Δ means better distribution and spread of the solutions. In this condition, $\Delta = 0$ is the perfect condition indicating that extreme solutions of the $PF_{optimal}$ have been found and that $d_i = \bar{d}$ for all

non-dominated points. Figure 4 shows a schematic view of the Δ performance metric for a given Pareto front.

The metric of MS measures how ‘well’ the $PF_{optimal}$ is covered by the PF_g through hyper-boxes formed by the extreme function values observed in the $PF_{optimal}$ and PF_g . It is defined as follows (Kaveh & Laknejadi, 2011):

$$MS = \left[\frac{1}{m} \sum_{i=1}^m \left[\frac{\min(f_i^{max}, F_i^{max}) - \max(f_i^{min}, F_i^{min})}{F_i^{max} - F_i^{min}} \right]^2 \right]^{\frac{1}{2}}, \tag{12}$$

where f_i^{max} and f_i^{min} are the maximum and minimum of the i^{th} objective in the $PF_{optimal}$ respectively, F_i^{max} and F_i^{min} are the maximum and minimum of the i^{th} objective in the $PF_{optimal}^g$ respectively. Larger value of MS implies a better spread of solutions.

5. Multi-objective WCA for portfolio optimisation

5.1. Water cycle algorithm

The WCA mimics the flow of rivers and streams toward the sea and derived by the observation of water cycle process. Let us assume that there are some rain or precipitation phenomena. An initial population of design variables (i.e., population of streams) is randomly generated after raining process. The best individual (i.e., the best stream), classified in terms of having the minimum cost function (for minimisation problem), is chosen as the sea.

Then, a number of good streams (i.e., cost function values close to the current best record) are chosen as rivers, while the other streams flow into the rivers and sea. In an D dimensional optimisation problem, a stream is an array of $1 \times D$. Starting an optimisation algorithm, an initial population representing a matrix of streams of size $N_{pop} \times D$ is generated. Hence, the matrix of initial population, which is generated randomly, is given as (rows and column are the number of population and the number of design variables, respectively):

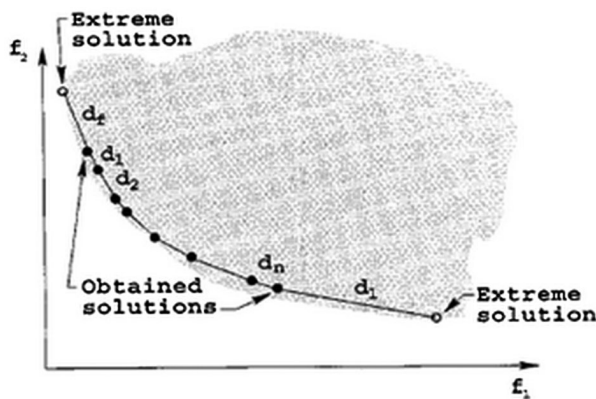


Figure 4. Schematic view of diversity metric (Δ) for the MOPs. Source: Deb et al. (2002).

$$\text{Total Population} = \begin{bmatrix} \text{Sea} \\ \text{River}_1 \\ \text{River}_2 \\ \text{River}_3 \\ \vdots \\ \text{Stream}_{N_{sr}+1} \\ \text{Stream}_{N_{sr}+2} \\ \text{Stream}_{N_{sr}+3} \\ \vdots \\ \text{Stream}_{N_{pop}} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & \dots & x_D^1 \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_D^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{N_{pop}} & x_2^{N_{pop}} & x_3^{N_{pop}} & \dots & x_D^{N_{pop}} \end{bmatrix}, \quad (13)$$

where N_{pop} and D are population size and the number of design variables, respectively. Each of the decision variable values (x_1, x_2, \dots, x_D) can be represented as floating point number (real values) or as a predefined set for continuous and discrete problems, respectively. The cost of a stream is obtained by the evaluation of cost function (fitness function).

At the first step, N_{pop} streams are created. A number of N_{sr} from the best individuals (minimum values) are selected as a sea and rivers. The stream which has the minimum value among others is considered as the sea. In fact, N_{sr} is the summation of number of rivers (which is defined by user) and a single sea. The rest of the population (i.e., streams flow into the rivers or may directly flow to the sea) are considered as streams.

Depending on magnitude of flow, each river absorbs water from streams. The amount of water entering a river and/or the sea, hence, varies from stream to stream. In addition, rivers flow to the sea which is the most downhill location. The designated streams for each rivers and sea are calculated using the following equation (Eskandar et al., 2012):

$$C_n = Cost_n - Cost_{N_{sr}+1} \quad n = 1, 2, 3, \dots, N_{sr}, \quad (14)$$

$$NS_n = \text{round} \left\{ \frac{C_n}{\sum_{n=1}^{N_{sr}} C_n} \times N_{Streams} \right\}, \quad n = 1, 2, \dots, N_{sr}, \quad (15)$$

where NS_n is the number of streams which flow to the specific rivers and sea. As it happens in nature, streams are created from the raindrops and join each other to generate new rivers. Some stream may even flow directly to the sea. All rivers and streams end up in the sea that corresponds to the current best solution. Let us assume that there are N_{pop} streams of which $N_{sr}-1$ are selected as rivers and one is selected as the sea. Figure 5a shows the schematic view of a stream flowing towards a specific river along their connecting line.

At the exploitation phase in the WCA, new positions for streams and rivers have been suggested as follows (Eskandar et al., 2012):

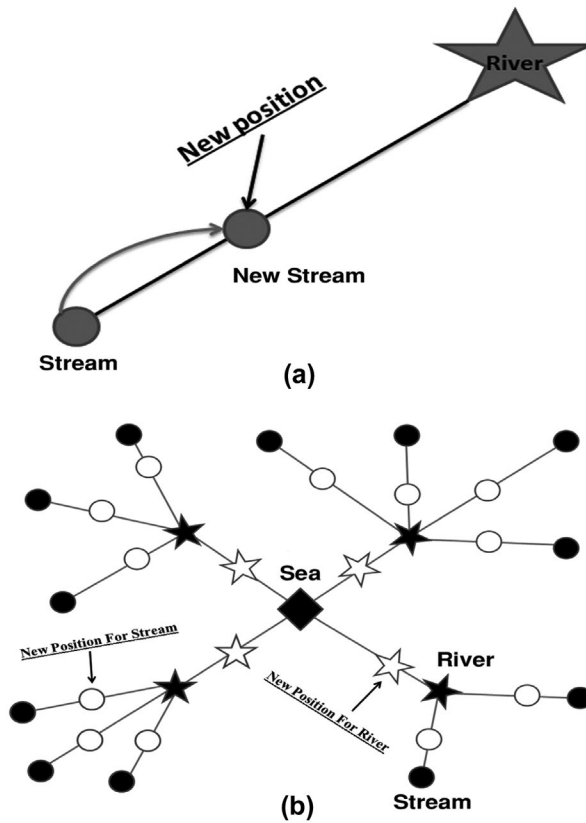


Figure 5. Schematic description of: (a) the stream's flow to a specific river; (b) the WCA optimisation process. Source: Eskandar et al. (2012).

$$\vec{X}_{Stream}^{t+1} = \vec{X}_{Stream}^t + rand \times C \times (\vec{X}_{Sea}^t - \vec{X}_{Stream}^t) \tag{16}$$

$$\vec{X}_{Stream}^{t+1} = \vec{X}_{Stream}^t + rand \times C \times (\vec{X}_{River}^t - \vec{X}_{Stream}^t), \tag{17}$$

$$\vec{X}_{River}^{t+1} = \vec{X}_{River}^t + rand \times C \times (\vec{X}_{Sea}^t - \vec{X}_{River}^t), \tag{18}$$

where $1 < C < 2$ and the best value for C may be chosen as 2 and $rand$ is an uniformly distributed random number between zero and one. Equations (16) and (17) are for streams which flow into the sea and their corresponding rivers, respectively. Notations having vector sign correspond to vector values, otherwise the rest of notations and parameters are considered as scalar values. If the solution given by a stream is better than its connecting river, the positions of river and stream are exchanged (i.e., the stream becomes a river and the river becomes a stream). A similar exchange can be performed for a river and the sea.

The evaporation process operator also is introduced to avoid premature (immature) convergence to local optima (exploitation phase) (Sadollah et al., 2015). Basically, evaporation

causes sea water to evaporate as rivers/streams flow into the sea. This leads to new precipitations. Therefore, we have to check if the river/stream is sufficiently close to the sea to make the evaporation process occur. For that purpose, the following criterion is utilised for evaporation condition (Eskandar et al., 2012):

$$\text{if } \left\| \vec{X}_{Sea}^t - \vec{X}_{River_j}^t \right\| < d_{max} \quad \text{or} \quad rand < 0.1 \quad j = 1, 2, 3, \dots, N_{sr} - 1$$

Perform raining process by unifrom random search

end

where d_{max} is a small number close to zero. After evaporation, the raining process is applied and new streams are formed in the different locations (similar to mutation in the GAs). Hence, in the new generated sub-population, the best stream will act as a new river and other streams move toward their new river. This condition will also apply for streams that directly flow to the sea.

Similarly, the best newly formed stream is considered as a river flowing to the sea. The rest of new streams are assumed to flow into the rivers or may directly flow into the sea. The following equation is used only for the streams which directly flow to the sea. It encourages the creation of streams which directly flow to the sea in order to improve the exploration near the sea (the optimum solution) in the feasible region for constrained problems (Eskandar et al., 2012):

$$\vec{X}_{Stream}^{t+1} = \vec{X}_{Sea}^t + \sqrt{\mu} \times randn(1, D), \quad (19)$$

where μ is a coefficient which shows the range of searching region near the sea, $randn$ is the normally distributed random number. The larger μ increases the possibility to exit from feasible region. The smaller μ leads the algorithm to search in smaller region near the sea. Its suitable value is set to 0.1. Indeed, term $\sqrt{\mu}$ represents the standard deviation. The generated individuals with variance μ are distributed around the best obtained optimum point (sea).

Therefore, the evaporation operator is responsible for the exploration phase in the WCA.

A large value for d_{max} prevents extra searches and small values encourage the search intensity near the sea. Therefore, d_{max} controls the search intensity near the sea (i.e., best obtained solution). The value of d_{max} adaptively decreases as follows:

$$d_{max}^{t+1} = d_{max}^t - \frac{d_{max}^t}{Max_Iteration} \quad t = 1, 2, 3, \dots, Max_Iteration \quad (20)$$

where t is an iteration index. The development of the WCA optimisation process is illustrated by Figure 5b where circles, stars, and the diamond correspond to streams, rivers, and sea, respectively. The white (empty) shapes denote the new positions taken by streams and rivers.

5.2. Multi-objective WCA

In order to convert the WCA to an efficient multi-objective optimisation algorithm, the pre-dominant features of the algorithm (i.e., sea and rivers) should be defined correctly. In ordinary optimisation problems for the WCA, only one objective function needs to

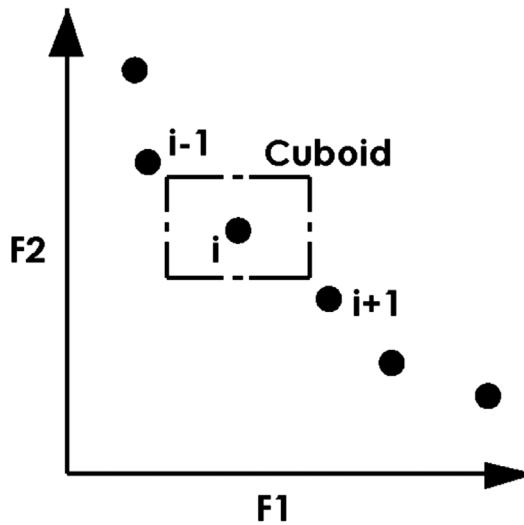


Figure 6. Schematic view of crowding-distance calculation. Source: Deb et al. (2002).

be minimised (maximised). In this condition, a number of best obtained solutions in the population are selected as a sea (i.e., best obtained solution so far) and rivers.

Nevertheless, in MOPs, there is more than one function to be minimised (maximised). Therefore, the definition in the WCA for selecting sea and rivers has to be changed in the multi-objective space. A crowding-distance mechanism is used for selecting the most efficient (best) solutions in the population as a sea and rivers. At first, Deb et al. (2002) defined the concept of crowding-distance mechanism. This criterion shows the distribution of non-dominated solutions around a particular non-dominated solution. Figure 6 illustrates how to calculate the crowding-distance (i.e., the average side length of the cuboid) for point i .

A lower value for the crowding-distance indicates greater distribution of the solutions in a specific region. In MOPs, this parameter is calculated in objective spaces. Hence, all non-dominated solutions must be sorted in terms of values for one of the objective functions. It is necessary to compute this parameter for each non-dominated solution. A vital step in the MOWCA is to select the sea and rivers from the obtained population as the best solutions for other solutions at each iteration. It affects the convergence capability of the MOWCA and maintains a good spread of non-dominated solutions.

Therefore, the crowding-distance for all non-dominated solutions must be calculated for all iterations. It is necessary to determine which solutions have the highest crowding-distance values. Afterwards, the obtained non-dominated solutions are designated as a sea and rivers and, furthermore, the flow intensity of rivers and sea is calculated based on the crowding-distance values. Some non-dominated solutions will most likely be created around the sea and rivers at the next iteration, and their crowding-distance values will be amended and reduced.

Moreover, it is very important to save non-dominated solutions in an archive to generate the Pareto front sets. This archive is updated at each iteration, and any dominated solutions are eliminated from the archive. Therefore, whenever the number of members in the Pareto archive becomes greater than the Pareto archive size, the crowding-distance is applied again to eliminate as many as necessary of the non-dominated solutions having the lowest crowding-distance values among the Pareto archive members. The MOWCA has the great

potential for exploitation in the design space as it focuses in near optimum solutions and exploits far distance solutions.

Mostly, the MOWCA starts with exploitation approach, moving streams toward rivers and rivers toward the sea. However, at early iteration, these movements act as an exploration operator in the MOWCA due to the diversity of early population. Afterwards, the MOWCA applies evaporation condition in order to perform exploration phase. This trend can be considered as the potential of the MOWCA to search a wide range of design space, while concentrating to near optimum non-dominated solutions (Sadollah et al., 2015).

Many MOPs are subjected to a set of constraints (e.g., inequality, equality, linear, nonlinear, and so forth). Therefore, finding the good and simple strategies to handle the imposed constraints and detect solutions in the feasible area is important. Hence, a simple approach is defined here to apply on the MOWCA. After generating a set of solutions at each iteration, all constraints are checked and some solutions that are in the feasible area are selected. Afterward, non-dominated solutions are selected from the feasible solutions and inserted into the Pareto archive. Finally, sea and rivers are chosen from this archive for the next iteration (Sadollah et al., 2015).

The detailed steps of MOWCA are described as follows (Sadollah et al., 2015):

Step 1: Choose the initial parameters of the MOWCA: N_{sr} , d_{max} , N_{pop} , maximum iteration number, and Pareto archive size.

Step 2: Create random initial population and form the initial streams, rivers, and sea.

Step 3: Create the value of multi-objective functions for each stream.

Step 4: Determine the non-dominated solutions in the initial population and save them in the Pareto archive.

Step 5: Determine the non-dominated solutions among the feasible solutions and save them in the Pareto archive.

Step 6: Calculate the crowding-distance for each Pareto archive member.

Step 7: Select a sea and rivers based on the crowding-distance value.

Step 8: Determine flow intensity of rivers and sea based on the crowding distance values by Equations (14) and (15).

Step 9: Some streams may directly flow into the sea using Equation (16).

Step 10: Exchange positions of sea with a stream which gives the best solution.

Step 11: Streams flow into the rivers by Equation (17).

Step 12: Exchange positions of river with a stream which gives the best solution.

Step 13: Rivers flow into the sea using Equation (18).

Step 14: Exchange positions of sea with a river which gives the best solution.

Step 15: Check the evaporation condition using the pseudo-code in Section 4.1.

Step 16: The raining process will occur if the evaporation condition is satisfied.

Step 17: Reduce the value of d_{max} which is a user defined parameter using Equation (19).

Step 18: Determine the new feasible solutions in the population.

Step 19: Determine the new non-dominated solutions among the feasible solutions and save them in the Pareto archive.

Step 20: Eliminate any dominated solutions in the Pareto archive.

Step 21: Go to the Step 22 if the number of member in the Pareto archive is more than the determined Pareto archive sizes, other-wise, go to the Step 23.

Step 22: Calculate the crowding-distance value for each Pareto archive member and remove as many members as necessary with the lowest crowding-distance value.

Step 23: Calculate the crowding-distance value for each Pareto archive member to select new sea and rivers.

Step 24: Check the convergence criterion. The WCA will be stopped if the stopping criterion is satisfied, otherwise return to the Step 9.

6. Computational experiments and discussion

Portfolio selection problem seeks the best allocation of wealth among different investment opportunities in a market consisting of risky assets. Determination of optimal portfolios is a rather complex problem depending on the objective of the investor. Generally, an investor tends to lowest risk and highest return. Therefore, they are faced with a multi-objective function. High return is accompanied with high risk and so, an investor must explicitly consider the trade-off between the risk and return in deciding which portfolio to adopt.

In this section, a set of computational experiments are used to clarify the performance of MOWCA. In this article, the MOWCA is compared with the NSGA-II (Deb et al., 2002) and MOPSO (Li, 2003). The MOWCA, NSGA-II, and MOPSO were coded in MATLAB programming software, and the optimisation task of each experiment was executed using 20 independent runs.

For all experiments, the initial parameters for the MOWCA were selected as 50, 8 and $1E-05$, for N_{pop} , N_{sr} , and d_{max} , respectively. In terms of MOPSO and NSGA-II, population size of 50 is considered for all experiments. Values of 0.75 and 1.5 were set for inertia weight and both learning coefficients used in MOPSO, respectively. Talking about user parameters of NSGA-II, the crossover and mutation rates were 0.5 and 0.1, accordingly. It is worth mentioning that the used initial parameters for all reported optimisers have been fine-tuned and the optimal initial values are used for the optimisation task using sensitivity analyses.

In order to have fair and reliable comparison, the maximum number of function evaluations (NFEs) used and the Pareto archive size for all MOPs were set to 10,000 and 100, respectively. In fact, the maximum NFEs is taken as the stopping condition is assumed in this article. Four performance metrics, widely used in the literature, have been utilised for evaluating the efficiency of MOWCA and the other optimisers. It is worth mentioning that both applied optimisers (i.e., NSGA-II and MOPSO) have been coded and implemented in this article.

Four MOPs are investigated to assess the capability of the MOWCA in handling constrained portfolio optimisation problems involving daily data from January 2012 to December 2014. Data obtained from the following indices: 97 firms from S&P 100 in the US, 50 firms from Hang Seng (HSI) in Hong Kong, 96 firms from FTSE 100 in the UK, and 30 firms from DAX in Germany.

Table 1. Statistical optimisation results obtained by the GD and S metrics for the S&P problem using different optimisers.

| Method | GD | | | | S | | | |
|---------|----------|----------|----------|----------|----------|----------|----------|----------|
| | Worst | Average | Best | SD | Worst | Average | Best | SDa |
| NSGA-II | 1.55E-03 | 1.16E-03 | 9.32E-04 | 1.92E-04 | 7.25E-01 | 4.50E-01 | 3.15E-01 | 1.04E-01 |
| MOPSO | 2.01E-02 | 9.86E-03 | 5.01E-03 | 4.44E-03 | 3.93E-01 | 2.49E-01 | 1.82E-01 | 7.40E-02 |
| MOWCA | 1.50E-03 | 1.07E-03 | 6.57E-04 | 2.06E-04 | 3.70E-01 | 2.22E-01 | 1.66E-01 | 5.25E-02 |

Source: Data processing was performed off-line using a commercial software package MATLAB 8.5, The MathWorks Inc., Natick, MA, 2000.

^aStandard deviation.

Table 2. Statistical optimisation results obtained by the MS and Δ metrics for the S&P problem using different optimisers.

| Method | Δ | | | | MS | | | |
|---------|----------|----------|----------|----------|----------|----------|----------|----------|
| | Worst | Average | Best | SD | Worst | Average | Best | SD |
| NSGA-II | 6.29E-01 | 4.96E-01 | 3.69E-01 | 7.68E-02 | 5.69E-01 | 7.50E-01 | 9.20E-01 | 1.09E-01 |
| MOPSO | 7.37E-01 | 4.76E-01 | 2.91E-01 | 2.14E-01 | 0.97E-01 | 6.38E-01 | 9.12E-01 | 2.88E-01 |
| MOWCA | 6.30E-01 | 3.40E-01 | 2.41E-01 | 8.37E-02 | 4.44E-01 | 8.32E-01 | 9.21E-01 | 1.05E-01 |

Source: Data processing was performed off-line using a commercial software package MATLAB 8.5, The MathWorks Inc., Natick, MA, 2000.

6.1. S&P100 (97 stocks)

Tables 1 and 2 show statistical optimisation results for the considered evaluators using the NSGA-II, MOPSO, and MOWCA for the S&P100 problem. As it can be seen from Table 1, MOWCA could find the non-dominate solutions with the minimum distance from the $PF_{optimal}$ (see GD metric in Table 1) and better distribution (see S metric in Table 1) compared with the other optimisers. In terms of Δ and MS metrics, as the statistical results show in Table 2, the MOWCA have found better non-dominated solutions against the MOPSO and NSGA-II.

Moreover, the extreme solutions (ranges) obtained by the MOWCA are (0.454, 0.152) and (1.464, 0.511). In contrast, the minimum and maximum returns obtained by the NSGA-II are 0.412 units with a risk of 0.151 and 1.375 units with a deflection of 0.445, respectively. Hence, NSGA-II has found the lowest return. On the other hand, MOWCA could find better maximum return against the NSGA-II.

Figure 7 demonstrates computed (generated) Pareto front using the reported multi-objective optimisers for solving S&P problem. Figure 7 shows the capability of each algorithm to find the $PF_{optimal}$ and also how well the final non-dominated solutions are distributed

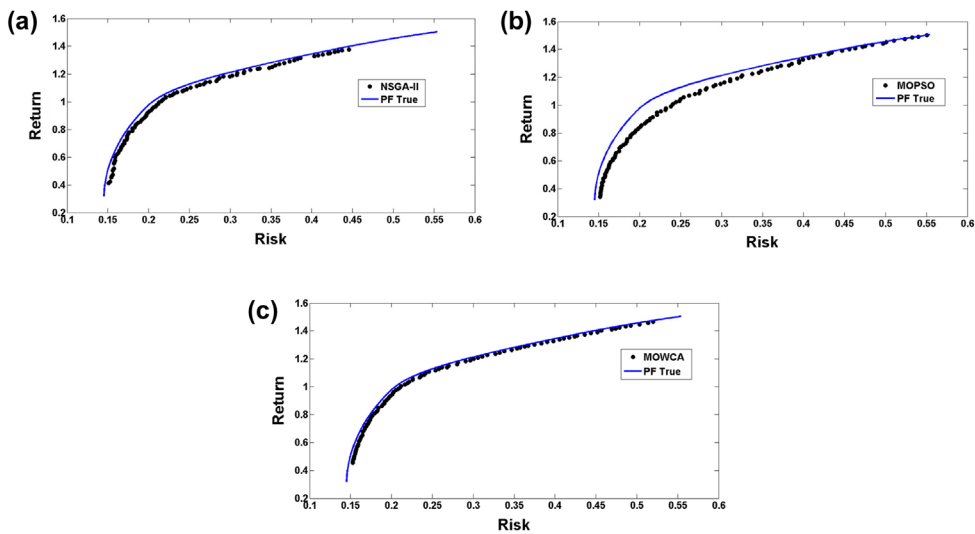


Figure 7. Generated Pareto front versus optimal (true) Pareto front (PF True) for solving S&P problem obtained by the: (a) NSGA-II; (b) MOPSO; (c) MOWCA. Source: Data processing was performed off-line using a commercial software package MATLAB 8.5, The MathWorks Inc., Natick, MA, 2000.

next to each other. Looking at Figure 7c, it can be concluded that the MOWCA was successful in covering the optimal Pareto front, having satisfactory distribution and spread for the non-dominated solutions.

6.2. Hang Seng (50 stocks)

In terms of statistical optimisation results reported by the evaluators, the MOWCA has the best convergence to the $PF_{optimal}$ (i.e., GD metric in Table 3), as shown in Table 3 for the HSI problem. Regarding the metric of spacing, the MOWCA has the best distribution of non-dominated solutions compared to the other considered methods (see S metric in Table 3).

Talking about the MS and Δ metrics reported in Table 4, the MOWCA has surpassed the NSGA-II and MOPSO (only for Δ metric). However, the MOPSO is superior over the MOWCA in terms of MS metric as shown in Table 4.

It is worth pointing out that the NSGA-II could perform better in finding the lowest return, while the MOWCA had a better performance in finding the maximum return. In terms of extreme solutions' locations, the MOWCA has found the extreme solutions with the position of (0.353, 0.174) and (1.305, 0.512), whereas, the minimum and maximum values achieved by the NSGA-II are (0.360, 0.174) and (1.292, 0.505), respectively. As it can be seen, the interval span covered by the MOWCA is [0.952, 0.338], while the detected interval span obtained by the NSGA-II is [0.932, 0.331] which shows the capability of MOWCA in finding wider range of non-dominate solutions.

Figure 8 displays a graphical comparison among reported optimisers in finding Pareto optimal solution for the HSI experiment. The non-dominated solutions obtained by the MOWCA are finely distributed compared to the NSGA-II as can be seen in Figures 8a and 8c. In terms of MOPSO (Figure 8b), it shows fine coverage of non-dominated solutions in

Table 3. Statistical optimisation results obtained by the GD and S metrics for the HSI problem using different optimisers.

| Method | GD | | | | S | | | |
|---------|----------|----------|----------|----------|----------|----------|----------|----------|
| | Worst | Average | Best | SD | Worst | Average | Best | SD |
| NSGA-II | 6.87E-04 | 5.85E-04 | 4.53E-04 | 7.40E-05 | 5.44E-01 | 3.02E-01 | 2.22E-01 | 8.59E-02 |
| MOPSO | 1.10E-01 | 1.15E-02 | 2.26E-03 | 2.43E-02 | 2.15E-01 | 1.99E-01 | 1.80E-01 | 2.16E-02 |
| MOWCA | 9.21E-04 | 4.40E-04 | 2.01E-04 | 1.75E-04 | 2.28E-01 | 1.97E-01 | 1.50E-01 | 2.81E-02 |

Source: Data processing was performed off-line using a commercial software package MATLAB 8.5, The MathWorks Inc., Natick, MA, 2000.

Table 4. Statistical optimisation results obtained by the MS and Δ metrics for the HSI problem using different optimisers.

| Method | Δ | | | | MS | | | |
|---------|----------|----------|----------|----------|----------|----------|----------|----------|
| | Worst | Average | Best | SD | Worst | Average | Best | SD |
| NSGA-II | 4.16E-01 | 3.38E-01 | 2.76E-01 | 3.23E-02 | 8.00E-01 | 8.80E-01 | 9.53E-01 | 3.73E-02 |
| MOPSO | 3.65E-01 | 2.98E-01 | 1.88E-01 | 6.04E-02 | 8.67E-01 | 9.46E-01 | 1.09E+00 | 5.26E-02 |
| MOWCA | 3.51E-01 | 2.92E-01 | 1.77E-01 | 4.63E-02 | 8.43E-01 | 8.85E-01 | 9.65E-01 | 3.26E-02 |

Source: Data processing was performed off-line using a commercial software package MATLAB 8.5, The MathWorks Inc., Natick, MA, 2000.

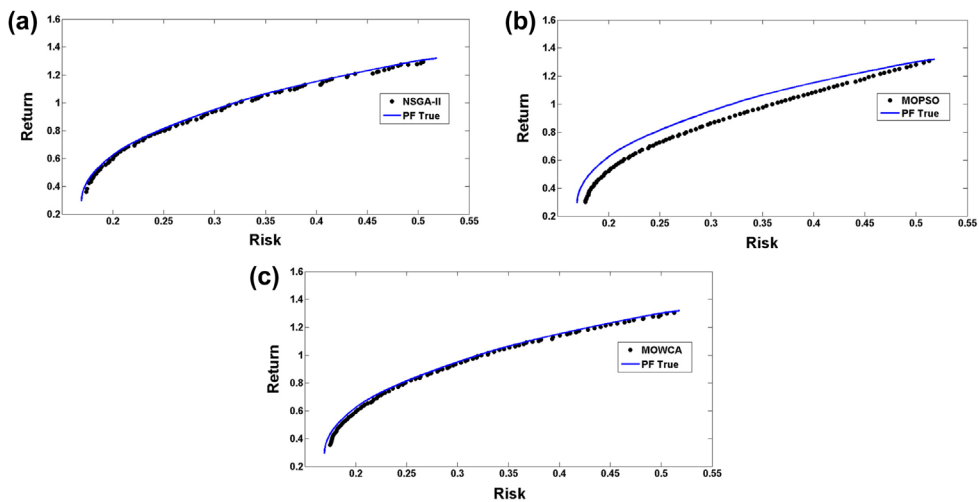


Figure 8. Generated Pareto front versus optimal (true) Pareto front (PF True) for solving HSI problem obtained by the: (a) NSGA-II; (b) MOPSO; (c) MOWCA. Source: Data processing was performed off-line using a commercial software package MATLAB 8.5, The MathWorks Inc., Natick, MA, 2000.

Table 5. Statistical optimisation results obtained by the GD and S metrics for the FTSE problem using different optimisers.

| Method | GD | | | | S | | | |
|---------|----------|----------|----------|----------|----------|----------|----------|----------|
| | Worst | Average | Best | SD | Worst | Average | Best | SD |
| NSGA-II | 1.80E-03 | 1.17E-03 | 7.61E-04 | 2.55E-04 | 9.13E-01 | 5.33E-01 | 3.04E-01 | 1.78E-01 |
| MOPSO | 4.94E-02 | 1.98E-02 | 8.61E-03 | 1.14E-02 | 5.67E-01 | 3.49E-01 | 2.37E-01 | 7.89E-02 |
| MOWCA | 3.52E-03 | 1.98E-03 | 6.14E-04 | 8.34E-04 | 5.23E-01 | 2.79E-01 | 1.66E-01 | 1.07E-01 |

Source: Data processing was performed off-line using a commercial software package MATLAB 8.5, The MathWorks Inc., Natick, MA, 2000.

Table 6. Statistical optimisation results obtained by the MS and Δ metrics for the FTSE problem using different optimisers.

| Method | Δ | | | | MS | | | |
|---------|----------|----------|----------|----------|----------|----------|----------|----------|
| | Worst | Average | Best | SD | Worst | Average | Best | SD |
| NSGA-II | 7.23E-01 | 6.00E-01 | 5.19E-01 | 6.34E-02 | 5.64E-01 | 6.71E-01 | 7.78E-01 | 5.28E-02 |
| MOPSO | 7.27E-01 | 4.58E-01 | 3.35E-01 | 1.67E-01 | 3.48E-01 | 6.52E-01 | 7.65E-01 | 2.33E-01 |
| MOWCA | 6.62E-01 | 3.86E-01 | 2.83E-01 | 7.83E-02 | 6.65E-01 | 7.81E-01 | 8.95E-01 | 1.14E-01 |

Source: Data processing was performed off-line using a commercial software package MATLAB 8.5, The MathWorks Inc., Natick, MA, 2000.

the objective space, however, it suffers the capability of being close to the Pareto optimal solution (i.e., GD metric) with respect to MOWCA.

6.3. FTSE100 (96 stocks)

The final statistical optimisation results obtained by the reported methods for the performance measurements are shown in Tables 5 and 6 for the FTSE100 problem. As can be seen

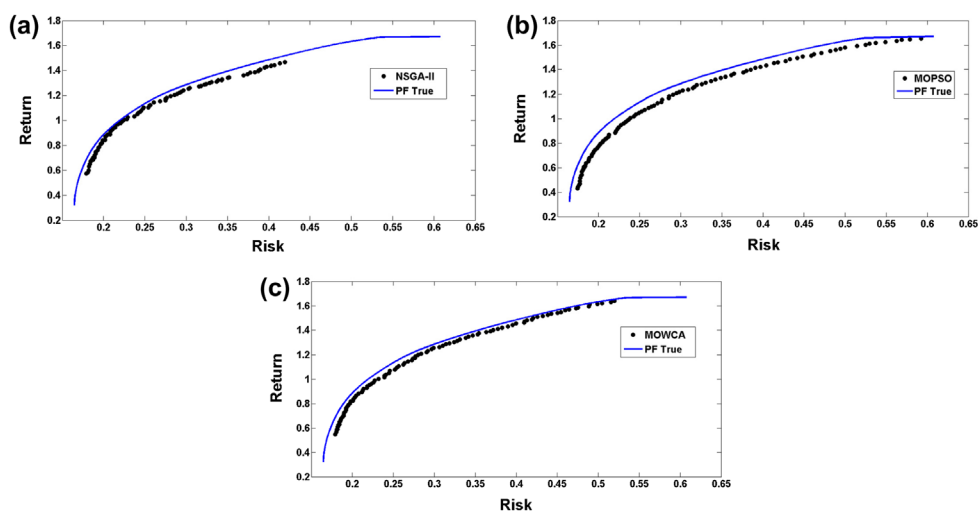


Figure 9. Generated Pareto front versus optimal (true) Pareto front (PF True) for solving FTSE problem obtained by the: (a) NSGA-II; (b) MOPSO; (c) MOWCA. Source: Data processing was performed off-line using a commercial software package MATLAB 8.5, The MathWorks Inc., Natick, MA, 2000.

from Table 5, the NSGA-II shows better statistical results for the GD metric compared to the MOPSO and MOWCA. However, for the rest of the reported evaluators, the NSGA-II was not successful as can be observed in Tables 5 and 6.

The non-dominated solutions obtained by the MOWCA have the best distribution and spread, which can be inferred from the statistical results of the S , Δ , and MS metrics (see Tables 5 and 6). In other words, the MOWCA was superior over the reported methods for the spacing, spread, and MS metrics.

Except of GD metric, the MOWCA has represented better results compared to those obtained by the MOPSO and NSGA-II. Moreover, note that the non-dominated solutions obtained by the MOWCA are distributed in the range between (0.5487, 0.1792) and (1.6393, 0.5199), whereas non-dominated solutions obtained by the NSGA-II are spread in the range of (0.5729, 0.1789) and (1.4673, 0.4196).

Figure 9 depicts how well the considered optimisation method could cover the $PF_{optimal}$. Looking at Figure 9c, it can be emphasised that the non-dominated solutions detected by the MOWCA have the highest accuracy with the best distribution for the FTSE experiment.

6.4. DAX100 (30 stocks)

Tables 7 and 8 tabulate obtained statistical results by the reported evaluators in this article for the DAX100 problem. By observing statistical results obtained by the GD metric, it indicates that the MOWCA has surpassed the others. In this regard, the MOPSO has shown the worst results in terms of GD metric, as can be seen in Table 7. Moreover, the MOWCA not only could reach the best GD value but also possesses the first rank in terms of having the minimum values for the metrics of spacing and spread (see Tables 7 and 8).

However, in terms of MS , the MOPSO has obtained the best MS metric value as can be seen in Table 8. In contrast, with respect to average and worst results for the MS metric, the MOWCA has been placed at the first rank among the considered optimisers.

Table 7. Statistical optimisation results obtained by the GD and S metrics for the DAX problem using different optimisers.

| Method | GD | | | | S | | | |
|---------|----------|----------|----------|----------|----------|----------|----------|----------|
| | Worst | Average | Best | SD | Worst | Average | Best | SD |
| NSGA-II | 3.89E-04 | 3.16E-04 | 2.30E-04 | 4.75E-05 | 4.99E-01 | 3.33E-01 | 2.32E-01 | 7.37E-02 |
| MOPSO | 2.90E-02 | 7.93E-03 | 2.11E-03 | 7.97E-03 | 3.51E-01 | 2.76E-01 | 2.41E-01 | 3.23E-02 |
| MOWCA | 6.31E-04 | 3.12E-04 | 1.84E-04 | 1.33E-04 | 3.76E-01 | 1.96E-01 | 1.39E-01 | 5.14E-02 |

Source: Data processing was performed off-line using a commercial software package MATLAB 8.5, The MathWorks Inc., Natick, MA, 2000.

Table 8. Statistical optimisation results obtained by the MS and Δ metrics for the DAX problem using different optimisers.

| Method | Δ | | | | MS | | | |
|---------|----------|----------|----------|----------|----------|----------|----------|----------|
| | Worst | Average | Best | SD | Worst | Average | Best | SD |
| NSGA-II | 5.60E-01 | 3.78E-01 | 2.60E-01 | 6.56E-02 | 7.68E-01 | 8.73E-01 | 9.56E-01 | 5.52E-02 |
| MOPSO | 6.16E-01 | 2.64E-01 | 2.65E-01 | 1.36E-01 | 5.08E-01 | 8.84E-01 | 9.79E-01 | 1.50E-01 |
| MOWCA | 2.59E-01 | 2.18E-01 | 1.92E-01 | 1.88E-02 | 9.06E-01 | 9.44E-01 | 9.68E-01 | 1.77E-02 |

Source: Data processing was performed off-line using a commercial software package MATLAB 8.5, The MathWorks Inc., Natick, MA, 2000.

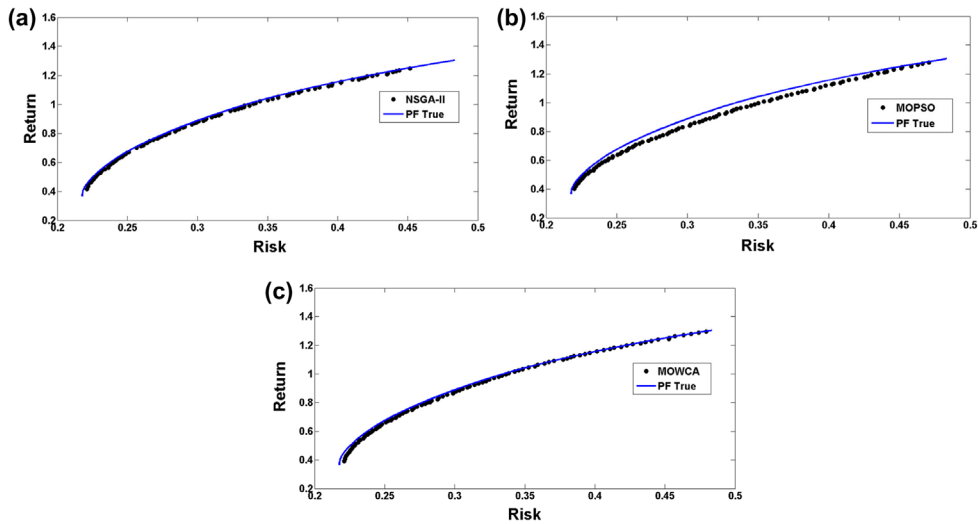


Figure 10. Generated Pareto front versus optimal (true) Pareto front (PF True) for solving DAX problem obtained by the: (a) NSGA-II; (b) MOPSO; (c) MOWCA. Source: Data processing was performed off-line using a commercial software package MATLAB 8.5, The MathWorks Inc., Natick, MA, 2000.

Talking about extreme non-dominated solutions, the MOWCA has found the minimum and maximum solutions of (0.3903, 0.2212) and (1.2965, 0.4793), whereas the minimum and maximum returns obtained by the NSGA-II are 0.4159 units with a risk of 0.2210 and 1.2482 units with a risk of 0.4517, respectively.

These observations, summarised in Tables 7 and 8, will be completed by the results presented in Figure 10. Looking at Figure 10c, one can conclude that the MOWCA was

successful in covering the optimal Pareto front, having satisfactory distribution and spread for the non-dominated solutions.

To sum up, it can be concluded that the MOWCA showed better performance in finding suitable non-dominated solutions sufficiently near to the $PF_{optimal}$. Not only the MOWCA could stand in the top rank with regard to GD metric, also, it was able to generate uniform distribution of non-dominated solutions. It is worth highlighting that as another contribution in this article, the applied optimisers have been implemented to analyse the daily data for the period January 2012 to December 2014 for the multi-objective portfolio optimisation problems. The results can lead to this conclusion which even in the real markets with a real number of stocks, the portfolio model with the usage of WCA is applicable and it can reach reasonable optimal solutions.

7. Conclusions and future studies

In this article, a portfolio selection model which is based on Markowitz's portfolio selection problem including the cardinality constraints is considered. The results can lead Markowitz's model to a more practical one. Thus, the proposed estimators facilitate the Markowitz M-V optimisation procedure, making it desirable, implementable, practically useful, and applicable in a variety of situations.

Moreover, this study proposes a cardinality constrained M-V model, and employed metaheuristic optimisation algorithms to solve the problem. It utilised an efficient multi-objective optimisation technique known as MOWCA. The fundamental concepts and ideas which underlie the method are inspired by nature and based on the water cycle process in nature. In this study, the MOWCA is represented for solving four constrained benchmark portfolio optimisation problems.

Furthermore, using the applied optimisers, computational results are attained for analyses of daily data for the period January 2012 to December 2014 including S&P100 in the US, Hang Seng in Hong Kong, FTSE100 in the UK, and DAX100 in Germany. The efficiency and performance of MOWCA were carried out using four well-known performance metrics (i.e., GD metric, metric of spacing, metric of spread, and metric of MS).

The statistical optimisation results attained by the performance metrics illustrated that the MOWCA was able to offer non-dominated solutions sufficiently near to the optimal Pareto front in addition to providing better optimal non-dominated solutions in comparison with NSGA-II and MOPSO. Note that the robustness and exploratory abilities of MOWCA depends on the nature and complexity of the problems. However, the optimisation results obtained in this article showed that the MOWCA can be considered as a proper alternative multi-objective optimiser, with a comparable degree of accuracy to find the optimal Pareto fronts for different scales of portfolio optimisation multi-objective problems.

For future studies some areas of research are proposed as follows: (1) considering other constraints of a real market such as transaction costs; (2) studying other periods such as crisis periods (2007–2009) and comparing the results with the stable growth period (2010–2016) results; (3) selecting the samples from other countries and markets; and (4) utilising other risk measures such as semi-variance, mean absolute deviation, and variance with skewness for investigating the efficiency of MOWCA.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Ali Sadollah  <http://orcid.org/0000-0002-7782-4126>

References

- Ali, M. M., & Kaelo, P. (2008). Improved particle swarm algorithms for global optimization. *Applied Mathematics and Computation*, 196, 578–593.
- Bermúdez, J. D., Segura, J. V., & Vercher, E. (2012). A multi-objective genetic algorithm for cardinality constrained fuzzy portfolio selection. *Fuzzy Sets and Systems*, 188, 16–26.
- Chang, T.-J., Meade, N., Beasley, J. E., & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimization. *Computers & Operations Research*, 27, 1271–1302.
- Chang, T.-J., Sang-Chin, Y., & Chang, K.-J. (2009). Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications*, 36, 10529–10537.
- Chen, W. (2015). Artificial bee colony algorithm for constrained possibilistic portfolio optimization problem. *Physica A*, 429, 125–139.
- Coello, C. A. C. (2000). An updated survey of GA-based multi-objective optimization techniques. *ACM Computing Surveys*, 32, 109–143.
- Corazza, M., Fasano, G., & Gusso, R. (2013). Particle swarm optimization with non-smooth penalty reformulation, for a complex portfolio selection problem. *Applied Mathematics Computing*, 224, 611–624.
- Crama, Y., & Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research*, 150, 546–571.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester: Wiley.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6, 182–197.
- 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, 4558–4566.
- Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, 110–111, 151–166.
- Goldberg, D. (1989). *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley.
- Golmakani, H. R., & Fazel, M. (2011). Constrained portfolio selection using particle swarm optimization. *Expert Systems with Applications*, 38, 8327–8335.
- Guney, K., & Basbug, S. (2014). A quantized water cycle optimization algorithm for antenna array synthesis by using digital phase shifters. *International Journal of RF and Microwave CE*, 25, 21–29.
- Haddad, O., Moravej, M., & Loáiciga, H. (2014). Application of the water cycle algorithm to the optimal operation of reservoir systems. *Journal of Irrigation and Drainage E-ASCE*, 141, 04014064.
- Haupt, R. L., & Haupt, S. E. (2004). *Practical genetic algorithms* (2nd ed.). USA: John Wiley & Sons Inc.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Michigan: University of Michigan Press.
- Jabbar, A., & Zainudin, S. (2014). Water cycle algorithm for attribute reduction problems in rough set theory. *JATIT*, 61, 107–117.
- Jiang, Y., Hu, T., Huang, C., & Wu, X. (2007). An improved particle swarm optimization algorithm. *Applied Mathematics and Computation*, 193, 231–239.
- Kaveh, A., & Laknejadi, K. (2011). A novel hybrid charge system search and particle swarm optimization method for multi-objective optimization. *Expert Systems with Applications*, 20, 1–14.

- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization, in Proc. of the IEEE international conference on neural networks, Perth, Australia, 1942–1948.
- Kennedy, J., Eberhart, R. C., & Shi, Y. (2001). *Swarm intelligence*. San Francisco, CA: Kaufmann.
- Kolm, P. N., Tütüncü, R., & Fabozzi, F. J. (2013). 60 Years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research*, 236, 258–267.
- Kyong, J. O., Tae, Y. K., & Sungky, M. (2005). Using genetic algorithm to support portfolio optimization for index fund management. *Expert Systems with Applications*, 28, 371–379.
- Lenin, K., Reddy, B. R., & Kalavathi, M. S. (2014). Water cycle algorithm for solving optimal reactive power dispatch problem. *Journal of Engineering Technology and Research*, 2(2), 1–11.
- Li, X. (2003, July). A non-dominated sorting particle swarm optimizer for multiobjective optimization. In E. Cantu-Paz, J. A. Foster, K. Deb, L. D. Davis, R. Roy, U.-M. O'Reilly, ... A. C. Schultz (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2003)* (Vol. 2723, pp. 37–48). Madrid: Springer, Lecture Notes in Computer Science.
- Li, J., & Xu, J. (2007). A class of possibilistic portfolio selection model with interval coefficients and its application. *Fuzzy Optimization and Decision Making*, 6, 123–137.
- Lin, C.-C., & Liu, Y.-T. (2008). Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operational Research*, 185, 393–404.
- Liu, Y., Wu, X., & Hao, F. (2012). A new chance-variance optimization criterion for portfolio selection in uncertain decision systems. *Expert Systems with Applications*, 39, 6514–6526.
- Maringer, D., & Kellerer, H. (2003). Optimization of cardinality constrained portfolios with a hybrid local search algorithm. *OR Spectrum*, 25, 481–495.
- Markowitz, H. M. (1952). Portfolio selection *Journal of Finance*, 7, 77–91.
- Markowitz, H. M. (1991). *Portfolio selection: Efficient diversification of investments*. New York, NY: Yale University Press, John Wiley.
- Najafi, A. A., & Mushakhian, S. (2015). Multi-stage stochastic mean-semivariance-CVaR portfolio optimization under transaction costs. *Applied Mathematics Computing*, 256, 445–458.
- Ruiz-Torrubiano, R., & Suarez, A. (2010). Hybrid approaches and dimensionality reduction for portfolio selection with cardinality constraints. *IEEE Computational Intelligence Magazine*, 5, 92–107.
- Sadollah, A., Eskandar, H., Kim, J. H., & Bahreininejad, A. (2014). Water cycle algorithm for solving multi-objective optimization problems. *Soft Computing*, 19, 2587–2603. doi:10.1007/s00500-014-1424-4
- Sadollah, A., Eskandar, H., & Kim, J. H. (2015). Water cycle algorithm for solving constrained multi-objective optimization problems. *Applied Soft Computing*, 27, 279–298.
- Schott, J. R. (1995). *Fault tolerant design using single and multicriteria genetic algorithm optimization* (Master's thesis), Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA.
- 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, 5058–5063.
- Tsai, S.-J., Sun, T.-Y., Liu, C.-C., Hsieh, S.-T., Wu, W.-C., & Chiu, S.-Y. (2010). An improved multi-objective particle swarm optimizer for multi-objective problems. *Expert Systems with Applications*, 37, 5872–5886.
- Veldhuizen, D. A. V., & Lamont, G. B. (1998). *Multi-objective evolutionary algorithm research: A history and analysis*. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH.
- Woodside-Oriakhi, M., Lucas, C., & Beasley, J. E. (2011). Heuristic algorithms for the cardinality constrained efficient frontier. *European Journal of Operational Research*, 213, 538–550.
- Yang, X., Yuan, J., Yuan, J., & Mao, H. (2007). A modified particle swarm optimizer with dynamic adaptation. *Applied Mathematics and Computation*, 189, 1205–1213.
- Zhu, H., Wang, Y., Wang, K., & Chen, Y. (2011). Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem. *Expert Systems with Applications*, 38, 10161–10169.