International Journal of Automation and Computing 05(1), January 2008, 67-80
DOI: 10.1007/s11633-008-0067-2

Evolutionary Multi-objective Portfolio Optimization in Practical Context

S. C. Chiam* K. C. Tan A. Al Mamum

Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117576, Singapore

Abstract: This paper addresses evolutionary multi-objective portfolio optimization in the practical context by incorporating realistic constraints into the problem model and preference criterion into the optimization search process. The former is essential to enhance the realism of the classical mean-variance model proposed by Harry Markowitz, since portfolio managers often face a number of realistic constraints arising from business and industry regulations, while the latter reflects the fact that portfolio managers are ultimately interested in specific regions or points along the efficient frontier during the actual execution of their investment orders. For the former, this paper proposes an order-based representation that can be easily extended to handle various realistic constraints like floor and ceiling constraints and cardinality constraint. An experimental study, based on benchmark problems obtained from the OR-library, demonstrates its capability to attain a better approximation of the efficient frontier in terms of proximity and diversity with respect to other conventional representations. The experimental results also illustrated its viability and practicality in handling the various realistic constraints. A simple strategy to incorporate preferences into the multi-objective optimization process is highlighted and the experimental study demonstrates its capability in driving the evolutionary search towards specific regions of the efficient frontier.

Keywords: Evolutionary computation, multi-objective optimization, portfolio optimization, preference-based multi-objective optimization, constraint handling.

1 Introduction

The allocation of limited capital to the different financial assets available is one of the paramount problems in financial management. Normally, the decision will be based on some forms of quantitative measurement, most typically the expected return of the portfolio and its associated risk, i.e., return variance. Intuitively, for a given level of return objective, portfolio managers will seek to reduce the risk as much as possible. An optimal portfolio is one that has the maximum return with the minimum risk and the set of all the optimal portfolios will form the efficient frontier. Essentially, the ultimate objective in portfolio optimization is to balance the expected risk and return via diversification and obtain the efficient frontier under various practical constraints arising from business and industry regulations.

In the construction of a portfolio, portfolio managers must select both the type of assets, as well as its quantity (proportion or units). This combinatorial optimization problem has a highly complex search space due to the abundant choices of financial assets available. Thus, portfolio optimization continues to pose a challenge for efficient optimization techniques. Although many computational techniques have been developed for this purpose, most of them are single objective approaches, even though this problem clearly consists of two conflicting objectives. However, there are an increasing number of multi-objective approaches being developed, particularly multi-objective evolutionary algorithms (MOEA)[1]. The main advantage of evolutionary multi-objective portfolio optimization (EMOPO), i.e., the application of MOEA for portfolio optimization, is that an estimation of the efficient risk-return frontier can be obtained in a single run as opposed to the multiple runs needed in the case of single objective approaches.

Manuscript received August 2, 2007; revised October 30, 2007
*Corresponding author. E-mail address: g0500055@nus.edu.sg

Most of the early works on EMOPO adopt the unconstrained Markowitz mean-variance model[2−4], which is impractical in the real world of investment management, as portfolio managers often face a number of realistic constraints arising from business regulations, practical matters, and industry regulations[5]. While subsequent research did incorporate them into the optimization model, in-depth analysis that examined how these constraints affect the evolutionary search progress and the efficient frontier attainable are sorely lacking.

This paper aims to consider a more realistic model of the portfolio optimization problem by considering floor and ceiling constraint and cardinality constraint, and analyze their effects on the efficient frontier attainable. For this purpose, an order-based representation that can be easily extended to handle these constraints will be proposed. Furthermore, this paper will improve the current experimental platform for EMOPO by introducing diversity measures and statistical analysis used typically in the performance assessment of multi-objective optimizers. Lastly, there has been an increased interest in incorporating decision makers′ preferences in the evolutionary search process of multi-objective optimization[6] recently. This is especially relevant in portfolio optimization as well where portfolio managers are only interested in specific regions or points along the efficient frontier during the actual execution of their investment order. Such techniques will be explored and evaluated in the context of EMOPO.

The remainder of the paper is structured as follows. The paper will start with a formal introduction of the basic portfolio optimization model and the various practical constraints available. This will be followed by a description of the proposed evolutionary platform, focusing on the order-based representation and its corresponding constraint handling techniques. Subsequently, experimental results for both the unconstrained and constrained problem will be analyzed to evaluate the viability and practicality of the proposed algorithmic model. The last part of the paper will investigate the effects of incorporating preference in EMOPO before the conclusions are drawn in the last section.

2 Portfolio optimization

The foundation of portfolio optimization was laid by Markowitz[7], where he proposed a mean-variance optimization model for designing an optimum portfolio based on the idea of minimizing the risk for some expected return. The following equations give an outline of the model.

$$\min\ F_1 = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij} \tag{1}$$

$$\max\ F_2 = \sum_{i=1}^{N} w_i \mu_i \tag{2}$$

subjected to

$$\sum_{i=1}^{N} w_i = 1 \tag{3}$$

$$0 \le w_i \le 1,\ i = 1, \cdots, N \tag{4}$$

where N is the number of assets available, μ_i is the expected return of asset i, σ_{ij} represents the covariance between assets i and j, and w_i is the decision variable denoting the composition of asset i in the portfolio as a proportion of the total available capital. Equation (3) gives the budget constraint for a feasible portfolio, while (4) requires all investment to be positive, i.e., no short selling is allowed. The goal in portfolio optimization is to find portfolios amongst the N assets that can simultaneously satisfy the two conflicting objectives, i.e., minimize the total variance (1), denoting the risk associated with the portfolio, while maximizing its profits (2).

Related literatures in EMOPO have extended the mean-variance optimization model by modifying the existing objective functions. Particularly, Arnone et al.[3] and Loraschi et al.[4] considered downside risk (i.e., distribution of the downside returns) in place of the return variance (1). Alternatively, additional objective functions have been incorporated to enhance the original model. To handle the cardinality constraints, Fieldsend et al.[8] considered the cardinal as an additional objective to be optimized. This approach allows the direct extraction of the 2-dimensional cardinality constrained frontier for any particular cardinality. Other additional objectives considered in literature include surplus variance[9], portfolio value at risk[9], annual dividend[10], and asset ranking[10]. Nevertheless, most related literature improved the realism of the mean-variance model by incorporating realistic constraints encountered in actual practice.

2.1 Practical constraints

In the real world of investment management, portfolio managers often face a number of realistic constraints arising from business regulations, practical matters, and industry regulations[5]. Examples of such realistic constraints include floor and ceiling constraint, cardinality constraint, round-lot constraint, turnover constraint, trading constraint, buy-in threshold and transaction cost inclusion[11]. The first three constraints will be considered in this paper. However, due to the limitations of the problem set, round-lot constraint will not be included in the experimental study though the corresponding constraint handling technique will be mentioned. The remaining constraints will be considered in future work.

The floor and ceiling constraint specifies the lowest and highest limits on the proportion of each asset that can be held in a single portfolio. The former prevents excessive administrative costs for very small holdings, which have negligible influence on the performance of the portfolio, while the latter rules out excessive exposure to any one portfolio constituent as part of institutional diversification policy. This constraint is formulated as

$$a_i \le w_i \le b_i, \quad 0 \le a_i \le b_i \le 1 \tag{5}$$

where a_i and b_i denote respectively the minimum and maximum weights that can be held for asset $i\,(i = 1, \cdots, N)$. While floor constraint has been actively studied in [5, 10, 12–17], the general floor and ceiling constraint has been less explored.

Cardinality constraint specifies the maximum and minimum number of assets that a portfolio can hold due to monitoring, diversification or transaction cost control reasons. It can be expressed as follows:

$$C_l \le \sum_{i=1}^{N} z_i \le C_u \tag{6}$$

where $z_i = 1$ if $w_i > 0$, and otherwise, $z_i = 0$. This constraint has been simplified in several related works, where either the inequality restriction in (6) is replaced by an equality restriction instead, i.e., portfolios are restricted to a particular fixed value of cardinality[10−16,18] or only the maximum cardinality constraint is considered[5,17,19].

Round-lot constraint requires the number of any asset included in the portfolio to be in exact multiples of the normal trading lots[5,20]. The round-lot constraint can be expressed as

$$w_i = \frac{c_i y_i}{C} \tag{7}$$

where C is the total capital budget, c_i is the purchasing price for the minimum lot of asset i, and $y_i \in \mathrm{Z}$ denotes the number of lots purchased for asset i. The inclusion of round-lot constraint most likely will require a relaxation of the budget constraint as the total capital might not be of exact multiples of the minimum lot prices for the various assets. Also, since the round-lot constraint requires the exact definition of the available capital and the minimum lot prices for each asset, this constraint will not be considered in the experimental study as the latter is not available in the test problems studied.

2.2 Preferences in portfolio optimization

Essentially, the optimization problem is to find portfolios amongst the N assets that can satisfy the two objectives of return (2) and risk (1) simultaneously. Thus, the optimal portfolio is one that has the maximum return with the minimum risk and the set of all the optimal portfolios will form the efficient frontier illustrating the tradeoff between the conflicting objectives, as represented by FF in Fig. 1.

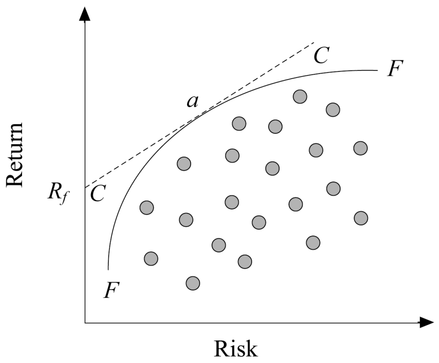


Fig. 1 Efficient frontier (FF) illustrating the tradeoff between return and risk for various portfolios of assets (Suboptimal portfolios are denoted in gray; the line CC and point a denote the capital market line and efficient portfolio, respectively.)

Even though knowledge of the efficient frontier is important, portfolio managers are only interested in specific regions or points along the efficient frontier in practical situations. However, there have been limited developments in the incorporation of preferences in EMOPO despite its increasing popularity in the general field of multi-objective optimization. The only available work in literature was proposed by Subbu et al.[9], where they augmented the evolutionary search with a target objective genetic algorithm, which is a non-Pareto and non-aggregating function approach that allows solutions that are as close as possible to a pre-defined target for one or more criterions to be found. The limitations of this approach are that the decision maker should have enough domain knowledge to select good combination of objectives and determine scaling factors for each objective.

In the context of portfolio optimization, portfolio managers are often interested in a particular point on the efficient frontier known as the efficient portfolio as described in the capital asset pricing model (CAPM). The key principles underlying CAPM are illustrated in Fig. 1. The point, R_f represents the risk-free return available in the market to an individual, for example through short-term government treasury bills. Line CC denotes the capital market line, which is a straight line that passes through R_f and is tangential to the efficient frontier FF. The point, a, at the intersection of CC and FF is the efficient portfolio. The significance of the efficient portfolio is that any combination of it and the risk-free asset, attainable by either lending or borrowing at the rate of R_f, will allow the individual to operate at any point on the capital market line, above the efficient frontier, resulting in higher return for any given amount of risk than any optimal portfolio on FF.

Mathematically, the efficient portfolio is the point on the efficient frontier that can maximize the objective function (8).

$$\max\ F_3 = \frac{(F_2 - R_f)}{F_1} \tag{8}$$

where R_f is the risk-free rate. This fitness measure is more commonly known as the Sharpe ratio, a well-known financial indicator that calculates the risk-adjusted return for an asset/portfolio. The latter part of the paper will discuss how the incorporation of preference-based techniques could drive the evolutionary search towards the efficient portfolio.

3 Evolutionary algorithm

The incorporation of constraints to improve the realism of the portfolio optimization problem has obsoleted classical optimization techniques[1] and motivated the development and application of meta-heuristics techniques like evolutionary algorithms, ant colony optimization[16,21], particle swarm optimization[22] and etc. Amongst them, MOEA is the more popular approach[1] due to its ability in solving complex multi-objective optimization problems with respect to the proximity and diversity goals. Based on basic concepts from the biological model of evolution, the search dynamic of MOEA is driven by biologically inspired evolutionary operators like selection, crossover and mutation, which will explore and exploit the associated search space for the optimal solution. The crossover and mutation operator manipulate and create potential solutions, while the selection operator provides the necessary convergence pressure. When extending MOEA for portfolio optimization, several issues need to be taken into consideration, namely representation, variation operator and constraint handling techniques.

3.1 Representation

MOEA maintains a population of chromosome, where each of them represents a potential solution to the optimization problem, which in the context of EMOPO is a portfolio of assets. In the related literature, different types of representation have been proposed. The most direct representation is to use a real-number vector that denotes the weight composition of the various assets in the portfolio[23]. Before the fitness evaluation, the total weight is normalized to one to satisfy the budget constraint (3).

However, better algorithmic performance can be obtained, if a problem specific representation is adopted instead. Streichert et. al.[15] observed that the optimal portfolio normally comprised of only a limited number of the available assets. Thus, a hybrid representation was proposed, where an additional binary string is included to reflect the existence of the assets in the portfolio. Such a scheme facilitates the removal and adding of assets to portfolios, resulting in smaller portfolios generally. This representation has been popular in [5, 14, 15, 17]. Alternatively, the weight vector can just comprise a few assets that are randomly chosen prior to the algorithmic run[16,24]. This approach provides a simple solution to the fixed cardinality constraint that limits the portfolio size to a particular value.

Contrary to previous works, this paper proposes an order-based representation for EMOPO. Each chromosome comprises of two vectors, i.e., a real number vector and an integer vector that contains the identity tags of the various assets available, denoting their corresponding weights. Fig. 2 shows an instance of this representation for a problem with eight assets available.

Asset vector	7	2	5	1	3	8	6	4
Weight vector	0.54	0.25	0.85	0.14	0.05	0.67	0.55	0.40

Fig. 2 A chromosomal instance for the ordered-based representation proposed based on eight assets available

To find the portfolio associated with this chromosome, an empty portfolio will first be initialized and assets will then be added to it, as per their order in the asset vector. This procedure will terminate once the total weight of the portfolio exceed one or when all the available assets are in the portfolio. The total weights for the assets in the portfolio will then be normalized to one to satisfy the budget constraint. After which, the returns and risk can be calculated to determine its optimality. Fig. 3 illustrates the fitness evaluation procedures for the chromosome in Fig. 2. In Fig. 3, the assets are iteratively added into the portfolio until the total weights exceed one. The various weights in the portfolio are then normalized to one to satisfy the budget constraint.

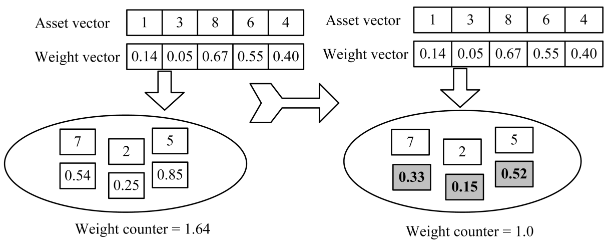


Fig. 3 Fitness evaluation for the chromosome

As each asset is added iteratively into the portfolio for the order-based representation, direct monitoring and control of the weight values for each asset is possible at their point of inclusion. Thus, constraint handling techniques can be executed instantaneously to repair any infeasibility. Further details of the constraint handling techniques will be furnished in the later section.

The chromosome initialization process involves randomly permuting the order of the asset vector and generating the weights from certain probability distribution. The most direct approach is to adopt the uniform distribution ranging from 0 to 1 so as to satisfy constraint (4) directly. However, this will correspond to a mean weight of 0.5, which implies that only around 2 to 3 assets are required to fill up an empty portfolio. Thus, simple implementation of this representation will tend to generate small portfolio. Specifically, the average portfolio size for 100 000 randomly generated chromosomes is around 2.7 with a standard deviation of 0.9.

Of course, the average portfolio size can be increased by setting a maximum limit for the various weight values during initialization, as this will increase the assets required to fill up the portfolio. For example, imposing a maximum weight limit of 0.1 will increase the average portfolio size to 20.7 with a standard deviation of 2.6. However, this simple initialization strategy might not be able to improve the diversity of the initial population by much. Fig. 4 plots the average portfolio size attained for various maximum weight limits, i.e., {5.0, 2.0, 1.0, 0.5, 0.2, 0.1, 0.05, 0.01}. Clearly, although a smaller maximum weight limit will result in larger portfolio size, the diversity of the population remained unaffected.

Thus, an alternative initialization technique is suggested here. Different maximum weight limits will be assigned to the various chromosomes during the initialization process. This will arbitrarily enhance the diversity of the initial population, as clearly evident in Fig. 4.

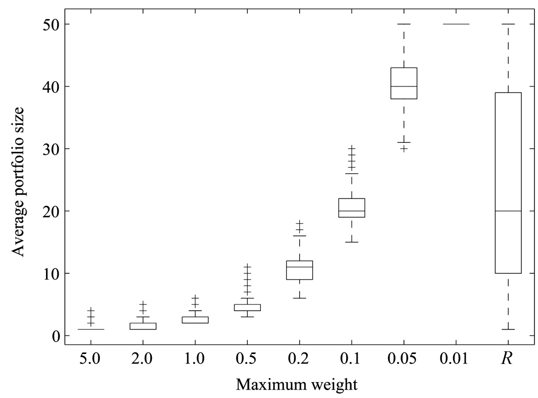


Fig. 4 Average portfolio size (maximum 50) for 100 000 randomly generated chromosomes with different weight limits ("R" denotes the case where each chromosome is assigned a different limit.)

3.2 Variation operator

Since conventional crossover or mutation operators are not suitable for the data structure of this order-based representation, different variation operators have to be designed. The proposed crossover operation is illustrated in Fig. 5. In Fig. 5, genes after the crossover point are swapped between the two parent chromosomes. Given two parent chromosomes, a crossover point will be randomly selected. Each chromosome will retain their original value before the crossover point and the remaining values after it will be rearranged in accordance with the order in the other chromosome, i.e., the values {5, 2, 7} in chromosome 1 are rearranged to {2, 5, 7}, in the order where these three values appear in chromosome 2. The corresponding weight vector will also be reshuffled accordingly.

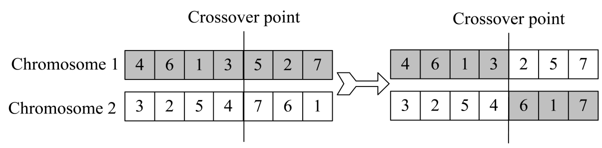


Fig. 5 Single-point crossover

However, not all the assets are included in the portfolio as illustrated in Fig. 3. Thus, neutral variation[25] (where redundancy in the genotype nullifies the effects of variation) might arise if the crossover point is selected amongst those irrelevant assets. Thus, the crossover point should be randomly selected amongst the assets considered in the portfolio. Specifically, the crossover point will be chosen within the mean portfolio sizes for the two parent chromosomes.

The mutation operation is just a simple procedure of swapping the asset and weights of two randomly selected alleles in a single chromosome, as illustrated in Fig. 6. In Fig. 6, the position of randomly chosen genes are swapped. Again to prevent neutral variation, it should be ensured that at least one of the selected assets should be within the portfolio. Both the variation operations discussed earlier are typically used for order-based representation.

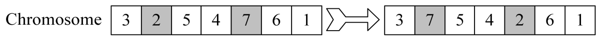


Fig. 6 Bit-swap mutation

3.3 Constraint handling techniques

The main advantage of this representation is that it can easily be extended to handle the various realistic constraints in portfolio optimization. This section will discuss the corresponding constraint handling techniques.

3.3.1 Floor and ceiling constraint

This constraint requires weight values to be within a specific range. Thus, the conventional strategy of normalizing the total weight to one so as to meet the budget constraint is no longer applicable here, since the normalized weights might not be within the limits. Related works[10−17] focus only on floor constraint and the conventional approach is to arbitrarily add the minimum weight to any infeasible assets.

A simple technique is proposed here to handle the general floor and ceiling constraints, which involves modifying the fitness evaluation operation while maintaining the same representation and variation operation and other evolutionary operators. The modified fitness evaluation will still initialize with an empty portfolio where asset is being added iteratively. The representation nature allows direct control on the manner in which the assets are introduced into the portfolio and any infeasibility can be immediately repaired. Thus, the floor and ceiling constraints are regarded as hard constraints by ensuring that the various weight values are adjusted to the floor and ceiling constraint as shown in (9).

$$w_i^{'} = a_i + (b_i - a_i) \cdot w_i. \tag{9}$$

Subsequently, assets will be added to the portfolio until the total weight of the portfolio exceeds one. At this stage, case-dependent correction techniques will be applied to ensure the feasibility of the final portfolio constructed. A total of 3 different cases have been identified:

1) After removing the last added asset, the remaining weight is between the floor and ceiling limits. In this case, the weight of the last added asset can simply be reassigned so that its adjusted weight is equivalent to the remainder needed to attain a total weight of one.

2) After removing the last added asset, the remaining weight is less than the floor limits. This case can be further subdivided into two different scenarios.

i) After removing the second last added asset, the remaining weight is between the floor and ceiling limits. This will mean that its weight can simply be reassigned so that the adjusted weight is equivalent to the remainder that it needed to attain a weight counter of one. The portfolio will contain all the assets considered so far and the adjusted asset.

ii) After removing the second last added asset, the remaining weight is outside the floor and ceiling limits. For this case, all the weight vectors will simply be readjusted by either increasing or decreasing them by a predefined percentage.

This modified fitness evaluation will ensure that all the solutions generated during the evolutionary search progress will always be feasible with respect to this constraint.

3.3.2 Cardinality constraint

Related works in literature have considered cardinality constraint as a hard constraint and generalized the inequality restriction by an equality constraint[11−16,18]. Thus, a fixed number of assets are arbitrarily selected based on the fixed cardinal value before the weights are normalized to satisfy the other constraints. Similar techniques can be employed to satisfy the maximal cardinality constraints[5,17,19] by setting the weights of excess assets to be zero. However, such techniques might have difficulties dealing with ceiling constraints as the excess weights cannot be arbitrarily assigned to other assets.

This paper considers the general cardinality constraint and regards it as a soft constraint instead. The repair operation described as below, is used to correct the feasibility of the chromosome.

```
If number of asset > maximum cardinal
    Increase all weights by k %
else If number of asset < minimum cardinal
    Decrease all weights by k %
end If
```

Specifically, the various values in the weight vector will be increased/decreased when its associated portfolio size is too high/low, so that fewer/more assets will be required in the re-evaluation. This simple procedure will help to adjust the portfolio size of infeasible chromosomes back to the feasible range.

Because of the presence of infeasible solutions in the evolving population, the selection operation will also have to factor this into consideration. That is, the feasibility of the portfolio will take priority over the optimality of the solutions. This is applicable for both the parent selection and the survivor selection.

3.3.3 Round-lot constraint

Because of the representation nature, round-lot constraints can be easily handled in a similar fashion as in the case of floor and ceiling constraints. Essentially, for every asset that is added into the portfolio, they will first be adjusted based on the floor and ceiling constraints. Following that, they will be rounded down to the largest weight available (10). Similar to techniques proposed by Skolpadungket et al.[5], the remainder of the budget will be allocated to the assets in the existing portfolio provided that the ceiling constraint is not satisfied, and to the assets outside the portfolio if the floor constraint can be satisfied.

$$w_i^{''} = w_i^{'} - w_i^{'} \cdot \mathrm{mod}\left(\frac{c_i}{C}\right). \tag{10}$$

4 Experimental setup

To evaluate the performance of the proposed order-based representation, it will be applied to a set of portfolio optimization problems obtained from the OR-library[26]. These problems contain the estimated returns and the covariance matrix for groups of assets in different stock market indices. Their details are summarized in Table 1. The difficulty of these problems is directly related to the number of assets available.

Table 1 Description of experimental data sets

Problem index	Data source	Number of assets
PORT1	Hong Kong, Hang Seng	31
PORT2	German, DAX 100	85
PORT3	British FTSE 100	89
PORT4	US S&P 100	98
PORT5	Japanese Nikkei 225	225

The evolutionary platform adopted was a generic elitist MOEA that maintained a fixed-size population and an archive to store the best solution discovered. Both the population and the archive are assigned a size of 100 each. The order-based representation proposed was adopted and the length of each chromosome depends on the number of assets available in each problem. In each generation, mating individuals were selected via binary tournament from the combined population of the existing evolved solutions and archive. The selection criterion is based on Pareto dominance. In the event of a tie, the niche count will be employed. Specifically, a niche radius of 0.01 in the normalized objective space was considered. The mechanism of niche sharing is used in the tournament selection as well as diversity maintenance in the archive. The mating individuals would subsequently undergo variation operation (i.e., crossover probability of 0.8 and bit-wise mutation of $1/N$) to produce offspring for the next generation. The generational stopping criteria were varied for each problem based on their level of difficulty. Specifically, each problem was run sufficiently until their performance can be properly differentiated.

Unlike single-objective optimization, there are several goals in multi-objective optimization[27, 28], most notably proximity and diversity. The former describes the accuracy of the solution set while the latter measures how well the solution set is defined. Despite so, most of the experimental studies in EMOPO do not involve diversity measures and statistical analysis that are commonly used in the performance assessment of multi-objective optimizers. While generational distance[5] and average relative distance to the efficient frontier[17,24] have been used on separate occasions, other related works merely portray the efficient frontier attained[9,16,20].

In this paper, a set of proximity and diversity measures will be adopted that is commonly used in multi-objective optimization. The generational distance metric, GD, is used to measure proximity. It quantifies how "far" the approximation of the efficient frontier found (EF_{known}) is from the actual efficient frontier[29,30] and is defined as

$$GD = \sqrt{\left(\frac{1}{m}\sum_{i=1}^{m} d_i^2\right)} \tag{11}$$

where m is the number of solutions found, d_i is the Euclidean distance (in objective space) between the member i in EF_{known} and its nearest member of the efficient frontier. A low value of GD signifies that EF_{known} is very close to the efficient frontier.

As for diversity, it depends on factors like the spread and spacing of the solution set. The former can be measured by the maximum spread, MS metric[28] which measures how well the efficient frontier is covered by EF_{known} through the hyper-boxes formed by the extreme function values observed in both fronts. To normalize the metric, this metric is modified as

$$MS = \sqrt{\frac{1}{L}\sum_{l=1}^{L}\left[\frac{\left(\max\limits_{1\le i\le m} f_l^i - \min\limits_{1\le i\le m} f_l^i\right)}{(F_l^{\max} - F_l^{\min})}\right]^2} \tag{12}$$

where f_l^i is the l-th objective of member i, $F_l^{\max}$ and $F_l^{\min}$ are the maximum and minimum of the l-th objective in EF_{known}. The greater the value of MS is, the more the area of the efficient frontier is covered by EF_{known}. For the latter, the metric of spacing, S, which measures how "evenly" solutions in EF_{known} are distributed is chosen. It is defined as

$$S = \frac{\sqrt{\frac{1}{m}\sum_{i=1}^{m}(d_i - \overline{d})^2}}{\overline{d}} \tag{13}$$

where d_i is the Euclidean distance (in objective space) between the member i and its nearest member in EF_{known}. S will be low if the members in EF_{known} are evenly distributed.

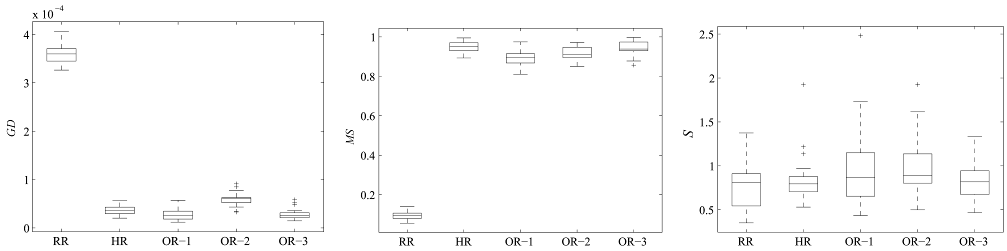
5 Experimental results and discussion about unconstrained portfolio optimization

In this section, the unconstrained portfolio optimization model (1)-(4) will be considered to analyze the performance of the proposed evolutionary model. The algorithmic performance of the order-based representation will be compared with two other representations, namely real vector representation[23] and hybrid representation[15]. The various algorithm configurations are described in Table 2. Prior investigations have revealed that uniform crossover resulted in better algorithmic performance for these representations. As for the proposed order-based representation, the different initialization techniques mentioned earlier will be considered.

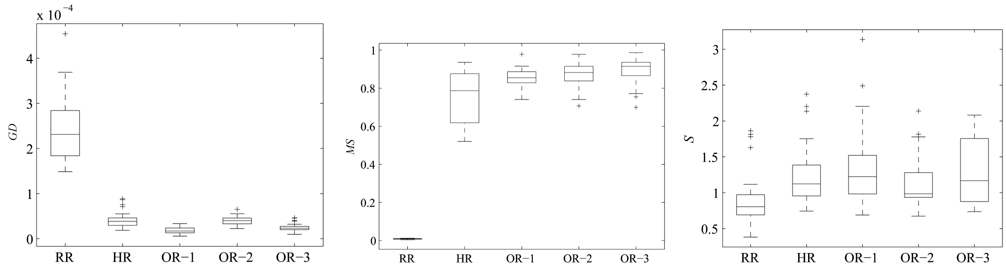
Table 2 Description of the various algorithm configurations in the experimental study for unconstrained portfolio optimization

Algorithm configurations	Notations
Real number representation with uniform crossover	RR
Hybrid representation with uniform crossover	HR
Order-based representation without initialization limit	OR-1
Order-based representation with initialization limit of 0.1	OR-2
Order-based representation with initialization technique	OR-3

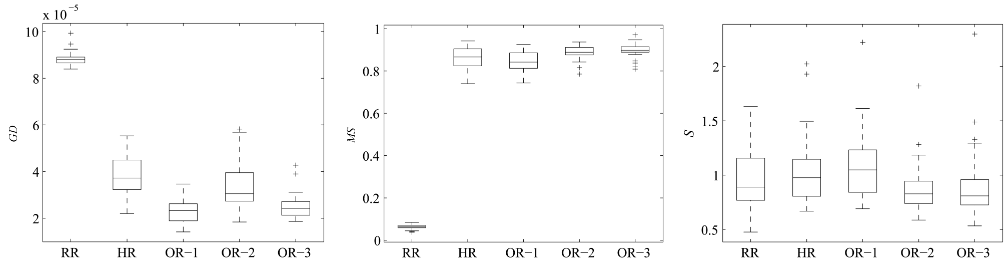
30 independent simulation runs were performed for all experiments and the same random seed was assigned to each set of the runs so that all algorithms start with the same



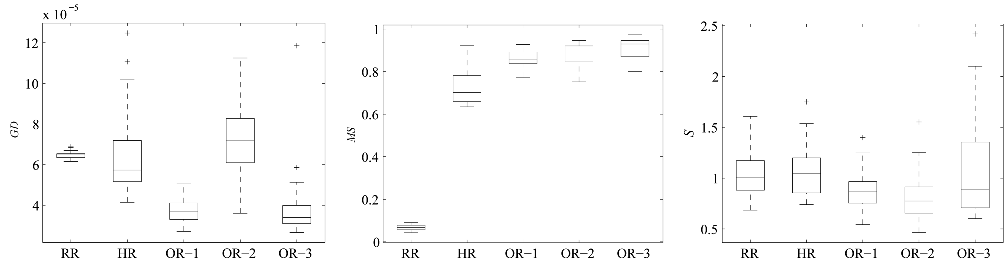
(a) PORT1 (100)



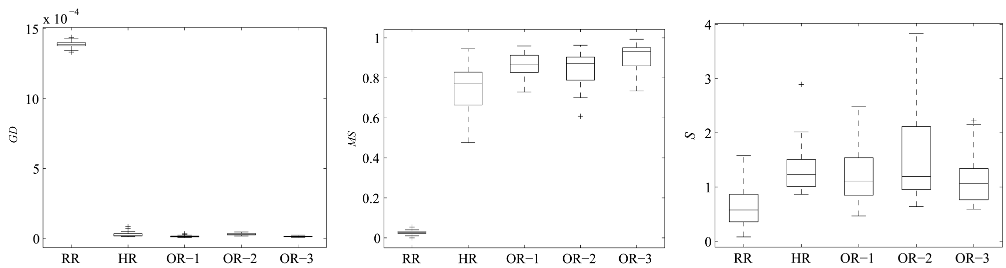
(b) PORT2 (300)



(c) PORT3 (300)



(d) PORT4 (300)



(e) PORT5 (500)

Fig. 7 GD, MS, and S obtained under the different algorithms for the different problems with varying stopping criteria

initial population. The simulation results are illustrated by box plots to provide a statistical comparison of the performances for the various algorithms. Since a mere difference in the average of the qualitative metrics cannot be blindly regarded as performance difference between the algorithms, statistical test, namely the analysis of variance (ANOVA), is used to examine the significance of the mean difference between the various results.

The various performance metrics are illustrated in Fig. 7. The performance of RR was significantly poorer than the rest, especially in terms of diversity of the EF_{known} attained, as reflected by their low values of MS. This was due to the nature of the representation, which favored large portfolio sizes that were near to N, as verified by Table 3. Then, EF_{known} for RR was limited to the region of low return and risk due to excessive diversification, and thus failed to cover the entire efficient frontier. Fig. 8 shows the Pareto front obtained by RR in PORT4.

Table 3 The average portfolio size and its corresponding standard deviation for the various solutions attained by the various algorithms in the different problems

	RR	HR	OR-1	OR-2	OR-2
PORT1	30.99	4.74	3.30	4.38	3.52
	(0.031)	(0.48)	(0.35)	(0.62)	(0.28)
PORT2	84.98	15.68	4.64	9.77	7.21
	(0.043)	(2.67)	(0.82)	(1.08)	(1.50)
PORT3	88.98	16.84	4.27	9.23	7.02
	(0.042)	(2.97)	(0.74)	(1.67)	(0.91)
PORT4	97.99	24.09	6.15	13.60	12.08
	(0.0264)	(3.08)	(0.63)	(1.23)	(2.30)
PORT5	224.95	65.81	4.48	8.73	5.65
	(0.089)	(11.16)	(0.57)	(1.19)	(1.24)

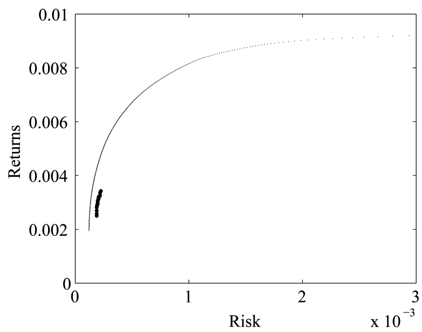
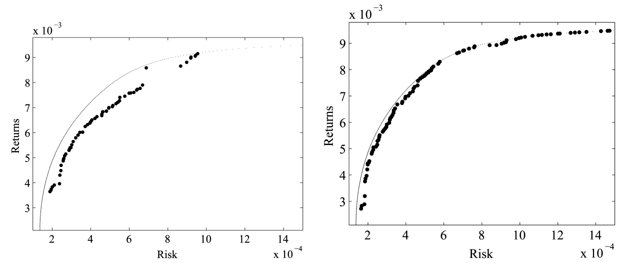


Fig. 8 EF_{known} of PORT4 obtained by RR in one of the algorithmic runs (The dotted-line denotes the efficient frontier.)

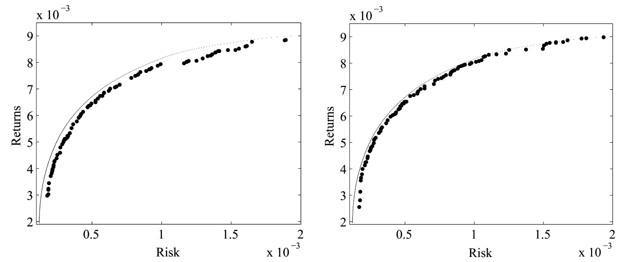
ANOVA tests revealed no significant differences between the GD attained by HR and OR-3. However, there are significant differences in the degree in which they satisfy the diversity goal of attaining a solution set that spans the entire efficient frontier. Except for PORT1 and PORT3, the ANOVA test reveals that OR-3 actually attains a significantly higher value of MS as compared to HR. Figs. 9 and 10 compare the EF_{known} obtained by HR and OR-3 in PORT2 and PORT4. They clearly illustrate the difference in diversity under these two representations, in accordance with Fig. 7. It is evident from Fig. 11 that OR-3 was able to attain a set of solutions that is close to the efficient frontier with sufficient level of diversity for the rest of the problems. However, it is noticeable that certain regions of the efficient frontier were not well-defined. Hence, to further improve the algorithmic performance of OR-3 in terms of diversity, local search operators could be deployed in future works to improve the algorithmic convergence.

A closer examination in Fig. 7 reveals differences in the algorithmic performance for the various initialization techniques. As discussed earlier, OR-1 will favor smaller portfolios, thus the algorithm will work with fewer assets initially and then gradually increases the portfolio size during the evolutionary search progress. This can be observed from the evolutionary traces of the portfolio sizes in Fig. 12. On the other hand, the application of the fixed initialization limits of 0.1 or random initialization increases the initial portfolio sizes, with the latter providing more diversity in the initial population, as observed in Fig. 12 (b).



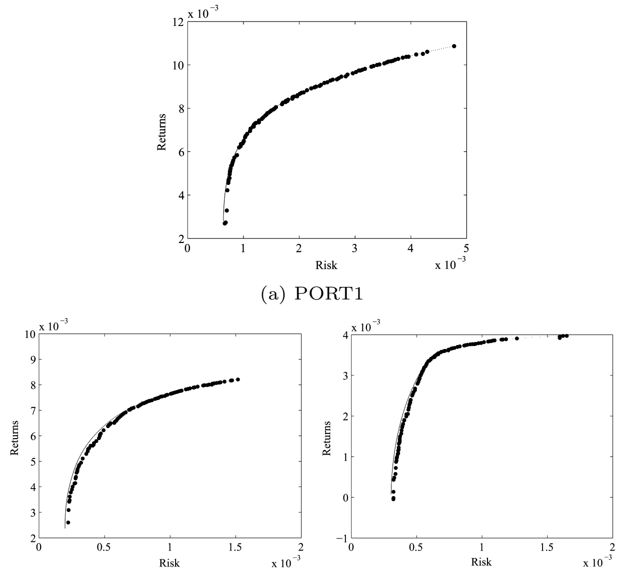
(a) HR (b) OR-3

Fig. 9 EF_{known} obtained by HR and OR-3 for PORT2 in one of the algorithmic runs (The dotted-line denotes the efficient frontiers.)



(a) HR (b) OR-3

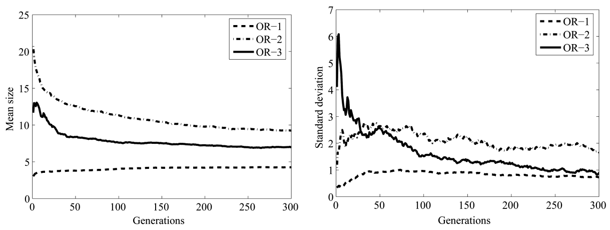
Fig. 10 EF_{known} obtained by HR and OR-3 for PORT4 in one of the algorithmic runs (The dotted-line denotes the efficient frontiers.)



(a) PORT1

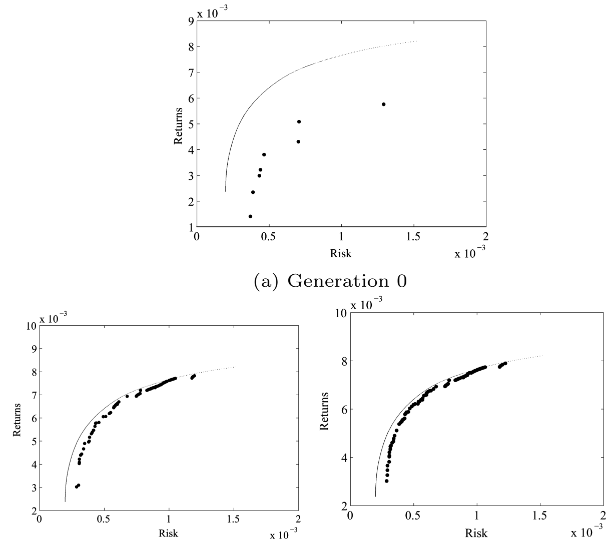
(b) PORT3 (c) PORT5

Fig. 11 EF_{known} obtained by OR-3 for selected problems in one of the algorithmic runs (The dotted-line denotes the efficient frontiers.)



(a) (b)

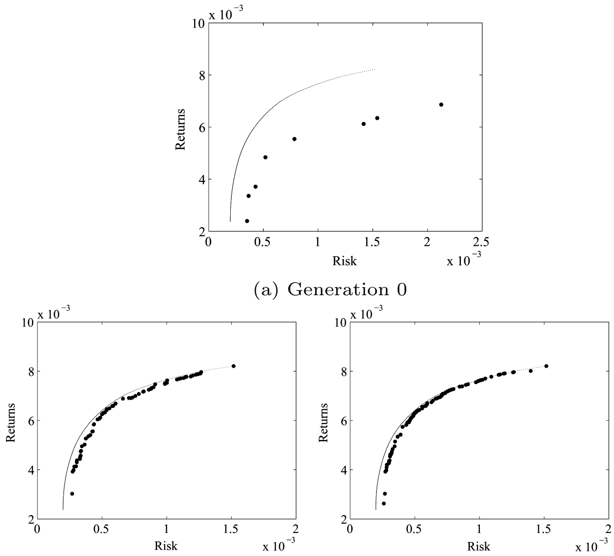
Fig. 12 Evolutionary traces of (a) the average portfolio sizes and (b) the corresponding standard deviation in PORT3 for three different algorithms, i.e., {OR1, OR2, OR3}



(a) Generation 0

(b) Generation 50 (c) Generation 100

Fig. 13 EF_{known} attained by OR1 at different generations in PORT3 (The dotted-line denotes the efficient frontiers.)



(a) Generation 0

(b) Generation 50 (c) Generation 100

Fig. 14 EF_{known} attained by OR3 at different generations in PORT3 (The dotted-line denotes the efficient frontiers.)

The importance of diversity in the initial population is reflected in the evolutionary traces of the objective space. The diverse initial population generated by OR-3 as compared with OR-1 (Fig. 13 (a) versus Fig. 14 (a)) resulted in a more diverse set of solutions (in terms of MS) being evolved eventually at generation 100.

6 Experimental results and discussion about constrained portfolio optimization

The experimental results earlier have demonstrated the capability of the proposed order-based representation in generating better approximation of the efficient frontier as compared with the other representations. The experimental study in this section will extend the evolutionary platform to the constrained portfolio optimization model and evaluate its constraint handling ability with respect to the floor and ceiling constraint and cardinality constraints. Particularly, the study will be restricted to PORT3 and the generational stopping criteria will be extended to 1000 to ensure algorithmic convergence.

Before examining the results for the constrained portfolio optimization model, it will be instructive to analyze how portfolio size changes along the efficient frontier. Fig. 15 plots the risk against the portfolio size for all the solutions obtained by OR3 for PORT3 in the 30 experimental runs. Clearly, smaller portfolio sizes are associated with higher risk bands while larger portfolio sizes possess smaller risk due to diversification. Thus, the imposition of floor and ceiling constraint and cardinality constraint, which will limit the portfolio sizes, will influence the level of return and risk attainable, and then restricting the constrained efficient frontier to certain regions of the efficient frontier.

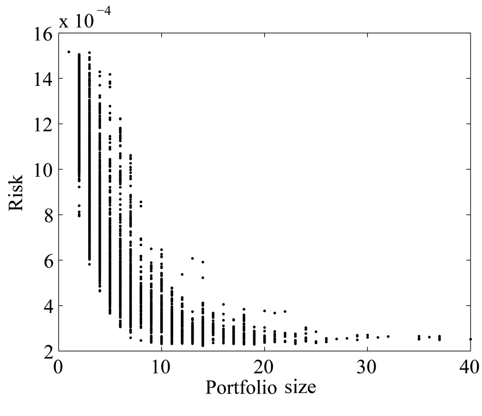
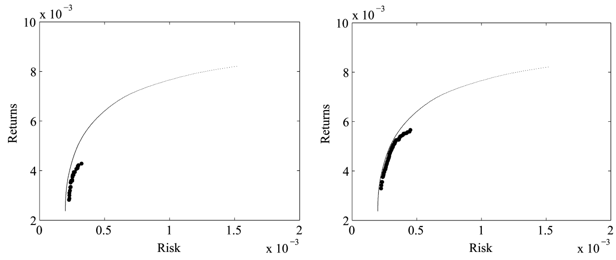


Fig. 15 The risk against portfolio size obtained by OR3 in PORT3

Particularly in the context of floor and ceiling constraint, the former will force a minimal exposure to those lower-returning assets, while the latter will prevent the high optimal level of exposure to high returns assets, again forcing an exposure to lower-returning assets. This will ultimately reduce the overall portfolio′s return, resulting in suboptimal portfolio. To verify this hypothesis, two sets of floor and ceiling constraints, {1%, 2%} and {10%, 11%}, are considered. Fig. 16 shows the constrained EF_{known}. Clearly, with this constraint, it is not possible to approximate the entire efficient frontier, and $EF_{\rm known}$ attained are limited to the low risk-return region.



(a) {1%, 2%} (b) {10%, 11%}

Fig. 16 Constrained $EF_{\rm known}$ attained for PORT3 with different floor and ceiling constraint (The dotted-line denotes the unconstrained efficient frontiers.)

To further investigate the effects of floor and ceiling constraint on portfolio sizes, different values of the constraint were considered and the average portfolio sizes obtained under the various instances is shown in Fig. 17. The dark region denotes the infeasible case where the floor constraint is higher than the ceiling constraint. With the floor and ceiling constraint, the average portfolio size generally increased as compared with 7.02 in the unconstrained case. By using a higher ceiling constraint, larger weight values are possible, resulting in the reduction of the portfolio size. Similarly, increasing the floor constraint will have the same effect as larger weight values were required. Comparing the set of constraint considered, {1%, 2%} attains a larger portfolio size, resulting in the attainable Pareto front to be situated near the low risk and return region in Fig. 16 (a) due to excessive diversification. On the other hand, increasing the constraints values to {10%, 11%} allows higher risk–return portfolio to be attained and stretched the attainable $EF_{\rm known}$ upwards as shown in Fig. 16 (b).

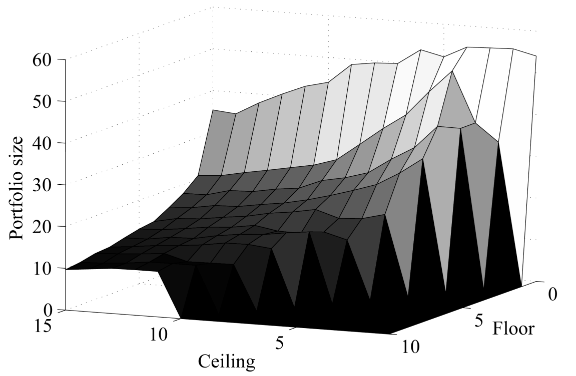
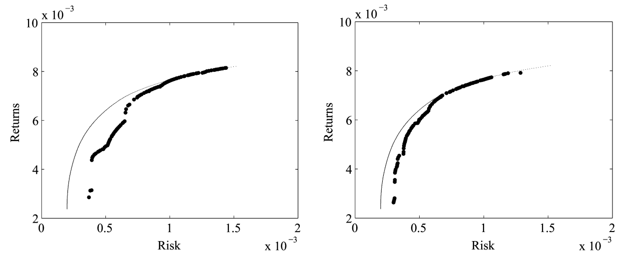


Fig. 17 Average portfolio size obtained for various values of floor and ceiling constraint

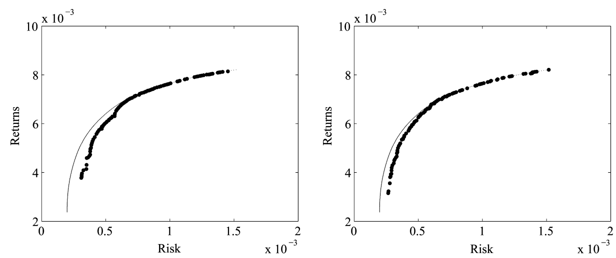
Contrary to the floor and ceiling constraints, cardinality constraint influences the portfolio size directly. Thus, the cardinality-constrained efficient frontier might be discontinuous, as certain portfolios will not be available for the rational investor[18]. Fig. 18 shows the effects of adopting a fixed cardinality constraint. The discontinuity phenomenon is clearly evident here where the tight cardinality constraint confined the constrained $EF_{\rm known}$ to the high risk region as efficient risk diversification is ruled out.



(a) {2, 2} (b) {3, 3}

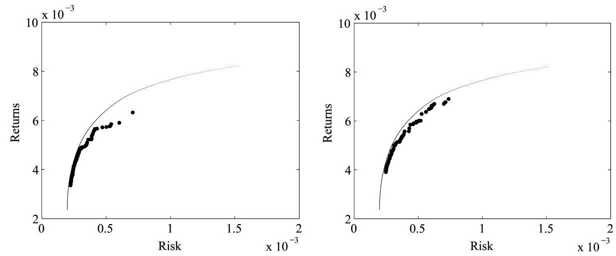
Fig. 18 Constrained $EF_{\rm known}$ attained for PORT2 with different cardinality constraints (The dotted-line denotes the unconstrained efficient frontiers.)

However, as the cardinality limits were relaxed, the constrained $EF_{\rm known}$ became more continuous as illustrated in Fig. 19 where the constraints were relaxed to {2, 3} and {1, 4}, respectively. However, the low risk-return regions are not very well defined since large portfolio sizes are not possible under these constraints. Nevertheless, it should be highlighted that the actual effects on the constrained frontier ultimately depend on the extent of relaxation in the cardinality constraints. Conversely, in the case for higher values of cardinality constraints in Fig. 20, the constrained $EF_{\rm known}$ are now confined to the low-risk region, since only large portfolio sizes are allowed.



(a) {2, 3} (b) {1, 4}

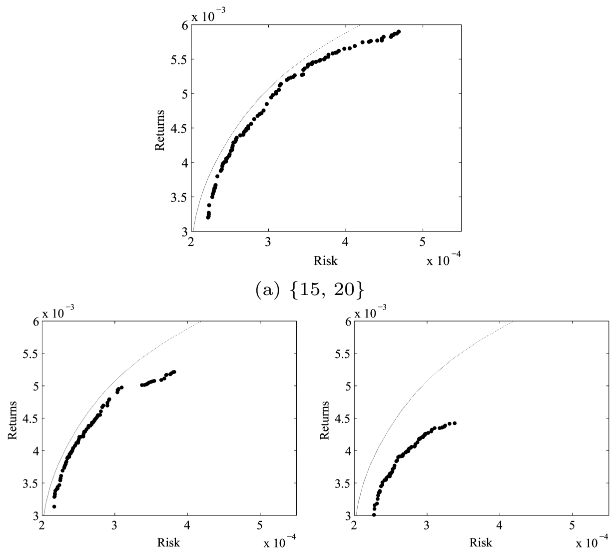
Fig. 19 Constrained $EF_{\rm known}$ attained for PORT3 with different cardinality constraints (The dotted-line denotes the unconstrained efficient frontiers.)



(a) {35, 35} (b) {32, 38}

Fig. 20 Constrained $EF_{\rm known}$ attained for PORT3 with different cardinality constraints (The dotted-line denotes the unconstrained efficient frontiers.)

To further evaluate the generality of the constraint handling technique, both floor and ceiling constraint and cardinality constraint is considered together. Previous result shows that for a floor and ceiling constraint of {1%, 12%}, the portfolio size ranged from 15 to 35 with a mean value of 23. Adopting this value of floor and ceiling constraint, different cardinality constraints were considered and the constrained EF_{known} was compared with that obtained without the cardinality constraint. Generally, the proposed constraint handling technique is able to attain an EF_{known} that satisfies both the constraints. Different levels of cardinality constraints restrict the constrained EF_{known} to different risk-return regions, i.e., the cardinality constraint {15, 20} which is below the mean portfolio size corresponds to the higher risk-return region while the higher cardinality constraint corresponds to the lower risk-return region. Fig. 21 (c) shows that if the cardinality constraint is fixed outside the optimal portfolio size range, it will result in a suboptimal Pareto front. It should be highlighted that if these constraints are too rigid, there might be a possibility that there will not be any feasible portfolio. Take for example a maximum cardinality of 3 and a ceiling constraint of 0.1, the minimum portfolio size based on the latter, i.e., 10, could not possibly satisfy the cardinality constraint under any circumstances.



(a) {15, 20}

(b) {25, 30}

(c) {50, 55}

Fig. 21 Constrained EF_{known} attained for PORT3 with combined floor and ceiling constraints at {1%, 12%} and different cardinality constraints (The dotted-line denotes the unconstrained efficient frontiers.)

This simple exercise has illustrated the viability of the proposed approach in attaining a feasible EF_{known} under these two constraints. Nevertheless, an experimental study based on proper statistical tests, similar to Section 5, is necessary for the proper evaluation of the constraint handling capability of the algorithm in terms of its effectiveness and efficiency. This will be reserved for future work, as currently, there are no other proposed MOEA that can operate under these two constraints.

7 Experimental results and discussion about preference-based portfolio optimization

As mentioned earlier, there has been an increased interest in incorporating preferences in the evolutionary search process of multi-objective optimization lately[6]. Even though the use of preference information essentially casts the optimization problem into the single-objective domain, thus possibly undermining the multi-objective optimization approach, the main advantage of adopting the latter with preferences-based techniques with respect to the former is that they generate higher diversity in the search efforts and provides alternatives in the proximity of the preferred region.

A simple technique to incorporate preferences in the optimization process is considered here. In the event of tie during the selection process which is based on Pareto dominance, the objective function (8) will be used instead of the niche count. This technique is incorporated into the MOEA used earlier and is denoted as preference-based multi-objective evolutionary algorithms (PMOEA). For comparison, a single-objective evolutionary algorithm (SOEA) based on (8) is also considered as listed in Table 4.

Table 4 Description of the various algorithm configurations in the experimental study for preference-based portfolio optimization

Algorithms	Notations
Single-objective evolutionary algorithms	SOEA
Multi-objective evolutionary algorithms	MOEA
Preference-based multi-objective Evolutionary algorithms	PMOEA

The various algorithms are applied to the same set of portfolio optimization problems used earlier. Three different values of R_f are considered for each problem, namely 25%, 50%, and 75% of the returns, which correspond to three capital market lines and efficient portfolios. The efficient portfolios are located at different regions of the efficient frontier. It should be highlighted that the risk-free rate will normally be situated at lower return-percentiles. The various values are listed in Table 5 and Fig. 22 gives a clear illustration of how these values are obtained in each problem.

Table 5 Values of R_f for the various problems (optimal F3 highlighted in parentheses)

Problem	PORT1	PORT2	PORT3	PORT4	PORT5
25% R_f	0.0034	0.0030	0.0026	0.0028	0.0010
	(3.2402)	(11.0774)	(8.2546)	(9.1543)	(4.0154)
50% R_f	0.0068	0.0059	0.0053	0.0056	0.0020
	(0.9591)	(4.0934)	(2.5864)	(2.6803)	(2.3330)
75% R_f	0.0102	0.0089	0.0079	0.0083	0.0030
	(0.1315)	(0.3799)	(0.1835)	(0.3351)	(0.8283)

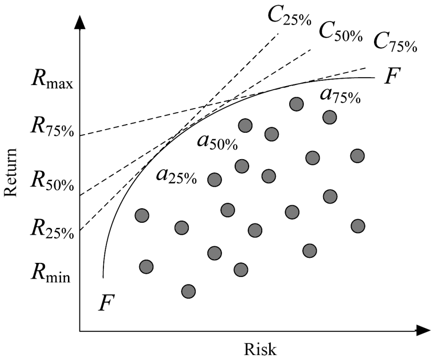
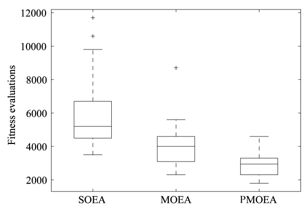
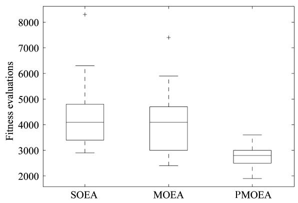


Fig. 22 Three values of R_f are derived based on 25%, 50%, and 75% returns

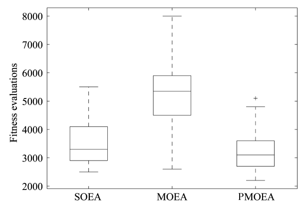
The algorithmic performances are evaluated based on the number of fitness evaluations required to reach within 5% of the optimal fitness (8) for the respective problem and R_f. The mean fitness evaluations taken in 30 experimental runs are illustrated in Fig. 23.



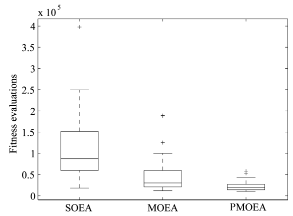
(a) PORT1 (25% R_f)



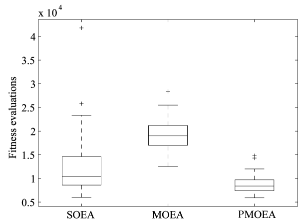
(b) PORT1 (50% R_f)



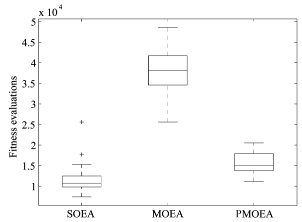
(c) PORT1 (75% R_f)



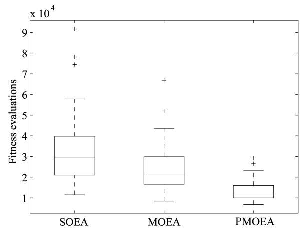
(d) PORT2 (25% R_f)



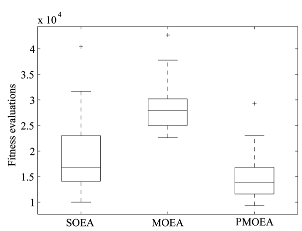
(e) PORT2 (50% R_f)



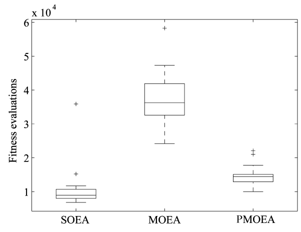
(f) PORT2 (75% R_f)



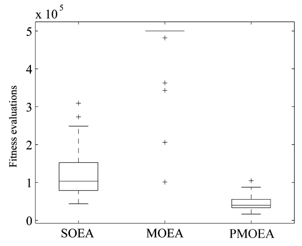
(g) PORT3 (25% R_f)



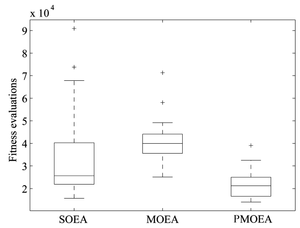
(h) PORT3 (50% R_f)



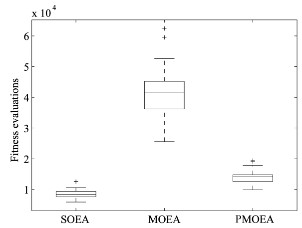
(i) PORT3 (75% R_f)



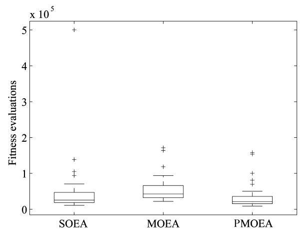
(j) PORT4 (25% R_f)



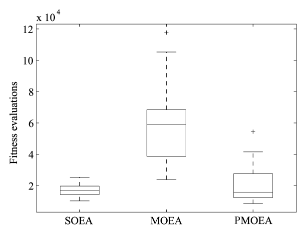
(k) PORT4 (50% R_f)



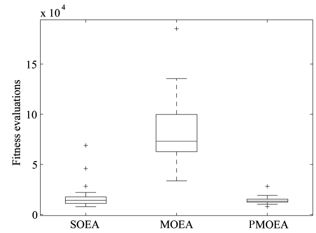
(l) PORT4 (75% R_f)



(m) PORT5 (25% R_f)



(n) PORT5 (50% R_f)

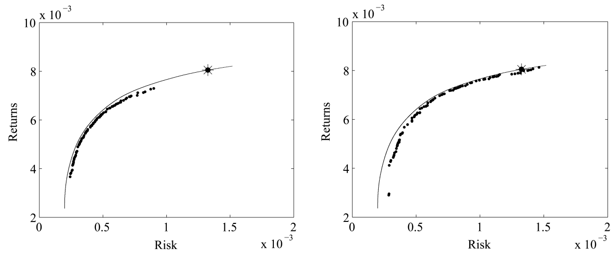


(o) PORT5 (75% R_f)

Fig. 23 The mean fitness evaluation

In Fig. 23, the mean fitness evaluation to reach within 5% of the optimal fitness (8) is obtained under the different algorithms for the different problems and R_f. In most cases, MOEA locates the preferred region since the efficient portfolio is ultimately part of the efficient frontier. However, there are improvements in the convergence time to obtain solutions in the vicinity of the preferred region after introducing the preference knowledge in the selection criteria. Interestingly, these improvements are especially significant whenever there is a large performance difference between SOEA and MOEA. For example, the significant algorithmic improvement in PORT2 for 75% R_f corresponds to the huge performance differences between SOEA and MOEA, while the contrary is observed for 25% R_f. The performance difference actually reflects the difficulty in locating the efficient portfolio on the efficient frontier and hence the level of improvement attainable. Thus, application of PMOEA will be more justified in such cases. Nevertheless, PMOEA is able to converge faster to the efficient portfolio generally as compared with SOEA, yet at the same time, offers more alternatives in the vicinity of the preferred region.

Fig. 24 shows the approximated efficient frontier for MOEA and PMOEA for PORT3. Clearly, while MOEA blindly drives solutions towards the efficient frontier, the incorporation of preference in PMOEA was able to direct solutions towards the efficient portfolio. The close-up illustration in Fig. 25 shows that PMOEA is able to obtain solutions near the efficient portfolio as opposed to SOEA. In Fig. 25, solution obtained by SOEA is included as a reference. Furthermore, solutions attained by PMOEA provide alternatives around the targeted region, allowing the portfolio to have more choices and a better understating of the problem.



(a) MOEA (b) PMOEA

Fig. 24 Estimation of the efficient frontier attained by MOEA and PMOEA for PORT3 with R_f =0.0079 within 10 000 fitness evaluations (The star and dotted-line denote the corresponding efficient portfolio and the efficient frontier, respectively.)

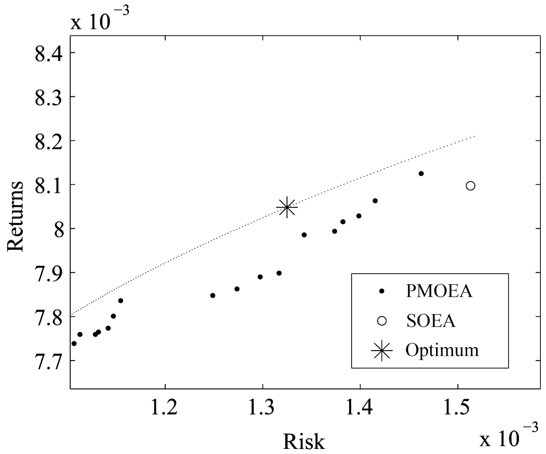


Fig. 25 Close-up illustration of the efficient frontier in the vicinity of the efficient portfolio (The star and dotted-line denote the corresponding efficient portfolio and the efficient frontier, respectively.)

8 Conclusions

This paper first introduced an order-based representation for EMOPO, which is capable of generating better approximation of the efficient frontier with respect to other conventional representations. Second, experimental results illustrate that the floor, ceiling constraint and cardinality constraint can be handled simultaneously by the proposed approach, enabling the constrained efficient frontier to be studied. Finally, the incorporation of preference-based techniques into the proposed evolutionary platform enhances its capability as a decision support system for portfolio managers in real-world implementation. Nevertheless, it is necessary to extend the evolutionary optimization model to handle other realistic constraints like round-lot constraints and transaction costs and evaluate its viability and practicality on more comprehensive test problems.

References

[1] M. G. C. Tapia, C. A. C. Coello. Applications of Multi-Objective Evolutionary Algorithms in Economics and Finance: A Survey. In *Proceedings of IEEE Congress on Evolutionary Computation*, Singapore, pp. 532–539, 2007.

[2] G. V. Louis, C. Chan, D. E. Goldberg. Investment Portfolio Optimization Using Genetic Algorithms. In *Late Breaking Papers at the Genetic Programming 1997 Conference*, California, pp. 255–263, 1997.

[3] S. Arnone, A. Loraschi, A. Tettamanzi. A Genetic Approach to Portfolio Selection. *Neural Network World*, vol. 3, no. 6, pp. 597–604, 1993.

[4] A. Loraschi, A. Tettamanzi, M. Tomassini, P. Verda. Distributed Genetic Algorithms with an Application to Portfolio Selection Problems. In *Proceedings of International Conference on Artificial Neural Networks and Genetic Algorithms*, pp. 384–387, 1995.

[5] P. Skolpadungket, K. Dahal, N. Harnpornchai. Portfolio Optimization Using Multi-objective Genetic Algorithms. In *Proceedings of IEEE Congress on Evolutionary Computation*, Singapore, pp. 516–523, 2007.

[6] H. Sato, H. Aguirre, K. Tanaka. Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs. In *Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization, Lecture Notes in Computer Science*, Springer, Matsushima, Japan, pp. 5–20, 2007.

[7] H. Markowitz. *Portfolio Selection: Efficient Diversification of Investments*, John Wiley & Sons, New York, USA, 1959.

[8] J. E. Fieldsend, J. Matatko, M. Peng. Cardinality Constrained Portfolio Optimization. In *Proceedings of 5th International Conference on Intelligent Data Engineering and Automated Learning*, Exeter, UK, pp. 788–793, 2004.

[9] R. Subbu, P. P. Bonissone, N. Eklund, S. Bollapragada, K. Chalermkraivuth. Multiobjective Financial Portfolio Design: A Hybrid Evolutionary Approach. In *Proceedings of IEEE Congress on Evolutionary Computation*, NY, USA, vol. 2, pp. 1722–1729, 2005.

[10] M. Ehrgott, K. Klamroth, C. Schwehm. An MCDM Approach to Portfolio Optimization. *European Journal of Operational Research*, vol. 155, no. 3, pp. 752–770, 2004.

[11] D. Maringer. *Portfolio Management with Heuristic Optimization*, Springer, New York, USA, 2005.

[12] F. Streichert, M. Tanaka-Yamawaki. The Effect of Local Search on the Constrained Portfolio Selection Problem. In *Proceedings of IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, pp. 2368–2374, 2006.

[13] F. Streichert, H. Ulmer, A. Zell. Evaluating a Hybrid Encoding and Three Crossover Operators on the Constrained Portfolio Selection Problem. In *Proceedings of Congress on Evolutionary Computation*, IEEE Press, Portland, USA, vol. 1, pp. 932–939, 2004.

[14] F. Streichert, H. Ulmer, A. Zell. Comparing Discrete and Continuous Genotypes on the Constrained Portfolio Selection Problem. In *Proceedings of Conference on Genetic and Evolutionary Computation, Lecture Notes in Computer Science*, Springer, Washington, USA, vol. 3103, pp. 1239–1250, 2004.

[15] F. Streichert, H. Ulmer, A. Zell. Evolutionary Algorithms and the Cardinality Constrained Portfolio Selection Problem. In *Proceedings International Conference on Operations Research*, Heidelberg, Germany, pp. 253–260, 2003.

[16] R. Armananzas, J. A. Lozano. A Multiobjective Approach to the Portfolio Optimization Problem. In *Proceedings of IEEE Congress on Evolutionary Computation*, Edinburgh, UK, vol. 2, pp. 1388–1395, 2005.

[17] R. Ruiz-Torrubiano, A. Suárez. Use of Heuristic Rules in Evolutionary Methods for the Selection of Optimal Investment Portfolios. In *Proceedings of IEEE Congress on Evolutionary Computation*, Singapore, pp. 212–219, 2007.

[18] T. J. Chang, N. Meade, J. E. Beasley, Y. M. Sharaiha. Heuristics for Cardinality Constrained Portfolio Optimization. *Computers and Operations Research*, vol. 27, no. 13, pp. 1271–1302, 2000.

[19] D. Maringer, H. Kellerer. Optimization of Cardinality Constrained Portfolios with a Hybrid Local Search Algorithm. *OR Spectrum*, vol. 25, no. 4, pp. 481–495, 2004.

[20] D. Lin, S. Wang, H. Yan. A Multiobjective Genetic Algorithm for Portfolio Selection. In *Proceedings of the 5th International Conference on Optimization: Techniques and Applications*, Hong Kong, pp. 15–17, 2001.

[21] K. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, C. Stummer. Ant Colony Optimization in Multiobjective Portfolio Selection. In *Proceedings of the 4th International Conference on Metaheuristics*, Porto, Portugal, pp. 243–248, 2001.

[22] F. Xu, W. Chen, L. Yang. Improved Particle Swarm Optimization for Realistic Portfolio Selection. In *Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, IEEE Press, Qingdao, China, vol. 1, pp.185–190, 2007.

[23] L. Diosan. A Multi-objective Evolutionary Approach to the Portfolio Optimization Problem. In *Proceedings of International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, Vienna, Austria, vol. 2, pp. 183–187, 2005.

[24] R. Moral-Escudero, R. Ruiz-Torrubiano, A. Suarez. Selection of Optimal Investment Portfolios with Cardinality Constraints. In *Proceedings of IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, pp. 2382–2388, 2006.

[25] F. Rothlauf, D. E, Goldberg. Redundant Representations in Evolutionary Computation. *Evolutionary Computation*, vol. 11, no. 4, pp. 381–415, 2003.

[26] J. E. Beasely. OR-Library: Distributing Test Problems by Electronic Mail. *Journal of the Operational Research Society*, vol. 41, no. 11, pp. 1069–1072, 1990.

[27] K. Deb. *Multi-objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, New York, USA, 2001

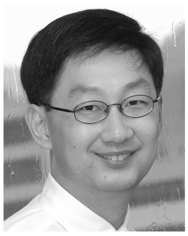
[28] E. Zitzler, K. Deb, L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[29] D. A. V. Veldhuizen, G. B. Lamont. Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-arts. *Evolutionary Computation*, vol. 8, no. 2, pp. 125–147, 2000.

[30] C. A. C. Coello, D. A. V. Veldhuizen, G. B. Lamont. *Evolutionary Algorithms for Solving Multi-objective Problems*, Kluwer Academic Publishers, New York, USA, 2002.

S. C. Chiam received the B. Eng. degree (the first class honors) in electrical engineering from the National University of Singapore in 2005. He is currently a Ph. D. candidate at the Centre for Intelligent Control, National University of Singapore.

His research interests include evolutionary computation and neural networks, specifically in the application of evolutionary multi-objective optimization techniques in the field of computational finance, i.e., portfolio optimization and time series forecasting.



K. C. Tan received the B. Eng. degree (the first class honors) in electronics and electrical engineering and the Ph. D. degree from the University of Glasgow, Glasgow, Scotland, in 1994 and 1997, respectively. He is currently an associate professor in the Department of Electrical and Computer Engineering, National University of Singapore. He has authored or coauthored two books and more than 140 journal and conference publications. He is also an international program committee member for over 50 conferences and served in the organizing committee for over 15 international conferences, including the technical program co-chair for *IEEE Congress on Evolutionary Computation* in 2005, program chair for *IEEE Conference on Cybernetics and Intelligent Systems* in 2004 and 2006, general co-chair for *IEEE Congress on Evolutionary Computation* in 2007 in Singapore and *IEEE Symposium on Computational Intelligence in Scheduling* in 2007 in Hawaii. He is an associate editor for *IEEE Transactions on Evolutionary Computation*.

His research interests include computational intelligence, evolutionary computation, multi-objective optimization, and engineering design optimization.

A. Al Mamun graduated from the Indian Institute of Technology, Kharagpur, India in 1985. He received the Ph. D. degree from the National University of Singapore in 1997. In his professional career, he worked as a research engineer at the Data Storage Institute, Singapore, and as staff engineer at Maxtor Peripherals prior to joining the faculty of Department of Electrical and Computer Engineering, National University of Singapore.

His research interests include precision servomechanism, mechatronics, intelligent control, and autonomous mobile robots.