Applied Soft Computing 146 (2023) 110652

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A knowledge-based constructive estimation of distribution algorithm for bi-objective portfolio optimization with cardinality constraints

Zhi-Xuan Zhang [a,b], Wei-Neng Chen [a,b,*], Xiao-Min Hu [c]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[b] State Key Laboratory of Subtropical Building Science, South China University of Technology, Guangzhou 510006, China
[c] School of Computers, Guangdong University of Technology, Guangzhou 510006, China

ARTICLE INFO

Article history:
Received 19 October 2021
Received in revised form 13 July 2023
Accepted 18 July 2023
Available online 22 July 2023

Keywords:
Portfolio optimization
Mean–variance model
Constraint optimization
Ant colony optimization
Estimation distribution algorithm
Multi-objective optimization

ABSTRACT

Portfolio optimization is an essential and practical model for financial decision making. With the consideration of some real-world constraints, especially the cardinality constraints, the problem becomes much more challenging as it converts to a mixed-integer quadratic multi-objective optimization problem. To solve this problem, we propose a knowledge-based constructive estimation of distribution algorithm (KC-EDA) with the following three features. First, a hybrid design of Ant colony optimization (ACO) and Estimation distribution algorithm (EDA) is used to solve this mixed-variable optimization problem based on knowledge information. Second, a knowledge accumulation mechanism is designed to discover the internal relationship among the assets. The mechanism can not only guide the selection of assets effectively but also enable the use of historical information during evolution to direct the allocation of investment proportion. Third, a constructive approach is applied to construct portfolios under the constraints. This hybrid and constructive approach is incorporated into the multi-objective evolutionary framework and the experiment has been performed on the SZ50, SZ180, and SZ380 datasets (from January 2014 to December 2018). The experimental results demonstrate the effectiveness of KC-EDA in solving the portfolio optimization problem with cardinality constraints.

Code metadata

Permanent link to reproducible Capsule: https://doi.org/10.24433/CO.4030045.v1

1. Introduction

As early as 1952, Markowitz [1] proposed the mean–variance (MV) model in his article "Portfolio selection", which built the asset risk measurement mechanism firstly. This model has led the era of quantitative finance and further evolved into the basis of modern financial investment theory. Following Markowitz's theory, the mean–variance portfolio optimization problem (MVPO) is defined to maximize the expected return and minimize the risk, i.e., the variance of the expected return, by selecting appropriate assets and determining their investment proportions.

According to the general assumption of modern classical finance, investors in the market are supposed to be perfectly rational, which means they will always select the portfolio with the least risk at the same return or that with the highest return at the same risk. However, return maximization and risk minimization are two conflicting objectives that cannot be fulfilled simultaneously. Investors tend to select their preferred portfolio from various return and risk combinations. Therefore, the MVPO problem was further developed into a multi-objective optimization problem. The conventional methods for handling this problem include the weighted sum method [2–5] and the Pareto-based method [6–8]. Beyond the mean and variance objectives, more consequential objectives were introduced by researchers, such as downside risk [9–11], value-at-risk (VaR) [12], Sharpe ratio [13,14], trend ratio [15], investor preference [16,17], etc.

Multi-objective optimization evolutionary algorithms (MOEAs) have been widely applied to solve this problem. MOEAs can get a set of dominant solutions in a single run. Each solution represents a portfolio with an expected return value and the lowest risk value that matches this return level. The investors can select a portfolio from the Pareto front found by the algorithm according to their risky preference and gain the equipotent expected return at the same time. More detailed surveys about

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author at: School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China.
E-mail address: cwnraul634@aliyun.com (W.-N. Chen).

https://doi.org/10.1016/j.asoc.2023.110652
1568-4946/© 2023 Published by Elsevier B.V.

using MOEAs to solve portfolio optimization can be referred to Metaxiotis et al. [18], Ponsich et al. [19], and Kalayci et al. [20].

In the original MV model, it only requires the total investment proportion is equal to one, which is called the budget constraint. In order to improve the practicability of the MV model, some real-world constraints have been taken into consideration, such as boundary constraint [21–23], grouping constraint [3,24], cardinality constraint [3,21,25,26], round lot constraint [21,24,27], turnover constraint [28] and transaction cost constraint [29–32]. With the consideration of these real-world constraints, especially the cardinality constraint, the MVPO problem becomes even more challenging, as its search space becomes non-consecutive, and some traditional methods become invalidate [33,34].

In the literature, there are three main methods used to handle constraints: repair mechanism, penalty function, and infeasibility tolerance, which can also be used for MVPO problem with constraints. The main idea of repair mechanism is to fix infeasible solutions into feasible ones. Chang et al. [25] proposed a repair procedure to fix the boundary constraint and cardinality constraint. The idea of penalty function is to assign a penalty value to each infeasible solution according to its violation degree [28,35,36]. The infeasibility tolerance technique [37] allows the existence of infeasible solutions. Deb [38] proposed a selection rule which indicates feasible solutions are prior to infeasible solutions, and infeasible solutions are sorting by their violation degree of constraints. Moral et al. [39] combined GA and quadratic programming to construct a hybrid strategy to find out the lowest risk for a given return. They used the random assorting recombination (RAR) operator to design the crossover operation which can satisfy cardinality constraint during evolution. Pai et al. [3] used the k-means clustering method to handle the cardinality constraint. They clustered the candidate assets into K kinds and then selected one asset from each kind to form the portfolio. They also designed a particularly refined weight standardization algorithm to handle the boundary and grouping constraints.

These methods can solve MVPO problem with cardinality constraints, but they consider the candidate assets as relatively independent entities when designing evolutionary operators or heuristic strategies. In fact, the interrelation among candidate assets can significantly influence the risk of the portfolio. The risk of the portfolio consists of two parts: the weighted average of the variance of every asset and the weighted average of the covariance between every two assets. The risk is hedged by the combination of the appropriate assets. Regarding the candidate assets as separated entities cannot correctly reflect the property of MVPO problem.

To provide a new thought to MVPO problem with cardinality constraints, we intend to propose a knowledge-based constructive estimation of distribution algorithm (KC-EDA). The innovations of this method mainly include the following three points:

(1) Based on the mixed-variable optimization characteristics of MVPO problem, the proposed method applies a hybrid design to combine the advantages of Ant Colony Optimization (ACO) and Estimation of Distribution Algorithm (EDA). We apply ACO to accomplish asset selection before using EDA to carry on investment proportion allocation.

(2) We design a knowledge-based assets selection and proportion allocation process to construct portfolios. It considers the relationship between assets and keeps discovering the interrelation among the assets during evolution. It can also utilize the distribution information from the historical solution set and learn from the proper investment.

(3) We design a constructive approach to handle budget constraint, boundary constraint, grouping constraint, and cardinality constraint. This constructive approach guarantees to build feasible solutions, and thus it can improve efficiency as it avoids searching in infeasible areas.

The proposed KC-EDA is more in line with the property of MVPO problem and the constructive approach is more efficient in searching for feasible portfolios. The experimental results show that it can obtain a better Pareto front compared with the existing approaches.

The rest of this paper is organized as follows: Section 2 explains the mathematical model of MVPO problem with cardinality constraints in detail. Section 3 presents the background of this research. Section 4 introduces the knowledge-based constructive estimation of distribution algorithm and its comprehensive implementation. The experimental studies are shown in Section 5, and conclusions are finally drawn in Section 6.

2. Mathematical model of the portfolio optimization problem with cardinality constraints

Selecting appropriate assets from lots of candidate assets is a prevalent investment practice in the financial market. According to the MV model, the portfolio's evaluation can be calculated by its constituent assets' property and investment proportion. Besides the budget constraint, this paper considers three additional constraints: boundary constraint, grouping constraint, and cardinality constraint. The mathematical model is described in detail as follows.

2.1. The mean–variance model of the portfolio optimization problem

Let N be the number of the candidate assets in the market, the mathematical formulation of the MV model can be described as

$$\text{Max} \quad \mu_p = \sum_{i=1}^{N} w_i \mu_i \tag{1}$$

$$\text{Min} \quad \sigma_p^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i \sigma_{ij} w_j \tag{2}$$

$$\text{S. t.} \quad \sum_{i=1}^{N} w_i = 1 (0 \le w_i \le 1) \tag{3}$$

where w_i is the investment proportion of candidate asset i, μ_i denotes the expected return of candidate asset i and σ_{ij} denotes the covariance between asset i and asset j. The estimated values of μ_i and σ_{ij} can be calculated by using historical data. According to Markowitz's theory, the expected return μ_p and risk σ_p^2 of the portfolio can be calculated by (1) and (2). The investor will appropriately arrange the investment proportion w_i of each asset to maximize return and minimize risk. As shown in (3), the total investment proportion should be equal to one, which means the total available fund (budget constraint). Besides, the investment proportion w_i should be greater than zero (no short-selling) and less than one (no leveraging).

2.2. The constraints

2.2.1. The boundary constraint

For each candidate asset i, its investment proportion w_i should lie in the boundary range set by the investor.

$$\begin{aligned} &lowerbound_i \le w_i \le upperbound_i \\ &(0 \le lowerbound_i < upperbound_i \le 1) \end{aligned} \tag{4}$$

In practical investment activities, investors generally avoid holding only a tiny proportion of an asset ($lowerbound_i$), which will lead to high transaction costs and almost impossible to trade. On the other hand, investors will also avoid investing a very high proportion in a single asset ($upperbound_i$), which is incapable of risk dispersion.

2.2.2. The grouping constraint

For a given group G_j, the total investment proportion of its inclusive assets should lie in the boundary range.

$$\begin{aligned} & lowergroup_j \leq \sum_{i \in G_j} w_i \leq uppergroup_j \\ & (0 \leq lowergroup_j < uppergroup_j \leq 1) \end{aligned} \tag{5}$$

The candidate assets in the market may belong to several different groups. For example, the stock assets can divide into financial industry, energy industry, technology industry, and so on. The stock assets that belong to the same industry can be viewed as a group. The investor can assign the minimum investment proportion ($lowergroup_j$) of each group G_j according to their preferences. To avoid the investment gathers in a single group, the investors can also set the maximum investment proportion ($uppergroup_j$) of each group G_j.

2.2.3. The cardinality constraint

For N candidate assets in the market, the number of assets selected in the portfolio is denoted as K, which should lie in the boundary range set by the investor.

$$\begin{aligned} & K_{lower} \leq K \leq K_{upper} \\ & (1 \leq K_{lower} < K_{upper} \leq N) \end{aligned} \tag{6}$$

To avoid that the number of assets in the portfolio is too small, the investor can declare the minimum number of the selected assets K_{lower}. The number of candidate assets N may be multitudinous, but the investor can only hold a limited number of assets. The investor can declare the maximum number of selected assets K_{upper}, which prevents over-disperse investment.

2.3. The mathematical model of the portfolio optimization problem with cardinality constraint

The objective functions of MVPO problem with cardinality constraint are maximizing the return and minimizing the risk, which are defined by (1) and (2). Meanwhile, the portfolio should satisfy budget constraint (3), boundary constraint (4), grouping constraint (5), and cardinality constraint (6). Due to the cardinality constraint, portfolio optimization becomes a mixed-integer quadratic optimization problem. This problem consists of two tasks: asset selection and investment proportion allocation. The first task is a combinatory optimization problem and the second task is a continuous optimization problem. The traditional methods cannot deal very well with this problem.

3. Background

3.1. Multi-objective evolutionary algorithm

A multi-objective optimization problem (MOP) with N different objectives can be formulated as

$$\max F(x) = \{f_1(x), f_2(x), \ldots, f_N(x)\}, x \in \Omega \tag{7}$$

where x is the decision variable defined in domain Ω and $f_1(x), f_2(x), \ldots, f_N(x)$ are N objective functions needed to be optimized simultaneously. The Pareto dominance relation is widely used to compare fitness values between solutions.

Definition 1 (*Pareto Dominance*)**.** : Pareto dominance is a binary relationship defined in Ω. Given two feasible solutions $u, v \in \Omega$, u Pareto dominates v if and only if $f_i(u) \geq f_i(v)$, $i \in \{1, 2, \ldots, N\}$ for all objectives and $f_j(u) > f_j(v)$, $j \in \{1, 2, \ldots, N\}$ for at least one objective.

For a solution x^*, if no other solution x from Ω such that x dominates x^*, then x^* is called a Pareto optimal solution. The set of all Pareto optimal solutions form the Pareto set (PS). The objective function vectors of all solutions in PS form the Pareto front (PF). Multi-objective evolutionary algorithms (MOEAs) aim to approximate the PS of multi-objective optimization problems by a single run of evolutionary algorithms. Some popular MOEAs include NSGA-II [40], NSGA-III [41], SPEA II [42], MOEA/D [43], MOEA/HD (Hierarchical Decomposition) [44], Multi-population based MOEA [45], Fuzzy Decomposition-based MOEA [46], etc.

3.2. Estimation of distribution algorithm

Estimation of distribution algorithm (EDA) is a kind of evolutionary computation method bases on the theory of probability. EDA can predict the optimal area of solution space by sampling and statistical learning. The main steps of EDA include initialization of population, selection of promising individuals, estimation of the probability model, sampling new individuals, and these steps iterate until the terminal condition is met. In theoretical study, Zhang and Muhlenbein [47] analyzed the convergence of EDA and discussed the condition of converging to global optimum. Rastegar and Meybodi [48] analyzed the upper bound of global convergence iteration times and the exact iteration times of converging to global optimum. Chen et al. [49] analyzed the average time complexity of two kinds of EDA: Univariate Marginal Distribution Algorithm (UMDA) and the Incremental UMDA. In recent years, research of EDA mainly focused on improving probabilistic model [50–53], maintaining population diversity [54], designing hybrid algorithms [55], solving multimodal optimization problems [56,57], etc. For MVPO application, Ruiz et al. [33] further proposed an EDA with pruning heuristics. It uses EDA to generate a portfolio with K assets and keeps removing the assets which are unlikely to appear in the optimal solutions. The pruning heuristics can reduce the number of candidate assets and satisfy the cardinality constraint.

3.3. Ant colony optimization algorithm

Ant colony optimization (ACO) algorithm is a kind of population-based stochastic optimization algorithms which are widely used to solve combinatorial optimization problems. Dorigo et al. [58,59] proposed the ant system in 1992, where they simulated the behavior of real ant colonies to solve the traveling salesman problem (TSP). ACO constructs an information sharing and knowledge accumulating mechanism by using the pheromone matrix and heuristic information matrix. ACO has the characteristics of distributed computation, self-organization, positive feedback of information, and robustness. Because of these properties, ACO is further applied in solving dynamic traveling salesman problem (DTSP) [60], evacuation path optimization (EPO) [61], autonomous underwater vehicles (AUVs) path planning [62,63], controlling the Pollutant Spreading [64], dynamic data stream clustering [65], cloud workflow scheduling [66,67], etc. In theoretical study, Kong and Tian [68] mathematically described the simplified Ant System as a Markov chain and proofed it has probability 1 to get to global optimum. Badr and Fahmy [69] modeled ant algorithm as branching random process which can reach a stable distribution. Duan et al.[70] provided a theoretical proof for the convergence of ACO by using Markov chains and martingale theory, the pheromone trail vector will converge to global optimum when the iteration time approaching infinite. Yang et al.[71] provided a runtime analysis of ACO by modeled

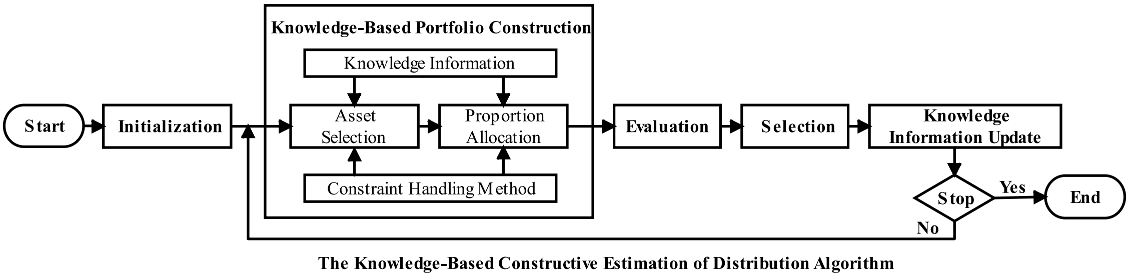

Knowledge-Based Portfolio Construction
Knowledge Information
Asset Selection
Proportion Allocation
Constraint Handling Method
Start
Initialization
Evaluation
Selection
Knowledge Information Update
Stop
Yes
No
End
The Knowledge-Based Constructive Estimation of Distribution Algorithm

Fig. 1. Main frame of the proposed algorithm.

as an absorbing Markov chain. For MVPO application, Doerner et al.[72] applied ACO in portfolio selection and showed the candidate assets are not isolated individuals. These assets have a complicated internal correlation, which can be multidimensional.

4. Design of the knowledge-based constructive estimation of distribution algorithm

To solve the portfolio optimization problem with cardinality constraint, we design a knowledge-based constructive estimation of distribution algorithm (KC-EDA). It can approximate the Pareto front (PF) according to budget constraint, boundary constraint, grouping constraint, and cardinality constraint. The main framework of the proposed algorithm is shown in Fig. 1. It is composed of five steps, i.e., initialization, knowledge-based portfolio construction, evaluation, selection, and knowledge information update. Among these, knowledge-based portfolio construction is the kernel of KC-EDA. It uses a hybrid design of ACO and EDA to construct a portfolio based on the knowledge information. This knowledge information includes the assets combinatorial information and proportion distribute information, which is updating during the evolution process. In addition, a constructive approach is well designed to handle the constraint, which can construct the portfolio in the feasible solution space.

4.1. Encoding scheme

A solution of MVPO problem with cardinality constraint consists of two parts:

Part1: The index number list of K kinds of assets included in the portfolio:

$$\text{index number list}: (a_1, a_2, \ldots, a_K) a_i \in N \tag{8}$$

Part2: The investment proportion of each asset included in the portfolio:

$$\text{investment proportion}: (w_1, w_2, \ldots, w_K) \tag{9}$$

It should be noticed that the index number list should satisfy the cardinality constraint given by investors (6). Besides, the investment proportion should fulfill budget constraint (3), boundary constraint (4), and grouping constraint (5). Once the selected K kinds of assets and their investment proportions are determined, we can apply (1) and (2) to calculate the *Return* and *Risk* of the solution.

4.2. The detailed design of KC-EDA

As mentioned above, the MVPO problem's solution is mixed-variable encoding. The index number list is a discrete variable list while the investment proportion is a continuous variable list. These two lists should be optimized as a whole but asynchronous, which brings a challenge to algorithm design. The investment proportion is corresponding to a specific index number list, which will become invalid with the changing of assets. It is another challenge to allocate the investment proportion under multi constraints. Based on this situation, we propose KC-EDA to solve MVPO problem with cardinality constraint. It uses the hybrid design of ACO and EDA to optimize mixed-variable and applies a constructive approach to handle constraints. Different from the existed approaches, KC-EDA considers the relationship between assets and learns from the proper investment which is defined as knowledge information. This knowledge information will update during evolution. The KC-EDA consists of the following main steps:

Step 1 An initial population with size *Popsize* is generated and saved as the preserve population in the first generation. These solutions are randomly sampled feasible solutions. Then we evaluate the objective functions of each individual in the initial population.

Step 2 A new population is constructed by using *Asset_Selection*() and *Weight_Assignment*() functions. Then we evaluate the objective functions of each individual in the new population.

Step 3: The preserve population and the new population are merged into a united population that contains 2**Popsize* individuals. We perform the Non-dominated sorting with respect to the objective functions *Return* and *Risk*, each individual will obtain a Pareto level *rank*. After that, we evaluate the crowding distances of all individuals in *R*.

Step 4: We use *Individual_Selection*() function to select *Popsize* optimal individuals from the united population which constitutes the preserve population. *Individual_Selection*() function applies the individual selection mechanism proposed in the NSGA-II [40].

Step 5: We update the pheromone values according to the elite individuals in the united population. These pheromone values can discover the internal relation between assets and increase the occurring probability of suitable combinations.

Step 6: We apply *Estimate_Distribution*() function to estimate the distribution information. It allows the proportion allocation process to utilize the distribution information from the proper investment.

This procedure repeats until it achieves the specified generation. The pseudo-code for KC-EDA is given in Algorithm 1.

```
Algorithm 1: KC-EDA
Input:
•  Gen: maximum generations
•  Popsize: population size
•  K: cardinality constraints
•  P: pheromone matrix
•  $\bar{\mu}$: mean investment proportion of each asset
•  $\bar{\sigma}$: standard deviation of the investment proportion
Output: The nondominated solutions in $\boldsymbol{S}_i$
$i \leftarrow 0$;
$\boldsymbol{S}_0 \leftarrow$ Generate Popsize individuals as initial population;
$\boldsymbol{S}_0 \leftarrow$ Individual_Evaluation ( );
$\boldsymbol{Q}_0 \leftarrow \emptyset$  new sampled population initializes as an empty set;
while $i < Gen$
    $\boldsymbol{Q}_i \leftarrow$ Sample Popsize individuals using Asset_Selection ( ) and
            Weight_Assignment ( );
    $\boldsymbol{Q}_i \leftarrow$Individual_Evaluation ( );
    $\boldsymbol{R} \leftarrow \boldsymbol{S}_i + \boldsymbol{Q}_i$;
    $\boldsymbol{R} \leftarrow$ Non-dominated Sorting ( );
    $\boldsymbol{R} \leftarrow$ Crowding-distance Assignment ( );
    $\boldsymbol{S}_{i+1} \leftarrow$ Individual_Selection ( );
    $P \leftarrow$Update_Pheromone ( );
    $\bar{\mu}, \bar{\sigma} \leftarrow$ Estimate_Distribution ( );
    $i \leftarrow i+1$;
end
return the nondominated solutions in $\boldsymbol{S}_i$
```

4.3. Knowledge-based asset selection

The first step of knowledge-based portfolio construction is asset selection, we design a knowledge-based asset selection process to construct the index number list encoded by (8). Selecting K kinds of assets from N kinds of candidate assets is a combinational optimization problem. The candidate assets have a complicated internal correlation, which is multidimensional. Based on this characteristic, the feasible combination of several kinds of assets could generate a low-risk portfolio with a considerable return, because the investment risk is appropriately spread out over every asset. To dig out this internal correlation between candidate assets, we apply the pheromone matrix of ACO as a knowledge information carrier. The pheromone matrix is shown below.

$$\text{pheromone matrix}: \begin{pmatrix} \tau_{11} & \tau_{12} & \cdots & \tau_{1N} \\ \tau_{21} & \tau_{22} & \cdots & \tau_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{N1} & \tau_{N2} & \cdots & \tau_{NN} \end{pmatrix} \tag{10}$$

Pheromone value τ_{ij} represents the association strength between asset i and asset j, the higher value means they are closely related and more likely to include in the portfolio at once. We define $\tau_{ij} = \tau_{ji}$ because of symmetry, which means the association strength is directionless. Besides, pheromone value τ_{ii} denotes the significance of asset i, the higher value means this asset is more likely to be selected at the beginning of asset selection process. The pseudo-code for asset selection process is given in Algorithm 2.

The *Asset_Selection*() function can select K kinds of assets from N kinds of candidate assets according to the pheromone matrix. The selection process applies binary tournament selection strategy to select the first kind of asset according to the diagonal pheromone value τ_{ii}. After that, the function selects the next kind of asset based on the pheromone value $\tau_{a_{k-1},i}$ correlates to the previously selected asset a_{k-1}, the unselected assets with higher pheromone value are more likely to be selected. This process repeats till reaching the cardinality constraint K. Then, the *verification*() function will check if these K kinds of assets can satisfy budget constraint, boundary constraint, and grouping constraint. If so, the index number list will be returned; if not, the selection process will repeat in again and try to find out a feasible combination. When running out of the trying time T, it will randomly reuse a feasible combination from the optimal solution set S_i.

The knowledge information uses to direct the assets selecting process here can update during evolution. At the beginning of KC-EDA, pheromone values τ_{ij} are all initialized to $\tau_0 = 1$.

$$\tau_{ij} = \tau_{ji} = \tau_0 = 1 \forall i, j \in \{1, 2, \ldots, N\} \,\&\, i \neq j \tag{11}$$

```
Algorithm 2: Asset_Selection ( )
Input:
•  K: cardinality constraint
•  T: maximum trying time
•  C: candidate assets list
•  P: pheromone matrix
Output: index number list of solution x: $(a_1, a_2, \ldots, a_K)$
$t \leftarrow 0$, $C' \leftarrow C$;
while $t < T$
    $i, j \leftarrow$ Randomly select two assets from $C'$;
    if $\tau_{ii} \geq \tau_{jj}$
        $a_1 = i$, remove $i$ from $C'$;
    else
        $a_1 = j$, remove $j$ from $C'$;
    end
    for $k = 2 : K$
        $i, j \leftarrow$ Randomly select two assets from $C'$;
        if $\tau_{a_{k-1},i} \geq \tau_{a_{k-1},j}$
            $a_k = i$, remove $i$ from $C'$;
        else
            $a_k = j$, remove $j$ from $C'$;
        end
    end
    if verification ( ) == True
        return $(a_1, a_2, \ldots, a_K)$;
    else
        $t \leftarrow t + 1$, $C' \leftarrow C$;
    end
end
$(a_1, a_2, \ldots, a_K) \leftarrow$ Randomly select an index number list from $\boldsymbol{S}_i$;
return $(a_1, a_2, \ldots, a_K)$;
```

τ_0 is the lower bound of all pheromone values, and the pheromone updating process will guarantee $\tau_{ij} \geq \tau_0$. Suppose the L largest values of $\tau_{ij} \forall j \in \{1, 2, \ldots, N\} \,\&\, i \neq j$ are $(\tau_{i1}, \tau_{i2}, \ldots, \tau_{iL})$, the diagonal pheromone value τ_{ii} is the mean value of these *Top* pheromone values.

$$\tau_{ii} = \frac{\sum_{t=1}^{Top} \tau_{it}}{Top} \forall i \in \{1, 2, \ldots, N\} \tag{12}$$

The pheromone value will update at each generation. After applying the Non-dominated sorting on the united population R, each individual will obtain a Pareto level *rank*. For example, $rank = 1$ means the individuals are non-dominated solutions, the individuals with smaller *rank* value mean they are more optimized solutions. The individuals with $rank < Th$ can update the pheromone matrix. Th is a threshold value, only the individuals in the first Th-1 Pareto level can update the pheromone value. Suppose Δ_{ij} is the adding value of τ_{ij}. Giving an adequate individual with index number list $(a_1, a_2, \ldots, a_K)$ and investment proportion $(w_1, w_2, \ldots, w_K)$, the adding value Δ_{ij} is calculated as below:

$$\begin{aligned} &\Delta_{a_i a_j} = \Delta_{a_i a_j} + \tfrac{\xi \times w_i w_j \times Th}{rank} \\ &\forall i, j \in \{1, 2, \ldots, K\} \,\&\, i \neq j \end{aligned} \tag{13}$$

Each pair of assets a_i and a_j in the index number list will contribute an increment $\frac{\xi \times w_i w_j \times Th}{rank}$ to adding value $\Delta_{a_i a_j}$. Here, ξ is the increasing rate that controls the numerical value of increment. The increment is proportional to investment proportion and inversely proportional to the Pareto level *rank*.

After all individuals with $rank < Th$ have updated the adding value Δ_{ij}, pheromone value τ_{ij} is updated as below:

$$\begin{aligned} &\tau_{ij} = (1-\rho)\tau_{ij} + \Delta_{ij} \;\forall i, j \in \{1, 2, \ldots, N\} \,\&\, i \neq j \\ &\tau_{ij} = \begin{cases} \tau_{\min} \text{ if } \tau_{ij} < \tau_{\min} \\ \tau_{\max} \text{ if } \tau_{ij} > \tau_{\max} \end{cases} \end{aligned} \tag{14}$$

ρ is the volatilization rate. In each iteration, the primitive pheromone value will only reserve $(1-\rho)$ proportion in the current value. After volatilizing, the current pheromone value τ_{ij} adds with its adding value Δ_{ij} and goes through the boundary check process. This process can limit the pheromone value in an interval $[\tau_{min}, \tau_{max}]$ to guarantee the convergence. After all pheromone values $\tau_{ij}(i \neq j)$ are updated by (14), the diagonal pheromone value τ_{ii} will be updated by (12).

We apply this knowledge accumulation mechanism to discover the internal relation between assets. The appropriate asset combination will release pheromone to strengthen their connection. This positive feedback can direct the asset selection process so that it can construct more effective portfolio. The occurring probability of good combination goes up while the occurring probability of bad combination goes down. The pheromone matrix will demonstrate in the experiment section for an intuitive understanding.

4.4. Knowledge-based proportion allocation

After obtaining an index number list, we need to decide the investment proportion of each asset, which is a continuous optimization problem. The available allocations of investment proportion are endless, which brings a challenge to the algorithm design. EDA can predict the optimal area of the solution space by sampling and statistical learning. We use this mechanism as knowledge information to solve the investment proportion assignment task by assuming the investment proportion subject to the Gaussian distribution.

$$p(w_i) = \frac{1}{\sqrt{2\pi}\overline{\sigma_i}} \exp\left(-\frac{(w_i - \overline{\mu_i})^2}{2\overline{\sigma_i}^2}\right) i \in \{1, 2, \ldots, N\} \tag{15}$$

$\overline{\mu_i}$ represents the mean investment proportion of asset i and $\overline{\sigma_i}$ represents the standard deviation of the investment proportion. $\overline{\mu_i}$ and $\overline{\sigma_i}$ are estimated by the investment proportion of asset i in the optimal solution set. It is different from μ_i and σ_{ij} which are calculated by real historical transaction data. Based on this assumption, we design a *Weight_Assignment*() function to construct the investment proportion encoded by (9). *Weight_Assignment*() function uses Gaussian random sampling method to generate an investment proportion value w_k for asset k. w_k needs to be adjusted by minimum necessary proportion L and the maximum usable proportion U which are calculated by *Boundary_Calculation*() function. After that, w_k is subtracted from *available_weight*. Repeating this process K times, we can obtain an investment proportion list $(w_1, w_2, \ldots, w_K)$ that satisfies all constraints. The detail pseudo-code is given in Algorithm 3.

Algorithm 3: *Weight_Assignment* ( )

```
Input:
•   $K$: cardinality constraints
•   $(\overline{\mu_1}, \overline{\mu_2}, \ldots, \overline{\mu_K})$: mean investment proportion of each asset
•   $(\overline{\sigma_1}, \overline{\sigma_2}, \ldots, \overline{\sigma_K})$: standard deviation of the investment proportion
Output: investment proportion list of solution $x$: $(w_1, w_2, \ldots, w_K)$
available_weight ← 1;
for $k = 1 : K$
  $L$,$U$ ← Boundary_Calculation ( );
  $w_k$ ← random sampling according to the Gaussian distribution
          $N(\overline{\mu_\iota}, \overline{\sigma_\iota}^2)$ in range [0,1];
  if $w_k < L$
    $w_k = L$;
  end
  if $w_k > U$
    $w_k = U$;
  end
  available_weight ←available_weight - $w_k$;
end
return $(w_1, w_2, \ldots, w_K)$;
```

In the proportion allocation process, we use the proportion's distribution as knowledge information which will update during evolution. In the beginning, the mean value $\overline{\mu_i}$ is initialized to the median value of each asset's upper bound and lower bound.

$$\overline{\mu_i} = \frac{ub_i + lb_i}{2} \;\forall i \in \{1, 2, \ldots, N\} \tag{16}$$

The standard deviation $\overline{\sigma_i}$ is initialized to the half-length of each asset's upper bound and lower bound interval.

$$\overline{\sigma_i} = \frac{ub_i - lb_i}{2} \;\forall i \in \{1, 2, \ldots, N\} \tag{17}$$

After using *Individual_Selection*() function to generate a new preserve population S_{i+1}, we can update $\overline{\mu_i}$ and $\overline{\sigma_i}$ according to the individuals in S_{i+1}. Suppose asset i is selected by H individuals ($H \in \{0, 1, 2, \ldots, Popsize\}$) and its investment proportion is $(w_1, w_2, \ldots, w_H)$.

The update rule of the distribution information is given below:

$$\overline{\mu_i} = \begin{cases} (1-\eta)\overline{\mu_i} + \frac{\eta}{H}\sum\limits_{h=1}^{H} w_h & \text{if } H \geq 1 \\ (1-\eta)\overline{\mu_i} + \eta \times lb_i & \text{if } H = 0 \end{cases} \tag{18}$$

$$\overline{\sigma_i} = \begin{cases} (1-\eta)\overline{\sigma_i} + \eta \times \left(\sqrt{\frac{\sum_{h=1}^{H}(w_h - \overline{\mu_i})^2}{H}} + \varepsilon\right) & \text{if } H \geq 1 \\ (1-\eta)\overline{\sigma_i} + \eta \times (ub_i - lb_i) & \text{if } H = 0 \end{cases} \tag{19}$$

where η is the smoothing parameter and $\eta \in (0, 1]$. It can avoid violent fluctuation of distribution information and control the convergence speed. If asset i is selected by at least one individual ($H \geq 1$), the mean value $\overline{\mu_i}$ is the previous mean value, with a weight $(1-\eta)$, add with the mean value estimate from the preserve population S_{i+1}, with a weight η. The standard deviation $\overline{\sigma_i}$ is the previous standard deviation value, with a weight $(1-\eta)$, add with the standard deviation value estimate from the preserve population S_{i+1}, with a weight η. To avoid the standard deviation converges to zero, the estimated standard deviation has added a small ε value. If no individual selects asset i ($H = 0$), the estimated mean value will be set as the lower bound of asset i. As thus, the mean value of fewer selected assets will gradually converge to their lower bound lb_i. The estimated standard deviation value will be set as the length of the upper bound and lower bound interval of asset i. In this way, the standard deviation $\overline{\sigma_i}$ will gradually converge to the bound length $(ub_i - lb_i)$, which can remain a relatively loose distribution range of proportion assignment.

The investment proportion assignment process can utilize the distribution information from the preserve population S_{i+1}. Since the preserve population keeps updating, the distribution information is constantly renewing. The heavily invested assets, name as core assets, have a larger mean value $\overline{\mu_i}$. Meanwhile, the marginal assets have a smaller mean value $\overline{\mu_i}$. The distribution information will demonstrate in the experiment section for an intuitive understanding.

4.5. Constructive constraint handling method

Constraint handling is another challenge of the portfolio optimization problem with cardinality constraint. The most commonly used methods include repair mechanism, penalty function, and infeasibility tolerance, but they still exist some problems for solving MVPO problem. The repair mechanism is normally very complicated and difficult to deal with the irreparable solution. For example, a solution without any asset from a particular group is irreparable under the grouping constraint. A feasible way is trying to swap assets which will further increase the complexity of the repair mechanism. The penalty function and infeasibility tolerance method also exist a weakness, they cannot ensure to get enough feasible solutions. Based on these above issues, we apply a constructive approach to handle the constraints. This approach bases on the idea of construct solutions within the feasible region and includes two functions: *verification*() and *Boundary_Calculation*().

In the knowledge-based asset selection process, we apply *verification*() function to guarantee the selected assets have a feasible investment proportion allocate space. The pseudo-code of the verification process is given in Algorithm 4.

Algorithm 4: *verification* ( )

```
Input:
•   K: cardinality constraint
•   M: group number
•   $(ub_1, ub_2, \ldots, ub_K)$: upper bound of the selected assets
•   $(lb_1, lb_2, \ldots, lb_K)$: lower bound of the selected assets
•   $(g_1, g_2, \ldots, g_K)$: group information of the selected assets
•   $(ug_1, ug_2, \ldots, ug_M)$: upper bound of each group
•   $(lg_1, lg_2, \ldots, lg_M)$: lower bound of each group
Output: Boolean
if $\sum_{k=1}^{K} lb_k > 1$ return false;
if $\sum_{m=1}^{M} lg_m > 1$ return false;
for m=1:M
    if $\sum_{k=1 \&\& g_k==m}^{K} ub_k < lg_m$ return false;
    if $\sum_{k=1 \&\& g_k==m}^{K} lb_k > ug_m$ return false;
end
if $\sum_{m=1}^{M} \max(\sum_{k=1 \&\& g_k==m}^{K} lb_k, lg_m) > 1$ return false;
if $\sum_{m=1}^{M} \min(\sum_{k=1 \&\& g_k==m}^{K} ub_k, ug_m) < 1$ return false;
return true;
```

Firstly, *verification*() function verifies the sum of all assets' lower bound and the sum of all groups' lower bound. If it greater than one, this asset combination will violate the budget constraint. Secondly, it verifies the boundary of each group. The verify process returns false if the lower group bound greater than the sum of included assets' upper bound or the upper group bound smaller than the sum of included assets' lower bound. Thirdly, it finds out exact lower bound and upper bound of each group. The asset combination will violate budget constraint if the sum of exact lower bound greater than one or the sum of exact upper bound smaller than one. If the asset combination passes all verification, *verification*() function will return a true value, which means the asset combination has a feasible solution space under all constraints.

In the knowledge-based proportion allocation process, we apply *Boundary_Calculation*() function to calculate the minimum necessary proportion L and the maximum usable proportion U of asset k. This function can guarantee the investment proportion assignment process does not violate the constraints. The core idea of this constructive approach is maintaining a feasible assign space for remaining assets. If we assign an appropriate proportion for current asset k and remain a feasible assign space for the rest K-k assets in each step, we can construct an investment proportion list that satisfies all constraints. The detailed pseudo-code of the boundary calculation process is given in Algorithm 5.

Algorithm 5: *Boundary_Calculation* ( )

```
Input:
•   K: cardinality constraints
•   M: group number
•   k: current assign asset
•   available_weight: current unassigned proportion
•   $(w_1, w_2, \ldots, w_{k-1}, 0, \ldots, 0)$: current proportion assignment list
•   $(ub_1, ub_2, \ldots, ub_K)$: upper bound of the selected assets
•   $(lb_1, lb_2, \ldots, lb_K)$: lower bound of the selected assets
•   $(g_1, g_2, \ldots, g_K)$: group information of the selected assets
•   $(ug_1, ug_2, \ldots, ug_M)$: upper bound of each group
•   $(lg_1, lg_2, \ldots, lg_M)$: lower bound of each group
Output: L, U
$(c_1, c_2, \ldots, c_M) \leftarrow (ug_1, ug_2, \ldots, ug_M)$;
$(d_1, d_2, \ldots, d_M) \leftarrow (lg_1, lg_2, \ldots, lg_M)$;
$(e_1, e_2, \ldots, e_M) \leftarrow (0, 0, \ldots, 0)$;
$(f_1, f_2, \ldots, f_M) \leftarrow (0, 0, \ldots, 0)$;
L = U = available_weight;
if k < K
    if k > 1
        for i = 1 : (k-1)
            $c_{g_i} = c_{g_i} - w_i$;
            $d_{g_i} = d_{g_i} - w_i$;
        end
    end
    for i = (k+1) : K
        $e_{g_i} = e_{g_i} + ub_i$;
        $f_{g_i} = f_{g_i} + lb_i$;
    end
    for m = 1 : M
        if m != $g_k$
            L = L – min ($c_m$, $e_m$);
            U= U – max ($d_m$, $f_m$);
        end
    end
    L = L - $e_{g_k}$;
    U = U - $f_{g_k}$;
    L = max (L, $d_{g_k} - e_{g_k}$, $lb_k$);
    U = min (U, $c_{g_k} - f_{g_k}$, $ub_k$);
end
return L, U
```

c_m is the remaining usable proportion of group m which is calculated by subtracting the assigned proportion from the upper group bound ug_m. e_m is the sum of remaining assets' upper bound of group m. The maximum usable proportion of group m is the smaller value of c_m and e_m. In order to meet budget constraint, we need to make sure the unassigned proportion *available_weight* can be exhausted. We assume that each group except group g_k takes the maximum usable proportion and the remaining assets in group g_k take their upper bound value e_{g_k}. After that, the remaining proportion should be assigned to asset k to exhaust *available_weight*. In order to meet grouping constraint, we need to make sure the unreached lower group bound value d_{g_k} can be fulfilled. We assume that the remaining assets in group g_k take their upper bound value e_{g_k} from d_{g_k}, the remaining proportion should be assigned to asset k to fulfill the lower group bound constraint. In order to meet boundary constraint, the investment proportion of asset k should greater than lb_k. The minimum necessary proportion L is the maximum value of the above three values.

d_m is the unfulfilled proportion of group m which is calculated by subtracting the assigned proportion from the lower group bound lg_m. f_m is the sum of the remaining assets' lower bound of group m. The minimum necessary proportion of group m is the bigger value of d_m and f_m. To fulfill budget constraint, we assume that each group takes the minimum necessary proportion and the remaining assets in group g_k take their lower bound value f_{g_k}. After that, the remaining proportion is the maximum proportion that can be assigned to asset k. In this way, the remaining K-k assets still have a feasible assign space. We assume that the remaining assets in group g_k take their lower bound value f_{g_k} from c_{g_k}, the remaining proportion is the maximum proportion of asset k under grouping constraint. The investment proportion of asset k should smaller than ub_k under boundary constraint. The maximum usable proportion U is the minimum value of the above three values. After finding out the minimum necessary proportion L and the maximum usable proportion U of asset k, $Weight\_Assignment()$ function can adjust the proportion value w_k for asset k to satisfy all constraints.

5. Experimental studies

5.1. Experimental settings

In order to validate the effectiveness of KC-EDA, experiments are done on three real-world datasets: SZ50, SZ180, and SZ380. These datasets are generated by using historical transaction information from January 2014 to December 2018 on Shanghai Stock Exchange, China. Two groups of experiments are implemented. The first group is to demonstrate the effect of KC-EDA. The second group is to compare KC-EDA with three other approaches for MVPO problem with cardinality constraints. The first one is genetic algorithms with random assorting recombination (RAR-GA) [39], which uses random assorting recombination operator to design a crossover operation to keep cardinality constraint during evolution. RAR-GA uses crossover probability 1.00 and mutation probability 0.01. The second one is estimation of distribution algorithm with pruning heuristics (P-EDA) [33], which uses pruning heuristics to reduce the number of candidate assets and improve efficiency. P-EDA applies estimation of multivariate normal algorithm (EMNA) and pruning heuristic threshold as $lb_i/2$. If the estimated mean value of asset i smaller than half of its lower bound value, this asset will be pruned. These two algorithms consider budget constraint, boundary constraint [0.01,1], and cardinality constraint, so they are organized with the proposed algorithm as one subgroup of comparison studies. The third one is genetic algorithms with k-means cluster method (Kmean-GA) [3], which uses k-means cluster method to handle cardinality constraint and designs a particular method to repair solutions under proportion constraints. Kmean-GA uses crossover probability 0.5 and mutation probability 0.01. It fully considers budget constraint, boundary constraint [0.01,1], grouping constraint, and cardinality constraint, so it is organized with the proposed algorithm as the second subgroup of comparison studies. The parameters of KC-EDA include: pheromone value interval is set to [1, 100], volatilization rate ρ is set to 0.05, increasing rate ξ is set to 10^3, threshold value Th is set to 10, maximum number Top used in (12) is set to 10, and smoothing parameter η is set to 0.25. The maximum trying time T in Algorithm 2 takes different values {100, 10, 0} during evolution in order to balance combination discover and proportion optimize. The population size of all algorithms is set to 200, but the maximum generations depend on the number of candidate assets. The experimental configurations of each algorithm are listed in Table 1.

Table 1
Experimental configurations of each algorithm.

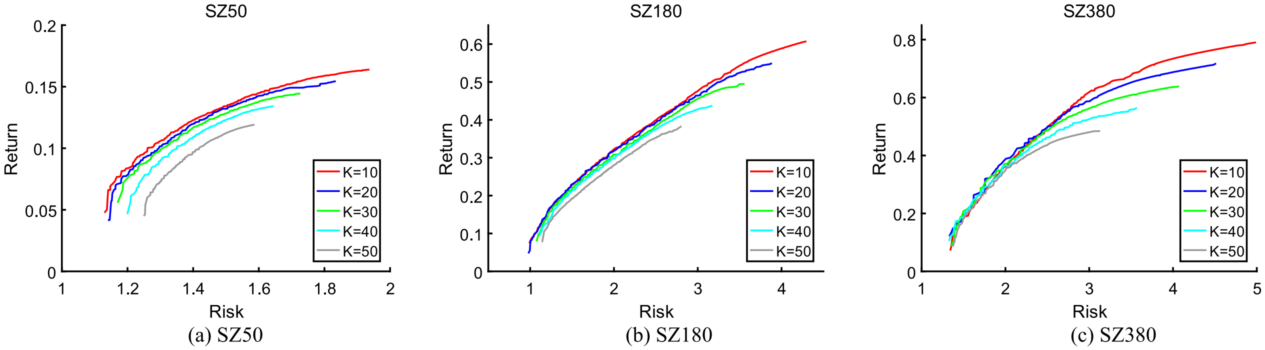
Algorithm	Parameter name	Parameter value
RAR-GA	crossover probability	1.00
	mutation probability	0.01
P-EDA	EDA model	EMNA
	pruning threshold	$lb_i/2$
Kmean-GA	crossover probability	0.5
	mutation probability	0.01
KC-EDA	pheromone value interval	[1,100]
	volatilization rate ρ	0.05
	increasing rate ξ	10^3
	threshold value Th	10
	maximum number Top	10
	smoothing parameter η	0.25
	maximum trying time T	100 for [0, 1/2*Gen*] 10 for [1/2 *Gen*, 3/4 *Gen*] 0 for [3/4 *Gen*, *Gen*]
Common parameters	population size *Pop*	200
	maximum generations *Gen*	200 for SZ50 400 for SZ180 600 for SZ380

5.2. Effect of KC-EDA

In MVPO problem with cardinality constraint, the main task is to select K kinds of assets and assigning their investment proportion according to the constraints. The objectives are obtaining higher return and lower risk. The proposed algorithm applies a hybrid design of ACO and EDA to construct portfolios based on knowledge information. The knowledge information includes the assets combinatorial information and proportion distribute information, which keeps updating during evolution. In addition, a constructive approach is designed to handle the constraints, which can construct the portfolio in the feasible solution space. To validate the effectiveness of our algorithm, we run the algorithm in three datasets with cardinality constraint K from 10 to 50. Boundary constraint is set to [0.01, 1]. Without setting the lower bound of assets, the investment proportion of the selected assets can be 0, which will counteract cardinality constraint. All assets are assigned to only one group, and grouping constraint is set to [0, 1]. This experiment can facilitate the effect of cardinality constraint. The Pareto fronts obtained by our algorithm are shown in Fig. 2. For ease of observation, we use the daily rate of return to express portfolio's return and the standard deviation to express portfolio's risk in the diagram.

From the figure, we can find out that KC-EDA can obtain a relatively complete Pareto front with different cardinality constraint values. We can observe that with the increment of cardinality constraint K, Pareto fronts gradually deviate from the high-risk area. Because higher cardinality constraint value will decentralize investment and reduce the investment proportion of risky high-yield assets. For small dataset SZ50, when K takes value 40 or 50, nearly all candidate assets are included in the portfolio. But the Pareto front is inferior to those with smaller K values. This phenomenon names as excessive decentralization. So, the cardinality constraint value K should be relatively small.

To demonstrate the knowledge information, we use the arc diagram to show the pheromone matrix after iteration with $K=10$ in Fig. 3. As shown in the figure, the size of points is proportional to the value of diagonal pheromone value τ_{ii}. For ease of observation, we sort all points from largest to smallest and paint them according to their value. The arcs link between each pair of points represents their correlate pheromone value τ_{ij}. We divide correlate pheromone value τ_{ij} into four levels and paint by different colors: [0, 25] gray, [25, 50] blue, [50, 75] green, [75,



(a) SZ50 (b) SZ180 (c) SZ380

Fig. 2. The Pareto fronts obtain by the KC-EDA with different cardinality constraints K.

100] yellow. For dataset SZ50, we draw all fifty points, but for SZ180 and 380, we only draw the first fifty points which will not affect observation. We can visually see that only part of points have big pheromone value, which means they are important and frequently used (points 1st to 25th in SZ50, points 1st to 42th in SZ180, and points 1st to 34th in SZ380). And these frequently used points have more arcs with big pheromone value (yellow, green, and blue arcs). It indicates those points have stronger interrelation and commonly select together. For the points with small values, they have gray arcs with almost all points, which means they do not have prominent relations with any points. By using pheromone as knowledge information, we can discover the internal relation between assets and increase the occurring probability of suitable combinations. The number of core assets only occupies a small part of candidate assets as showed in SZ180 and SZ380.

We also demonstrate the distribution information of investment proportion in the optimal solution set. The mean value $\overline{\mu_i}$ is represented by a blue square and the standard deviation $\overline{\sigma_i}$ is represented by a blue vertical line. From the figure, we can see the frequently used points have relatively big mean values. It indicates those points are always allocated with a high investment proportion. Take point 1st in SZ50 as an example, it has the biggest mean value than any other points and a relatively big standard deviation, which means its investment proportion is various in different portfolios. For the points with a small mean value, they have a relatively big standard deviation, which means they are seldomly used in the portfolio.

5.3. Comparison results and discussion

The proposed algorithm is compared with RAR-GA and P-EDA with $K = 10$, 20, and 30 in three datasets. Boundary constraint is set as [0.01, 1], and all assets belong to one group with bound [0, 1] because these two algorithms cannot handle grouping constraint. The results are shown in Fig. 4. From the figure, the Pareto fronts obtained by RAR-GA (black line) are worse than the other two algorithms in optimization degree and smoothness. The heuristic strategy of RAR-GA is trying to preserve the proper asset combinations in individuals' encoding, so population size will limit the accumulation and transmission of heuristic information. The Pareto fronts obtain by the P-EDA (blue line) and KC-EDA (red line) are relatively close, but the smoothness of KC-EDA is better. The pruning strategy of P-EDA can rapidly reduce the number of candidate assets. It may converge to local optimality because candidate assets are narrow. On the other hand, KC-EDA can discover the internal relation between assets and preserve them in the pheromone matrix, which can guide the asset selection. After finding out the proper combination, KC-EDA uses the reuse strategy $T = 0$ to modulate the investment proportion of each portfolio. In this way, KC-EDA can find out a more optimal solution and lead to a smooth Pareto front.

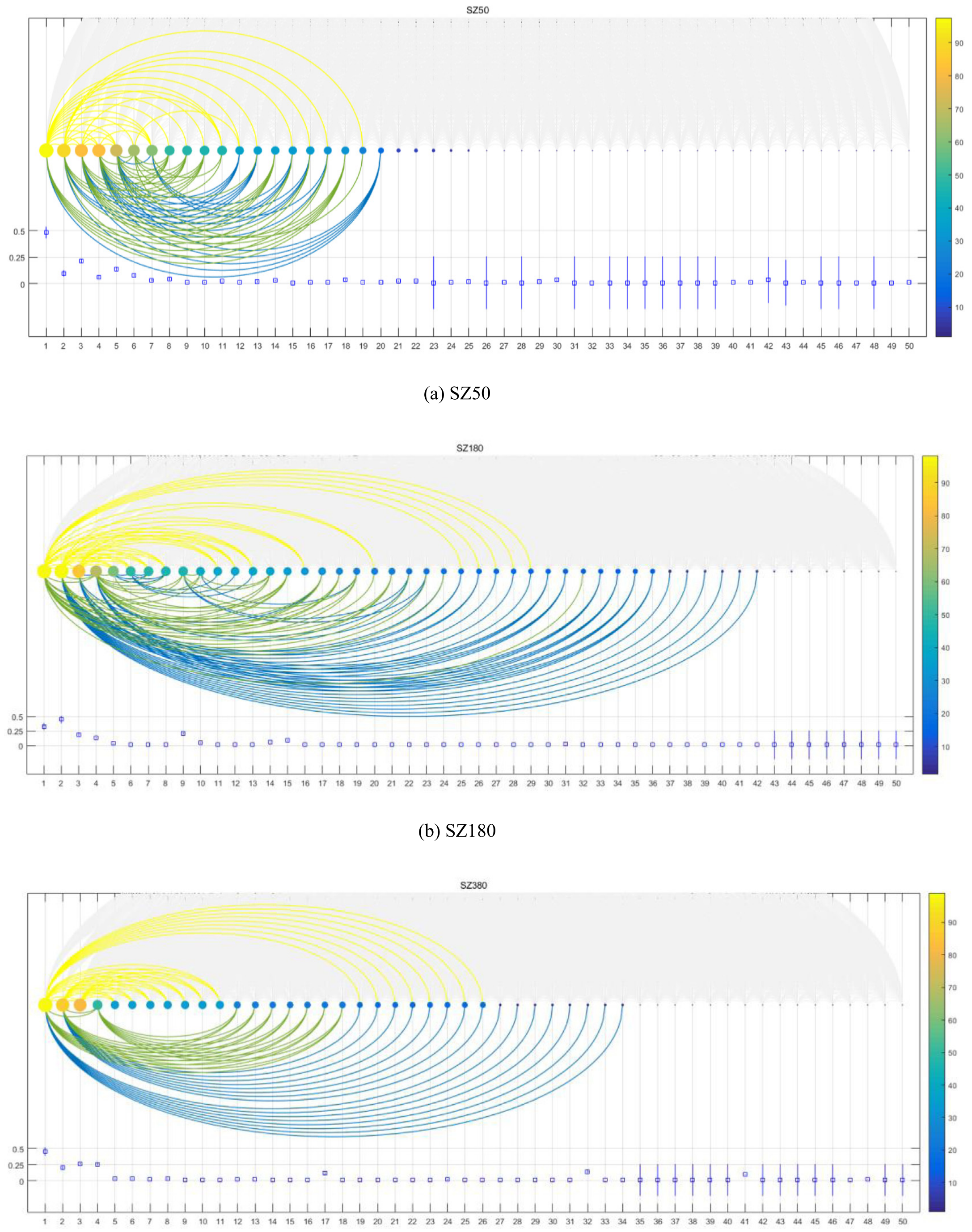
We also compared the proposed algorithm with Kmean-GA with $K = 10$, 20, and 30 in three datasets. Boundary constraint is set as [0.01, 1] and candidate assets are divided into four groups: banking [0.01, 0.3], technology [0.01, 0.4], energy [0.01, 0.3] and others [0, 1]. The results are shown in Fig. 5. For ease of comparison, we also draw the Pareto front obtain by KC-EDA without grouping constraint (green line). The Pareto front obtain by the Kmean-GA (blue line) is relatively worse. Kmean-GA randomly selects one asset from one cluster to handle cardinality constraints, but it may be unable to separate candidate assets into appropriate clusters. Some highly relative assets may be divided into the same cluster and cannot be selected simultaneously. KC-EDA still can obtain an optimal Pareto front, the red line almost overlaps with the green line in SZ50 and SZ180. But in SZ380, the red line diverges from the green line in the high-risk region. Without grouping constraint, the upper investment proportion of risky high-yield assets is 1, but it is limited by the upper bound of its group, which results in the deviation.

We further calculate the average hypervolume (HV) of each algorithm over 30 independent runs and the results are shown in Table 2. The (min (σ), max (μ)) point of each dataset is set as the reference point, a small average hypervolume denotes the better performance. In the first group of experiment, KC-EDA performs better than RAR-GA and P-EDA in all datasets. In the second group of experiment, KC-EDA performs better than Kmean-GA in all datasets. These results have passed the Wilcoxon's rank-sum test at a 0.01 significance level, which shows the effectiveness of KC-EDA.

Discussion

(1) In RAR-GA, the asset combination information is preserved in the encoding of each individual. The information sharing can only happen in the crossover operation which limits the scope and speed of information sharing. In KC-EDA, we apply the pheromone matrix of ACO as a knowledge information carrier. This mechanism can collect knowledge information from all individuals and share the knowledge information to all individuals which helps KC-EDA obtain better solutions.

(2) In P-EDA, it uses the pruning strategy to rapidly reduce the number of candidate assets. This mechanism will give up part of the searching space together with the optimal solutions within the abandoned space. In KC-EDA, the pheromone matrix can discover and preserve the internal relation between assets.



(a) SZ50

(b) SZ180

(c) SZ380

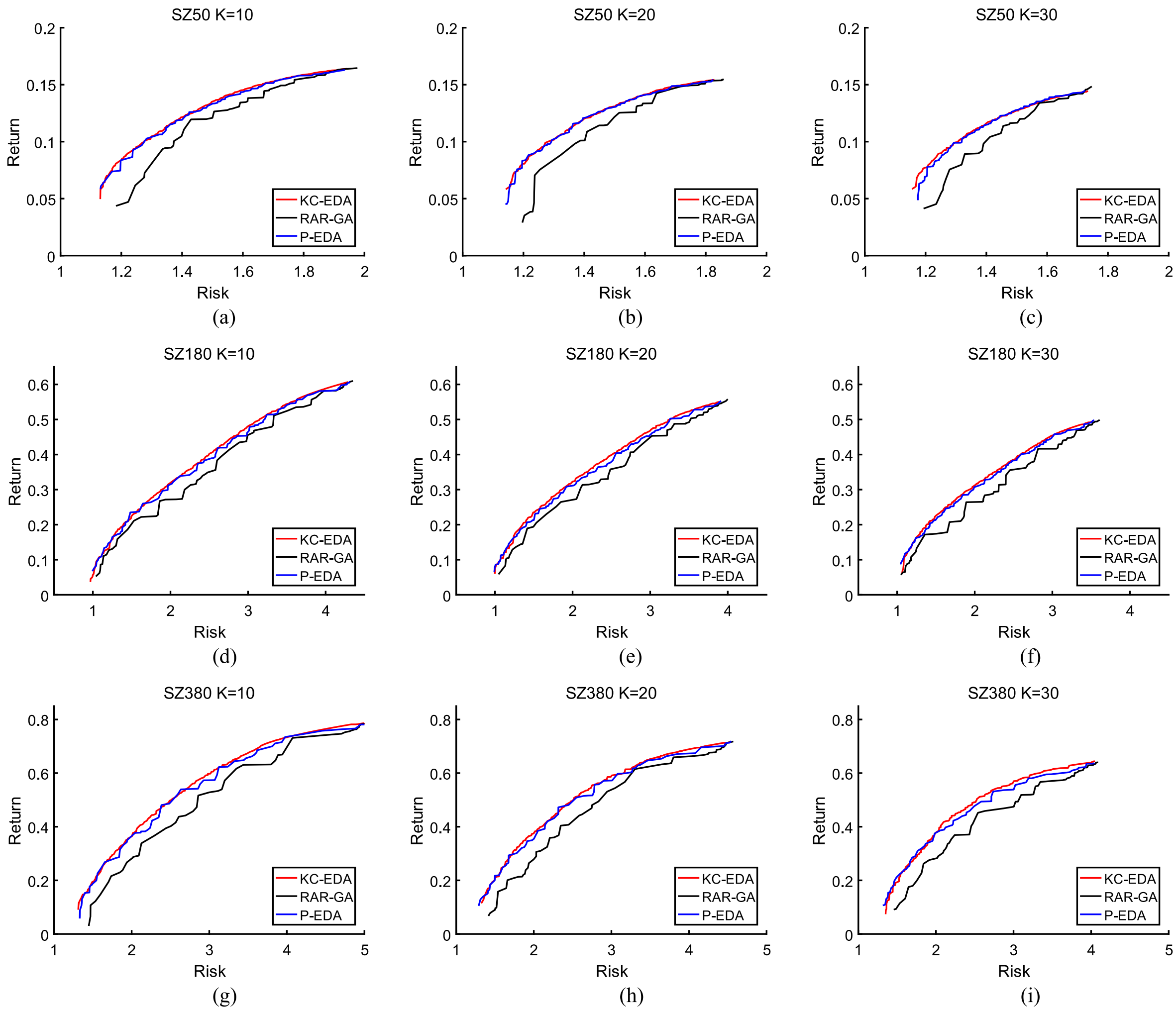
Fig. 3. Visualization of knowledge information with K=10.

It only reduces the possibility of those seldom-used assets rather than abandon them which keeps the integrity of searching space. Therefore, KC-EDA can obtain smooth Pareto fronts.

(3) In K-means GA, it uses the clustering method to handle cardinality constraints. It is hard to correctly classify candidate assets without knowledge information from the optimal solutions. The KC-EDA can accumulate knowledge information while searching the optimal solutions. Meanwhile, the constructive approach of KC-EDA is more high-efficiency than the repair approach of K-means GA. Because the constructive approach enables KC-EDA to carry on searching only in the feasible space.

6. Conclusion

In this paper, a knowledge-based constructive estimation of distribution algorithm (KC-EDA) is proposed to solve the portfolio



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

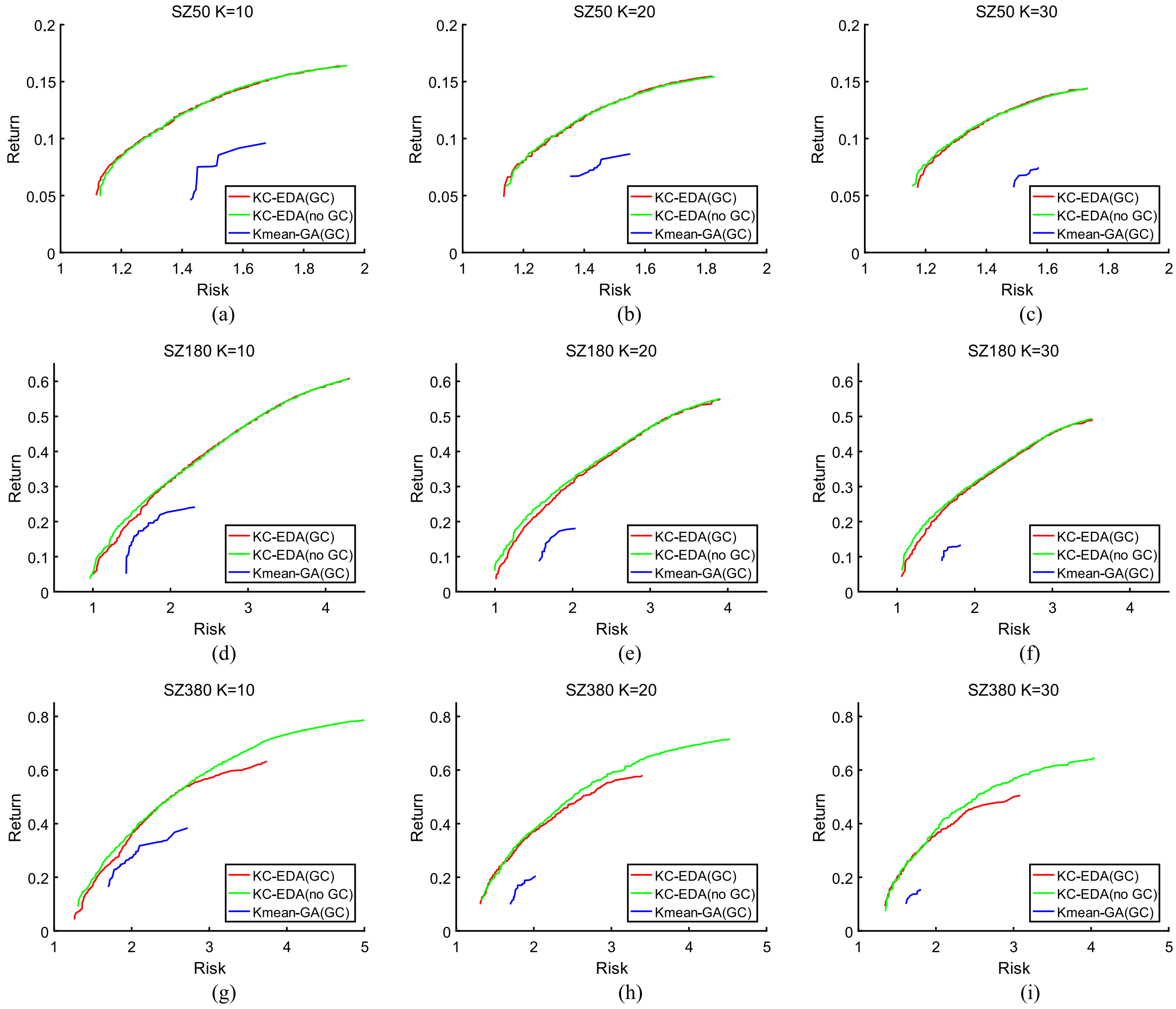
Fig. 4. The comparison between KC-EDA, RAR-GA and P-EDA.

Table 2
Comparison of the HV values among the RAR-GA without grouping constraint (no GC), P-EDA without grouping constraint (no GC), KC-EDA without grouping constraint (no GC), Kmean-GA with grouping constraint (GC) and KC-EDA with grouping constraint (GC).

Dataset	K	RAR-GA (no GC)	P-EDA (no GC)	KC-EDA (no GC)	Dataset	K	Kmean-GA (GC)	KC-EDA (GC)
SZ50	10	0.23749	0.22543	**0.22289**$^{*\wedge}$	SZ50	10	0.29562	**0.22260**$^{+}$
	20	0.24010	0.22893	**0.22821**$^{*\wedge}$		20	0.30316	**0.22788**$^{+}$
	30	0.24451	0.23594	**0.23550**$^{*\wedge}$		30	0.30921	**0.23522**$^{+}$
SZ180	10	1.6098	1.5194	**1.4886**$^{*\wedge}$	SZ180	10	2.3237	**1.5055**$^{+}$
	20	1.6584	1.5674	**1.5530**$^{*\wedge}$		20	2.5613	**1.5648**$^{+}$
	30	1.7325	1.6512	**1.6435**$^{*\wedge}$		30	2.6714	**1.6560**$^{+}$
SZ380	10	2.4561	2.2718	**2.1986**$^{*\wedge}$	SZ380	10	2.5866	**2.4946**$^{+}$
	20	2.5289	2.3593	**2.3092**$^{*\wedge}$		20	3.9125	**2.6181**$^{+}$
	30	2.6543	2.5326	**2.4769**$^{*\wedge}$		30	4.0690	**2.7800**$^{+}$

'*' indicates KC-EDA (no GC) is significantly better than RAT-GA (no GC) according to Wilcoxon's rank sum test at 0.01 significance level.
'^' indicates KC-EDA (no GC) is significantly better than P-EDA (no GC) according to Wilcoxon's rank sum test at 0.01 significance level.
'+' indicates KC-EDA (GC) is significantly better than Kmean-GA (GC) according to Wilcoxon's rank sum test at 0.01 significance level.

optimization problem with cardinality constraints. The knowledge-based portfolio construction consists of three components: asset selection, proportion allocation and constraint handling. KC-EDA uses a hybrid design of ACO and EDA to construct portfolio based on the knowledge information. It considers the relationship between assets and keeps discovering the interrelation during evolution. It can utilize the distribution information from the historical solution set and learn from the proper investment. Rather than using build and repair mechanism to adjust the solution, we apply a constructive approach to construct portfolios under constraints. This knowledge-based portfolio construction is incorporated into the multi-objective evolutionary framework, and the



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 5. The comparison between KC-EDA with grouping constraint, KC-EDA without grouping constraint and Kmean-GA with grouping constraint.

experimental results show that KC-EDA is effective in solving the portfolio optimization problem with cardinality constraints.

In the future, more real-world constraints can take into consideration. Besides, the proposed algorithm can extend to solve the multi-period portfolio optimization problem.

CRediT authorship contribution statement

Zhi-Xuan Zhang: Methodology, Software, Writing – original draft. **Wei-Neng Chen:** Conceptualization, Methodology, Writing – review & editing, Supervision. **Xiao-Min Hu:** Validation, Investigation, Writing – review & editing.

Declaration of competing interest

This manuscript is the authors' original work and has not been published nor has it been submitted simultaneously elsewhere. All authors have checked the manuscript and have agreed to the submission.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61976093, 62272108 and Guangdong Regional Joint Foundation Key Project, China 2022B1515120076.

References

[1] H. Markowitz, Portfolio selection, Harry Markowitz Sel. Work. 7 (1) (2009) 15–30, http://dx.doi.org/10.2307/2975974.

[2] D. Lin, X. Li, M. Li, A Genetic Algorithm for Solving Portfolio Optimization Problems with Transaction Costs and Minimum Transaction Lots, in: Lecture Notes in Computer Science, vol. 3612, 2005, pp. 808–811, http://dx.doi.org/10.1007/11539902_99, no. PART III.

[3] G.A.V. Pai, T. Michel, Evolutionary optimization of constrained k-means clustered assets for diversification in small portfolios, IEEE Trans. Evol. Comput. 13 (5) (2009) 1030–1053, http://dx.doi.org/10.1109/TEVC.2009.2014360.

[4] Q. Ni, X. Yin, K. Tian, Y. Zhai, Particle swarm optimization with dynamic random population topology strategies for a generalized portfolio selection problem, Nat. Comput. 16 (1) (2017) 31–44, http://dx.doi.org/10.1007/s11047-016-9541-x.

[5] A. Kumar, S. Yadav, P. Gupta, M. Mehlawat, A credibilistic multiobjective multiperiod efficient portfolio selection approach using data envelopment analysis, IEEE Trans. Eng. Manag. (2021) 1–15, http://dx.doi.org/10.1109/TEM.2021.3072543.

[6] K.P. Anagnostopoulos, G. Mamanis, A portfolio optimization model with three objectives and discrete variables, Comput. Oper. Res. 37 (7) (2010) 1285–1297, http://dx.doi.org/10.1016/j.cor.2009.09.009.

[7] M. Ehrgott, K. Klamroth, C. Schwehm, An MCDM approach to portfolio optimization, European J. Oper. Res. 155 (3) (2004) 752–770, http://dx.doi.org/10.1016/S0377-2217(02)00881-0.

[8] K. Liagkouras, K. Metaxiotis, A new efficiently encoded multiobjective algorithm for the solution of the cardinality constrained portfolio optimization problem, Ann. Oper. Res. 267 (1–2) (2018) 281–319, http://dx.doi.org/10.1007/s10479-016-2377-z.

[9] S. Arnone, A. Loraschi, A. Tettamanzi, A genectic approach to portfolio selection, Neural Netw. World 3 (6) (1993) 597–604.

[10] T.J. Chang, S.C. Yang, K.J. Chang, Portfolio optimization problems in different risk measures using genetic algorithm, Expert Syst. Appl. 36 (7) (2009) 10529–10537, http://dx.doi.org/10.1016/j.eswa.2009.02.062.

[11] W. Yan, R. Miao, S. Li, Multi-period semi-variance portfolio selection: Model and numerical solution, Appl. Math. Comput. 194 (1) (2007) 128–134, http://dx.doi.org/10.1016/j.amc.2007.04.036.

[12] P.C. Lin, P.C. Ko, Portfolio value-at-risk forecasting with GA-based extreme value theory, Expert Syst. Appl. 36 (2 PART 1) (2009) 2503–2512, http://dx.doi.org/10.1016/j.eswa.2008.01.086.

[13] S.C. Chiam, K.C. Tan, A. Al Mamum, Evolutionary multi-objective portfolio optimization in practical context, Int. J. Autom. Comput. 5 (1) (2008) 67–80, http://dx.doi.org/10.1007/s11633-008-0067-2.

[14] F.C. Duran, C. Cotta, A.J. Fernández, On the use of sharpe's index in evolutionary portfolio optimization under markowitz's model, in: Adaptive and Emergent Behaviour and Complex Systems - Proceedings of the 23rd Convention of the Society for the Study of Artificial Intelligence and Simulation of Behaviour, AISB 2009, 2009, pp. 9–14.

[15] Y.H. Chou, S.Y. Kuo, Y.C. Jiang, A novel portfolio optimization model based on trend ratio and evolutionary computation, IEEE Trans. Emerg. Top. Comput. Intell. 3 (4) (2019) 337–350, http://dx.doi.org/10.1109/TETCI.2018.2868939.

[16] E. Fernandez, J. Navarro, E. Solares, C.C. Coello, A novel approach to select the best portfolio considering the preferences of the decision maker, Swarm Evol. Comput. 46 (2019) 140–153, http://dx.doi.org/10.1016/j.swevo.2019.02.002.

[17] C.R.B. Azevedo, F. Von Zuben, Learning to anticipate flexible choices in multiple criteria decision-making under uncertainty, IEEE Trans. Cybern. 46 (3) (2016) 778–791, http://dx.doi.org/10.1109/TCYB.2015.2415732.

[18] K. Metaxiotis, K. Liagkouras, Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review, Expert Syst. Appl. 39 (14) (2012) 11685–11698, http://dx.doi.org/10.1016/j.eswa.2012.04.053.

[19] A. Ponsich, A.L. Jaimes, C.A.C. Coello, A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications, IEEE Trans. Evol. Comput. 17 (3) (2013) 321–344, http://dx.doi.org/10.1109/TEVC.2012.2196800.

[20] C.B. Kalayci, O. Ertenlice, M.A. Akbay, A comprehensive review of deterministic models and applications for mean–variance portfolio optimization, Expert Syst. Appl. 125 (2019) 345–368, http://dx.doi.org/10.1016/j.eswa.2019.02.011.

[21] F. Streichert, H. Ulmer, A. Zell, Comparing Discrete and Continuous Genotypes on the Constrained Portfolio Selection Problem, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3103, 2004, pp. 1239–1250, http://dx.doi.org/10.1007/978-3-540-24855-2_131.

[22] J. Branke, B. Scheckenbach, M. Stein, K. Deb, H. Schmeck, Portfolio optimization with an envelope-based multi-objective evolutionary algorithm, European J. Oper. Res. 199 (3) (2009) 684–693, http://dx.doi.org/10.1016/j.ejor.2008.01.054.

[23] K. Lwin, R. Qu, G. Kendall, A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization, Appl. Soft Comput. 24 (2014) 757–772, http://dx.doi.org/10.1016/j.asoc.2014.08.026.

[24] H. Soleimani, H.R. Golmakani, M.H. Salimi, Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm, Expert Syst. Appl. 36 (3 PART 1) (2009) 5058–5063, http://dx.doi.org/10.1016/j.eswa.2008.06.007.

[25] T.J. Chang, N. Meade, J.E. Beasley, Y.M. Sharaiha, Heuristics for cardinality constrained portfolio optimisation, Comput. Oper. Res. 27 (13) (2000) 1271–1302, http://dx.doi.org/10.1016/S0305-0548(99)00074-X.

[26] I. Bavarsad Salehpoor, S. Molla-Alizadeh-Zavardehi, A constrained portfolio selection model at considering risk-adjusted measure by using hybrid meta-heuristic algorithms, Appl. Soft Comput. 75 (2019) 233–253, http://dx.doi.org/10.1016/j.asoc.2018.11.011.

[27] P. Skolpadungket, K. Dahal, N. Harnpornchai, Portfolio optimization using multi-objective genetic algorithms, in: 2007 IEEE Congress on Evolutionary Computation, CEC 2007, 2007, pp. 516–523, http://dx.doi.org/10.1109/CEC.2007.4424514.

[28] Y. Crama, M. Schyns, Simulated annealing for complex portfolio selection problems, European J. Oper. Res. 150 (3) (2003) 546–571, http://dx.doi.org/10.1016/S0377-2217(02)00784-1.

[29] Y. Xia, B. Liu, S. Wang, K.K. Lai, A model for portfolio selection with order of expected returns, Comput. Oper. Res. 27 (5) (2000) 409–422, http://dx.doi.org/10.1016/S0305-0548(99)00059-3.

[30] F.D. Paiva, R.T.N. Cardoso, G.P. Hanaoka, W.M. Duarte, Decision-making for financial trading: A fusion approach of machine learning and portfolio selection, Expert Syst. Appl. 115 (2019) 635–655, http://dx.doi.org/10.1016/j.eswa.2018.08.003.

[31] Y. Dai, Z. Qin, Multi-period uncertain portfolio optimization model with minimum transaction lots and dynamic risk preference, Appl. Soft Comput. 109 (2021) http://dx.doi.org/10.1016/j.asoc.2021.107519.

[32] S.S. Meghwani, M. Thakur, Multi-objective heuristic algorithms for practical portfolio optimization and rebalancing with transaction cost, Appl. Soft Comput. 67 (2018) 865–894, http://dx.doi.org/10.1016/j.asoc.2017.09.025, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) held as part of IEEE World Congress on Computational Intelligence (IEEE WCCI).

[33] R. Ruiz-Torrubiano, A. Suárez, Hybrid approaches and dimensionality reduction for portfolio selection with cardinality constraints, IEEE Comput. Intell. Mag. 5 (2) (2010) 92–107, http://dx.doi.org/10.1109/MCI.2010.936308.

[34] Q. Liu, C. Dang, T. Huang, A one-layer recurrent neural network for real-time portfolio optimization with probability criterion, IEEE Trans. Cybern. 43 (1) (2013) 14–23, http://dx.doi.org/10.1109/TSMCB.2012.2198812.

[35] H.R. Golmakani, M. Fazel, Constrained portfolio selection using particle swarm optimization, Expert Syst. Appl. 38 (7) (2011) 8327–8335, http://dx.doi.org/10.1016/j.eswa.2011.01.020.

[36] M. Corazza, G. Fasano, R. Gusso, Particle swarm optimization with non-smooth penalty reformulation, for a complex portfolio selection problem, Appl. Math. Comput. 224 (2013) 611–624, http://dx.doi.org/10.1016/j.amc.2013.07.091.

[37] C.B. Kalayci, O. Ertenlice, H. Akyer, H. Aygoren, An artificial bee colony algorithm with feasibility enforcement and infeasibility toleration procedures for cardinality constrained portfolio optimization, Expert Syst. Appl. 85 (2017) 61–75, http://dx.doi.org/10.1016/j.eswa.2017.05.018.

[38] K. Deb, An efficient constraint handling method for genetic algorithms, Comput. Methods Appl. Mech. Engrg. 186 (2–4) (2000) 311–338, http://dx.doi.org/10.1016/S0045-7825(99)00389-8.

[39] R. Moral-Escudero, R. Ruiz-Torrubiano, A. Suárez, Selection of optimal investment portfolios with cardinality constraints, in: 2006 IEEE Congress on Evolutionary Computation, CEC 2006, 2006, pp. 2382–2388, http://dx.doi.org/10.1109/cec.2006.1688603.

[40] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 1917 (6), 2000, pp. 849–858, http://dx.doi.org/10.1007/3-540-45356-3_83.

[41] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2014) 577–601, http://dx.doi.org/10.1109/TEVC.2013.2281535.

[42] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, Evol. Methods Des. Optim. Control Appl. Ind. Probl. 103 (2001) 95–100, doi: 10.1.1.28.7571.

[43] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731, http://dx.doi.org/10.1109/TEVC.2007.892759.

[44] H. Xu, W. Zeng, D. Zhang, X. Zeng, MOEA/HD: A multiobjective evolutionary algorithm based on hierarchical decomposition, IEEE Trans. Cybern. 49 (2) (2017) 517–526, http://dx.doi.org/10.1109/tcyb.2017.2779450.

[45] H. Ma, M. Fei, Z. Jiang, L. Li, H. Zhou, D. Crookes, A multipopulation-based multiobjective evolutionary algorithm, IEEE Trans. Cybern. 50 (2) (2020) 689–702, http://dx.doi.org/10.1109/TCYB.2018.2871473.

[46] S. Liu, Q. Lin, K.C. Tan, M. Gong, C.A. Coello Coello, A fuzzy decomposition-based multi/many-objective evolutionary algorithm, IEEE Trans. Cybern. 52 (5) (2022) 3495–3509, http://dx.doi.org/10.1109/TCYB.2020.3008697.

[47] Q. Zhang, H. Mühlenbein, On the convergence of a class of estimation of distribution algorithms, IEEE Trans. Evol. Comput. 8 (2) (2004) 127–136, http://dx.doi.org/10.1109/TEVC.2003.820663.

[48] R. Rastegar, M.R. Meybodi, A study on the global convergence time complexity of estimation of distribution algorithms, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3641 LNAI, 2005, pp. 441–450, http://dx.doi.org/10.1007/11548669_46.

[49] T. Chen, K. Tang, G. Chen, X. Yao, On the analysis of average time complexity of estimation of distribution algorithms, in: 2007 IEEE Congress on Evolutionary Computation, CEC 2007, 2007, pp. 453–460, http://dx.doi.org/10.1109/CEC.2007.4424506.

[50] S. Gao, L. Qiu, C. Cao, Improved estimation of distribution algorithms based on gaussian distribution, Res. J. Appl. Sci. Eng. Technol. 6 (10) (2013) 1841–1845, http://dx.doi.org/10.19026/rjaset.6.3912.

[51] Y. Liang, Z. Ren, X. Yao, Z. Feng, A. Chen, W. Guo, Enhancing Gaussian estimation of distribution algorithm by exploiting evolution direction with archive, IEEE Trans. Cybern. 50 (1) (2020) 140–152, http://dx.doi.org/10.1109/TCYB.2018.2869567.

[52] H. Karshenas, R. Santana, C. Bielza, P. Larranaga, Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables, IEEE Trans. Evol. Comput. 18 (4) (2014) 519–542, http://dx.doi.org/10.1109/TEVC.2013.2281524.

[53] W. Shi, W.N. Chen, T. Gu, H. Jin, J. Zhang, Handling uncertainty in financial decision making: A clustering estimation of distribution algorithm with simplified simulation, IEEE Trans. Emerg. Top. Comput. Intell. 5 (1) (2021) 42–56, http://dx.doi.org/10.1109/TETCI.2020.3013652.

[54] K. Kim, R.I. McKay, Stochastic diversity loss and scalability in estimation of distribution genetic programming, IEEE Trans. Evol. Comput. 17 (3) (2013) 301–320, http://dx.doi.org/10.1109/TEVC.2012.2196521.

[55] S.Y. Wang, L. Wang, An estimation of distribution algorithm-based memetic algorithm for the distributed assembly permutation flow-shop scheduling problem, IEEE Trans. Syst. Man, Cybern. Syst. 46 (1) (2016) 139–149, http://dx.doi.org/10.1109/TSMC.2015.2416127.

[56] Q. Yang, W.N. Chen, Y. Li, C.L.P. Chen, X.M. Xu, J. Zhang, Multimodal estimation of distribution algorithms, IEEE Trans. Cybern. 47 (3) (2017) 636–650, http://dx.doi.org/10.1109/TCYB.2016.2523000.

[57] P. Yang, K. Tang, X. Lu, Improving estimation of distribution algorithm on multimodal problems by detecting promising areas, IEEE Trans. Cybern. 45 (8) (2015) 1438–1449, http://dx.doi.org/10.1109/TCYB.2014.2352411.

[58] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: Optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. B 26 (1) (1996) 29–41, http://dx.doi.org/10.1109/3477.484436.

[59] M. Dorigo, L.M. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem, IEEE Trans. Evol. Comput. 1 (1) (1997) 53–66, http://dx.doi.org/10.1109/4235.585892.

[60] M. Mavrovouniotis, F.M. Muller, S. Yang, Ant colony optimization with local search for dynamic traveling salesman problems, IEEE Trans. Cybern. 47 (7) (2017) 1743–1756, http://dx.doi.org/10.1109/TCYB.2016.2556742.

[61] Z.M. Huang, W.N. Chen, Q. Li, X.N. Luo, H.Q. Yuan, J. Zhang, Ant colony evacuation planner: An ant colony system with incremental flow assignment for multipath crowd evacuation, IEEE Trans. Cybern. 51 (11) (2021) 5559–5572, http://dx.doi.org/10.1109/TCYB.2020.3013271.

[62] X. Yu, W.N. Chen, T. Gu, H. Yuan, H. Zhang, J. Zhang, ACO-A∗: Ant colony optimization plus a∗ for 3-D traveling in environments with dense obstacles, IEEE Trans. Evol. Comput. 23 (4) (2019) 617–631, http://dx.doi.org/10.1109/TEVC.2018.2878221.

[63] X. Yu others, Path planning in multiple-AUV systems for difficult target traveling missions: A hybrid metaheuristic approach, IEEE Trans. Cogn. Dev. Syst. 12 (3) (2020) 561–574, http://dx.doi.org/10.1109/TCDS.2019.2944945.

[64] W.N. Chen, D.Z. Tan, Q. Yang, T. Gu, J. Zhang, Ant colony optimization for the control of pollutant spreading on social networks, IEEE Trans. Cybern. 50 (9) (2020) 4053–4065, http://dx.doi.org/10.1109/TCYB.2019.2922266.

[65] C. Fahy, S. Yang, M. Gongora, Ant colony stream clustering: A fast density clustering algorithm for dynamic data streams, IEEE Trans. Cybern. 49 (6) (2019) 2215–2228, http://dx.doi.org/10.1109/TCYB.2018.2822552.

[66] Z.G. Chen, et al., Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach, IEEE Trans. Cybern. 49 (8) (2019) 2912–2926, http://dx.doi.org/10.1109/TCYB.2018.2832640.

[67] Y.H. Jia, et al., An intelligent cloud workflow scheduling system with time estimation and adaptive ant colony optimization, IEEE Trans. Syst. Man, Cybern. Syst. 51 (1) (2021) 634–649, http://dx.doi.org/10.1109/TSMC.2018.2881018.

[68] M. Kong, P. Tian, A convergence proof for the ant colony optimization algorithms, in: Proceedings of the 2005 International Conference on Artificial Intelligence, ICAI'05, 2005, pp. 118–121, 1.

[69] A. Badr, A. Fahmy, A proof of convergence for ant algorithms, Inf. Sci. (Ny) 160 (1–4) (2004) 267–279, http://dx.doi.org/10.1016/j.ins.2003.08.018.

[70] H. Duan, D. Wang, X. Yu, Markov chains and martingale theory based convergence proof of ant colony algorithm and its simulation platform, in: Proceedings of the World Congress on Intelligent Control and Automation (WCICA), 2006, pp. 3057–3061, http://dx.doi.org/10.1109/WCICA.2006.1712928, 1.

[71] Z.M. Yang, H. Huang, Z. Cai, Y. Qin, A theoretical framework for runtime analysis of ant colony optimization, in: 2010 International Conference on Machine Learning and Cybernetics, ICMLC 2010, 2010, pp. 1817–1822, http://dx.doi.org/10.1109/ICMLC.2010.5580959, 4.

[72] K. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, C. Stummer, Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection, Ann. Oper. Res. 131 (1–4) (2004) 79–99, http://dx.doi.org/10.1023/B:ANOR.0000039513.99038.c6.