COMPLEX SYSTEM MODELING AND SIMULATION
ISSN 2096-9929 02/06 pp 191−201
Volume 3, Number 3, September 2023
DOI: 10.23919/CSMS.2023.0012

$I_{\epsilon+}$LGEA: A Learning-Guided Evolutionary Algorithm Based on $I_{\epsilon+}$ Indicator for Portfolio Optimization

Feng Wang*, Zilu Huang, and Shuwen Wang

Abstract: Portfolio optimization is a classical and important problem in the field of asset management, which aims to achieve a trade-off between profit and risk. Previous portfolio optimization models use traditional risk measurements such as variance, which symmetrically delineate both positive and negative sides and are not practical and stable. In this paper, a new model with cardinality constraints is first proposed, in which the idiosyncratic volatility factor is used to replace traditional risk measurements and can capture the risks of the portfolio in a more accurate way. The new model has practical constraints which involve the sparsity and irregularity of variables and make it challenging to be solved by traditional Multi-Objective Evolutionary Algorithms (MOEAs). To solve the model, a Learning-Guided Evolutionary Algorithm based on $I_{\epsilon+}$ indicator ($I_{\epsilon+}$LGEA) is developed. In $I_{\epsilon+}$LGEA, the $I_{\epsilon+}$ indicator is incorporated into the initialization and genetic operators to guarantee the sparsity of solutions and can help improve the convergence of the algorithm. And a new constraint-handling method based on $I_{\epsilon+}$ indicator is also adopted to ensure the feasibility of solutions. The experimental results on five portfolio trading datasets including up to 1226 assets show that $I_{\epsilon+}$LGEA outperforms some state-of-the-art MOEAs in most cases.

Key words: portfolio optimization; evolutionary algorithm; sparse solution space; indicator-based Evolutionary Algorithm (EA)

1 Introduction

The portfolio selection is a significant and established issue in financial practice. In the previous research, technical analysis is used to optimize portfolio decisions. It has been proved that the technical analysis performs well in developed countries[1]. However, the Chinese stock market is semi-efficient so the technical analysis only includes past information and has nonsustainable meanings in the real world[2]. Therefore, it is necessary to replace technical analysis with others.

Markowitz proposed a classical methodology, and it maximizes the mean return while minimizes the variance of the portfolio, which is also called mean-variance model. However, the mean-variance model is not flawless. Variance is an indicator that measures the total risks of a portfolio, symmetrically delineating both positive and negative sides. However, in the practical world, people are more afraid of downside fluctuations rather than the upside. Variance ignores this important feature and loses the accurate meaning of risk in the real world[3]. Then Salehpoor and Molla-Alizadeh-Zavardehi[4] started to use semi-variance, Mean Absolute Deviation (MAD), and skewness to manage the asymmetric nature of risk, which suit better for the continuous situations. Therefore, another measurement of risk called idiosyncratic volatility in

• Feng Wang and Zilu Huang are with the School of Computer Science, Wuhan University, Wuhan 430072, China. E-mail: fengwang@whu.edu.cn; huangzilu@whu.edu.cn.
• Shuwen Wang is with the Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. E-mail: shuwen.w828@gmail.com.
∗ To whom correspondence should be addressed.
※ This article was recommended by Associate Editor Wenyin Gong.
Manuscript received: 2023-04-17; revised: 2023-04-27; accepted: 2023-05-09

finance is considered.

Idiosyncratic volatility was proposed to be the most efficient indicator to capture the unsystematic risk and illustrate the uncertainty that only belongs to the company itself. A controversial fact is discovered that stocks with higher return performance also have lower idiosyncratic volatility. Based on the flaws of previous risk indicators and economic rationale behind idiosyncratic volatility, a model is developed that replaces the traditional risk measurements such as variance and semi-variance with idiosyncratic volatility to construct a more stable and practical portfolio in the volatile market. In this paper, a momentum-volatility model is developed with cardinality constraints that considers the whole volatility impact on the market. Due to the sparsity and irregularity of the variables in the novel model, it is difficult for typical Multi-Objective Evolutionary Algorithms (MOEAs) to solve it.

Portfolio optimization is a large-scale optimization problem. This kind of problem is currently being addressed by numerous intelligent optimization methods based on reinforcement learning[5]. To solve the complex optimization problem, Wang et al.[6] suggested a reinforcement learning level based particle swarm optimization technique. To break the issue into a few low-dimensional subproblems, Sun et al.[7] suggested using a random grouping technique. In addition, more and more intelligent optimization algorithms are being proposed[8−10].

Many MOEAs have been proposed to solve the Portfolio Optimization Problems (POPs)[3]. In order to solve the POPs with various risk metrics, Kaucic et al.[11] proposed a novel version of the NSGA-II and the SPEA2. This kind of method focuses on the risk measures of the POPs. Based on the new NBI-style Tchebycheff technique, Zhang et al.[12] suggested using MOEA/D to solve the POPs with inconsistently scaled objectives. This kind of method focuses on the scale of the POPs. A five-objective assistant reference point guided evolutionary algorithm was presented by Ma et al.[13] for the fuzzy portfolio selection. This kind of method focuses on the objective of the POPs. However, few scholars use indicators to solve the POPs. In this paper, an indicator is used to guide the evolution to guarantee the sparsity of the solutions and can help improve the algorithm's convergence. Since the $I_{\epsilon+}$ indicator can accurately assess the solution's convergence[14] and it is also parameterless with a low computational cost which is suitable to solve the POPs, in this paper, the $I_{\epsilon+}$ indicator is used to guide the evolution.

To summarize, the major contributions are concluded as follows.

- A novel portfolio optimization model is proposed to replace traditional risk measurements with idiosyncratic volatility factors. It can capture the risks of the portfolio in a more accurate way which makes the model more practical and stable.
- A Learning-Guided Evolutionary Algorithm based on $I_{\epsilon+}$ indicator ($I_{\epsilon+}$LGEA) is developed to solve the new portfolio optimization model. The $I_{\epsilon+}$ indicator which is parameterless with a low computational cost is incorporated into initialization and genetic operators to ensure the solution's sparsity. In addition, a new constraint-handling method based on the $I_{\epsilon+}$ indicator is adopted to guarantee the feasibility of the solutions.
- To show the efficiency of the proposed $I_{\epsilon+}$LGEA, extensive experimental tests are conducted on five portfolio datasets and show that $I_{\epsilon+}$LGEA outperforms in most cases.

The remainder is organized as follows. Section 2 introduces the new momentum-volatility model. Section 3 details the $I_{\epsilon+}$LGEA. The experimental studies are presented in Section 4. Section 5 concludes our work.

2 Momentum-Volatility Portfolio Optimization Model

This section introduces the related assumptions and notations, the classical portfolio model, and our new model.

2.1 Assumption and notation

Here are some basic assumptions.

- Transaction costs are zero.
- This portfolio that we create is not time-changing and dynamic.

The following notations are defined as follows:

- $i = 1, 2, \ldots, n$: index of assets,
- w_i: weighting of stock i,
- $\mathrm{MOM}_{i,t}$: cumulative return of stock i at time t,
- $r_{i,t}$: return of stock i at time t,
- MKT_t: excess market return at time t,
- SMB_t: size factor at time t,
- HML_t: value factor at time t,
- α_i: intercept of asset i in the regression,
- $\varepsilon_{i,t}$: idiosyncratic volatility for asset i at time t,

- p_i: price of stock i.

2.2 Classical model

The most classical portfolio optimization model was proposed by Markowitz. The traditional POPs can be expressed mathematically as follows:

$$\max \quad R = \sum_{i=1}^{N} w_i r_i \tag{1}$$

$$\min \quad V = \sum_{i=1}^{N} w_i^2 \mathrm{Var}_i \tag{2}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} w_i = 1 \tag{3}$$

where r_i is the return of stock i, Var_i is the variance of stork i, and w_i is the weight that stock i takes up in that portfolio. R is the sum return of this portfolio. V is the total variance of this portfolio that includes i stocks.

However, the risk measurement, variance, is very sensitive to one single movement in the solution space. Furthermore, it measures risk symmetrically, which is not true for the real-world situation.

2.3 New momentum-volatility model

The traditional measure of the risk loses accurate meaning of risk in the real world. The idiosyncratic volatility can capture the unsystematic risk and illustrate the uncertainty that only belongs to the company itself. In addition, the average return is replaced with the cumulative return (momentum) to reflect historical information in our model. Hence, a new momentum-volatility model is proposed which can help make the portfolio model more applicable. The absolute momentum is defined as follows:

$$\mathrm{MOM}_{i,t} = p_{i,t-l} \prod_{i=0}^{K} (1 + r_{i,t-l}) \tag{4}$$

where $p_{i,t}$ is the price of stock i at time t, l is the lagging time periods (months) of holding this stock, and $r_{i,t-l}$ is the return rate of stock i at time $t-l$ (lagging l months). In the objective functions, the averaged momentum factor is adopted for stock i over a specific period t. In this step, the stock is eliminated with negative absolute momentum.

The calculation of the idiosyncratic volatility metric is shown as follows. Since it is the unsystematic risk for each stock, it is required to filter out the common systematic risk. By extracting the variance of residuals from the Fama-French model[15], the idiosyncratic uncertainty can be captured. First, the Fama-French model is implemented for each stock. The regression on the Fama-French three factors equation of each asset is provided as follows:

$$\begin{cases} r_{1,t} = \alpha_1 + \beta_{1,\mathrm{MKT}}\mathrm{MKT}_t + \\ \quad \beta_{1,\mathrm{SMB}}\mathrm{SMB}_t + \beta_{1,\mathrm{HML}}\mathrm{HML}_t + \varepsilon_{1,t}, \\ r_{2,t} = \alpha_2 + \beta_{2,\mathrm{MKT}}\mathrm{MKT}_t + \\ \quad \beta_{2,\mathrm{SMB}}\mathrm{SMB}_t + \beta_{2,\mathrm{HML}}\mathrm{HML}_t + \varepsilon_{2,t}, \\ r_{3,t} = \alpha_3 + \beta_{3,\mathrm{MKT}}\mathrm{MKT}_t + \\ \quad \beta_{3,\mathrm{SMB}}\mathrm{SMB}_t + \beta_{3,\mathrm{HML}}\mathrm{HML}_t + \varepsilon_{3,t}, \\ \ldots \\ r_{N,t} = \alpha_N + \beta_{N,\mathrm{MKT}}\mathrm{MKT}_t + \\ \quad \beta_{N,\mathrm{SMB}}\mathrm{SMB}_t + \beta_{N,\mathrm{HML}}\mathrm{HML}_t + \varepsilon_{N,t} \end{cases} \tag{5}$$

By adding them up with certain weights, the sum residual of this whole portfolio can be obtained.

$$\begin{aligned} r_{p,t} = & \sum_{i=1}^{N} w_i \alpha_i + \sum_{i=1}^{N} w_i \beta_{i,\mathrm{MKT}}\mathrm{MKT}_t + \\ & \sum_{i=1}^{N} w_i \beta_{i,\mathrm{SMB}}\mathrm{SMB}_t + \\ & \sum_{i=1}^{N} w_i \beta_{i,\mathrm{HML}}\mathrm{HML}_t + \\ & \sum_{i=1}^{N} w_i \varepsilon_{i,t} \end{aligned} \tag{6}$$

where $\sum_{i=1}^{N} w_i \varepsilon_{i,t}$ stands for the total idiosyncratic volatility for the portfolio at time t.

The momentum-volatility model is built as follows:

$$\max \quad F_1(t, r(t)) = \sum_{i=1}^{N} w_i \mathrm{MOM}_{i,t} \tag{7}$$

$$\min \quad F_2(t) = \sum_{i=1}^{N} w_i^2 \mathrm{Var}^2(\varepsilon_{i,t}) \tag{8}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} w_i = 1 \tag{9}$$

$$\sum_{i=1}^{N} s_i = K \tag{10}$$

$$l_i \leqslant w_i \leqslant h_i \tag{11}$$

$$s_i \in \{0, 1\} \tag{12}$$

where $\mathrm{MOM}_{i,t}$ is the cumulative return of stock i, a description of the absolute momentum factor. $\mathrm{Var}(\varepsilon_{i,t})$ is the residual term of the regression on size factor SMB_t, value factor HML_t, and market return term MKT_t. The cardinality limitation is shown in Eq. (10), where K is the total number of assets held. The ratio of stock i is defined as w_i in Formula (11), and the floor and ceiling restrictions are l_i and h_i, respectively. If s_i is 1, it indicates that stock i has been selected. Otherwise, the stock i is not chosen.

As mentioned above, the momentum-volatility model includes inequality constraints such as Formula (11) and equality constraints such as Eq. (10) which involve the sparsity and irregularity of variables and make it challengeable to be solved by traditional MOEAs.

3 $I_{\epsilon+}$LGEA

The new population initialization and genetic operators incorporate the $I_{\epsilon+}$ indicator to guarantee the sparsity of the solutions and potentially enhance algorithm convergence. The constraint-handling methods based on the $I_{\epsilon+}$ indicator are adopted to ensure the feasibility of the solutions. Hence, $I_{\epsilon+}$LGEA can handle the momentum-volatility model effectively.

3.1 Framework of $I_{\epsilon+}$LGEA

The flowchart of the proposed $I_{\epsilon+}$LGEA is shown in Fig. 1. Firstly, a population is initialized and the Score of each asset is obtained based on $I_{\epsilon+}$ indicator which is shown in Algorithm 1. Next N offsprings are generated with new genetic operators which is shown in Algorithms 2 and 3. Then the constraint-handling method based on $I_{\epsilon+}$ indicator is adopted to make the offspring solutions feasible which is shown in Algorithm 4. After then, the parent population is joined with the offspring population. The solutions will survive to the following generation baed on environmental selection.

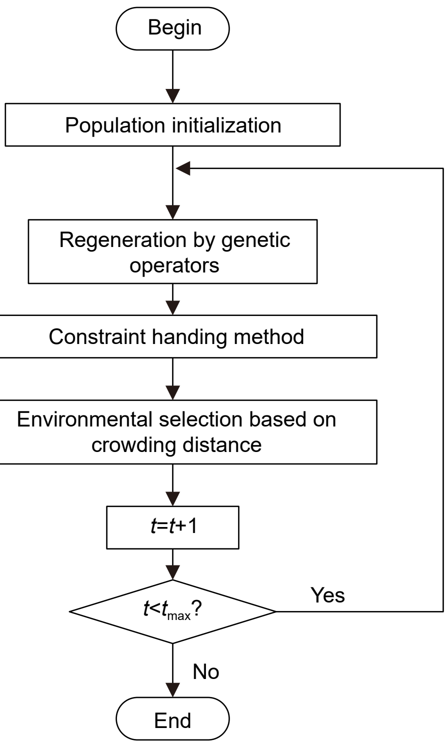

Fig. 1 Flowchart of $I_{\epsilon+}$LGEA.

3.2 Population initialization

A hybrid representation is used to solve portfolio optimization. The solution X is composed of the real vector W and the binary vector B. W denotes the weight of the associated asset. B determines whether the associated asset is chosen or not. The final decision variables of X are obtained via normalization, i.e.,

$$\begin{cases} X_i = 0, \quad \text{if} \quad B_i = 0; \\ X_i = l_i B_i + \dfrac{W_i B_i}{\sum\limits_{i=1}^{D} W_i B_i}\left(1 - \sum\limits_{i=1}^{D} l_i B_i\right), \quad \text{otherwise} \end{cases} \tag{13}$$

Algorithm 1 Initialization (N, K)

Input: N (population size), K (max cardinality), and $D\leftarrow$ number of assets;
Output: P (initial population) and Score (score of decision variables);
$B \leftarrow D\times D$ identity matrix;
$W \leftarrow D\times D$ identity matrix;
$Q \leftarrow$ A population whose i-th solution is generated by the i-th rows of W and B according to Eq. (13);
$B \leftarrow N\times D$ matrix of zeros;
$W \leftarrow$ Uniformly randomly generate the decision variables of N solutions;
for i = 1 to N **do**
  for j = 1 to rand()$\times K$ **do**
    $[m,n] \leftarrow$ Randomly select two decision variables;
    if $\mathrm{Score}_m > \mathrm{Score}_n$ **then**
      Set the m-th element in the i-th binary vector in B to 1;
    else
      Set the n-th element in the i-th binary vector in B to 1;
    end if
  end for
end for
$P\leftarrow$ A population whose i-th solution is generated by the i-th rows of W and B according to Eq. (13);
return P and Score

Algorithm 2 Crossover (P', Score)

Input: P' (a set of parents) and Score (score of decision variables);
Output: O (a set of offsprings by crossover);
$O \leftarrow \varnothing$;
while P' is not empty **do**
  $[p,q] \leftarrow$ Randomly select two parents from P';
  $P' \leftarrow P' \setminus \{p,q\}$;
  $o.B \leftarrow p.B$; //$p.B$ denotes the binary vector B of solution p
  if rand() < 0.5 **then**
    $[m,n] \leftarrow$ Randomly select two decision variables from the nonzero elements in $p.B \cap q.\bar{B}$;
    if $\text{Score}_m < \text{Score}_n$ **then**
      Set the m-th element in $o.B$ to 0;
    else
      Set the n-th element in $o.B$ to 0;
    end if
  else
    $[m,n] \leftarrow$ Randomly select two decision variables from the zero elements in $p.\bar{B} \cap q.B$;
    if $\text{Score}_m > \text{Score}_n$ **then**
      Set the m-th element in $o.B$ to 1;
    else
      Set the n-th element in $o.B$ to 1;
    end if
  end if
  $O \leftarrow O \cup \{o\}$
end while
return O

Algorithm 3 Mutation (O)

Input: O (a set of offsprings by crossover);
Output: O (a set of offsprings by mutaion);
if rand() < 0.5 **then**
  $[m,n] \leftarrow$ Randomly select two decision variables from the nonzero elements in $o.B$;
  if $\text{Score}_m < \text{Score}_n$ **then**
    Set the m-th element in $o.B$ to 0;
  else
    Set the n-th element in $o.B$ to 0;
  end if
else
  $[m,n] \leftarrow$ Randomly select two decision variables from the nonzero elements in $o.\bar{B}$;
  if $\text{Score}_m > \text{Score}_n$ **then**
    Set the m-th element in $o.B$ to 1;
  else
    Set the n-th element in $o.B$ to 1;
  end if
end if
$o.W \leftarrow$ Perform simulated binary crossover and polynomial mutation based on $p.W$ and $q.W$;
$O \leftarrow O \cup \{o\}$
return O

Algorithm 4 Constraint-handling (P, Score, K)

Input: P (combined population), Score (score of decision variables), and K (max cardinality);
Output: P (repaired population);
$N \leftarrow$ the size of P;
for i = 1 to N **do**
  if sum of $B_i > K$ **then**
    sort the Score of the selected asset;
    keep only the K largest score of selected asset to 1 and set all surplus asset to 0;
  end if
end for
$P \leftarrow$ A population whose i-th solution is generated by the i-th rows of W and B according to Eq. (13);
return P

where l_i is a small predefined lower limit of the portfolio optimization. In $I_{\epsilon+}$LGEA, W and B are initialized and evolved by various methods.

The population initialization of $I_{\epsilon+}$LGEA is outlined in Algorithm 1. Firstly, the population Q with D solutions is generated, where D indicates the number of assets. The variables in W and B are set to 0, and the i-th variable of the i-th solution is set to 1. Then, the $I_{\epsilon+}$ indicator of population Q is calculated. $I_{\epsilon+}$ indicator is regarded as the Score of the asset represented as S. The Score_i of the i-th decision variable indicates the probability of the i-th asset should be selected. A smaller Score_i of the i-th decision variable means a lower probability that the i-th asset should be selected. The $I_{\epsilon+}$ indicator and corresponding Score are defined as

$$I_{\epsilon+}(x,y) = \min_{\epsilon}(f_i(x) - \epsilon \leqslant f_i(y), i \in (1,2,\ldots,m)) \tag{14}$$

$$S(x) = \sum_{y \in Q, y \neq x} -\mathrm{e}^{I_{\epsilon+}(x,y)/0.05} \tag{15}$$

Every variable in B is set to 0, and every variable in W is set to a random value for each solution in P. Then, using a binary tournament selection method, rand()$\times K$ variables are chosen from B and set to 1 according to the Score of the decision variables, where rand() stands for a random number with uniform distribution in the range $[0,1]$. It can guarantee that the solutions that are generated are feasible.

An example of initializing the population is shown in Fig. 2. Assuming that the number of initial assets is 5, the initialization will generate five individuals P_1–P_5,

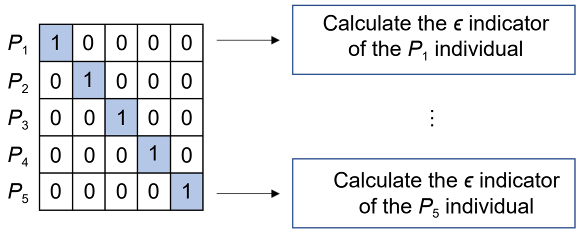

Fig. 2　Initial population example.

the binary vectors of these five individuals are the identity matrix of 5×5, and then the i-th individual will be calculated. The ϵ^+ indicator represents its score, which represents the probability of the i-th asset being selected. The higher the score, the more likely the asset will be selected. For example, the ϵ^+ indicator of P_1 individual represents the probability of the i-th asset being selected. Initialization can help ensure the sparsity of the initial population.

3.3　Genetic operators

The genetic operators include crossover and mutation. It is provided in Algorithms 2 and 3. In order to produce an offspring o, two parents p and q are initially chosen at random from P. When B of o and p are equal, one of the following two operations is carried out with the same probability: binary tournament selection is used to choose a variable from the nonzero variables in $p.B\cap q.\bar{B}$ based on the score of the decision variables (Lines 9−15 in Algorithm 2), and changing this component of an offspring's B to 0; or binary tournament selection is used to choose a variable from the nonzero variables in B based on the score of the decision variables (Lines 16−23 in Algorithm 2), and changing this component of an offspring's B to 1.

The crossover process is shown in Fig. 3. "Parent 1" and "Parent 2" represent two randomly selected parent individuals, and the black dotted box indicates that the selected assets of the two parent individuals are the same, which are directly inherited by the offspring individual "Offspring". And then randomly select two of the non-zero assets, that is, the third asset and the sixth asset in Fig. 3 (shown by the red arrow in Fig. 3), and the indicator of the smaller one is set to 0. Assuming the indicator of the sixth asset is smaller, the sixth asset of offspring individuals is 0, and the third asset is 1. Randomly select two assets in the zero elements, that is, the second asset and the fourth asset in Fig. 3 (shown by the black arrow in Fig. 3). Assuming the indicator of the fourth asset is larger, then the fourth asset of offspring individuals is 1, and the second asset is 0.

The mutation process is shown in Fig. 4. Randomly select two assets among the non-zero elements, that is, the third asset and the fourth asset in Fig. 4 (shown by the red arrow in Fig. 4), and set the asset whose indicator is smaller to 0. Randomly select two assets in the zero element, that is, the second decision variable and the sixth decision variable in Fig. 4 (shown by the black arrow in the Fig. 4), and set the asset whose indicator is larger to 1.

B of o is mutated by any of the next two operations with the same probability after crossover: using a binary tournament to select a variable from the nonzero variables in $o.B$ based on the score of the decision variables (Lines 3−9 in Algorithm 3), and setting this element to 0; or choosing an element through binary tournament selection with the score of the decision variables from the nonzero items in $o.\bar{B}$ (Lines 10−16

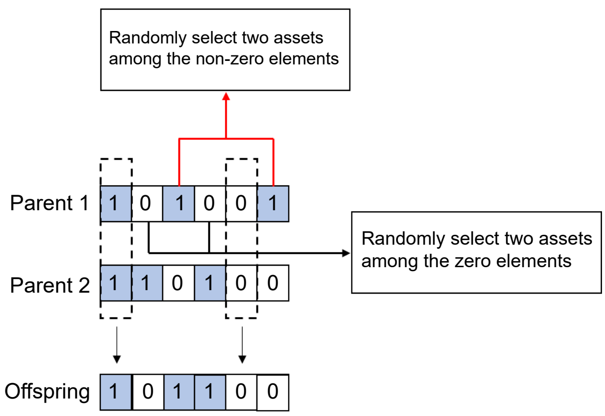

Fig. 3　Crossover example.

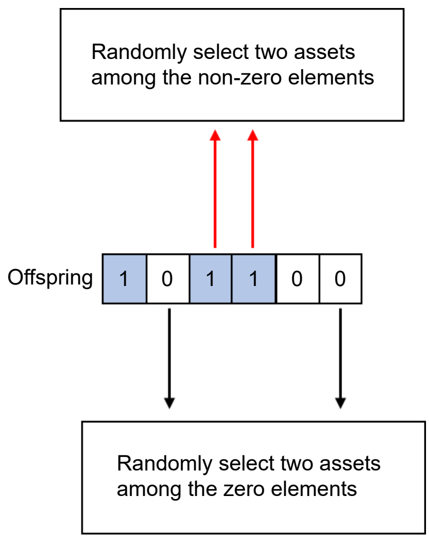

Fig. 4　Mutation example.

in Algorithm 3), and setting this element to 1. The same operators used in many MOEAs[16, 17] currently are used to construct W of o.

3.4 Constraint-handling method based on $I_{\epsilon+}$ indicator

The process of the constraint-handling method is introduced in Algorithm 4. To begin with, calculate the sum of the binary B_i of the i-th solution. If the result is larger than max cardinality K, then sort the Score of the selected asset in the i-th solution. Then keep only the K largest Score of the selected asset to 1 and label all surplus asset to 0. Finally, the final decision variables of X are obtained via normalization according to Eq. (13). $I_{\epsilon+}$ indicator can evaluate convergence and it is regarded as the Score of the asset. The Score$_i$ of the i-th asset indicates the probability that the i-th asset should be selected. Hence, the constraint-handling method selects the K largest Score of the asset to improve the convergence.

An example of constraint handling is shown in Fig. 5. Assuming that the cardinality constraint is 3 and the selected stocks are 4, which exceed the cardinality constraint, so a stock needs to be discarded. The selected stocks are sorted according to the score corresponding to the selected stocks, and the stocks with the smallest score are set to 0. Since the greater the score of stocks, the greater the potential of stocks. When the number of selected stocks exceeds the number of cardinality constraints, the stocks with the least potential should be unselected.

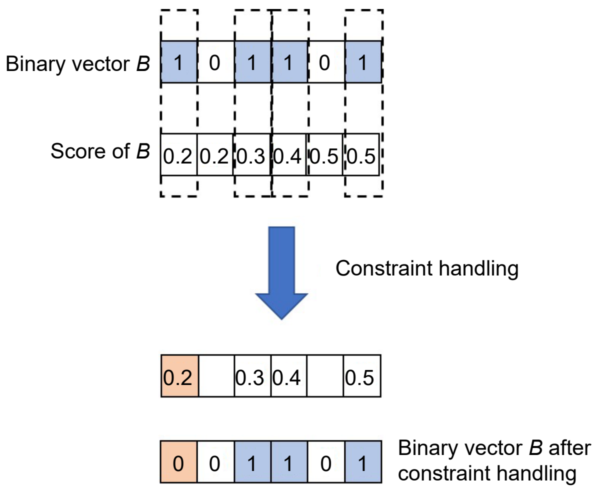

Fig. 5 Constraint handling example.

4 Experimental Study

We compared the proposed method to four MOEAs, namely NSGA-II, MOEA/D, IBEA, and HypE, on five well-known portfolio datasets. The evolutionary multi-objective optimization platform PlatEMO is used for all of the experiments[18].

4.1 Parameter settings and datasets

All compared algorithms have a population size of 100. On all portfolio datasets, the maximum number of function evaluations is set to 30 000, and 30 independent runs are carried out.

The datasets are available from Wharton Research Data Services (WRDS). In the smallest instance of the datasets, there are 22 assets, and in the greatest case, there are up to 1226 assets. The details of the datasets are introduced in Table 1.

4.2 Performance metrics

The performance metrics of Inverted Generational Distance (IGD) and Hypervolume (HV) are applied for evaluating the results. we define the reference set for each instance as the collection of non-dominated solutions acquired from all comparison algorithm runs.

4.3 Experimental results on portfolio datasets

This section tests the effectiveness of the algorithms based on portfolio data from five distinct financial markets. It is clear from Tables 2 and 3 that the final solutions generated by $I_{\epsilon+}$LGEA are superior to those of other algorithms for all test problems. Especially on NASDAQComp and SP500 datasets, $I_{\epsilon+}$LGEA is significantly better than other comparison algorithms. The main reason is that the number of stocks in the NASDAQComp and SP500 datasets exceeds 100, which leads to a larger scale of the problem and a

Table 1 Details of portfolio datasets.

Dataset	Number of assets	Cardinality K	Time interval
DowJones	22	5	Jan 2000−Aug 2020
Industries49	49	12	Jan 2000−Aug 2020
NASDAQ 100	82	20	Jan 2000−Aug 2020
NASDAQComp	1226	300	Jan 2000−Aug 2020
SP500	386	100	Jan 2000−Aug 2020

Table 2 HV metric's mean and standard deviation values obtained by the comparing algorithms.

Dataset	HV metric's mean (standard deviation value)				
	NSGA-II	MOEA/D	IBEA	HypE	$I_{\epsilon+}$LGEA
DowJones	4.0382×10^3 (2.83)−	3.6812×10^3 (5.29×10^2)−	4.0365×10^3 (4.93)−	3.3656×10^3 (9.37×10^1)−	**4.0439×10^3 (1.03)**
Industries49	1.5627×10^5 (3.27×10^2)−	1.5426×10^5 (2.21×10^3)−	1.5666×10^5 (8.04×10^2)−	1.0075×10^5 (5.03×10^3)−	**1.5729×10^5 (6.73×10^1)**
NASDAQ100	7.6840×10^4 (5.76×10^2)−	7.5004×10^4 (3.20×10^3)−	7.7324×10^4 (2.71×10^2)−	4.2257×10^4 (2.49×10^3)−	**7.7881×10^4 (3.39×10^1)**
NASDAQComp	2.5033×10^{12} (7.29×10^{11})−	1.0280×10^{12} (5.56×10^{11})−	2.7547×10^{12} (3.53×10^{11})−	1.2620×10^{11} (1.08×10^{10})−	**2.8274×10^{12} (8.86×10^8)**
SP500	2.7683×10^4 (1.16×10^3)−	2.6191×10^4 (2.43×10^3)−	2.8447×10^4 (2.67×10^2)−	1.6084×10^4 (7.52×10^2)−	**2.8953×10^4 (3.53×10^1)**
+/ − / ≈	0/5/0	0/5/0	0/5/0	0/5/0	−

Note: Symbols "+", "−", and "≈" indicate that the result by another MOEA is significantly better, significantly worse, and statistically similar to that obtained by $I_{\epsilon+}$LGEA, respectively.

Table 3 IGD metric's mean and standard deviation values obtained by the comparing algorithms.

Dataset	IGD metric's mean (standard deviation value)				
	NSGA-II	MOEA/D	IBEA	HypE	$I_{\epsilon+}$LGEA
DowJones	1.8384 (7.91×10^{-2})≈	2.0004×10^1 (3.83×10^1)−	2.0630 (6.33×10^{-2})−	1.5783×10^2 (1.25×10^1)−	**1.8195 (5.57×10^{-2})**
Industries49	2.6175 (1.20×10^{-1})−	2.1094×10^1 (3.45)−	2.9128 (6.49×10^{-1})−	4.0055×10^2 (2.96×10^1)−	**2.3799 (8.99×10^{-2})**
NASDAQ100	4.9032 (3.19×10^{-1})−	9.9617 (4.57)−	6.0687 (2.59×10^{-1})−	5.8094×10^2 (5.98)−	**4.4690 (2.48×10^{-1})**
NASDAQComp	1.3520×10^7 (2.92×10^7)−	6.9734×10^7 (2.20×10^7)−	3.0845×10^6 (1.41×10^7)−	7.8548×10^7 (1.43×10^3)−	**6.2253×10^5 (2.99×10^4)**
SP500	3.4997 (3.96)−	2.9591×10^1 (2.94×10^1)−	3.4989 (5.35×10^{-1})−	1.9364×10^2 (2.37)−	**1.4261 (1.01×10^{-1})**
+/ − / ≈	0/4/1	0/5/0	0/5/0	0/5/0	−

Note: Symbols "+", "−", and "≈" indicate that the result by another MOEA is significantly better, significantly worse, and statistically similar to that obtained by $I_{\epsilon+}$LGEA, respectively.

sparser solution space. Other algorithms cannot handle large-scale sparse optimization problems well. This also further verifies that special initialization methods and genetic operators in $I_{\epsilon+}$LGEA can effectively guarantee the performance of the algorithm and ensure the sparsity of the solutions.

In Fig. 6, the final solution sets for the 30 runs of five POPs are presented together with the median HV values. Figure 6 demonstrates that $I_{\epsilon+}$LGEA has little better convergence than other techniques but obtains the best diversity and uniformity on most POPs.

4.4 Effect of different components

It is noted that the initialization and genetic operators in $I_{\epsilon+}$LGEA are inspired by SparseEA[19]. To study whether the number of $I_{\epsilon+}$LGEA components has an influence on the improvement of performance, different experiments are designed by combining different components of $I_{\epsilon+}$LGEA. For a simpler description, we name the two major components $I_{\epsilon+}$ indicator computation and constraint-handling method based on $I_{\epsilon+}$ indicator as C_1 and C_2, respectively. C_1 means that we employ the $I_{\epsilon+}$ indicator of the i-th solution in Q as the score of the i-th asset. C_1+C_2 means that we also employ the new constraint-handling method, so the complete $I_{\epsilon+}$LGEA is C_1+C_2.

To testify the effectiveness of the proposed $I_{\epsilon+}$LGEA more comprehensively, we compare SparseEA, C_1, and C_1+C_2 for fairness. HV-metric values and IGD-metric values of the final solutions for five portfolio selection problems are shown in Tables 4 and 5, respectively.

On the most of the test problems, it is clear from Table 4 that the final solutions provided by C_1 are superior to SparseEA in terms of HV-metric. Table 5 reveals that, for all test problems except the NASDAQ100 problem, the final solutions generated by C_1 are superior to SparseEA in terms of IGD-metric. It can be explained that the non-dominated front number of the i-th solution in Q is the score of the i-th asset in SparseEA. The non-dominated front number of the i-th

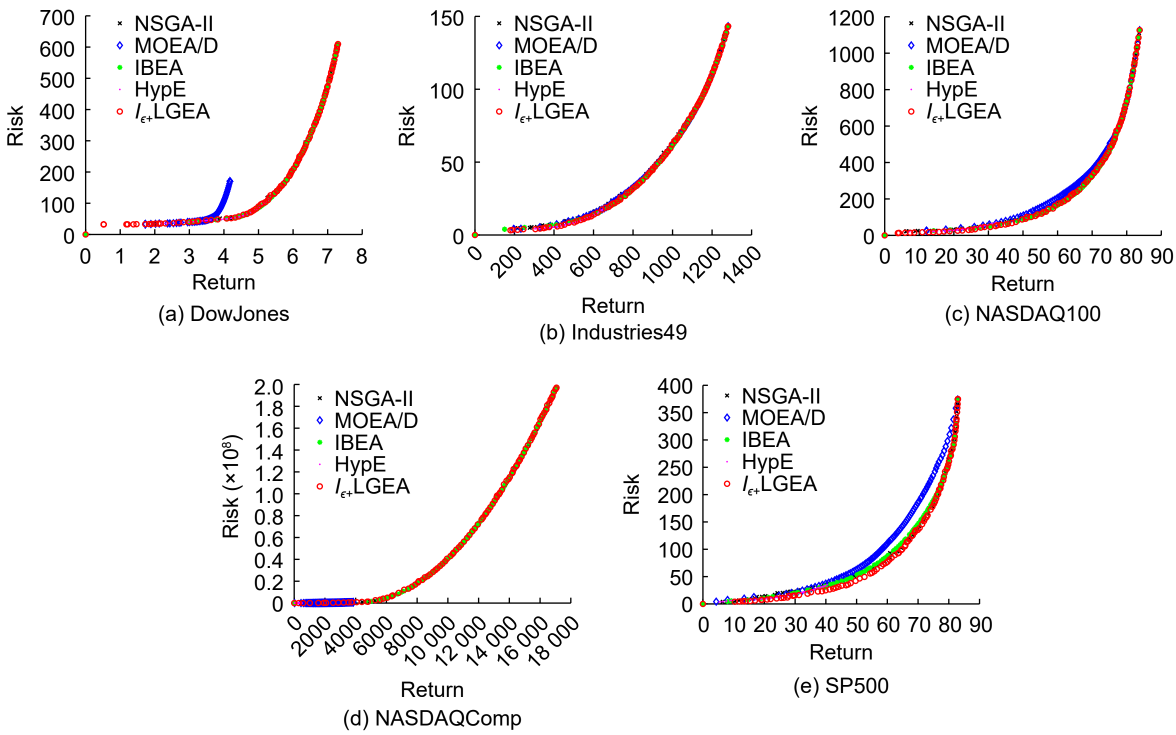

Fig. 6 Plots for five datasets.

Table 4 HV metric's mean and standard deviation values determined by different components of $I_{\epsilon+}$LGEA.

Dataset	HV metric's mean (standard deviation value)		
	SparseEA	C_1	C_1+C_2
DowJones	$4.0427\times10^3(1.06)-$	$4.0427\times10^3(1.23)-$	$\mathbf{4.0444\times10^3(7.79\times10^{-1})}$
Industries49	$1.5719\times10^5(7.68\times10^1)-$	$1.5721\times10^5(8.02\times10^1)-$	$\mathbf{1.5726\times10^5(6.53\times10^1)}$
NASDAQ100	$7.7822\times10^4(4.55\times10^1)-$	$7.7844\times10^4(3.00\times10^1)-$	$\mathbf{7.7874\times10^4(2.57\times10^1)}$
NASDAQComp	$\mathbf{2.8275\times10^{12}(1.19\times10^9)}\approx$	$2.8275\times10^{12}(1.37\times10^9)\approx$	$2.8271\times10^{12}(1.24\times10^9)$
SP500	$2.8898\times10^4(4.23\times10^1)-$	$2.8948\times10^4(2.99\times10^1)\approx$	$\mathbf{2.8962\times10^4(2.94\times10^1)}$
$+/-/\approx$	0/4/1	0/3/2	–

Note: Symbols "+", "–", and "≈" indicate that the result by another MOEA is significantly better, significantly worse, and statistically similar to that obtained by $I_{\epsilon+}$LGEA, respectively.

Table 5 IGD metric's mean and standard deviation values determined by different components of $I_{\epsilon+}$LGEA.

Dataset	IGD metric's mean (standard deviation value)		
	SparseEA	C_1	C_1+C_2
DowJones	$1.9057(1.13\times10^{-1})-$	$1.8810(7.61\times10^{-2})\approx$	$\mathbf{1.8396(7.84\times10^{-2})}$
Industries49	$3.4302(1.72\times10^{-1})-$	$3.3666(1.30\times10^{-1})-$	$\mathbf{3.2846(1.54\times10^{-1})}$
NASDAQ100	$\mathbf{3.9519(1.87\times10^{-1})}\approx$	$3.9643(1.76\times10^{-1})\approx$	$4.0081(2.21\times10^{-1})$
NASDAQComp	$5.7228\times10^5(2.42\times10^4)\approx$	$5.7196\times10^5(3.19\times10^4)\approx$	$\mathbf{5.7124\times10^5(2.20\times10^4)}$
SP500	$1.9736(1.01\times10^{-1})-$	$1.9170(1.01\times10^{-1})\approx$	$\mathbf{1.8811(9.60\times10^{-2})}$
$+/-/\approx$	0/3/2	0/1/4	–

Note: Symbols "+", "–", and "≈" indicate that the result by another MOEA is significantly better, significantly worse, and statistically similar to that obtained by $I_{\epsilon+}$LGEA, respectively.

asset may be the same as the j-th asset. Hence, it can not distinguish the potentiality of different assets comprehensively. However, $I_{\epsilon+}$LGEA employs the $I_{\epsilon+}$ indicator of the i-th solution as the score of the i-th asset. The $I_{\epsilon+}$ indicator can distinguish the potentiality of different assets.

Additionally, to demonstrate the effectiveness of the suggested constraint-handling approach, $I_{\epsilon+}$LGEA is compared with constraint-handling method in Ref. [20] (C_1) and $I_{\epsilon+}$LGEA (C_1+C_2). It is clear that for every test problem, C_1+C_2's final solutions are superior to C_1's. It can be explained that the constraint-handling method based on weight keeps only the K largest weight of the selected asset to 1 and set all surplus assets to 0. The weight can not evaluate the potentiality of the asset. However, a constraint-handling method based on the $I_{\epsilon+}$ indicator can evaluate the convergence of the solutions to select the promising assets.

For the discussion on the above results, in general, the proposed $I_{\epsilon+}$LGEA can help ensure the sparsity of the solutions and improve the convergence of the algorithm. Besides, the new constraint-handling method performs better than traditional constraint-handling methods.

5 Conclusion

In this paper, a novel portfolio optimization model is developed, which can help capture the risks of the portfolio more accurately. To deal with the novel portfolio optimization, a learning-guided evolutionary algorithm based on the $I_{\epsilon+}$ indicator is proposed to obtain better convergence and distributed solutions. The new population initialization and genetic operators use $I_{\epsilon+}$ indicator which is parameterless and low time-consuming to guide the evolutionary process. It can ensure the sparsity of the generated solutions and help improve the convergence of the algorithm. The new constraint-handing method based on $I_{\epsilon+}$ indicator can ensure the feasibility of the generated solutions. The performance of the proposed algorithm is superior to other MOEAs according to experimental results on a variety of portfolio optimization problems. The $I_{\epsilon+}$LGEA algorithm proposed in this paper focuses on solving the constrained portfolio optimization problem with two objectives. However, there are still some portfolio models with more complex constraints and more objectives, so in future work, we will further study these more complex portfolio models to solve the problem.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 62173258 and 61773296).

References

[1] T. Hendershott, X. Zhang, J. L. Zhao, and Z. Zheng, FinTech as a game changer: Overview of research frontiers, *Inf. Syst. Res.*, vol. 32, no. 1, pp. 1–17, 2021.

[2] B. S. Alhashel, F. W. Almudhaf, and J. A. Hansz, Can technical analysis generate superior returns in securitized property markets? Evidence from East Asia markets, *Pac. Basin Finance J.*, vol. 47, pp. 92–108, 2018.

[3] O. Ertenlice and C. B. Kalayci, A survey of swarm intelligence for portfolio optimization: Algorithms and applications, *Swarm Evol. Comput.*, vol. 39, pp. 36–52, 2018.

[4] I. B. Salehpoor and S. Molla-Alizadeh-Zavardehi, A constrained portfolio selection model at considering risk-adjusted measure by using hybrid meta-heuristic algorithms, *Appl. Soft Comput.*, vol. 75, pp. 233–253, 2019.

[5] L. Wang, Z. Pan, and J. Wang, A review of reinforcement learning based intelligent optimization for manufacturing scheduling, *Complex System Modeling and Simulation*, vol. 1, no. 4, pp. 257–270, 2021.

[6] F. Wang, X. Wang, and S. Sun, A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization, *Inf. Sci.*, vol. 602, pp. 298–312, 2022.

[7] M. Sun, C. Sun, X. Li, G. Zhang, and F. Akhtar, Large-scale expensive optimization with a switching strategy, *Complex System Modeling and Simulation*, vol. 2, no. 3, pp. 253–263, 2022.

[8] W. Gong, Z. Liao, X. Mi, L. Wang, and Y. Guo, Nonlinear equations solving with intelligent optimization algorithms: A survey, *Complex System Modeling and Simulation*, vol. 1, no. 1, pp. 15–32, 2021.

[9] X. Yan, H. Zuo, C. Hu, W. Gong, and V. S. Sheng, Load optimization scheduling of chip mounter based on hybrid adaptive optimization algorithm, *Complex System Modeling and Simulation*, vol. 3, no. 1, pp. 1–11, 2023.

[10] C. Hu, R. Qiao, Z. Zhang, X. Yan, and M. Li, Dynamic scheduling algorithm based on evolutionary reinforcement learning for sudden contaminant events under uncertain environment, *Complex System Modeling and Simulation*, vol. 2, no. 3, pp. 213–223, 2022.

[11] M. Kaucic, M. Moradi, and M. Mirzazadeh, Portfolio optimization by improved NSGA-II and SPEA 2 based on different risk measures, *Financial Innov.*, vol. 5, no. 1, p. 26, 2019.

[12] Q. Zhang, H. Li, D. Maringer, and E. Tsang, MOEA/D with NBI-style Tchebycheff approach for portfolio management, in *Proc. IEEE Congress on Evolutionary Computation*, Barcelona, Spain, 2010, pp. 1–8.

[13] X. Ma, J. Chen, Y. Sun, and Z. Zhu, Assistant reference point guided evolutionary algorithm for many-objective fuzzy portfolio selection, *Swarm Evol. Comput.*, vol. 62, p. 100862, 2021.

[14] E. Zitzler and S. Künzli, Indicator-based selection in multiobjective search, in *Proc. 8th International Conference on Parallel Problem Solving from Nature*, Birmingham, UK, 2004, pp. 832–842.

[15] E. F. Fama and K. R. French, Common risk factors in the returns on stocks and bonds, in *The Fama Portfolio:*

Selected Papers of Eugene F. Fama, J. H. Cochrane and T. J. Moskowitz, eds. Chicago, IL, USA: University of Chicago Press, 2017, pp. 392–449.

[16] K. Deb and R. B. Agrawal, Simulated binary crossover for continuous search space, *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.

[17] K. Deb and M. Goyal, A combined genetic adaptive search (GeneAS) for engineering design, *Journal of Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.

[18] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum], *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, 2017.

[19] Y. Tian, X. Zhang, C. Wang, and Y. Jin, An evolutionary algorithm for large-scale sparse multiobjective optimization problems, *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 380–393, 2019.

[20] F. Streichert, H. Ulmer, and A. Zell, Evaluating a hybrid encoding and three crossover operators on the constrained portfolio selection problem, in *Proc. 2004 Congress on Evolutionary Computation* (*IEEE Cat. No. 04TH8753*), Portland, OR, USA, 2004, pp. 932–939.

Feng Wang received the BSc and PhD degrees in computer science from Wuhan University, Wuhan, China in 2003 and 2008, respectively. She is currently a professor at the School of Computer Science, Wuhan University. Her research interests include evolutionary computation, intelligent information retrieval, and machine learning. She serves as a reviewer for several international journals and conferences.

Zilu Huang received the BSc degree in computer seience and technology from Hefei University of Technology, Hefei, China in 2020, and the MSc degree in computer seience and technology from Wuhan University, Wuhan, China in 2023. His research interest focuses on evolutionary computation.

Shuwen Wang received the bachelor degree in finance from Wuhan University, China in 2019, and the master degree in finance from MIT in 2021. Her research interest is computational finance.