

# A Population Based Incremental Learning Algorithm for the Multiobjective Portfolio

Shicheng Hu<sup>1</sup>, Fang Li<sup>1</sup>, Xiaona Deng<sup>1</sup> and Yang Liu<sup>2\*</sup>

<sup>1</sup>School of Economics & Management, Harbin Institute of Technology, Harbin 150001, China

<sup>2</sup>School of Computer Science & Technology, Harbin Institute of Technology, Harbin 150001, China

\*Corresponding author

**Abstract**—Multiobjective optimization problem Portfolio is a hard–decision–making problem in investment management. With respect to how to obtain the multiobjective candidate decisive solutions for Portfolio, a multiobjective optimization method M–PBIL (Multiobjective Population Based Incremental Learning) to Portfolio is proposed. Different from the traditional evolutionary algorithms which generate individuals based on the recombination of the current ones, M–PBIL follows the strategy of PBIL to generate individuals based on probability model and researches three key technologies in solving multiobjective optimization problems. First, for the multiobjective optimization problem in continuous space, a real number–based coding scheme is proposed, which can overcome the defects of binary coding such as code redundancy and probability conflict. In the second place, a variable probability model for the gene bit is designed so as to realize the dynamic partition for the intervals of decision variables. Next, a dominance and representativeness–based assessment mechanic is employed for the selection of non–dominated solutions of multiobjective optimization problem. The performances of the M–PBIL are evaluated by convergence and distribution and compared with the representative NSGAI on benchmark data. The experimental results show that M–PBIL outperforms NSGAI in convergence and distribution.

**Keywords**—portfolio; multiobjective optimization algorithm; evolutionary algorithm

## I. INTRODUCTION

Portfolio—a very important task in management, which involves how to manage amount of limited capital to purchase different kinds of financial products to maximize returns and minimize risk [1]. Portfolio is a typical classic puzzle in multiobjective optimization [2, 3]. Evolutionary algorithms are commonly used to solve multiobjective optimization problems as they can generate multiple solutions at a time. The evolutionary algorithm can be defined to two types from the generation mechanism of the solution: one is evolutionary algorithm based on existing solution recombination and another is evolutionary algorithm based on existing solution learning. The former usually generates a new set of solutions through recombination operations such as crossover and mutation of the current solution, among which NSGAI (Nondominated Sorting Genetic Algorithm II) [4, 5] is typical; the latter mainly learns according to the distribution of existing solutions to generate a new set of solutions, PBIL (Population Based Incremental Learning) [6] is typical in this case. NSGAI

introduces fast and non–inferior sorting and congestion calculation in traditional genetic algorithms, and adopts the elite retention selection mechanism. It has achieved significant advantages in both solving multiobjective optimization problems and solving the Portfolio problem [7]. While the main problem is the lack of diversity of the solution and the early convergence to the local optimal solution [8], which makes it difficult to find the preferred solution of the decision maker's satisfaction in the IMCDM decision–making process. The PBIL algorithm obtains a probability model based on the learning of current excellent individuals, and generates a new group through probability model control. Since the genetic operator is not used, avoiding the deception problem and the chain problem of the traditional genetic algorithm [9]. Thus, preventing the algorithm from falling into premature convergence or local optimization. Although PBIL started late, it is mainly used to solve single–objective optimization problems now in biological information, engineering optimization, aviation military, production scheduling, and combinatorial optimization [10].

According to the significant effect of PBIL in solving single–objective optimization problems, this paper takes further research on PBIL and propose a solution method named M–PBIL for multiobjective optimization problem. Section 2 shows the detail of the M–PBIL method, including three key issues and algorithmic processes involved in solving Portfolio; Section 3 uses standard datasets for M–PBIL Simulation experiments were performed and compared with the computational performance of NSGAI. Finally, the conclusion is presented.

## II. M–PBIL—THE SOLUTION OF PORTFOLIO

For the single–objective optimization problem, PBIL updates the probability model according to the optimal individual for each generation. The single–objective optimization problem usually only has one optimal individual, and the selection of the optimal individual and the update scheme of the probability model are relatively simple. As the optimum of multiobjective optimization problem is a non–inferior solution set. The optimal individual selection of PBIL for single–objective optimization problems and the update scheme of probability model cannot be applied to multiobjective optimization problems. Therefore, for multiobjective optimization problems, M–PBIL has to solve

three key problems: the coding of the solution, the probability model and its update, the evaluation and selection of the solution.

#### A. The Coding of the Solution

PBIL usually takes binary coding, which leads to coding redundancy and probability conflict, which reduces the efficiency of the algorithm [11]. In order to overcome these problems, M-PBIL uses real-value coding for the multiobjective optimization problem of Portfolio as follows:

Individual coding:

$$b_1, b_2, b_3, b_4, \dots, b_i, \dots, b_N. \quad (1)$$

Each of the gene coding codes  $b_i$  is a random value taken from  $[0, 1]$ . Solution to the Portfolio problem  $w_i = b_i / (\sum b_i)$ ,  $i = 1, 2, \dots, N$ .

To obtain the value of  $b_i$  in a specific interval and speed up the convergence of the algorithm, M-PBIL gradually subdivides the value range of  $b_i$  according to the evolution effect in the algorithm process. Set a probability threshold  $TH$ , assuming that  $b_i$  takes a value in a certain interval, and if the value of  $b_i$  exceeds  $TH$ , then the interval will be divided into  $D$  equal-sized sub-intervals. Obviously, at the beginning of the algorithm, the range of  $b_i$  is  $[0, 1]$ , and  $b_i$  takes the value of probability 1 in the interval. During the execution of the algorithm, the interval is gradually subdivided, and  $b_i$  takes values with different probabilities with different subintervals.

#### B. The Probability and Update of the Model

The encoding of each gene position takes a value from  $\{0, 1\}$  with a probability value by binary coding that is updated based on the optimal solution. M-PBIL uses real-value coding. The interval of each gene-encoding is gradually subdivided. Each sub-interval has a corresponding probability value. Therefore, each gene-bit coding corresponds to a variable length probability vector. Set the value range of  $b_i$  to  $T_i = [T_1, T_2] \cup (T_2, T_3] \cup (T_3, T_4] \cup \dots \cup (T_k, T_{k+1}] \cup \dots \cup (T_{d_i}, T_{d_i} + 1]$ , where  $T_1=0, T_{d_i+1}=1, k = 1, 2, \dots, d_i$ .  $T_i$  is dynamically changing, and  $T_i$  corresponds to a probability vector  $P_i$ .

$$P_i = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{ik}, \dots, p_{id_i}), \quad \sum_k p_{ik} = 1 \quad (2)$$

The component  $p_{ik}$  of  $P_i$  represents the probability that  $b_i$  takes a subinterval  $(T_k, T_{k+1}]$  of  $T_i$ . If the subinterval  $(T_k, T_{k+1}]$  is subdivided into  $D$  equal parts, the corresponding  $p_{ik}$  is also equally divided into  $D$  values. The probability vector  $P_i$  is used to generate the value  $b_i$  of the  $i$ -th ( $i = 1, 2, \dots, N$ ) gene positions of the individual: a cumulative probability vector  $CP_i = (CP_{i1}, CP_{i2}, CP_{i3}, \dots, CP_{ik}, \dots, CP_{id_i})$ , where  $CP_{ik} = \sum_{t \leq k} p_{it}$ .  $Z$  is a randomly generated number over the interval  $[0, 1]$ , then  $k = \min \{k \mid CP_{ik} \geq Z\}$ , ie the value of  $b_i$  from  $T_i$ 's  $k$ -th subinterval  $(T_k, T_{k+1}]$  is randomly generated.

For the multiobjective optimization problem, the optimal value in the evolution process is a non-inferior solution set, and the value probability vector  $P_i$  of each gene position is updated based on the coded value on the corresponding gene position of the non-inferior solution set. Each component of  $P_i$  is updated as follows:

$$p_{ik} = (1 - LR) \times p_{ik} + LR \times \frac{num_k}{pop^*}, k = 1, 2, \dots, d_i \quad (3)$$

Where  $LR$  is learning probability,  $pop^*$  is the number of individuals in the current non-inferior solution set,  $num_k$  represents the number of individuals in the sub-interval  $(T_k, T_{k+1}]$  on the  $i$ -th gene position in  $pop^*$  individuals.

The new population of M-PBIL is generated based on a probability model, which is updated based on the current optimal solution set. Therefore, according to the probabilistic model update, the algorithm gradually evolves toward the Pareto optimal solution set, embodying the intensification of the evolutionary algorithm. To avoid the algorithm falling into local optimum, M-PBIL also designs the variation of the probability model. Each component of  $P_i$  is mutated as follows:

$$p_{ik} = (1 - MUTE\_SHIFT) \times p_{ik} + MUTE\_SHIFT \times \text{random}(\{0, 1\}) \quad (4)$$

$MUTE\_SHIFT$  is the variation range, which is a real number between  $(0, 1]$ . The frequency of variation is controlled by the mutation probability  $MUTE\_PROB$ . M-PBIL realizes the diversification of evolutionary algorithms through the variation of probability model.

#### C. Evaluation and Selection of Solutions

The multiobjective optimization problem has a Pareto optimal solution set. In a group, dominance evaluates the pros and cons of the solution from the perspective of the value of the objective function. Due to the large number of Pareto optimal solutions, evolutionary algorithms can only find a finite solution set, so the representativeness of the solution is another important indicator to evaluate the merits. M-PBIL evaluates the pros and cons of the evolutionary process with both dominant and representative indicators.

For each individual  $I$  in the current population, there are two indicators related to dominance:  $sup_I$  and  $inf_I$ . The former refers to the number of individuals who dictate  $I$ , and the latter refers to the number of individuals that are governed by  $I$ . The smaller the  $sup_I$  value is, the larger the  $inf_I$  value is, the stronger the individual's dominance will be. The representativeness adopts the congestion degree of NSGAI[5] as an evaluation index, and the congestion degree of the individual  $I$  is recorded as  $crd_I$ . Obviously, the larger the  $crd_I$  value, the stronger the representativeness of the individual.

Prior to selection, individuals in the current population are prioritized by dominant and representative indicator values. The sorting process works as follows: in the first step, the  $crd_I$

and  $inf_i$  values of each individual are calculated for all unsorted individuals, and the individuals whose value of  $crd_i$  is 0 are classified as the first type S1; the second step is calculated the  $crd_i$  value of each individuals within the range of S1. The  $crd_i$  value of each individuals; the third step, sorting the individuals in S1: first sorting according to the size of the  $inf_i$  value, then sorting the individuals with the same  $inf_i$  value according to the size of the  $crd_i$  value, and finally performing random sorting for the individuals of the same order. After the ordering of the individuals in S1 is completed, the individuals in S1 are removed, and the remaining individuals are sequentially sorted according to the above three steps. Thus, the current population is divided into several ordered classes S1, S2, ..., and the individuals in each class are ordered, resulting in an ordering of all individuals in the population. The selection of individuals in the current group is based on the prioritization above. Obviously, individuals in S1 compose the non-inferior solution set of the current group.

#### D. Steps of Algorithm

The above shows the three key techniques of M-PBIL to solve the multiobjective optimization problem Portfolio. The main solution steps of the algorithm are given below.

- Initialize the parameter:

M-PBIL includes the constant quantization parameter below.

*GEN*: number of generations of circulation of the algorithm;

*pop*: The size of the newly generated group in each generation;

*D*: The number of equal parts of the interval;

*TH*: Probability threshold;

*LR*: Learning rate;

*MUTE\_PROB*: Mutation probability;

*MUTE\_SHIFT*: Variability;

- Construct initial probability vector  $P_i$ :

$b_i$  ( $i = 1, 2, \dots, N$ ), which initial value interval is  $T_i = [0, 1]$ . Therefore, its corresponding probability vector has only one component which is  $P_i = (1)$ .

- Building an initial group  $POP_0$ :

Generating the individual's gene position based on the probability vector  $P_i$   $b_i$  ( $i = 1, 2, \dots, N$ ), generating pop individuals. For the initial group  $POP_0$ , As  $P_i = 1$ ,  $T_i = [0, 1]$ ,  $b_i$  randomly generated from  $[0, 1]$ .

- While  $g \leq GEN$  do:
- Generating new groups  $POP_g$ :

Generating the individual's gene position based on the probability vector  $P_i$   $b_i$  ( $i = 1, 2, \dots, N$ ), generating pop individuals.

- Merged group:

Current group  $POP_g$  combines with last generation  $POP_{g-1}$  then becomes  $POP_g \cup POP_{g-1}$ .

- Selection:

Select individuals based on dominance and representative ordering from the combined group  $POP_g \cup POP_{g-1}$ . Then, the *pop* individuals ranked first are selected, and the newly selected *pop* individuals replace the *pop* individuals in the  $POP_g$ .

- Updating probability vectors  $P_i$ :

According to (3), updating component  $P_{ik}$  of probability variable  $P_i$  ( $i = 1, 2, \dots, N$ ).

- Mutation:

For each component  $P_{ik}$  of probability vector  $P_i$  ( $i = 1, 2, \dots, N$ ), if the randomly generated random  $(0, 1) \leq MUTE\_PROB$ , then mutate according to (4).

- Interval subdivision:

For each component  $P_{ik}$  of probability vector  $P_i$  ( $i = 1, 2, \dots, N$ ), if  $P_{ik} > TH$ , then corresponding interval  $(T_k, T_{k+1}]$  is divided into  $D$  equal parts. At the same time, the  $P_{ik}$  is also equally divided into  $D$  values, each value corresponding to the probability value of each subinterval.

- End while.

### III. EMPIRICAL COMPUTATION

In this paper, the MPBIL algorithm is programmed in C++ language and implemented on a personal computer with a 1.86 GHz CPU speed. The validity of the M-PBIL method is tested using the Hong Kong Hang Seng Index data in the published Portfolio standard data set OR-Library [12]. The Portfolio example data includes 31 stocks. The values of the constant parameters used in the algorithm are set as:  $GEN=500$ ;  $pop=2000$ ;  $D=10$ ;  $TH=0.8$ ;  $LR=0.1$ ;  $MUTE\_PROB=0.05$ ;  $MUTE\_SHIFT=0.02$ . The evaluation of the quality of the solution is carried out in two ways: the convergence and the distribution of the solution, combined with qualitative and quantitative methods [13]: the qualitative evaluation of convergence and distribution is the scatter of the solution set by the observation algorithm; Quantitative evaluation of convergence uses generation distance as the evaluation index, the smaller the generation distance, the better the convergence; the quantitative index of spatial evaluation for distribution, the smaller the spatial evaluation value, the better the distribution. At the same time, the calculation results of M-PBIL are compared with the calculation results of NSGAI. NSGAI is a representative evolutionary algorithm based on existing solution recombination, and it is also an algorithm with superior solution effect and significant influence in the industry [7, 14].

Figure 1 shows the real solution set (PF) of the Portfolio problem instance, the solution set obtained by M-PBIL, and the solution set obtained by NSGAI. The scatter points of the three solution sets are marked as black asterisks and blue small dots and red circles respectively. It can be seen from the figure

that the scatter of the solution set obtained by M-PBIL basically coincides with the scatter of the real solution set, except that there is a small part of the data missing at the two extremes, indicating that M-PBIL can converge to the true solution set; and NSGAI Although the scatter point of the obtained solution set is roughly consistent with the scatter point of the real solution set, as the gain and risk increase, it slightly deviates from the scatter point of the real solution set, and there is also a relatively obvious data loss. It can also be seen from the figure that the scatter distribution of M-PBIL is relatively uniform, and only a small number of defects appear at the front end and the end; NSGAI scatter points are dense in some places, some places are sparse, and a large number of points are concentrated at the front end. The mid-end, especially the end distribution, is less, reflecting the greater disadvantage of the NSGAI algorithm in terms of distribution.

TABLE I. COMPARISON OF QUANTITATIVE INDEX VALUES BETWEEN THE TWO METHODS

Methods	Generation distance	Special evaluation
M-PBIL	0.000326	0.000145
NSGAI	0.000867	0.000352

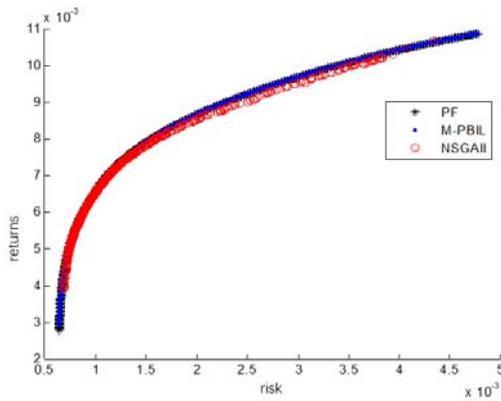


FIGURE I. COMPARISON OF SCATTER PLOTS OF THREE SOLUTION SETS

Table 1 is the generation distance and spatial evaluation index values calculated according to the solution set of the M-PBIL and NSGAI algorithms. It can be seen from the table that the generation distance and spatial evaluation index values of the M-PBIL algorithm are significantly smaller than the generation distance and spatial evaluation index values of NSGAI. The two quantitative index values also fully demonstrate that the convergence and distribution of M-PBIL are significantly better than NSGAI. But we found that the efficiency of M-PBIL is lower than that of NSGAI while the population size becomes larger.

#### IV. CONCLUSION

This paper proposes an optimization method M-PBIL for how to solve the multiobjective candidate decision-making scheme of Portfolio. M-PBIL follows PBIL's strategy for generating individuals based on probability model. To solve the

optimization problem of continuous space, M-PBIL adopts the real number coding scheme, which overcomes the coding redundancy and probability conflict of traditional PBIL binary coding, which is beneficial to improve the efficiency of the algorithm; M-PBIL design changes for each gene position. The long probability vector realizes the gradual subdivision of the decision interval of the decision variable and strengthens the convergence of the algorithm. M-PBIL adopts the evaluation mechanism based on dominance and representation for the solution in the evolution process to ensure that the algorithm can converge to the real solution set has good distribution.

#### ACKNOWLEDGMENT

This work was supported in part by NSFC under GrantNo. 61702139 and NSFS under Grant No. ZR2016FM27.

#### REFERENCES

- [1] T. L. Khin, "Evolutionary Approaches for Portfolio Optimization," PHD Dissertation: University of Nottingham, 2015.
- [2] C. L. He, "Mean-variance Model With Optimal Portfolio Selection Under General Uncertainty," Chinese Management Science, 2015, vol.23, no.12, pp. 63-70.
- [3] Z. Su, L. Qin, T. Fang, "Sparse Minimum Variance Portfolio Model with Graph Structure Constraints," Chinese Management Science, 2015, vol.23, no.9, pp. 65-70.
- [4] D.Meignan, S. Knust, J. M. Frayret, G. Pesant, N. Gaud, "A review and taxonomy of interactive optimization methods in operations research," ACM Transactions on Interactive Intelligent Systems, 2015, vol.5, no.3, pp. 1-43.
- [5] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, 2002, vol.6, no.2, pp. 182-197.
- [6] S. Baluja, "Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning," Pittsburgh, Pennsylvania: Carnegie Mellon University, 1994.
- [7] S. K. Mishra, G. Panda, R. Majhi, "A comparative performance assessment of a set of multiobjective algorithms for constrained Portfolio assets selection," Swarm and Evolutionary Computation, 2014, vol.16, pp. 38-51.
- [8] D. Mukhlisullina, A. Passerini, R. Battiti, "Learning to diversify in complex interactive multiobjective optimization," Proceedings of the 10th Metaheuristics International Conference (MIC 2013), Singapore Management University, Singapore: 2013: pp. 230-239.
- [9] D. E. Goldberg, B. Korb, K. Deb, "Messy genetic algorithms: motivation, analysis, and first results," Complex Systems, 2010, vol. 3, no.3, pp. 493-530.
- [10] Y. Wu, "Research on Distribution Estimation Algorithm and Its Application in Dynamic Optimization Problems," Doctoral Thesis, Xidian University, 2009.
- [11] S. D. Zhou, Z. Q. Sun, "A Survey of Distribution Estimation Algorithms," Acta Automatica Sinica, 2007, vol. 33, no. 2, pp. 113-124.
- [12] J. E. Beasley, "OR-Library: distributing test problems by electronic mail," Journal of the Operational Research Society, 1990, pp. 1069-1072.
- [13] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," IEEE Transactions on Evolutionary Computation, 2003, vol.7, no.2, pp. 117-132.
- [14] R. F. Huang, X. W. Luo, B. Ji, P. Wang, A. Yu, et al, "Multiobjective optimization of a mixed-flow pump impeller using modified NSGA-II algorithm," Chinese Science: Technical Science, 2015, vol.58, no.12, pp. 2122-2130.