Comparison between genetic algorithm and differential evolution algorithm applied to one dimensional bin-packing problem

Shi-Yuan Han, Xiao-Yu Wan, Lin Wang*, Jin Zhou
School of Information Science and Engineering,
University of Jinan,
Jinan, 250022, Shandong, China
Email: {ise_hansy, ise_wanglin, ise_zhouj}@ujn.edu.cn
* Corresponding author: ise_wanglin@ujn.edu.cn

Xiao-Fang Zhong
School of Information Technology,
Shandong Womens University,
Jinan 250002, Shandong, China
Email: 215046321@qq.com

***Abstract*—This paper discusses the one dimensional bin-packing problem (BPP) based on the genetic algorithm and differential evolution. First, the mathematical model for one bin-dimensional packing problem is established. Then, the detailed processes for one bin-dimensional packing problem are designed based on the genetic algorithm and differential evolution algorithms, and the difference performance between the designed algorithms are discussed. Finally, some simulation results are shown to prove the effectiveness of designed algorithms and verified the discussed conclusion.**

I. INTRODUCTION

In our modern worldlogistics distribution problem has been widely occurred in petroleum, online shopping goods and so on. Generally, logistics distribution problem consists of several challenging combinatorial optimization problemssuch as inventory management problemgoods packing problemvehicle routing problemetc. One dimensional bin-packing problem is one of the key factors in logistics distributions problem. As the classical combinatorial optimization problem, a growing number of scholars have paid more attention to this research area. For example, an island parallel grouping genetic algorithms were designed to solve the one-dimensional bin packing optimization problem in [1]; [2] proposed a consistent neighborhood search method for solving one-dimensional bin packing and two-dimensional vector packing; an average-weight-controlled bin-oriented heuristics algorithm was proposed for the one-dimensional bin-packing problem in [3]; [4] discussed the one-dimensional cutting stock problem with multi-objective optimization.

Generally speaking, the packing problem can be divided into three types according to the number of dimensions of the goods. But, even for one of the most easily one-dimensional problems, it is a NP-hard problem. The issues of packing problem with computer in combination optimization has closely related to order scheduling problems and in-depth research on this issue can advance our understanding of NP hard problem, expecting to find a good way to deal with the related combinatorial optimization problems. With the developing of intelligent computing technology, and the intelligent theory applied in the algorithm design area, it arises great challenge for the classical combinatorial optimization problems in this new computing environment. Especially, the genetic algorithm and differential evolution, as the classic intelligent algorithm, have much potential to solve the packing packing problem. Thus, the research of this problem has great theoretical significance. [5] designed a practical evolutionary algorithm for one dimensional stock cutting problem; some heuristic algorithms are proposed to solve the one-dimensional bin packing problem in [6], [7]; Massively parallel computing algorithm was applied to the one-dimensional bin packing problem in [8]; [9] proposed a parallelizing generalized algorithm to solve the one-dimensional bin packing problem by using map-reduce technology.

The motivation of this paper is to apply the genetic algorithm and differential evolution to solve the one-dimensional bin packing problem. A mathematical model is established to describe the one-dimensional bin packing problem as the combination optimization problem. The algorithm processes for the genetic algorithm and differential evolution are designed to obtained the packing solution. Simulation results prove the effectiveness of designed algorithms.

II. PROBLEM FORMULATION

The one-dimensional bin packing problem can be descried that packing n items into a number of same size and shape bins with a capacity of c with respect to minimize the number of bins required to contain n items [5], [8]. Assume that the the number of bins is unlimited with capacity $c > 0$. A set of n items is to be packed into bins and the size of item $i \in 1, ..., n$ is $s_i > 0$, as shown in Table 1.

Then the mathematical model of one-dimensional bin packing problem can be described as:

$$\begin{aligned} \min \ & f(x) = \sum_{i=1}^{n} x_i, \\ s.t. \ & \sum_{j=1}^{n} s_j y_{ij} < l_i, \\ & \sum_{i=1}^{n} y_{ij} = 1, \\ & x_i = 0 \quad or \quad 1, \quad i \in N, \\ & y_{ij} = 0 \quad or \quad 1, \quad i, j \in N, \end{aligned} \tag{1}$$

While $x_i = 1$, it denotes that the ith bin is with the items, otherwise it is empty. While $y_{ij} = 1$, it denotes that the jth item packs into the ith pin, otherwise it does not pack.

TABLE I. WEIGHT OF TIMES AND INFORMATION OF BINS

Item's Number	1	2	...	n
Weight	s_1	s_2	...	s_n
Upper Limits	l_1	l_2	...	l_n

The goal of this paper is to find the minimum number M of bins for packing n items into bins with respect to minimum $f(x)$ in Eq. (1) based on the genetic algorithm and differential evolution algorithm.

III. DESIGN OF ALGORITHM FOR ONE-DIMENSIONAL BPP

In this section, we will describe the genetic algorithm and differential evolution for one-dimensional BPP.

A. Design of genetic algorithm for one-dimensional BPP

Referring to the biological theory of evolution, the genetic algorithm transforms the one-dimension BPP into a biological evolution process. Through the process of reproduction, crossover and mutation operations, the next generation of solutions will be obtained, in which the solution with smallest value of fitness function will be gradually obtained. Finally, the optimal solution for one-dimensional BPP will be generated.

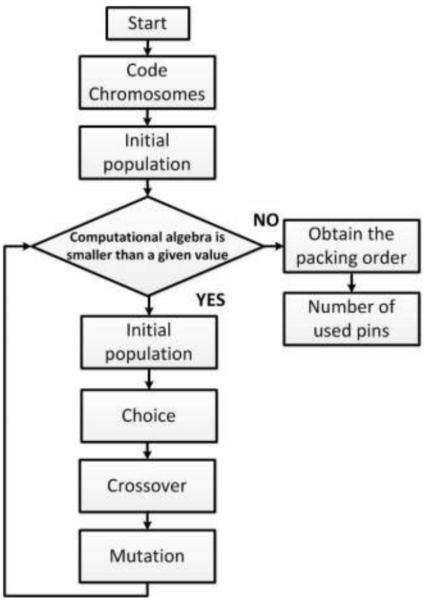

Fig. 1. The Process of Genetic Algorithm for One-Dimensional BPP

The process of genetic algorithm for one-dimensional BPP is described as follows, which is shown in Fig. 1.

Step 1 **Chromosome representation and initialization**
Each item needs to be loaded into the pins. In order to record the detained parking process, each item could be given a specifical ID number $1, 2, 3, ..., N$. Therefore, we can use a fixed length for chromosome encoding in this paper. Then the chromosome of each individual can just be obtained from a random sequence among all of the number of items in the population. Chromosome representation and initialization can be done in this step.

Step 2 **Fitness function**
The packing problem is an optimization problem essentially. The basic target is to park all items into pins with respect to the minimum number of pins. Based on this target, a fitness function is designed as follows:

$$f = \frac{\sum_{i=1}^{n} Fi/c^K}{n}, \tag{2}$$

where n denotes the number of required pins, F_i is the total weight after packing pins, f denote the sum of weight of packed items, c is the weight limit of items, $K(K > 1)$ is the constant, which represents the emphasis on packing pins.

Step 3 **Selection**
Selection is based on the evaluation of the population fitness in all the individual's adaptation of the operation. We use the roulette method to select cross individual. As the dart game of roulette, roulette selection probability of the individuals is proportional in the transformation. It should be pointed that the individuals with higher fitness are selected with higher probability, otherwise, the opposite case is with lower probability.

Step 4 **Cross**
Cross is the primary way to generate new individuals in genetic algorithm, which exchanges genes on the chromosomes of the parent to produce new genes. But crossing will also generate some of inappropriate conditions of the chromosome. According to the chromosome encoding, chromosome represents the order of the items. In this paper, we chose the selection method proposed in [9] to produce new genes.

Step 5 **Mutation**
The mutation can make the chromosome produce new characters, and it has a great influence on the genetic algorithm. Here is the method of crossover and mutation, nondirectional produce chromosome in the two position number and the region between two numbers reverses the order.

B. Design of differential evolution for one-dimensional BPP

The basic idea of difference evolution algorithm is that: first, the parent individual mutation operation constitutes the individual variation; based on the ensured probability, the parent individuals and individual of variation generate individual of test generation by the crossover operation; finally, between the parent individuals and trial individual,the greedy selection operation is decided based on the adaptation degree to retain superior and achieve the population evolution.

The process of designed differential evolution algorithm for one-dimensional BPP is described as follows, which is shown in Fig. 2.

Step 1 **Initialization parameters**
The defined ID for each item is $1, 2, 3, ...N$. Consider a fixed length chromosome to encode and put these items in the sequence as a chromosome. The initialization parameters include minimum cycle times

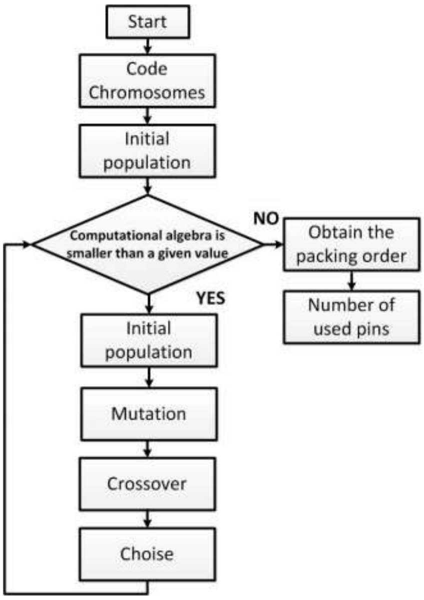

Fig. 2. The Process of Differential Evolution for One-Dimensional BPP

min, initial population NP, scaling factor F, and cross constant.

Step 2 **Population initialization**
Initializing population of $X(0)$ that each individual chromosome is as $1, 2, 3, ...N$ sequence and calculates the fitness of each individual. Three different individuals are randomly selected except the target individuals.
While population NP, dimension D, evolutionary population $X(i)$ are given, a feasible solution to the problem of random generation can be described as

$$\begin{cases} X(0) = \{\ x_1^0, \ \cdots, \ x_{NP}^0\ \}, \\ x_i^0 = \{\ x_{i,1}^0, \ \cdots, \ x_{i,D}^0\ \}, \\ x_{i,j}^0 = x_{j,\min} + rand(x_{j,\min} - x_{j,\max}), \end{cases} \quad (3)$$

where $x_{j,\min}$ and $x_{j,\max}$ denote the upper and lower bounds of the solution space of dimension.

Step 3 **Mutation** In order to realize the mutation operation on X, the variation of individual are calculated.
Generation of variation vector t_i are given by:

$$t_i = x_{r1} + k(x_{r2} - x_{r3}), \quad i = 1, 2, \cdots, NP, \quad (4)$$

where x_{r1}, x_{r2}, x_{r3} is a random selection of three individuals in the parent population, k is the scaling factor.

Step 4 **Cross**
X and T were crossed to produce individual in Step 5. Through the component of variation vector T_i and random reorganization of the target vector X_i in each dimension. The crossover operation aims to improve population diversity algorithm by using the following equation and generate new cross vector $v_i = [v_{i,1}, v_{i,2}, ..., v_{i,D}]$:

$$v_{i,j} = \begin{cases} t_{i,j}, & randb \leq CR \ or \ j = rand_j, \\ x_{i,j}, & randb > CR \ or \ j \neq rand_j, \end{cases}$$
$$i = 1, \cdots, NP,$$
$$j = 1, \cdots, NP, \quad (5)$$

where CR is a constant in the range of $[0, 1]$, $Randb$ is a random number between $[0, 1]$. The greater value of CR, the more cross in the process. $rand_j \ in[1, D]$ ensures that at least one of the V component is obtained from the t component.

Step 5 **Choice**
The choice standard is based on fitness value of individual.
When the individual fitness of the new vector value is better than the individual fitness of target vector, the V population would be accepted, otherwise, x will remain in the next generation, which is given by:

$$x_i^{t+1} = \begin{cases} v_i, & f(v_i) < f(x_i^t) \\ x_i, & others \end{cases} \quad (6)$$

IV. Simulation

In order to demonstrate the validity and effectiveness of the designed algorithm for one-dimensional BPP, some simulation results will shown in this section, in which the packing results and the number of used pins will be given. Also, we will discuss the simulation result to demonstrate the analysis between two types algorithm.

The information of item's weights and bin's parameters are shown in Table 1. Applied to the designed genetic algorithm and differential evolution algorithm to on-dimensional BPP, the mainly simulation results are shown in Table III and IV. Through the comparison of the simulation results, the genetic algorithm and differential evolution algorithm can effectively solve the problem of one dimensional packing problem, but there are some differences, such as the packing order of pins, the number of items in each case.

Actually, the diversities among simulation results are mainly caused by:

TABLE II. Weights of items and parameters of bins

Number of items	$1-9$	$10-15$
Weights	0.3	0.2

TABLE III. The results based on genetic algorithm

Pins	Load Weight	Total Weight
1	$0.3, 0.2, 0.3, 0.2$	1.0
2	$0.2, 0.3, 0.2, 0.3$	1.0
3	$0.3, 0.2, 0.3$	0.9
4	$0.3, 0.2, 0.2, 0.3$	1.0

TABLE IV. The results based on differential evolution algorithm

Pins	Load Weight	Total Weight
1	$0.3, 0.3, 0.2$	0.9
2	$0.2, 0.3, 0.3, 0.2$	1.0
3	$0.3, 0.2, 0.2, 0.3$	1.0
4	$0.3, 0.2, 0.3, 0.2$	1.0

1. The traditional genetic algorithm uses the binary code, but the differential evolution algorithm uses the real number code.
2. In the genetic algorithm, produce two individual sub individuals through the two generations of the individual, but in the differential evolution algorithm, the difference vectors of second or more individuals are perturbed to generate new individuals.
3. In the traditional genetic algorithm, the individuals with a certain probability to replace its parent individual, but in differential evolution, newly produced individuals only replace the individual in the population when it is superior to the individual in the population.

V. CONCLUSION

The problem of one dimensional packing problem based on genetic algorithm and differential evolution algorithm was researched in this paper. Based on the established mathematical model, the algorithms was designed for the one-dimensional packing problem based on genetic algorithm and differential evolution algorithm, in which the detailed process was described. The simulation results shown that the designed algorithms can solve the one-dimensional packing problem effectively based on the genetic algorithm and differential evolution algorithm . However, there are still some differences comparison results between two algorithms, such as the packing order of the box, the number of items in each case and so on.

ACKNOWLEDGMENT

This work was supported by the Supported by Program for Scientific Research Innovation Team in Colleges and Universities of Shandong Province, National Natural Science Foundation of China under Grant No. 61573166, the Shandong Provincial Natural Science Foundation, China, under Grant ZR2015JL025, Shandong Province Higher Educational Science and Technology Program under Grant J16LN07, and the Shandong Distinguished Middle-Aged and Young Scientist Encourage and Reward Foundation Grant BS2014DX015.

REFERENCES

[1] D. Tansel, C. Ahmet, *Optimization of one-dimensional Bin Packing Problem with island parallel grouping genetic algorithms*, Computers and Industrial Engineering, 75: 176-186, 2014.

[2] B. Mirsad, V. Michel, *Consistent neighborhood search for one-dimensional bin packing and two-dimensional vector packing*, Computers and Industrial Engineering, 76: 12-21, 2016.

[3] F. Krzysztof, C. Christoforos, *Average-weight-controlled bin-oriented heuristics for the one-dimensional bin-packing problem*, European Journal of Operational Research, 210(2): 176-187, 2011.

[4] F. Ge, L. Liu, J. Pei, X. Zhang, *Study of one-dimensional cutting stock problem with multi-objective optimization*, 2012 International Conference on Computer Science and Information Processing (CSIP), 571-574, 2012.

[5] J. Zhang, Y. Li, Y. Yang, J. Li, *A practical evolutionary algorithm of one dimensional stock cutting problem*, Proceedings of the 3rd World Congress on Intelligent Control and Automation, 5: 3729 - 3732, 2000.

[6] C.D.T. Suarez, E.P. Gonzlez, M.V. Rendon, *A heuristic algorithm for the offline one-dimensional bin packing problem inspired by the point Jacobi Matrix iterative method* , 2006 Fifth Mexican International Conference on Artificial Intelligence, 281 - 286, 2006.

[7] S. Abidi, S. Krichen, E. Alba, J.M. Molina, *Improvement heuristic for solving the one-dimensional bin-packing problem*, 2013 5th International Conference on Modeling, Simulation and Applied Optimization(ICMSAO), 1-5, 2013.

[8] J. O. Berkey, *Massively parallel computing applied to the one-dimensional bin packing problem*, the 2nd Symposium on the Frontiers of Frontiers of Massively Parallel Computation, 317 - 319, 1988.

[9] Anika, D. Garg, *Parallelizing generalized one-dimensional bin packing problem using MapReduce*, 2014 IEEE International on Advance Computing Conference (IACC), 628 - 635, 2014.