A variable relationship excavating based optimization algorithm for solving 0-1 knapsack problems

Minyi Zheng[†], Fangqing Gu[*†], Xuesong Chen[†] and Hao-Tian Wu[‡]
[†] Guangdong University of Technology, Guangzhou, China.
[‡] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.
[*]Email: fqgu@gdut.edu.cn.

***Abstract*—The past decades have seen an extensive investigation of evolutionary algorithms. The recombination operators of most evolutionary algorithms are either single-point crossover or multi-point crossover for solving 0-1 combinatorial optimization problems. There is a little studies on excavating the variable relationship to improve the efficiency of the recombination operator. Hence, we propose an optimization algorithm based on excavating variable relationship for solving 0-1 combinational problems. The aim of the recombination operator is fully utilizing the inherent information from every slice component of all individuals. We compared the proposed algorithm with the classic evolutional algorithm with single-point crossover operation on several 0-1 knapsack problems, which is a classical combinatorial optimization problem. The simulation results show the convergence efficiency of the proposed algorithm.**

***Index Terms*—Optimization algorithm, Knapsack problem, Knowledge excavating, Combinatorial optimization.**

I. Introduction

During the past few decades, optimization problems have attracted a great deal of attentions due to its wide applications in real world. Number of evolutionary algorithms (EAs) have been successfully applied to solve optimization problems. Combinatorial optimization is a discrete optimization problem. It widely exists in reality commonly in the project field and studies of science. Typical combinatorial optimization problems [1] include minimum set cover [2], 0/1 knapsack problem [3], resource allocation [4], clustering problem [5] graph coloring problem [6], etc. These problems have very precise mathematical descriptions and high computational complexity. Conventional methods for solving combinatorial optimization problems such as branch and bound method, can obtain feasible solutions for the problems with small scale. While the increase of the scale of the problem may lead to combinatorial explosion of the searching space. Evolutionary algorithms have stimulated the interest of the researchers [7]–[9] for solving the combinational problems.

Researchers have presented some evolutionary algorithm for solving 0-1 knapsack problem, which is a classic combinatorial optimization problem [10]. For example, Yuan et al. [11] used the decomposition-based multi-objective evolutional algorithm with an improved greedy repair strategy to solve multiobjective backpacks. He et al. [12] proposed a new greedy strategy, and integrate the greedy strategy into a genetic algorithm to form a new hybrid genetic algorithm: Greedy Genetic Algorithm. Combining particle swarm optimization algorithm and greedy idea, a very greedy Particle Swarm Optimization algorithm is proposed for knapsack problem in [13]. Ye et al. [14] proposed a hybrid particle swarm algorithm to solve the 0/1 knapsack problem, which improves the search ability and the probability of finding the optimal solution. These algorithms improve the performance by using a problem specific greedy local search. However, there are little studies on excavating the variable relationship to improve the efficiency of the recombination operator. Hence, we propose an optimization algorithm based on excavating variable relationship for solving 0-1 combinational problems. The aim of recombination operator is fully utilizing the inherent information from every slice component of all individuals. We compared the proposed algorithm with the classic evolutional algorithm with single-point crossover operation on several 0-1 knapsack problems.

The remainder of this paper is organized as follows. In section II, we give a detailed description of the probability matrix and the main framework of proposed algorithm. Section III compares the proposed algorithm with classical evolutionary algorithm used one-point crossover. The simulation results are also presented in this section. The conclusion of this paper is drawn in section IV.

II. The Framework of the Proposed Algorithm

A. Variable relationship

Knapsack Problem (KP) is regarded as a resource allocation problem. In general, KP is a combinatorial optimization problem which requires to select a subset of items from available set to maximize the collective profit without exceeding its capacities. It's mathematically described as:

$$\begin{aligned} \max f(\mathbf{x}) &= \sum_{i=1}^{n} p_i x_i, \\ s.t. \sum_{i=1}^{n} r_i x_i &\le c, \\ x_i &\in \{0,1\}, i = 1, \cdots, n. \end{aligned} \tag{1}$$

Where $\mathbf{x} = (x_1, \cdots, x_n)$ with x_i for $i = 1, \cdots, n$ is decision variables and n is the dimension of the decision variables,

978-1-7281-6092-4/19/$31.00 ©2019 IEEE
DOI 10.1109/CIS.2019.00016

i.e., the number of items. $x_i = 1$ if and only if ith item is selected. The knapsack has a capacity c, and each item has a profit p_i and size r_i. A well-stated KP assumes that $p_i > 0$ and $r_i \leq c < \sum_{i=1}^{n} r_i$ for all $i = 1, \cdots, n$. KP belongs to the class of NP-hard problems.

Most of the existing optimization algorithms barely consider the relationship between variables when generating new individuals. However, in reality, there is a certain relationship between variables for many combinational optimization problems. Taking KP as an example, it is observed that a correlation between variables. There are four situations for any two items i and j:

1) both items i and j should be excluded, i.e., $x_i = 0$ and $x_j = 0$.
2) item i should be selected while item j should be excluded, i.e., $x_i = 1$ and $x_j = 0$.
3) item i should be excluded while item j should be selected, i.e., $x_i = 0$ and $x_j = 1$.
4) both items should be selected, i.e., $x_i = 1$ and $x_j = 1$.

It is a prospective conjecture that mining these relationships by searching history, i.e. historical population, helps to improve the searching efficiency.

B. Expression of variable relationship

In this paper, we use variable relationship matrix to represent the relationship between variables. Although there are four relationships among variables, two variable relationship matrices are enough. The reason is that case 1) and case 2) are statistically equivalent, as well as cases 3) and 4). Therefore, we use two conditional probability matrices $P = [p_{ij}]_{n\times n}$ and $Q = [q_{ij}]_{n\times n}$ to represent case 1) and case 3), respectively. $p_{ij} = p(x_j = 0|x_i = 0)$ denotes the conditional probability of $x_j = 0$ under the condition $x_i = 0$ and $q_{ij} = p(x_j = 0|x_i = 1)$ denotes the conditional probability of $x_j = 0$ under the condition $x_i = 1$.

Generally, the conditional probability between the decision variables in advance. Fortunately, we can approximate the conditional probability matrices by the population, i.e., a solution set, for the population-based optimization algorithms. For example, for a given solution set

$$X = \begin{array}{c} \begin{array}{cccccc} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 & \mathbf{x}_6 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array} \begin{array}{l} \\ item_1 \\ item_2 \\ item_3 \\ item_4 \\ item_5 \end{array}, \tag{2}$$

we use the frequency to estimate the conditional probability matrices P and Q.

$$P = \begin{pmatrix} 1 & 1 & \frac{2}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{3}{4} & 1 & \frac{2}{4} & \frac{2}{4} & \frac{1}{4} \\ \frac{2}{3} & \frac{2}{3} & 1 & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & 0 & 1 & \frac{2}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{2}{3} & 1 \end{pmatrix}, \quad Q = \begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 \\ \frac{1}{3} & \frac{2}{3} & 0 & 1 & \frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} & 1 & 0 & \frac{1}{3} \\ \frac{2}{3} & 1 & \frac{2}{3} & \frac{1}{3} & 0 \end{pmatrix}. \tag{3}$$

Algorithm 1: The framework of the proposed algorithm

Input :
- A stopping criterion;
- N: the size of the population.

Output: The best solution in the population

Step 1. Initialization
Step 1.1. Initialize a population X with a size of N.
Step 1.2. Use the improved greedy repair strategy to fix the population members.
Step 1.3. Compute the conditional probability matrices P and Q.
Step 2. Update
foreach $\mathbf{x}_i \in X$ **do**
  Compute the probability vector $\mathbf{p}$ of $\mathbf{x}_i$ by using the strategy described in section II.(C).
  Generate a new solution $\mathbf{x}_i^{new}$ according to $\mathbf{p}$ and fix it by the improved greedy repair strategy.
end
Select the first best N solutions form the combination of parents and the newly generated solutions to form the population X by using tournament selection.
Update the condition probability matrices P and Q.
Step 3. Stopping Criteria
If the stopping criteria is unsatisfied, then go to Step 2.
Otherwise, stop the algorithm and output the best solution in X.

C. Recombination operator based on the variable relationship matrixes

In this subsection, we propose a recombination operator based on the variable relationship matrices P and Q to generate a new solution $\mathbf{x}^{new}$ for a given solution $\mathbf{x} = (x_1, \cdots, x_n)$. The probability of $x_j^{new} = 0$ for $j = 1, \cdots, n$ are computed as follows:

$$p(x_j^{new} = 0) = \frac{1}{n-1} \sum_{i=1, i\neq j}^{n} (1 - x_i)p_{ij} + x_i q_{ij}. \tag{4}$$

Then, we compute the probability vector $\mathbf{p} = (p_1, \cdots, p_n)^T$ of the elements equaled to zeros of the new solution $\mathbf{x}^{new}$, which is generated by the solution $\mathbf{x}_1 = (1, 1, 1, 0, 0)^T$ in Eq (2). For instance, the probability of the first element equals to zeros, i.e., $p_1 = p(x_1^{new} = 0)$, can be calculated by:

$$\begin{aligned} p(x_1^{new} = 0) &= \frac{1}{5-1} (\sum_{i=2}^{n} (1 - x_i)p_{i1} + x_i q_{i1}, \\ &= \frac{1}{4}(q_{2,1} + q_{3,1} + p_{4,1} + p_{5,1}) = \frac{1}{4}. \end{aligned} \tag{5}$$

By this way, we can obtain the probability vector $\mathbf{p} = (\frac{1}{4}, \frac{5}{12}, \frac{7}{24}, \frac{17}{24}, \frac{3}{4})$. And we can generate a new solution according to this probability vector.

D. The framework of the proposed algorithm

The framework of the proposed optimization algorithm based on variable relationship excavating is given in Algorith-

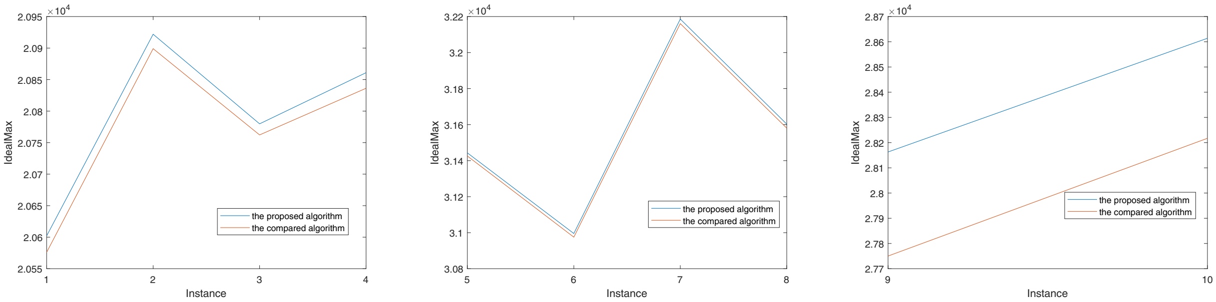

Fig. 1. Average best values of the proposed algorithm and the compared algorithm in 31 independent runs.

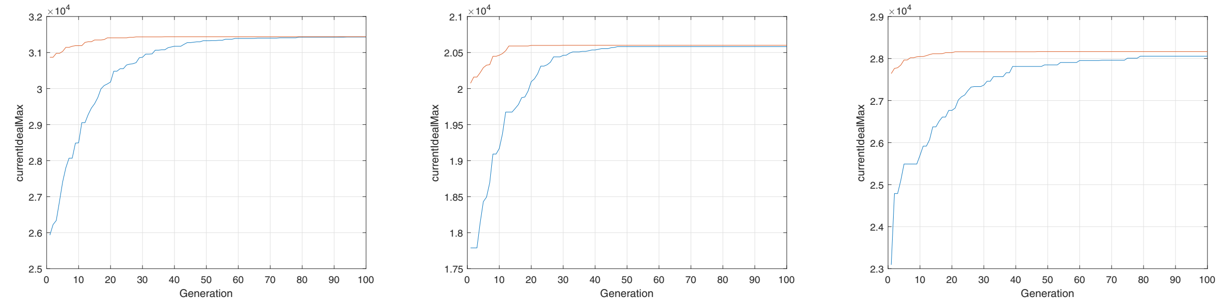

Fig. 2. The Convergence Efficiency of Test Instance 1, 5 And 9 in Two Algorithms

m 1. Different from traditional EAs method, instead of using one-point crossover or multi-point crossover operation which just chooses points into the mating pool randomly without guiding thoughts, we imply the recombination operator based on the variable relationship matrix which has a fully utilization of the whole population to guide the search process in every iteration. Besides, the proposed algorithm applies tournament selection to select the best individuals.

III. Experimental Study

In this section, 10 KPs [15] are considered to verify the effectiveness of the proposed algorithm. All 10 test instances are maximization problems. The amount of the knapsacks of these test instances has only one case, i.e. 1 knapsack, while the amount of items in each test instance has three types, i.e. 500, 750 and 1000. We compare the proposed algorithm with the algorithm using the multi-point crossover to generate the new solution in our experimental studies.

For each test instance, the population size N is set to be 100 in the proposed algorithm and the compared algorithm. Two algorithms stop after 100 generations, and each algorithm is running 31 independent times for each test instance.

A. Experimental results

Table I gives the best results obtained by the proposed algorithm and the compared algorithm in 31 independent runs for the each test instance, respectively.

From Table I and Fig. 1 we can see that, the value of the best solution in proposed algorithm is greater than the value of the compared one in all the 10 test instances.

The advantages of the proposed algorithm can be observed intuitively from Fig. 2. This figure shows the convergence efficiency of test instance 1, 5 and 9 in two algorithms, respectively. Because we only test one objective problem, we only present the figures of three typical instances having 500, 750, 1000 items respectively. From these figures, it is obvious that the proposed algorithm obtains superior converged solutions than the compared algorithm.

According to the results of different algorithms for solving the same example, it can be seen from Fig. 1 that the first and the second examples find almost same result because of the smaller scale. Through the experimental results of different scale KPs, it can be seen that with the increase of the size of the KPs, the performance of the compared algorithm deteriorates drastically, and the gap between the performance of the proposed algorithm and compared algorithms is larger. Thus, the compared one is not well suited for solving large-scale knapsack problems. For the second experiment, it can be seen from Fig. 2 that the proposed algorithm and the compared one get the best calculation results, but from the convergence efficiency, the iteration scale of the proposed algorithm is small. In summary, it can be seen from this set of experiments that the algorithm with variable relationship

TABLE I
THE AVERAGE OF BEST VALUES OF THE PROPOSED ALGORITHM AND MOEA/D-M2M IN 31 INDEPENDENT RUNS.

Instance	proposed	compared
1 (1, 500)	20602.13	20576.10
2 (1, 500)	20921.97	20899.06
3 (1, 500)	20779.81	20762.23
4 (1, 500)	20860.97	20836.23
5 (1, 750)	31443.13	31425.61
6 (1, 750)	30995.42	30975.19
7 (1, 750)	32186.00	32160.90
8 (1, 750)	31602.81	31580.48
9 (1, 1000)	28162.94	27750.19
10 (1, 1000)	28613.87	28217.23

matrix based crossover operation proposed in this paper is superior and efficient.

IV. CONCLUSION

This paper proposed optimization algorithm with a novel recombination operator. The recombination operator can excavate the variable relationship for solving 0-1 combinatorial optimization problems. It can make full use of the inherent information from multiple component of multiple individuals. The simulation results on ten knapsack problems show that the proposed algorithm is better than the compared algorithm in the ability of finding optimal solution and the efficiency.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation of China under Grant 61703108, the Natural Science Foundation of Guangdong Province 2017A030310467, 2018A030313505.

REFERENCES

[1] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multi-objective evolutionary algorithms on many-objective knapsack problems," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 264–283, 2014.

[2] Y. Yu, X. Yao, and Z.-H. Zhou, "On the approximation ability of evolutionary optimization with application to minimum set cover," *Artificial Intelligence*, vol. 180-181, pp. 20–33, 2012.

[3] H. Seada, M. Abouhawwash, and K. Deb, "Multiphase balance of diversity and convergence in multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 503–513, 2019.

[4] H.-L. Liu, F. Gu, Y.-M. Cheung, S. Xie, and J. Zhang, "On solving WCDMA network planning using iterative power control scheme and evolutionary multiobjective algorithm," *IEEE Computational Intelligence Magazine*, vol. 9, no. 1, pp. 44–52, 2014.

[5] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. P. L. F. de Carvalho, "A survey of evolutionary algorithms for clustering," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 39, no. 2, pp. 133–155, 2009.

[6] E. E. Korkmaz, "Multi-objective Genetic Algorithms for grouping problems," *Applied Intelligence*, vol. 33, no. 2, pp. 179–192, 2010.

[7] P. Merz and B. Freisleben, "Fitness landscape analysis and memetic algorithms for the quadratic assignment problem," *Evolutionary Computation IEEE Transactions on*, vol. 4, no. 4, pp. 337–352, 2000.

[8] V. A. Shim, K. C. Tan, and C. Y. Cheong, "A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem," *IEEE Transactions on Systems Man and Cybernetics Part C*, vol. 42, no. 5, pp. 682–691, 2012.

[9] Y. W. Leung and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimizationg," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, pp. 41–53, 2002.

[10] A.-B. M. K. A. Lim, T.Y., "Taming the 0/1 knapsack problem with monogamous pairs genetic algorithm," *Expert Systems with Applications*, vol. 54, no. 4, pp. 241–250, 2016.

[11] Z. He and G. Yen, "Many-objective evolutionary algorithm: Objective space reduction + diversity improvement," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 145–160, 2016.

[12] L.-K. Z. C. He, Y.C., "Greedy genetic algorithm for solving knapsack problem and its application use," *Comput. Eng. Des.*, vol. 28, no. 11, pp. 2655–2657, 2007.

[13] X. Zhao and T. Yang, "Very greedy particle swarm optimization algorithm for knapsack problem," *Computer Engineering and Applications*, vol. 45, no. 36, pp. 32–34, 2009.

[14] Y. E. Yong chun, L. X. Che, and H. E. Bing, "A kind of hybrid particle swarm optimization algorithm for solving 0-1 knapsack problems," *Journal of Changsha University of Electric Power*, vol. 21, no. 4, 2006.

[15] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.