

A Genetic Algorithm for Solving Knapsack Problems Based on Adaptive Evolution in Dual Population

YAN Tai-shan, GUO Guan-qi, Li Hong-min, He Wei
 School of Information and Communication Engineering
 Hunan Institute of Science and Technology
 Yueyang, Hunan, China
 Yantaishan163@163.com

Abstract—In order to solve knapsack problems efficiently, an improved genetic algorithm based on adaptive evolution in dual population (called DPAGA) is proposed. In DPAGA, the new population produced by selecting operation is regarded as main population. The population composed by the individuals washed out by selecting operation is regarded as subordinate population. The individual evolution strategy of main population is different from that of subordinate population. The crossover operators and mutation operators are all adjusted non-linearly and adaptively. DPAGA is used to solve knapsack problems. The experimental results show that its convergence speed and solution quality are all better than that of simple genetic algorithm. It is also suited to solve other optimization problems.

Keywords- Genetic algorithm; Dual population; Adaptive evolution; Knapsack problem

I. INTRODUCTION

Knapsack problem[1-7] is a well-known and well-studied problem in combinatorial optimization. It has a wide range of applications, for example network planning, network routing, parallel scheduling, and budgeting etc.. Mathematically, the 0/1 knapsack problem may be formulated as:

$$\text{Maximize: } f(x_1, x_2, \dots, x_n) = \sum_{i=1}^n v_i x_i$$

$$\text{Subject to: } \sum_{i=1}^n w_i x_i \leq c$$

where, $x_i \in \{0,1\}, i = 1, 2, \dots, n$. c denotes the capacity of knapsack, v_i denotes the value of object i , and w_i denotes the weight of object i . Generally, we assume that w_i, v_i and c are all positive integers.

Since the knapsack problem is a NP problem, the traditional algorithms have a large calculation and need a long running time when solve the knapsack problems with large scale[8]. In order to solve knapsack problems efficiently, an improved genetic algorithm based on adaptive evolution in dual population will be proposed in this paper.

II. GENETIC ALGORITHM BASED ON ADAPTIVE EVOLUTION IN DUAL POPULATION (DPAGA)

A. Basic Idea of the algorithm

Genetic algorithm based on adaptive evolution in dual population (DPAGA) is a novel improved adaptive genetic algorithm. Its population includes a new population formed by selecting operation (the main population) and a population eliminated by selecting operation (the subordinate population). Dual crossover operator and dual mutation operator are introduced in this algorithm. Dual crossover operator means the main crossover operator P_{bc} and the subordinate crossover operator P_{sc} . Dual mutation operator means the main mutation operator P_{bm} and the subordinate mutation operator P_{sm} . In the evolution of the main population, the individuals perform adaptive crossover operation with high probability and adaptive mutation operation with low probability according to the main crossover operator and the main mutation operator. And in the evolution of the subordinate population, the individuals perform adaptive crossover operation with low probability and adaptive mutation operation with high probability according to the subordinate crossover operator and the subordinate mutation operator.

B. Key operators

In the genetic algorithm based on adaptive evolution in dual population, key operators include selection operator, dual crossover operator and dual mutation operator. Selection operator is often realized by the roulette wheel selection method. The dual crossover operator and dual mutation operator are adjusted adaptively according to the individual fitness in the evolution of population. Their formulas can be described as follows:

$$P_{bc} = \begin{cases} P_{bc \min} + \frac{P_{bc \max} - P_{bc \min}}{1 + \exp(-(\frac{f' - f_{bavg}}{f_{b \max} - f_{bavg}}))}, & f' \geq f_{bavg} \\ P_{bc \max}, & f' < f_{bavg} \end{cases} \quad (1)$$

$$P_{bm} = \begin{cases} P_{bm \min} + \frac{P_{bm \max} - P_{bm \min}}{1 + \exp(-(\frac{f - f_{bavg}}{f_{b \max} - f_{bavg}}))}, & f \geq f_{bavg} \\ P_{bm \max}, & f < f_{bavg} \end{cases} \quad (2)$$

$$P_{sc} = \begin{cases} P_{sc \min} + \frac{P_{sc \max} - P_{sc \min}}{1 + \exp(-(\frac{f' - f_{savg}}{f_{s \max} - f_{savg}}))}, & f' \geq f_{savg} \\ P_{sc \max}, & f' < f_{savg} \end{cases} \quad (3)$$

$$P_{sm} = \begin{cases} P_{sm \min} + \frac{P_{sm \max} - P_{sm \min}}{1 + \exp(-(\frac{f - f_{savg}}{f_{s \max} - f_{savg}}))}, & f \geq f_{savg} \\ P_{sm \max}, & f < f_{savg} \end{cases} \quad (4)$$

Where, $f_{b \max}$ and $f_{s \max}$ denotes the maximal fitness of the main population and subordinate population respectively, f_{bavg} and f_{savg} denotes the average fitness of the main population and subordinate population respectively, f' denotes the higher fitness of the two crossing individuals, f denotes the fitness of the mutating individuals, $P_{bc \min}$ and $P_{bc \max}$ denotes the lower limit and the upper limit of the crossover probability of the main population respectively, $P_{sc \min}$ and $P_{sc \max}$ denotes the lower limit and the upper limit of the crossover probability of the subordinate population respectively, $P_{bm \min}$ and $P_{bm \max}$ denotes the lower limit and the upper limit of the mutation probability of the main population respectively, $P_{sm \min}$ and $P_{sm \max}$ denotes the lower limit and the upper limit of the mutation probability of the subordinate population respectively. Generally, their values may chosen as follows. $P_{bc \min} = 0.6$, $P_{bc \max} = 0.9$, $P_{sc \min} = 0.001$, $P_{sc \max} = 0.1$, $P_{bm \min} = 0.001$, $P_{bm \max} = 0.1$, $P_{sm \min} = 0.1$, $P_{sm \max} = 0.5$.

From formula (1) ~formula (4), we can see that the probability of crossover and the probability of mutation are adjusted nonlinearly according to the function

$$f(x) = \frac{1}{1 + \exp(-x)}$$

between the average fitness and the maximal fitness. When the fitness of the majority of individuals is near and the average fitness is close to the maximal fitness, the probability of crossover and the probability of mutation will be elevated. So, the individuals nearby the maximal fitness are preserved as many as possible. The ability of the algorithm to jump out the local convergence is enhanced greatly.

C. Steps and working flow of the algorithm

The working flow of DPAGA can be described as follows:

Step1: Code genetic individuals and initialize population P_t ;

Step2: Calculate individual fitness;

Step3: Perform selecting operation according to the roulette wheel selection method, and gain the new population P_{t+1} (main population) and the eliminated population P'_{t+1} (subordinate population);

Step4: For the main population P_{t+1} , perform adaptive crossover operation with high probability and adaptive mutation operation with low probability;

Step5: Judge convergence condition of P_{t+1} . If it is congruous, perform Step9;

Step6: For the subordinate population P'_{t+1} , perform adaptive crossover operation with low probability and adaptive mutation operation with high probability;

Step7: Form the new population P_t from the results of Step4 and Step6;

Step8: Repeat Step2~Step6, till the convergence condition is congruous;

Step9: Stop.

Fig.1 illustrates the working flow of DPAGA.

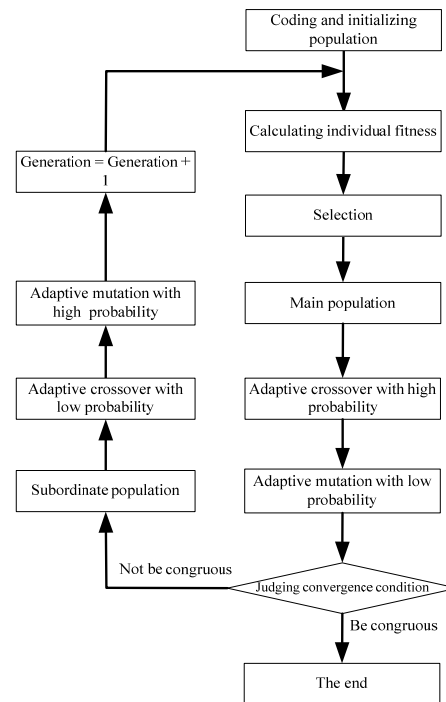


Figure 1. The working flow of DPAGA

III. EXPERIMENTS ON KNAPSACK PROBLEMS

In order to test the performance and effectiveness of the algorithm, the following two examples of knapsack problem are chosen in the experiment.

Example 1 The knapsack problem composed of 50 objects. It is described as:

$V = \{220, 208, 198, 192, 185, 180, 165, 162, 160, 158, 155, 130, 125, 122, 120, 118, 115, 110, 105, 101, 100, 100, 98, 96, 95, 90, 88, 82, 80, 77, 75, 73, 72, 70, 69, 66, 65, 63, 60, 58, 56, 50, 30, 25, 15, 10, 8, 5, 3, 1\}$,

$W = \{80, 82, 85, 70, 72, 70, 66, 50, 55, 25, 50, 55, 40, 48, 50, 32, 22, 60, 30, 32, 40, 38, 35, 32, 25, 28, 30, 22, 50, 30, 45, 30, 60, 50, 20, 65, 20, 25, 30, 10, 20, 25, 15, 10, 10, 10, 4, 4, 2, 1\}$, $c=1000$.

Example 2 The knapsack problem composed of 100 objects. It is described as:

$V = \{998, 997, 991, 986, 978, 977, 939, 936, 924, 920, 911, 901, 901, 885, 880, 866, 866, 863, 856, 842, 809, 794, 792, 789, 778, 767, 764, 764, 763, 759, 756, 747, 739, 708, 707, 706, 694, 693, 684, 680, 676, 652, 644, 640, 628, 12, 607, 597, 593, 570, 560, 556, 556, 556, 542, 538, 530, 530, 520, 498, 487, 466, 464, 461, 459, 456, 452, 443, 412, 399, 391, 383, 381, 378, 377, 359, 353, 351, 327, 317, 311, 295, 289, 287, 283, 269, 249, 248, 235, 193, 189, 189, 134, 108, 93, 74, 51, 48, 23, 8\}$,

$W = \{353, 180, 377, 230, 87, 174, 157, 390, 186, 213, 56, 86, 77, 215, 252, 90, 360, 187, 294, 299, 372, 384, 93, 328, 283, 99, 114, 374, 383, 183, 248, 164, 323, 263, 266, 318, 296, 196, 10, 324, 128, 376, 19, 280, 229, 225, 217, 134, 233, 35, 361, 302, 166, 374, 392, 319, 241, 15, 384, 82, 158, 322, 139, 239, 110, 44, 115, 23, 267, 82, 30, 198, 173, 70, 329, 125, 220, 107, 148, 159, 351, 56, 17, 99, 308, 396, 327, 235, 213, 223, 372, 376, 191, 299, 304, 277, 292, 391, 120, 37\}$, $c=9803$.

In this study, the simple genetic algorithm (SGA) and DPAGA are used to solve the above two knapsack problems. The parameters are set as: $T = 300$, $M = 100$, $p_c = 0.7$, $p_m = 0.1$. The results of example 1 are shown as Table 1. The results of example 2 are shown as Table 2.

From Table 1 and Table 2, we can see that when solve the knapsack problems, DPAGA is superior to SGA whether look from convergence speed or the quality of solutions.

IV. CONCLUSION

In this study, an improved genetic algorithm based on adaptive evolution in dual population was proposed. Its population includes a main population and a subordinate population. In the main population and subordinate population, the genetic operators are adjusted adaptively and nonlinearly with different strategies. The premature convergence of genetic algorithm is conquered and its global searching ability is enhanced greatly. Successful results were gained when experiments were taken on knapsack problems. Its convergence speed and solution

quality are all better than that of simple genetic algorithm. It is also suited to solve other optimization problems.

TABLE I. THE OPTIMAL SOLUTION OF EXAMPLE 1

Algorithm	Code of the optimal solution	Total value	Total weight	Time
SGA	11011011111010011011111111 1010000001010011000000011	3082	999	0.432
DPAGA	10011101111011011011011111 1110100001010011001001000	3105	1000	0.229

TABLE II. THE OPTIMAL SOLUTION OF EXAMPLE 2

Algorithm	Code of the optimal solution	Total value	Total weight	Time
SGA	11111111111111111111000111 11001111110011110111111111 0010000101101011110110010 1010001110000000000000000	40599	9796	6.152
DPAGA	11111111111111111101001111 11001111111011010111111111 0010000101101011110110010 1010001110000000000000000	40611	9799	3.627

ACKNOWLEDGMENTS

Support From the Science Research Project Foundation of Hunan Province Education Department under Grant No. 10C0757, the National Natural Science Foundation of China under Grant No. 60975049, the Natural Science Foundation of Hunan Province under Grant No.11JJ2037 is gratefully acknowledged.

REFERENCES

- [1] Lee Hae Sang, Lie Chang Hoon. A computation method for evaluating importance-measures of Gates in a fault tree. *IEEE Trans. on Reliability* , 1997 , 46 (3) :360 – 365
- [2] WANG Li, SHAO Ding-hong, LU Jin-gui. The Genetic Algorithm of Solving 0/1knapsack Problem. *Computer Simulation*, 1998,23(3):154~156
- [3] LI Kenli, LI Qinghua. An Adaptive algorithm for knapsack problem. *Computer research and development*, 2004,7(41):1292-1297
- [4] ZHU Xiaorong, ZHANG Xinghua. A quantum genetic algorithm with repair function and its application in knapsack question. *Computer applications*, 2007,22(5):1187-1190
- [5] HUANG Xiyue, ZHANG Zhuhong. *Morden intelligent algorithm - theory and application*. Beijing: Science Publishing Company, 2005.
- [6] He Yichao, Liu Jianqin etc.. A GREEDY DS_BPSO ALGORITHM FOR GENERAL KNAPSACK PROBLEM. *Computer Applications and Software*, 2008,25(4):230-232
- [7] YAN Tai-shan, He Wei. An Algorithm For Solving Knapsack Problems Utilizing Knowledge Evolution Principle. 2011 3rd International IEEE Workshop on Intelligent Systems and Applications (ISA2011), Wuhan, China, May 28-29, 2011, Vol.1: 1-4
- [8] Li, K.L., G.M. Dai and Q.H. Li. A genetic algorithm for the unbounded knapsack problem. *Proceedings of the 2nd International Conference on Machine Learning and Cybernetics*, USA., 2003, 1586-1590.