http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (5): 974-978, 2010
ISSN 1812-5638

An Improved Adaptive Genetic Algorithm Based on Human Reproduction Mode for Solving the Knapsack Problem

Yan Taishan
School of Information and Communication Engineering,
Hunan Institute of Science and Technology, 414000, Yueyang, Hunan, People's Republic of China

Abstract: In this study, human reproduction mode was introduced into genetic algorithm and an improved adaptive genetic algorithm based on human reproduction mode for solving 0/1 knapsack problem was presented. The genetic operators of this algorithm included selection operator, help operator, adaptive crossover operator and adaptive mutation operator. The genetic individuals' sex feature, age feature and consanguinity feature were considered. Two individuals with opposite sex could reproduce the next generation if they were distant consanguinity individuals and their age was allowable. Experiments were taken on two knapsack problems with different scale. The validity and excellent performance of this algorithm was proved by the experimental results. Its global convergence speed and optimal solutions were all better than that of many other algorithms.

Key words: Knapsack problem, human reproduction mode, genetic algorithm, adaptive operator, global optimization

INTRODUCTION

Knapsack problem (Sinnamon and Andrews, 1997; Wang and Cao, 2002) 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:

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

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

where, $x_i \in \{0, 1\}$, $i = 1, 2, \ldots, n_0$ 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 NP problem, various approaches presently available such as dynamic programming, backtracking, branch and bound etc. are not very useful for solving it. These exact algorithms have a large calculation and need a long running time when solve the knapsack problems with large scale (Li *et al*., 2003). So, it is very difficult for these algorithms to obtain the exact solutions in case of many large scale knapsack instances which come from practical applications. For those large scale instances, it has to rely on heuristic algorithms to obtain the near optimal solutions to them.

Amongst the heuristic algorithms for the knapsack problem, genetic algorithm is a search technique to find exact or approximate solutions to optimization and search problems (Wang and Cao, 2002; Tareeq *et al*., 2007; Ze-Su *et al*., 2007; Rong-Chen *et al*., 2009; Liu and Yan, 2007; Zhang *et al*., 2008). It is categorized as global search heuristics and has been used to solve the knapsack problems (Li *et al*., 2003; Shi and Dong, 2007; Dong, 2008; Wang *et al*., 1998; Bao, 2008; Ma and Ye, 2008; Li *et al*., 1998; He *et al*., 2008). However, the simple genetic algorithm has its own limitations such as large population, premature convergence etc. And its solutions for knapsack problems are still not satisfactory. Therefore, human own reproduction mode is introduced into genetic algorithm and an improved adaptive genetic algorithm based on human reproduction mode for solving the knapsack problem (named HRAGA) is presented in this study. The purpose is to improve the solutions' quality and the algorithm's work efficiency for the knapsack problems.

THE IMPROVED ADAPTIVE GENETIC ALGORITHM BASED ON HUMAN REPRODUCTION MODE FOR SOLVING THE KNAPSACK PROBLEM

The basic idea of improved adaptive genetic algorithm: Human reproduction mode is essentially different from that of other animals. So, the advancement of human quality and human civilization is superior to other animals. Firstly, in order to improve the quality of population, laws make explicit provision to marry age. Certainly, individual bearing ability will disappear naturally after a certain age. Secondly, human reproduction mode is a strict far consanguinity reproduction, but in the animal kingdom, incest phenomenon takes place usually. In order to avoid near consanguinity reproduction, corresponding laws are established to restrict human reproduction. Thirdly, human are able to adopt all kinds of effective measures actively to help offspring individuals including little embryos in order to ensure their healthy growth. And other animals only conform to environment passively.

Human reproduction mode is used for reference and an adaptive genetic algorithm based on human reproduction mode (HRAGA) is formed. The genetic individuals of this algorithm are separated into male individuals and female individuals, the age feature and consanguinity feature are fused into individuals. Two individuals with opposite sex can reproduce the next generation if they are distant consanguinity individuals and their age is allowable. The genetic operators include selection operator, help operator, adaptive crossover operator and adaptive mutation operator. The working flow of HRAGA can be described as follows:

Step 1: Code and initialize population
Step 2: Generation = 0
Step 3: Calculate individual fitness
Step 4: Perform the selection operation
Step 5: Judge individual age. Be permitted, perform step 6; otherwise, jump to step 4
Step 6: Perform the help operation and individual mating
Step 7: Judge individual difference. Be far consanguinity, perform step 8 directly; otherwise, perform step 8 after modification
Step 8: Perform the adaptive crossover and adaptive mutation operation
Step 9: Judge convergence condition. Be congruous, perform step 11; otherwise, perform step 10
Step 10: Generation = Generation+1, jump to step 3
Step 11: Stop

Coding method: Binary system is used in HRAGA. The individual code is consisted of two parts, the first of which is the code of individual exhibition and the second of which is the code of individual sex (Fig. 1).

Where, the code of individual sex is 0 or 1, 0 may be used to denote female individuals and 1 may be used to denote male individuals. Individual exhibition code is the genetic code of knapsack problem. It is arranged according to x_i ($i = 1, 2,\ldots,n$). Namely, individual exhibition code is X ($x_1, x_2,\ldots x_n$) where, $x_i = 0$ or 1. $x_i = 1$ means object i should be put into the knapsack; $x_i = 0$ means object i should not be put into the knapsack. For example, 110001100000000......0000000011 denotes a solution, it means that object 1, 2, 6, 7,..., n-1, n should be put into the knapsack and the other objects should not be put into the knapsack.

Individual exhibition	**Individual sex**

Fig. 1: The structure of individual code

Fitness function: As fitness is the only certainty index of individual choice in the population, so the form of fitness function determines the evolution behavior of population directly. In this paper, penalty function method is used to deal with the constraint condition of knapsack problem. The genetic individual's fitness function is chosen as follows:

$$\text{fit}(X_i^t) = \left[\sum_{j=1}^{n} v_j x_{i,j}^t\right] - P\left|\min\{0, c - \sum_{j=1}^{n} w_j x_{i,j}^t\}\right| \qquad (1)$$

where, I = 1, 2,..., G, G, G denotes the population size; t = 1, 2,..., T, T, T denotes the maximal generation; c denotes the capacity of knapsack; P is a large enough positive number.

Genetic operator: Selection operator: The selection method that lines up individual order between two generations is adopted to select optimal individuals in HRAGA. Re-line up all male individuals and female individuals of father generation and child generation respectively, then choose the anterior G/2 excellent individuals from male individuals set and female individuals set respectively as objects of crossover operation according to the population size G in allow of the age scope. Thus, not only the effective mating for individuals in crossover operation can be ensured, but also the excellent individuals of every generation can be reserved and the ill individuals may be washed out.

Help operator: In order to reinforce the ability of algorithm to jump out local optimal solutions and increase the speed of convergence, a help operator is used in HRAGA to help individuals according to the probability of help. Help operation is located after selecting operation and before mating operation. Under the condition that the individual's fitness doesn't reduce, its key operation is to change the individual's gene whose value is 0 according to the probability of help.

Adaptive crossover operator and adaptive mutation operator: Under binary coding mode, one-point crossover is a very effective crossover operation, it is adopted in HRAGA. Mutation means to choose some individuals stochastically from population under a small probability of mutation, then choose one bit or some bits of the chosen individuals stochastically and overturn their numbers, namely turn 1 to 0 or turn 0 to 1.

In the course of crossover and mutation, the probability of crossover and the probability of mutation are adjusted adaptively according to the following formulas:

$$P_c = \begin{cases} P_{c1} - \dfrac{(P_{c1} - P_{c2})(f' - f_{avg})}{f_{max} - f_{avg}}, & f' \geq f_{avg} \\ P_{c1}, & f' < f_{avg} \end{cases} \quad (2)$$

$$P_m = \begin{cases} P_{m1} - \dfrac{(P_{m1} - P_{m2})(f_{max} - f)}{f_{max} - f_{avg}}, & f \geq f_{avg} \\ P_{m1}, & f < f_{avg} \end{cases} \quad (3)$$

where, P_{cmin} and P_{cmax} denotes the lower limit and the upper limit of the probability of crossover respectively, f_{max} and f_{avg} denotes the maximal fitness and the average fitness of the population respectively, f' denotes the higher fitness of the two crossover individuals, fdenotes the fitness of mutation individual.

Judging individual age: The age of the individuals who will enter mating pool is judged after selection operation in HRAGA. Suppose that the maximal permitted mating age is N. The individual age of initial population P (0) is n_i (0) = 1 and individual age will be add one when the population evolves one times. The individual age n_i (t) of evolved new population will be noted after an evolution. If n_i (t)<N, the individual may be put into the mating pool. Generally, the maximal permitted age N is not bigger then 90% of the maximal evolution generation in order to prevent the individuals without reproduction ability from evolving.

Mating and examining individual consanguinity: In HRAGA, individuals with same sex are not permitted to mate. That is to say a male individual can only mate with a female individual. Further more, individuals mate in order of their excellence sequence, namely male individuals that lined up according to the excellence sequence mate with female individuals that lined up according to the excellence sequence one by one. It is advantageous for genetic algorithm to elevate the speed to search the global optimal solution and boost up the global convergence ability.

In order to avoid near relative reproduction, two opposite sex individuals still need to carry on an examination on individual difference. If two individuals are equal or the difference between their corresponding binary values is one, they can't cross and it is necessary to revise them. Here, the high bit value of individual code with low fitness should be modified by the different value of the high bit value of individual code with high fitness. Thus, far consanguinity reproduction between individuals can be ensured.

EXAMPLES AND EXPERIMENTAL RESULTS

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, 379,372, 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.

where, V denotes the value of objects, W denotes the weight of objects, c denotes the capacity of knapsack.

Table 1: The optimal solution of example 1

Algorithm	The code of the optimal solution	Total value	Total weight
SGA	1101101111101001101111111 1010000001010011000000011	3082	999
HRAGA	1101101111101001101101111 1010001001010011001000011	3117	1000

Table 2: The optimal solution of example 2

Algorithm	The code of the optimal solution	Total value	Total weight
SGA	1111111111111111111000111 1100111111001111011111111 0010000101101011110110010 1010001110000000000000000	40599	9796
HRAGA	1111111111111111111000111 1100111111001101011111111 0010000101101011110110110 1011001110000000000000000	40627	9797

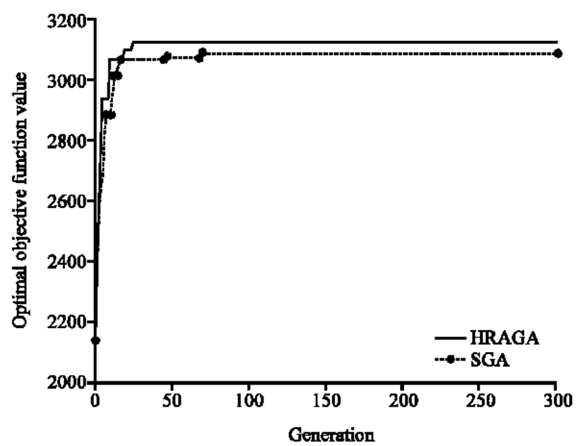

Fig. 2: The convergence curve of SGA and HRAGA for example 1

In this study, the Simple Genetic Algorithm (SGA) and HRAGA are used to solve the above two knapsack problems. The parameters are set as: T = 300, M = 100, P_{c1} = 0.9, P_{c2} = 0.6, P_{m1} = 0.1, P_{m2} = 0.01, p_h = 0.1. The algorithm programs run 50 times respectively. The results of example 1 are shown as Table 1 and Fig. 2. The results of example 2 are shown as Table 2 and Fig. 3.

DISCUSSION

In the present experiments, example 1 and 2 are two knapsack problems with different scale. They are solved by SGA and HRAGA presented in this study, respectively.

For example 1, the optimal objective function value of SGA is 3082 and its average evolution generation is 70. The optimal value of HRAGA is 3117 and its average evolution generation is 25. The optimal value of the greedy algorithm (Li *et al.*, 1998) is 3036. The optimal value of the hybrid genetic algorithm (Li *et al.*, 1998) is 3103. The optimal value of the Parallel Particle Swarm Optimization Algorithm Based on Cultural Evolution (Ma and Ye, 2008) is 3103. It is shown that the optimal value of HRAGA for example1 is better than that of SGA, the greedy algorithm (Li *et al.*, 1998), the hybrid genetic algorithm (Li *et al.*, 1998) and the Parallel Particle Swarm Optimization Algorithm Based on Cultural Evolution (Ma and Ye, 2008).

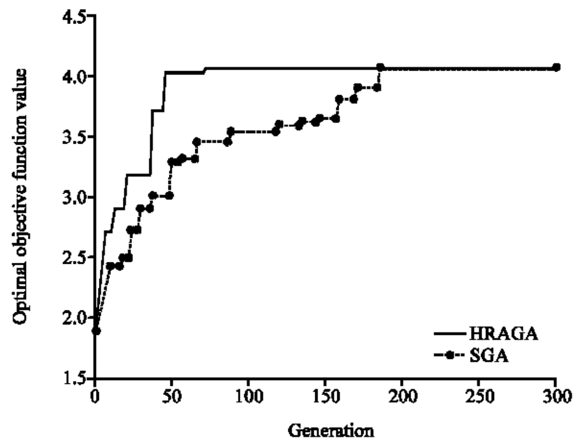

Fig. 3: The convergence curve of SGA and HRAGA for example 2

For example 2, the optimal objective function value of SGA is 40599 and its average evolution generation is 182. The optimal value of HRAGA is 40627 and its average evolution generation is 71. The optimal value of the hybrid genetic algorithm (He *et al.*, 2008) is 40613. The optimal value of the GDS_BPSO algorithm (He *et al.*, 2008) is 40627. It is shown that the optimal value of HRAGA for example 2 is better than that of SGA and the hybrid genetic algorithm (He *et al.*, 2008) and it is the same as that of the GDS_BPSO algorithm (He *et al.*, 2008).

From the above comparison, we can see that good results have been achieved by HRAGA presented in this study when solve the knapsack problems. Its performance is better than many other algorithms.

CONCLUSION

Human reproduction mode is introduced into genetic algorithm and an improved adaptive genetic algorithm for solving 0/1 knapsack problem is presented in this study. The algorithm simulates human reproduction mode which is different from other animals. It is not easy to fall into the local optimum and its ability to search the global optimum is improved. Experiments show that the searching speed and accuracy of this algorithm are all better than many other algorithms. It is a more efficient algorithm for solving knapsack problems.

ACKNOWLEDGMENTS

Support from the Science Foundation of Hunan Institute of Science and Technology under Grant No. 2009Y08 is gratefully acknowledged.

REFERENCES

Bao, J.H., 2008. Solution to multiple-choice knapsack problem using penalty function method implemented by genetic algorithm. Comput. Eng. Design, 29: 4518-4520.

Dong, P., 2008. Hybrid genetic algorithm with high convergent speed for solving 0/1 knapsack problem. Comput. Eng. Appl., 40: 47-49.

He, Y.C., J.Q. Liu and W.L. Qu, 2008. A Greedy ds_bpso algorithm for general knapsack problem. Comput. Appl. Software, 25: 230-232.

Li, J., P. Fang and M. Zhou, 1998. A hybrid genetic algorithm for knapsack problem. J. Nanchang Inst. Aeronautical Technol., 3: 31-35.

Li, K.L., G.M. Dai and Q.H. Li, 2003. A genetic algorithm for the unbounded knapsack problem. Proceedings of the 2nd International Conference on Machine Learning and Cybernetics, (MLC'03), IEEE Xplore Press, USA., pp: 1586-1590.

Liu, M. and J.W. Yan, 2007. An adaptively annealing genetic algorithm based scheduling method of workshop daily operating planning. Chinese J. Comput., 30: 1164-1172.

Ma, H.M. and C.M. Ye, 2008. Parallel particle swarm optimization algorithm based on cultural evolution. Comput. Eng., 34: 193-195.

Rong-Chang, C., C. Ting-Tsuen and F. Wei-Luen, 2009. Assignment of external off-the-job training courses to employees using genetic algorithm. Inform. Technol. J., 8: 147-155.

Shi, L. and H.L. Dong, 2007. Genetic algorithm based on active evolution for large scale 0-1 knapsack problem. Comput. Eng., 33: 31-33.

Sinnamon, R.M. and J.D. Andrews, 1997. New approaches to evaluating fault trees. Reliability Eng. Syst. Safety, 58: 89-96.

Tareeq, S.M., R. Parveen, L.J. Rozario and M. Al-Amin-Bhuiyan, 2007. Robust face detection using genetic algorithm. Inform. Technol. J., 6: 142-147.

Wang, L., D.H. Shao and J.G. Lu, 1998. The genetic algorithm of solving 0/1knapsack problem. Comput. Simulation, 23: 154-156.

Wang, X.P. and L.M. Cao, 2002. Genetic Algorithm-Theory, Application and Software Realization. 1st Edn., Xian Jiao Tong University Press, Xian, China, ISBN:7-5605-1448-0.

Ze-Su, C., H. Bing-Rong and L. Hong, 2007. An improved polar scan matching using genetic algorithm. Inform. Technol. J., 6: 89-95.

Zhang, Q., B. Wang, R. Zhang and C.X. Xu, 2008. Genetic algorithm-based design for DNA sequences sets. Chinese J. Comput., 31: 2193-2199.