Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 7 – No. 6, September 2017 – www.caeaccess.org

Greedy Approach for Optimizing 0-1 Knapsack Problem

Shafiqul Abidin
Department of Information Technology,
HMR Institute of Technology and Management
GGSIP University, Delhi, India

ABSTRACT
Abstract: 0--1 knapsack problem is a typical NP complete problem in field of computer. Traditional knapsack problem is solved recursively using backtracking, branch and bound, dynamic programming and greedy methods. The advantages of using recursive backtracking to solve knapsack problem algorithm are – its simplicity, completely traverse the search space and sure to find the optimal solution. But the solution space is exponential growth, when extended to certain extent. This algorithm will solve the knapsack problem unrealistically. The greedy algorithm can only be used to get Approximate Solution in a certain range near the optimal solution.

This paper is based on 0-1 knapsack problem, a mathematical model, and analysis of the greedy strategy. Effort is made to give a genetic algorithm that solves the knapsack problem. Greedy strategy along with the traditional genetic algorithm has been improved that shorten the time of solution. Further, it improves the accuracy of the solution.

Keywords
Genetic Algorithm, 0-1 Knapsack Problem, Greedy Strategy.

1. INTRODUCTION
Knapsack problem is a typical NP-hard problem. It is actually a mathematical model for 0 - 1 programming problems. Let us assume that there are N objects, and its weight is w_i with that profit $p_{i\ where\ <}$ i = 1, 2, ···, n> . the Knapsack maximum weight capacity is A. If i th object is selected then definition of selection variable x_i = 1, otherwise x_i = 0. to consider the choice of objects within the

total backpack weight is $\sum_{i=1}^{n} w_i * x_i$ the total value is

$\sum_{i=1}^{n} p_i * x_i$ how to the decision goods variables x_i (i = 1, 2, ···, n)to profit, so that the total profit of the objects within the knapsack is maximize. Its mathematical model as follows

$$\begin{cases} \max f(x_1, x_2 \cdots x_n) = \sum_{i=1}^{n} p_i x_i & 1 \\ \sum_{i=1}^{n} w_i x_i < A & 2 \\ x_i \in \{0,1\} & 3 \end{cases}$$

Expression: the describes of Knapsack problem

Expression 1 which describes the total profit of the selected objects, Expression 2 describes the total feasible weight a knapsack can accommodate, Expression 3 describes that each selection variable can hold Boolean values.(1 if object is selected ,0 otherwise)

2. ANALYSIS OF GREEDY STRATEGY
Traditional use of greedy algorithm to solve the 0-1 knapsack problem is because it largely depends on the structure and problems. Sometimes the solution is to be verified that is different from the optimal solution. We have used and analysis this strategy for the following key features.

2.1 Greedy Criteria of Profit
The steps are as follows: Objects are loaded into knapsack according to the decreasing order of their weight(i.e. large weight object is select first then next lower weight object select next and so on till capacity of knapsack). Consider n = 3, w = [20,10,8], p = [15,10,10], capacity A =20. When using the profit of the greedy strategy, the solution obtained for x = (1,0,0), the total profit of this program is 15. But The optimal solution for the (0,1,1), the total profit of 20. This can be seen from this example, this strategy can not guarantee the optimal solution. The reason is that, although every step of the current benefits was the largest increase in profit, but some cases, excessive consumption of backpacks available capacity.

2.2 Greedy Criteria of Capacity
This strategy considers the selection of objects whose weight is smallest, and each time to find the weight of the smallest from remaining unselected objects put to the knapsack . Using this strategy optimal solution of the previous example can be obtained (0,1,1). But under normal circumstances it is not possible to get the optimal solution. Consider the example of n = 2, w = [10,20], p = [5,100], capacity A = 25. The use of this strategy to get the solution for (1.0) profit is 5 and the actual optimal solution for the opposite (0,1) the profit is 100.This is possible because sometime, the volume gets consumed slowly, but the profit does not increase rapidly and accordingly.

2.3 Greedy Criteria of Profit Density
This strategy considers both options of the above 2 strategy. The steps of this strategy are as follows: Choose from the remaining unselected objects who has the largest rate of vi / wi in the package. This type of strategy can not guarantee that will get the optimal solution. Solution using this strategy considering that n = 3, w = [20,15,15], v = [40,25,25], capacity A= 30 and when the solution obtained for x = (1,0,0), while the optimal solution for x = (0,1,1).

3. GENETIC ALGORITHM
Genetic algorithm came into existent from Holland in 1960 and its basic idea comes from Darwin's theory of evolution and

Mendel's genetics, The purpose of the initial study of the adaptive behavior of natural systems is to design the adaptive software systems It is a space-based search method. This is generated by natural selection, genetic variation, such as operations, as well as Darwin's theory of survival of the fittest mechanism. This is helpful for simulation of natural biological evolution of an algorithm. That is to follow the survival of the fittest, the law of survival of the fittest, that is, useful in optimizing the process of reservation, the removal of useless. In the science and practice production performance best to meet the conditions required by the solution from all possible solutions of the problem of how, that is, to find a optimal solution. The general problem-solving steps are as follows:

i. Initialize the population
ii. Calculation of the population on the fitness of each individual profit
iii. According to the individual fitness profit
iv. Determined by a rule choose to enter the next generation of individual
v. By a crossover probability
vi. By a mutation probability
vii. If it does not meet the conditions, then
viii. Transferred to step 2, or enter the next step
ix. Output the profit of in the best fitness
x. Chromosome as a satisfactory solution and the optimal solution calculation will stop as per the following conditions

A: completed a pre-evolution of a given algebra
B: there are individual profits of several generations did not improve the average fitness for several generations.

4. GREEDY STRATEGY BASED ON GENETIC ALGORITHM IN KNAPSACK PROBLEM

4.1 Gene Encoding & Population Initialization

Population initialization is the first step of genetic algorithm, the first generation that is produced. Here because of our solution space is N of the Objects are selected, , taking the profit of X[i] 0 or 1, the traditional genetic algorithms generally use a random number sequence 2, the profit of stocks as the initialization code, as the result of random Iterative calculation time is too many. So here we also use *Xi* (i in□<0,1 ... n>) which take the profit of 0,1 to 1 if object is selected, 0 is not selected on behalf of the traditional genetic algorithm is set up a group of random numbers to initialize the population, However, we choose to use a greedy strategy for the approximate solution to the binary code as part of our population initialization code. This algorithm can effectively reduce the number of iterations.

4.2 The Fitness Function

Genetic algorithm fitness function calculation for determining decision whether or not choice individual. Here we adopted the profit of the current program dividing by the total profit of objects as the fitness function.

4.3 Select Computing

The use of genetic algorithm selection operator (or copy operations) is to achieve the group for the survival of the fittest individual steps:. Individuals with high fitness to the next generation by genetic probability of large groups; low fitness individuals are genetic to the probability of the next generation in small groups. Here we use the profit of individual fitness in the fitness profit of all proportion and the selection of the probability of conversion and then use the roulette wheel to choose individual.

$$\text{Probability of individual choice, is } p_i = \frac{\text{adapt}_i}{\sum_{i=1}^{n} \text{adapt}_i} \text{ in}$$

Which individual fitness for the purpose of calculating the profit of adapt*i.*

4.4 Computing of Crossover

Crossover-operator, is the pairing of two chromosomes in one way or another by exchanging some of its genes, to form two new individuals. Crossover-operator in genetic algorithm is different from the other important feature of evolutionary algorithm, genetic algorithm it plays a key role is to generate a new individual in the main method we have adopted a uniform cross-here it refers to the two pairs of each individual gene in order to The probability of the same exchange, to form two new individuals. Specific operation is as follows: randomly generated with the same length of individual binary encoding mask word W = w1 w2 ... wn; by the following rules from the A, B two parent individuals produce two new individuals X, Y: if wi = 0 Then the i genes of X, inherited A corresponding gene, Y gene of the first i of the succession of B the corresponding gene; if wi = 1, then A, B, paragraph i of the mutual exchange of genes, which generate X, Y of the first i genes.

4.5 Computing of Mutation

Mutation operator, refers to the individual encoded string profit of some of the genes used to replace the profit of other genes to form a new individual. Variation in the genetic algorithm operation is a new method of individual support, but it is an essential step in computing, because it determines the genetic algorithm local search capabilities. Crossover operator and mutation operator of the mutual co-completion of the search space of the global search and local search.

4.6 Algorithm based on mutation

Probability p_m * n randomly selected individually to carry out mutation operation for each individual, a variation of random mutation position, that is, if the individual will be a corresponding gene for turning it into 0, but if the corresponding gene is 0, you need to determine the variation arising from the new profit of the individual constraints is greater than A, if more than A, while for corrective action. Calculation of born is to be bounded by the conditions of individual profits. Algorithm termination is accordance with the above mentioned conditions. For the termination of genetic algorithms one of the following conditions will be required to complete the set genetic algebra.

5. ANALYSIS OF EXPERIMENTAL RESULTS

In the experiment, we checked the size knapsack for N =50, with the weight of W [50] = (35,34,31, ... 28,27) and profit of P [50] = (13,18,21 ... ... 25 , 23), we get 100 iterations, crossover probability of 0.15 mutation probability of 0.05. The use of

traditional genetic algorithm executed 50 times to obtain the optimal solution for an average of 3382.24, with an average of 38 algebra, and genetic algorithm based on greedy strategy solution. We have used the greedy strategy result as to be the first to use the initialization code for the population (1101 ... ... .101) optimal solution for the average 1412.19, with an average of 27 algebra. again and again regardless of the time deposit or to be the optimal solution. This is to be analyzed based on the greedy strategy of the genetic algorithm significantly superior to traditional algorithms.

6. CONCLUSION

In this paper we have analyzed three kinds of commonly used greedy strategies to solve 0-1 knapsack problem. Then combine with the basic principles of genetic algorithms. We have given a genetic algorithm based on Greedy strategy to solve 0-1 knapsack problem. The improvement of this genetic algorithm lies in the establishment of the original population, to use of greedy strategy solution as the original code of the algorithm. This approach reduces the Diego generation time and increases efficiency, So this algorithm has some reference profit for solving the 0-1 knapsack problem.

7. REFERENCES

[1] Yang Liu The Genetic Algorithm of Solving 0-1'sKnapsack Problem and Its Improvement[J] Tianjin Normal University Natural Science Edition 9/2003J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Wenlan Chen, Jian Hua Liu a genetic algorithm in Solving a knapsack problem Computer of FuJian 5/2006.

[3] TanShan Yan Greedy algorithm based on Hybrid Genetic Algorithm for Solving 0 / 1 knapsack problem Research and Development 7/2006.

[4] HongWei Huo,Jin Xu,Zheng Bao Solv ing 0-1 knapsack problem us ing genetic algorithm The Journal of Xi'an University of Electronic Science and Technology August 1999.

[5] GuangLin Shi, WeiXiang Shi Genetic Algorithm and its New Progress in Research and Application Basic science April of 1997.

[6] Shafiqul Abidin, A Novel Construction of Secure RFID Authentication Protocol", International Journal of Security, Computer Science Journal, Malaysia, Vol. 8, Issue 8, October 2014.

[7] Shafiqul Abidin, Key Agreement Protocols and Digital Signature Algorithm" International Journal of Current Advanced Research, August, 2017.