A New Approach to Solve Knapsack Problem

Prof.A.V.Pande

Department of Computer Science and Engineering, Sant Gadge Baba Amravati University

ankita.pande2@gmail.com

Prof.Y.A.Thakare

Department of Computer Science and Engineering, Sant Gadge Baba Amravati University

yugathakare@gmail.com

***Abstract*— The Knapsack Problem belongs to a large class of problems known as Optimization Problem. This problem is to maximize the obtained profit without exceeding the knapsack capacity. It is a very special case of the well-known Integer Linear Programming Problem. The purpose of this paper is to analyze several feasible solutions to a Fractional Knapsack Problem using greedy approach. Based on the knapsack algorithm to take different feasible solutions, in this set of feasible solutions, particular solution that satisfies the objective of the function. Such solution is called optimum solution. The optimum selection is without revising previously generated solutions. The greedy choices are made one after the other, reducing each given problem instance to smaller one. The greedy choices bring efficiency in solving the problem with the help of sub problems.**

***Keywords-* Knapsack, greedy approach,feasible solution,optimal solution,optimization problem.**

I. INTRODUCTION TO GREEDY APPROACH

There are several categories of algorithms such as greedy approach, dynamic programming, backtracking, branch and bound and so on. Knapsack problem is kind of algorithm which falls in greedy category of algorithms. Greedy approach is kind of an algorithm used for optimization problem. The algorithm which is greedy in nature tries to find a solution which is suitable at that moment. Greedy approach does not always guarantee optimal solution but when it is used it will gives you most simplest and efficient solution. Let's consider real life example of greedy approach, if you have to travel from source A to source B. And there are numerous ways to reach to the destination B then in that case greedy method will choose an option that will be best or suitable at that moment. But it is not necessary that you will definitely reach to destination B.Greedy method is a straight forward method. This method is popular for obtaining the optimized solutions. In a greedy technique, the solution is constructed through a sequence of steps, each step examining a partially constructed solution obtained so far, until a complete solution to the problem is reached. At each step the choice made should be

Feasible – It should satisfy the problem's constraints.

Locally optimal – Among all feasible solutions the best choice is to be made.

Irrevocable – Once the particular choice is made then it should not get changed on subsequent steps.

Actually there are two types of solutions produced by greedy method. One is feasible solution and another one is optimal solution. Feasible solution is the one which is generalized solution. That means feasible solution is not exact or not more appropriate solution to the given problem.

For instance consider a quadratic equation

$x2 + 5x + 6 = 0$……Equation(1)

The problem is to find out maximum root of the given equation. Then according to the feasible solution answer for equation (1) would be x = 3 and x = 2. The next category of solutions is optimal solution. Optimal solution is one which gives more exact or accurate solution to the given problem. Optimal solution for equation (1) would be x=3.

II. KNAPSACK PROBLEM WITH EXAMPLE

The Knapsack Problem is an example of a optimization problem, which seeks for a best solution from among many other solutions. A thief robbing a store and can carry a maximum weight of W_i into their knapsack.There are n objects, from i=1,2,3…n. Each object has a particular weight W_i and obtains a particular profit P_i.What object should thief take? This version of problem is known as Fractional knapsack problem. The setup is same,but the thief can take fractions of items, meaning that the items can be broken into smaller pieces so that thief may decide to carry only a fraction of x_i of item i, where $0 \leq x_i \leq 1$. The aim is to fill the knapsack using various items so that the total weight of the items does not exceed the capacity of the knapsack. To choose only those objects that give maximum profit of the included objects. The total weight of selected objects should be ≤ M.

For a knapsack problem we have given a sack with ‘n’ number of data items has to be filled in that sack. Every data item has associated its weight value. We have to fill that bag/sack with given data items up to its maximum capacity of sack. And if that data item is sell out in the market will get its profit value. The main objective of this algorithm is to achieve maximum profit value in any situation. As this problem is kind of a greedy method then it also has two solutions. These two solutions with its formulas are as follows.

Feasible Solution

$\sum Wi * xi <= M$……Equation(2)

Optimal Solution

Total Profit $= \sum Pi * xi$…… Equation(3)

Where, Wi= Weight of an i^{th} object.

Pi=Profit of an ith object

xi= It is the fraction of object that will be inserted in sack.

M=Maximum capacity of the back

There are four strategies to solve the given problem using greedy method.

1) Random Approach

2) Minimum Weight

3) Maximum Profit

4) Maximum Profit/Weight

1) **Random Approach-**According to this strategy, select an object randomly and add it to that object into the sack. And then calculate the total profit value of selected object.

2) **Minimum Weight-**According to this strategy, select an object that is having minimum weight. And then calculate the total profit value of selected object.

3) **Maximum Profit-** According to this strategy selects an object that is having maximum weight. And then calculate the total profit value of selected object.

4) **Maximum Profit/Weight-** According to this strategy, select an object that is having maximum ratio of Profit/Weight. And then calculate the total profit value of selected object.

Example 1:-

Consider an example to find an optimal solution to the following instance of knapsack problem if n=5 M=100 (p1,p2,....p5)=(20,30,66,40,60) and (w1,w2,...w5)=(10,20,30,40,50)

Given:-

N	1	2	3	4	5
W	10	20	30	40	50
P	20	30	66	40	60

1) Random Approach

In this strategy, select any of the data items as per your choice randomly. Only make sure that maximum capacity of sack should not exceed the overall weight of the sack.

N	1	3	4	2	5
W	10	30	40	20	50
P	20	66	40	30	60
Random Approach	1	1	1	1	0

Feasible Solution

$\sum Wi * xi <= M$

$10*1+30*1+40*1+20*1<=100$

$100<=100$ // Condition satisfied

Optimal Solution

Total Profit $=\sum Pi * xi$

Total Profit $=20*1+66*1+40*1+30*1$

Total Profit $=156$

2) **Minimum Weight:-** In this strategy arrange the given data in ascending order of their weights. And each time check the weight of the selected object and current capacity of sack so that it should not exceed the maximum capacity of sack.

N	1	2	3	4	5
W	10	20	30	40	50
P	20	30	66	40	60
Minimum Weight	1	1	1	1	0

Feasible Solution

$\sum Wi * xi <= M$

10*1+20*1+30*1+40*1<=100

100<=100 // Condition satisfied

Optimal Solution

Total Profit =$\sum Pi * xi$

Total Profit=20*1+30*1+66*1+40*1

Total Profit=156

3) **Maximum Profit:-**In this strategy arrange the given data in descending order of their profits. And each time check the weight of the selected object and current capacity of sack so that it should not exceed the maximum capacity of sack.

N	3	5	4	2	1
W	30	50	40	20	10
P	66	60	40	30	20
Maximum Profit	1	1	20/40	0	0

Feasible Solution

$\sum Wi * xi <= M$

30*1+50*1+40*20/40<=100

100<=100 // Condition satisfied

Optimal Solution

Total Profit =$\sum Pi * xi$

Total Profit =66*1+60*1+40*20/40+30*0+20*0

Total Profit =146

4) **Maximum Profit/Weight:-** This is suggested new approach of this paper.In this strategy, first step is to calculate the ratio of Profit/Weight.

N	1	2	3	4	5
W	10	20	30	40	50
P	20	30	66	40	60
Profit/Weight	2	1.5	2.2	1	1.2

Then arrange the given data in descending order of their profits. Each time check the weight of the selected object and current capacity of sack so that it should not exceed the maximum capacity of sack.

N	3	1	2	5	4
W	30	10	20	50	40
P	66	20	30	60	40
Profit/Weight	2.2	2	1.5	1.2	1
Maximum Profit/Weight	1	1	1	40/50	0

Feasible Solution

$\sum Wi * xi <= M$

30*1+10*1+20*1+50*40/50<=100

100<=100 // Condition satisfied

Optimal Solution

Total Profit =$\sum Pi * xi$

Total Profit =66*1+20*1+30*1+60*40/50

Total Profit =164

After analyzing the above four strategies for solving knapsack problem, it is clearly observed that the last strategy i.e. Maximum Profit/Weight ration strategy gives us an optimal solution with maximum profit value.

III. ALGORITHM AND ITS DESCRIPTION

Algorithm for Knapsack Problem

Step 1: For i←1 to n do

Step 2: If (W[i]>M) then

Step 3: return;

Step 4: else

Step 5: x[i]=1.0;

Step 6: M=M-W[i]

Step 7: Max_profit=P[i] * x[i]

Step 8: End of If Loop

Step 9: End of For Loop

Step 10: If(i<n) then

Step 11:x[i]=M/W[i];

Step 12: Max_profit = Max_profit + (P[i] * x[i]);

Step 13: End of If Loop

Step 14:Print the Max_profit value.

Where Max_proft=Maximum Profit Value

Wi= Weight of an i^{th} object.

Pi=Profit of an ith object

xi= It is the fraction of object that will be inserted in sack.

M=Maximum capacity of the back

Analysis of Knapsack Algorithm

If the provided items are already sorted into a decreasing order of pi/wi then the for loop takes a time in O(n); Therefore, the total time including the sort is in O(n logn).

IV. CONCLUSION

This paper proposed an new approach of algorithm to solve the knapsack problem. This algorithm originates from a well-known continued fraction method and runs in polynomial time of the input length. We conjecture that for any fixed n>2, the knapsack problem with n variables may be solved in polynomial time. The proof seems difficult and generalization of the method used in this paper doesn't seem to work. It is hoped that the modest beginning presented in this paper will draw the attention of more researchers and will bring about the eventual solution of this problem.

REFERENCES

[1] Horowitz, E and Sahni, S. "Computing partitions with applications to the knapsack problem", JACM 21, 2(April 1974), 277-292.

[2] Sri Lakshmi Kanagala, "A Study of Analyzing Greedy Approach for Fractional Knapsack Problem", IJARCCE,Volume 5, Issue 2, February 2016.ISSN : 2278-102.

[3] Sartaj Sahni, "Data Structures, Algorithms, and Applications in C++", China Machine Press, (2008).

[4] Parag H. Dave and Himanshu B.Dave , "Design and Analysis of Algorithms", Pearson Education India, 2009.

[5] Aho,Hopcroft & Ullman "The Design & Analysis of Computer Algorithms", Addison-Wesley.

[6] Cormen, T.H, Lierson & Rivest: " Introduction to Algorithms", McGraw-Hill.