

#### JHANVI PAREKH

60009210033

**CSE(Data Science)** 

#### **Experiment 4**

(Greedy Algorithm)

**Aim:** Implementation of fractional Knapsack using greedy algorithm.

#### **Theory:**

Given a set of items, each with a weight and a value, determine a subset of items to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.

The knapsack problem is in combinatorial optimization problem. It appears as a subproblem in many, more complex mathematical models of real-world problems. One general approach to difficult problems is to identify the most restrictive constraint, ignore the others, solve a knapsack problem, and somehow adjust the solution to satisfy the ignored constraints.

#### **Applications:**

In many cases of resource allocation along with some constraint, the problem can be derived in a similar way of Knapsack problem. Following is a set of example.

- Finding the least wasteful way to cut raw materials
- portfolio optimization
- Cutting stock problems

In this case, items can be broken into smaller pieces, hence the thief can select fractions of items.

According to the problem statement,

- There are n items in the store
- Weight of  $i^{th}$  item  $w_i > 0$
- Profit for i<sup>th</sup> item p<sub>i</sub>>0 and
- Capacity of the Knapsack is W

## **Pseudocode:**

Greedy-Fractional-Knapsack (w[1..n], p[1..n], W) for i = 1 to n do x[i] = 0 weight = 0 for i = 1 to n

if weight  $+ w[i] \le W$  then x[i] = 1weight = weight + w[i]

else x[i] = (W - weight) / w[i] weight = W break return x

#### **Complexity:**

Time Complexity: O(n logn).

#### **Example:**

Problem: Consider the following instances of the fractional knapsack problem: n = 3, M = 20, V = (24, 25, 15) and W = (18, 15, 20) find the feasible solutions.

#### Solution:

Arrange items by decreasing order of profit density. Assume that items are labeled as X = (I1, I2, I3), have profit  $V = \{24, 25, 15\}$  and weight  $W = \{18, 15, 20\}$ .

Item (x <sub>i</sub> )	Value (vi)	Weight (w <sub>i</sub> )	$p_i = v_i / w_i$
$I_2$	25	15	1.67
$I_1$	24	18	1.33
I <sub>3</sub>	15	20	0.75

Initialize, Weight of selected items, SW = 0,

Profit of selected items, SP = 0,

Set of selected items,  $S = \{ \},$ 

Here, Knapsack capacity M = 20.

Iteration 1 :  $SW = (SW + W_2) = 0 + 15 = 15$ 

SW  $\leq$  M, so select I<sub>2</sub>

$$S = \{ I_2 \}, SW = 15, SP = 0 + 25 = 25 \}$$

Iteration 2 : SW +  $w_1$  > M, so break down item  $I_1$ .

The remaining capacity of the knapsack is 5 unit, so select only 5 units of item I<sub>1</sub>.

frac = 
$$(M - SW) / W[i] = (20 - 15) / 18 = 5 / 18$$

```
S = \{ I_2, I_1 * 5/18 \} SP = SP + v_1 * frac = 25 + (24 * (5/18)) = 25 + 6.67 = 31.67 SW = SW + w_1 * frac = 15 + (18 * (5/18)) = 15 + 5 = 20
```

The knapsack is full. Fractional Greedy algorithm selects items  $\{I_2, I_1 * 5/18\}$ , and it gives a profit of **31.67 units**.

#### **Lab Assignment to Complete:**

CODE: # include<stdio.h>

The capacity of the knapsack W = 60 and the list of provided items are shown in the following table –

Item	A	В	С	D
Profit	280	100	120	120
Weight	40	10	20	24

```
void knapsack(int n, float weight[], float profit[], float capacity) {
float x[20], tp = 0;
int i, j, u;
u = capacity;
for (i = 0; i < n; i++)
x[i] = 0.0;
for (i = 0; i < n; i++) {
if (weight[i] > u)
break;
else {
x[i] = 1.0;
tp = tp + profit[i];
u = u - weight[i];
if (i < n)
x[i] = u / weight[i];
tp = tp + (x[i] * profit[i]);
printf("\nThe result vector is:- ");
for (i = 0; i < n; i++)
printf("%f\t", x[i]);
printf("\nMaximum profit is:- %f", tp);
int main() {
float weight[20], profit[20], capacity;
int num, i, j;
float ratio[20], temp;
printf("60009210033 - Jhanvi Parekh");
printf("\nEnter the no. of objects:- ");
scanf("%d", &num);
printf("\nEnter the wts and profits of each object:- ");
```

```
for (i = 0; i < num; i++) {
scanf("%f %f", &weight[i], &profit[i]);
printf("\nEnter the capacity of knapsack:- ");
scanf("%f", &capacity);
for (i = 0; i < num; i++) {
ratio[i] = profit[i] / weight[i];
}
for (i = 0; i < num; i++) {
for (j = i + 1; j < num; j++) {
if (ratio[i] < ratio[j]) {
temp = ratio[j];
ratio[j] = ratio[i];
ratio[i] = temp;
temp = weight[j];
weight[j] = weight[i];
weight[i] = temp;
temp = profit[j];
profit[j] = profit[i];
profit[i] = temp;
knapsack(num, weight, profit, capacity);
return(0);
}
```

```
main.c
                                                                      -<u>;</u>o:-
                                                                             Rur
1 # include<stdio.h>
 2 void knapsack(int n, float weight[], float profit[], float capacity) {
3 float x[20], tp = 0;
4 int i, j, u;
5 u = capacity;
 6 for (i = 0; i < n; i++)
 7 x[i] = 0.0;
8 for (i = 0; i < n; i++) {
9 if (weight[i] > u)
10 break;
11 else {
12 x[i] = 1.0;
13 tp = tp + profit[i];
14  u = u - weight[i];
15 }
16 }
17 if (i < n)
18 x[i] = u / weight[i];
19 tp = tp + (x[i] * profit[i]);
20 printf("\nThe result vector is:- ");
21 for (i = 0; i < n; i++)
22 printf("%f\t", x[i]);
23 printf("\nMaximum profit is:- %f", tp);
24 }
25 int main() {
```

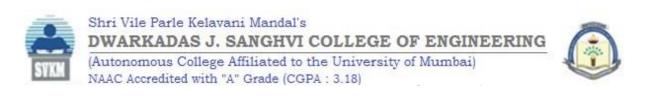
# DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING



(Autonomous College Affiliated to the University of Mumbai) NAAC Accredited with "A" Grade (CGPA: 3.18)

```
main.c
                                                                      -<u>;</u>o-
                                                                             Run
44
25 int main() {
26 float weight[20], profit[20], capacity;
27 int num, i, j;
28 float ratio[20], temp;
29 printf("60009210033 - Jhanvi Parekh");
30 printf("\nEnter the no. of objects:- ");
31 scanf("%d", &num);
32 printf("\nEnter the wts and profits of each object:- ");
33 for (i = 0; i < num; i++) {
34 scanf("%f %f", &weight[i], &profit[i]);
35 }
36 printf("\nEnter the capacity of knapsack:- ");
37 scanf("%f", &capacity);
38 for (i = 0; i < num; i++) {
39 ratio[i] = profit[i] / weight[i];
40 }
41 for (i = 0; i < num; i++) {
42 for (j = i + 1; j < num; j++) {
43 if (ratio[i] < ratio[j]) {
44
45 temp = ratio[j];
46 ratio[j] = ratio[i];
47 ratio[i] = temp;
48 temp = weight[j];
49 weight[j] = weight[i];
```

```
-<u>;</u>ċ-
                                                                                  Run
main.c
   Scalli ( 101 101 , AMCIBILLI), API DIILLI),
35
   }
36 printf("\nEnter the capacity of knapsack:- ");
37 scanf("%f", &capacity);
38 - \text{ for } (i = 0; i < \text{num}; i++) 
39 ratio[i] = profit[i] / weight[i];
40 }
41 \cdot \text{ for } (i = 0; i < \text{num}; i++) 
42 for (j = i + 1; j < num; j++) {
43 if (ratio[i] < ratio[j]) {
44
45 temp = ratio[j];
46 ratio[j] = ratio[i];
47 ratio[i] = temp;
48 temp = weight[j];
49 weight[j] = weight[i];
50 weight[i] = temp;
51 temp = profit[j];
52 profit[j] = profit[i];
53 profit[i] = temp;
54 }
55 }
56 }
57 knapsack(num, weight, profit, capacity);
58 return(0);
59 }
```



#### Output

#### /tmp/oOPgg7dC3E.o

60009210033 - Jhanvi Parekh Enter the no. of objects: 4

Enter the wts and profits of each object:- 40 280

10 100

20 120

24 120

Enter the capacity of knapsack:- 60

The result vector is:- 1.000000 1.000000 0.500000 0.000000

Maximum profit is:- 440.000000



#### Shri Vile Parle Kelavani Mandal's

## DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING



(Autonomous College Affiliated to the University of Mumbai) NAAC Accredited with "A" Grade (CGPA: 3.18)

60009210	033			Page No.:	
Jhanvi P	arekh			(Date. / /	
ab work			HILLIAN	THE BURNES	-
Item	Profit (	(vi)	weight (	wi) Pi=	Vi
Α	280		40	100 + Cu 107	
В	100		OU TO	10	
C	120		20	6	
D	120		24	3949 5.	
				010 0001 = 9	
Bout the	items o	according	to profit	in descende	no
order		J	1183 11	west bones	
Item	vi	wi	Pi	Duni hills	
Δ	100	10	10	25 - 3	
£	280	40	7	FOR HOUR SOR	
С	120	20	6	105 x 102 3 CW	
D	120	24	5	0-17-2-00	
tue in	itial value	s are	weight (	w) -0, Projet	CF
	M = 60				
select I		*	78 - O.D		
$B \rightarrow 1$		*			
w = w			6.2		
	+10	2 1 13 1			
	4		6, 4 X		
W=10	Pi		100000	( + JE - + Q)	
P = P+1				* 0 0 'e 111 's	
P = P + 1 $P = 0 + 1$ $P = 10$	0 0			0x17-7 17	
P = P + 1 $P = 0 + 1$ $P = 10$	100	£B G	(2-0×a)	1) + 1/2 · 3	
P = P + 1 $P = 0 + 1$ $P = 10$	0 0	£вζ	(2, g×90	in management	
P = P + 1 $P = 0 + 1$ $P = 10$ selected	0 0	EB G	(7 0×00	in the state of th	

# Shri Vile Parle Kelavani Mandal's

## DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING



(Autonomous College Affiliated to the University of Mumbai) NAAC Accredited with "A" Grade (CGPA: 3.18)

Department of Computer Science and	Engineering (Data Ser	ience)
	Face No:	
selected item A	2,050	1
A -> H	Civi spie mas	
w = w +w;	SAE A	
W = 80+20 10+40	8	
W = 50	001-101	
P = P + Pi	1 041 1 0 0 0	
P = 100 + 280		
P = 380	many many and a	
selected item = {BIAS	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	100
Edect item c	10 - 10 ms	r
C→ 20	1 ' 00) A	
$\omega = \omega + \omega = \omega$	U . 58.5 2	
w = 50 + 20	2 0.51	
w = 70	0 001 1	
The weight of Item c exc	ceeds the capacity	of b
breakdown the items using	practional thapsack	tee
fraction = M-W	1. 1.3 = mil (1991)	102
usi ·	03-M	
= 60-50	ह प्राची करा	10
20.	01-8	
= 0.5	(01) + (1)	15
.: Selected item = { B, A	, 0.5 x c }	0
w= w+wixos	01.20	
w = 50 + 20 x 0.5	19490	9
w = 60	001+02	9
P= P+Pixo-s		7
P = 380 + (120 x 0.5)	283 = Will begin	
P = 440 units		
P = 440 units  She capacity of phapeack	is full therepore s	silecte
	is full therefore s 3 and gives a prop	silecte it of