Varun Khadayate
A016

25-10-2021

0/1 Knapsack Problem

Code:

```
#include<stdio.h>
#include<conio.h>
#define MAX 20
float final_profit;
int w[MAX];
int p[MAX];
int n,m;
int temp[MAX],x[MAX];
float final_wt;

float Bound_Calculation(int,int,int);
void BackTracking(int,int,int);

int main()
{
   int i;
   printf("\tKNAPSACK PROBLEM USING BACKTRACKING");
   printf("\n Enter number of Objects you want:");
   scanf("%d",&n);
   for(i=1;i<=n;i++)
   {
      printf("\n Enter Weight and value for object%d:",i);
      scanf("%3d %3d",&w[i],&p[i]);
   }
   printf("\n Enter Capacity of Knapsack:");
   scanf("%d",&m);
   getch();
```

```
    printf("\n Weight\tProfit");

    for(i=1;i<=n;i++)
    {
        printf("\n %d \t %d",w[i],p[i]);
    }
    BackTracking(1,0,0);
    printf("\n Following Objects are included:");
    for(i=1;i<=n;i++)
    {
        if(x[i]==1)
            printf("\n%d",i);
    }
    printf("\n Final Weight:%0.2f",final_wt);
    printf("\n Final Profit:%0.2f",final_profit);
    return 0;
}

float Bound_Calculation(int cp,int cw,int k)
{
    int ub,c,i;
    ub=cp;
    c=cw;
    for(i=k+1;i<=n;i++)
    {
        c=c+w[i];
        if(c < m)
            ub=ub+p[i];
        else
            return (ub+(1-(c-m)/w[i])*p[i]);
    }
```

```
    return ub;
}

void BackTracking(int k,int cp,int cw)
{
    int new_k,new_cp,new_cw,j;
    if(cw+w[k]<=m)
    {
        temp[k]=1;
        if(k < n)
        {
            new_k=k+1;
            new_cp=cp+p[k];
            new_cw=cw+w[k];
            BackTracking(new_k,new_cp,new_cw);
        }
        if((new_cp>final_profit)&&(k==n))
        {
            final_profit=new_cp;
            final_wt=new_cw;
            for(j=1;j<=k;j++)
            {
             x[j]=temp[j];
            }
        }
    }
    if(Bound_Calculation(cp,cw,k)>=final_profit)
    {
        temp[k]=0;
        if(k < n)
            BackTracking(k+1,cp,cw);
```

```
    if((cp>final_profit)&&(k<=n))
    {
       final_profit=cp;
       final_wt=cw;
       for(j=1;j<=n;j++)
          x[j]=temp[j];
    }
  }
}
```

Output:

```
E:\TY\DAA\PRAC_4_Knapsack_0-1_Approach.exe
 Enter number of Objects you want:4

 Enter Weight and value for object1:2
4

 Enter Weight and value for object2:3
6

 Enter Weight and value for object3:4
8

 Enter Weight and value for object4:1
5

 Enter Capacity of Knapsack:20

 Weight Profit
 2       4
 3       6
 4       8
 1       5
 Following Objects are included:
1
2
3
 Final Weight:9.00
 Final Profit:18.00
Process returned 0 (0x0)   execution time : 11.972 s
Press any key to continue.
```