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:

Text

Description automatically generated