Vellore-632014, Tamil Nadu, India

School of Mechanical Engineering

MEE437 Operations Research – PBL – Slot E1

Faculty: Siva Prasad Darla

SOLVING TRANSPORTATION PROBLEM USING THE BEST CANDIDATES METHOD AND ITS C PROGRAM

BY
Shivansh Jagga
13BCE0188

A project report submitted as a part of internal assessment of MEE437 Operations Research Course

Fall Semester 2016-17

3RD NOVEMBER 2016

CONTENT

ABSTRACT

The optimization processes in mathematics, computer science and economics are solving effectively by choosing the best element from set of available alternatives elements.

Most of the currently used methods for solving transportation problems are trying to reach the optimal solution, whereby, most of these methods are considerd complex and very expansive in term of the execution time.In this study we use the best candidate method (BCM), in which the key idea is to minimize the combinations of the solution by choosing the best candidates to reach the optimal solution.

Comparatively, applying the BCM in the proposed method obtains the best initial feasible solution to a transportation problem and performs faster than the existing methods with a minimal computation time and less complexity. The proposed methods is therefore an attractive alternative to traditional problem solution methods.

1. INTRODUCTION

The optimization processes in mathematics, computer science and economics are solving effectively by choosing the best element from set of available alternatives elements. The most important and successful applications in the optimaization refers to transportation problem (TP), that is a special class of the linear programming (LP) in the operation research (OR).

Approach: The main objective of transportation problem solution methods is to minimize the cost or the time of transportation. Most of the currently used methods for solving transportation problems are trying to reach the optimal solution, whereby, most of these methods are considerd complex and very expansive in term of the execution time.

In this study we use the best candidate method (BCM), in which the key idea is to minimize the combinations of the solution by choosing the best candidates to reach the optimal solution.

Conclusion/Recommendations: The BCM can be used successfully to solve different business problems of distrbution products that is commonly referd to a transportation problems.

2. PROBLEM DESCRIPTION

Most of the currently used methods for solving transportation problems are trying to reach the optimal solution, whereby, most of these methods are considerd complex and very expansive in term of the execution time.

In this study we use the best candidate method (BCM), in which the key idea is to minimize the combinations of the solution by choosing the best candidates to reach the optimal solution. Comparatively, applying the BCM in the proposed method obtains the best initial feasible solution to a transportation problem and performs faster than the existing methods with a minimal computation time and less complexity. The proposed methods is therefore an attractive alternative to traditional problem solution methods.

3. METHODOLOGY

Transportation model: In a transportation problem, we are focusing on the original points. These points may represent factories to produce items, and to supply a required quantity of the products to a certain number of destinations. This process must be done successfully in such a way as to maximize the profit or minimize the cost transportation.

Therefore, the places of production and supply are collected as the original points and the destinations respectively. Sometimes the original and destinations points are also termed as sources and sinks. However, to illustrate a typical transportation model, suppose that m factories supply certain items to n warehouses. As well as, let factory i (i = 1, 2, …, m) produces a_i units, and the warehouse j (j = 1, 2, …, n) requires b_j units.

Furthermore, suppose the cost of transportation from factory i to warehouse j is c_{ij}. The decision variables xij is being the transported amount from the factory i to the warehouse j. Typically, our objective is to find the transportation pattern that will minimize the total of the transportation cost (see Table 1).

Table 1: The model of a transportation problem

Origins (Factories)	**Destinations (Warehouses)**		**Available**	
	1 2 …….. n			
1	c_{11}	c_{12}	c_{1n}	a_1
2	c_{21}	c_{22}	c_{2n}	a_2
…	…	…	…	…
m	c_{m1}	c_{m2}	c_{mn}	a_m
Required	b_1	b_2	b_n	

4. SOLUTION PROCEDURE

4.1 Choosing best combination using BCM

- Step1: Prepare the matrix. If the matrix unbalanced, we balance it and don't use the added row or column candidates in our solution procedure.
- Step2: Election the best candidates, that is for minimization problems minimum cost and for maximize profit max cost: Elect the best two candidates in each row, if the candidate repeated more than two times elect it also. Check the columns that not have candidates and elect one candidate for them, if the candidate repeated more than one time elect it also
- Step3: Find the combinations. Determine only one candidate for each row and column starting from the row that have least candidates and delete that row and column If there is situation that have no candidate for some rows or columns elect directly the best available candidate. Repeat step3 (1, 2) by determining the next candidate in the row that started from. Compute and compare the total sum of candidates for each combination to determine the best combination that give the optimal solution

Example of Using BCM : Consider the problem of assigning five jobs to five persons. The assignment costs are given as follows.

Table 1: Matrix

	1	2	3	4	5
A	8	4	2	6	1
B	0	9	5	5	4
C	3	8	9	2	6
D	4	3	1	0	3
E	9	5	8	9	5

Table 2: Elect the best candidates

	1	2	3	4	5
A	8	4	2	6	1
B	0	9	5	5	4
C	3	8	9	2	6
D	4	3	1	0	3
E	9	5	8	9	5

Table 3: Find the combinations 1

	1	2	3	4	5
A	8	4	2	6	1
B	0	9	5	5	4
C	3	8	9	2	6
D	4	3	1	0	3
E	9	5	8	9	5

Table 4: Find the combinations2

	1	2	3	4	5
A	8	4	2	6	1
B	0	9	5	5	4
C	3	8	9	2	6
D	4	3	1	0	3
E	9	5	8	9	5

4.2 Solving Transportation Problem using the Proposed Method

- Step1: We must check the matrix balance, If the total supply is equal to the total demand, then the matrix is balanced and also apply
- Step2: If the total supply is not equal to the total demand, then we add a dummy row or column as needed to make supply is equal to the demand. So the transportation costs in this row or column will be assigned to zero.
- Step3: Appling BCM to determine the best combination that is to produce the lowest total weight of the costs, where is one candidate for each row and column.
- Step4: Identify the row with the smallest cost candidate from the choosen combination. Then allocate the demand and the supply as much as possible to the variable with the least unit cost in the selected row or column. Also, we should adjust the supply and demand by crossing out the row/column to be then assigned to zero. If the row or column is not assigned to zero, then we check the selected row if it has an elemnt with lowest cost comparing to the determined element in the chossen combination, then we elect it.
- Step5**:** Elect the next least cost from the choosen combination and repeat Step 3 untill all columns and rows is exhusted.

4.3 Solving the problem using the method

Finding the optimal cost for the problem of assigning five jobs to five persons.

Plant	D1	D2	D3	D4	Supply	
S1	10	30	25	15	14	
S2	20	15	20	10	10	
S3	10	30	20	20	15	
S4	30	40	35	45	12	
Demand	10	15	12	15	***52***	***51***

Step 1: Use BCM to get candidates

Plant	D1	D2	D3	D4	Supply	
S1	10	30	25	15	14	
S2	20	15	20	10	10	
S3	10	30	20	20	15	
S4	30	40	35	45	12	
S5	0	0	0	0	1	
Demand	10	15	12	15	***52***	***52***

Step 2: Choose smallest candidate first

Plant	D1	D2	D3	D4	Supply	
S1	10	30	25	15	14	
S2	20	15	20	10	10	
S3	10 **10**	30	20	20	~~15~~ 5	
S4	30	40	35	45	12	
S5	0	0	0	0	1	
Demand	~~10~~	15	12	15	***52***	***52***

0

Step 3: Follow 3 of pseudo algo and repeat

Plant	D1	D2	D3	D4	Supply	
S1	10	30	25	15	14	
S2	20	15	20	10	10	
S3	10 **10**	30	20 **5**	20	~~15~~ ~~5~~ 0	
S4	30	40	35	45	12	
S5	0	0	0	0	1	
Demand	~~10~~	15	~~12~~	15	***52***	***52***

0 7

Step 4: Do this till demand and supply =0

Plant	D1	D2	D3	D4	Supply	
S1	10	30	25	15 **14**	~~14~~ 0	
S2	20	15 **10**	20	10	~~10~~ 0	
S3	10 **10**	30	20 **5**	20	~~15~~ ~~5~~ 0	
S4	30	40 **5**	35 **7**	45	~~12~~ 0	
S5	0	0	0	0 **1**	~~1~~ 0	
Demand	~~10~~	~~15~~	~~12~~	~~15~~	***52***	***52***

0 5 ~~7~~ ~~1~~

0 0 0

Step 5: Result that we get using proposed method

15*14 + 15*10 +10*10 + 20*5 + 40*5 + 35*7 = 1005

5. TESTING/ PROGRAMMING USING DEV C++

5.1 Program Code

```
#include<iostream>
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
using namespace std;
#define  F(i,a,b) for(int i = (int)(a) ; i<(int)(b); i++)


int c[20][20],sup[20],dem[20];        //For input

 int m,n; //Input of number of rows and columns
 int M,N;

void sort(int a[],int n)
{ int temp,j,k;
  for(j=0;j<n;j++)
  { for(k=j+1;k<n;k++)
    { if(a[j]>a[k])
      { temp=a[j];
            a[j]=a[k];
            a[k]=temp;
      }
    }
  }
}
//////////////////////////////////////////////////////////////////////////////////////////
/////Supporting function for BCM

int path[4][4]={0}; //For visited places or the places we can't tread
int best_sum=10000 , best_s[4][4]; //Best sum for each combination like f(0,0) or f(0,1) or f(0,2) .. is stored
                                             //best_s saves path matrix for the current best_sum of combination

void func(int g, int h, int sum)
{
   if(g>4 || g<0 ||h>4 ||h<0)
    return ;
   if(path[g][h]==1||path[g][h]==2)
   return ;

   sum+=c[g][h];


   F(j,0,n)
    path[g][j]=1;
   F(i,0,m)
            path[i][h]=1;

            path[g][h]=2;

            if(g==m-1)
```

```
  {
                        if(best_sum > sum)  //Saving this best_sum if better than sums of all combinations under the combos under
func(0,0,sum)
                        {
                                    best_sum=sum;
                                    //Saving path as best path found till now
                                     F(k,0,4)
                                     F(j,0,4)
                                                best_s[k][j]=path[k][j];
                        }
  }
            //Recursive function selecting all columns for every next row
            F(i,1,n)
            {
            func(g+1,(h+i)%4 , sum);
            }

            //Undo changes
            F(j,0,n)
            path[g][j]=0;
             F(i,0,m)
            path[i][h]=0;

            F(k,0,4)
            F(j,0,4)
            { if(path[k][j]==2)
                        {
                           F(i,0,n)
                                    path[k][i]=1;
                           F(i,0,m)
                                    path[i][j]=1;

                                    path[k][j]=2;
                        }
            }

            return;
}

/////////////////////////////////////////////////////////////
//////////////////////Function for getting best candidate

int bcm()
{
            int best=10000, best_pos; //Best records best of all best_sums. and best_pos stores the position of that best starting
point
F(i,0,n)
{
//Re-Making path matrix as 0.
            F(k,0,m)
            F(j,0,n)
            path[k][j]=0;
            best_sum=10000 ;
            func(0,i,0);
            if(best_sum < best)
            {   best=best_sum;
```

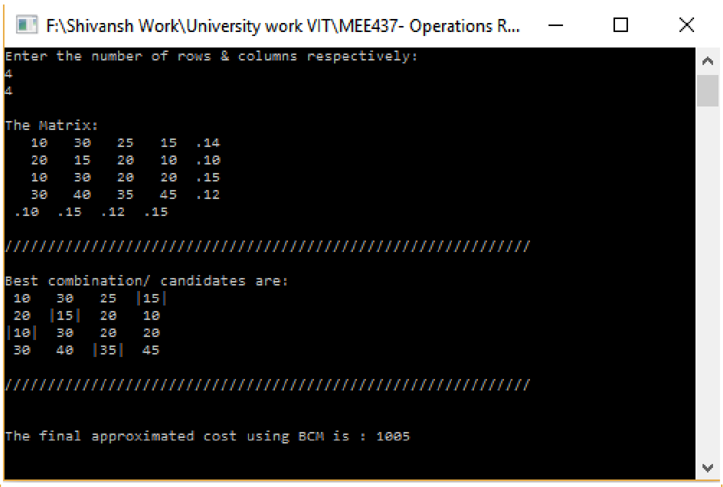
```
                        best_pos=i;
            }
}
//To get the final path matrix of the best sum
func(0,best_pos,0);

cout<<"\nBest combination/ candidates are: \n";
F(k,0,M)
            {
            F(j,0,N)
            {           if(best_s[k][j]==2)
                  cout<<"|"<<c[k][j]<<"|"<<" ";
               else
                        cout<<" "<<c[k][j]<<"  ";
            }
            cout<<endl;
}
return 0;
}
/////////////////////////////////////////////////////////////////////
///////////////////////////

///////////////////////////////////////////MAIN function & TRANSPORTATION PROBLEM
int main()
{ int i,j,b,p,d,k;

  int cf[20],rf[20],a[20],cp[20],rp[20];

  int max,min,s,t,sum=0;
  system("cls");
  printf(".............Transportation problem Using Best Candidates Method........\n");
  printf("\nEnter the number of rows & columns respectively:\n");
  scanf("%d%d",&m,&n);

  //For balancing the matrix to perform BCM
  if(m<n)
            {   M=m;
               N=n;
                        F(i,m,n)
                        F(j,0,n)
                                    c[i][j]=0;
                        m=n;
            }
            else
            {    M=m;
               N=n;
                        F(i,0,m)
                        F(j,n,m)
                                    c[i][j]=0;
                        n=m;
            }

//For INPUT
int c[20][20],dem[20],sup[20];

 printf("\nEnter the cost:");
```

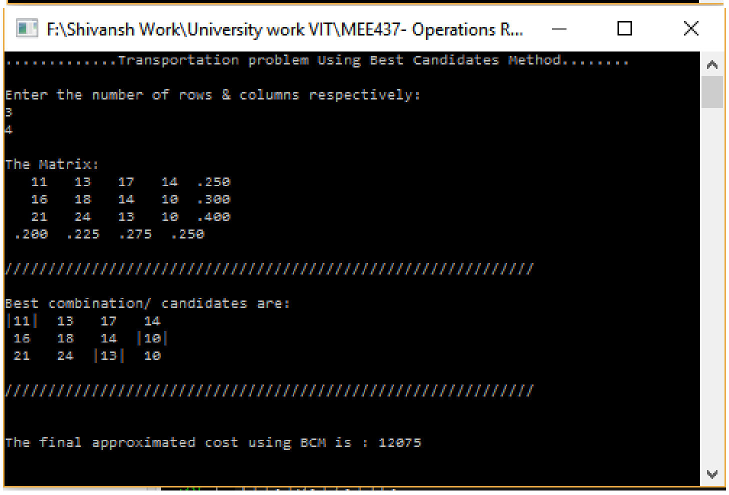
```
 for(i=0;i<m;i++)
 { for(j=0;j<n;j++)
    scanf("%d",&c[i][j]);
 }
 printf("\nEnter the demand:");
 for(i=0;i<n;i++)
  scanf("%d",&dem[i]);
 printf("\nEnter the supply:");
 for(i=0;i<m;i++)
  scanf("%d",&sup[i]);


 printf("\nThe Matrix:\n");
 for(i=0;i<M;i++)
 { for(j=0;j<N;j++)
    printf("   %d",c[i][j]);

   printf("  .%d",sup[i]);
   printf("\n");
 }
 for(j=0;j<n;j++)
  printf(" .%d ",dem[j]);

  cout<<"\n\n//////////////////////////////////////////////////////////// \n";
            //BCM
            bcm();
cout<<"\n//////////////////////////////////////////////////////////// \n";
 for(i=0;i<m;i++)
  rf[i]=0;
 for(i=0;i<n;i++)
  cf[i]=0;
 b=m,d=n;
 while(b>0&&d>0)
 { for(i=0;i<m;i++)
    rp[i]=-1;
   for(i=0;i<n;i++)
    cp[i]=-1;
   for(i=0;i<m;i++)
   { k=0;
     if(rf[i]!=1)
     { for(j=0;j<n;j++)
            { if(cf[j]!=1)
                a[k++]=c[i][j];
            }
            if(k==1)
             rp[i]=a[0];
            else
            { sort(a,k);
              rp[i]=a[1]-a[0];
            }
     }
   }
   for(i=0;i<n;i++)
   { k=0;
     if(cf[i]!=1)
     { for(j=0;j<m;j++)
```

```
          { if(rf[j]!=1)
             a[k++]=c[j][i];
          }
    if(k==1)
           cp[i]=a[0];
          else
          { sort(a,k);
            cp[i]=a[1]-a[0];
          }
  }
}
for(i=0;i<m;i++)
 a[i]=rp[i];
for(j=0;j<n;j++)
 a[i+j]=cp[j];
max=a[0];
p=0;
for(i=1;i<m+n;i++)
{ if(max<a[i])
  {       max=a[i];
          p=i;
  }
}

min=1000;
if(p>m-1)
{ p=p-m;
  if(cf[p]!=1)
  { for(i=0;i<m;i++)
          { if(rf[i]!=1)
            { if(min>c[i][p])
              { min=c[i][p];
                s=i;
                t=p;
              }
            }
          }
  }
}
else
{ if(rf[p]!=1)
  { for(i=0;i<n;i++)
          { if(cf[i]!=1)
            { if(min>c[p][i])
              { min=c[p][i];
                s=p;
                t=i;
              }
            }
          }
  }
}

if(sup[s]<dem[t])
{ sum+=c[s][t]*sup[s];
  dem[t]-=sup[s];
```

```
      rf[s]=1;
      b--;
    }
    else
    if(sup[s]>dem[t])
    { sum+=c[s][t]*dem[t];
      sup[s]-=dem[t];
      cf[t]=1;
      d--;
    }

    else
    if(sup[s]==dem[t])
    { sum+=c[s][t]*dem[t];
      cf[t]=1;
      rf[s]=1;
      b--;
      d--;
    }
  }
  printf("\n\nThe final approximated cost using BCM is : %d \n\n",sum);

  return 0;
}
```

5.2 Output





6. RESULTS AND DISCUSSION/ CONCLUSION

Actually, using BCM based on the best candidates election is to minimize the number of combinations solutions. As we can obtain the combinations without any intersect means, such that; one candidate for each row and column. Eficentaly, BCM works to obain the optimal solution or the closest to optimal solution than other available methods. Since it has a better starting solution.

In this study, we have proposed a BCM for solving transportation problems, because of its wide applicability in different area. Whereby, it refers to choose the best distrubtion of cost or time. The BCM obtained the optimal solution or the closest to optimal solution with a minimum computation time. As well as, using BCM will reduces the complexity with a simple and a clear solution manner which is can be easily used on different area for optimization problems.

7. REFERENCES

[1] Elaine L.Y. Man and Adam Baharum , 2011. A Qualitative Approach of Identifying Major Cost Influencing Factors in Palm Oil Mills and the Relations towards Production Cost of Crude Palm Oil. American Journal of Applied Sciences, 8: 441-446. DOI: 10.3844/ajassp.2011.441.446.

[2] Sudha, S.V. and K. Thanushkodi, 2012. A genetic based neuro fuzzy technique for process grain sized scheduling of parallel jobs. J. Comput. Sci., 8:48-54. DOI: 10.3844/jcssp.2012.48.54

[3] Amir, S., H.Z. Aashtiani and K.A. Mohammadian, 2009. A Shor-term Management strategy for Improving transit network efficiency. Am. J. Applied Sci., 6: 241-246. DOI: 10.3844/ajassp.2009.241.246.

[4] Zuhaimy Ismail , Diyana A. Mahad and Tan S. Ching , 2011. Modeling of Multi-Level Capacitated Lot-Size Scheduling Problem. American Journal of Applied Sciences, 8: 290-296. DOI: 10.3844/ajassp.2011.290.296.