Available online at www.elixirpublishers.com (Elixir International Journal)

Applied Mathematics

Elixir Appl. Math. 72 (2014) 25583-25585

ISSN: 2229-712X

A study of North east corner method and use of Object Oriented Programming model

R. Palaniyappa[1] and V. Vinoba[2]

[1]Department of Mathematics, IRT Polytechnic College, Chromepet, Chennai – 44, India.

[2]Department of Mathematics, K. N. G. C. (W), Thanjavur – 7, India.

ARTICLE INFO

Article history:

Received: 26 May 2014;

Received in revised form: 19 June 2014;

Accepted: 4 July 2014;

Keywords

Transportation problem,

LPP, Optimal solution,

North east corner rule,

Object oriented programming.

ABSTRACT

In this paper, the North east corner [NEM] procedure is successfully coded and tested via many randomly generated problem instances . Based on the results we can conclude that the correctness of the newly coded NEM is promising as compared with the previously coded one.

Introduction

The term 'OR' was coined in 1940 by M. C. Closky & T.ref then in a small town of Bawdsey in England. It is a science that came into existence in a military content. During world war II, the military management of UK called an Scientists from various disciplines & organized them into teams to assist it in solving strategic & tactical problems relating to air & land defence of the country.

The transportation problem is a special class of LPP that deals with shipping a product from multiple origins to multiple destinations. The objective of the transportation problem is to find a feasible way of transporting the shipments to meet demand of each destination that minimizes the total transportation cost while satisfying the supply & demand constraints. The two basic steps of the transportation method are

Step 1: Determine the initial basic feasible solution

Step 2: Obtain the optimal solution using the solution obtained from step 1.

In this paper the corrected coding of NEM in C++ is implemented. Then its correctness is verified via many randomly generated instances. The remainder of this paper is organized as follows :

Section II deals with the mathematical formulation of the transportation problem. In section III NEM is summarized. In section IV potential significance of the new object oriented program of VAM is illustrated with a numerical example.

Finally, conclusion by highlighting the limitations and future research scope on the topic is made in section V.

Mathematical formulation of the transportation problem

A. In developing the LP model of the transportation problem the following notations are used

a_i - Amounts to be shipped from shipping origin i ($a_i \geq 0$.

b_j - Amounts to be received at destination j ($b_j \geq 0$).

c_{ij} - Shipping cost per unit from origin i to destination j ($c_{ij} \geq 0$).

x_{ij} - Amounts to be shipped from origin i to destination j to minimize the total cost ($x_{ij} \geq 0$) .

We assumed that the total amount shipped is equal to the total amount received, that is,

$$\sum_{i=1}^{m} a_i \geq \sum_{i=1}^{m} b_j$$

B. Transportation problem

$$\text{Min } \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

Subject to $\sum_{j=1}^{n} x_{ij} \leq a_i$, i = 1 , 2, … , m

$\sum_{i=1}^{m} x_{ij} \leq b_j$, j = 1 , 2 , … , n , where $x_{ij} \geq 0 \ \forall$ i , j .

Feasible solution : A set of non negative values x_{ij} , i = 1 , 2 , …, n and j = 1 , 2 , …, m that satisfies the constraints is called a feasible solution to the transportation problem .

Optimal solution : A feasible solution is said to be optimal if it minimizes the total transportation cost .

Non degenerate basic feasible solution: A basic feasible solution to a (m × n) transportation problem that contains exactly m + n – 1 allocations in independent positions.

Degenerate basic feasible solution: A basic feasible solution that contains less that m + n – 1 non negative allocations.

Balanced and Unbalanced Transportation problem: A Transportation problem is said to be balanced if the total supply from all sources equals the total demand in the destinations and is called unbalanced otherwise.

Thus, for a balanced problem, $\sum_{i=1}^{m} a_i = \sum_{i=1}^{m} b_j$ and for unbalanced problem, $\sum_{i=1}^{m} a_i \neq \sum_{i=1}^{m} b_j$

North East Corner Rule

Procedure:

North East Corner Method :

➢ i) The method starts at the North – East corner cell (route) of the tableau (Variable X1n).

➢ Allocate as much as possible to the selected cell and adjust the associated amounts of supply and demand by subtracting the allocated amount.

➢ ii) Cross out the row or column with zero supply or demand to indicate that no further assignments can be made in the row or column. If both a row and a column net to zero simultaneously cross out one only and leave a zero supply (demand in the uncrossed out row or column).

Tele:

E-mail addresses: shivaram2003@gmail.com

- If exactly one row or column is left uncrossed out or below if exactly one row or column is left uncrossed out, stop. Otherwise, move to the cell to the right if a column has just been crossed out or below if a row has been crossed out. Go to step (i).
- Start with X1n and end must be Xm1.

Object oriented program code for North east corner method for initial basic feasible solution:

```
 ****************************************************
     NORTH EAST CORNER
****************************************************/
#include<stdio.h>
#include<conio.h>
void main()
{
   int sn,dn,i,j,ss=0,ds=0,sum=0;
   int sup[10],dem[10];
   int a[10][10],c[10][10];
   clrscr();
   //getting no.of supply & demand.
   printf("Enter the num of supply:");
   scanf("%d",&sn);
   printf("Enter the num of demand:");
   scanf("%d",&dn);
   //clearing values in array.
   for(i=0,j=0;i<10,j<10;i++,j++)
     {
       sup[i]=0;
       dem[j]=0;
     }
   //input supply values & calculate sum of supply.
   for(i=0;i<sn;i++)
   {
       printf("\n Enter supply value sup[%d]:",i);
       scanf("%d",&sup[i]);
       ss=ss+sup[i];
   }
   //input demand values & calculate sum of demand.
   for(j=0;j<dn;j++)
   {
       printf("\n Enter demand value dem[%d]:",j);
       scanf("%d",&dem[j]);
       ds=ds+dem[j];
   }
   if(ss!=ds)

{
      printf("\n unbalanced problem..");
      getch();
      exit(0);
  }
   //input transportation cost.
   printf("\n Enter array values:");
   for(i=0;i<sn;i++)
   {
       for(j=0;j<dn;j++)
       {
            scanf("%d",&a[i][j]);
       }
   }
   //clearing cost array.
   for(i=0;i<10;i++)
   {
       for(j=0;j<10;j++)
       {
            c[i][j]=0;
       }
   }
   //calculation.
   for(i=0,j=dn-1;i<sn,j>=0;)
     {
       if(i==sn||j==-1)
            goto L1;
       if(dem[j]==sup[i])  //Checking demand=supply
        {
          c[i][j]=dem[j]*a[i][j];
          printf("\n c[%d][%d]=%d and sup=%d",i,j,c[i][j],sup[i]);
          j--;
          i++;
        }
       else if(dem[j]<sup[i])  //checking demand< supply
        {
          c[i][j]=dem[j]*a[i][j];
          sup[i]=sup[i]-dem[j];
          printf("\n c[%d][%d]=%d and sup=%d",i,j,c[i][j],sup[i]);
          dem[j]=0;
          j--;
        }
       else   //demand >= supply
       {
         c[i][j]=sup[i]*a[i][j];
         dem[j]=dem[j]-sup[i];
         printf("\n                c[%d][%d]=%d                and
dem=%d",i,j,c[i][j],dem[j]);
         i++;
       }
     }

  L1:
  for(i=0;i<sn;i++)
       for(j=0;j<dn;j++)
            sum=sum+c[i][j];
   printf("\n\n sum of transportation cost = %d",sum);
   getch();
}
```

Example :

Solve the transportation problem

5	8	6	6	3	800
4	7	7	6	5	500
8	4	6	6	4	900
Demand 400	400	50000	400	800	

Proof :

The given Problem unbalanced transportation problem. Because demand # supply i.e. 2,200 # 2,500 (2500-2200=300) so we add a dummy row all the entries are zero and put the supply is 300.

5	8	6	6	3	800
4	7	7	6	5	500
8	4	6	6	4	900
0	0	0	0	0	300
400	400	500	400	800	

By using north East Method

5	8	6	6	800 3	800
4	7	100 7	400 6	5	500/100
100 8	400 4	400 6	6	4	900/500/100
300 0	0	0	0	0	300
400/300	400	500/400	400	800	

The Initial basic feasible solution is given as
X15 =800, X23 = 100, X24 = 400, X31 = 100,
X32 = 400, X33 = 400, X41 = 300
The Objective function is
Min Z = 800x3 + 100x7+400x6+100x8+400x4+400x6+300x0 = $ 10,300

Output :

```
Enter the num of supply:4
Enter the num of demand:5
Enter supply value sup[0]:800
Enter supply value sup[1]:500
Enter supply value sup[2]:900
Enter supply value sup[3]:300
Enter demand value dem[0]:400
Enter demand value dem[1]:400
Enter demand value dem[2]:500
Enter demand value dem[3]:400
Enter demand value dem[4]:800
Enter array values:5 8 6 6 3
                   4 7 7 6 5
                   8 4 6 6 4
                   0 0 0 0 0

c[0][4]=2400 and sup=800
c[1][3]=2400 and sup=100
c[1][2]=700 and dem=400
c[2][2]=2400 and sup=500
c[2][1]=1600 and sup=100
c[2][0]=800 and dem=300
c[3][0]=0 and sup=300
sum of transportation cost = 10300
```

Conclusion

The optimal solution obtained in this present investigation shows much more closeness with initial basic feasible solution obtained by North east corner rule. The comparison of optimal solution have been made with other methods of finding initial solutions and observe that North east corner method give the better initial feasible solutions which are closer to optimal solution. The objected oriented programs using c++ have been developed . This shows that the computed results tally with the results obtained c++ programming. **.** Object oriented program code for said programs is given for better understanding.

References

[1] Reeb, J.E. and S.A., Leavengood, “Transportation problem: a special case for linear programming problems”, EM8779. Corvallis: Oregon State University Extension Service, pp. 1-35, 2002.

[2] Charnes, A. and W.W. Cooper, "The stepping stone method of explaining linear programming calculations in transportation problems", Management Science, **1**(1): pp. 49-69, 1954.

[3] Dantzig, G.B., “Linear Programming and extensions”, Princeton, NJ: Princeton University press, 1963.

[4] Taha Hamdy A., “Operation Research: An introduction”, Prentice-Hall of India, 8th edition. 2006.

[5] Reinfeld, N.V and Vogel, W.R., “Mathematical Programming”, Englewood Cliffs, New Jersey: prentice-Hall, pp. 59-70, 1958.

[6] Nabendu Sen et al., “A study of transportation problem for an essential item of southern part of north eastern region of India as an OR model and use of object oriented programming”, International Journal of Computer Science and Network Security, 10(4), pp. 78-86, 2010.

[7] Saleem, Z.R. and Imad, Z.R., “Hybrid two-stage algorithm for solving transportation problem”, Modern Applied Science, 6(4), pp. 12-22, 2012.

[8] R. Palaniyappa and V. Vinoba , “A new type of transportation problem using object oriented model “ , International Journal of Mathematical Archive-4(11), 2013, 71-77