



Two-Stage Multi Objective Variant Bulk TP Model

Guravaraju Pokala^{*1}, Purusotham.S², Suresh Babu C³, Madhu Mohan Reddy P⁴

¹ Research Scholar, Dept of Mathematics, S. V. University, Tirupati, Andhra Pradesh, India

² Asst. Professor, Dept of Mathematics, Vellore Institute of Technology (VIT), Vellore, Tamil Nadu, India

^{3,4} Asst. Professor, Dept of Mathematics, SIETK, Puttur, Andhra Pradesh, India

Abstract: In recent years so many researchers have studied different models in transportation problem. In this problem the commodities can be transported to a particular destination through some transient points. Here the place of manufacturing is called Main source. The transient points are called Sources and the various points receiving the objects from sources are called Destinations. The transportation and supply of Main sources and Sources should be subject to their capacities. Each destination should get its entire requirement from one source.

In this problem “Two-Stage Multi Objective Variant Bulk TP Model” there are p main sources, m sources and n destinations. Out of p main sources p_0 , m_0 sources from m sources and n_0 destinations from n destinations are selected. Here the process of transportation is in two stages. In first stage the transportation of a product from main sources to sources and in second stage the product supplied from sources to destinations. In the first schedule we consider the transportation as distance and in second schedule we consider the supply as cost. Distance and cost are two independent factors(units).

The objective of the problem is to minimize both the total distance in first schedule and total cost in second schedule subjected to the constraints. The problem comes under combinatorial programming problem. We solve this problem by Lexi-search Algorithm using pattern recognition Technique.

Key words: Transportation, Main sources, Sources, Destinations, combinatorial programming, Lexi-search Algorithm, Pattern Recognition technique.

I. INTRODUCTION

In general the transportation of commodities from sources to destinations is known as Transportation Problem. But instead of direct transportation to destinations, the commodities/objects can be transported to a particular destination through some transient points called sources. This problem is called Transportation Problem with some transient Nodes between Main sources and Destinations or Two-Stage Transportation Problem (PanneerSelvam).

In recent years so many researchers studied different models in Transportation Problem. The product is manufactured at different industries/places called as main-sources. From the main-sources, the product is transported to various destinations through some transient points called sources. The various points receiving the objects from sources are called destinations. Thus the process of transportation of a product from main sources to destination is done in two stages.

The first stage involves the transportation of the product from main-sources to the sources and the second stage involves the supply of product from sources to destinations. In this problem from main sources to sources we consider distances and from sources to destinations we consider costs. Here distances and costs are two independent factors. The conditions of transportation are

- (i) The main source is supplying its capacity to more than one source subject to its capacity.
- (ii) A source should get its entire requirement from one or more main sources and source is supplying its capacity to more than one destination
- (iii) A destination should get its entire requirement from one source.

The objective of the problem “Two-Stage Multi Objective Variant Bulk TP Model” (TSMVBTP) is to minimize the total distance and the total bulk transportation cost on transportation of the products in two stages subjected to the constraints. In this problem we have studied a variation of Transportation Problem called Two-Stage Transportation Problem.

1.1. Variations of Bulk Transportation Problem:

There are various types of transportation models. The problem of Bulk Transportation was first investigated by Maio and Roveda[6] who presented an algorithm. Srinivasan and Thompson [13] also offered an algorithm for the Bulk Transportation Problem based on Branch and Bound procedure. The well known and much simpler “Plant Location” models Balinski and Spielberg [1] Ellwein and Gray[2] for surveys. Marks, Liebman and Bellmore [7] report reasonably good computational experience with a conventional branch and bound algorithm to capacitated transshipment problems.

Sundaramurthy [14] has studied the Bulk Transportation Problem and solved it using Lexi-search algorithm using pattern recognition technique developed by him. This problem with the additional restriction that a destination should get its supply from one source only is studied in another Bulk-Transportation problem. Pandit and Sundaramurthy [9], Sundaramurthy presented a Lexi-Search algorithm which takes out the drawbacks of Maio De and Rovedo algorithm and takes the advantage of the simple structure of the problem to get an optimal feasible solution. Naganna [8] has studied a general Three Dimensional Bulk Transportation Problem. Here the problem takes the form of zero – one programming problem.

Junginer[4] who proposed a set of logic problems to solve multi-index transportation problems has also conducted a detailed investigation regarding the characteristics of multi-index transportation problem model. Rautman et al [11] used multi-index transportation problem model. In the case of cost-time trade- off bulk transportation problem further more Sobhan Babu and Sundaramurthy [12] presented Lexi Search algorithms and studied that the efficiency of this algorithm over branch and bound algorithm.

Hinojosa[3] has investigated a multi-period two-echelon multi-commodity capacitated plant location problem. Linda van Norden and Steef van de Velde[5] studied a multi-product lot-sizing model in exchange for a guaranteed price. Naganna Sobhan Babu et al. have studied a variety of the problems. Purusotham and Sundaramurthy [10] presented a Multi-Product Bulk Transportation Problem to minimize the total cost of the bulk transportation.

The Time Minimizing Transportation Problem also known as the Bottleneck Transportation Problem. Furthermore Sobhan Babu and Sundaramurthy have studied a variety of problems and presented Lexi Search algorithms and mentioned that the efficiency of this algorithm over branch and bound algorithm. Suresh Babu [15] studied A Variant Bulk Transportation Problem and Vidhyullatha [16] presented three dimensional time minimization bulk transportation problem to minimize the total time of goods transportation.

II. PROBLEM DESCRIPTION

The present problem namely “Two-Stage Multi Objective Variant Bulk TP Model” (TSMOVBTM) is a more generalized model of the two-stage transportation problem and comes under combinatorial programming problems. In this problem there is a set of p main sources, m sources and n destinations i.e. $M=\{1,2,\dots,p\}$, $S=\{1,2,\dots,m\}$ and $DR=\{1,2,\dots,n\}$. The quantity availability at the main source is given by $M(i)$ where $M(i)$ is the quantity available in the i^{th} main source. Similarly $S(i)$ is the capacity at the i^{th} source and $DR(i)$ is the requirement at the i^{th} destination. Being the problem is truncated out of m sources and n destinations m_0 sources and n_0 destination has to satisfy it.

$$\sum_{i \in DR} DR(i) \leq \sum_{i \in S} S(i) \leq \sum_{i \in M} M(i)$$

Now D is the distance matrix where $D(i,j)$ is the distance from i^{th} main source to j^{th} source, $i \in M, j \in S$. C is the cost matrix where $C(i,j)$ is the bulk cost of transporting from i^{th} source to j^{th} destination. $i \in S, j \in DR$.

The requirement of destinations should be satisfied from the sources and the requirement of the sources should be satisfied by main sources. From main sources to sources we consider the distances in the first schedule and from sources to the destinations we consider the cost in the second schedule.

The objective of the problem is to find the supply in first schedule from main sources to the sources and the second schedule from sources to the destinations such that the total distance is minimum and the total cost is also minimum

III. MATHEMATICAL FORMULATION

$$\text{Min}_{(x,y)} Z = \text{Min}_x \left[\sum_{i \in M} \sum_{j \in S} D(i,j) \cdot x(i,j) \right] + \text{Min}_y \left[\sum_{i \in S} \sum_{j \in DR} C(i,j) \cdot y(i,j) \right] \dots\dots\dots(1)$$

Subject to the constraints

$$\left. \begin{aligned} \sum_{i \in M} x(i,j) \alpha(i,j) &= S(j), \quad \forall j \in S \\ y(i,j) DR(j) &= DR(j), \quad \forall j \in DR \\ \text{for one } i \in S \end{aligned} \right\} \sum_{i \in M} \sum_{j \in S} x(i,j) \geq m_0, \quad \sum_{i \in S} \sum_{j \in DR} y(i,j) = n_0 \dots\dots\dots(3)$$

$$x(i,j) = 1 \text{ or } 0, \quad y(i,j) = 1 \text{ or } 0 \dots\dots\dots(4)$$

Equation (1) represents objective function of the problem i.e. to minimize the total distance and total bulk transportation cost subjected to the constraints. Equation (2) states that the transportation of available capacity from one or more main sources to source and requirement of any destination should be supplied from one source. Equation (3) denotes that number of connectivities in first schedule from main sources to sources should be greater than or equal to m_0 (sources= m_0) and in second schedule requirement at destinations from m_0 sources exactly equal to n_0 (destinations= n_0). Equation (4) illustrates that if i^{th} main source supply to j^{th} source then it is 1, otherwise it is 0 and also if supply from i^{th} source to j^{th} destination then it is 1, otherwise it is 0.

IV. NUMERICAL ILLUSTRATION:

The concepts and algorithm developed will be illustrated by a numerical example. In which the number of main sources (p) is 2, number of Sources (m) is 3 and number of Destinations (n) is 7. i.e. $M=\{1,2\}$, $S=\{1,2,3\}$ and $DR=\{1,2,3,4,5,6,7\}$. $M(i)$ is the availability at the main source, $S(i)$ is the demand at sources and $DR(i)$ is the requirement of

the product at the destinations. The capacity of $M(i)$, $S(i)$ and $DR(i)$ are given in Tables 3 and 4. Here $p_0=2, m_0=2$ and $n_0=6$ has to satisfy being truncated problem. The distance and cost matrices $D(i,j), \forall i \in M, j \in S$ and $C(i,j), \forall i \in S, j \in DR$ are given in following Tables 1 and 2.

Distance Matrix

Table-1

	1	2	3
1	1	2	2
2	2	2	3

Cost Matrix

Table-2

	1	2	3	4	5	6	7
1	1	2	3	2	1	1	10
2	2	3	5	12	1	6	2
3	3	7	9	9	6	8	11

The availability of Main sources, sources and requirement at Destinations are given in the following tables 3 and 4.

Table-3

Available Capacities

Main Source	Sources
M1=90	60
M2=90	60
	60

Table-4

Requirement of Destinations

Destination	1	2	3	4	5	6	7
DR(i)	20	40	10	20	10	20	10

V. THE CONCEPTS AND DEFINITIONS

5.1 Definition of a pattern:

An indicator the three dimensional array X which is associated with an allocation is called a "pattern". A pattern is said to be feasible if X is a feasible solution. The feasible solution in figure -1 can be represented by an indicator array X (Three dimension). If $X(i,j,k)=1$ when $k=1$ it indicates that i^{th} main source supplying to j^{th} source with distance $D(i,j)$ which is also equal to $DC(i,j,1)$ and if $k=2$ then the j^{th} destination get its requirement from i^{th} source with cost $C(i,j)$ which is also equal to $DC(i,j,2)$. The third dimensional array X is also called a pattern for the above feasible solution. The ordered triple set $\{(1,1,1), (1,4,2), (1,5,2), (1,2,1), (2,2,1), (1,1,2), (1,3,2), (2,6,2), (2,2,2)\}$ represents the pattern as given in tables 5 and 6 represent a feasible pattern

Table-5

$$X(i,j,1) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Table-6

$$X(i, j, 2) = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Now the value of the pattern X is defined as:

$$Z = \sum_i \sum_j \sum_k DC(i,j,k)X(i,j,k) \text{ where } i \in M \cup S, j \in S \cup DR \text{ and } k \in \{1,2\} \text{ for the numerical example. The value}$$

$V(X)$ gives the total distance and cost of the allocation for the solution represented by X . Thus the value of the feasible pattern gives the total distance and cost represented by it. In the sequel an algorithm is developed in which a search is made for a feasible pattern with least value. $V(X) = \sum \sum DC(i,j,1)X(i,j,1) + \sum \sum DC(i,j,2)X(i,j,2)$

Consider the ordered triples and the value of such pattern are

$$V(X) = DC(1,1,1) + DC(1,4,2) + DC(1,5,2) + DC(1,2,1) + DC(2,2,1) + DC(1,1,2) + DC(1,3,2) + DC(2,6,2) + DC(2,2,2)$$

$$V(X) = 1+1+1+2+2+2+2+2+5 = 18$$

5.2. Alphabet – Table and a Word:

There are $p \times m$ $D(i,j)$'s and $m \times n$ $C(i,j)$'s in two dimensional array. $D(i,j)$ represents distance and $C(i,j)$ represents cost which are two independent factors. $DC(i,j,k)$ is a three dimensional array which represents both $D(i,j)$ and $C(i,j)$ such that $DC(i,j,1) = D(i,j)$ and $DC(i,j,2) = C(i,j)$. The elements of $DC(i,j,k)$ are arranged in increasing order of their corresponding values and are indexed from 1,2,3,...[Sundaramurthy 1979]. Let $SN = \{1, 2, 3, \dots\}$ be the set of indices. Let DC be the corresponding array of distances and costs. For convenience same DC if $a, b \in SN$ and $a < b$ then $DC(a) \leq DC(b)$. Let CDC be the array of cumulative sum of the elements of DC . Also let R, C, K be arrays of row, column indices respectively of the ordered triples represented by SN . The arrays of $SN, DC, CDC, R, C,$ and K for the numerical example which is given in Table-7 is an Alphabet-Table. If $a \in SN$ then $[R(a), C(a), K(a)]$ is the ordered triplet and

$$\text{the value of the ordered triplet } CDC(a) = \sum_{i=1}^a DC(i)$$

Table-7 Alphabet-Table

S.No	DC	CDC	R	C	K
1	1	1	1	1	1
2	1	2	1	4	2
3	1	3	1	5	2
4	1	4	2	4	2
5	2	6	1	2	1
6	2	8	1	3	1
7	2	10	2	1	1
8	2	12	2	2	1
9	2	14	1	1	2
10	2	16	1	3	2
11	2	18	2	6	2
12	3	21	2	3	1
13	3	24	1	2	2
14	3	27	2	1	2
15	5	32	2	2	2
16	5	37	3	7	2
17	6	43	2	5	2
18	6	49	3	4	2
19	7	56	3	1	2
20	8	64	1	7	2
21	8	72	3	5	2
22	9	81	3	2	2
23	9	90	3	3	2
24	10	100	1	6	2
25	11	111	3	6	2
26	12	123	2	3	2
27	12	135	2	7	2

Let us consider $10 \in SN$. It represents the ordered triple $[R(10), C(10), K(10)] = (1,3,2)$ and the value of $CDC(10) = 16$. Let $L_r = (a_1, a_2, a_3 \dots a_r), a_i \in SN$ be an ordered sequence of r indices from SN . The pattern represented by the ordered triple whose indices are given by L_r is independent of the order of a_i in the sequence. Hence for uniqueness, the indices are arranged in the increasing order such that $a_i < a_{i+1}$, for $i = 1, 2 \dots r-1$. The ordered sequence L_r is defined as a 'word' of length r . The word is called a "sensible word" if $a_i < a_{i+1}$, for $i = 1, 2 \dots r-1$ and if this condition is not met it is called a "non-sensible word". A word L_r is said to be partially feasible word if the block of words represented by L_r has at least one feasible word or equivalently the partial pattern represented by L_r should not have any inconsistency.

5.3. Value of the Word:

The value of the word L_r is defined as $V(L_r) = V(L_{r-1}) + DC(a_r)$ with $V(L_0) = 0$, the distance and cost array arranged such that $DC(a_r) \leq DC(a_{r+1})$. The values of $V(L_r)$ and $V(X)$ will be the same for the pattern X represented by L_r . For example consider $L_4 = (1, 2, 3, 5)$. Then $V(L_4)$ will be described as follows.

$$V(L_4) = V(L_3) + DC(5) = V(L_2) + DC(3) + DC(5) = V(L_1) + DC(2) + DC(3) + DC(5) = V(L_0) + DC(1) + DC(2) + DC(3) + DC(5) = 0 + 1 + 1 + 1 + 2 = 5$$

5.4. Lower bound of a Partial Word $LB(L_r)$:

A lower bound $LB(L_r)$ for the values of the block of words represented by $L_r = (a_1, a_2, a_3, \dots, a_r)$ can be defined as follows

$$LB(L_r) = V(L_r) + CDC[a_r + (n_0 + m_0 - NA - NB)] - CDC(a_r)$$

where NA is number of main sources covered and NB is number of sources covered. Consider the partial word $L_4 = (1, 2, 3, 5)$ then $V(L_4) = 1 + 1 + 1 + 2 = 5$ and $LB(L_r) = V(L_r) + CDC[a_r + (n_0 + m_0 - NA - NB)] - CDC(a_r) = V(L_4) + CDC[5 + 6 + 2 - 1 - 2] - CDC(5) = 5 + CDC(10) - CDC(5) = 5 + 16 - 6 = 15$

5.5. Feasibility Criterion of a Partial Word:

A feasibility criterion is developed in order to check the feasibility of a partial word $L_{r+1} = (a_1, a_2, a_3, \dots, a_k, a_{r+1})$ given that L_r is a partial word. We will introduce some more notations which will be useful in the sequel.

IR: IR be an array where $IR(i)=1, i \in M$ indicates that i^{th} main source is connected to some source of $j, j \in S$ otherwise $IR(i)=0$

IC: IC be an array where $IC(j)=1, j \in S$ indicates that j^{th} source is connected by some main source of $i, i \in M$ otherwise $IC(j)=0$

SB: SB be an array where $SB(i)=1, i \in S$ indicates that i^{th} source is connected to some destination of $j, j \in DB$ otherwise $SB(i)=0$

DB: DB be an array where $DB(j)=1, j \in DR$ indicates that j^{th} destination is connected by some source of $i, i \in S$ otherwise $DB(j)=0$.

IK: IK be an array where $IK(j)=1, j \in S$ indicates j^{th} source is connected to some main source $i \in M$ $IK(j)=2, j \in DB$ that j^{th} destination is connected to some source of $i, i \in S$ otherwise $IK(j)=0$.

L: L be an array where $L(i) = a_i$ is the letter in the i^{th} position of a word.

For example consider a sensible partial word $L_4 = (1, 2, 3, 5)$ which is feasible. The recursive algorithm for checking the feasibility of a partial word L_p is given as follows. In the algorithm we equate $IX = 0$. At the end if $IX = 1$, then the partial word is feasible otherwise it is infeasible. For this algorithm we have Lexi-Search algorithm to find an optimal feasible word developed as follows:

Table-8

	1	2	3	4	5	6	7	8	9
L	1	2	3	5	-	-	-	-	-
IR	1,1	-	-	-	-	-	-	-	-
IC	1	1	-	-	-	-	-	-	-
SB	1,1	-	-	-	-	-	-	-	-
DB	-	-	-	1	1	-	-	-	-
IK	1	2	2	1	-	-	-	-	-

VI. ALGORITHMS

6.1. Algorithm – 1: (Algorithm for feasible checking)

```

Step 100:  IX=0                                go to 101
Step 101 :  Is KA=1                            Yes Distance go to 102
                                                No cost go to 201
Step 102:  Is IR(RA)=0                        Yes PA=PA+1,
                                                IR(RA)= IR(RA)+1 go to103,
                                                No go to 207
Step 103:  Is PA=p0                            Yes go to 207
                                                No go to 104
Step 104:  Is NA=m0                            Yes go to 207
                                                No go to 105
Step 105:  LD (RA) = MX (RA)- ML (RA) go to 106
Step 106:  Is IC(CA)=0                        Yes go to107
                                                No go to 207
Step 107:  LE(CA)= SX (CA)-SL(CA) go to 108
Step 108:  Is LD (RA) >LE (CA)
          A=LE(CA), YES go to 206
          No A=LD(RA), go to 206
Step201:  KA=2,
          SA=RA,
          DA=CA                                goto202
Step 202:  Is NB = n0                          yes go to 207
                                                no go to 203
Step203:  Is DB (DA) =1                      yes go to 207
          N{NB=NB+1&B(DA)=DB(DA)+1
                                                goto204A
Step 204A: Is SB(SA) =0;
          SB(SA) = SB(SA) +1 yes go to 204
          No go to 204
    
```

Step 204: $LF(SA) = SC(SA) - SD(SA)$ go to 205
 Step 205: Is $LF(SA) \geq DL(DA)$
 Yes $B=DL(DA)$ yes go to 206
 no go to 207
 Step 206: $IX=1$
 Step 207: STOP

6.2. Algorithm – 2: (Lexi-search Algorithm)

Step 1: (Initialization)

The arrays SN, DC, CDC, R, C, K, MX, SX, DX, LD,LE,LF and MAX are made available. DB, SB, IR, IC, L, V, LB, SL, DL. ML, PA, NA, NB are initialized to zero. The values $VT=999, I=1, j=0, p=p_0-2, m=3, m_0=2, n=7, n_0=6$

Step 2: $j=j+1$
 Is $j < MAX$ Yes go to 3; No go to 16
 Step 3: $L(i)=j, RA=R(j), CA=C(j),$
 $KA=K(j)$ go to 4
 Step 4: (Check feasibility by using Algorithm)
 Is $IX=0$ Yes go to 2 No go to 5
 Step 5: $V(i) = V(i-1)+DC(j)$
 $LB(i) = V(i) + CDC[j+m_0+n_0+-NA-NB] - CDC [j]$ go to 6
 Step 6: Is $LB(i) \geq VT$ Yes goto 16, No go to 7
 Step 7: Is $KA=1$ Yes go to 8; No go to 8A

Fixation

Step 8: $ML(RA) = ML(RA)+A$
 $SL(CA) = SL(CA)+A$
 Is $SX(CA)=SL(CA)$
 Yes $NA=NA+1$
 $IC(CA)=IC(CA)+1$ go to 9
 No go to 9

Fixation

Step 8A: $SD(SA) = SD(SA)+B$
 $DX(DA) = DX(DA)+B$ go to 9
 Step 9: Is $PA=p_0 \& NA=m_0, NB= n_0$ yes go to 11
 No go to 10
 Step 10: $I=I+1$ go to 2
 Step 11: VT a Solution goto 11 F

Nullification

Step 11F: $L(i)=j$
 Is $KA=1$ Yes go to 11A, No go to 11B
 Step 11A: $RA=R(j), CA=C(j), ML(RA)= ML(RA)-A$
 Is $ML(RA)=0$ Yes $PA=PA-1$ go to 11D
 No go to 11D

Step 11D: $SL(CA)= SL(CA)-A$
 Is $SL(CA)=0$ yes $A=NA-1$ go to 11C
 $IC(CA)= IC(CA)-1$ No go to 11C

Step 11B: $SD(SA)= SD(SA)-B,$
 $DL(DA)=DL(DA)-B, NB=NB-1$
 $DB(DA)=DB(DA)-1$ go to 11C

Step 11C: $L(i+1)=0, V(i+1)=0$ go to 12 Step 12: $i=i-1$ go to 12A

Step 12A: $j=L(i)$
 Is $KA=1$ Yes go to 13, No go to 14

Step 13: $RA=R(j), CA=C(j)$
 $ML(RA)=ML(RA)-A$

Is $ML(RA)=0$ Yes $PA = PA-1$ goto 13 $IR(RA)=IR(RA)-1$ No go to 13A

Step 13A: $SL(CA)= SL(CA)-A$
 Is $SX(CA)=0$ Yes $NA=NA-1, IC(CA)= IC(CA)-1$ go to 15
 No go to 15

Step 14: $SA=RA, DA=CA, SD(SA)= SD(SA)-B$
 $DX(DA)=DX(DA)-B, NB=NB-1, DB(DA)=DB(DA)-1$ go to 15

Step 15: $L(i+1)=0, V(i+1)=0$
 $LB(i+1)=0$ go to 2

Step 16: Is $i=1$ Yes go to 17
 No go to 12

Step 17: End.

This recursive algorithm will be used as sub routine in the Lexi-Search algorithm. We start the algorithm with a very large value say 999 as a trail value of VT. If the value of the feasible word is known, we can as well start with that value as VT. During the search the current value of VT gives the optimal feasible word. We start with the partial word $L_1 = (a_1) = 1$. A partial word $L = L_r * (a_r)$ where * indicates chain form or concatenation. We will calculate the value of LB (L_r). Then two cases arises (one for branching and other for continuing the search):

- (i) If $LB(L_r) < VT$. Then we check whether L_r is feasible or not. If it is feasible we proceed to consider a partial word of order (r+1), which represents a sub-block of the block of words represented by L_r . If L_r is not feasible then consider the next partial word of order r by taking another letter which succeeds a_r in the r^{th} position. If all the words of order p are exhausted then we consider the next partial word of order (r-1).
- (ii) If $LB(L_r) \geq VT$. In this case we reject the partial word meaning that the block of words with L_r as leader is rejected for not having an optimal word and we also reject all partial words of order p that succeeds L_r .

The current value of VT at the end of the search is the value of the optimal feasible word. At the end if $VT = 999$, it indicates that there is no feasible solution.

VII. SEARCH – TABLE

The working details of getting an optimal word using the above algorithm for the illustrative numerical example is given in Table – 9. The columns (1), (2), (3), (4), (5), (6), (7) ,(8) and (9) gives the letters in first, second, ..., ninth places respectively. The indices R,C and K give row, column and facility(distance/cost) are indices of the letter. The corresponding $V(i)$ and $LB(i)$ give the value and lower bound of a word. The last column gives the remarks regarding the acceptability of the partial words. In the following table A indicates Acceptance and R indicates Rejection.

Table-9 Search –Table

S.NO	1	2	3	4	5	6	7	8	9	V	LB	R	C	K	REM
										Distance+Cost					
1.	1									1+0	14	1	1	1	A
2.		2								1+1	14	1	4	2	A
3.			3							1+ 2	14	1	5	2	A
4.				4						1+2	14	2	4*	2	R
5.				5						3+2	15	1	2	1	A
6.					6					5+2	15	1*	3*	1	R
7.					7					5+2	15	2	1*	1	R
8.					8					5+2	16	2	2	1	A
9.						9				5+4	16	1	1	2	A
10.							10			5+6	16	1	3	2	A
11.								11		5+8	16	2	6	2	A
12.									12	8+8	16	2	3*	1	R
13.									13	5+11	16	1*	2	2	R
14.									14	5+11	16	2	1*	2	R
15.									15	5+13	18	2	2	2	A
16.								12		8+6	17	2	3*	1	R
17.								13		8+6	17	1*	2	2	R
18.								14		8+6	19	2	1*	2	R,>VT
19.									11	5+6	17	2	6	2	A
20.								12		8+6	17	2	3*	1	R
21.								13		5+9	17	1*	2	2	R
22.								14		5+9	19	2	1	2	R,>VT
23.									12	8+4	18	2	3*	1	R,=VT
24.								10		5+4	17	1	3	2	A
25.									11	5+6	17	2	6	2	A
26.								12		8+6	17	2	3*	1	R
27.								13		5+9	17	1*	2	2	R
28.								14		5+9	19	2	1	2	R,>VT
29.									12	8+4	18	2	3*	1	R,=VT
30.									11	5+4	18	2	6	2	R,=VT
31.					9					3+4	17	1	1	2	A
32.						10				3+6	17	1	3	2	A
33.							11			3+8	17	2	6	2	A
34.								12		6+8	17	2	3*	1	R
35.								13		3+11	17	1*	2	2	R
36.								14		3+11	19	2	1*	2	R,>VT
37.									12	6+6	18	2	3*	1	R,=VT

38.					11				3+6	18	2	6	2	R,=VT
39.				10					3+4	18	1	3	2	R,=VT
40.			6						3+2	15	1	3	1	A
41.				7					5+2	15	2	1*	1	R
42.				8					5+2	16	2	2*	1	R
43.				9					3+4	17	1	1	2	A
44.					10				3+6	17	1*	3	2	A
45.					11				3+6	18	2*	6	2	R,=VT
46.				10					3+4	18	1	3	2	R,=VT
47.			7						3+2	16	2	1*	1	R
48.			8						3+2	17	2	2	1	A
49.				9					3+4	17	1	1	2	A
50.					10				3+6	17	1	3	2	A
51.						11			3+8	17	2	6	2	A
52.							12		6+8	17	2	3*	1	R
53.							13		3+11	17	1*	2	2	R
54.							14		3+11	19	2	1*	2	R,>VT
55.						12			6+6	18	2	3*	1	R,=VT
56.					11				3+6	18	2	6	2	R,=VT
57.				10					3+4	18	1	3	2	R,=VT
58.			9						1+4	18	1	1	2	R,=VT
59.		4							1+2	15	2	4*	2	R
60.		5							3+1	16	1	2	1	A
61.			6						5+1	16	1*	3*	1	R
62.			7						5+1	17	2	1*	1	R
63.			8						5+1	18	2	2	1	R,=VT
64.		6							3+1	17	1*	3*	1	R
65.		7							3+1	18	2	1	1	R,=VT
66.	3								1+1	15	1	5	2	A
67.		4							1+2	15	2	4	2	A
68.			5						3+2	15	1	2	1	A
69.				6					5+2	15	1	3*	1	R
70.				7					5+2	15	2	1*	1	R
71.				8					5+2	16	2	2	1	A
72.					9				5+4	16	1	1	2	A
73.						10			5+6	16	1	3	2	A
74.							11		5+8	16	2	6	2	A
75.							12		8+8	16	2	3*	1	R
76.							13		5+11	16	1*	2	2	R
77.							14		5+11	16	2	1*	2	R
78.							15		5+13	18	2	2	2	R,=VT
79.							12		8+6	17	2	3*	1	R
80.							13		5+9	17	1*	2	2	R
81.							14		5+9	19	2	1*	2	R
82.						11			5+6	17	2	6	2	A
83.							12		8+6	17	2	3*	1	R
84.							13		5+9	17	1*	2	2	R
85.							14		5+9	19	2	1*	2	R,>VT
86.						12			8+4	18	2	3*	1	R=VT
87.					10				5+4	17	1	3	2	A
88.						11			5+6	17	2	6	2	A
89.							12		8+6	17	2	3*	1	R
90.							13		5+9	17	1	2	2	A
91.							14		5+12	17	2	1	2	A
92.							14		5+9	19	2	1	2	R,>VT
93.						12			8+4	18	2	3	1	R,>VT
94.					11				5,+4	18	2	6	2	R,>VT
95.				9					3+4	17	1	1	2	R,=VT
96.			6						3+2	15	1	3	1	A
97.				7					5+2	15	2	1*	1	R

98.				8					5+2	16	2	2*	1	R
99.				9					3+4	17	1	1	2	R,=VT
100.			7						3+2	16	2	1*	1	R
101.			8						3 +2	17	2	2	1	R,=VT
102.		5							3 +1	16	1	2	1	A
103.			6						5 +1	16	1	3*	1	R
104.			7						5 +1	17	2	1*	1	R,=VT
105.		6							3 +1	17	1	3	1	R,=VT
106.	4								1+1	16	2	4	2	A
107.		5							3+1	16	1	2	1	A
108.			6						5+1	16	1*	3*	1	R
109.			7						5+1	17	2	1*	1	R,=VT
110.		6							3+1	17	1*	3*	1	R,=VT
111.	5								3+0	18	1	2	1	R,>VT
112.	2								0+1	15	1	4	2	A
113.	3								0+2	15	1	5	2	A
114.		4							0+3	15	2	4*	2	R
115.		5							2 +2	16	1	2	1	A
116.			6						4 +2	16	1	3*	1	R
117.			7						4 +2	17	2	1	1	R,=VT
118.		6							2+2	17	1	3	1	R,=VT
119.	4								0+2	16	2	4	2	A
120.		5							2 +2	16	1	2	1	A
121.			6						4 +2	16	1	3*	1	R
122.			7						4 +2	17	2	1	1	R,=VT
123.		6							2 +2	17	1	3	1	R,=VT
124.	5								2 +1	18	1	2	1	R,>VT
125.	3								0+1	16	1	5	2	A
126.	4								0+2	16	2	4	2	A
127.		5							2 +2	16	1	2	1	A
128.			6						4 +2	16	1	3*	1	R
129.			7						4 +2	17	2	1	1	R,=VT
130.		6							2 +2	17	1	3	1	R,=VT
131.	5								2 +1	18	1	2	1	R,>VT
132.	4								0+1	18	2	4	2	R,>VT

At the end of the search the value of VT is 17 and it is the value of the optimal feasible word $L_9 = \{1, 66, 67, 68, 71, 72, 87, 90, 91\}$. It is represented in 91st row of the search table-9 with corresponding ordered triples. It involves selection of sources $\geq m_0$ and exactly equal to n_0 destinations. The ordered pairs from distance $D(i,j)$ and cost $C(i,j)$ are two independent factors in two dimensions are taken as a single three dimensional array $DC(i,j,k)$ and select three ordered pairs from distance and six ordered pairs from cost for obtaining optimum solution. The ordered triples (i,j,k) are: (1,1,1), (1,5,2), (2,4,2), (1,2,1), (2,2,1), (1,3,2), (2,6,2), (1,2,2),(2,1,2),

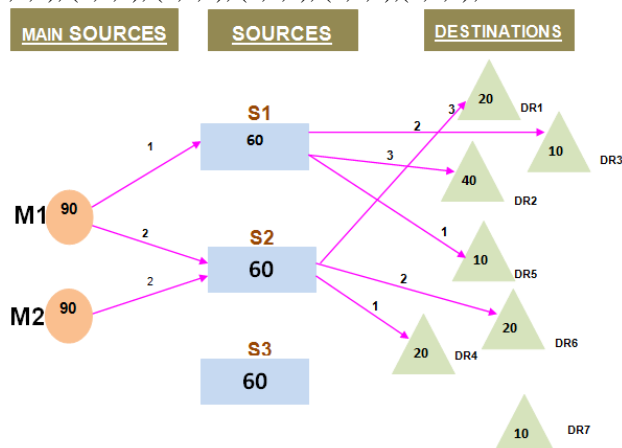


Figure -1 [Optimal solution]

The ordered triple set $\{(1,1,1), (1, 5,2), (2,4,2), (1,2,1), (2,2,1), (1,3,2), (2,6,2), (1,2,2),(2,1,2)\}$ represents the pattern given in the tables 9 and 10 which is an optimal solution for the above numerical example.

Table – 10

$$X(i,j,1)= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Table 11

$$X(i, j, 2) = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The above pattern represents graphically in the figure – 3 with clear terms. the above figure-3, Source S1 is gets its capacity of 60 units from Main source M1 with 1 unit of distance, Source S2 is gets its capacity of 60 units 30 units from Main source M1 having 2 units of distance and 30 units from Main source M2 with 2 units of distance. The third source S3 is left out ie. $m_0=2$ in first schedule. In second schedule destinations DR2,DR3, and DR5 are getting their requirements of 40,10and 10 units from Source S1 with costs of 3,2 and 1 units respectively. Similarly destinations DR1, DR4 and DR6 are getting their entire requirement of 20,20 and 20 units from Source S2 with 3,1 and 2 units of cost respectively. The seventh destinationDR7 is left out.i.e. $n_0=6$

In the above solution there are 3 connectivities from Main sources to sources i.e., $3 \geq m_0$ (where $m_0=2$ in this problem) and exactly 6 connectivities from sources to destinations i.e., $n_0=6$.Here in this bulk transportation process source S3 and destination DR7 are truncated. The above figure-3 satisfies all the constraints in the mathematical formulation. It is an optimal feasible solution. The two schedules of total distance and total cost of bulk transportation is given. The distances and costs of corresponding ordered triples are represented with their values in the set as {DC(1,1,1)=1, DC(1, 5,2)=1, DC(2,4,2)=1, DC(1,2,1)=2, DC(2,2,1)=2, DC(1,3,2)=2, DC(2,6,2)=2, DC(1,2,2)=3, DC(2,1,2)=3}. Thus the optimal solution is:

$$Z(X)=DC(1,1,1)+DC(1,5,2)+DC(2,4,2)+DC(1,2,1) +DC(2,2,1)+DC(1,3,2)+DC(2,6,2)+DC(1,2,2)+DC(2,1,2) = 1+1+1+2+2+2+2+3+3=17$$

VIII. COMPUTATIONAL EXPERIENCE AND RESULTS

A Computer program for the proposed Lexi – Search Algorithm is written in C language and is tested for different random numbers. The experiments are carried out on a COMPAQ (dx2280 MT) system by generating the uniformly random [0,1000] values for distance Matrix D (i, j) and cost matrix C (i, j). The values of D (i, j) and C (i, j) are tried a set of problems by giving different values to p, m and n. The results are shown in the Table -3.9 below. For each instance, five data sets are tested. It is seen that the time required for the search of the optimal solution is less.

Table 3.9

SN	P	m	n	P ₀	m ₀	n ₀	NPT	VT	CPU Run Time in seconds
1	2	3	7	2	2	6	5	999	0.000
2	3	5	10	2	3	7	5	999	0.1098
3	4	6	10	3	5	9	5	999	0.5943
4	5	10	16	4	8	13	5	999	0.3846
5	6	10	15	4	9	12	5	999	0.1648
6	8	10	20	5	8	16	5	999	2.5495
7	8	13	22	5	10	18	5	999	3.0175
8	9	15	23	6	11	17	5	999	2.7847
9	10	15	25	6	12	20	5	999	3.2578
10	9	10	25	7	9	22	5	999	3.3824
11	10	15	26	7	12	20	5	999	1.8850
12	8	15	30	7	10	24	5	999	3.9072
13	9	16	35	6	14	24	5	999	4.2568

In the Table-3.9, SN = serial number, p = number of main-sources, m = number of sources, n = number of destinations, p₀ = number of main-sources covered, m₀ = number of sources covered, n₀ = number of destinations covered, NPT= number of problems tried and CPU run time for searching an optimal solution.

IX. CONCLUSION

In this model, we presented an exact algorithm called Lexi-search algorithm based on pattern recognition technique to solve the TSPMJF Problem. Lexi-search algorithms are proved to be more efficient in many combinatorial problems. First the model is formulated into a zero-one programming problem. A Lexi-Search Algorithm based on Pattern Recognition Technique is developed for getting an optimal solution. The problem is illustrated with help

of suitable numerical example in detail. We planned the proposed algorithm using C-language. The computational details are reported and compare the existing results with the available model. As an observation the CPU run time is reasonably less for higher values of the problem to obtain an optimal solution than the available model. Based on this experience we strongly consider that this algorithm can perform larger size problems.

ACKNOWLEDGEMENTS

The first author express my deep sense of reverence and gratitude to the research supervisor Prof. M .Sundara Murthy,(Rtd), Department of Mathematics, S.V.University, Tirupati, Andhra Pradesh, for suggesting this problem for investigation. It is solely due to his immense interest, critical analysis, exceptional guidance and concrete suggestions enlightened discussions, which cumulatively are responsible for the successful execution of this work.

REFERENCES

- [1] Balinski, M. L. and Spielberg, K. Methods for Integer Programming: Algebraic, Combinatorial and Enumerative, J. S. Aronofsky (ed), Progress in Operations Research, vol. III, Wiley, New York; 1969.
- [2] Ellwein, L.B., and Gray, P. Solving Fixed Charge Location – Allocation Problems with Capacity and configuration constraints. AIIE Transactions, III, 4, 290-298, 1971.
- [3] Hinojosa, Y.A multiperiod two-echelon multicommodity capacitated plant location problem. European Journal of Operational Research, 123(2), 271 -291, 2000.
- [4] Junginer, W. On representatives of multi-index transportation problems.European Journal of Operational Research, 66, 353-371, 1993.
- [5] Linda van norden and Steef van de velde (2005): “Multi-product lot-sizing with a transportation capacity reservation contract”, European Journal of Operational Research, Vol: 165(1), pp: 127-138.
- [6] Maio Adriano De and Roveda, C. An all zero-one algorithm for a certain class of transportation problems. Operations Research, 19, 1406-1418. 1971.
- [7] Marks, D. H., Liebman, J.C., and Bellmore, M. Optimal Location of Intermediate Facilities in a Trans-shipment Network, paper R-TP3.5 presented at the 37th National ORSA Meeting, Washington, D.C, 1970.
- [8] Naganna. B: (2007): “Operations Research. Ph. D thesis”, SriVenkateswara University, Tirupati, India
- [9] Pandit, S.N.N. andSundara Murthy, M. Allocation of sources to Destinations - paper presented in ORSI, held at Bombay, 1975 & PANDIT, S.N.N: (1962): “The Loading Problem”, Opns, Res., Vol:10, pp: 639-646.
- [10] Purusotham, S. & Sundara Murthy, M: (2011) “An Exact Algorithm for Multi Product Bulk Transportation Problem”, International Journal on Computer Science and Engineering (IJCSE) , Vol:3 (9),pp: 3222-3236.
- [11] Rautman, C.A., Reid, R.A., and Ryder, E.E. Scheduling the disposal of nuclear waste material in a geologic repository using the transportation model.Operations Research 41, 459-469, 1993.
- [12] SobhanBabu, K. and Sundara Murthy, M. (2010). An Efficient Algorithm for Variant Bulk Transportation Problem.International Journal of Engineering Science and Technology, vol. 2(7), pg. N0.2595-2600.
- [13] Srinivasan, V&Thompson, G.L :(1973):“An algorithm for assigning uses to sources in a special class of transportation problem”, Operations Research,vol: 21,pp: 284-295.
- [14] Sundara Murthy, M - (1976). A Bulk Transportation Problem Opsearch, 13, pp. 143-155.
- [15] Suresh Babu .C-2013.A Variant Bulk Transportation Problem, “Operations Research. Ph. D thesis”, Sri Venkateswara University, Tirupati, India
- [16] Vidyullata,A–1992,Combinatorial Programming Problems. M.Phil,dissertation, S.V.U.C.E., Tirupati, India