Chapter - V

A VARIANT BULK TRANSPORTATION PROBLEM WITH MULTIPLE BULK COST CONSTRAINT
5.1. Introduction:

The Classical Transportation Problem is to minimize the total cost for shipping the various capacities of the goods on the requirement of destinations from the available sources. The model often can be built as a linear programming model, an NP-hard problem. Generally the transportation cost of one unit of a commodity is depending on the source and the destination. In some situations the cost may not depend on the actual amount is transported, in that situation the cost is treated as 'Bulk-Transportation cost'. Sundara Murthy studied this problem with the additional restriction that a destination should get its supply from one source only, solved with the efficient Lexi-Search algorithm using the Pattern Recognition Technique and discussed that the efficiency of his algorithm over branch and bound algorithm.

The transportation problem can be generalized in many directions. The cost $c(i, j)$ given above, is a unit cost. In some situations the transportation cost may depend only on $(i, j)$ the source and the destination, but not on the actual amount transported. This cost is called the 'Bulk-Transportation cost'. This leads to one more generalisation of the transportation problem. The objective function in this case is $\sum \sum C(i, j) \cdot H(x(i, j))$, where $H$ is a step function with $H(\alpha) = 1$ if $\alpha > 0$ and $H(\alpha) = 0$ if $\alpha = 0$, and $C(i, j)$ is theBulk-Transportation cost. This problem with the additional restriction that a destination should get its supply from one source only is studied in another Bulk-Transportation problem (Sundara Murthy - 1976).

There are various types of transportation models and the simplest of them was first presented by Hitchcock (1941). It was further developed by Koopmans (1949) and Dantzig (1951). Several extensions of transportation models and methods have been subsequently developed. In these models, the given cost of transportation is for a unit quantity of item or commodity or product. But in Bulk Transportation Problem, the bulk transportation cost (independent of quantity transported) from a source $i$ to destination $j$ is taken into consideration.

The problem of Bulk Transportation was first investigated by Maio and Roveda (1971) who presented an algorithm. Later Srinivasan and Thompson (1973) also offered an algorithm. Both the algorithms are based on Branch and Bound procedure. Furthermore Sundara Murthy (1976), Sobhan Babu and Sundara Murthy (2010) presented Lexi Search algorithms and
mentioned that the efficiency of this algorithm over branch and bound algorithm. These investigations come under single commodity bulk transportation problems.

A few additional remarks are in order concerning how the present model fits into the existing literature. Its chief ancestors are, of course, the well known and much simpler “Plant Location” models (Balinski and Spielberg (1969); Ellwein and Gray (1971) for surveys). Marks, Liebman and Bellmore (1970) report reasonably good computational experience with a conventional branch and bound algorithm in which the linear programs, which specialize to capacitated trans-shipment problems.

Multi-index transportation problems are the extension of conventional transportation problems, and are appropriate for solving transportation problems with multiple supply points, multiple demand points as well as problems using diverse modes of transportation demands or delivering different kinds of products. Thus the forwarded problem would be more complicated than conventional transportation problems. Junginer (1993) 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. (1993), used multi-index transportation problem model to solve the shipping scheduling suggested that the employment of such transportation efficiency but also optimize the integral system. These references are only a single objective model and its constraints are not fuzzy member. In the case of cost-time trade-off bulk transportation problem prakash et al. (2008), (2009) studied multi objective models and presented efficient heuristic for this multi objective bulk transportation problem.

Hinojosa, Y. (2000) has investigated A multi-period two-echelon multi-commodity capacitated plant location problem, the problem deal with a facility location problem where one desires to establish facilities at two different distribution levels by selecting the time periods. , Linda van Norden and Steef van de Velde (2005) studied a multi-product lot-sizing model where, in any period, any portion of a reserved transportation capacity can be used in exchange for a guaranteed price. Miller et al. (1960), perskalla (1966), Picard et al. (1978), Bhavani and Sundara Murthy (2005), S. Das et al. (2006), Naganna (2007), Soban Babu et al. (2010) have studied a variety of the problems. Purusotham S and Sundara Murthy M (2011)
present 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 and has been studied by Hammer (1969), Garfinkel and Rao (1971) and Szwarc (1971). Hammer (1969) and Szwarc (1971) used labeling techniques to solve the problem. Garfinkel and Rao (1971) solved the problem by introducing sufficiently large cost M on certain routes. H. L. Bhatia, Kanti Swaroop and M. C. Puri (1976) also studied this problem and developed a technique which involved finite iterations. It was a mini-max problem with two dimensions. Later Pandit and Sundara Murthy - 1975, Sundara Murthy - 1976 presented a Lexi-Search algorithm which takes out the drawbacks of deMaio and Rovedo algorithm and takes the advantage of the simple structure of the problem to get an optimal feasible solution.

Naganna (2007) introduced a three dimensional time dependent Bulk transportation problem. Purusotham (2010) discussed a Multi-Product Bulk Transportation Problem (MPBTP) where one requests to get the requirement of different products depending on the availability from any source. The problem is to minimize the total cost of the bulk transportation for gathering the demands of all products specified over the planning subject of various warehouses. The above problem involves three dimensions. Under this consideration Miller et al. (1960), perskalla (1966), Picard et al. (1978), Bhavani and Sundara Murthy (2005), S. Das et al. (2006), Soban Babu et al. (2010) have studied a variety of problems. Vidhyullatha (2012) presented three dimensional time minimization bulk transportation problem to minimize the total time of goods transportation.

5.2. Pattern Recognition Technique based Lexicographic Search Approach:

Lexicographic Search Approach is a systematized Branch and Bound approach, developed by Pandit in the context of solving of loading problem in 1962. In principle, it is essentially similar to the Branch and Bound method as adopted by Little et.al.-1963. This approach has been found to be productive in many of the Combinatorial Programming Problems. It is significance mentioning that Branch and Bound can be viewed as a particular case of Lexicographic Search approach [Pandit - 1965]. The name Lexicographic Search itself suggests
that, the search for an optimal solution is done in a systematic manner, just as one searches for
the meaning of a word in a dictionary and it is derived from Lexicography the science of
effective storage and retrieval of information. This approach is based on the following grounds
Pandit - 1963].

(i) It is possible to list all the solutions or related configurations in a structural hierarchy
which also reflects a hierarchical ordering of the corresponding values of these
configurations.

(ii) Effective bounds can be set to the values of the objective function, when structural
combinatorial restraints are placed on the Allowable configurations.

The basic principle is described as follows [Rajbongshi-1982]. Consider a set of
symbols \( A = \{1, 2, 3, \ldots, n\} \) and the different possible sequences of length \( k \) of these symbols.
Thus \( (\alpha_1, \alpha_2, \ldots, \alpha_k) \) is a \( k \)-word, formed from the alphabet of \( n \) symbols 1, 2, 3 \ldots, \( n \). The \( i^{th} \)
letter in this word is \( \alpha_i \in A \). By defining an alphabetic order on the elements of \( A \), we will be
able to define a unique ordered list of words of length not exceeding \( m \), where \( m \) is finite.
Words of length \( k \leq m \) are called incomplete words standing for the set or block of the \( (m-k)! \)
Words of length \( k \). Searching for an optimum word is a problem of finding the word of minimum
value (in the case of a minimizing problem) in the Lexi Search defined by the solution of the
problem. The search efficiency of a Lexi Search algorithm is based in this approach depends on
the choice of an appropriate Alphabet-Table, where two conflicting characteristics of the search
list have to be taken into account: one is the difficulty in setting bounds to the values of the
partial words (that defines partial solutions representing subsets of solutions). The other
difficulty is in checking the feasibility of a partial word. Thus we get two situations in the choice
of the alphabet-table [Sundara Murthy (1979)]. In this problem we get the process of checking
the feasibility of a partial word is easy, while the calculations of a lower bound is bulky

When the process of feasibility checking of a partial word becomes difficult and
the lower bound computation is easy, a modified Lexi Search i.e. Lexi Search with recognizing
the Pattern of the Solution known as Pattern Recognition Technique (which was the main
contribution of Sundara Murthy 1979) can be adopted. In this method, in order to improve the
efficiency of the algorithm, first the bounds are calculated and then the partial word, for which the value is less than the initial (trial) value are checked for the feasibility. The pattern-recognition technique can be described as follows.

"A unique pattern is associated with each solution of a problem. Partial pattern defines a partial solution. An alphabet-table is defined with the help of which the words, representing the pattern are listed in a Lexicographic order. During the search for an optimal word, when a partial word is considered, first bounds are calculated and then the partial words for which the value is less than the trail value are checked for the feasibility".

Using Pattern Recognition technique reduces the dimensions requirement of the problem. For this problem find an optimal solution X which is a three dimensional array. the problem can be reduced to a linear form of finding an optimal word of length n. This reduction in the dimension for some problems reduces the computational work in getting an optimal solution [Sundara Murthy -1979, Vidyullata – 1992, Ramana and Umashankar –1995]. The present paper uses the Lexicographic Search in general and makes use of the Pattern Recognition approach.

5.3. Problem Description:

In this chapter we have studied a variation of the problems called "A Variant Bulk Transportation Problem with Multiple Bulk Cost Constraint". This is a more generalized model and comes under combinatorial programming problems. For understanding of the problem easily, where the products are supplied simultaneously to the destinations according to their requirement from the sources. The cost of transportation of products from the sources to destinations is given. Objective of the problem is to minimize the total bulk transportation cost subjected to the availability and requirement constraints.

In the bulk transportation problem C (i, j) is the cost of transport of the requirement of the destination J (j) from I (i) and it is independent of the units of the products, hence it is called the bulk cost. Hence in the pattern X (i, j)=0 or 1; if it is 1 it means the source i is supplying destination j, otherwise 0 i.e., X (i, j) takes 0 or 1 values. When quantities are large this model is not considered because C (i, j) will be bulk cost only but the quantity is fixed at say [140]
$\alpha$, as a result $C(i, j)$ will be the bulk cost which is one bulk unit $\alpha$ supplying from source $i$ to destination $j$. If the destination $j$ requires $k\alpha$, i.e., $k$ times the bulk unit and it can be supplied from source $i$ subjected to the availability, then the cost is $k \times C(i, j)$, then $X(i, j) = k$. In many practical cases $k = 1, 2$ or $3$. If $k$ is more than $3$ then the source person is will supply to extra quantity freely, because of competition. As a result in many cases $k$ is restricted to some finite number $1, 2, 3$ or $4$. The model where $X(i, j)$ can take $1, 2, 3$ or $4$ is more practical and useful. So this model we call it as A Variant Bulk Transportation Problem with multiple Bulk Cost Constraint.

There is a set of $I = \{1, 2, 3... m\}$ sources which produces a particular product and set of $J = \{1, 2, 3... n\}$ destinations with the set $K = \{1, 2, 3... l\}$ of $l$ bulk units. Here in this problem we are introducing a bulk unit which makes this problem more versatile. The requirement of place $j \in J$ is $DR(j)$ and the capacity of the source $i \in I$ is $SA(i)$. The cost for bulk unit $\alpha$ transportation from source $i$ to the place $j$ is $C(i, j)$ for $i \in I, j \in J$. The objective is to minimize the total bulk transportation cost subjected to the availability and requirement constraints. The model can be built as an Integer programming problem. The Pictorial representation of the problem is given in the following figure -5.1. In figures shape of cylinder represents the sources and the boxes of rectangular shapes represent destinations. $k$ represents the number of bulk units transported from the corresponding source $I$ to destination $J$. 
The mathematical formulation is given in the following section.

\[
\text{Minimize } Z = \sum_{i \in I} \sum_{j \in J} C(i, j) X(i, j) \tag{1}
\]

Subject to

\[
\sum_{i = 1}^{m} S_A(i) \geq \sum_{j = 1}^{n} D_R(j) \tag{2}
\]

\[
\sum_{j = 1}^{n} a_X(i, j) \leq S_A(i) \quad \forall i \in I \tag{3}
\]

\[
\sum_{i = 1}^{m} a_X(i, j) = D_R(j) \quad \forall j \in J \tag{4}
\]

\[
X(i, j) = \begin{cases} 0 & \forall i \in I, j \in J \\ 1 & \xi = 2 \\ 2 & \xi = 3 \end{cases} \tag{5}
\]
The constraint (1) describes the objective function of the problem i.e. to minimize the total bulk transportation cost subjected to the constraints. The constraint (2) takes care of the restriction of availability and requirement of the product between sources and destinations i.e. the total availability of the product at the sources is greater than or equal to the total requirement of the product at the destinations. The constraint (3) represents the feasibility criterion that the total capacity available at the sources supplies to different destinations i.e. The total supply of a product to the different destinations from a source should be less than or equals to the availability of such source. The constraint (4) represents the feasibility criterion that the total capacity requirement at the destination gets from different sources i.e. the total supply from different sources of a product to a destination is strictly equals to the requirement of such destination. The constraint (5) describes that if a source ‘i’ does not supply its capacity to the requirement of destination j then \( X(i, j) = 0 \). Otherwise it is not equals to 0. Suppose \( X(i, j) = 2 \) represents that \( i^{th} \) source supplies \( 2a \) bulk product to \( j^{th} \) destination at twice cost. Here source \( i \) supply its bulk product to any destination at most thrice. In the sequel we developed a Lexi-search algorithm based on the “Pattern Recognition Technique” to solve this problem which takes care of simple combinatorial structure of the problem.

5.5. Numerical Illustration:

The concepts and the algorithm developed will be illustrated by a numerical example. In which we have taken number of Sources (S) as 4 and number of Destinations (D) as 6. \( SA \) is the availability of a product at sources and \( DR \) is requirement of a product at the destinations. Let \( a=100 \) be the bulk unit. The total destinations classified into three levels. They are Level 1, Level 2 and Level 3 based on bulk unit \( a \). i.e., Level 1, Level 2 and Level 3 have the requirements of destinations less than or equal to \( a \), \( 2a \), and \( 3a \) respectively. The number of destinations of Level 1, Level 2 and Level 3 called \( p \), \( q \) and \( r \) respectively. In this problem consider \( p = 2 \), \( q = 2 \) and \( r = 2 \). In the above numerical example given in Table – 5.1, \( C(i, j) \), where \( C(i, j) \) are taken as non negative integers it can be easily seen that this is not a necessary condition and the cost can well be any positive quantity. Then the cost array is given below.

[143]
A Variant Bulk Transportation Problem with multiple Bulk Cost Constraint

Table - 5.1

<table>
<thead>
<tr>
<th></th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
<th>D6</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>2</td>
<td>9</td>
<td>18</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>600</td>
</tr>
<tr>
<td>S2</td>
<td>11</td>
<td>5</td>
<td>15</td>
<td>4</td>
<td>12</td>
<td>16</td>
<td>400</td>
</tr>
<tr>
<td>S3</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>13</td>
<td>6</td>
<td>10</td>
<td>400</td>
</tr>
<tr>
<td>S4</td>
<td>7</td>
<td>10</td>
<td>12</td>
<td>5</td>
<td>14</td>
<td>3</td>
<td>500</td>
</tr>
<tr>
<td>DR</td>
<td>180</td>
<td>300</td>
<td>85</td>
<td>270</td>
<td>200</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Suppose \(C(i, j) = 6\) means that the cost of the product supply from source 3 to destination 4 is 6. Moreover, \(SA\) and \(DR\) represent that the availability of sources and requirements of destinations. The problem is to minimize the total bulk transportation cost from given 4 sources to 6 destinations. The source and destination arrays \(SA\) and \(DR\) are given below for our connivance.

Table-5.2

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>600</td>
<td>400</td>
<td>400</td>
<td>500</td>
</tr>
</tbody>
</table>

Table-5.3

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR</td>
<td>180</td>
<td>300</td>
<td>85</td>
<td>270</td>
<td>200</td>
<td>100</td>
</tr>
</tbody>
</table>

From the above tables 5.2 & 5.3, \(SA(i) = \alpha\) means that the availability of source \(i\) is \(\alpha\) and \(DR(j) = \theta\) means that the requirement of destination \(j\) is \(\theta\). Here \(SA(2) = 400\) means that the availability of source 2 is 400 and \(DR(4) = 270\) means that the requirement of destination is 270.

5.6. Concepts and Definitions:

An indicator two-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. Assignment of the pattern can be represented by an approximate \(m \times n\) indicator array \(X = \{x(i, j); x(i, j) = 0\text{ or } 1\text{ or } 2\text{ or } 3\}\) in which if a source \(i\) does not supply to the requirement of destination \(j\) then \(X(i, j) = 0\). Suppose \(X(i, j) = 2\) represents that \(i^{th}\) source supplies its availability bulk product to \(j^{th}\) destination \(2 \times 100 = 200\).
5.6.1. Feasible Solution:

Consider an ordered pair set \{(1,1), (1,2), (2,2), (2,4), (3,2), (3,3), (3,5), (4,6)\} represents the pattern given in Table - 5.4. In which (1,2), (2,2), (3,2), (3,3), (4,6) give Level 1, (3,5) and (2,4) give the Level 2 and Level 3 respectively. i.e., suppose \(X(2,4)=3\) means from source 2 the supply of products to destination 4 is \(3 \times 100 = 300\). \(X\) is feasible solution.

Table - 5.4

\[
X(i,j) = \begin{bmatrix}
2 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 3 & 0 \\
0 & 1 & 1 & 0 & 2 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

The above solution destination 1 gets its requirement of product from source 1 at two bulk units, destination 2 gets its requirement of product from the sources 1, 2 and 3 at one bulk unit from each respectively, destination 3 gets its requirement of product from source 3 at once, destination 4 gets its requirement of product from source 2 at thrice, destination 5 gets its requirement of product from source 3 at twice and destination 6 gets its requirement of product from source 4 at once. Here the all destinations satisfy with the respective requirements by the above allocation. So the solution gives a feasible solution and the following figure - 5.2 represents the feasible allocation.

As earlier mentioned in the figures 5.2 and 5.3, shape of cylinder represents the sources and the boxes of rectangular shapes represent destinations. The values at cylinders represent the capacity of source availability and the values at boxes represent the requirements of destinations. The values at arrows indicates that the number of bulk units supply from source to destinations and the values of parenthesis indicates the bulk cost from source to destination per bulk unit. Then the figure - 5.2 represent the feasible solution as follows

[145]
In the above allocation, source 3 supplies its product one bulk unit to destination 2, one for destination 3 and 2 units for destination 5 at bulk cost per bulk unit 8, 1 & 6 respectively. The all sources supply as same manner shown in the above figure -5.2. Destination 4 gets its total requirement of 3 bulk units at 4 per unit of bulk cost from second source only and also remaining destinations get its requirements as in above figure - 5.2. The cost calculated can be done as follows. Suppose the destination 1 gets its product from source 1 at twice (i.e., 2×100=200). Then the transportation cost of this allocation is 2×C(1,1)×2×2=4. Similarly calculate the transportation costs of remaining all destinations. Then the total bulk transportation cost of the above allocation (figure -5.2) is as follows.

Sources supply to destinations in three different ways. In the above solution, number of allocations at level 1 (p) are 5, number of allocations at level 2 (q) are 2, number of allocations at level 3 (r) are 1. We calculate the total transportation cost (TP) based on these levels. i.e., the total transportation cost is sum of the costs of these three levels.

Total cost =\[\text{level } 1\text{ TP cost}\]+\[\text{level } 2\text{ TP cost}\]+\[\text{level } 3\text{ TP cost}\]

=\[(C(1,2)+C(2,2)+C(3,2)+C(3,3)+C(4,6))+2(C(1,1)+C(3,5))+3(C(2,4))\]
5.6.2. Infeasible Solution:

Consider an ordered pair set \((1, 1), (1, 2), (2, 2), (3, 4), (3, 5), (4, 3)\) and \((4, 6)\) represents the pattern given in Table - 5.5, in which \((1,1), (4,3), (4,6)\) give Level 1 \((1,2), (4,5)\) and \((2,2), (4,4)\) give the Level 2 and Level 3 respectively, this gives a infeasible solution.

Table-5.5

\[
X(i,j) = \begin{bmatrix}
1 & 2 & 0 & 0 & 0 & 0 \\
0 & 3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 3 & 2 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 \\
\end{bmatrix}
\]

In the above figure 5.3, Destination 2 gets the amount of product from source 2 at thrice of bulk unit and from source 1 at twice of bulk unit. It is more than actual requirement of destination 2. Source 3 supplies 3 units for destination 4 and 2 units for destination 5. The total amount of supply is greater than actual amount at source 3. So the solution gives a infeasible
solution and the above figure-5.3 represents an infeasible allocation. The cost calculated can be done as earlier mentioned. Suppose the destination 4 gets its product from source 3 at thrice (i.e., $3 \times 100 = 300$). Then the transportation cost of this allocation is $3 \times C(1,1) = 2 \times 13 = 39$. Similarly calculate the transportation costs of remaining all destinations.

$$\text{Total cost} = I[C(1,1)+C(4,3)+C(4,6)]+2[C(1,2)+C(3,5)]+3[C(2,2)+C(3,4)]$$


5.6.3. Definition of a Pattern:

An indicator two-dimensional array which is associated with an assignment is called a 'pattern'. A Pattern is said to be feasible if X is a solution.

$$V(X) = \sum_i \sum_j C(i,j)X(i,j)$$

The value $V(X)$ is gives the total cost of the tour for the solution represented by X. The pattern represented in the table – 5.4 is a feasible pattern. The value $V(X)$ gives the total transportation cost for the solution represented by X. Thus the value of the feasible pattern gives the total cost represented by it. In the algorithm, which is developed in the sequel, a search is made for a feasible pattern with the least value. Each pattern of the solution X is represented by the set of ordered pairs $[(i,j)]$ for which $X(i,j) = 1, 2, 3$ with understanding that the other $X(i,j)'s$ are zeros. The ordered pair set $\{(1, 1), (1, 2), (2, 2), (2, 4), (3, 2), (3, 3), (3, 5) and (4, 6)\}$ represents the pattern given in table -5.4, which is feasible and the ordered pair set $\{(1, 1), (1, 2), (2, 2), (3, 4), (3, 5), (4, 3) and (4, 6)\}$ represents the pattern given in the table – 5.5, which is an infeasible solution.

There are $M = m \times n$ ordered pairs in the two-dimensional array X. For convenience these are arranged in ascending order of their corresponding distance and are indexed from 1 to M (Sundara Murthy-1979). Let $SN=\{1, 2, 3, \ldots, m \times n\}$ be the set of $m \times n$ indices. Let $D$ be the corresponding array of cost. If $a, b \in SN$ and $a < b$ then $D(a) \leq D(b)$ Also let the arrays $R, C$ (for convenience same notation) be the array of row and column indices of the ordered pair represented by $SN$. The arrays $SN, D, R, and C$ for the numerical example are given in the table-5.6. If $a \in SN$ then $(R(p), C(p))$ is the ordered pair and $D(a) = C(R(a), C(a))$ is the value of the ordered pair and $CD(a) = \sum_{i=1}^{a} D(i)$
From the above table – 5.6, let us consider 11 ∈ SN. It represents the ordered pair \( D(R(11), C(11)) = (3, 2) \). Then \( D(11) = C(3, 2) = 8 \), i.e. the transportation cost from source 3 to destination 2 with one bulk unit is 8 and \( CD(11) = 51 \).
5.6.4. Value of the Word:

The value of the (partial) word \( L_0 \), \( V (L_0) \) is defined recursively as

\[
V (L_0) = V (L_{0,1}) + k \cdot D (a_0) \quad \text{with} \quad V (L_0) = 0 \quad \text{where} \quad D (a_0) \quad \text{is the cost array arranged such that} \quad D (a_0) < D (a_{0+1}) \quad \text{for} \quad V (L_0) \quad \text{and} \quad V(x) \quad \text{the values of the pattern} \ X \quad \text{will be the same. Since} \ X \quad \text{is the (partial) pattern represented by} \ L_0, \quad \text{(Sundara Murthy - 1979).}
\]

For example the word \( L_3 = (1, 5, 8) \) then value of \( L_3 \) is \( V (L_3) = V (L_2) + k \cdot D (a_3) \). In this partial word \( a_3 = 8 \) represents serial number of 8 and corresponding source and destinations are 3 and 5 respectively. since source 3 can supply its availability to destination 5 in two bulk units. i.e., \( k = 2 \).

Therefore \( V (L_3) = 5 + 2 \cdot D(8) = 5 + 2 \cdot 6 = 5 + 12 = 17, \) where \( V (L_2) = 5 \).

5.6.5. Lower Bound of a Partial Word \( LB (L_0) \):

A lower bound \( LB (L_0) \) for the values of the block of words represented by \( L_0 \) can be defined as follows

\[
LB (L_0) = V(L_0) + 3(CD(\theta+r)-CD(\theta)) + 2(CD(\theta+r+q)-CD(\theta+r)) + (CD(\theta+r+q+p)-CD(\theta+r+q))
\]

For example the word \( L_3 = (1, 5, 8) \) then value of \( L_3 \) is \( V (L_3) = 17, \) \( p = 2, \) \( q = 1 \) and \( r = 1 \)

Then the lower bound of \( L_3 \) is

\[
LB (L_3) = V(L_3) + 3(CD(8+1)-CD(8)) + 2(CD(8+1+1)-CD(8+1)) + 1(CD(8+1+1+2)-CD(8+1+1))
\]

\[
= 17 + 3(36-29) + 2(43-36) + 1(60-43)
\]

\[
= 17 + 3(7) + 2(7) + 1(17) = 17 + 21 + 14 + 17 = 69.
\]

5.6.6. Feasibility Criterion of a Partial Word:

A feasibility criterion is developed, in order to check the feasibility of a partial word \( L_{0+1} = (a_1, a_2, \ldots, a_0, a_{0+1}) \) given that \( L_0 \) is a feasible word. We will introduce some more notations which will be useful in the sequel.

\textbf{L}: An array where \( L (i) \) is the letter in \( i^{\text{th}} \) position of the word.

The values of the arrays \( DR, SA \) and \( L \) are as follows

\[
L (i) = a_i, \quad i = 1, 2, \ldots, 0, \quad \text{and} \quad L (j) = 0, \quad \text{for other elements of} \ j
\]
For example consider a sensible partial word $L_8 = (1, 2, 4, 5, 6, 8, 11$ and 12) which is feasible. The array DR, SA and L takes the values represented in table - 5.5 given below.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>15</td>
<td>400</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

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 $RA = R(a_{p+1})$ and $CA = C(a_{p+1})$.

5.7. Algorithms:

5.7.1. Algorithm 1 (checking for feasible):

<table>
<thead>
<tr>
<th>STEP</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IX = 0</td>
</tr>
<tr>
<td>2</td>
<td>GO TO 2</td>
</tr>
<tr>
<td>3</td>
<td>IS DR (CA) = 0</td>
</tr>
<tr>
<td>4</td>
<td>IF YES GO TO 5</td>
</tr>
<tr>
<td>5</td>
<td>IF NO GO TO 3</td>
</tr>
<tr>
<td>6</td>
<td>IS SA (RA) = 0</td>
</tr>
<tr>
<td>7</td>
<td>IF YES GO TO 5</td>
</tr>
<tr>
<td>8</td>
<td>IF NO GO TO 4</td>
</tr>
<tr>
<td>9</td>
<td>IX = 1</td>
</tr>
<tr>
<td>10</td>
<td>GOTO 5</td>
</tr>
<tr>
<td>11</td>
<td>STOP</td>
</tr>
</tbody>
</table>

We start with the partial word $L_1 = (a_1) = (1)$. A partial word $L_9$ is constructed as $L_9 = L_{9-1} \ast (a_9)$. Where * indicates chain formulation. We will calculate the values of $V(L_9)$ and $LB(L_9)$ simultaneously. Then two situations arises one for branching and other for continuing the search.

1. $LB(L_9) < VT$. Then we check whether $L_9$ is feasible or not. If it is feasible we proceed to consider a partial word of under $(9+1)$ which represents a sub-block of the block of words represented by $L_9$? If $L_9$ is not feasible then consider the next partial word $p$ by
taking another letter which succeeds a₀ in the position. If all the words of order p are
exhausted then we consider the next partial word of order (θ-1).

2. LB (L₀) ≥ VT. In this case we reject the partial word L₀. We reject the block of word with
L₀ as leader as not having optimum feasible solution and also reject all partial words of
order p that succeeds L₀.

Now a Lexi-search algorithm to find an optimal feasible word is developed along
with the flow chart for easy understanding.

5.7.2. Algorithm 2 (Lexi-search Calculation):

STEP0 : Initialization

The arrays SN, R, C, D, α, P1, Q1, R1, SA, DR and values of N are made
available. The values I=1, J=0, α, VT=999 and Max= m×n – n

STEP1 : p=P1,q=Q1,r=R1 GOTO2

STEP 2 : J = J+1

IS (J>Max) IF YES GOTO 21

IF NO GOTO 3

STEP 3 : L (I) = J

RA = R (J)

CA = C (J) GOTO 4

STEP4 : Check feasible (Using Algorithm 1)

IS (IX=0) IF YES k = 0 GO TO 2

IF NO GO TO 5

STEP5 : IS SA (RA)≥DR (CA) IF YES CV=DR(CA),BR= 0

GOTO8

IF NO CV=SA(RA),BR=DR(CA)-CV

GOTO 6

STEP6 : IS BR ≤ α IF YES p = p+1 GOTO 8

IF NO GO TO 7

STEP7 : IS BR ≤ 2α IF YES q = q+1 GOTO 8

IF NO r = r+1GO TO 8
A Variant Bulk Transportation Problem with multiple Bulk Cost Constraint

STEP 8: \( \text{IS } DR(CA) \leq \alpha \)  
  IF YES \( p = p-1 \) GOTO 10  
  IF NO GOTO 9

STEP 9: \( \text{IS } TR \leq 2\alpha \)  
  IF YES \( q = q-1 \) GOTO 10  
  IF NO \( r = r+1 \) GOTO 10

STEP 10: \( \text{IS } CV \leq \alpha \)  
  IF YES \( k = 1 \) GOTO 12  
  IF NO GOTO 11

STEP 11: \( \text{IS } CV \leq 2\alpha \)  
  IF YES \( k = 2 \) GOTO 12  
  IF NO \( k = 3 \) GOTO 12

STEP 12: \( V(I) = V(I-1) + k \times D(J) \)  
  \( LB(I) = V(I) + 3(CD(J+r)-CD(0)) + 2(CD(J+r+q)-CD(J+r)) \)  
  \( + (CD(\theta + r + q + p) - CD(0 + r + q)) \) GOTO 13

STEP 13: \( \text{IS } LB(I) \geq VT \)  
  IF YES GOTO 22  
  IF NO \( \text{DR}(CA) = DR(CA) - CV \) GOTO 14

STEP 14: \( \text{IS } DR(i) = 0, \text{do } i=1,2,...n \)  
  IF YES GOTO 16  
  IF NO GOTO 15

STEP 15: \( L(I) = J \)  
  \( \text{SA}(RA) = \text{SA}(RA) - CV \)  
  \( CV1(I) = CV \)  
  \( KP(I) = p, KQ(I) = q, KR(I) = r \)  
  \( I = I + 1 \) GOTO 2

STEP 16: \( \text{VT} = V(I), L(I) = J, L(I) \) is full length word and is feasible  
  Record \( L(I) \) and \( VT \)  
  \( \text{DR}(CA) = \text{DR}(CA) + CV \) GOTO 17

STEP 17: \( I = I - 1 \) GOTO 18

STEP 18: \( J = L(I) \)  
  \( RA = R(J) \)  
  \( CA = C(J) \)  
  \( \text{SA}(RA) = \text{SA}(RA) + CV1(I) \)  
  \( \text{DR}(CA) = \text{DR}(CA) + CV1(I) \)  
  \( \text{IS } (I = 1) \) IF YES GOTO 1

{153]
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.

5.8. Search Table:

The working details of getting an optimal word using the above algorithm for the illustrative numerical example is given in the following table-5.8. The columns named (1), (2), (3), (4), (5), (6), (7), (8)and (9) gives the letters in the first, second, third, fourth, fifth, sixth, seventh, eighth and ninth places of a word respectively, the corresponding value and lower bound of the partial word indicated V and LB, represented in the next two columns. The next columns R and C give the row and column indices of the letter respectively. The last column gives the remarks regarding the acceptability of the partial words. In the following Table-5.8, A means Acceptance and R means Rejectance of the partial word. In Search Table -5.8, fifth row we have 5-3 represent that the respective ordered pair (2, 4) is at level 3. i.e., destination 4 gets 3α bulk units from source 2.

[154]
A Variant Bulk Transportation Problem with Multiple Bulk Cost Constraint

**Search Table 5.8**

<table>
<thead>
<tr>
<th>SN</th>
<th>Search Procedure</th>
<th>R</th>
<th>C</th>
<th>V</th>
<th>LB</th>
<th>REM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>35</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>2-2</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>36</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>44</td>
<td>R</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>45</td>
<td>A</td>
</tr>
<tr>
<td>5</td>
<td>5-3</td>
<td>2</td>
<td>4</td>
<td>20</td>
<td>45</td>
<td>A</td>
</tr>
<tr>
<td>6</td>
<td>6-2</td>
<td>2</td>
<td>2</td>
<td>30</td>
<td>52</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>37</td>
<td>51</td>
<td>R</td>
</tr>
<tr>
<td>8</td>
<td>8-2</td>
<td>3</td>
<td>5</td>
<td>42</td>
<td>56</td>
<td>A</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>51</td>
<td>51</td>
<td>R</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>1</td>
<td>5</td>
<td>51</td>
<td>53</td>
<td>A</td>
</tr>
<tr>
<td>11</td>
<td>11-2</td>
<td>3</td>
<td>2</td>
<td>58</td>
<td>67</td>
<td>A</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>67</td>
<td>67</td>
<td>VT</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>3</td>
<td>2</td>
<td>50</td>
<td>59</td>
<td>A</td>
</tr>
<tr>
<td>14</td>
<td>12</td>
<td>59</td>
<td>59</td>
<td>VT</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>12</td>
<td>1</td>
<td>2</td>
<td>60</td>
<td>60</td>
<td>R,≥VT</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>36</td>
<td>57</td>
<td>A</td>
</tr>
<tr>
<td>17</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>43</td>
<td>57</td>
<td>A</td>
</tr>
<tr>
<td>18</td>
<td>10</td>
<td>4</td>
<td>1</td>
<td>43</td>
<td>-</td>
<td>R</td>
</tr>
<tr>
<td>19</td>
<td>11</td>
<td>3</td>
<td>2</td>
<td>59</td>
<td>-</td>
<td>R</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>4</td>
<td>1</td>
<td>43</td>
<td>-</td>
<td>R</td>
</tr>
<tr>
<td>21</td>
<td>11-2</td>
<td>3</td>
<td>2</td>
<td>52</td>
<td>61</td>
<td>R,≥VT</td>
</tr>
<tr>
<td>22</td>
<td>9-2</td>
<td>1</td>
<td>5</td>
<td>44</td>
<td>58</td>
<td>A</td>
</tr>
<tr>
<td>23</td>
<td>10</td>
<td>4</td>
<td>1</td>
<td>51</td>
<td>-</td>
<td>R</td>
</tr>
<tr>
<td>24</td>
<td>11-2</td>
<td>3</td>
<td>2</td>
<td>60</td>
<td>60</td>
<td>R,≥VT</td>
</tr>
<tr>
<td>25</td>
<td>9</td>
<td>1</td>
<td>3</td>
<td>37</td>
<td>59</td>
<td>R,≥VT</td>
</tr>
<tr>
<td>26</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>25</td>
<td>47</td>
<td>A</td>
</tr>
<tr>
<td>27</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>29</td>
<td>-</td>
<td>R</td>
</tr>
<tr>
<td>28</td>
<td>8-2</td>
<td>3</td>
<td>5</td>
<td>37</td>
<td>51</td>
<td>A</td>
</tr>
<tr>
<td>29</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>44</td>
<td>-</td>
<td>R</td>
</tr>
<tr>
<td>30</td>
<td>10</td>
<td>4</td>
<td>1</td>
<td>44</td>
<td>-</td>
<td>R</td>
</tr>
</tbody>
</table>

[155]
|   | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|   | 11-2 | 3 | 2 | 53 | 62 | $R \geq VT$ | 8 | 3 | 5 | 31 | 52 | $A$ | 9 | 1 | 5 | 38 | 52 | $A$ | 10 | 4 | 1 | 45 | - | $R$ | 11-2 | 3 | 2 | 54 | 54 $= VT$ | $A$ | 10 | 4 | 1 | 38 | - | $R$ | 11-2 | 3 | 2 | 47 | 56 | $R \geq VT$ | 9-2 | 1 | 5 | 39 | 53 | $A$ | 10 | 4 | 1 | 46 | - | $R$ | 11-2 | 3 | 2 | 55 | 55 | $R \geq VT$ | 9 | 1 | 5 | 32 | 54 | $R \geq VT$ | 10 | 4 | 1 | 39 | - | $R$ | 11-2 | 3 | 2 | 48 | 66 | $R \geq VT$ | 8 | 3 | 5 | 26 | 54 | $R \geq VT$ | 2 | 4 | 16 | 47 | $A$ | 2 | 2 | 26 | 49 | $A$ | 7 | 4 | 4 | 31 | 50 | $A$ | 8-2 | 3 | 5 | 43 | 50 | $A$ | 9 | 1 | 5 | 50 | - | $R$ | 10 | 4 | 1 | 50 | - | $R$ | 11 | 3 | 2 | 51 | 51 $= VT$ | $A$ | 8 | 3 | 5 | 37 | 51 | $R \geq VT$ | 8-2 | 3 | 5 | 38 | 52 | $R \geq VT$ | 6 | 2 | 2 | 21 | 50 | $A$ | 7 | 4 | 4 | 26 | 52 | $R \geq VT$ | 7 | 4 | 4 | 21 | 53 | $R \geq VT$ | 5 | 2 | 4 | 12 | 49 | $A$ | 6-3 | 2 | 2 | 27 | 49 | $A$ | 7-2 | 4 | 4 | 37 | 49 | $A$ | 8-2 | 3 | 5 | 49 | 49 $= VT$ | $A$ | 156 |
The shaded rows in the above table – 5.8 gives optimal solution of the taken numerical example and at the end of the search the current value of VT is 49. Then the partial word is \( L_7 = (1, 2, 4, 5, 6, 7, \text{and } 8) \) is a optimal feasible partial word. It is given in the 38th row of the search table. For this partial word the array L, DR, SA and IK are given in the following Table – 5.9

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>96</td>
<td>3-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>48</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6</td>
<td>9</td>
<td>52</td>
<td></td>
<td>VT</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>52</td>
<td></td>
<td>VT</td>
</tr>
</tbody>
</table>

Table – 5.9

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>420</td>
<td>000</td>
<td>115</td>
<td>230</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

From the above table – 5.9, the pattern \( X \) represented by the above optimal feasible word is given in the following table – 5.10.

\[
X (i, j) = \begin{bmatrix}
2 & 0 & 0 & 0 & 0 & 0 \\
0 & 3 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 2 & 0 \\
0 & 0 & 0 & 2 & 0 & 1
\end{bmatrix}
\]

From the above optimal feasible solution, the pictorial representation of the optimal allocation is given in the following figure – 5.4.
5.9. Computational results:

A Computer program for the proposed Lexi – Search Algorithm is written in C language and is tested. The experiments are carried out on a COMPAQ (dx2280 MT) system by generating the values randomly for Cost Matrix \( C(i, j) \). The values for the availability and requirement of the product are also uniformly randomly generated between \( [1, 1000] \). We tried a set of problems by giving different values to \( m, n, p, q, \) and \( r \). The results are tabulated in the Table-5.9 below. For each instance, five data sets are tested. It is seen that the time required for the search of the optimal solution is fairly less. The following table-5.11 shows the CPU run time taken by the proposed Lexi-Search Algorithm (LSA) to find the optimal solution of various hard instances.
In the above Table-5.1, SN = serial number, m = number of sources, n = number of destinations, p = number of destinations, which have the requirement less than or equals to bulk quantity \( \alpha \), q = number of destinations, which have the requirement lies between \( \alpha \) to \( 2\alpha \) and \( r = number \) of destinations, which have the requirement more than \( 2\alpha \), where \( \alpha \) is bulk quantity (say 100) NPT= number of problems tried, AT = CPU run time for formation of the alphabet table, ST=CPU run time for searching an optimal solution.

The graphical representation of the above tabulated values with proposed algorithm as follows. In the graph X axes taken the SN and Y axes taken the values of CPU run time for getting optimal solution in the graph 5.1
In the above graph 5.1, the bars of different colors indicate CPU run time for formation of alphabet table and search time for optimal solution respectively.

5.10. Comparison Details:

We implement the Pattern Recognition Technique based Lexi Search Algorithm (LSA) using C language for this model. We tested the proposed algorithm by different set of problems and compared the computational results with the published model by Purusotham, S et al. International Journal on Computer Science and Engineering (IJCSIE). In this model $X(i,j) \sim 0$ or 1. The products are multiple products. Then Table-5.12 shows that the comparative results of different sizes.

<table>
<thead>
<tr>
<th>SN</th>
<th>n</th>
<th>NPT</th>
<th>CPU Run Time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Published model</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>6</td>
<td>0.0091</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>6</td>
<td>0.0000</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>6</td>
<td>0.03430</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>6</td>
<td>0.9832</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>6</td>
<td>1.4961</td>
</tr>
</tbody>
</table>
In the above table-5.12, \( n \) = number of destinations, \( NPT \) = number problems tried. The last two columns show the CPU run time of published model and proposed model. As compared the two models of size \( n=25 \) The runtime of this instance with the existing model is 1.4961 Sec., and the proposed model took 1.53846 Sec., it is reasonably close. The present model takes less computational times like the published model. Hence, the present model is also very useful.

The graphical representation of the CPU run time for the two models presented in the above instances is given below. In the Graph-5.2, X axes taken the SN and Y axes taken the values of CPU run time for the published and proposed models.

**Graph-5.2**

![Graph-5.2](image)

From the above Graph-5.2, series1 represent that CPU run time for getting optimal solution by published model and series 2 represent that CPU run time for searching the optimal solution by the proposed model.

**5.11 Conclusion:**

In this chapter, we have studied a problem “A Variant Bulk Transportation Problem with Multiple Bulk Cost Constraint” to minimizing the total transportation cost and
A Vartnt Bulk Transportation Problem with multiple Bulk Cost Constraint

developed a Lexi-search algorithm based on pattern recognition technique to solve it. First the model is formulated into an integer programming problem. A Lexi-Search Algorithm based on Pattern Recognition Technique is developed for getting an optimal solution. The problem is discussed in detail with help of a numerical illustration. We have programmed the proposed algorithm using C-language. The computational details are reported. As an observation the CPU run time is fairly less for higher values to the parameters of the problem to obtain an optimal solution. Moreover, Lexi-search algorithms are proved to be more efficient in many combinatorial problems.