Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 10, Number 4 (2014), pp. 631-646

http://www.ripublication.com

A Variant Constrained Bulk Transshipment Problem

[1]Sangeetham Prasad, [2]*Purusotham S, [1]Suresh Babu C, [1]Sundara Murthy M

[1]Department of Mathematics, Sri Venkateswara University, Tirupati, Andhra Pradesh, India
sangeetham1729@gmail.com, suresh8044@gmail.com, profmurthy@gmail.com
*[2]*Department of Mathematics, Statistics and Operational Research Division, School of Advanced Sciences, Vellore Institute of Technology University, Vellore, Tamil Nadu, India.*
drpurusotham.or@gmail.com

Abstract:-

The usual Transshipment Problem is to minimize the total cost for shipping the various capacities of the goods on the requirement of destinations from the available sources, well known N.P-Hard problem. Generally, the transshipment problem consist a unit cost on supplied goods to destinations from the sources. But in bulk transshipment the cost is independent of number of goods supplied to destinations, it is practical. In this paper we investigated a "variant of constrained bulk transshipment problem". Let there are *m*-sources and *n*-destinations. The destinations can get its complete requirement from a source directly or through some destination. The practical constraint is considered as only fewer destinations will act as transshipment nodes i.e. a destination can get its complete requirement from a source through a permitted destination; this transshipment node can be utilized with the specified finite number of times in the optimal schedule. The cost of transportation of products from the sources to destination and destinations to destination is given. The total availability of the product at the sources is greater than or equal to the total requirement of the product at the destinations. Generally movement of a product from source to source or destinations to source is not natural, hence these possibilities are avoided and movement from destinations to destination is only considered. This is more generalized problem and comes under combinatorial programming problem. Often, the model is expressed as a zero-one programming problem. The objective of the problem is to minimize the total bulk cost of supplying the required products to the destinations with the restriction that any destination should get its supply from one source only, even when it gets from a destination. In the sequel, we have developed a Lexi-

Search Algorithm using Pattern Reorganization Technique to obtain the optimal solution. The concepts and algorithms developed are explained with a suitable example. The algorithm is tested and the experimental results indicate that the hard instance problems also solved in reasonable time.

Keywords:- Transportation problem, Bulk Transshipment Problem, Zero-One Programming, Lexi-Search Algorithm and Pattern Reorganization Technique.

1. Introduction:

The Classical Transportation Problem was first formulated by Hitchcock [1], but Koopmans [2] was the first to notice how graph theory could be used to develop efficient solution techniques. Hence, it is usually called the Hitchcock-Koopmans model. The model often can be built as a linear programming model, an NP-hard problem. There is no restriction on the number of sources which can supply to a given destination. The objective is to minimize the total transportation cost with the assumption that the total capacity of all sources equals the total demand of all destinations. This condition can always be achieved by the introduction of 'dummy' sources or 'dummy' destinations. 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 [3] studied this problem with the additional restriction that a destination should get its supply from one source only, solved with Lexi-Search algorithm using pattern recognition technique, which takes out the drawbacks of Maio and Roveda [4] algorithm with the advantage of the simple combinatorial structure of the problem. Several extensions of transportation models and methods have been subsequently developed.

Foulds and Gibbons [5], discussed the bulk zero-one time-mini-max in depth. Two new algorithms for this model are outlined - one based on branch and bound enumeration and the other on a backtracking technique and commented the merits of the algorithms. Later VanitaVerma and Puri [6] solved the bulk transportation problem with branch and bound method. Prakash et al. [7] studied a cost–time trade-off bulk transportation problem with the objectives to minimize the total cost and duration of bulk transportation without according priorities to them and the entire requirement of each destination is to be met from one source only; however a source can supply to any number of destinations subject to the availability of the commodity. Anju Gupta et al. [8] also presented an algorithm to solve the Time-cost Trade-off-Relations in Bulk Transportation Problem.

More recently, Naganna [9] introduced a three dimensional time dependent Bulk transportation problem with the objective is to supply the requirements of the destinations with a minimum cost subjected to the conditions. A variation to the above problem is that if a plant is producing different commodities instead of single commodity and the objective is to shipping all the commodities to warehouses with the minimum bulk cost then the problem becomes the multi-commodity bulk

transportation problem studied by SobanBabu and Sundara Murthy [10]. Purusotham and Sundara Murthy [11] studied a Multi-Product Bulk Transportation Problem (MPBTP) where one requests to get the requirement of different products depending on the availability from any source such that the total cost of the bulk transportation to meet the demands of all products specified over the planning subject of various warehouses is minimized. Vidhyullatha [12] presented three dimensional time minimization bulk transportation problem to minimize the total time of transportation. All the above variants are solved with the pattern recognition based Lexi-Search algorithm.

One of the other important variants of the transportation models is a Transshipment model. Transshipment problem is a real life transportation problem which is quite widely used for planning bulk distribution. In Transshipment problem, the objective is to minimize the total cost of the shipments, and thus the shipment passes through one or more intermediated nodes before it reaches its desired destination. Due to this advantage, the total transportation cost in the transshipment problem becomes less than the classical total transportation cost. Both the transportation problem and the transshipment problem are also quite widely used for planning bulk distribution, especially where the (road) distances travelled are large. Transshipments provide an effective mechanism for correcting discrepancies between the locations' observed demand and their available inventory. As a result, transshipments may lead to cost reductions and improved service (through higher flexibility and responsiveness) in the supply chain without increasing system-wide inventories (see, e.g., Lee et al. [13] and Guinet [14], Khurana and Arora [15]). Amit Kumar et al. [16] suggested two new methods to find the fuzzy optimal solution of fuzzy transportation problems with some additional transshipment. The next section describes the mathematical model of the proposed model.

2. Mathematical formulation:

Let there are *'m'(S= {1, 2, 3, ..., m})*sources and *'n'(D = {1, 2, 3, ..., n})* destinations. We have considered that each source can supply its available capacity of the product to any destination subject to conditions. $SA\ (i) = A_i$,and$DR\ (j) = R_j$, be the amount of availability and requirement of i^{th}source and j^{th}requirement respectively. Let $D^1(\subset D) = \{1, 2, 3, \ldots, k\}$,where$k < n$be the sub set of destinations, which act as transshipment nodes to supply the product to other destinations subject to availability of the product at sources. Let S1 be the number of effective sources including the transshipment nodes (D^1), because the supply of the product from source to destination can pass through these transshipment nodes, i.e. $S^1=SUD^1= \{1, 2, 3,\ldots, m^1\}$, here $m^1= m + k$. The cost of bulk transshipment from source i to the destination j is denoted by $C\ (i, j);\ i \in S1,\ j \in D$.The objective is to minimize the total bulk transshipment cost subjected to the availability and requirements. The model can be built as Zero-One programming problem and its formulation is given below.

$$\text{Minimize } Z = \sum_{i \in s^1} \sum_{j \in D} C(i, j).X(i, j) \qquad \ldots\ldots\ldots\ldots\ldots (1)$$

Subject to the constraints:

$$\sum_{i \in s^1} X(i, j) = 1, \; j \in D \qquad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

$$\sum_{i \in s^1} \sum_{j \in D} X(i, j) = n \qquad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

$$\sum_{i=1}^{m} SA(i) \geq \sum_{j=1}^{n} DR(j) \qquad \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

$$X(i, j) = 0 \; or \; 1; \forall i \in I, \forall j \in J \qquad \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

The constraint (1) describes the minimization of the total bulk transshipment costsubjected to the constraints. The constraint (2) represents that a destination should get itscomplete requirement exactly once (i.e., from a source or via shipment node). The constraint (3)indicates the supply schedule to the n destinations. The constraint (4) represents that the sum ofthe requirements at different destinations should less than or equal to the sum of availabilities atvarious sources. The last constraint (5) indicates that if there is a transportation from i to j then*X(i, j) = 1,* Otherwise it is equals to 0. In the sequel, we developed a Lexi-Search algorithm usingPattern Recognition Technique to solve this problem which takes care of simple combinatorialstructure of the problem.

3. Pattern Recognition Technique based Lexicographic Search Approach:

Pandit [17] was developed a Lexicographic Search Approach in the context of solving the loading problem, it is a systematized Branch and Bound approach. The existence of Lexi-search came into the light before the branch-bound (Little et al. [18]). This approach has been found to be productive in many of the Combinatorial Programming Problems. Gupta [19], ShilaGhose [20], Shila Das [21], Sundara Murthy [22], and Arora and Puri [23] are some of the early applications. It is significance mentioning that Branch and Bound can be viewed as a particular case of Lexicographic Search approach (Pandit [24]). The search mechanism in the algorithm is done in a systematic manner like one search the meaning of a word in a dictionary. This approach is based on the following grounds [17].

(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.

Alphabet Table:

The search process of the algorithm is based on the choice of a suitable Alphabet Table (*Table - 3*); it is constructed in such a way that the *M (= m1 x n)*elements of

cost matrix is arranged in ascending order with their respective indices and are indexed from 1 to M. Here two situations of the search list have to be taken into consideration (Sundara Murthy [22]).

(i) The process of checking the feasibility of a partial word is easy, while the calculations of a lower bound is bulky and

(ii) Computation of lower bound is easy, while the feasibility checking is difficult.

When the process of checking feasibility of a partial word becomes complex and setting the effective bounds are easy, a modified Lexi Search i.e. Lexi Search with recognizing the Pattern of the Solution known as Pattern Recognition Technique (PRT) [22] 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 PRT 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".

In general, Lexi Search algorithm takes the less memory due to the existence of the Lexicographic order of the partial words and PRT reduces the dimensions. For instance, in the three dimensional assignment problem one requires to find an optimal solution X which is again a three dimensional array. If Pattern Recognition is used, 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 (Ramana and Umashankar [25], Purusotham et al. [26], Vidhyullatha [12]).

3.1 Definitions:

An indicator two dimensional arrayX which is associated with Boolean elements is called a "pattern". If X is feasible solution for a given problem then the pattern is called a feasiblepattern. This pattern can be represented by an appropriate m x n indicator array $X = \{X(i, j)/ X(i,j)= 0 \text{ or } 1\}$. A feasible solution is a solution which satisfies all the conditions of the problem.

The proposed problem requires n ordered pairs in the optimal solution. Consider a set of symbols $A = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ 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 Table -3. If $k = n$the word is called a complete word, if $k < n$, the word is called partial word of length k. Every feasible complete word will provide a solution. Searching for an optimum word is a problem of finding the word of minimum value (for minimization problem) in the Lexi Search defined by the solution of the problem.

4. Numerical Illustration:

The concepts and the algorithm is developed will be illustrated by a suitable numerical example. For which we have taken the number of sources as *m = 4 (S = {1, 2, 3, 4})*and the number of destinations as *n = 7 (D= {1, 2, 3, 4, 5, 6, 7}).* Let D^1*= {1, 4}* be the set of transshipment nodes, and at most two destinations are permitted to get its requirement from a source through the same transshipment node, $D^1 \subset D$. Let S^1be the number of sources including transshipment nodes i.e. $S^1 = SUD^1 = \{1, 2, 3, 4, 5, 6\}$, here $S^1(5)=D^1(1)$ and *S1(6)=D1(2).* Let *SA(i)*is the availability of a product at source *i*and *DR(j)*is the requirement of a product at the destinations *j*. Then we consider the random cost matrix C as given in Table-1. From Table-1 *C(3,4) = 5* refers that the cost of the transportation from source 3 to destination 4. Here, *C (i, j)'s*randomly taken as non-negative integers but it is not a necessary condition and the cost can beconsidered any positive real quantity.

Table-1: Cost Matrix

C(i, j) =

	1	2	3	4	5	6	7	SA(i)
1	1	17	24	21	30	41	8	120
2	18	3	12	11	47	16	21	100
3	2	1	20	5	15	7	44	90
4	6	4	17	28	39	32	2	80
5	∞	19	5	23	4	54	59	-
6	27	13	49	∞	50	6	3	-
DR(j)	40	30	35	45	30	45	50	

4.1 Feasible Solution:

Consider an ordered pair set *{(1,1), (3,2), (4,7), (5,5), (3,4), (5,3), (2,6)}*represents the pattern given in Table –2, which is a feasible solution for the given problem. This feasible schedule shows that, destination 1 getting its requirement from source 1, destinations 2 and 4 getting its requirement from the source 3, destinations 3 and 5 are getting its requirement from source 1 via the transshipment node 1 (source 5). Finally, destinations 7 and 6 are getting its requirement from the sources 4 and 2 respectively.

Table –2

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

This schedule satisfies all the constraints involved in the problem. Therefore, the total bulk transshipment cost of this feasible schedule will be computed using the formula $V(X) = \sum_{i \in I} \sum_{j \in J} C(i,, j).X(i, j)$. Here V(X) is the value of the pattern represented by X and

provides the total cost of the schedule.
Total cost = C (1, 1) + C (3, 2) + C (4, 7) + C (5, 5) +
C (3, 4) + C (5, 3) + C (2, 6)
= 1 + 1 + 2 + 4 + 5 + 5 + 16 = 34.

Each pattern of the solution X is represented by the set of ordered pairs {(i, j)} for which X(i, j)=1 with understanding that the other X(i, j)'s are zeros. The alphabet order of the given cost matrix (Table –1) is listed in Table -3.

Table-3: Alphabet Table (AT)

SN	C	DC	IR	IC		SN	C	DC	IR	IC
1	1	1	1	1		22	18	176	2	1
2	1	2	3	2		23	19	195	5	2
3	2	4	3	1		24	20	215	3	3
4	2	6	4	7		25	21	236	1	4
5	3	9	2	2		26	21	257	2	7
6	3	12	6	7		27	23	280	5	4
7	4	16	4	2		28	24	304	1	3
8	4	20	5	5		29	27	331	6	1
9	5	25	3	4		30	28	359	4	4
10	5	30	5	3		31	30	389	1	5
11	6	36	4	1		32	32	421	4	6
12	6	42	6	6		33	39	460	4	5
13	7	49	3	6		34	41	501	1	6
14	8	57	1	7		35	44	545	3	7
15	11	68	2	4		36	47	592	2	5
16	12	80	2	3		37	49	641	6	3
17	13	93	6	2		38	50	691	6	5
18	15	108	3	5		39	54	745	5	6
19	16	124	2	6		40	59	804	5	7
20	17	141	1	2		41	∞	∞	5	1
21	17	158	4	5		42	∞	∞	6	4

In Table-3; SN= {1, 2, 3… M} be the set of $m^1 \times n$ indices. C denotes the array of costelements. $DC(a) = \sum_{i \in I} C(i), a \in SN$, represents the cumulative cost. If a,b∈SN and a < bthenC(a) ≤C(b). Also the arrays IR, IC indicates the row and column indices of the respectivecost element. Let us consider 14∈SN, represents the ordered pair C (IR (14), IC (14)) = (1, 7).Then C(14) = C(1, 7) = 8 refers the transportation cost from source 1 to destination 7 and thecumulative cost DC(14) = 57.

4.2 Effective Bounds of a word:

A lower bound of the partial word L_k denoted by LB(L_k) and is defined as follows
LB (L_k) = V (L_k) + DC (a_k + n- k) – DC (a_k), a_k∈SNand the upper bound of LB (L_k) is

taken as VT a trial value. Here V (L_k) denotes the value of the partial word $L_k$of length 'k'defined as follows.

$$V(L_k) = V(L_{k-1}) + C(a_k) \text{ with } V(Lo) = 0,$$

Consider the partial word $L_5 = (1, 2, 4, 8, 9)$, then from Table -3

$$V(L_5) = C(1) + C(2) + C(4) + C(8) + C(9) = 1 + 1 + 2 + 4 + 5 = 13;$$

And $LB(L5) = V(L_5) + DC(a_5 + 7\text{-}5) - DC(a_5)$

$$= V(L=) + DC(9+2) - DC(9) = 13 + 36 - 25 = 24.$$

5. **Algorithms:**

5.1 Feasibility Criterion of a Partial Word:

In this section a feasibility criterion has been developed to verify the feasibility of a partial word $L_{k+1}=(a_1, a_2 \ldots a_k, a_{k+1})$given that $L_k$is a feasible word. We start with the partial word $L_1= (a_1) = (1)$. A partial word $L_k$is constructed as$L_k = L_{k-1} * (a_k)$. Where '*' indicates the concatenation. We will calculate the values of $V(L_k)$and $LB(L_k)$simultaneously. Then two situations arises one for branching and the other for continuing the search for optimality.

(i) LB (L_k) < VT. Then we check the feasibility of L_k(Algorithm 1). If $L_k$is feasible proceed to consider a partial word of length (k+1). If L_k is not feasible then construct the partial word L_k by taking another letter a_{k+1} which succeeds $a_k$in the k^{th}position of $L_k$and proceed to check the feasibility. Furthermore, if all the words of length k are exhausted then we consider the next partial word of length (k-1).

(ii) LB (L_k) > VT. In this case we reject the partial word $L_k$and concluding that the block of wordwith $L_k$is leader is not having an optimum feasible solution and. Further we reject all partialwords which succeed L_k. Again take the next partial word of length (k-1) for checking theoptimality.

The following notations will be useful in the sequel of algorithm.

CI: An array, if $CI(j) = 1, j \epsilon J = \{1, 2 \ldots n\}$, it indicates that the requirements of j^{th}destination is met from some source, otherwise $CI(j) = 0$.

DR: An array refers the quantity of requirement at destinations.

SA: An array represents the amount of availability at sources.

L: An array where L (*i*) is the letter in i^{th}position of the word.
The recursive algorithm for checking the feasibility of a partial word $L_k$is given in 5.2. In the algorithm initially take $IX = 0$ and proceed to check the feasibility. At the end if $IX = 1$ then the partial word is feasible otherwise it is infeasible. Here $RA = R(a_k)$ and $CA = C(a_k)$.

5.2 Algorithm 1 (checking for feasibility):

STEP0:	IX = 0 GO TO 1	
STEP1:	IS (CI [CA] = 1)	IF YES GO TO 13

		IF NO GO TO 2
STEP2:	IS (RA > m)	IF YES GO TO 3
		IF NO GO TO 8
STEP3:	IS (count [RA] > Q)	IF YES GO TO 13
		IF NO GO TO 4
STEP4:	RD = DRS [RA] + DR [CA]	
	IS (IC [SS [RA]] = 1)	IF YES GO TO 5
		IF NO GO TO 6
STEP5:	IS (RD ≤ (SA [SW [SS [RA]]])	IF YES GO TO 9
		IF NO GO TO 13
STEP6:	DRS [RA] = RD IF YES GO TO 7	
STEP7:	IS (IC [SS [RA]] =0) IF YES	
	{DR [SS [RA]] =DR [SS [RA]] +DRS [RA] GOTO 12}	
		IF NO GO TO 8
STEP8:	IS (DR [CA] ≤ SA [RA])	IF YES GO TO 9
		IF NO GO TO 13
STEP9:	IS (I=n)	IF YES GOTO 12
		IF NO GOTO 10
STEP10:	IS (RA>m)	
	IF YES {SA [SW [SS [RA]]] = SA [SW [SS [RA]]]-DR [CA] GGOTO 12}	
STEP11:	SA [RA] =SA [RA]-DR [CA]	GOTO 12
STEP12:	IX = 1	GOTO 13
STEP13:	STOP	

To find the Optimal Feasible word a Lexi-Search Algorithm using PRT is developed and is given in 5.2

5.2. Algorithm 2 (Lexi-Search Algorithm):

STEP0: Initialization
The arrays SN, CI, R, C, SS, DRS, SW, SA, DR and the values Q, m, n, k are made available. The values I=1, J=0, VT=9999 and Max= m1×n–n

STEP1: J = J + 1		
	IS J > Max	IF YES GOTO 9
		IF NO GOTO 2
STEP2:	V [I] = V [I-1] + C [J]; V [0] = 0	
	LB [I] = V [I] + DC [J + n –I] – DC [J]	
	IS (LB [I] ≥ VT)	IF YES GOTO 9
		IF NO GOTO 3
STEP3:	RA =IR [J], CA = CI [J] GOTO 4	
STEP4:	Check Feasibility using algorithm 1	
	IS (IX = 1)	IF YES GOTO 5
		IF NO GOTO 1
STEP5:	IS (I = n)	IF YES GOTO 8
		IF NO GOTO 6

STEP6: CI [CA] = 1
SW [CA] = RA
L [I] = J
Count [RA] =count [RA] +1 GOTO 7

STEP7:
I = I + 1
Max = Max + 1 GO TO 1

STEP8: VT = V [I], L [I] = J, L [I] is full length word and is feasible
Record L [I] and VT GO TO 10

STEP9: IS (I = 1) IF YES GO TO 12
IF NO GO TO 10

STEP10: I = I – 1;
J = L [I], RA = R [J], CA =C [J];
SW [CA] = 0; Count [RA] =count [RA]-1;
L [J] =0;
IS (RA > m) IF YES 10a;
IF NO {SA [RA] =SA [RA] +DR [CA] GOTO 11}

STEP10a:
IS (SW [SS [RA]] ≠ 0) IF YES
{SA [SW [SS [RA]]] = SA [SW [SS [RA]]] +DR [CA] GOTO 10b}
IF NO GOTO 10b

STEP10b: IS (DRS [RA] ≠ 0)
IF YES {DR [SS [RA]] = DR [SS [RA]] - DRS [RA]
DRS [RA] = DRS [RA] - DR [CA]} GOTO 11

STEP11: SW [CA] = 0 GOTO 1

STEP12: STOP

The current value of VT at the end of the search is the value of the optimal feasible word. At the end if VT = 9999 it indicates that there is no feasible solution.

6. **Search Table:**

The working details of getting an optimal word using the algorithm 5.2 for theillustrative numerical example are given Table -*4*. The columns *(1), (2), (3), (4), (5), (6)* and *(7)*gives the letters in the first, second, third, fourth, fifth, sixth and seventh positions of a wordrespectively. The next two columns V and LB are indicate the value and lower bound of the respective partial word. The columns R and C gives the row and column indices of the letter. Thelast column gives the remarks regarding the acceptability of the partial words (i.e. if a partial wordis feasible word then accept the letter otherwise reject the letter) and here A indicates theacceptance and R for rejectance of the letter in the respective position.

Table – 4: Search Table (ST)

SN	1	2	3	4	5	6	7	V	LB	R	C	Remarks
1	1							1	16	1	1	A
2		2						2	16	3	2	A
3			3					4	16	3	1	R
4			4					4	18	4	7	A
5				5				7	18	2	2	R
6				6				7	20	6	7	R
7				7				8	22	4	2	R
8				8				8	24	5	5	A
9					9			13	24	3	4	A
10						10		18	24	5	3	A
11							11	24	24	4	1	R
12							12	24	24	6	6	R
13							13	25	25	3	6	R
14							14	26	26	1	7	R
15							15	29	29	2	4	R
16							16	30	30	2	3	R
17							17	31	31	6	2	R
18							18	33	33	3	5	R
19							19	34	34	2	6	A, VT=34
20						11		19	25	4	1	R
21						12		19	26	6	6	R
22						13		20	28	3	6	R
23						14		21	32	1	7	R
24						15		24	36	2	4	R, >VT
25					10			13	25	5	3	A
26						11		19	25	4	1	R
27						12		19	26	6	6	A
28							13	26	26	3	6	R
29							14	27	27	1	7	R
30							15	30	30	2	4	A, VT=30
31						13		20	28	3	6	A
32							14	28	28	1	7	R
33							15	31	31	2	4	R, >VT
34						14		21	32	1	7	R, >VT
35					11			14	27	4	1	R
36					12			14	29	6	6	A
37						13		21	29	3	6	R
38						14		22	33	1	7	R, >VT
39					13			15	34	3	6	R, >VT
40				9				9	26	3	4	A
41					10			14	26	5	3	A

42						11		20	26	4	1	R
43						12		20	27	6	6	R
44						13		21	29	3	6	R
45						14		22	33	1	7	R, >VT
46					11			15	28	4	1	R
47					12			15	30	6	6	R, =VT
48				10				9	28	5	3	A
49					11			15	28	4	1	R
50					12			15	30	6	6	R, =VT
-	-	-	-	-	Do	-	-	-	-	-	-	-
85		4						3	22	4	7	A
86			5					6	22	2	2	A
87				6				9	22	6	7	R
88				7				10	24	4	2	R
89				8				10	26	5	5	A
90					9			15	26	3	4	A
91						10		20	26	5	3	A
92							11	26	26	6	6	A, VT=26
93							12	21	27	6	6	R, >VT
94						11		15	27	5	3	R, >VT
95					10			11	28	3	4	R, >VT
96				9				6	24	6	7	R
97			6					7	27	4	2	R, >VT
98			7					4	25	2	2	A
99		5						7	25	3	4	A
-	-	-	-	-	Do	-	-	-	-	-	-	-
132		5						5	26	2	2	R, =VT
133	4							2	27	4	7	R, >VT

In Table -5 the shaded rows represents the optimal solution of the given numerical example. At the end of the search the current value of VT is 26 and it is not improved in the further search and hence 26 is an optimal value. The partial word L7 = (1, 4, 5, 8, 9, 10, 12) is an optimal feasible word. It is given in the 93rd row of the search table. The optimal feasible pattern of L7 is shown in Table -5

Table –5

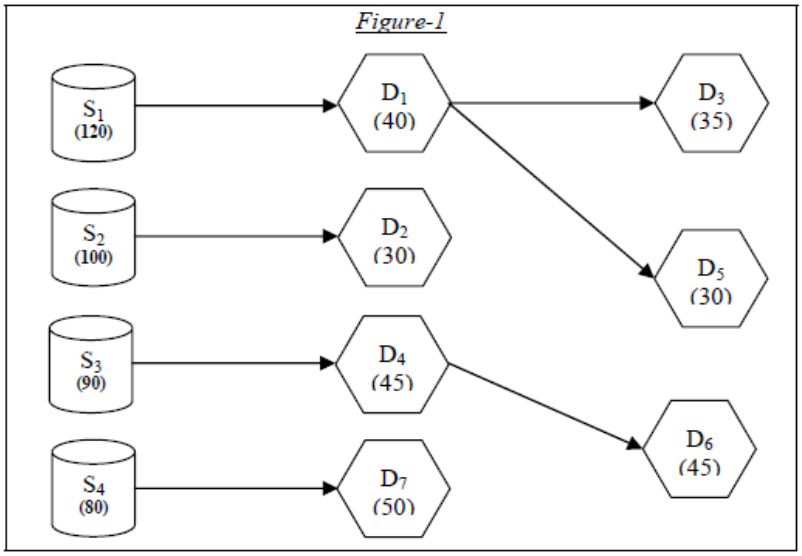
$$X(i,j)=\begin{bmatrix}1&0&0&0&0&0&0\\0&1&0&0&0&0&0\\0&0&0&1&0&0&0\\0&0&0&0&0&0&1\\0&0&1&0&1&0&0\\0&0&0&0&0&1&0\end{bmatrix}$$

Therefore the total bulk transshipment cost of the optimal feasible pattern is obtained by summing the costs involved in the cost matrix in respective non-zero positions of optimal feasible pattern

= C(1, 1)+C(2, 2)+C (3, 4)+C (4, 7)+C (5, 3)+C (5, 5)+C(6, 6)

= 1 + 2 + 3 + 4 + 5 + 5 + 6 = 26

The pictorial representation of the optimal schedule is shown in figure – 1.



7. Computational results:

A Computer program for the proposed Lexi – Search Algorithm is well written in C-language and is tested at various hard instances. The experiments are carried out on a COMPAQ (dx2280 MT) system. The inputs like Cost Matrix C (i, j), source capacities (SA), and destination requirements (DR) are randomly generated for different instances. The cost values are uniformly generated in the interval [1, 100]. The values for the availability and requirement of the product are also randomly generated between [1, 1000]. For different values of m, k, n, and Q a set of problems have been tested and their computational run time is recorded in seconds. The obtained results are tabulated in Table -6. It is observed that, the time required for the search of the optimal solution is fairly less for higher dimensions also.

Table-*6: Computational results*

SN	m	k	n	Q	Trailsolution	OptimalSolution	CPURUNIME(AT+ST)
1	4	2	7	2	200	26	0.000000
2	5	2	8	2	200	58	0.000000
3	5	3	10	2	200	103	0.000000
4	8	2	10	2	200	73	0.000000
5	8	3	12	2	200	123	0.000000
6	10	3	10	3	300	55	0.000000
7	10	5	15	3	300	104	0.109890
8	10	5	20	3	300	147	0.109890
9	15	5	25	3	300	141	0.274725
10	20	3	25	3	300	106	0.274725
11	20	5	30	3	400	113	0.384615
12	25	8	40	3	400	175	0.549445
13	30	5	40	4	400	135	0.604396
14	30	5	50	4	400	187	0.659341

15	40	10	50	4	400	125	0.879121
16	45	8	60	5	500	166	0.824176
17	50	10	55	5	500	109	0.934066
18	55	5	70	5	500	159	1.043956
19	60	10	70	6	500	140	1.153460

In Table-8, SN = serial number, m = number of sources, k = number of destinations which are acting as shipment nodes, n = number of destinations, Q = the number of times a source can supply to the destinations via the same shipment node. AT = CPU run time for formation of the alphabet table, ST=CPU run time for searching an optimal solution.

8. Conclusions:

In this paper an exact algorithm called Lexi-search algorithm using pattern recognition technique is developed to solve the A Variant Constrained Bulk Transshipment Problem. Firstly, the model is formulated into a zero-one programming problem. The problem is having many applications in logistics, transportation GIS, network routings and allied areas. The problem is then illustrated with the help of a suitable numerical example and explained in detail. A well written code has been developed for the proposed algorithm using C-language. Lexi-searchalgorithms are proved to be more efficient in many combinatorial programming problems.The experimental result shows that the CPU run time is reasonably less to obtain an optimalsolution for higher order instances. Hence it is suggested that the algorithm is useful to obtain anoptimal solution of a class of this variant transshipment problems.

9. References:

[1] Hitchcock F. L, 1941: The distribution of a product from several sources to numerous localities, Journal of Mathematics and Physics, 20, 224-230.

[2] Koopmans, Tjalling C, 1949: Optimum utilization of the transportation system, Econometrica17 (Supplement) 136–146.

[3] Sundara Murthy M, 1976: A Bulk Transportation Problem, Opsearch, Vol. 13(3), 143-155.

[4] Maio A. D., Roveda C, 1971: An all zero-one algorithm for a certain class of transportation problems, Operations Research, 19, 1406-1418.

[5] Foulds L. R., Gibbons P. B, 1980: New Algorithms for the Bulk, Zero-One Time Mini-Max Transportation Model, NZOR, Vol. 8 (1), 109-120.

[6] VanitaVerma, Puri M.C, 1996: A branch and bound procedure for cost minimizing bulk transportation problem, Opsearch, Vol. 33(3), 145-161.

[7] Prakash S, Kumar P, Prasad B.V.N.S., Gupta A, 2008: Pareto optimal solutions of a cost–time trade-off bulk transportation problem, European Journal of Operational Research, Vol. 188 (1), 85–100.

[8] Anju Gupta, VanitaVerma, Puri M. C., 1996: Time-cost Trade-off- Relations in Bulk Transportation Problem, Journal of Information and Optimization Sciences, 16(2), 317-325.

[9] Naganna B, 2007: Operations Research, a Ph. D. thesis, Sri Venkateswara University, Tirupati, India.

[10] SobhanBabu K., Sundara Murthy M, 2010: An Efficient Algorithm for Variant Bulk Transportation Problem, International Journal of Engineering Science and Technology, Vol. 2(7), 2595-2600.

[11] Purusotham S., Sundara Murthy M, 2011: An Exact Algorithm for Multi-Product Bulk Transportation Problem, International Journal on Computer Science and Engineering, Vol. 3(9), 3222-3236.

[12] Vidhyullatha A, 2012: Pattern Recognition Lexi – Search Exact Algorithms for Variant ASP and Bulk TP Models, a Ph. D. thesis, Sri Venkateswara University, Tirupati, India.

[13] Lee Y.H, Jung J.W, Jeon Y.S, 2007: An effective lateral transshipment policy to improve service level in the supply chain, International Journal of Production Economics, Vol. 106, 115–126.

[14] Guinet A, 2001: Multi-site planning: a transshipment problem, International Journal ofProduction Economics, Vol. 74, 21–32.

[15] Khurana A., Arora S., (2011): Solving transshipment problems with mixed constraints, International Journal of Management Science and Engineering Management, 6(4):292–297.

[16] Amit Kumar, AmarpreetKaur, Anila Gupta. 2011: New methods for solving fuzzy transportation problems with some additional transshipments, ASOR Bulletin, 30(1), 42-61.

[17] Pandit S.N.N, 1962: The Loading Problem', Operational Research, Vol. 10, 639-646.

[18] Little J. D. C., Murthy K. G., Sweeny D. W, 1963: An algorithm for the TSP, OperationsResearch, Vol. 11, 972-982.

[19] Gupta, J.N.D, 1967: TSP—A survey of theoretical developments and applications. Opsearch, Vol. 5(4), 181–192.

[20] GhoseShila (1971): The Maximum Capacity Routes: A Lexi Search Approach, OperationsResearch, Vol. 8, 209-255.

[21] Shila Das, 1976: Routing and Allied Combinatorial Problems, a Ph.D. Thesis, Dibrugrah University, Dibrugrah Assam, India.

[22] Sundara Murthy M. 1979: Combinatorial Programming - A Pattern Recognition Approach, a Ph. D. Thesis, REC Warangal, India.

[23] Arora S., Puri M.C., (1998): A Variant of Time Minimizing Assignment Problem. EuropeanJournal of Operational Research, 110, 314-325.

[24] Pandit S.N.N. 1963: Some Combinatorial Search Problems, PhD thesis, IIT Kharagpur, India.

[25] VenkataRamana V. V., Umasankar C., 1998: "On a Class of Assignment Problems," Opsearch, Vol. 35(2), 127-138.

[26] Purusotham S., Suresh Babu C., Sundara Murthy M., 2011: Pattern Recognition Technique based Lexi-search Approach to the Multi-Dimensional Assignment Problem, International Journal of Engineering Science and Technology, Vol. 3(8), 6350-6363.