TEST Engineering & Management

March - April 2020
ISSN: 0193-4120 Page No. 10584 - 10595

A New Vision on Solving a Variant Constraint Bulk Transshipment Problem using C-Language

Sangeetham Prasad[1], V. B. V. N. Prasad[2], T.Nageswara Rao[2], G. Balaji Prakash[2], B.Mahaboob[2], M.Sundara Murthy[3]
[1]*S.V.University, Tirupathi.*
[2]*Department of Mathematics, Koneru Lakshmiah Education Foundation, Vaddeswaram, Guntur Dist., A.P. INDIA.*
[3]*Retd. Professor, S.V.University, Tirupathi.*

Article Info
Volume 83
Page Number: 10584 - 10595
Publication Issue:
March - April 2020

Article History
Article Received*: 24 July 2019*
Revised*: 12 September 2019*
Accepted*: 15 February 2020*
Publication*: 13 April 2020*

Abstract: This research article explores on an error-free algorithm entitled by Lexi Search Algorithm (LSA) built on Pattern Recognition Technique (PRT) and its application in solving a Variant Constraint Bulk Transshipment Problem. Lexi Search Algorithms are shown to be dynamic in a large number of combinatorial situations. Here an innovative algorithm called LSA established on PRT has been proposed in order to acquire the optimal solution and the discussions made here are demonstrated by means of numerical examples.Using C- Language the proposed algorithm is designed and it can be observed that this algothim has the ability to execute enormous problems.

Keywords: *Source, Destination, Lexi-Search Approach (LSA), Pattern Recognition Technique (PRT), Distance Matrix, Feasible and Infeasible Solutions, Search Table.*

I. INTRODUCTION

A variant constraint bulk transhipment problem was first formulated by Hitchcock (1941), but Koopmans (1949) was the first to notice how graph theory could be used to develop efficient solution techniques. For the original reference see Hitchcock (1941). Hence, it is usually called the Hitchcock-Koopmans model.. There is no restriction on the number of sources which can supply a given destination. The aim 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.. Several extensions of transportation models and methods have been subsequently developed.

Foulds and Gibbons (1980), 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. Computational experience gained from the use of all these algorithms presented in detailed. Later VanitaVerma and Puri (1996) solved the BTP by BB echnique..Anju Gupta *et al.* (1996) also presented an algorithm to solve the Time-cost Trade-off- Relations in BTP.One of the other important variants of the transportation models is a Transshipment model.

Naganna (2007) proposed a 3D time dependent BTP with objective is to fund the requirements of the destinations with a minimum cost subjected to the conditions. The set of destinations are known and fixed. But there are no sources which are

March - April 2020
ISSN: 0193-4120 Page No. 10584 - 10595

existing, only a set of potential places where sources can be located are known. This set may include some destinations. The problem is to locate some sources in the potential places and supply to the terminuses with the restrictions (i) that a terminus must get its supply from one source only and (ii) the number of sources to be located should not be more than a specified number. Here it is assumed that the sources have an unlimited capacity and this makes the specification of the requirement of the destinations and also the location of more than the one source in a place irrelevant. The destinations themselves are the only set of potential places for locating a source. If a source is located in a destination then it should get its supply from that source only. The numbers of sources to be located are fixed in number.

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 α, as a result C (i, j) will be the bulk cost which is one bulk unit α supplying from source i to destination j. if the destination j requires kα. i.e., k times the bulk unit and it can be supplied from source i subjected to the availability, then the cost is k×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.

In this research article a variant transportation problem called **"A Variant Constrained Bulk Transshipment Problem"is**presented**.** The purpose of this chapter is to propose an efficient Lexi-Search Algorithm using pattern recognition technique for solving "A Variant Constrained Bulk Transshipment Problem" on a scalable multicomputer platform and to obtain an optimal solution. The derived outputs depicted the proposed algorithm is highly competitive on a set of benchmark problems. The remainder of this paper is organized as follows.

II. PROBLEM DESCRIPTION

Let $D^1 = \{\propto_1, \propto_2, \propto_3 \ldots \propto_k\}$ where k<n (i.e.,$D^1 \subset D$) be the sub set of destinations, which supply the product to destinations subject to its availability of product. Let S^1 be a set of effective sources including destinations (D^1) because we allowed supply the product from destination to destination also i.e., $S^1=SUD^1 = \{1, 2, 3,\ldots\ldots\ldots,m^1\}$, where $m^1 = m + k$. If source is i and destination is *j* then cost of bulk transshipment is C (i, j); $i \in S^1$, $j \in D$. The objective is to reduce the TBTC subjected to the obtainability and requirements of conditions. This model can be built as 0 or1 programming problem.

III. MATHEMATICAL FORMULATION

$$Minimize\, Z = \sum_{i \in S^1} \sum_{j \in D} D(i, j) X(i, j) \quad \text{---------------}(1)$$

Subject to the constraints:

$$\sum_{i \in S^1} X(i, j) = 1, \qquad j \in D \quad \text{------------}(2)$$

$$\sum_{i \in S^1} \sum_{j \in D} X(i, j) = n \quad \text{-----------------}(3)$$

$$\sum_{i=1}^{m} SA(i) \geq \sum_{j=1}^{n} DR(j) \quad \text{--------------------}(4)$$

$$X(i, j) = 0 \; or \; 1 \quad \forall i \in I,\; j \in J \quad \text{------------}(5)$$

March - April 2020
ISSN: 0193-4120 Page No. 10584 - 10595

The constraint **(1)** describes the minimization of the total bulk transshipment cost subjected to the constraints. The constraint **(2)** represents that a destination should get its complete 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 of the requirements at different destinations should less than or equal to the sum of availabilities at various sources. The last constraint **(5)** indicates that if there is a transportation from i to j then *X (i, j)* is unity, Else it is equals to zero.

IV. NUMERICAL FORMULATION

Let SA is the availability of a product at sources and DR is the requirement of a product at the destinations. Let D^1 = {1, 4} and S^1 be the number of sources including destinations (D^1) because the transshipment problem allow supplying product from destination to destination also. So the total number of sources will be increase to 6(4+2).

But in this numerical example we consider, only two destinations can supply the product to the destinations subject to its availability of product. Let D^1 = {1, 4} be the set of destinations, which supply the product to the destinations subject to its availability of product. This is subset of set D. Number of sources rises from 4 to 6. Let S^1 be the total number of sources such that $S^1 = SUD^1$ = {1,2,3,4,5,6}.Where $S^1(5)=D^1(1)$ and $S^1(6)=D^1(4)$. Cost matrix D is presented here.. (For convenience same notation D is taken for the distance matrix).

Table- 1

		1	**2**	**3**	**4**	**5**	**6**	**7**	**SA**
D	**1**	1	17	24	21	30	41	8	**120**
(i,j)=	**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	**40**	**30**	**35**	**45**	**30**	**45**	**50**	

Suppose D (3, 4) = 5 means that the cost of the product supply from source 3 to destination 4 is 5. Similarly D (6, 7) = 3 means that the cost of the product supply from source 6 (i.e., destination 4) to destination 7 is 3.

V. FEASIBLE SOLUTION

The collection {(1,1), (3,2), (4,7), (5,5), (3,4), (5,3), (2,6)} represents a feasible allocation and gives the feasible solution.. The values at cylinders represent the capacity of source availability and the values at boxes represent the requirements of destinations. The values at arrows indicate that the bulk cost from the corresponding source i to destination j. Then the **figure – 1** represent the feasible solution as follows.

In the above solution, source 1 supplies its product to destination 1. Source 5 (destination 1) supplies its product to the destinations 5 and 3. Source 3 supplies its product to the destinations 2 and 4.

March - April 2020
ISSN: 0193-4120 Page No. 10584 - 10595

Source 4 supplies its product to the destinations 7 and source 2 supplies its product to the destinations 6. So the solution gives the feasible solution.

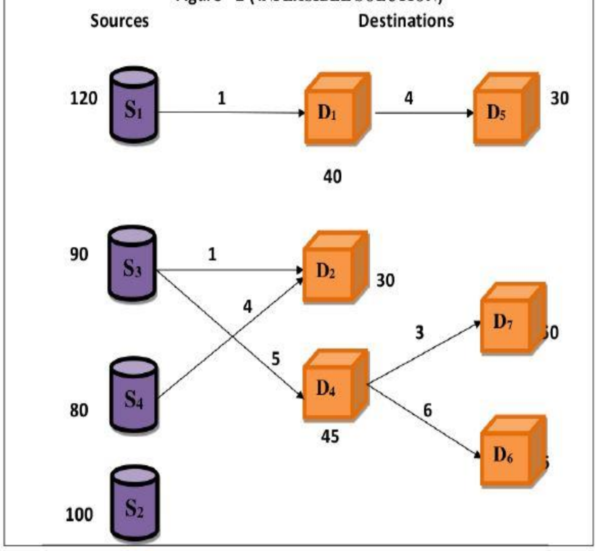
Then the total bulk transshipment cost from given 6 sources to 7 destinations with respective source is as follows.

Total cost = 1 + 1 + 2 + 4 + 5 + 5 + 16 = 34 units.

VI. INFEASIBLE SOLUTION

The collection {(1,1), (3,2), (6,7), (4,2), (5,5), (3,4), (6,6)} represents an infeasible allocation and gives the infeasible solution. From the below **figure-2**, source 1 supplies its product to destination 1. The destination 2 gets its required product from source 3. Source 6 (i.e., destinsation4) supplies its product to destination 7. Source 4 supplies its product to destination 2. But destination 2 already satisfied by source 3. Source 5 (i.e., destination1) satisfies the requirement of destination 5. Source 3 supplies its product to destination 4. From the below **figure-2**, source 1 supplies its product to destination 1. The destination 2 gets its required product from source 3. Source 6 (i.e., destinsation4) supplies its product to destination 7. Source 4 supplies its product to destination 2. But destination 2 already satisfied by source 3. Source 5 (i.e., destination1) satisfies the requirement of destination 5. Source 3 supplies its product to destination 4

Figure –2 (INFEASIBLE SOLUTION)



But the total sum of the requirement of the destinations 2, 4 and 7 is greater to the availability of the source 3, i.e., the total amount of supply is greater than actual amount of availability of source 3. Source 6 (i.e., destination 4) supplies its product to destination 6. Here the all destinations do not satisfy with the respective requirements by the above allocation. So the solution gives the infeasible solution.

Total cost= 1 + 1 + 3 + 4 + 4 + 5 + 6 = 24 units.

VII. ALGORITHMS

Algorithm 1 (checking for feasible)

STEP0: IX = 0 GO TO 1

STEP1: IS (IC [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]

March - April 2020
ISSN: 0193-4120 Page No. 10584 - 10595

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)

{SA [SW [SS [RA]]] = SA [SW [SS [RA]]]-DR
[CA] IF YES GOTO 12}

IF NO GOTO 8

STEP11: SA [RA] =SA [RA]-DR [CA]

GOTO 12

STEP12: IX = 1 GOTO 13

STEP13: STOP

Algorithm 2 (Lexi-search Calculation)

STEP0: Initialization

The arrays SN, IC, R, C, SS, DRS, SW, SA, DR, DC, RA, CA, L, V, LB and values Q, m, n, k are made available. The values I=1, J=0, VT=9999 and Max= m×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 GO TO 9
IF NO GO TO 3

STEP3: RA = IR [J], CA = IC [J] GOTO 4

STEP4: Check Feasibility using algorithm 1

IS (IX = 1) IF YES GO TO 5
IF NO GO TO 1

STEP5: IS (I = n) IF YES GO TO 8
IF NO GO TO 6

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

STEP7: I = I + 1

Max = Max + 1 GOTO 1

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

Record L [I] and VTGO TO 10

TEST Engineering & Management

March - April 2020
ISSN: 0193-4120 Page No. 10584 - 10595

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]

IC [CA] = 0

SW [CA] = 0

Count [RA] =count [RA] -1;

L [J] = 0;

IS (RA > m) IF YES GOTO 11 IF NO {SA [RA] =SA [RA] + DR [CA]
GOTO 11}

STEP11: IS (SW [SS [RA]] ≠ 0) IF YES

{SA [SW [SS [RA]]] = SA [SW [SS [RA]]]
+ DR [CA] GOTO 12}
IF NO GOTO 12

STEPP12: IS (DRS [RA] ≠0)

IF YES {DR [SS [RA]] =DR [SS [RA]] - DRS [RA]
DRS [RA] =DRS [RA] - DR [CA]}
GOTO 11

STEP13: SW [CA] = 0 GOTO 1

STEP14: 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 = 999 it indicates that there is no feasible solution.

VIII. SEARCH TABLE

If a partial word is feasible word then accept the letter otherwise reject the letter) and here A indicates the acceptance and R for rejectance of the letter in the respective position.

Table-6

SN	1	2	3	4	5	6	7	V	LB	R	C	Remark
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

March - April 2020
ISSN: 0193-4120 Page No. 10584 - 10595

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								28	28	1	7	R

							14					
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								15	30	6	6	R,

March - April 2020
ISSN: 0193-4120 Page No. 10584 - 10595

					12							=VT
48				10				9	28	5	3	A
49					11			15	28	4	1	R
50					12			15	30	6	6	R, =VT
51				11				10	31	4	1	R, >VT
52			5					5	21	2	2	R
53			6					5	23	6	7	A
54				7				9	23	4	2	R
55				8				9	25	5	5	A
56					9			14	25	3	4	R
57					10			14	26	5	3	A
58						11		20	26	4	1	R
59						12		20	27	6	6	A
60							13	27	27	3	6	R
61							14	28	28	1	7	R
62							15	31	31	2	4	R, >VT
63						13		21	29	3	6	A
64							14	29	29	1	7	R
65							15	32	32	2	4	R, >VT
66						14		22	33	1	7	R, >VT
67					11			15	28	4	1	R
68					12			15	30	6	6	R, =VT
69				9				10	27	3	4	R
70				10				10	29	5	3	A
71					11			16	29	4	1	R
72					12			16	31	6	6	R, >VT
73				11				11	32	4	1	R, >VT
74			7					6	26	4	2	R
75			8					6	28	5	5	A
76				9				11	28	3	4	A
77					10			16	28	5	3	A
78						11		22	28	4	1	R
79						12		22	29	6	6	R
80						13		23	31	3	6	R, >VT
81					1			17	30	4	1	R, =VT

TEST Engineering & Management

March - April 2020
ISSN: 0193-4120 Page No. 10584 - 10595

					1							
82				10				11	30	5	3	R, =VT
83			9					7	31	3	4	R, >VT
84		3						3	19	3	1	R
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	4	1	R
93							12	26	26	6	6	A, VT=26
94						11		21	27	4	1	R, >VT
95					10			15	27	5	3	R, >VT
96				9				11	28	3	4	R, >VT
97			6					6	24	6	7	R
98			7					7	27	4	2	R, >VT
99		5						4	25	2	2	A
100			6					7	25	6	7	A
101				7				11	25	4	2	R
102				8				11	27	5	5	R, >VT

103			7					8	28	4	2	R, >VT
104		6						4	28	6	7	R, >VT
105	2							1	19	3	2	A
106		3						3	19	3	1	A
107			4					5	19	4	7	A
108				5				8	19	2	2	R
109				6				8	21	6	7	R
110				7				9	23	4	2	R
111				8				9	25	5	5	R
112				9				10	27	3	4	R, >VT
113			5					6	22	2	2	R
114			6					6	24	6	7	A
115				7				10	24	4	2	R
116				8				10	26	5	5	R, =VT
117			7					7	27	4	2	R, >VT
118		4						3	22	4	7	A
119			5					6	22	2	2	R
120			6					6	24	6	7	R
121			7					7	27	4	2	R, >VT
122		5						4	25	2	2	R
123		6						4	28	6	7	R, >VT
124	3							2	23	3	1	A
125		4						4	23	4	7	A
126			5					7	23	2	2	A
127				6				10	23	6	7	R
128				7				11	25	4	2	R

March - April 2020
ISSN: 0193-4120 Page No. 10584 - 10595

129				8				11	27	5	5	R, >VT
130			6					7	25	6	7	R
131			7					8	28	4	2	R, >VT
132		5						5	26	2	2	R, =VT
133	4							2	26	4	7	R, =VT

IX. COMMENTS

In the sixth tablerows shaded are the desired one

Table –7

	1	2	3	4	5	6	7
L	1	4	5	8	9	10	12
IC	1	1	1	1	1	1	1

.

Figure-3 represents the optimal solution to the problem

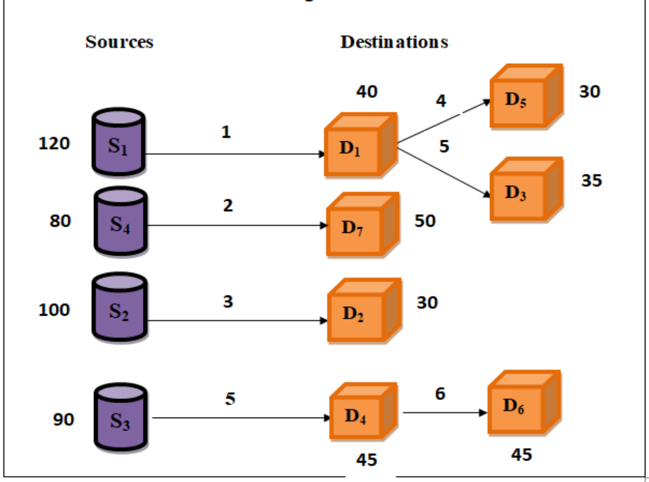
In the below **figure-3**, source 1 supplies its product to destination 1. Source 5 (destination 1) supplies its product to the destinations 5 and 3. Source 3 supplies its product to destination 4. Source 6 (destination 4) supplies its product to the destinations 6. Destination 7 and destination 2 gets its requirement of the product from the source 4 and source 2 respectively. The transshipment cost for the proposed pairs are

Total cost = 1 + 2 + 3 + 4 + 5 + 5 + 6 = 26 units.

Table-8

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

. Figure –3



From the above optimal feasible solution the total bulk transshipment cost from given sources to 7 destinations as follows.

Total cost = 1 + 2 + 3 + 4 + 5 + 5 + 6 = 26 units

X. EXPERIMENTAL RESULTS

A C-code for the suggested LSA is verified at various hard instances.. The inputs like D (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]. 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 -9**. It is observed that, the searching of the OS takes fairly less time. Here microseconds are represented by zero

Table-9

SN	m	K	N	Q	Trail Solution	Optimal Solution	CPU RUN TIME AT+ST
1	4	2	7	2	200	26	0.000000
2	5	2	8	2	200	58	0.000000

TEST Engineering & Management

March - April 2020
ISSN: 0193-4120 Page No. 10584 - 10595

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
20	70	5	80	6	500	140	1.153460

Here m = sources count, *k* = destinations count which are acting as shipment nodes, n = number of destinations. Q = the number of times can supply the requirement of destinations by a destinations which are acting as shipment nodes.

XI. COMPARISON DETAILS

LSA using PRT based with C code is implemented here.. In the following table microseconds are represented by zero.

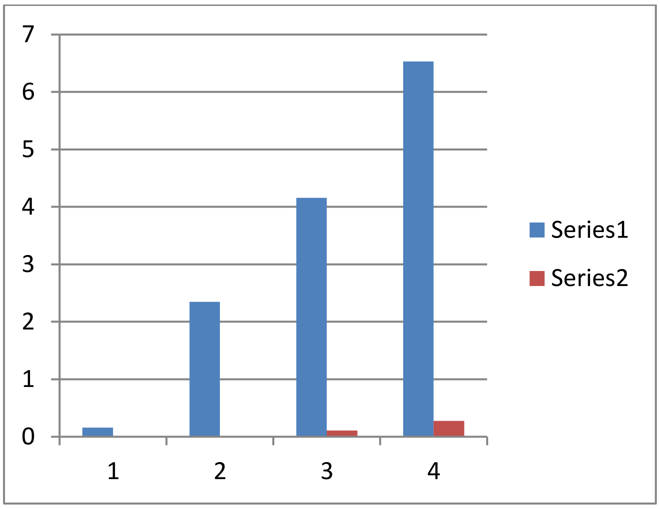
Table-10

S.No.	No. of Sources	No.of Destinations	Pb	Pr
1	4	7	0.16	0.000000
2	8	10	2.35	0.000000
3	10	15	4.16	0.109890
4	15	25	6.53	0.274725

Series 1=Run time of CPU in order to receive OS by Published model

Series 2=Run time of CPU in order to find OS by Proposed model

Graph-1



XII. CONCLUSIONS AND FUTURE RESEARCH

The above discussion focuses on Lexi Search Algorithm constructed on PRT by which A Variant Constraint Transshipment Problem can be solved. In the context of future research two problems namely Vehicle Routing Problem with Inter-Loading Facilities and Minimum Spanning Connectivity of Clustered Cities to the Head Quarter City can be proposed and analyzed by means of C-Language

REFERENCES:

1. G.Vijayalakshmi (2013), Lexisearch Approach to Travelling Salesman Problem, IOSR Journal of Mathematics (IOSR-JM), Volume6, Issue4, May-June2013,Pp1-08
2. K.ChendraSekhar et.al (2012), A Problem Recognition Lexisearch Approach to traveling salesman problem with additional constraints, International

Journal on Computer Science and Engineering (IJSCE), vol4.No 02, Feb2012 pgs307-320
3. ZakirHussain Ahmed (2011): A data-guided lexis arch algorithm for the bottleneck travelling salesman problem, International Journal of Operations Research 12(1):20-33
4. Abuali, F. N., Wainwright, R. L. and Schoenefeld, D.A.(1995): “Determinant factorization: A new encoding scheme for spanning trees applied to the probabilistic minimum spanning tree problem,” in: L.J. Eshelman (Ed.), Proceedings of the Sixth International Conference on Generic Algorithms, Morgan Kaufmann Publishers, San Francisco, California, pp. 470 – 475
5. ShaliniArora et.al (2017):”A Lexisearch Algorithm for a time minimizing assignment problem”, OPSEARCH 35,193-213(1998), Springer Link
6. Amit Kumar, AmarpreetKaur, and Anila Gupta. (2011): “New methods for solving fuzzy transportation problems with some additional transshipments,” ASOR Bulletin, 30(1), 42-61.
7. Archetti, C., Hertz, A., and Speranza, M.G. (2006): “A tabu search algorithm for the split delivery vehicle routing problem,” Transportation Science 40, 64 - 73.
8. Balakrishna, U. (2009): [a] “Generalized Time Dependent Travelling Salesman Problem with Cyclic constraint [GTSP]”, [b] “Generalized Time Dependent Travelling Salesman Problem (cluster constraint) [GTDTSP] models,”
9. Bhavani, V. &Sundara Murthy M. (2006): “Truncated M-Travelling Salesmen Problem,” OPSEARCH, 43(2).
10. Borovska P., Lazarova, M. and Bahudejla, S. (2007): “Strategies for Parallel Genetic Computation of Optimization Problems,” Journal Biotechnologies & Biotechnological Equipment, Vol.21, No.2, pp.241-246.
11. Lee, Y.H., Jung, J.W., and 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.
12. Khurana, A. and Arora, S. (2011): “Solving transshipment problems with mixed constraints,” International Journal of Management Science and Engineering Management, 6(4):292–297.
13. Kek, A.G.H., Cheu,R.L., and Meng,Q.(2008): “Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots,” Mathematical and Computer Modelling, vol. 47, no. 1-2, pp. 140–152.
14. Karuno, Y., Tachibana, T., and Yamashita, K. (2009): “A transsipment problem with a permutable transit vector,” Proceedings of JSME/SSJ International Symposium on Scheduling2009, 163–169.
15. Fagerholt, K. (2004): “Designing optimal routes in a liner shipping problem,” Maritime Policy & Management 31, 259–268.