March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

C -Language Approach to Vehicle Routing Problem with Inter Loading Facilities

Sangeetham Prasad[1], G. Balaji Prakash[2], V. B. V. N. Prasad[2], T.Nageswara Rao[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: 10623 - 10637
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 paper primarily focuses on the problem **'Vehicle Routing Problem with Inter Loading Facilities'.** Suppose there are some cities/stations available. Among them some of the cities act as sources including head quarter and remaining cities act as destinations. All sources have some availability of goods/commodity/load and all destinations have the requirements of goods. The vehicle starts from the head quarter with given load capacity and supply requirements of some destinations. If all the destination requirements are satisfied then the vehicle comeback to the head quarter city. Suppose the load/goods of a vehicle is low while supplying the destinations, then there is a facility that the vehicle fill the sufficient load by visiting the near source station and supply the destinations. Here the vehicle need not visit all source cities while supplying the destination requirements, but all the destinations must be satisfied. The availability of goods at the source cities are always greater than or equal vehicle capacity. The aim of the problem is to get minimum distance/cost for satisfy all destinations subject to some conditions.

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

I. INTRODUCTION

Let **N** be the set of n stations defined as **N= {1, 2, 3, 4… n}** and here the city **'1'** taken as the home city. Among them **S** is set of **k** sources including city **1** and defined as **S= {α_1, α_2, α_3...α_k}.** Let N^1 be the set of **n-k** destinations defined as N^1**= {β_1, β_2, β_3,….,β_{n-k}}.** Let the requirement of destination **j** ϵ N^1 is **DR (j)** and the capacity of the source **i** ϵ **S** is **SC (i).** Let the vehicle load capacity be **'α'**. The vehicle starts its tour from the home city (say 1) and come back to it after supplying the requirement of all **n-k** destinations. The vehicle may or may not visit all the k source stations in its tour. While supplying the destination requirements, if the load of a vehicle is less than the requirement of destination requirements then the vehicle must visit the nearest source city to filling the load up to its capacity and supply the destination requirements. Hence the objective of the problem is to find a minimum total distance in its tour while completing all the destination requirements subject to the conditions.

March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

II. MATHEMATICAL MODELING

Min (Z) $\equiv \sum_{i\in N}\sum_{j\in N} D\ (i,j)X(i,j)$,

$S \cup N^1 = N\ , 1 \in S$& S$\cap N^1 = \emptyset$…….…… (1)

Under the conditions

$$\sum_{i\in N}\sum_{j\in N} X\ (i,j)\ = m = \ |M|\ ,$$

$n - k \le m \le n, \quad M \subseteq N$ ……..… (2)

$\sum_{i\in S} SC\ (i)\ \ge\ \sum_{j\in N^1} DR\ (j)$ ……..…… (3)

$$\left.\begin{array}{l} \alpha, \beta\ \in S \quad Let\ \alpha, \gamma_1\ , \gamma_2, \gamma_3, \ldots\ldots\ldots\ldots\ldots, \gamma_p, \beta \\ \text{be the sequence of cities } \alpha\ to\ \beta \text{ in the tour} \\ \sum_{i=1}^{p} DR\ (\gamma_i) \le VL\ , \text{Where VL } = \boldsymbol{\alpha} \end{array}\right\} (4)$$

X(i,j)=1or0…… (5)

Equation (**1**) describes the objective function i.e. to minimize the total distance/cost subjected to constraints. Constraint **(2)** represents that the trip includes m cities.

The constraint **(3)** takes care of the restriction of availability and requirement of the product between sources and destinations, i.e., the sum of the available capacities at sources is more than the sum of the demands at destinations of a product. The constraint **(4)** assumes that the demand at the destination from source to source should be lesser than or equals vehicle capacity. **(5)**denotes that if the vehicle is traversed from i to j its value is 1, else zero.

III. NUMERICAL FORMULATION

The ideas and the algorithms are depicted by a suitable arithmetical specimen. where cities total no.is taken as **n=9(N = 1 to 9)** among them **1** be the head quarter (**HQ**), number of loading points/source facilities **k = 3 (S = {1, 4, 8})** and remaining **6** cities are destinations **(N^1 = {2, 3, 5, 6, 7, 9}).** Let **SC (i)** is the availability of a product at sources and **DR (j)** is the condition of a product at the destinations. Now distance matrix **D** is shown in **Table-1**.

Table – 1

D(i,j)=

	1	2	3	4	5	6	7	8	9	SC
1	∞	0	24	40	3	10	29	27	39	100
2	31	∞	40	8	18	0	19	5	22	-
3	19	32	∞	1	38	8	31	10	21	-
4	22	18	24	∞	41	34	2	20	13	200
5	4	24	17	25	∞	31	21	20	32	-
6	26	7	23	1	23	∞	27	11	40	-
7	30	37	3	24	14	29	∞	35	28	-
8	26	29	14	34	28	6	11	∞	9	150
9	27	16	12	37	5	28	38	36	∞	-
DR	-	40	60	-	40	50	30	-	40	

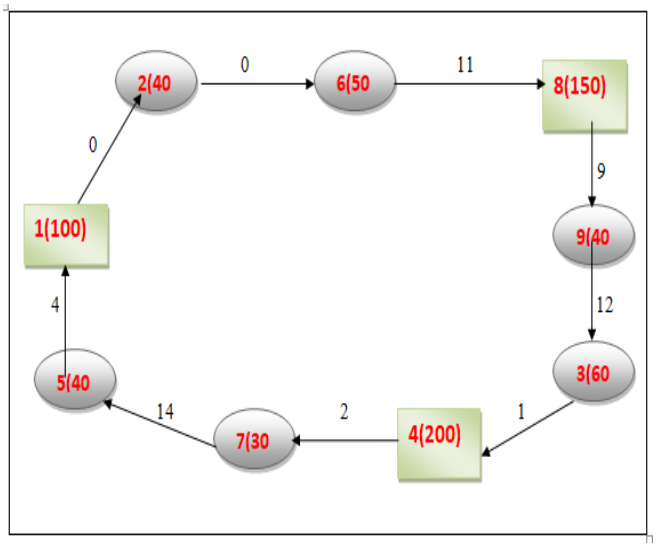
Suppose **D (4, 7) = 2** means that the distance between the cities **4** and **7** is **2**. More over **SC** and **DR** represent that the availability of sources and requirements of destinations. The source and destination arrays **SC** and **DR** are **SC (i) = b**means that the availability of source **i** is**b** and **DR (j) = c** means that the requirement of destination **j** is **c**. Here **SC (4) = 200**means that the availability of source **4** is **200** and **DR (5) = 40** means that the requirement of destination **5** is **40**. Here we have taken the vehicle capacity**$\boldsymbol{\alpha = 100}$** units. The aim here is to get a least total distance to meet requirements at the destinations subjected to the constraints.

IV. FEASIBLE SOLUTION

The ordered pair set {(1, 2), (2, 6), (6, 8), (8, 9), (9, 3), (3, 4), (4, 7), (7, 5), (5, 1)} represents a feasible solution. In **fig-1**, values in ellipse denote name of the destination city and the values in the parenthesis of ellipse denotes the requirement of the destination city. The rectangle box represents sources/pickup points and its available capacity indicated in the respective parenthesis. The values along the arcs indicate the distance between the connected cities.

March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

Figure – 1



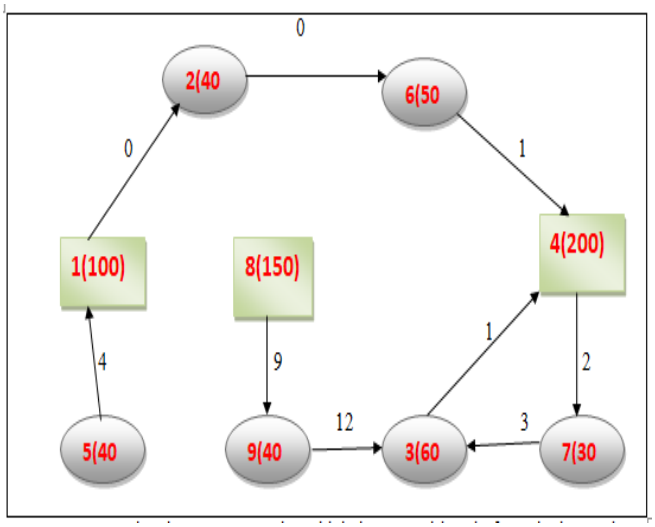
In the above **figure – 1,** the vehicle has started its trip from the home city 1 with sufficient load of **100** units. First, it has reached the destination city **2** and supplied the destination's requirement of **40** units, from city **2** the vehicle reached city **6** and supplied its requirement of **50** units. Now, the remaining load in the vehicle is only **10** units which are insufficient to the nearest destination city'srequirement. So, the vehicle reached the next nearest source city **8** to reload **90** units of its availability for shortage of vehicle loadand reached the destination city **9**andthen city**3,** there it supplied its requirements **40** and **60** units respectively. Now the load in the vehicle is **0** units. As above, the vehicle again reached the nearest source city **4** to reload **100** units of its availability for shortage of vehicle loadand reached its destination city**7**andthen city **5** to supply its requirement 3**0** and **40** units respectively. After completing the all destinations' requirements, the vehicle has come back to the home city. So, the trip has given a feasible solution. So the solution is

Z = D(1,2) + D(2,6) + D(6,8) + D(8,9) + D(9,3) + D(3,4) + D(4,7)+ D(7,5) + D(5,1)
= 0 + 0 + 11 + 9 +12 + 1 +2 + 14 + 4
= 53

V. INFEASIBLE SOLUTION

Infeasible solution is one which does not satisfy all the conditions.Consider an ordered pair set {**(1, 2), (2, 6), (3, 4), (6, 4), (4, 7), (7, 3), (5, 1), (8, 9), (9,3)}** characterizes an infeasible solution.

Fig-2



From the above **figure-2**, the vehicle has started its trip from the home city 1 with sufficient load of **100** units. First it has reached the destination city **2** and supplied the destination's requirement of **40** units,from city **2,** the vehicle reached city **6** and supplied its requirement of **50** units. After that the vehicle reached the source city **4** to reload, but the vehicle already approached the source city **4** from city **3**, this leads to contradict the feasible conditions. Because, in the definition of the tour the vehicle should reach each city exactly only once. Now the vehicle reached the destination city **7** from source city **4**, after supplied its requirement, the vehicle reached city **3**. Further the vehicle comeback to head quarter city from city **5.** This is again a contradiction that the vehicle has completed its tour without supplying the all destinations' requirements. Finally the vehicle started its trip from the source city **8** and reached the destination city **9,** there,it hassupplied its requirement from the vehicle; again it has reached another destination city **3** and supplied its requirement. The distance/cost for those pairs is

March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

Z = D(1,2) + D(2,6) + D(3,4) + D(6,4) + D(4,7) + D (7,3) +D (5,1) + D (8,9)+ D(9,3)
= 0 + 0 + 1 + 1 + 2 + 3 + 4 +9 + 12=32

6. Algorithms

(Algorithm for feasible checking)

STEP1 : IS (IC [CA] ==1) IF YES GOTO 1
IF NO GOTO 2

STEP2 : IS (b [RA] = =1) IF YES GOTO 4
IF NO GOTO 3
STEP3 : IS (b [CA]==1) IF YES GOTO 5
IF NO GOTO 20
STEP4: IS (b [CA]==1) IF YES
{DR [RA] =DR [RA] +DR [CA]; GOTO 10}
IF NO GOTO 20
STEP5 : IS (SA [RA]>=DR [CA])
IF YES GOTO 6
IF NO GOTO 1
STEP6 : W=CA
A=RA GOTO 7

STEP7 : IS (SW [W] ==0) IF YES
{SA [W] =SA [A]-DR [W]; GOTO 11}
IF NO GOTO 9
STEP8: ISb [W] ==1) IF YES GOTO 7
IF NO {SA [A]=SA [RA]-DR [CA];
GOTO 21}
STEP9 : A=W
W=SW [W]
LC=LC+1
IS (W==RA) IF YES GOTO 10
IF NO GOTO 8
STEP10: IS (X>=DR [RA]) IF YES GOTO11
IF NO {DR [RA] =DR [RA]-DR [CA];
GOTO 1}

STEP11: DR [RA] =DR [RA]-DR [CA];
W=RA;
A=CA; GOTO 12
STEP12: IS (SWI [W] ==0) IF YES
{DR [W]=DR[W]+DR[A];
GOTO 11}
IF NO GOTO 13
STEP13: M=SWI [W];
IS B [M] ==1) IF YES GOTO19
IF NO GOTO 14
STEP14: DR [W] =DR [W]+DR [A]
IS (SA [M]>=DR [W]) IF YES
{QT=SA [M] – DR [W]; GOTO 15}
IF NO GOTO 1
STEP15: W=CA GOTO 16

STEP16: IS (SW [W] ==0) IF YES
{SA [W] =QT; GOTO 11}
IF NO GOTO 17
STEP17: IS (b [W] ==1) IF YES GOTO 18
IF NO {SA [A] =QT; GOTO 21}
STEP18: A=W
W=SW [W]
LC=LC + 1
IS (W==RA) IF YES GOTO 10
IF NO GOTO 16
STEP19: DR [W] =DR[W]+DR [A]
DR[M]=DR[M]+DR [W]
IS (X>=DR[M]) IF YES
{W=M;A=W; GOTO 12}
IF NO GOTO 1
STEP20: W=CA; GOTO 21

STEP21: IS (SW[W]==0) IF YES GOTO 11
IF NO GOTO 22
STEP22: LC=LC+ 1
W=SW [W]
IS (W==RA) IF YES GOTO 10
IF NO GOTO 21
STEP 21: X = 1 GO TO STOP

STEP 22: STOP

Begin with the partial word $L_1 = (a_1) = (1)$. L_k is constructed as $L_k = L_{k-1} * (\alpha_p)$. * indicates chain formulation. One can compute the entries of V (L_k) and LB (L_k) at once. Two situations arises one for branching and other for continuing the search.

1. LB (L_k) < VT. VerifyL_k is feasible or not. If it is feasible go to a partial word of under (k+1). This denotes a sub-block of the block of words represented by L_k? If L_k is not feasible then consider the next partial word p by taking another letter which succeeds a_k in the position. If all the words of order p are exhausted then we

March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

consider the next partial word of order (k-1).

2. LB (L_k) ≥ VT. Here discardLk. Discard the block of word with L_k as leader as not having optimum feasible solution and also reject all partial words of order p that succeeds L_k.

Nowa Lexi-Search Algorithm to find an Optimal Feasible word is developed.**Lexi-Search Algorithm**

STEP 0: Initialization
The arrays SN, R, C, D, DC, B, M, LB, V, L, DR, r, c and values of N are made available IR, IC, SW, SWI and ST are initialized to zero. The values I=1, J=0, VT=999 and Max=N^2- N

STEP 1: J=J+1;
L[K]=J;
LC=1;
IS (J>Max) IF YES GOT 9
IF NO GOT 2

STEP 2: RA=r[J]
CA=c[J];
V[K]=V[K-1]+d[J];
LB[K]=V[K] +DC [J+n-K]-DC [J]
GOTO 3

STEP 3: IS (LB[K]>=VT) IF YES GOTO
IF NO GOTO 4

STEP 4: Check feasible (Using Algorithm 1)
IS (IX=0) IF YES GO TO 2
IF YES GO TO 5

STEP 5: IS (LC==n) IF YES GOTO 8
IF NO GOT 6

STEP 6: IS (b[RA]==1) IF YES
{DR[RA]=DR[RA] – DR[CA];
GOTO 1}
IF NO GOTO 1

STEP 7: L[K]=J
IC[CA]=1
IR[RA]= 1
SWI[CA]=RA
SW [RA] = CA
DR[CA]=0
K=K+1
GOTO STEP 1

STEP 8: VT=LB[K]
L[K]=J
IS (b[RA]==1) IF YES
{DR[RA]=DR[RA] – DR[CA];
GOTO 10}
IF NO GOTO 10

STEP 9: IS (K==1) IF YES GOTO 11
IF NO GOTO 10

STEP10: K=K-1
J = L [K]
RA=r[J]
CA = c [J]
L [J] = 0
IC[CA]=0
IR [RA] = 0
SWI[CA]=0
SW[RA]= 0
LB [K+1] = 0
L[K+1]=0
V[K+1]=0
IS (b[RA]==1) IF YES
{DR[RA]=DR[RA] – DR[CA];
GOTO 1}
IF NO GOTO 1

STEP11: STOP/END

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 shows that feasible solution does not exist

VII. SEARCH TABLE

The steps of getting an optimal word with the above algorithm for the numerical example isappeared in Tab-**6.** (1), (2), (3), (4), (5), (6), (7), (8) and (9) provides the letters in the first, second, third, fourth, fifth, sixth, seventh, eighth and ninthpositions of a word. Next two columns V and LB are indicate value and lower bound of the respective partial word. R and C contribute the row and column indices of the letter. The last column offers the remarks concerning the suitability of the partial words (i.e. if a partial word is feasible word then accept the letter otherwise reject the letter) and here A indicates

March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

the acceptance and R for rejectance of the letter in the respective position.

Tab-6

SN	1	2	3	4	5	6	7	8	9	V	LB	R	C	Remarks
1	1									0	19	1	2	A
2		2								0	19	2	6	A
3			3							1	19	3	4	A
4				4						2	19	6	4	R
5				5						3	23	4	7	A
6					6					6	23	1	5	R
7					7					6	26	7	3	R
8					8					7	30	5	1	A
9						9				12	30	2	8	R
10						10				12	33	9	5	R
11						11				13	36	8	6	R
12						12				14	39	6	2	R
13						13				15	42	2	4	R
14						14				15	44	3	6	R
15						15				16	47	8	9	A
16							16			26	47	1	6	R
17							17			26	48	3	8	R
18							18			27	50	6	8	A
19								19		38	50	8	7	R
20								20		39	52	9	3	A
21									21	52	52	4	9	R
22									22	53	53	7	5	A, VT=53
23								21		40	54	4	9	R, > VT
24							19			27	52	8	7	R
25							20			28	55	9	3	R, > VT
26						16				17	49	1	6	R
27						17				17	51	3	8	R
28						18				18	54	6	8	R, > VT
29					9					8	34	2	8	R
30					10					8	37	9	5	A
31						11				14	37	8	6	R
32						12				15	40	6	2	R

TEST Engineering & Management

March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

33						13				16	43	2	4	R
34						14				16	45	3	6	R
35						15				17	48	8	9	R
36						16				18	50	1	6	R
37						17				18	52	3	8	R
38						18				19	55	6	8	R, > VT
39					11					9	41	8	6	R
40					12					10	45	6	2	R
41					13					11	48	2	4	R
42					14					11	51	3	6	R
43					15					12	54	8	9	R, > VT
44				6						4	27	1	5	R
45				7						4	31	7	3	A
46					8					8	31	5	1	A
47						9				13	31	2	8	R
48						10				13	34	9	5	A
49							11			19	34	8	6	R
50							12			20	36	6	2	R

51							13			21	38	2	4	R
52							14			21	40	3	6	R
53							15			22	42	8	9	R
54							16			23	44	1	6	R
55							17			23	45	3	8	R
56							18			24	47	6	8	A
57								19		35	47	8	7	A
58									20	47	47	9	3	R
59									21	48	48	4	9	A, VT =48
60								20		36	49	9	3	R, > VT
61							19			24	49	8	7	R, > VT
62						11				14	37	8	6	R
63						12				15	40	6	2	R
64						13				16	43	2	4	R
65						14				16	45	3	6	R
66						15				17	48	8	9	R, = VT

TEST Engineering & Management

March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

67					9					9	35	2	8	R
68					10					9	38	9	5	A
69						11				15	38	8	6	R
70						12				16	41	6	2	R
71						13				17	44	2	4	R
72						14				17	46	3	6	R
73						15				18	49	8	9	R, > VT
74					11					10	42	8	6	R
75					12					11	46	6	2	R
76					13					12	49	2	4	R, > VT
77				8						5	36	5	1	A
78					9					10	36	2	8	R
79					10					10	39	9	5	A
80						11				16	39	8	6	R
81						12				17	42	6	2	R
82						13				18	45	2	4	R
83						14				18	47	3	6	R
84						15				19	50	8	9	R, >
														VT
85						11				11	43	8	6	R
86						12				12	47	6	2	R
87						13				13	50	2	4	R, > VT
88				9						6	40	2	8	R
89				10						6	44	9	5	A
90						11				12	44	8	6	R
91						12				13	48	6	2	R, = VT
92				11						7	49	8	6	R, > VT
93			4							1	23	6	4	A
94				5						3	23	4	7	A
95					6					6	23	1	5	R
96					7					6	26	7	3	A
97						8				10	26	5	1	A
98							9			15	26	2	8	R
99							10			15	28	9	5	A
100								11		21	28	8	6	R

TEST Engineering & Management

March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

101							12		22	30	6	2	R
102							13		23	31	2	4	R
103							14		23	32	3	6	R
104							15		24	34	8	9	A
105								16	34	34	1	6	R
106								17	34	34	3	8	A, VT =34
107							16		25	35	1	6	R, > VT
108						11			16	31	8	6	R
109						12			17	33	6	2	R
110						13			18	35	2	4	R, > VT
111					9				11	29	2	8	R
112					10				11	32	9	5	A
113						11			17	32	8	6	R
114						12			18	34	6	2	R, = VT
115					11				12	35	8	6	R, > VT

116					8				7	30	5	1	A
117						9			12	30	2	8	R
118						10			12	33	9	5	A
119						11			18	33	8	6	R
120						12			19	35	6	2	R, > VT
121						11			13	36	8	6	R, > VT
122				9					8	34	2	8	R, = VT
123				6					4	27	1	5	R
124				7					4	31	7	3	A
125					8				8	31	5	1	A
126						9			13	31	2	8	R
127						10			13	34	9	5	R, = VT
128					9				9	35	2	8	R, > VT
129				8					5	36	5	1	R, > VT
130			5						2	2	4	7	A

TEST Engineering & Management

March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

										8				
131				6						5	28	1	5	R
132				7						5	32	7	3	A
133					8					9	32	5	1	A
134						9				14	32	2	8	R
135						10				14	35	9	5	R, > VT
136					9					10	36	2	8	R, > VT
137				8						6	37	5	1	R, > VT
138			6							3	33	1	5	R
139			7							3	38	7	3	R, > VT
140		3								1	24	3	4	A
141			4							2	24	6	4	R
142			5							3	29	4	7	A
143				6						6	29	1	5	R
144				7						6	33	7	3	R
145				8						7	38	5	1	R, > VT
146			6							4	34	1	5	R, = VT
147		4								1	29	6	4	A
148			5							3	29	4	7	A
149				6						6	29	1	5	R
150				7						6	33	7	3	A
151					8					10	33	5	1	A
152						9				15	33	2	8	R
153						10				15	36	9	5	R, > VT
154					9					11	37	2	8	R, > VT
155				8						7	38	5	1	R, > VT
156			6							4	34	1	5	R, = VT
157		5								2	35	4	7	R, > VT
158	2									0	24	2	6	A
159		3								1	24	3	4	A
160			4							2	2	6	4	R

TEST Engineering & Management

March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

										4				
161			5						3	29	4	7	A
162				6					6	29	1	5	A
163					7				9	29	7	3	R
164					8				10	33	5	1	R
165					9				11	37	2	8	R, > VT
166				7					6	33	7	3	R
167				8					7	38	5	1	R, > VT
168			6						4	34	1	5	R, = VT
169		4							1	29	6	4	A
170			5						3	29	4	7	A
171				6					6	29	1	5	A
172					7				9	29	7	3	A
173						8			13	29	5	1	R
174						9			14	32	2	8	R
175						10			14	35	9	5	R, > VT
176					8				10	33	5	1	R
177						9			11	37	2	8	R, > VT
178				7					6	33	7	3	A
179					8				10	33	5	1	A
180						9			15	33	2	8	R
181						10			15	36	9	5	R, > VT
182					9				11	37	2	8	R, > VT
183				8					7	38	5	1	R, > VT
184			6						4	34	1	5	R, > VT
185		5							2	35	4	7	R, > VT
186	3								1	30	3	4	A
187		4							2	30	6	4	R
188		5							3	36	4	7	R, > VT
189	4								1	36	6	4	R, > VT

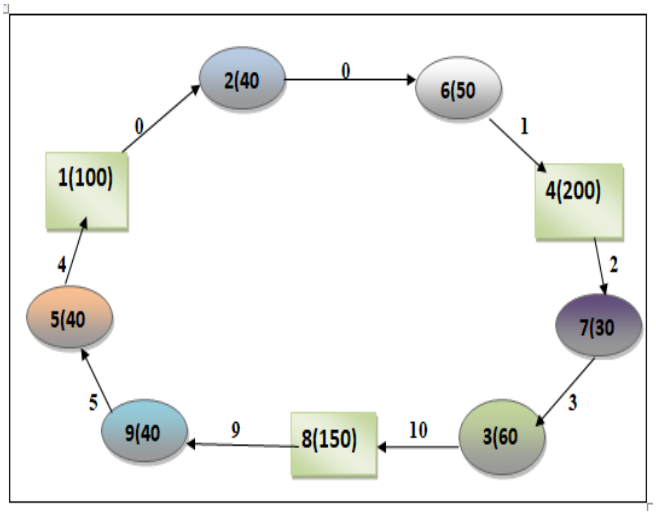
March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

VIII. COMMENTS

Shaded row in the **Tab-6**, presents optimal solution and at end of the search the current value of **VT** is **34.** Then the partial word is **L_9 = (1, 2, 4, 5, 7, 8, 10, 15, 17)** is an optimal feasible word. It is known in the **106th**row of the search table. For this optimal word the arrays L, IR, IC, SW and ST are appearing in the following **Tab – 7.Tab -7**

	1	2	3	4	5	6	7	8	9
L	1	2	4	5	7	8	10	15	17
IR	1	1	1	1	1	1	1	1	1
IC	1	1	1	1	1	1	1	1	1
SW	2	6	8	7	1	4	3	9	5

Figure –3



At the end of the search table the optimal solution value of **VT** is **34**and is the value of optimal feasible word **L_9 = (1, 2, 4, 5, 7, 8, 10 15, 17)**. **Fig-3** represents the optimal solution of the problem.

From the above **figure-3**, the vehicle has started its trip from the home city 1 with sufficient load of **100** units. First it has reached the destination city **2** and supplied its requirement of **40** units. From city **2,** the vehicle reached city **6** and supplied its requirement of **50** units. Now, the remaining load in the vehicle is only **10** units which are insufficient for the nearest destination city's requirement. So, the vehicle reached the nearest source city **4** to reload **90** units of its availability for shortage of vehicle loadand reached the destination city**7**andthen city **3,** there it supplied its requirement **30** and **60** units respectively. Now the load in the vehicle is only **10** units. As above, the vehicle again approached the nearest source city **8** to reload **90** units of its availability for shortage of vehicle loadand reached destination city**9** andthen **5,** there it supplied its requirement **40**and**40** units respectively. After completed the all destinations requirement, the vehicle has come back to the home city. So, the trip has given a feasible solution. Therefore the value of the solution is:

Z = D (1, 2)+D (2, 6)+D (6, 4)+D (4, 7)+

D (7, 3)+D (3, 8)+D (8, 9)+D (9, 5)+D (5, 1)

= 0 + 0 + 1 + 2 + 3 + 10 + 9 + 5 + 4

= 34 units

{(1, 2), (2, 6), (6, 4), (4, 7), (7, 3), (3, 8),(8, 9), (9, 5), (5, 1)} representing the pattern in **Tab-8,** is a feasible solution. In keeping with the pattern signified in **fig-3,** itsatisfies all the constraints in Mathematical Formulation.

Tab-8

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

IX. EXPERIMENTAL RESULTS

In the below **tab-10**, the last two columns display the CPU run time of published and

March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

proposed models. As compared the two models of sizes N=5, 10, 12, 15 & 20the runtime of this instance with the existing model are 0.054945sec, 0.109890sec, 0.164835sec, 0.439560sec & 1.318681sec and the proposed model took 0.0sec, 0.0sec, 0.0sec, 0.1035sec & 0.2481 sec., it is sensibly less time.The present model takes very less computational time for getting the optimal solution. Therefore the present model can be chosen for solving the higher dimensional problems also.

Table-9

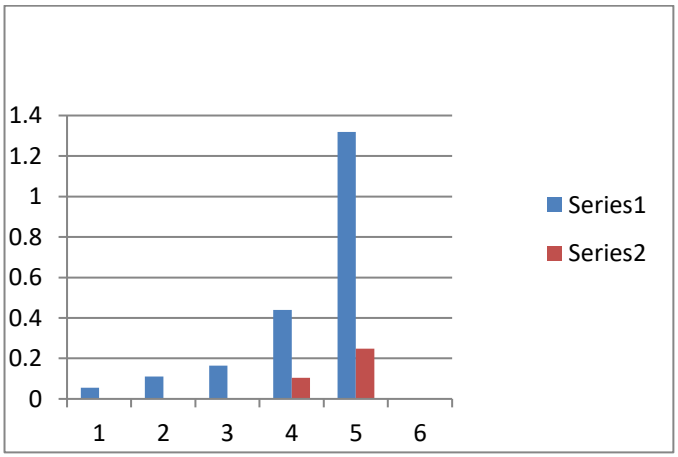
SN	N	S	D	NPT	VT	CPU Run Time in seconds
						Avg. AT+ ST
1	5	2	3	6	35	0.0000
2	10	4	6	6	55	0.0000
3	12	5	7	8	79	0.0000
4	15	6	9	7	87	0.1035
5	17	6	11	7	98	0.1089
6	20	8	12	6	107	0.2481
7	22	9	13	6	119	0.2735
8	25	10	15	6	124	0.3548
9	28	13	15	6	132	0.5196
10	30	14	16	6	148	0.6743
11	35	17	18	5	167	0.6894
12	40	18	22	5	185	0.8193

Table-10

S. No.	No. of cities	Published model	Proposed model
1	5	0.054945	0.0000
2	10	0.109890	0.0000
3	12	0.164835	0.0000
4	15	0.439560	0.1035
5	20	1.318681	0.2481

In the above **Table-10**, the last two columns show the CPU run time of published model and proposed model. As compared the two models of sizes N=5, 10, 12, 15 & 20. The runtime of this instance with the existing model are 0.054945sec, 0.109890sec, 0.164835sec, 0.439560sec & 1.318681sec and the proposed model took 0.0sec, 0.0sec, 0.0sec, 0.1035sec & 0.2481 sec., it is reasonably less time.The present model takes very less computational time for finding the optimal solution. Hence, suggested the present model for solving the higher dimensional problems also.The graphical picture of the CPU run time for the two models given in the above 5 instances is depicted below. In the **Graph-1**, horizontal axis take the SN and vertical axis take the values of CPU run time for the published and proposed models.

Graph-1



March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

X. CONCLUSIONS AND FUTURE RESEARCH

In the discourse an exact algorithm known as Lexi-Search Algorithm (LSA)usingPattern Recognition Technique(PRT)has been proposed in order to solve Vehicle Rooting Problem with Inter Loading Facilities. At the first stage the model is formulated into a Zero-One programming problem and in the next stage its optimal solution is obtained by using LSA built on PRT.The solution technique has been presented with appropriate numerical examples. The proposed algorithm has been sketched out using C-language. Furthermore all the computational details are displayed. Altogether the CPU run time is fairly less for higher values to parameters of the problem to obtain optimal solution. In the context of future researchIn the context of future research two problems namely a Variant Constraint Bulk Transshipment Problem and Minimum Spanning Connectivity of Clustered Cities to the Head Quarter City can be projected and investigated by the use of C-Language.

REFERENCES

1. 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.
2. Herer, Y.T., Tzur, M., and Yücesan, E. (2006)): "The multi-location transshipment problem," IIE Transactions, 38, pp. 185–200.
3 .Gendreau,M., Laporte, G., and Potvin, J.-Y.(2002): " Metaheuristics for the capacitated vehicle routing problem," in: P. Toth, D. Vigo (Eds.), The Vehicle Routing Problem, Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, pp. 129–154
4. G.Vijayalakshmi (2013), Lexi-Search Approach to Travelling Salesman Problem, IOSR Journal of Mathematics (IOSR-JM), Volume6, Issue4, May-June2013, Pp1-08
5. 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
6. ZakirHussain Ahmed (2011): A data-guided lexisearch algorithm for the bottleneck travelling salesman problem, International Journal of Operations Research 12(1):20-33
7. 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
8. ShaliniArora et.al (2017):"A Lexisearch Algorithm for a time minimizing assignment problem", OPSEARCH 35,193-213(1998), Springer Link
9. Amit Kumar, AmarpreetKaur, and Anila Gupta. (2011): "New methods for solving fuzzy transportation problems with some additional transshipments," ASOR Bulletin, 30(1), 42-61.
10. 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
11. 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.
12. Balakrishna, U. (2009): [a] "Generalized Time Dependent Travelling Salesman

March - April 2020
ISSN: 0193-4120 Page No. 10623 - 10637

Problem with Cyclic constraint [GTSP]", [b] "Generalized Time Dependent Travelling Salesman Problem (cluster constraint) [GTDTSP] models,"

13. Bhavani, V. &Sundara Murthy M. (2006): "Truncated M-Travelling Salesmen Problem," OPSEARCH, 43(2).
14. 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.
15. 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.
16. Khurana, A. and Arora, S. (2011): "Solving transshipment problems with mixed constraints," International Journal of Management Science and Engineering Management, 6(4):292–297.
17. 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.
18. 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.
19. Fagerholt, K. (2004): "Designing optimal routes in a liner shipping problem," Maritime Policy & Management 31, 259–268.
20. Laporte, G., Semet, F. (2002): *"Classical heuristics for the capacitated VRP,"* In: Toth, P., Vigo, D. (Eds.), The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, pp. 109–128