Commun. Optim. Theory 2016 (2016), Article ID 22

MATHRES
Communications in Optimization Theory
Available online at http://cot.mathres.org

SOLVING COST VARYING TRANSPORTATION PROBLEMS BY GENETIC ALGORITHM BASED ON A SPANNING TREE AND PRÜFER NUMBER

CHANDAN BIKASH DAS

Department of Mathematics, Tamralipta Mahavidyalaya, Tamluk, Purba Midnapore-721636, West Bengal, India

Abstract. This paper presents a mathematical model for the cost varying transportation problem (CVTP) in which cost is varied due to the capacity of vehicles as well as amount of transport quantity. The main purpose is to develop a bi-level mathematical model. This model determines the minimum total transportation cost by determining minimum cost of the transportation at the route(i, j). This model is also a mixed-integer mathematical model. To tackle such a problem, a genetic algorithm (GA) based on the spanning tree has been proposed. We focus on the use of Prüfer number encoding based on a spanning tree, which is adopted because it is capable of equally and uniquely representing all possible trees. From this point, the criteria by which chromosomes can always be converted to a CVTP tree is design. The procedures of crossover and mutation operators are newly designed. Numerical examples are presented to illustrate the problem with some conclusions.

Keywords. Cost varying transportation problem; Genetic algorithm; Spanning tree; Prüfer number.

1. Introduction

The classical transportation problem (TP) refers to a special class of linear programming. It is well known as a basic network problem. The first formulation and discussion of a planar transportation model was introduced by Hitchcock [5]. The objective is to determine the amounts shipped from each source to each destination that minimizes the total cost while satisfying both the supply limits and the demand requirements. Efficient methods of solution are derived from the simplex algorithm and were developed in 1947. The transportation problem

E-mail address: cdasmathematics@gmail.com

Received September 23, 2016

can be converted as a standard linear programming problem, which can be solved by the simplex method. However, because of its very special mathematical structure, it was recognized early that the simplex method applied to the transportation problem can be made quite efficient in terms of how to evaluate the necessary simplex-method information (variable to enter the basis, variable to leave the basis and optimality conditions). Charnes *et al.* [1] developed the Stepping Stone Method which provides an alternative way of determining the simplex-method information. Dantzig [2] used the simplex method in the transportation problem as the Primal simplex transportation method. An initial basic feasible solution for the transportation problem can be obtained by using the North West corner rule, Row minima, Column minima, Matrix minima, or the Vogels approximation method. The Modified Distribution method is useful for finding the optimal solution for the transportation problem.

In the past several decades a variety of deterministic and/or stochastic models have been developed considering constant unit transportation cost.In more real world applications, it is often that the unit transportation cost is not constant; it depends on amount of transport quantity and capacity of vehicles. If amount of quantity is small then small(capacity) vehicle is sufficient for deliver. Where as if amount of quantity is large then big(capacity) vehicle is needed. So, depend on amount of transport quantity and the capacity of vehicles, the unit transportation cost is varied. Panda and Das [10] present some transportation problems whose unit transportation cost is varied. This type of transportation problems are known as cost varying transportation problem(CVTP). They [11] modified some techniques to allocate initial basic feasible solutions. then optimize the objective function by deterministic way. Tn this paper we are going to present such type of TP where unit transportation cost is unknown but cost of single trip of all the vehicles are known for each route(i, j). Without considering the unit transportation cost, a bi-level mathematical programming model has been presented whose lower level determines the minimum cost to transport x_{ij} amount in the cell (i, j) and upper level presents the optimal transportation cost of the problem. This bi-level mathematical model is a mixed-integer programming problem.

Theoretically, any general mixed integer programming solution method can be used to solve this kind of problem, for example, branch and bound method, branch and cut method. However, these methods are generally inefficient and computationally expensive for the problems with large size.

Several efficient algorithms have been developed over the past decades for solving the transportation problem when the cost coefficients and the supply and demand values are known exactly.

Metaheuristic methods, can obtain global optimal solution or approximately global optimal solution at the cost of less computational time, are preferable in practical industry applications. To the best of our knowledge, the matrix encoding genetic algorithm for generic nonlinear transportation problems developed by Michalewicz, *et al.* [7] is an only meta heuristic method relevant directly to our problem.

A number of exact solution algorithms are developed to solve the fixed charge transportation (FCT) problem, which include cutting plane approaches [12] and branch and bound approaches [9]. Some heuristic methods are also proposed [14]. Some meta heuristic methods are developed for the FCP problem. Sun et al. [13] developed a tabu heuristic search procedure. Gottlieb and Paulmann [4] proposed a matrix representation genetic algorithm. Li et al. [6] proposed a genetic algorithm where prüfer numbers are use to encode spanning tree in unbalanced fixed charge TP. During decoding, it essential to repair some chromosomes. This paper presents a proposed GA where prüfer numbers are use to encode spanning tree and no repair is required and procedures of crossover and mutations are newly stated.

2. Preliminaries

2.1. Transportation Problem.

In transportation model, there are *m* sources (production units) and *n* destinations(customers). Distribute a single commodity from various sources to various destination o in such a manner that the total transportation cost is minimized. A transportation problem can be stated in **Model 1** as follows:

Model 1

$$\begin{aligned} \min \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}, \\ \text{subject to } \sum_{i=1}^{m} x_{ij} &= a_i,\ i=1,\ldots,m \\ \sum_{j=1}^{n} x_{ij} &= b_j,\ j=1,\ldots,n \\ \sum_{i=1}^{m} a_i &= \sum_{j=1}^{n} b_j \\ x_{ij} &\geq 0\ \forall i,\ \forall j, \end{aligned}$$

where,

x_{ij} quantity to be transport on the route (i,j) from source i to destination j.

c_{ij} unit transportation cost from source i to destination j.

a_i capacity of source i.

b_i number of units demand at destination j.

2.2. Cost varying transportation problem.

Suppose there are two types off vehicles V_1, V_2 from each source to each destination. Let C_1 and $C_2(> C_1)$ are the capacities(in unit) of the vehicles V_1 and V_2 respectively. $R_{ij} = (R^1_{ij}, R^2_{ij})$ represent transportation cost for each cell (i,j); where R^1_{ij} is the transportation cost from source $O_i, i = 1,\ldots,m$ to the destination $D_j, j = 1,\ldots,n$ by the vehicle V_1. And R^2_{ij} is the transportation cost from source $O_i, i = 1,\ldots,m$ to the destination $D_j, j = 1,\ldots,n$ by the vehicle V_2. So, cost varying transportation problem can be represent in the following tabulated form.

	D_1	D_2	..	D_n	$stock(a_i)$
O_1	R^1_{11}, R^2_{11}	R^1_{12}, R^2_{12}	....	R^1_{1n}, R^2_{1n}	a_1
O_2	R^1_{21}, R^2_{21}	R^1_{22}, R^2_{22}	....	R^1_{2n}, R^2_{2n}	a_2
....	....	....	....	....	....
O_m	R^1_{m1}, R^2_{m1}	R^1_{m2}, R^2_{m2}	....	R^1_{mn}, R^2_{mn}	a_m
$Demand(b_j)$	b_1	b_2	....	b_n	

Table 1: Tabular representation of cost varying transportation problem.

2.2.1. Determination of cost in (i, j)

In the above tabulated form we use the following notations.

$O_i, i = 1, \ldots, m$ are sources.

$D_j, j = 1, \ldots, n$ are destinations.

a_i capacity of source i.

b_i number of units demand at destination j.

V_1, V_2 types of vehicles,(Here consider only two types).

C_1 C_1 Capacity to carry of V_1.

$C_2(> C_1)$ Capacity to carry of V_2.

R^1_{ij} transportation cost for V_1. in a single trip from source i to destination j.

$R^2_{ij}(> R^1_{ij})$ transportation cost for V_2 in a single trip from source i to destination j.

P^1_{ij} number of vehicles used of type V_1.

P^2_{ij} number of vehicles used of type V_2.

Let x_{ij} amount of transport quantity from source i to destination j.

Suppose x_{ij} is known. If P^1_{ij} number V_1 type vehicles and P^2_{ij} number V_2 type vehicles are used to deliver x_{ij} units amount in the cell (i, j) i.e.$(x_{ij} \leq (P^1_{ij}C_1 + P^2_{ij}C_2))$.

And cost of transportation in the cell (i, j) is $W_{ij} = P^1_{ij}R^1_{ij} + P^2_{ij}R^2_{ij}$.

So, the transportation cost in (i, j) is determined by the following programming.

$$
\begin{aligned}
\min_{P^1_{ij}, P^2_{ij}} \quad & W_{ij} \\
W_{ij} &= P^1_{ij}R1_{ij} + P^2_{ij}R2_{ij} \\
x_{ij} &\leq P^1_{ij}C_1 + P^2_{ij}C_2) \\
x_{ij} &\geq 0;\ P^1_{ij}, P^2_{ij} \text{ are integers}
\end{aligned} \tag{1}
$$

Mathematical programming of cost varying transportation model The Bi-level mathematical programming cost varying transportation problem under 2-vehicle is formulated in **Model 2** as follows:

Model 2

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}, \\
\text{s. t.} \min_{P_{ij}^1, P_{ij}^2} \quad & W_{ij} \\
W_{ij} &= P_{ij}^1 R1_{ij} + P_{ij}^2 R2_{ij} \\
x_{ij} &\leq P_{ij}^1 C_1 + P_{ij}^2 C_2 \\
\sum_{i=1}^{m} x_{ij} &= a_i,\ i = 1,\ldots,m \\
\sum_{j=1}^{n} x_{ij} &= b_j,\ j = 1,\ldots,n \\
\sum_{i=1}^{m} a_i &= \sum_{j=1}^{n} b_j \\
x_{ij} &\geq 0\ \forall i,\ \forall j, \\
\text{where} \quad & P_{ij}^1,\ P_{ij}^2, i = 1,\ldots,m; j = 1,\ldots,n \text{ are integers}
\end{aligned}
$$

3. Proposed genetic algorithm

3.1. Parameters of GA

GA depends on different parameters like population size $(POPSIZE)$, probability of crossover $(PCROS)$, probability of mutation $(PMUTE)$ and maximum number of generation $(MAXGEN)$. According to the existing literature, there is no clear indication about the population size of GA(how large it should be?). However, there arises some difficulties in storing the data, if the population is too large. However, if it is too small, there may not be enough populations for good crossovers. In our present study, we have taken the values of these parameters as follows: $POPSIZE = 50$ $PCROS = 0.8$ $PMUTE = 0.1$ $MAXGEN = 500$

3.2. Chromosome representation

Spanning tree-based genetic algorithm. A TP is a kind of a network problem that its feasible solution has spanning tree topology. We encode the nodes of a transportation tree by prüfer number and is considered as a chromosome. Fig-1 illustrate a spanning tree and its prüfer number.

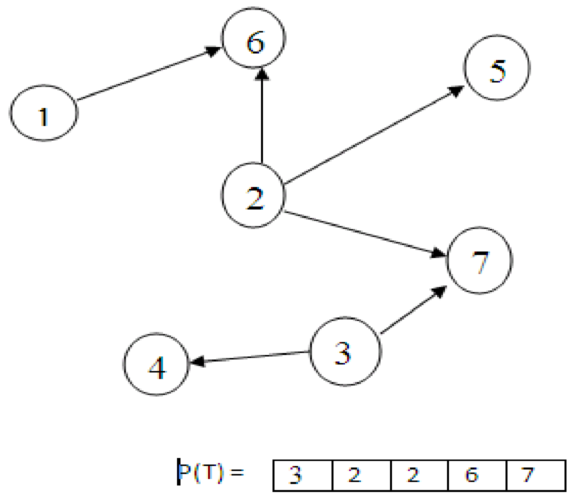

FIGURE 1. Spanning Tree and prüfer number .

3.3. Initialization

The initial chromosome is presented by a prüfer number which is performed from randomly generated $m+n-2$ digits in range $[1,m+n]$. There will be a possibility that it cannot be adopted into a transportation network graph. Due to this reason, the feasibility is checked before decoding the prüfer number.Gen and Cheng[3] developed feasibility criteria for the prüfer number. Molla-Alizadeh-Zavardehi et. al.[8] proposed a criteria for an unbalanced TP where there is no need to check or repair the invalid chromosome. The used feasibility criteria is as follows:

$$\sum_{i=1}^{m}(L_i+1) = \sum_{i=m+1}^{m+n}(L_i+1)$$

$$\text{i.e. } \sum_{i=1}^{m} L_i + m = \sum_{i=m+1}^{m+n} L_i + n \tag{2}$$

$$\text{and } \sum_{i=1}^{m} L_i + \sum_{i=m+1}^{m+n} L_i = m+n-2, \tag{3}$$

where, L_i is the appearance number of node i in the prüfer number $P(T)$.

If we consider

$$\sum_{i=1}^{m} L_i = n-1, \tag{4}$$

$$\sum_{i=m+1}^{m+n} L_i = m+n-2. \tag{5}$$

Then (3.2) and (3.3) are satisfied.

Prüfer number generation

Therefore, a prüfer number is generated by randomly selection of $(n-1)$ number of digits from set $[1,m]$ and $(m-1)$ number of digits from set $[m+1,m+n]$.

An example, is presented in the following Figure-2.

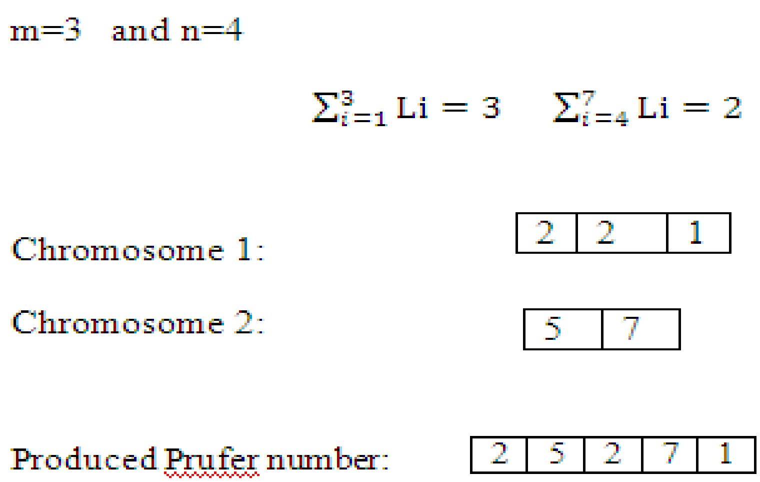

FIGURE 2. illustration of feasible prüfer number .

After generating a feasible prufer number, the transportation network graph can be determined by following decoding procedure.

3.4. Evaluation

Procedure : prüfer number to transportation tree

Step 1: Let $P(T)$ be a prüfer number, $P^c(T)$ the set of all nodes that are not part of $P(T)$ and design eligible for consideration.

Step 2: Repeat following (2.1) to (2.5)

(i) Let i be the lowest numbered eligible node in $P^c(T)$ and j be the left most digit of $P(T)$.

(ii) If i and j are not in the same set O or D, add edge (i,j) to the tree. Otherwise, select the next digit k from $P(T)$ that is not included to the same set with i, exchange j with k, and add the edge (i,k) to the tree T.

(iii) Remove $j(ork)$ from $P(T)$ and i from $P^c(T)$. If $j(ork)$ does not occur anywhere in the remaining part of $P(T)$, put it into $P^c(T)$.

(iv) Assign $x_{ij} = min(a_i,b_j)(ormin(a_i,b_k))$ to the edge $(i,j)(or(i,k))$.

(v) Set $a_i = a_i - x_{ij}$ and $b_j = b_j - x_{ij}(orb_k = b_j - x_{ik})$

Step 3: If no digits remain in $P(T)$, then there are exactly two nodes, i and j, still eligible in $P^c(T)$. Add edge (i,j) to the tree T and from a tree with $(m+n-2)$ edges.

Step 4: If there are no available units to assign, then stop.

Otherwise if, there are only one available and only one destination still remain, then add a edge between them and remove an edge which has a zero flow to avoid cycle so that the spanning tree have only $(m+n-2)$ edges. Else form a new spanning tree with $(m+n-2)$ edges.

Total transportation cost:

After determination of x_{ij} determine cost in i,j by (1) and determine the cost using $\sum_{i=1}^{m}\sum_{j=1}^{n}($ cost in $(i,j))$.

3.5. Selection

During the selection, the parent?individuals aimed at producing the child?chromosomes are chosen. The selection process works out a new population starting from current one by encouraging the chromosomes having the strongest fitness, i.e., those which are nearer to the global optima of the objective function. There exists several methods to obtain this task. In our work, we use the ranking selection method. The procedure is as follows:

Step 1:Sort all fitness $f_i, i = 1,\ldots,POPSIZE$ in descending order and change the corresponding chromosome accordingly.

Step 2: Generate a real random number c in $[0,1]$.

Step 3: Compute the probability pi of selection for each chromosome $P_i(T)$ by the formula $p_i = c(1-c)^{i-1}$.

Step 4: Compute the cumulative probability q_i for each chromosome $P_i(T)$ using the formula $q_i = \sum_{j=1}^{i} p_j$.

Step 5: Generate a random number $r \in [0,1]$.

Step 6: If $r < q_i$, then select the chromosome $P_i(T)$, otherwise go to **Step 7**.

Step 7: Repeat **Steps 5** and **6** *POPSIZE* times and obtain *POPSIZE* copies of chromosomes.

3.6. Genetic operators

Reproduction

The chromosomes with higher fitness value are more desirable, so $p_r\%$ of the chromosomes are automatically copied to the next generation.

Crossover

Crossover operator operates on two randomly selected parent chromosomes (solutions) at a time and generates offspring by combining both parent chromosomes (solutions) features. For this operation, expected $(PCROS * POPSIZE)$ number of chromosomes will take part. Hence, in order to perform the crossover operation, select $PCROS * POPSIZE$ number of chromosomes. Different steps of crossover between, $P_i(T)$ and $P_j(T)$, are given below.

Step 1: Generate a random real number $r \in [0,1]$.

Step 2: Select two chromosomes $P_i(T)$ and $P_j(T)$ randomly among populations for crossover if $r < PCROS$.

Step 3: Interchange the $k \in O$ of $P_i(T)$ with $l \in O$ of $P_j(T)$

Step 4: Repeat **Steps 1-3** for $PCROS * POPSIZE/2$ times. For example, (FIGURE 3)

Mutation. This operation is responsible for fine tuning capabilities of the system. It is applied to a single chromosome. For this operation, expected $(PMUTE * POPSIZE)$ number of chromosomes will take part. Hence, in order to perform the crossover operation, select $PMUTE * POPSIZE$ number of chromosomes. Different steps of crossover between, $P_i(T)$, are given below.

Step 1: Generate a random real number $r \in [0,1]$.

Step 2: Select chromosomes $P_i(T)$ randomly among populations for mutation if $r < PMUTE$.

Step 3: Select any k from $P_i(T)$, if it is in O then select any number l from O and replace k by

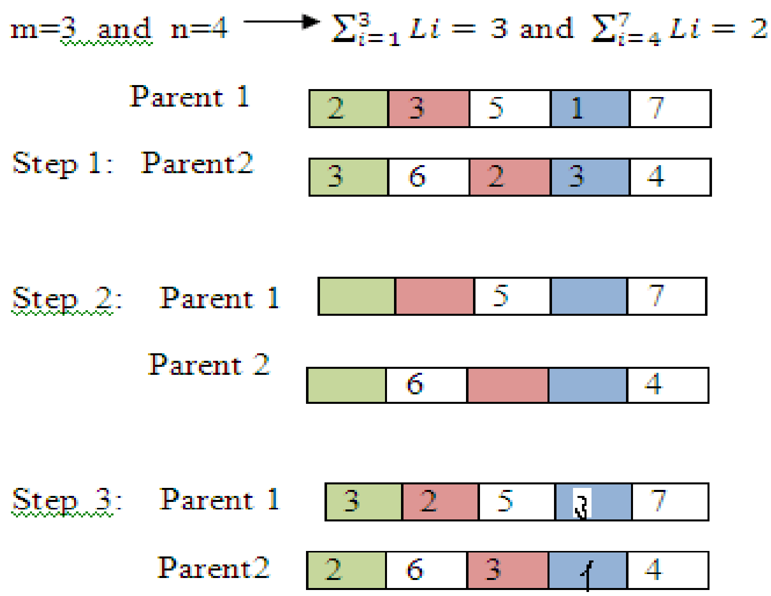

FIGURE 3. illustration of crossover procedure.

l. Otherwise select any number l from D and replace k by l.

Step 4: Repeat **Steps 1-3** for $PMUTE * POPSIZE/2$ times. (FIGURE 4)

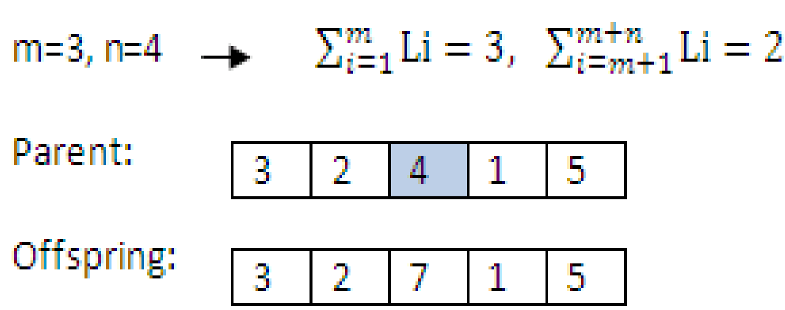

FIGURE 4. illustration of mutation procedure.

Overall procedure

The overall procedure of the spanning tree-based GA is shown in Fig.-5

4. Numerical examples

Example 4.1. Consider a cost varying transportation problem as

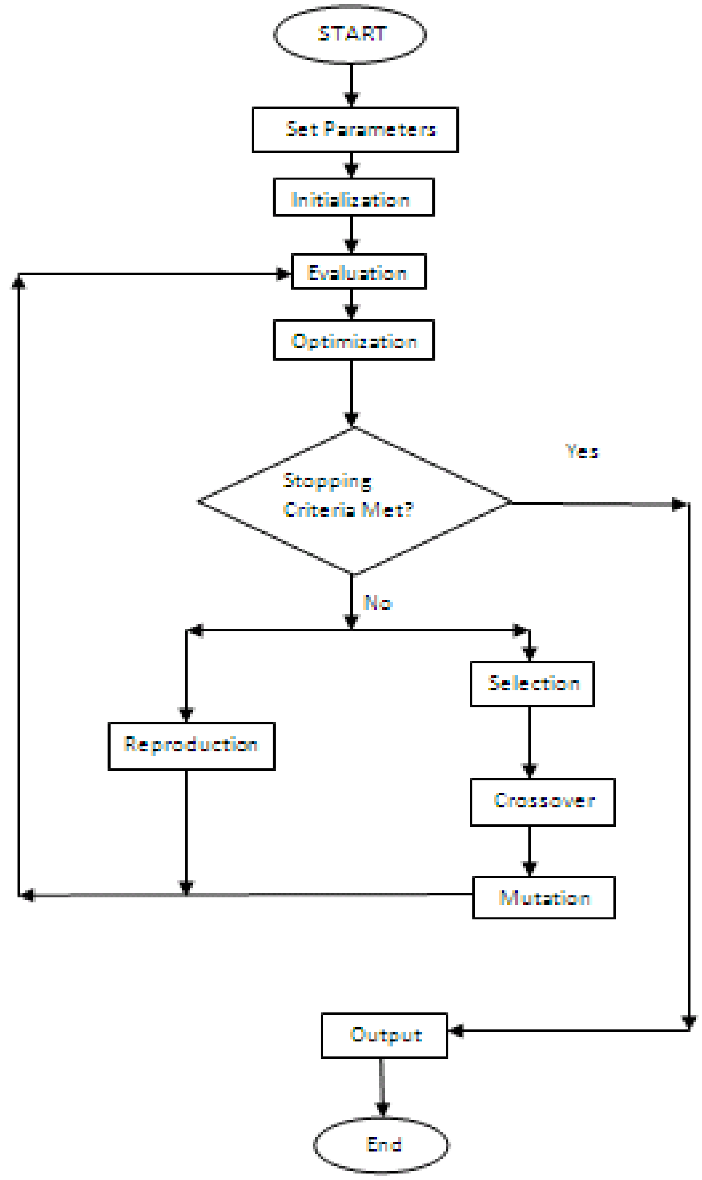

FIGURE 5. Proposed GA flowchart.

	D_1	D_2	D_3	D_4	$stock(a_i)$
O_1	$5,7$	$4,6$	$6,9$	$8,12$	30
O_2	$2,4$	$6,9$	$7,10$	$5,8$	40
O_3	$3,5$	$10,13$	$4,6$	$7,10$	20
$Demand(b_j)$	15	25	35	15	

Table 2: Tabular representation of Example 1.

The capacities of vehicles of V_1 and V_2 are respectively, $C_1 = 10$ and $C_2 = 16$.

The optimal solution of this problem is given in the following table

Popsize	*Maximum Generation*	*Prüfer Number*	*Route*	*Allocation*	*Minimum Cost*
10	20	$[2,6,7,3,1]$	$(2,4),(2,6),(3,5),$ $(3,7),(1,6),(1,7)$	$x_{12} = 25, x_{14} = 5,$ $x_{21} = 15, x_{23} = 25,$ $x_{33} = 10, x_{34} = 10,$	66
20	100	$[1,3,2,4,6]$	$(1,5),(1,4),(3,4),$ $(3,6),(2,6),(2,7)$	$x_{11} = 5, x_{12} = 25,$ $x_{23} = 25, x_{24} = 15,$ $x_{31} = 10, x_{33} = 10,$	64
50	500	$[4,1,3,6,2]$	$(1,5),(1,4),(3,4),$ $(3,6),(2,6),(2,7)$	$x_{11} = 5, x_{12} = 25,$ $x_{23} = 25, x_{34} = 15,$ $x_{31} = 10, x_{33} = 10,$	64

Table 3: Results of Example 1 by GA.

Example 4.2. Consider a cost varying transportation problem as

	D_1	D_2	D_3	D_4	*stock*(a_i)
O_1	$10,13$	$8,10$	$9,12$	$14,18$	40
O_2	$12,14$	$20,25$	$22,30$	$16,20$	50
O_3	$16,20$	$9,12$	$7,9$	$14,19$	60
Demand(b_j)	55	30	20	45	

Table 4: Tabular representation of Example 2.

The capacities of vehicles of V_1 and V_2 are respectively, $C_1 = 12$ and $C_2 = 18$.

The optimal solution of this problem is given in the following table

Popsize	*Maximum Generation*	*Prüfer Number*	*Route Route*	*Allocation Allocation*	*Minimum Cost*
10	20	$[3,3,2,4,4]$	$(1,4),(3,5),(3,6),$ $(3,4),(2,4),(2,7)$	$x_{11}=40, x_{21}=5,$ $x_{24}=45, x_{31}=10,$ $x_{32}=30, x_{33}=20,$	188
10	200	$[3,1,2,4,4]$	$(3,5),(3,4),(1,6),$ $(1,4),(2,4),(2,7)$	$x_{11}=20, x_{13}=20,$ $x_{21}=5, x_{24}=45,$ $x_{31}=30, x_{33}=30,$	183
20	200	$[4,3,2,7,1]$	$(3,5),(3,4),(2,4),$ $(2,7),(1,7),(1,6)$	$x_{13}=20, x_{14}=20,$ $x_{21}=25, x_{24}=25,$ $x_{31}=30, x_{32}=30,$	172
50	500	$[4,7,3,2,1]$	$(3,5),(3,4),(2,4),$ $(2,7),(1,7),(1,6)$	$x_{13}=20, x_{14}=20,$ $x_{21}=25, x_{24}=25,$ $x_{31}=30, x_{32}=30,$	172

Table 5: Results of Example 2 by GA.

Example 4.3. Consider a cost varying transportation problem as

	D_1	D_2	D_3	D_4	*stock(a_i)*
O_1	$10,14$	$9,12$	$25,30$	$18,22$	80
O_2	$12,146$	$10,15$	$24,28$	$20,25$	60
O_3	$11,15$	$8,10$	$30,35$	$17,20$	40
Demand(b_j)	75	50	30	25	

Table 6: Tabular representation of Example 3.

The capacities of vehicles of V_1 and V_2 are respectively, $C_1 = 15$ and $C_2 = 25$.

The optimal solution of this problem is given in the following table

Popsize	*Maximum Generation*	*prüfer Number*	*Route Route*	*Allocation Allocation*	*Minimum Cost*
10	20	$[1,2,4,3,5]$	$(1,6),(1,4),(2,4),$ $(2,5),(3,5),(3,7)$	$x_{11}=50, x_{13}=30,$ $x_{21}=25, x_{22}=35,$ $x_{32}=15, x_{34}=25,$	147
20	100	$[1,2,3,4,4]$	$(1,5),(1,4),(2,6),$ $(2,4),(3,4),(3,7)$	$x_{11}=30, x_{12}=50,$ $x_{21}=30, x_{23}=30,$ $x_{31}=15, x_{34}=25,$	147
20	500	$[1,2,3,4,4]$	$(1,5),(1,4),(2,6),$ $(2,4),(3,4),(3,7)$	$x_{11}=30, x_{12}=50,$ $x_{21}=30, x_{23}=30,$ $x_{31}=15, x_{34}=25,$	147
50	500	$[4,5,1,2,3]$	$(1,6),(1,4),(2,4),$ $(2,5),(3,5),(3,7)$	$x_{11}=50, x_{13}=30,$ $x_{21}=25, x_{22}=35,$ $x_{32}=15, x_{34}=25,$	147

Table 5: Results of Example 3 by GA.

Discussion. From the numerical results, it is seen that when the population size and the number of iterations are increased, after certain iteration, reach an optimal solution. It is also seen that there may exist many prüfer numbers give the same optimal path and unique optimal solution. It is also seen that there may exist alternative optimal solutions, i.e., prüfer numbers are different, routes are different allocations are different but optimal value is same though any basic cell does not have any zero(0) allocation.

5. Conclusion

This paper represents a new mathematical model on transportation problem where transportation cost is unknown, but the cost of transportation of the vehicles in a single trip from each source to each destination is known. This model is a bi-level mathematical programming model. It is not easy to solve by traditional method. By spanning tree based GA we solve this problem. By any finite number of vehicles this model can be generated and can be solved by our proposed methodology. This model is more efficient in reality.

REFERENCES

[1] C. Charnes, W.W. Cooper, The Stepping-stone method for explaining linear programming calculation in transportation problem, Management Sci. 1 (1954), 49-69.

[2] G.B. Dantzig, Linear Programming and Extensions, Princeton University Press, Princeton, 1963.

[3] M. Gen, R. Cheng, Genetic Algorithm and engineering optimization, NY, 2000.

[4] J. Gottlieb, L. Paulmann, Genetic algorithms for the fixed charge transportation problem, In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, IEEE Press, pp.330-335.

[5] F.L. Hitchcock, The distribution of a product from several sources to numerous localities, J. Math. Phy. 20 (1941), 224-230.

[6] Y. Li, M. Gen, K. Ida, Fixed charge transportation problem by spanning tree-based genetic algorithm, Beijing Math. 4(1998), 239-249.

[7] Z. Michalewicz, G.A. Vignaux, M. Hobbs, A nonstandard genetic algorithm for the nonlinear transportation problem, ORSA J. Comput. 3 (1991), 307-316.

[8] S. Molla-Alizadeh-Zavardehi, M. Hajiaghaei-Keshteli, R. Tavakkoli-Moghaddam, Solving a capacitated fixed-charge transportation problem by artificial immune and genetic algorithms with a Prufer number representation, Expert Systems Appl. 38 (2011), 10462-10474.

[9] F. Ortega, L.A. Wolsey, A branch-and-cut algorithm for the single commodity, uncapacitated, fixed-charge network flow problem, Networks, 41 (2003), 143-158.

[10] A. Panda, C.B. Das, 2-Vehicle Cost Varying Transportation Problem, J. Uncertain Sys. 8 (2014), 44-57.

[11] A. Panda, C.B. Das, Capacitated Transportation Problem under Vehicles, LAP LAMBERT Academic Publisher, Deutschland/Germany, 2014.

[12] J.M. Rousseau, A cutting plane method for the fixed cost problem, Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1973.

[13] M. Sun, J.E. Aronson, P.G. McKeown, D. Drinka, A tabu search heuristic procedure for the fixed charge transportation problem, Eur. J. Oper. Res. 106 (1998), 441-456.

[14] M. Sun, P.G. McKeown, Tabu search applied to the general fixed charge problems, Ann. Oper. Res. 41 (1993), 405-420.

[15] H.A. Taha, ‘Operation Research: an Introduction’, Fifth Ed., Macmillan, New York, 1992.