Solution of a Solid Traveling Purchaser Problem using Modified Genetic Algorithm and Comparison Crossover Technique

Gyei-Kark PARK[1] and Arindam Roy[2*]
[1]*Mokpo National Maritime University, South Korea and*
[2]*Dept. of Computer Science, Prabhat Kumar College, Contai, Purba Medinipur, W.B, India, 721401*

In this paper we develops a comparison crossover for Modified genetic algorithm (MGA) to solve a NP hard optimization problem. Here we consider a set of markets, a depot and some products for each of which a positive demand is specified. Each product is made available in a subset of markets in each of which only a given quantity, less than or equal to the required one, can be purchased at a given unit price. Here, this Traveling purchaser problem (TPP) searches for a cycle starting at and ending to the depot and visiting a subset of markets at a minimum traveling cost, also we introduce different vehicle to visit different markets say solid TPP (STPP). The effectiveness of our model are illustrated by numerical examples.

I. INTRODUCTION

The TSP (Traveling salesman problem) was first formulated as a mathematical problem in 1930 and became increasingly popular after 1950. TSP is a well-known NP-hard combinatorial optimization problem. Now, the traveling purchaser problem is an emerging concept in this area. In the traveling purchaser problem an agent must visit a set of outlets in-order to satisfy a minimum cost, demand requirement for the products. The cost is made up of two elements: travel cost and purchase cost. This problem is frequently faced by shoppers but its also has applications in the area of production scheduling and planning. An interesting generalization of the well-known traveling salesman problem (TSP) is the traveling purchaser problem (TPP) first introduced by Ramesh [1981]. The undirected version of this problem can be stated as follows. Consider a domicile denoted by 0, a set of markets denoted by M={1,2,........,m }, a travel cost c_{ij} on each edge (i,j) linking two markets, and a set K={1,2,........,n} of products. Denote by M_k the set of markets selling product k and by p_{ik} the price of product k at the market i. In what follows, c_{ij} must be interpreted as c_{ji} whenever i>j. The TPP is to construct a tour through a subset of the m markets and the domicile and to purchase each of the n products at one of these markets so as to minimize the sum of the travel and purchase costs. Under Ramesh's definition, it is implicitly assumed that if a product is available at a given market, its quantity is sufficient to satisfy the demand. This version of the problem is called uncapacitated traveling purchaser problem (UTPP). Laptore et. al [1987] have solved a generalization of the UTPP where the demand for product k is d_k, and the availability q_{ki} of product k at market i may be $< d_k$. This version is called the capacitated traveling salesman (CTPP).

*Communication address: Email:royarindamroy@yahoo.com

The most common TPP applications occur in vehicle routing and warehousing (Singh et. al. [1997]). Another application in the field of production scheduling is also described by Buzacott and Dutta [1971]. Here, a multi-purpose machine can assume several configurations i and each task $k \in$K must be performed using a configuration in a set M_k. Here, we consider different vehicle l=$\{1, 2...p\}$ to travel markets.

The TPP is NP-hard since it reduces to the TSP if each product is available only at one market and each market sells only one product. Now-a-days impreciseness plays a important role from our all aspects. Sometimes due to the lack of statistical data, probability theory does not work here. As a breakthrough to deal with non deterministic phenomena, especially expert data and subjective estimation, an uncertainty theory was founded by Liu[2007] and subsequently studied by many researchers.

SC is a term originally coined by Zadeh [1994, 1998]. A Genetic Algorithm (GA) is an optimization technique that is based on the evolution theory. It performs a random search having both exploitation and exploration. The first thing we must do in order to use a GA is to automatically build a set of solutions to the problem. In a TSP, every route that passes through all the cities is potentially a solution, although probably not the optimal one. Such randomly generated routes act as initial population of solutions for GA.

Many kinds of GA developed by the researchers such as Niched Pareto GA, Hybrid GA (HGA), Adaptive GA (AGA), etc, are available to get the optimal solutions in different research areas.

In the existing literature, many optimization methods, such as Simulated Annealing (SA) (Chiang & Russell1997), Tabu Search (TS) (Knoxl, 1989), Ant Colony System (ACS) (Bianchi, Dorigo & Gambardella, 2002), Genetic Algorithm (GA) (Holland, 1975), and Particle Swarm Optimization (PSO) (Eberhart & Kennedy, 1995); Marinakis & Marinakii, 2010) etc are used for TSP problems.

In this paper comparison crossover is used to solve the above model. Also probabilistic selection and conventional random mutation are consider for proposed MGA. A comparison crossover based Modified genetic algorithm (MGA) is developed with and presented here to solve this TPP. The model is illustrated by numerical examples. This paper is organized as follows: in Section-1, a brief introduction is given. In Section-2, MGA is presented. Now section-3 illustrate the solid TPP model. In Section-4 numerical experiments are performed. Finally we conclude the paper with conclusion in Section-5.

II. MODIFIED GENETIC ALGORITHM

Here we proposed GA using the probabilistic selection (Boltzmann Probability), nature based crossover and random mutation, among a set of potential solutions to get a new set of solutions. As usual, it is continued until terminating conditions are encountered. The proposed MGA and its procedures are presented below

i. Representation: Here a complete tour on M_k markets among M markets represents a solution. So an M_k dimensional integer vector $X_i = (x_{i1}, x_{i2}, ..., x_{iMk})$ is used to represent a solution, where x_{i1}, x_{i2}, ..., x_{iMk} represent M_k consecutive markets in a tour. Population size number of such solutions $X_i = (x_{i1}, x_{i2}, ..., x_{iMk})$, i = 1, 2, ..., N, are randomly generated by random number generator. Here N represents the number of chromosomes (solutions).

ii. Probabilistic Selection:

a. Probability of Selection Parameter (p_s): Here we introduce a predefined value say probability of selection parameter (p_s). For each solution of $f(X_i)$, generate a random number r from the range [0,1]. If $r < p_s$ then the corresponding chromosome is stored at matting pool.

b. Boltzmann-Probability: For minimum cost objective, it is better to choose that population which is in the neighborhood of the minimum solution of the entire solution space. So we get the convergence rate much high. From the initial population, choose the best fitted population for TPP. It is chosen as most minimum fitness value (say f_{min}). To form the matting pool, we use the **Boltzmann-Probability** of the each chromosome from the initial population.

Here $p_B = e^{((f_{min} - f(X_i))/T)}$, $T = T_0(1-a)^k$, $k = (1+100*(g/G))$, g=current generation number, G= maximum generation, T_0= rand[5,100], a=rand[0,1], $f(X_i)$ means the chromosome corresponding to X_i, i=chromosome number.

iii. Comparison Crossover:

Pseudo code of Comparison Crossover:

input: Matting Pool, p_c, Total number of node (N).
output: Offspring (child).
begin
 for (j=1; j<=N; j++) // N= total number of nodes.
  if (c (a_i, a_1) < c (a_i, s_1)) // $i \in \{1, 2, ..., N\}$, c (a_i, a_1) is the cost between nodes a_i and a_1
  {
   if (a_1 exist in Ch_1)
    {
     j++;
     compare next node from P_{r1};
    }
   else
    {
     concatenate a_1 in Ch_1;
     j++;
    }
  }
  else
  {
   if (s_1 exist in Ch_1)
    {
     j++;
     compare next node from P_{r2};
    }
   else
    {
     concatenate s_1 in Ch_1;
     j++;
    }
  }
 end for
end

During every comparison, concatenate a node such that the travel path satisfies the TSP conditions. Firstly in every comparison, check if the node already exists in the child, then the cost of the next node in modified parents will be considered i.e. repetition of the nodes are not allowed. Secondly comparison will occur until every node of the modified parents are checked i.e. every node must exist in the child.

(iv). p_m dependent Random Mutation:

a. Selection for mutation: For each solution of $p(t)$, generate a random number r from the range [0,1]. If $r < p_m$ then the solution is taken for mutation.

b. Mutation process: At first determined the total number of mutated node (T). To mutate a solution $X = (x_1, x_2, ..., x_N)$, number of mutated node T= p_m* N, N=total number of nodes.

c. Pseudo code of Mutation:

input: *pop_size*, (p_m) and total number of nodes (N).
output: Mutated offspring (child).
begin
Determine T= p_m*N // total number of mutated node
 for i=0 to *pop_size*
 r=rand(0,1)
 if(r< p_m){
  Select chromosome depending p_m
   for j=1 to T

Randomly select two different nodes between [1,N];
Swap the nodes;
end for
}
end for
end

Procedure of MGA:

procedure name: Modified Genetic Algorithm (MGA).
input: Max Gen (S_0), Population Size (pop_size), Probability of Selection (p_s), Probability of Crossover (p_c), Probability of Mutation (p_m), Problem Data (cost matrix).
output: The optimum and near optimum solutions.
1. **Start**
2. Set initial generation t ← 0.
3. (Initialization) Randomly generate initial population $p(t)$ where X_i, i=1,2...,pop$_size$ are the chromosomes, N numbers of node in each chromosome represent a solution/path of the TSP.
4. Evaluate the fitness of each solution of the initial population $p(t)$.
5. Check the condition **while** (t≤ S_0) **do** upto step 21.
6. Update the generation t ← t+1.
7. Selection Procedure.
8. Determine the Boltzmann Probability (p_B) of each chromosome of p(t)
9. Create the matting pool based on p_s and p_B.
10. Crossover Procedure.
11. Select the parents using p_c from matting pool.
12. According to Subsection 2(iii) perform the crossover operation
13. Modified the parents.
14. Generate off springs and replace the parents.
15. Repeat the Step 11 to Step 14 depend on p_c.
16. Mutation Procedure done according the Subsection 2(iv).
17. Select the off springs for mutation based on p_m.
18. Exchange the place of these nodes;
19. Store the new off springs into offspring set.
20. Compare the fitness and Store the local optimum and near optimum.
21. Repeat the Step 5 to Step 21.
22. (Optimum Solution) Store the optimum and near optimum results.
23. **Stop**.

III. PROPOSED SOLID TPP (STPP)

A. Classical TPP(2DTPP)

Now, TSP is a very well known problem. As for the TSP, a TPP modeled on a directed graph, where the cost c_{ij} is potentially different from c_{ji}, is named asymmetric (ATPP). Otherwise, if c_{ij}=c_{ji} for each arc (i,j) ∈ A, the problem is called symmetric TPP (STPP). In the literature, ATPP and STPP are often referred to as directed and undirected TPP, respectively. Another common classification concerns the availability of products at the suppliers. If the available quantity of a product $k \in K$ in a supplier $i \in M_k$ is defined as a finite value q_{ik}, potentially smaller than product demand d_k, then the TPP is called is called restricted (R-TPP). The unrestricted TPP(U-TPP), instead considers the case in which supplies are unlimited, i.e., where $q_{ik} \geq d_k$, $k \in K$, $i \in M_k$. Note that U-TPP represents a special case of R-TPP, since having unlimited supplies is equivalent to consider d_k=1 and q_{ik}=1, ∀k ∈ K and ∀$i \in M_k$. Many papers refer to R-TPP, and U-TPP as capacitated and uncapacitated TPP, respectively.

1. Symmetric TPP

The symmetric TPP is defined over a complete undirected graph $G_U = (V, E)$, where $E := \{e = (i, j) : i, j \in V, i < j\}$ is the edge set and a traveling cost c_e is associated with each edge $e \in E$. Let $x_e, e \in E$, be a binary variable taking value 1 if edge e is crossed, and 0 otherwise. Let also $\delta(V^{'}) := \{(i, j) \in E : i \in V^{'}, j \in V/V^{'}\}$ for any subset $V^{'}$ of nodes. Then, the Symmetric TPP can be defined as follows:

(Symmetric TPP) Minimize

$$\sum_{e\in E, l\in L} c_e x_e + \sum_{k\in K}\sum_{i\in M_k} p_{ik} z_{ik}, \quad L = 1, 2, \ldots\ldots, P \tag{1}$$

$$\text{subject to} \quad \sum_{i\in M_k} z_{ik} = D_k, \;\; k \in K \tag{2}$$

$$\sum_{e\in\delta(\{h\})} x_e = 2y_h \;\; h \in M \tag{3}$$

$$\sum_{e\in\delta(M^{'})} x_e \geq 2y_h \;\; M^{'} \subseteq M \;\; h \in M^{'} \tag{4}$$

$$x_e \in \{0, 1\} \;\; e \in E \tag{5}$$

$$y_i \in \{0, 1\} \;\; i \in M \tag{6}$$

$$z_{ik} \geq 0, \;\; k \in K, i \in M_k \tag{7}$$

Objective function (1) aims at the joint minimization of the traveling and purchasing costs. Equation (2) ensure that each product demand is satisfied exactly. Constraint (3) and (4) rule the visiting tour feasibility. Constraints (5) - (7) impose binary and non-negative conditions on variables. No integrality conditions are required

for z-variables, even if they actually represent the number of units purchased for each product in each supplier. If all input data are integer, in fact, then an optimal solution where all z-variables have integer values always exists.
Due to the use of an undirected graph, now in the degree constraints (3), two edges must be incident to each subset of suppliers containing a visited one. Note that this Symmetric TPP formulation does not follow solutions with less than three vertices, one being the depot. Two-vertex cycles containing the depot and one market can be easily generated and compared to the optimal solution given by the model.

B. Solid TPP (STPP)

In this section we present a solid traveling purchaser problem (STPP). In this paper, we first consider a STPP in two variables ie. traveling cost c_{el} and purchasing cost p_{ik}, the price of product k at market i. Here, c_{el} be the cost of traveling from i-th market to j-th market $\{e = (i, j)\}$ using l-th type of conveyance, $L = \{l : 1, 2, ..., P\}$. Let y_i, $i \in M$, be a binary variable taking value 1 if supplier i is selected, and 0 otherwise. Here The STPP is defined over a complete undirected graph $G_U = (V, E)$, where $E := \{e = (i, j) : i, j \in V, i < j\}$ is the edge set and a traveling cost c_e is associated with each edge $e \in E$. Let $x_e, e \in E$, be a binary variable taking value 1 if edge e is crossed, and 0 otherwise. Let also $\delta(V^{'}) := \{(i, j) \in E : i \in V^{'}, j \in V/V^{'}$ for any subset $V^{'}$ of nodes. Then, the STPP can be defined as follows:

(SGTPP) Minimize

$$\sum_{e\in E, l\in L} c_{el}x_e + \sum_{k\in K}\sum_{i\in M_k} p_{ik}z_{ik}, \quad L = 1, 2, ..., P \tag{8}$$

$$\sum_{i\in M_k} z_{ik} = D_k, \;\; k \in K \tag{9}$$

$$\sum_{e\in\delta(\{h\})} x_e = 2y_h \;\; h \in M \tag{10}$$

$$\sum_{e\in\delta(M')} x_e \geq 2y_h \;\; M^{'} \subseteq M, h \in M^{'} \tag{11}$$

$$x_e \in \{0, 1\} \;\; e \in E \tag{12}$$

$$y_i \in \{0, 1\} \;\; i \in M \tag{13}$$

$$z_{ik} \geq 0, \;\; k \in K, i \in M_k \tag{14}$$

Objective function (8) aims at the joint minimization of the traveling and purchasing costs. Equation (9) ensure that each product demand is satisfied exactly. Constraints (13) and (14) rule the visiting tour feasibility. Constraints (13) to (14) impose binary and non-negative conditions on variables.

IV. NUMERICAL EXPERIMENTS

A. Testing for proposed GA

The performance of the proposed algorithm MGA was found for 8 standard benchmarks using TSPLIB [1995]. Table 1 gives the results of MGA. The results are compared in terms of total cost. Under 25 independent run, the average results, best found results are presented here. We assume that each product must be available at the probability equal to 70% to be sold at a market, that is we build a tour from the TSPLIB by choosing 70% cities (markets).

Table 1: Results for TPP of single product with standard TSP data

Instances	No. of markets visit	No. of units purchased	Unit cost Unit cost Unit cost	Average Result	Best Result
bayg29	20	101	31	4593	4500
bays29	20	105	31	5050	4991
gr21	14	71	31	3722	3679
gr17	11	57	31	2608	2598
gr48	33	167	31	11035	10811
ulysses16	11	57	31	4632	4632
swiss42	29	147	31	6276	6240
brazil58	40	147	31	26899	23298

The parameters of the MGA are set as in Table 2 for different nodes of the TSP. As the size of the TSP increases pop$_{size}$, Maxgen, maximum initialization, number of initialization for convergence of the optimal solution.

Table 2: Parameters for MGA and simple GA

Size (N)	Maxgen	Number of Initialization	Maximum Initialization	Pop -size	p$_c$	p$_m$
N$\leq$ 20	200	80	120	30	.45	0.35
20$< N \leq$ 30	300	120	180	50	.45	0.3
30$< N \leq$ 40	400	200	300	80	.51	0.35
40$< N \leq$ 50	500	200	400	100	.51	0.4
40$< N \leq$ 60	600	250	450	100	.51	0.45

Table 3: Results of 2-dimensional TPP in Crisp Environment

Algorithm	Path	Total Cost	Total units
MGA	3-1-5-7-2-4-6	1489	35
	6-8-9-1-2-3-10	1482	35
	4-5-1-7-9-2-8	1481	35
	7-4-2-6-1-3-9	1490	35
	6-10-7-5-8-6-1	1483	35
GA	2-1-7-4-3-5-6	1495	35
	4-8-3-5-1-10-4	1494	35
	10-4-2-8-5-9-1	1498	35
	6-1-4-9-2-8-10	1493	35
	3-10-5-6-1-7-8	1496	35

Table 4: Input Data: Crisp STPP

	Crisp Cost Matrix(10*10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(27,28,23)	(28,29,32)	(50,52,45)	(23,25,26)	(15,16,17)	(35,39,33)	(34,33,38)	(49,38,45)	(37,35,43)
2	(28,29,24)	∞	(22,29,27)	(28,32,30)	(35,31,39)	(42,46,41)	(33,36,35)	(40,42,39)	(25,28,24)	(15,17,16)
3	(45,43,48)	(30,27,28)	∞	(41,39,37)	(28,22,29)	(39,28,34)	(30,29,23)	(33,32,35)	(39,40,41)	(27,38,34)
4	(29,27,28)	(32,35,37)	(18,17,19)	∞	(30,29,23)	(35,22,29)	(33,30,29)	(31,35,36)	(27,28,29)	(36,39,25)
5	(36,57,39)	(22,25,12)	(36,29,31)	(29,30,25)	∞	(25,24,22)	(27,25,29)	(35,34,29)	(18,19,16)	(21,29,23)
6	(28,22,26)	(34,29,29)	(28,26,28)	(18,19,17)	(38,29,34)	∞	(35,33,37)	(43,45,43)	(39,37,36)	(33,30,31)
7	(28,26,25)	(37,25,29)	(30,32,33)	(25,20,29)	(43, 37,45)	(40,45,36)	∞	(19,18,17)	(35,25,21)	(25,22,19)
8	(24,26,17)	(18,15,19)	(30,35,31)	(36,33,39)	(35,29,28)	(25,26,27)	(41,36,25)	∞	(22,26,19)	(36,29,26)
9	(35,32,34)	(38,37,40)	(34,36,33)	(25,26,27)	(20,22,12)	(20,18,21)	(31,25,28)	(32,33,36)	∞	(27,28,25)
10	(27,26,32)	(34,35,33)	(27,28,29)	(22,24,17)	(15,16,14)	(32,35,33)	(27,16,19)	(42,43,50)	(37,27,23)	∞

Table 5: Results of 3-dimensional STPP in Crisp Environment

Algorithm	Path(Vehicle)	Total Cost	Total units
MGA	9(1)-4(2)-10(3)-5(2)-1(2)-2(2)-7(1)	1512	35
	8(2)-6(2)-9(2)-3(1)-2(2)-5(3)-7(2)	1509	35
	5(1)-3(3)-1(3)-10(2)-9(2)-7(2)-2(1)	1498	35
	4(1)-9(3)-2(2)-10(3)-3(2)-7(3)-6(3)	1502	35
	4(1)-5(2)-3(1)-6(3)-10(2)-9(1)-7(2)	1525	35
GA	5(1)-3(2)-6(1)-8(3)-4(1)-2(1)-1(2)	1536	35
	5(1)-10(1)-2(3)-3(1)-7(1)-4(2)-8(1)	1554	35
	9(2)-5(2)-10(2)-7(1)-1(1)-3(3)-6(1)	1530	35
	8(1)-2(1)-6(1)-7(2)-9(1)-5(1)-3(3)	1595	35
	4(2)-5(1)-6(3)-10(2)-5(1)-3(2)-7(3)	1606	35

B. STPP in Crisp Environment

Now for a 3DTPP, where we consider three types of conveyances. The cost matrix for the 3DTPP is represented in Table 4.

For the above results, we consider maximum generation=1000. The problem is solved by MGA and simple GA the results are presented in Table 3 and here we consider first vehicle type through out the tour.
For the above results, we consider maximum generation=1000. The problem is solved by MGA and simple GA the results are presented in Table 5.

V. CONCLUSION

In this paper, a virgin comparison crossover for GA is proposed to solve an solid travelling purchaser problem. The proposed GA is tested with standard TSPLIB instances with considering maximum markets/supplier of the problem. The Solid TPP is introduced in the area of TPPs and regarded as highly NP-hard combinatorial optimization problems. Such STPP are here formulated in crisp costs and solved by the proposed GA. In future STPP with time window may consider. Here, development of GA is in general form and it can be applied in other discrete problems such as network optimization, graph theory, solid transportation problems, vehicle routing, VLSI chip design, etc. Again STPP can be used in production planning, equipment purchasing for multi national company project and many practical problem can be model solved.

[1] Bianchi L., Dorigo M. and Gambardella L. M., *Ant colony optimization approach to the probabilistic travelling salesman problem, PPSN VII, LNCS*, **2439**(2002), 883-892.

[2] Buzacott JA, Dutta SK, *Sequencing many jobs on a multipurpose facility*, Naval Research Logistics Quaterly, , **18**(1971), 75-82.

[3] Chiang W. C. and Russell R. A., *Simulated annealing meta heuristics for the vehicle routine problem with time windows, Annals of Operations Research*, **63**(1997), 3-27.

[4] Eberhart R.C. and Kennedy J., *Particle Swarm Optimization, In: Proc. IEEE Conference On Neural Networks*, **4**(1995), 1942-1948.

[5] Holland, J. H, *An efficient genetic algorithm for the traveling salesman problem, European Journal of Operational Research*, **145**(1975), 606-617.

[6] Knoxl J., *The application of Tabu search to the symmetric traveling salesman problem, Ph.D.Dissertation, University of Colorado*, (1989).

[7] Laptore G., Riera-Ledesma J, Salazar-Gonzalez J-J, (2000), *A branch-and-cut algorithm for the undirected traveling purchaser problem*, Opsearch, 51(6), (2000), 940-951.

[8] Marinakis Y. and Marinakii M., *A Hybrid Multi-Swarm Particle Swarm Optimization algorithm for the Probabilistic Traveling Salesman Problem*, Computers and Operations Research, **37(3)**(2010), 432-442.

[9] Ramesh T., (1981), *Traveling purchaser problem*, Opsearch, **18**, 78-91.

[10] Singh KN, Van Oudheusden DL, (1997), *A branch and bound algorithm for the traveling purchaser problem*, European Journal of Operational Research, **97**, 571-9.
[11] Zadeh, L. A., *Fuzzy Logic and Soft Computing: Issues, Contentions and Perspectives, In Proc. of IIZUKA'94: Third Int. Conf. on Fuzzy Logic, Neural Nets and Soft Computing*, **1-2**(1994), Iizuka, Japan.
[12] Zadeh, L. A., *Some reflection on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems, Soft Computing A Fusion of Foundations, Methodologies and Applications.* **2(1)**(1998), 23-25.