Volume 6, Issue 5, May 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Improvement in Shortest Route Path: A Technique to Solve Travelling Salesperson Problem Using Genetic TSPGO-Dijkstra

Preeti Singh
Dept. of Computer Science & M.TECH, IPEC (UPTU)
Uttar Pradesh, India

Alok Katiyar
Dept. of Computer Science & Asst. Prof – IPEC (UPTU)
Uttar Pradesh, India

Abstract— ***Every Traveling salesman problem (TSP) is shown to be NP-complete in most cases. The genetic algorithm (GA) is unique technique algorithms for solving this problem. The main approach is mining all sequential groups with cities of samples and changing the two central cities with each other. The next local optimization policy is similar to an extra mutation process. In this step with a low possibility a sample is designated. In this section four random cities are definite and the path between these cities is reversed. The calculation results display that the proposed method also discoveries better paths than the conventional GA within a suitable computation time. In this paper a conventional GA is compared with an improved hybrid TSPGO-Dijkstra in solving TSP. The enhanced or hybrid GA involve of conventional GA and two local optimization approaches contain the conventional GA and two local optimization approaches. We progress the TSPGO (travelling salesperson genetic optimization) then associate with existing TSPGA (travelling salesman genetic optimization with Dijikstra) at different number of generation and population size and a matrix illustration of the graph give the best results which is simulated in MATLAB tool.***

Keywords— Traveling salesman problem, genetic optimization algorithms optimize population, Dijikstra etc.

I. INTRODUCTION

We present and test a new method for the bi-objective routing problem known as the traveling salesman problem with profits. This problem compacts with the optimization of two incompatible objectives: the minimization of the tour length and the enlargement of the composed profits. This problem has been deliberate in the form of a single objective problem, where any the two objectives have been mutual or one of the objectives has been conserved as a constraint. The purpose of our study is to find keys to this problem using the notion of Pareto optimality, i.e. by searching for efficient solutions and constructing an effectual border. We have established an ejection chain local search and collective it with a multi-objective evolutionary algorithm which is used to produce expanded starting solutions in the neutral space.

A. Improvement Procedure for the TSPP (IP-TSPP)

The development process that we have advanced is similar to the PSEC LS proposed by Rego [9] for the TSP. A high level explanation of the improvement process is shown in Algorithm.

IP-TSPP precedes as an input a solution *S* for the TSPP. First, it calculates the goal point permitting to *S*. At each iteration it chooses a new node to become the root node of the stem-and-cycle process. This is essential as the traveled neighborhood highly depends on the meaning of the root node. The procedure is iterated till all the nodes have been used as the root without discovery a solution closer to the goal point. When an improved solution is found, the set of conceivable root nodes is rearrange so that every node can be chosen over as a root. A solution *s*1 is said to be better than another solution *s*2 if the Euclidean distance between *F (s*1*)* and *g* is smaller than the Euclidean distance between *F(s*2*)* and *g*. The search also appraises an archive *A* which covers the non-dominated solutions originate during the search. This collection is the final result of IP-TSPP. It permits finding solutions gained thanks to an oscillation around the local best found by the process.

Algorithm 1 IP-TSPP (Solution S)

Compute the goal point g according to S.
S∗ ← S
C ← V (C is the set of candidates to become the root node)
A ← ∅ (A is the archive of the non-dominated solutions found)
While C = ∅ do
Select vr randomly from C and set vr as the root node
C ←C \ {vr}
S' ←core_step(S∗, g, vr , A)
Apply a trial move on S_ (i.e. build a feasible tour as explained in next section)
if S' is better (i.e, closer to g) than S∗ then
S∗ ← S'
C ← V

end if
end while
Return A

The core step of IP-TSPP is detailed in Algorithm 2. First, the stem-and-cycle structure is completely defined by making the root node also the tip node. The starting solution is saved as the best solution found so far and as the current solution.

Algorithm 2 core_step (Solution *S*, Goal *g*, Root *vr* , Archive *A*)

vt ← *vr* (Initially, the root is also the tip)
k ← 1
S∗ ← *S*
Sc ← *S*
repeat
Find the best feasible move on *Sc*, i.e., themoves allowing the smallest Euclidean
distance between the resulting solution and *g*
Perform the move on *Sc*
Update the tabu short term memory structure
Try to include *Sc* in *A*
if *Sc* is better than *S*∗ **then**
S∗ ← *Sc*
end if
k ← *k* + 1
until *k* > *k*max
Return *S*∗

Then, the process is run for k_{max} iterations or until no move is possible. k_{max} is fixed to *n*. The move to perform is chosen first by considering the first set of moves and then the second set of moves. The chosen move is the one which provides the closest point in the objective space to the goal point when it is combined with a trial move. In case of a tie, the move from the first set is preferred; otherwise the choice is made randomly.

B. Dijkstra Algorithm

Dijkstra algorithm is also identified as single source shortest path algorithm. It computes the shortest path from the source to each of the unvisited vertices in the graph [1]. Dijkstra Algorithm used the technique of increasing node by node to get a shortest path tree which creates the starting point as its root. Here is the real method: in the Weighted directed graph, the shortest path node which Starts from the starting point s and ranges the initial must be the minimum point where all the nodes together to s and its length of arc is Chord length [4]. if the vertices of the graph signify cities and edge path costs signify driving distances between pairs of cities associated by a direct road, Dijkstra's algorithm can be used to discovery the shortest route between one city and all other cities if the vertices of the graph signify cities and edge path costs signify driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities.

C. Genetic Coding

To put on GA for any optimization problem, one has to reason a way for encoding solutions as possible chromosomes so that the crossovers of possible chromosomes result in feasible chromosomes. The methods for encoding solutions vary by problem and, include a positive amount of art. For the TSP, solution is classically signified by chromosome of length as the number of nodes in the problem.

Each gene of a chromosome takings a label of node such that no node can look twice in the same chromosome. There are principally two representation methods for representing tour of the TSP – adjacency representation and path representation. We deliberate the path representation for a tour, which basically lists the label of nodes. For example, let {1, 2, 3, 4, 5} be the labels of nodes in a 5 node instance, then a tour {1→ 3→4→ 2→ 5 →1} may be represented as (1, 3, 4, 2, 5).

II. PROBLEM STATEMENT

In nature, there exist numerous procedures which seek a stable state. These procedures can be seen as natural optimization procedures. Over the last 30 years, numerous efforts have been made to mature global optimization algorithms which pretend these natural optimization processes.

There are mostly three reasons why TSP has been involved the attention of many researchers and rests a vigorous research area. First, a large number of real-world problems can be demonstrated by TSP. Second, it was demonstrated to be NP-Complete problem. Third, NP-Complete problems are obstinate in the sense that no one has found any actually efficient way of solving them for large problem size. Also, NP complete problems are known to be more or less corresponding to each other; if one knew how to solve one of them one could solve the rest. An understood way of solving the TSP is basically to list all the feasible solutions, assess their objective function values and pick out the best. Though it is obvious that this "exhaustive search" is grossly incompetent and impracticable because of vast number of possible solutions to the TSP even for problem of moderate size.

III. SYSTEM MODEL

In universal TSP presented as a complete graph G (V, E), where V signifies cities and E represents different Path between cities correspondingly. Each edge has some values we call them, cost on the Edges for a graph G with *n* vertices. We have total of *n*× (*n*-1)/2 edges in this travelling Salesman Problem. We use frequency graph method to decrease the travelling salesman graph into sub graph by using optimal Hamiltonian circuit. A Hamiltonian circuit (HC) allowing to standard definition including *n* vertices once and precisely once is signified as HC= (*v*1, *v*2, *v*3,…, *v*i,…, *v*n, *v*1) and the two end vertices are identical.

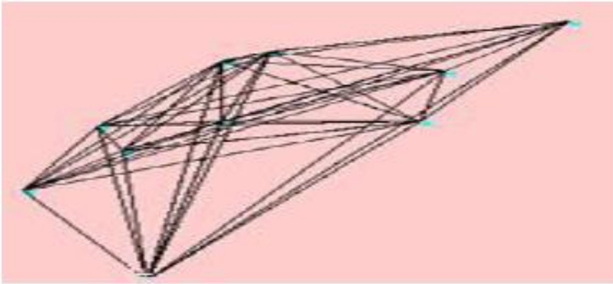

Fig.1 complete graph of travelling salesman problem with n vertices.

According to definition of complete graph, as we take the TSP in this paper as a complete graph So if Vertices n=10, Edges are n*(n-1)/2 i.e. 45 edges. Then It's sub graph can be as following:

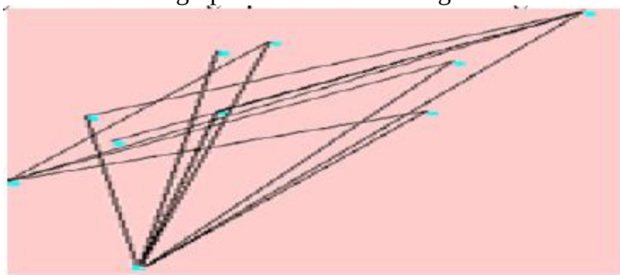

Fig 2 sub graph of complete graph of travelling salesman problem with n vertices.

IV. PROPOSED METHOD

A. Randomized Genetic optimization

Ongoing on the randomized path will take us to randomized Genetic optimization (RGO). RGO work in a way similar to nature. An evolutionary process takes place within a population of candidate solutions.

A basic GO starts out with a randomly generated population of candidate solutions. Some (or all) candidates are then reproduced to crop offspring and some go through a mutating process. Each applicant has a fitness value telling us how good they are. By choosing the fit candidates for mating and mutation the general fitness of the population will increase. Smearing RGO to the TSP includes applying a crossover routine, an amount of fitness, and also a mutation routine. A good measure of fitness is the actual length of the candidate solution.

The pseudo code of our proposed method RGO

1. Choose initial paths
2. Evaluate each path's length
3. Determine path's average length
4. Repeat
5. Select best-ranking paths to reproduce
6. Mute pairs at random
7. Apply crossover operator
8. Apply level 1 mutation operator
9. Apply level 2 mutation operator
10. Evaluate each path's length
11. Determine path's average lengths
12. Until terminating condition
    (E.g. until at least one path has the desired length or enough generations have passed)

B. Proposed Method Of Dijkstra With Genetic Optimization Algorithm

1. Initialize number of cities.
2. Initialize initial speed range
3. Compute the sum and average route of the path
4. The current cities n1 another field grid (n1) is added (means according to cities should be exist
5. Calculate typical size of considered area for cities if (the number of route greater to up-coming cities route) Then

6. The maximum route for all the paths, denoted respectively by seqnum and advertised-count if (the minimum distance of its neighbors for multi length path is greater to up comming route distance)
   Then
7. Intermediate route receives an RREQ, it increases the field esum (Pi(N0, Ni)) by the value of its finder route (means consider 2 condition concurrently)
8. esum- sum of route
9. Pi= probability of ith iteration
10. i=iteration
11. if (the maximum optimality (number of success route discovery) of its neighbors is greater to other rout)
    Then
12. RREQ message received with a sequence number and it's received in any previous RREQ message toward that destination is discarded (of highest weight route))
13. Weight should be proportional to path of repeated sending
14. Now counter for bit transmitted to Bases Station using route matrix
15. Now node forwarding data to neighbored distance sink
16. If (direct transmission if route of cities decreases below threshold)
    Then
17. Equalize upper and lower triangular matrix of route should be max infinity
18. if (route weight is smaller than previous value discard maximum distance toward base station of whole route)
    Then
19. Select next route for direct route finding and back track
20. Repeat steps until number of generation execution with number of round

V. RESULT

The results will obtain with the newly designed genetic operators in our algorithm are impressive, on practical random no of cities. Larger targets are to be tested next. This method can be easily adapted to solving the asymmetric TSP. Experiments on comparing those results with other existing solvers for asymmetric TSP also need to be performed. Application of the developed TSPGO-Dijkstra to real life problems like time complexity fragment assembly.

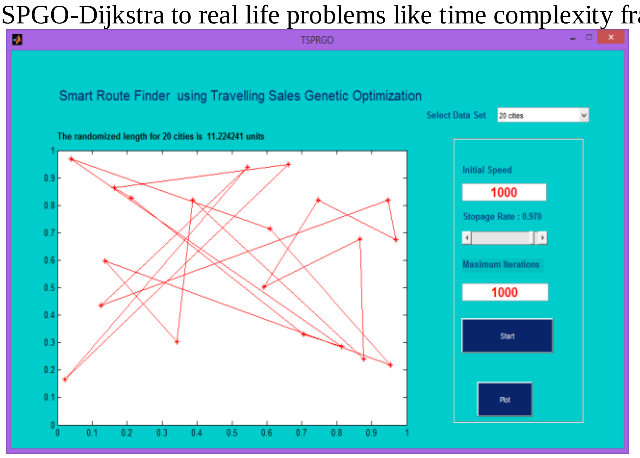

Fig. 5.1 TSP using Genetic optimization for 20 cities with 8.93 units efficiency

In above figure 5.1, we take the 20 cities from the dataset for optimize the route, its initial speed is 1000 and maximum optimization is 1000.

The experiments show that GOA is the worst approach after all. Since GA can fail in any cases, it is definitely unexpected finding, and the propose method absolutely improves the performance of GA. In the other side, in small amount of data, experiments show that GOA did not just give the same solution as TSPGO, but it also gave better solution. It does not mean that GA will constantly perform better than GOA because small data merely reflects simple problem. That advance method is not effective to be applied for simple problem, it is a common sense.

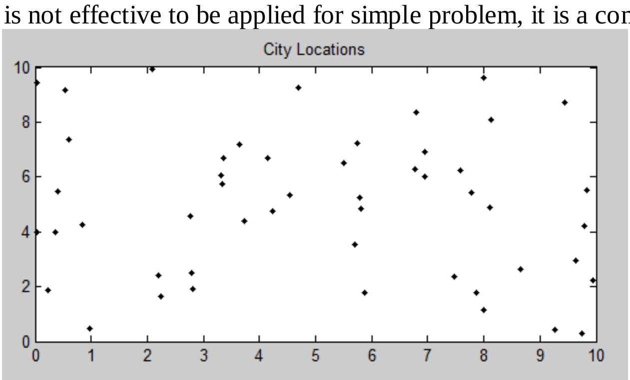

Figure 5.2TSP City location for Genetic optimization for 20 cities

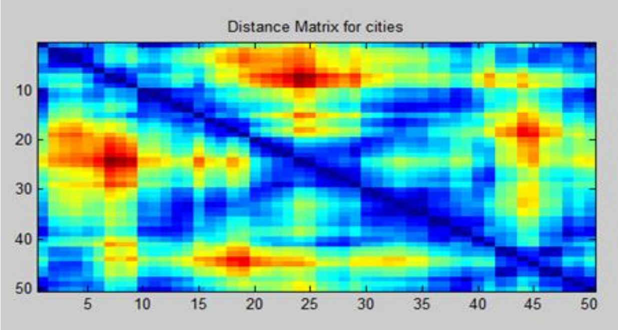

Figure 5.3 TSP Distance Matrix for Genetic optimization for 20 cities

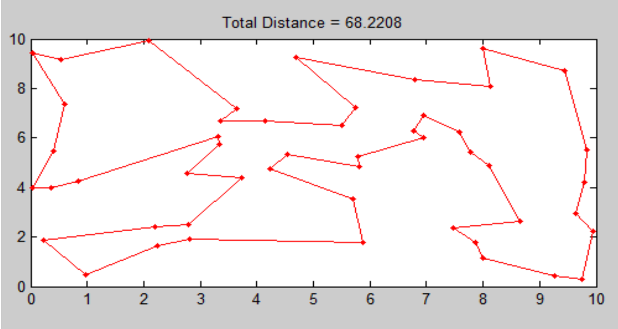

Figure 5.4 Total distance of TSP for Genetic optimization for 20 cities

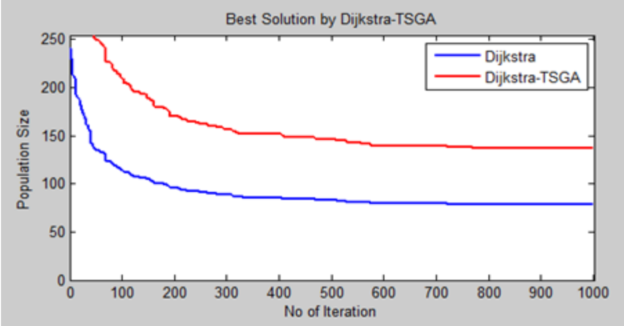

Figure 5.5 Comparison of Dijkstra and Dijkstra-TSGA for best solution

Figure 5.6 TSP using Genetic optimization for 29 cities with 10525.82 units efficiency

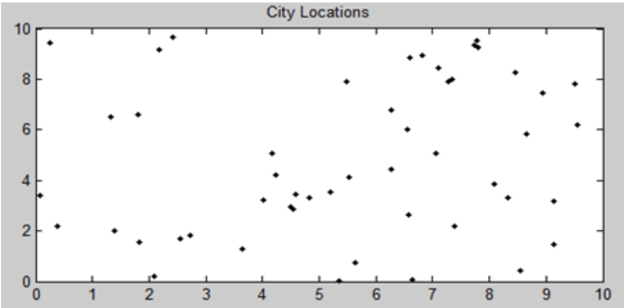

Fig.5.7 TSP City location for Genetic optimization for 29 cities

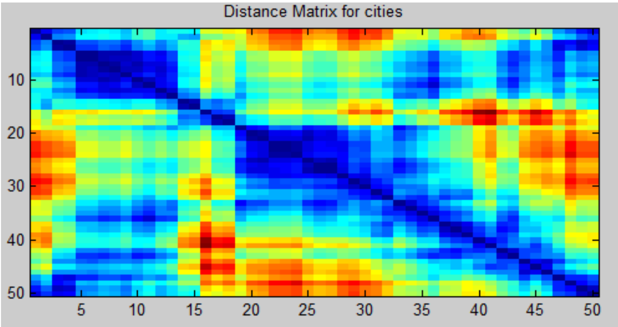

Fig. 5.8 TSP Distance Matrix for Genetic optimization for 29 cities

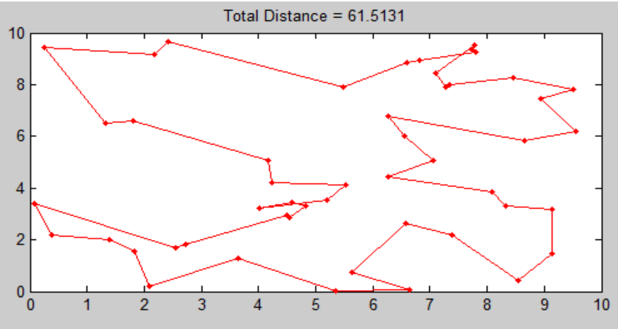

Fig.5.9 Total distance of TSP for Genetic optimization for 29 cities

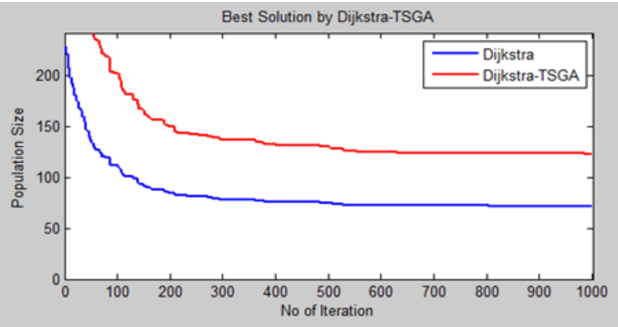

Fig. 5.10 Comparison of Dijkstra and Dijkstra-TSGA for best solution

Fig 5.11 TSP using Genetic optimization for 48 cities with 42370.21 unit's efficiency

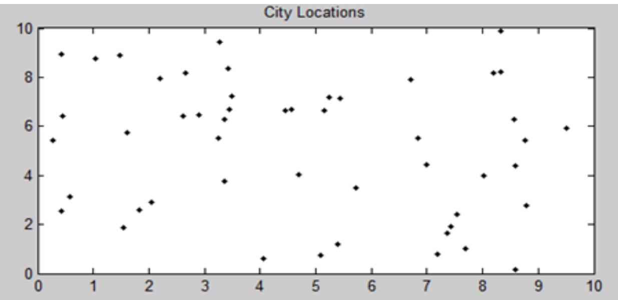

Fig 5.12 Tsp city location for Genetic Optimization for 48 cities

It has been observed from the Figures that the proposed TSPGOA has outperformed genetic algorithm in terms of convergence and optimal solution. The proposed crossover maintains more diversity in population and prevent algorithm to stick in local optima and genetic drift problem. The genetic algorithm usually results in premature convergence due to finite population size. But in proposed crossover, due to the incorporation of local search after normal crossover, the more fit offspring are generated that accelerates the search towards optima.

VI. CONCLUSION

This paper compares the proposed hybrid TSPGO-Dijkstra with pure GO. It was originated that proposed hybrid GA is actual in terms of convergence near the optimal result. The projected variation of crossover and mutation has the advantage of retaining the good solutions by using the value of fitness function.

The assistances to this field cover the growth of a TSPGO-Dijkstra process to solve TSP, the analysis and development of newly proposed algorithm, and the exploration of new applications of the precise sequencing problem. From the analysis and investigational work, the main assistances of the research can be summarized as follows:

1. Developed TSP procedure – developed a clear genetic algorithm procedure for TSP in which route repair based topological sort is inserted in the procedure in order to generate only feasible chromosomes.

2. Developed fitness evaluation procedure for TSP the objective function to evaluate the fitness of each chromosome has been developed.

3. Improved TSPGO-Dijkstra performance – the proposed algorithm has used 'earliest position' selection of tasks in order to reduce iteration time, hence improved GA performance.

4. Improved quality of the solution – the proposed algorithm has used simple linear order crossover and inversion mutation to introduce new fitter chromosomes from generations to generations in order to prevent premature convergence.

In future proposed approach can be tested and implemented on different NP Hard problems like vehicle routing problem, Job Shop Scheduling problem and knapsack problem. Study can also be conducted to analyze the performance of algorithm with hill climbing is applied to different percentage of population.

ACKNOWLEDGMENT

I am grateful to numerous local and global peers who have contributed towards shaping this thesis. At the outset, I would like to express my sincere thanks to Assistant prof., Mr. Alok katiyar for his advice during my thesis work. As my supervisor, he has constantly encouraged me to remain focused on achieving my goal. His observations and comments helped me to establish the overall direction of there search and to move forward with investigation in depth. He has helped me greatly and been a source of knowledge.

I am really thankful to my all friends and to everyone who has provided me with kind words, a welcome ear, new ideas, useful criticism, or their invaluable time, I am truly indebted.

I would also like to thank my parents and sister for standing beside me all the time and support me morally and ethically.

I must acknowledge the academic resources that I have got from IPEC.I would like to thank administrative and technical staff members of the department who have been kind enough to advise and help in their respective roles.

Last, but not the least, I would like to dedicate this thesis to my family, for their love, patience, and understanding.

REFERENCES

[1] ChetanChudasama, S. M. Shah and Mahesh Panchal, "Comparison of Parents Selection Methods of Genetic Algorithm for TSP", International Conference on Computer Communication and Networks (CSI- COMNET), 2011.

[2] Dwivedi, TarunaChauhan,SanuSaxena and PrincieAgrawal, "Travelling Salesman Problem using Genetic Algorithm", International Journal of Computer Applications(IJCA), 2012, pp. 25-30.

[3] Naveen kumar, Karambir and Rajiv Kumar, "A Genetic Algorithm Approach To Study Travelling Salesman Pr oblem", Journal of Global Research in Computer Science, 2012, Vol. 3, No. (3).

[4] Adewole Philip, AkinwaleAdioTaofiki and OtunbanowoKehinde, "A Genetic Algorithm for Solving Travelling Salesman Problem", International Journal of Advanced Computer Science and Applications, 2011, Vol. (2), No. (1).

[5] Ivan Brezina Jr.,ZuzanaCickova, "Solving the Travelling Salesman Problem using the Ant colony Optimization", Management Information Systems, 2011, Vol. (6), No. (4).

[6] ButhainahFahran, Al-Dulaimi, and Hamza A. Ali, "Enhanced Traveling Salesman Problem Solving by Genetic Algorithm Technique (TSPGA)", World Academy of Science, Engineering and Technology, 2008, Vol. (14).

[7] Rong Yang, "Solving Large Travelling Salesman Problems with Small Populations". IEEE 1997

[8] Chiung Moon, Jongsoo Kim, GyunghyunChoi ,YoonhoSeo," An efficient genetic algorithm for the traveling salesman problem with precedence constraints", European Journal of Operational Research 140 (2002) 606–617, accepted 28 February 2001.

[9] ShubhraSankar Ray, SanghamitraBandyopadhyay and Sankar K. Pal," New Operators of Genetic Algorithms for Traveling Salesman Problem", 2004 IEEE.

[10] Lawrence V. Snyder a,*, Mark S. Daskin ," A random-key genetic algorithm for the generalized traveling salesman problem", European Journal of Operational Research 174 (2006) 38–53, 2005.