An Efficient Hybrid Genetic Algorithm for Performance Enhancement in solving Travelling Salesman Problem

Navjot Kaur Dalip
Lovely Institute of Technology
Punjab Technical University
Jalandhar, India
E-mail: navjot.dalip@gmail.com

***Abstract*— This paper, proposes a solution for Travelling Salesman Problem (TSP) [1], using Genetic Algorithm (GA). The proposed algorithm works on data sets of latitude and longitude coordinates of cities and provides optimal tours in shorter time; giving convergence that is fast and better. To improve the solution few heuristic improvements are applied to prevent converging to local optima. The principle of natural selection here is based on both survival and reproduction capacities; that accelerate the convergence speed. Various factors affect the performance of GA(s), such as genetic operators, population etc. As the performance of GA is greatly affected by the initial population, the initial population for the algorithm is sorted first, using Quick Sort, this preserves the better fit population. Also, GA parameters such as selection and mutation probabilities are varied, to obtain enhanced and better performance. The computational results are compared with symmetric problems for some benchmark TSP LIB instances.**

***Keywords*-Genetic Algotihm(GA), Travelling Salesman Problem(TSP), Heuristic, Optimization Mutation.**

I. Introduction

The Travelling Salesman Problem is a classical NP-hard [8-9] optimization problem in which, a set of points and distance between the points is given and the shortest path that visits each point exactly once and returns to the initial point, is calculated.

TSP has always been an active research area and many researchers have focused their attention for this work, for various reasons, of which the main reasons are: (1) A large number of real world problems can be modeled by TSP. (2) It is an NP-complete problem. NP-complete problems are stubborn because no one has found any really efficient way of solving them for large problem size. For large populations, solution space for TSP grows exponentially; so it requires prodigious amount of computation resources.

Researchers have suggested several heuristic algorithms such as Genetic Algorithm (GA) [2-3, 12], Tabu Search [5], Neural Network [6], Ant Colony [7], Elastic Nets and Threshold Acceptance have been suggested to solve this problem. Among all these soft computation methods, GA has gained more attention due to its good performance in finding an optimal solution in small computation time. The TSP finds application in a variety of situations such as problem of computer wiring, scheduling of machines to drill poles in a circuit boards, transportation, logistics applications, X-ray crystallography etc.

The purpose of this paper is to develop a GA that solves the TSP in acceptable time. It involves generating random initial population, sorting it and dispensing the bad population with large distances. So, the population with higher fitness (small distances) is retained, providing us a better solution.

The rest of the paper is organized as follows. In the next section TSP and GA are reviewed briefly. Section III basic idea and describes the proposed algorithm process. Section IV presents the Experimental results of the algorithm. The final section contains conclusions and scope of the work.

II. Literature review

Travelling Salesman Problem (TSP) is one of the most compelling optimization problems. TSP was first proposed by W.R. Hamilton and the British Mathematician Thomas Kirkman.

It can be described as: In the graph G= (V,E), V is the set of nodes, or, cities, E is the set of edges, path. E= $\{(a,b)|a,b \in V\}$. The Euclidean distance between a and b is D_{ab}, assuming that $D_{ab}=D_{ba}$. The objective is to find a shortest length of closed tour, during which the travelling salesman starts at one city, visit each of the cities only one time each and finally returns to the starting point [10]. The closed tour is called Hamilton Cycle.

Genetic Algorithm (GA) is a process of iterative searching used in computers to find exact or appropriate solutions to optimization and search problems. It is based on the principles and mechanisms of natural selection and survival of the fittest concept. The Basic GA cycle is given in fig. 1.

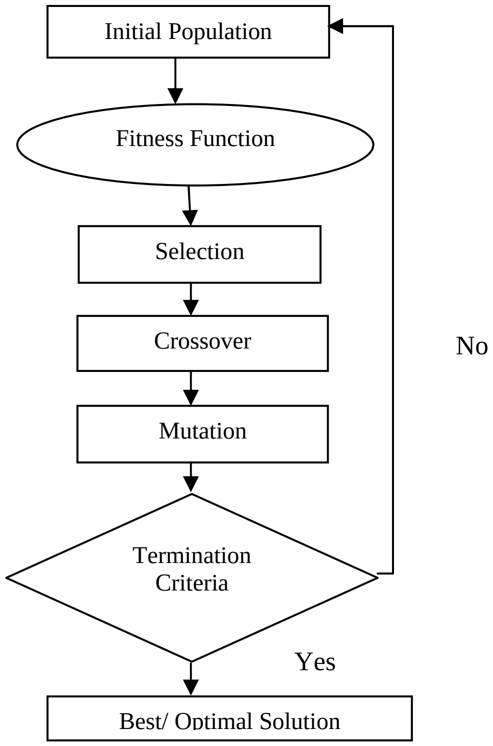

Figure1. Genetic Algorithm Cycle

A simple GA works by randomly generating initial population of solutions encoded as strings. These candidate solutions called "chromosomes" are allowed to evolve over a number of generations, and, then applying three genetic operators to create new and better populations as successive generations. First, selection operator chooses two members of present generation to participate for other two operations: - crossover and mutation. The crossover operator randomly selects pairs of strings, mate them and create new strings. Mutation is the random occasional alteration of the value at string position based on the mutation rate specified by the user. Sometimes Elitism is included to prevent losing best chromosomes after implementing crossover and mutation.

The hybrid approaches to Genetic Algorithms is combination of local searches such as stochastic hill climbing, simulated annealing etc. with Genetic Algorithms. Various hybrid approaches include changing or modifying the genetic operators, GA phases such as crossover, mutation, initial population generation, selection, and mutation. Example can be Genetic Algorithm by Li-Ying Wang where the author introduced untwist operator to improve the performance of GA by breaking the knots of route.

III. PRPOSED METHOD FOR SOLVING TSP

The new proposed GA in this paper applies two search strategies to solve the Travelling Salesman Problem: The Global Search and the Local Search. Large population is generated, to increase the probability of having good chromosomes, then, the algorithm applies quick sort for sorting the population in ascending order and deleting the bad 50 percent of population, with large distances. This will ensure more fit population for further generations. And, finally to guarantee global convergence, heuristic knowledge is employed for mutation.

A. Genetic Encoding and Initialization of TSP

The encoding stage decides the format for the chromosomes and to a large extent it effects the efficiency of algorithm. There are several ways for chromosome representation in genetic algorithm such as path representation, binary coding, adjacency representation, matrix representation etc. For the TSP, solution is typically represented by chromosome of length as the number of nodes in the problem. Each gene of a chromosome takes a label of node such that same node cannot appear twice in the same chromosome.

In this paper, path representation scheme is employed. A chromosome t_k(k=1,2,……….pop_size), pop_size is the population size is represented as, t_k (c_1,c_2,……….c_n) where c_i=the i^{th} city to be visited; i= 1,2,……….n. The route between the cities is described with an array. Each element of the array represents the number of the city.

Example: Considering six cities TSP, a route

4>2>1>7>3>6

can be represented using path representation as : (4,2,1,7,3,6). For array representation of such tour, a t [6] array is declared as:

4	2	1	7	3	6
t[0]	t[1]	t[2]	t[3]	t[4]	t[5]

Figure2. Array representation of chromosomes

This chromosome represents the tour starting from city4 to city2, city2 to city1, city1 to city7, city7 to city3, city3 to city6 and from city 6 to city4.

After the encoding of TSP, initial population is randomly generated.

B. Fitness Function

The fitness function implements the mechanism for deciding each chromosome's fitness. Hence, the constitution of the fitness function is vital to the algorithm, as it affect the convergence rate of GA. Since, TSP is a minimization problem; the reciprocal of objective function is treated as fitness function.

The fitness function is defined as:

f(x)=1/f(x),

$$\text{where } f(x)=\sum_{i=1}^{n} d(c_i,c_j)+d(c_n,c_1) \qquad (1)$$

$d(c_i,c_j)$:travelling distance from city i to city j

Euclidean distance d (c_i, cj) is calculated as:

$$C_{i,j} = \sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \qquad (2)$$

The longer the route length is, the smaller is the fitness value.

C. Selection Operators and Local Search

Local search is carried out on best individuals of each generation depending on their fitness values. The best individuals in the neighbor area are iteratively searched for better individuals; that will substitute the worst individuals. For local search, algorithm use elitism strategy, for selecting the better solutions. It allows the solutions to get better over time. By picking only the few best parents, and replacing the worst, population will converge quickly.

The major function of the selection operator is to select individuals with higher fitness value. Roulette-Wheel selection, Tournament selection, Rank selection, Steady-state selection are some of the frequently used selection operators.

In this paper, Roulette-Wheel selection along with strategy of best-individual to survive is adopted. According to Roulette-Wheel selection, if f_i, is the fitness of individual I in the population, its probability of being selected is:

$$P_i = f_i / \sum_{i=1}^{N} f_i \qquad (3)$$

N= number of individuals in population

And, Cumulative frequency is $Q_i = \sum_{J=1.}^{i} P_i \qquad (4)$

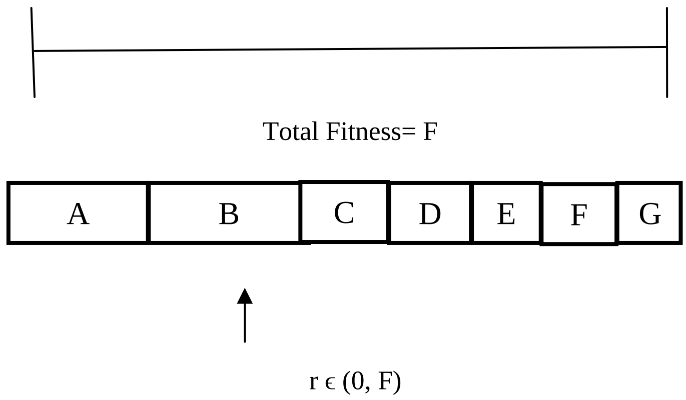

Figure3. Example of the selection of single individual

D. Crossover Opeartor

Crossover operator is used to generate new offspring. It selects two parent chromosomes, and produces better off springs by exchanging parent's bit information. The evaluation function gives each chromosome a score that decides the chromosome's probability of crossover. For TSP, Partially Matched Crossover (PMX), Cycle Crossover (CX), Order Crossover (OX), is frequently used.

In this paper, Partially Matched Crossover (PMX) operator is employed. Goldberg, in 1945 presented the PMX operator. In Partially Matched Crossover, two strings are aligned and two crossover points are picked uniformly at random and PMX proceeds by position wise exchange.

1. Two chromosomes as Parent1 and Parent 2 are aligned, and two chromosome sites are uniformly picked at random along the chromosomes.
2. Each element between the two crossover points in the alternate parent is mapped to the position held by this element in the first parent.
3. Inherit remaining elements from the parents.

Example to show the crossover operator performance:
Let two chromosomes be Parent1 and Parent2.

Parent1= (9 7 8 | 3 2 4 | 5 6 10 1)
Parent2= (8 3 2 | 7 1 10 | 6 4 5 9)

The cross points for parents are indicated by |.

If we perform a two point crossover on the chromosomes, offspring generation would be:

Child1= (9 7 8 7 1 10 5 6 10 1)
Child2= (8 3 2 3 2 4 6 4 5 9)

In Child1 city 7 and city 1 are visited twice and cities 2, 3, 4 are not visited, similarly, in Child2 city 3 and city2 are visited twice, while, not visiting cities 1, 7, 10.

So, it violates the closed tour of TSP.

PMX solves this conflict by modifying the procedure as: For any child, find the position of the element where there is conflict in the alternate parent, pick the element from that position in the original parent and place it to the position of conflict in the child.

For PMX the off springs will be generated as:

Child1 = (9 X 8 | 7 1 10 |5 6 X X)
Child 2 = (8 X X | 3 2 4 | 6 X 5 9)

Where, X shows position of conflict.

After resolving conflict,
Child1= (9 3 8 | 7 1 10 | 5 6 4 2)
Child2= (8 7 1 | 3 2 4 | 6 10 5 9)

So, PMX gives us legal chromosomes, all the cities are traversed exactly once.

E. Mutation Operator

Mutation is performed after crossover. Mutation factor increases the multiplicity of the individual, eliminating the possibility of evolution, limiting on some particular objects.
Mutation operator maintains the diversity of individuals in the population. The mutation operator consists of inversion, insertion, shift and swap, for TSP. The inversion mutation operators cannot fasten the algorithm convergence speed.
For TSP problem, mutation operator works on a single chromosome at a time, and, the genes are randomly altered.In this paper, Swap mutation operator, along with some heuristics accelerates the convergence speed, in TSP.
The procedure:

1. Randomly a tour is chosen and a city (c), the mutation point is chosen.
2. Closest city to c is chosen as second mutation point.
3. The cities at these two points are interchanged clockwise, preserving the adjacency information.

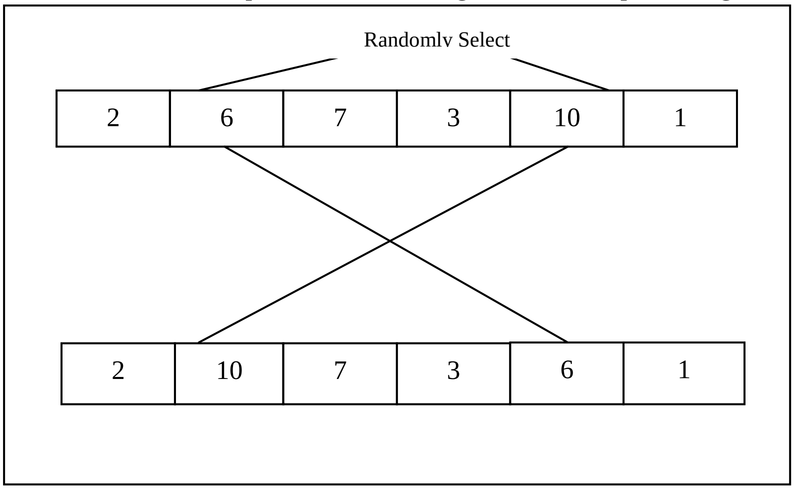

Figure 4: Swap Mutation Operator

F. Termination Condition

Genetic algorithm is an iterative searching method that gives us an optimal solution but, not surely the best one. For this, termination condition is set to number of iterations in the paper that acts as the standard of convergence. When number of iterations equals the defined maximum generation value, the solution is considered as the optimal route.
The flow chart for the proposed algorithm is given in fig5.

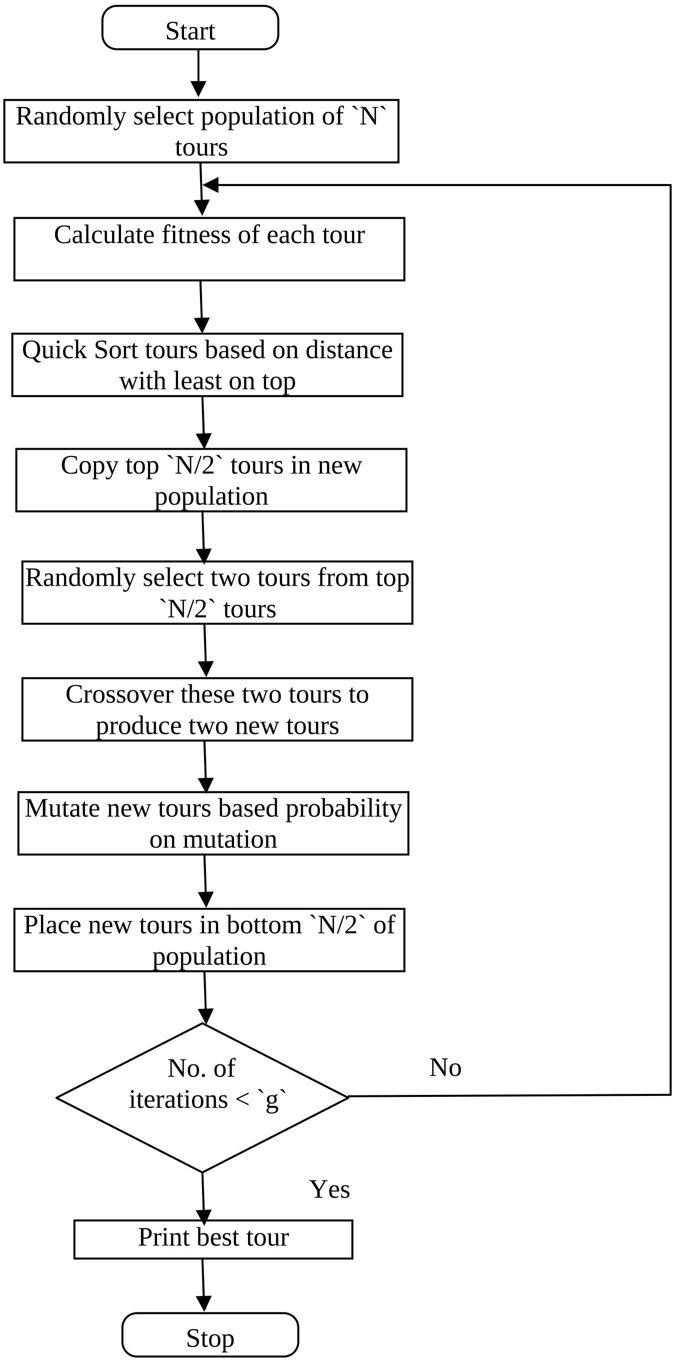

Figure 5: Proposed GA Algorithm

IV. EXPERIMENTAL RESULTS

The proposed algorithm is compared with other researcher's work in the same area, using the same dataset. To demonstrate the performance of the proposed algorithm, symmetric benchmark TSP problems from TSPLIB are used. The program is implemented in C and the results are compared with results of SWAP_GATSP [11], OX_SIM [2], MOC_SIM [2], FRAG_GA, ALGORITHM [4] and IGA [13].

TABLE I. RESULTS USING SWAP_GATSP, MOC_SIM, OX_SIM, IGA [13], FRAG_GA, ALGORITHM [4] AND PROPOSED ALGORITHM FOR DIFFERENT SYYMETRIC TSP INSTANCES

PROBLEM		SWAP_GATSP	MOC_SIM	OX_SIM	IGA	FRAG_GA	ALGORITHM [4]	PROPOSED ALGORITHM
eil51	best	439(220)	444(1600)	493(2500)	-	-	428(27)	436(1500)
n=51 optimal=426	average	442(700)	453(3000)	540(3000)	-	-	431(28)	469(1800)
st70	best	685(600)	698(4500)	823(4500)	-	-	679(30)	683(2000)
n=70 optimal=675	average	701(1000)	748(7500)	920(7500)	-	-	685(35)	769(2500)
eil76	best	548(700)	562(3800)	597(5000)	-	-	548(43)	568(4500)
n=76 optimal=538	average	555(1000)	580(7500)	620(7500)	-	-	552(43)	592(5000)
KroA100	best	21397(2000)	21514(8200)	21746(10000)	-	-	21285(36)	21447(8000)
n=100 optimal=21282	average	21740(3000)	21825(12000)	22120(12000)	-	-	21353(50)	21556(9500)
Rat783	best	9732(30000)	-	10810(70000)	8924(15000)	9007(15000)	-	8996(20000)
n=783 optimal=8806	average	10110(40000)	-	11520(10000)	9216(20000)	9442(20000)	-	9023(30000)
burma14	best	-	-	-	-	-	-	30.8785(20)
n=14 optimal=30.8785	average	-	-	-	-	-	-	31.7645(26)

V. CONCLUSIONS

In this paper a new Hybrid GA is proposed that is based on basic GA theory of achieving better population with better former population. The strategy of local search and heuristic approximation method is used to deal with premature convergence and to find the local optimum as an alternative to the other or non-fit individuals. Local and global search improves the search efficiency of the algorithm. Compared with other GAs, as shown in Table1 and Fig. 6, the performance of proposed GA is better.

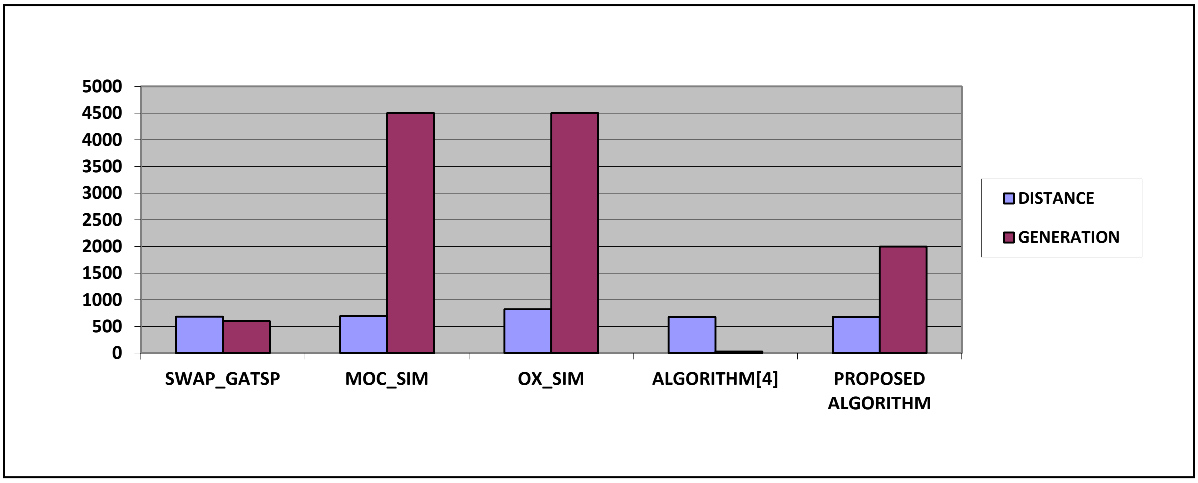

Figure6. Comparing results of SWAP_GATSP [11], MOC_SIM [2] OX_SIM [2], ALGORITHM [4], PROPOSED ALGORITHM for **st70** TSPLIB instance.

VI. ACKNOWLEDGEMENT

I would like to give my sincere thanks to my Guide, Prof. Amandeep Nagpal and Mr. Abhishek for their guidance, enthusiasm and inspiration during the research work. Their consistent help and valuable timely help and hearted cooperation were instrument to complete my thesis. They were ever willing to give me all kind of support and help. I am very grateful to them for all sorts of help they have extended to me during the course of this thesis.

I would admit that nothing would have been possible without the will of god almighty who gave me the courage during my thesis work. Last, but not the least, I express my gratitude to my loving husband and all other family members, who were the moral support behind this work.

VII. REFERENCES

[1] Goldberg David E, Lingle R Jr. “Alleles, Loci, and the Travelling Salesman Problem.” Proc. Of 1 st Int. Conf. on Genetic Algorithms and their Applications, Lawrence Erlbaum Associates, 1985, 154-159.
[2] S.S Ray, S.K. Pal, S. Bandyopadhyay, “Genetic operators for Combinatorial Optimization in TSP and Microarray Gene Ordering,” Applied Intelligence, vol. 26, no.3, pp. 183-195, 2007.
[3] L.Jiao, L.Wang, “ A Novel Genetic Algorithm Based On Immunity,” IEEE Transactions on Systems, Man and Cybernetics, Part A, pp. 552-561, 2002.
[4] Gohar Vahdati, Mehdi Yaghoubi, Mahdieh Poostchi, M.B. Naghibi S, “ A New Approach to Solve Travelling Salesman Problem using Genetic Algorithm Based on Heuristic Crossover and Mutation Operator”, International Conference of Soft Computing and Pattern Recognition, 2009.
[5] M.Dam, M. Zachariasen, “ Tabu Search on the geometric travelling salesman problem”, Meta-heuristics: Theory and Applications, Kluwer Academic Publishers, Boston, pp. 571-587, 1996.
[6] T. Nakaguchi, K. Jin’no, M. Tanaka, “ hysteresis Neural Networks for Solving Travelling Salesman Problems”, 2000 IEEE International Symposium on Circuits and Systems, Switzerland, vol. 03, pp. 153-156, 2000.
[7] Q. Zhu, S.Chen ,” A new Ant Evolution Algorithm to Resolve TSP Problem”, IEEE Sixth Conference On Machine Learning and Applications (ICMLA), pp. 62-66, 2007.
[8] E. Lawle,” Combinatorial Optimization: Networks and Matroids”, Holt, Rinhart, and Winston, New York 1,1976.
[9] P. Larranaga, C.M.H. Kujipers, R.H. Murga, I. Innza and S. Dizdarevic,” Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators”. P. 13(2), s.l.: Artif, Intell, 1999. 129-170
[10] Buthainah Fahran Al-Dulaimi, and Hamza, A. Ali “ Enhanced Travelling Salesman Problem Solving by Genetic Algorithm Technique (TSPGA) “ PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY VOLUME 28 APRIL 2008 ISSN 1307-6884, PP. 296-302.
[11] S.S. Ray, S.Bandyopadhyay, S.K. Pal,” New Operators of Genetic Algorithms for Travelling Salesman Problem”, Cambridge, UK, [CPR-042], vol.2, pp. 497-500, 2004.
[12] Z. Tao, “ TSP Problem Solution Based On Improved Genetic Algorithm,” In proceedings of the 2008 Fourth International Conference on Natural Computation, ICNC, IEEE Computer Society, Washington , DC, vol. 01, pp. 686-690,2008.
[13] Omar M. Sallabi, Younis El-Haddad, “ An Improved Genetic Algorithm to Solve the Travelling Salesman Problem”, World Academy of Science, Engineering and Technology 52 2009.

AUTHOR’S PROFILE

NAVJOT KAUR DALIP is currently pursuing her M.Tech in Information Technology from Lovely Institute of Technology. She received her M.Sc (Computer Science) degree from Guru Nanak Dev University in 2005.