

# Solving TSP using DARO

Nitin S. Choubey  
Professor & Head, Computer Department  
MPSTME, SVKM's NMIMS, Shirpur  
Maharashtra, India-425405

## ABSTRACT

Travelling Salesperson Problem is finding a Hamiltonian cycle with minimum weight. The paper presents solution travelling salesperson problem with Genetic Algorithm. Dynamic allocation of reproduction operators and social disaster technique are used for generating the population and based on the convergence analysis of evaluation process. The method is demonstrated and found to be effective in the data set for the cities in the state of Maharashtra.

## Keywords

Genetic Algorithm, Evolutionary Computation, Travelling Salesperson Problem, Premature Convergence problem.

## 1. INTRODUCTION

Travelling Salesperson Problem (TSP) is to find a tour of minimum cost [1]. A Tour is a directed simple cycle that includes every vertex in  $V$ . The cost of a tour is the sum of the cost of the edges on the tour. The travelling sales person problem find application in the variety of areas ranging from the robot arm motion for tightening screws in assembly line to finding the optimum rout to deliver the packets by a delivery van. The objective in every application is to get the minimum cost route. It is a classical problem of combinatorial optimization. The problem is already proved in the category of NP-Hard Problem. Potvin [2] presents survey of Genetic Algorithm (GA) approaches for the general TSP.

Genetic algorithms are the heuristics methods from the category of evolutionary algorithms which are based on the Darwin's principle of origin of species by means of natural selection [3]. GA's are invented by John Holland in 1960's [4]. In contrast with Evolution Strategies and Evolutionary Programming, Holland's original goal was not to design algorithms to solve specific problems, but rather to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be utilized into computer systems. Holland's 1975 book 'Adaptation in Natural and Artificial Systems' presented the GA as an abstraction of biological evolution and gave a theoretical framework for adaptation under the GA. Many problems in engineering and related areas require the simultaneous genetic optimization of a number of, possibly competing, objectives have been solve by combining the multiple objectives in to single scalar by some linear combination [5].

The perennial problem with GA is that of premature convergence, a non-optimal genotype taking over a population resulting in every individual being either identical or, the consequences of which is a population that does not contain sufficient genetic diversity to evolve further. To avoid the premature convergence, in a GA is imperative to preserve the population diversity during the evolution. An approach to increase the population size may not be enough to avoid the problem, as any increase in population size will incur the two fold cost of both extra computation time and more generations to converge on an optimal solution.

Paper focuses on the utilizing more than one reproduction operators by Dynamic Application of Reproduction Operators (DARO) Approach [6] for producing successive generations and use of Social Disaster Technique (SDT) [7] for handling the special case of potential convergence. Section 2 gives the description of the DARO and SDT, where as Section 3 covers the details of the data set used in the experiment. Section 4 describes the obtained result followed by concluding remarks in section 5.

## 2. METHODOLOGY USED

The GA produces successive generations of individuals by computing their "fitness" at each step and selecting the best of them, when the termination condition arises. Figure 1 shows a Simple Genetic Algorithm approach.

1. Create initial random Population.
2. Calculate fitness of the individuals in the population.
3. Repeat following steps until terminations criteria is reached.
  - a. Select best fit from the current population and generate offspring.
  - b. Evaluate fitness of each offspring.
  - c. Replace weak individuals from the population with newly generated ones.

**Fig 1: Simple Genetic Algorithm.**

In order to avoid the premature convergence problem the selection of a reproduction operator for the process of crossover and mutation plays an important role by producing successive generation. The figure gives steps for the Dynamic application of reproduction operators, where more than one operator is used for reproducing the proportionate amount of child population. The portion to be generated by any Crossover-Mutation Operator Combination (CMOC) is determined by the contribution made by the operator combination in generation of effective children in comparison with the parents based on fitness [8]. It is an extension of the Dynamic Application of crossover and mutation Operator Approach given in Elena Simona Nicoaria [6]. This extended version is also used for effective grammar induction [8] [9]. The GA run begins with the allocation of the equal probability to every reproduction operator combination (Crossover-Mutation operator combination, CMOC). The probability value determines the proportion of the child population to be generated with the application the CMOC.

The probability value for the CMOC is updated after every generation based on the ratio of the average fitness of the generated children to the average fitness value of the selected parent. The each CMOC will have chance to generate individuals in child population in proportionate to their probability value.

The process for Dynamic Application of Reproduction Operators is given below:

1. Create initial population of random individuals and calculate the fitness of all individuals.
2. Assign the initial probability values to the operator combination (m crossover × n mutation operators) by the formula,

$$p(u) = \frac{1}{(m * n)} \& \sum_{u=1}^{m*n} p(u) = 1$$

3. Sort the current population in increasing order of fitness.
4. Select the crossover operator,  $X_{i=1}$ . Select the first mutation operator,  $M_{j=1}$  and set  $u=1$ .
5. Set Resultant Efficiency, RE to 0.
6. Select two parents by using tournament selection method,  $P_1$  &  $P_2$  and produce two off-springs,  $C_1$  &  $C_2$ , with the crossover operator  $X_i$ .
7. Perform mutation on the newly generated off springs with the mutation operator,  $M_j$
8. Calculate the fitness of off-springs.
9. Calculate the Resultant Efficiency as,

$$RE = RE + \frac{(fitness(C_1) + fitness(C_2)) / 2}{(fitness(P_1) + fitness(P_2)) / 2}$$

10. Insert the newly generated off-spring in to child population.
11. Repeat step 6 through 10 until  $p(u)$  proportion of population is generated by the current operator population.
12. Update  $p(u)$  as,

$$p(u) = p(u) + RE$$

13. Select next mutation operator and increment u.
14. If all the mutation operators are applied with the current crossover operator then go to step 15 else go to step 6.
15. Select next crossover operator. If the entire crossover operators are applied, then go to step 16 else go to step 4.
16. Update  $p(u)$  for all u as,

$$p(u)_{next} = \frac{p(u)_{current}}{\sum_{u=1}^{m*n} p(u)_{current}} \& \sum_{u=1}^{m*n} p(u)_{next} = 1$$

17. Create new current population by overlapping the current population with child population.
18. If termination criterion is not reached, go to step 3.
19. Stop.

Three crossover operators and four mutation operators are used in the process. The Crossover Operators used are Single Point Order Based Crossover (SPOBC), Two Point Order Based Crossover (TPOBC), and Two Point Order Based Crossover with Internal Swapping (TPOBCIS) whereas the Mutation Operators are Swap Single bit Mutation (SSBM), Swap Block Mutation (SBM) [10]. The working of Mutation and crossover operators are shown in fig 2 and fig 3 respectively.

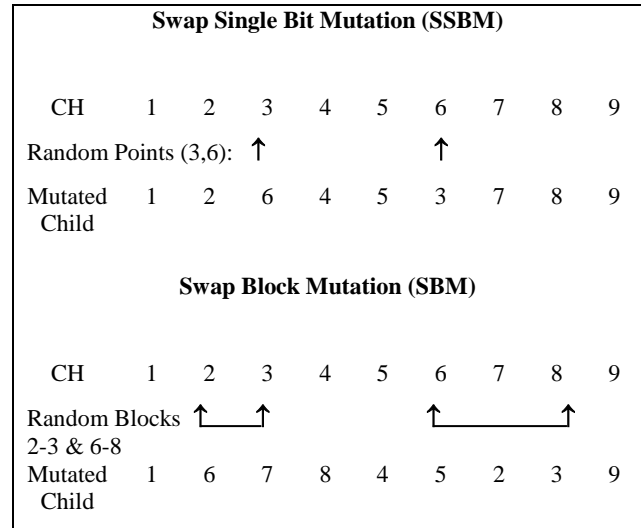


Fig 2: Mutation Operators Used

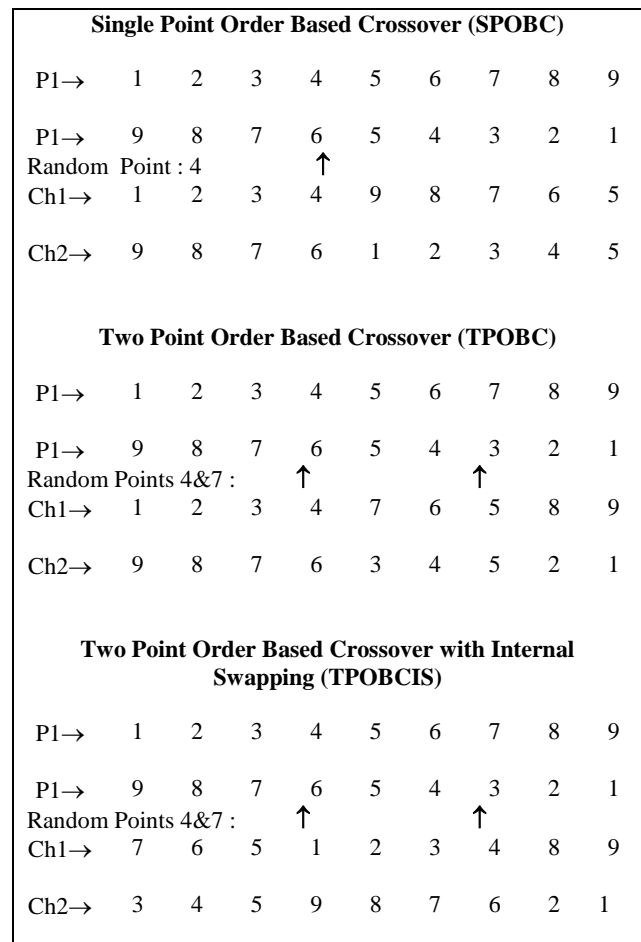


Fig 3: Crossover Operators Used

Potential convergence is examined with the help of the difference between best and worst value (D\_Range) obtained in any generation. If the value of D\_Range is less than the required minimum difference (Min\_Diff), the next generation is obtained by using Social Disaster Technique (SDT). In SDT, the entire population is operated with a WRAP (Judgment day) Operator by preserving the elite members of the population [7]. The process adopted for the experiment is shown in fig 4.

The process used in the experiment is shown in the fig 4.

1. Create initial random Population.
2. Calculate fitness of the individuals in the population.
3. Repeat following steps until the prescribed maximum number of generations are completed.
  - a. If D\_Range is less than Min\_Diff, generate next generation by using SDA, otherwise generate next generation by using DARO.
  - b. Evaluate fitness of each offspring.
  - c. Replace weak individuals from the population with newly generated ones.

Fig 4: GA process used

### 3. DATA SET USED

New data set is used for the experiment. The data about the various cities in the State of Maharashtra from India is used in the experiment. The data (Latitude & Longitude) about the location of the cities is taken from [http://www.mapsofindia.com/lat\\_long/maharashtra/](http://www.mapsofindia.com/lat_long/maharashtra/). The data set of the cities used for experiment is given in figure 5.

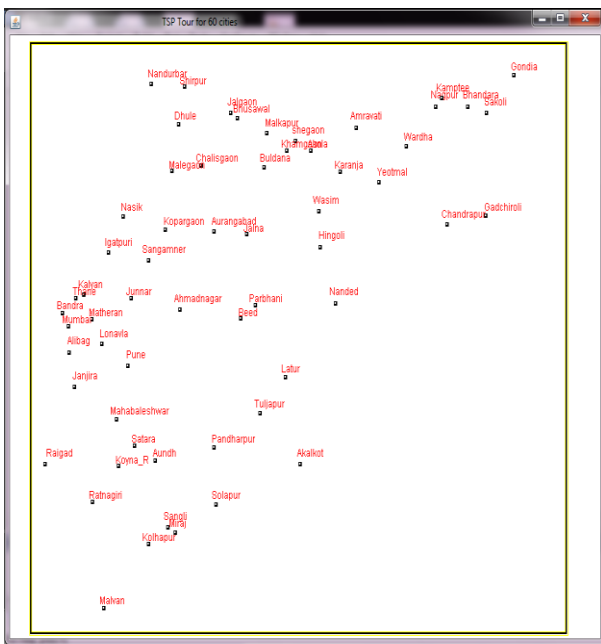


Fig 5. Locations of the cities in State of Maharashtra

### 4. EXPERIMENTAL SETUP AND RESULT ANALYSIS

The experiment is conducted JDK 1.6 on Experiment is done with JDK 1.4 on an Intel Core™2 CPU with 2.66 GHZ and 2 GB RAM. The Population size =200, Maximum number of generation = 5000, Crossover Rate = 0.8 and mutation rate = 0.05 is used for the purpose of experiment. Fig 6 shows the comparison of Adopted approach with Simple Genetic Algorithm (SGA) approach. The adopted approach is found to give better performance in term of the required number of generation for achieving result. It is found that the results obtained for n (n = 20, 25, 30, 35, 40, 45, 50, 55, 60) number of cities is shown in the fig 7. It is found that, the population is converged to the best value earlier in case of less number of cities, whereas it has converged late for the relatively more number of cities and

complex data sets. The adapted method is found to be working successfully on the data set considered. There is further scope for adoption of the same method for more complex data sets.

### 5. CONCLUSION

The proposed model has been implemented, and the results for the data set used are demonstrated successfully. The DARO approach found to be effective in utilizing the right combination of reproduction (crossover and mutation) operator for producing the better results. The combination of DARO approach and Social Disaster Technique resulted in to generating the optimal solution for the data set used. There is further scope for using the methods for larger number of cities and more complex data set.

### 6. ACKNOWLEDGMENTS

Author thanks to the Dr. Tapan Bagchi, Director, SVKM's NMIMS, Shirpur campus and Dr. M. V. Deshpande, Associate Dean, MPSTME, Shirpur Campus for providing necessary guidance and infrastructural facilities for conduction of experiment. Author also thanks to Ms. Manisha Kasar, Mr. Nilesh Pawar, and Ms. Shubhangi Patil for their necessary help in data collection.

### 7. REFERENCES

- [1] Horowitz E., Sahani S, and Rajasekaran S, 2007. Fundamentals of Computer Algorithm, University Press, 2007.
- [2] Potvin J.Y., 1996. Genetic algorithms for the travelling salesman problem, Annals of Operations Research, 339–370, 1996.
- [3] DARWIN C., 1859, The origin of species by means of natural selection, 1859.
- [4] Holland John H., 1992. Adaption in Natural and Artificial Systems- Introductory analysis with Application to biology, control and Artificial Intelligence, Bradford Book edition, The MIT Press, England.,1992.
- [5] Goldberg D., 1989. Genetic Algorithm in Search, Optimization, and Machine Learning. Addison Wesley, 1989.
- [6] Nicoar, E.S., 2009. Mechanisms to Avoid the Premature Convergence of Genetic Algorithms. Universitatea Petrol-Gaze din Ploieti.Matematica-Informatic-Fizica LXI(1), 87–96 2009.
- [7] Kureichick, Miagkikh, and Topchy, 1996. Genetic algorithm for solution of the travelling salesman problem with new feature against premature convergence, Working Paper, 1996.
- [8] Choubey N. S., and Kharat M. U., 2012. Hybrid System for Handling Premature Convergence in GA - Case of Grammar Induction, Applied Soft Computing 2012, doi: 10.1016/j.asoc.2012.03.069.
- [9] Choubey N. S., and Kharat M. U., 2011. Approaches for Handling Premature Convergence in CFG Induction Using GA, Soft Computing in Industrial Applications, 55-66, Springer Berlin/Heidelberg, 2011.
- [10] Sivanandam, Deepa "Introduction to Genetic Algorithm", Springer, 2008.

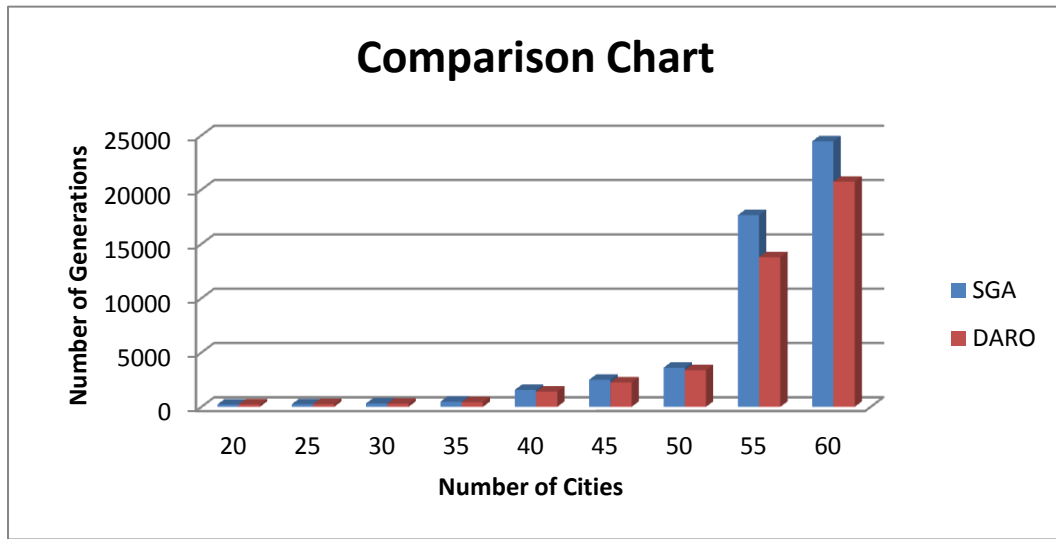


Figure 6. Comparison chart for SGA and DARO



Fig 7: Optimal Solution for n cities (n=20, 25, 30, 35, 40, 45, 50, 55, 60) from State of Maharashtra, India