



Variation in Selection Operator of Genetic Algorithm to Optimize TSP

Smita Sharma¹, Monika²

Research Scholar¹, Assistant Professor²

^{1,2}Department of Computer Science & Applications,
Kurukshetra University, Kurukshetra, Haryana, INDIA

Abstract: Genetic Algorithm (GA) is used to solve optimization problems. It depends on the selection operator, crossover and mutation rates. In this paper Roulette Wheel Selection operator with crossover and mutation probabilities, is used to solve well known optimization problem Traveling Salesman Problem (TSP). This paper compared effect of two different ways of selection on 2 different inputs from TSPLIB provided by Heidelberg University. Out of them one gives more optimized result but consumes more time as compared to other one.

Keywords: Selection; Selection Variation; Genetic Algorithm; Optimization; Travelling Salesman Problem

I. Introduction

Genetic algorithms are adaptive algorithms proposed by John Holland in 1975 [1] and were described as adaptive heuristic search algorithms [2] based on the evolutionary ideas of natural selection and natural genetics by David Goldberg. These are powerful optimization techniques that employ concepts of evolutionary biology to evolve optimal solutions of a given problem. Genetic algorithm (GA) is mainly composed of two processes. First process is the selection of chromosomes from the current population for the production of next generation and the second process is manipulating the selected individuals by crossover and mutation techniques. The main work of selection operation is to determine which individuals are chosen for reproduction. The main principle of selection is “the better is an individual; the higher is its chance of being parent.” [1][2] Selection reduces the search area by discarding the poor solutions. Crossover and mutation explore the search space for new promising solutions. The power of genetic algorithms comes from their ability to combine both exploration and exploitation in an optimal way [1]. It is important to find good balance between exploitation (i.e. better solutions go to the next generation more frequently than the poor solutions) and exploration (i.e. poor solutions must have chance to go to next generation) with the selection operation. Different selection strategy significantly affects the performance of GA differently. This study is intended to compare the performance of GA when using existing roulette wheel selection strategy with order crossover (OX) operator and roulette wheel selection strategy in a different way with same crossover operator in solving the traveling salesman problem (TSP).

TSP is a classical example of NP-Hard combinatorial optimization problem. Many other production problems can be reduced to TSP concept that a salesman who must travel city to city, visiting each city exactly once and returning to the home city. Salesman can select the orders of cities visited to minimize the distance traveled. The minimization of distance travelled will apparently save him time and money. Although, the problem is conceptually simple, but the solutions finding is hard. The main problem is the no. of tours; $(n-1)! / 2$ for symmetric n cities tour. As the number of cities increases, the number of valid permutations of tours will also increase significantly. So it's the factorial growth which makes the task of solving TSP immense even for modest n sized problems.

The remainder of this paper is organized as follows: Section II presents a literature review on selection strategy. Section III contains proposed algorithm. Section IV discusses the experimental results. Lastly, Section V contains the conclusion.

II. Literature Review

Several researchers have studied the performance of GA using different selection strategies. The performance is usually evaluated in terms of convergence rate and number of generations to reach the optimal solution. Selection stage of GA was examined for the problems and solutions of different selection operators, [3]. They also proposed a new selection operator called sexual selection and compared the performance with commonly used operators on Royal road problem. They claimed that new selection performs better or equal with roulette-wheel on average when no fitness scaling is used. It performs better than tournament selection in more difficult test cases. The results between proportional roulette wheel and Rank based roulette-wheel selections was compared on various mathematical functions and found that rank-based selection outperformed proportional roulette wheel in number of generations to reach optimal, [4]. They observe that rank selection is steadier, faster and more robust towards the optimum solutions. Mashohor et al. evaluated the performance of PCB inspection system using three GA selections; deterministic, tournament and roulette-wheel and found that deterministic one will overruled the other in reaching optimum in less generations, followed by roulette wheel and tournament

selection. A new PBS blended selection operator which has tradeoff between exploration and exploitation has been proposed, [5]. They compare the performance on standard TSP problem with roulette-wheel and Rank selection techniques. In roulette wheel selection, individuals with higher fitness have more probability of selection. It can be biased towards them too, [2] [6]. Rank-based selection can maintain a constant pressure in the evolutionary search where it introduces a uniform scaling across the population and is not influenced by super-individuals or the spreading of fitness values at all as in proportional selection, [7] [8]. The performance of PBS depends on the number of generations. In start of GA selection operator had explorative nature, as the search progress, selection pressure also increased and the nature of selection also changed to exploitative. The performance of PBS over other selection operators is superior.

III. Proposed Algorithm

In this section, two algorithms are described. Both the algorithms have the basic genetic algorithmic structure. In these algorithms, De Jong's guidelines, which is to start with a relatively high crossover probability ($P_c \geq 0.6$), relatively low mutation probability (P_m , 0.001-0.1), and a moderately sized population is used. The selections of parameter values are very depend on the problem to be solved [9] [10].

In the simple algorithm, following steps are followed:

- a) Initialization (using permutation encoding).
 - b) Roulette Wheel Selection (that selects only two individuals at a time that will take part in reproduction after that).
 - c) Order Crossover (OX, which will retain the order of genes in chromosome, is applied on two selected individuals given by selection operator).
 - d) Elitism (best individual of old generation will replace the worst individual of next generation).
- Steps b), c) and d) will be repeated population size times.
- e) Interchange Mutation (positions of two genes are swapped in an individual).
 - f) Total Replacement (Best individuals that will go in next generation will be only from the new generation got after mutation).

In the proposed algorithm, following steps are followed:

- a) Initialization (using permutation encoding).
- b) Roulette Wheel Selection (that selects as much individuals as in the population).
- c) Order Crossover (it considers all the possible combinations among selected individuals).
- d) Interchange Mutation (positions of two genes are swapped in an individual).
- e) Partial Replacement (Individuals that go in next generation will be the best out of both the populations before applying GA and after applying GA).

For both the algorithms in this study, termination is performed when number of generation reached the maximum number of generation. The maximum number of generation is entered at runtime of program. The size of population is also entered at runtime of program.

The proposed algorithm gives more optimized result as it considers all the possible combinations at the time of reproduction as compared to the second algorithm. But it also consumes more time to return the result as the population size increases.

IV. Experimental Result

The performance of both algorithms is compared at three TSP instances using MATLAB: the known optimal solution TSP instances taking from TSPLIB. Three problem instances were taken for experiments from standard TSP library TSPLIB are Eil51, Eil76 and Eil101, where 51, 76 and 101 cities are included respectively [11].

The following parameters are used in this implementation:

- Population size (N): 50, 100
- Number of generations (numofgen) : 100, 200
- Selection method: Roulette Wheel Selection (RWS)
- Crossover Operator: Order Crossover (OX) with crossover probability greater than 0.6
- Mutation: Interchange with mutation probability 0.1
- Replacement: Total Replacement and Partial Replacement
- Algorithm ending criteria: Execution stops on reaching numofgen generations
- Fitness Function: Objective value of function

Following are the tables to show the results and to observe the experiments.

Table I shows the experimental results with population size =50 and Generations=100

Algorithm/ Instance	Simple GA	GA with all combinations
Eil51	846	548
Eil76	1352	857
Eil101	1905	875

Table II shows the experimental results with population size =100 and Generations=200

Algorithm/ Instance	Simple GA	GA with all combinations
Eil51	797	522
Eil76	1202	655
Eil101	1795	852

The performance graphs in Figures below show the minimum distance found by the algorithm in each generation. Figure 1 and 2 are for Eil51 instance. Figure 3 & Figure 4 are for Eil76 instance. Figure 5 and Figure 6 are for Eil101 instance. First figure of all the instances is representing the result for population size=50 and generations=100 and second figure is representing the result for population size=100 and generations=200. It can be seen from the graphs that the distance with proposed algorithm reduced towards optimal solution as the generation increased and finally converged at a certain generation.

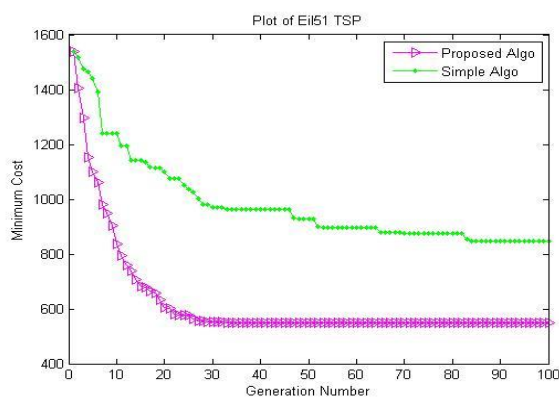


Figure 1: Eil51 Results

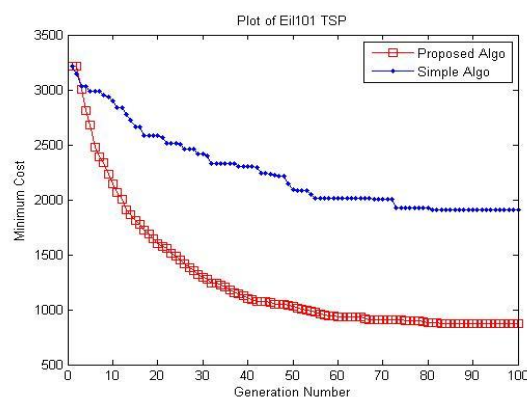


Figure 5: Eil101 Results

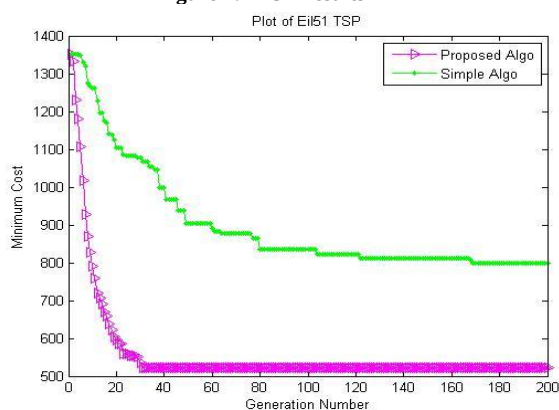


Figure 2: Eil51 Results

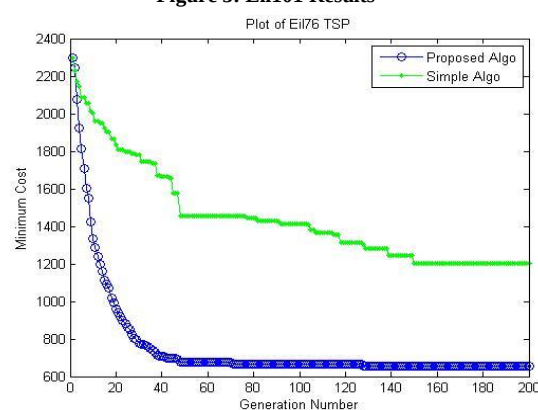


Figure 4: Eil76 Results

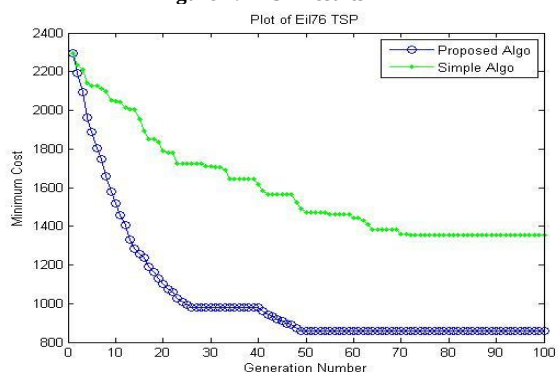


Figure 3: Eil76 Results

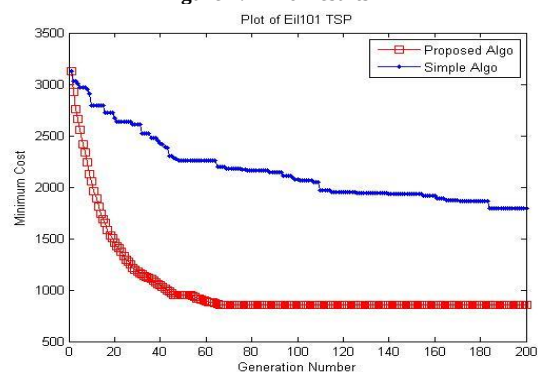


Figure 6: Eil101 Results

V. Conclusion

In this paper, genetic algorithm with extended selection operator is proposed. It uses the same selection operator as in the simple genetic algorithm i.e. roulettes wheel selection. But proposed GA considers all the possible combinations of selected individuals for the reproduction whereas in simple genetic algorithm only two

individuals are selected at a time that participated in reproduction. The performance of proposed algorithm is compared with simple genetic algorithm on standard TSP problems. It produces more optimized result than simple GA.

References

- [1] Holland J., "*Adaptation in natural and artificial systems*", University of Michigan Press, Ann Arbor, 1975.
- [2] Goldberg D. E., "*Genetic algorithms in search, optimization, and machine learning*", Addison Wesley Longman, Inc., ISBN 0-201-15767-5, 1989.
- [3] Goh K. S., Lim A., Rodrigues B., "*Sexual Selection for Genetic Algorithms*", Artificial Intelligence Review 19: 123–152, Kluwer Academic Publishers, 2003.
- [4] Jadaan O. A., Rajamani L., Rao C. R., "*Improved Selection Operator for GA*" Journal of Theoretical and Applied Information Technology, 2005
- [5] Kumar Rakesh, Jyotishree, 2011. "*Blending roulette wheel selection & rank selection in genetic algorithms*" In Proceedings of 3rd International conference on machine learning and computing, V4, IEEE catalog number CFP1127J-PRT, ISBN 978-1-4244-9252-7, 197202.
- [6] Goldberg D. E. and Deb K., "*A comparative analysis of selection schemes used in genetic algorithms*", Foundations of Genetic Algorithms, San Mateo, CA, Morgan Kaufmann, pp 69-93, 1991.
- [7] Whitley D., "*The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best*", Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann, pp 116-121, 1989.
- [8] Back T. and Hoffmeister F., "*Extended Selection Mechanisms in Genetic Algorithms*", ICGA4, pp. 92-99, 1991.
- [9] De Jong K., Spears W. M., "*Using Genetic Algorithms to Solve NP Complete Problems*," Proceedings of the Third International Conference on Genetic Algorithm, Morgan Kaufman, Los Altos, CA, 124 – 132, 1989
- [10] De Jong K. A., "*An Analysis of the behavior of a class of genetic adaptive systems*", (Doctoral dissertation, University of Michigan) Dissertation Abstracts International 36(10), 5140B University Microfilms No. 76/9381, 1975.
- [11] Reinelt G., "*TSPLIB – A Travelling Salesman Problem Library*". ORSA Journal on Computing, Vol.3, No.4, 376 – 384, 1991.