International Journal of Computer Applications (0975 – 8887)
Volume 70– No.21, May 2013

Alpha Cut based Novel Selection for Genetic Algorithm

Rakesh Kumar
Professor
DCSA, Kurukshetra University
Haryana-136119, India

Girdhar Gopal
Research Scholar
DCSA, Kurukshetra University
Haryana-136119, India

Rajesh Kumar
Assistant Professor
DCSA, Kurukshetra University
Haryana-136119, India

ABSTRACT

Genetic algorithm (GA) has several genetic operators that can be changed to improve the performance of particular implementations. These operators include selection, crossover and mutation. Selection is one of the important operations in the GA process. There are several ways for selection like Roulette-Wheel, Rank, and Tournament etc. This paper presents a new selection operator based on alpha cut as in Fuzzy Logic. This is compared with other selection in solving travelling salesman problem (TSP) using different parent selection strategy. Several TSP instances were tested and the results show that proposed selection outperformed proportional roulette wheel, achieving best solution quality with low computing times.

General Terms

Selection, Genetic Algorithm, Traveling Salesman Problem

Keywords

Alpha cut, Fuzzy Logic, Genetic algorithm, Optimization, Selection, Travelling Salesman problem

1. INTRODUCTION

Genetic algorithm (GA) is mainly composed of two processes. First 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 and how many offspring each selected individual will reproduce. 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 and crossover and mutation explore the search space for new promising solutions. Sometimes worst individuals must not be discarded totally, because they may produce some useful genetic material in future. So a good research is to find a trade-off between this exploitation and exploration to find the global optimum. Hence, 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 different crossover operator and new selection strategy with same crossover operators in solving the travelling 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 traveled 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 brief summary of previous works on selection strategy. Section III contains an overview of the genetic algorithm for TSP. Section IV describes the selection strategies in details and proposes a new selection mechanism Section V tests the performance of various selection operators and discusses the experimental results. Lastly, Section VI contains the conclusion.

2. RELATED WORD

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. 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.

3. GENETIC ALGORITHM FOR TSP

This section provides an overview of the genetic algorithm component and operation for solving TSP. The term "genetic algorithm"' (GA) is applied to any search or optimization algorithm that is based on Darwinian principles of natural selection. Genetic Algorithm is a population-based search and optimization method which mimics the process of natural evolution. Genetic Algorithms (GAs) were invented by John Holland in the 1960s and were developed by Holland in 1975 and his students and colleagues at the University of Michigan in the 1960s and the 1970s. Holland's GA is a method for moving from one population of "chromosomes" to a new population by using a kind of "natural selection" together with the genetics inspired operators like crossover, mutation, and inversion. A chromosome contains a group of numbers that completely specifies a candidate during the optimization process. There are a number of possible chromosome representations, due to a vast variety of problem types. The path presentation is more natural to represent the chromosome in TSP. TSP consists of number of cities, where each pair of cities has a corresponding distance. The goal is to visit all the cities such that the total travelling distance will be minimized. So, to represent the solutions in TSP one can use Order based encoding i.e. Permutation Encoding.

GA process starts by supplying some important information such as location of cities, maximum number of generations, population size, probability of crossover and probability of mutation etc. An initial random population of chromosomes (paths) is generated. The fitness function defined as the tour cost of a particular chromosome. This fitness is evaluated for initial population and this population is transformed in next generation by three genetic operators: selection, crossover and mutation. Selection operator chooses two parents to procreate new children by crossover/mutation. New generation contains higher proportion of the characteristics possessed by the good members of the previous generation. After each generation, a new set of chromosomes with equal size to the initial population is evolved. This new set is then used as initial population for the next generation. This evolution runs until the optimum value is reached, which generally occurs when a certain percentage of population has the same optimal chromosome in which the best individual is taken as the optimal solution.

4. SELECTION STRATEGY FOR REPRODUCTION

The selection strategy finds the chromosomes in current generation, which are used to produce new chromosomes. If we choose better chromosomes, the next generation will contain more and more better solutions, which ultimately leads to the optimum solution. Different selection strategies have different methods of calculating selection probability. All selection strategies develop solutions based on the principle of survival of the fittest. Fitter solutions are more likely to reproduce and pass their genetic material to the next generation in form of their offspring, [1] [2].

4.1 Roulette Wheel selection

It is the simplest selection approach. In this, all the chromosomes are placed on the roulette wheel according to their fitness value. Each chromosome has been assigned a segment of roulette-wheel according to the fitness – the bigger the value is, the larger the segment is. Then, the virtual roulette-wheel is spinned. The individual corresponding to the segment on which the roulette wheel stops are then selected. This process will repeat until the desired number of solutions is selected. Individuals with higher fitness have more probability of selection. It can be biased towards then too, [2] [6]. Also it might be the case, that the best one is missed due to non-stopping on that segment by chance. So there is no guarantee that the best one will go. Let f1, f2… fn be fitness values of individual 1, 2… n. Then the selection probability, Pi for individual i is define as,

$$p_i = \frac{f_i}{\sum_{j=1}^{n} f_j}$$

The basic advantage of this is that it gives a chance to all to be selected. For example, if an initial population contains one or two very fit but not the best individuals and the rest of the population are not good, then these fit individuals will quickly dominate the whole population and prevent the population from exploring other potentially better individuals. Such a strong domination causes a very high loss of genetic diversity which is definitely not advantageous for the optimization process. On the other hand, if individuals in a population have very similar fitness values, it will be very difficult for the population to move towards a better one since selection probabilities for fit and unfit individuals are very similar.

Roulette wheel selection

Set l=1, j=1, i=nogen

While l <= mpool

Begin

a) While j<=N

Begin

Compute $FRW_{i,j}$

End

b) Set j=1, S=0

c) While j<=N

Begin

Compute $S=S+FRW_{i,j}$

End

d) Generate random number r from interval (0,S)

e) Set j=1, S=0

f) While j<=N

Begin

Calculate $c_j=c_j-1+FRW_{i,j}$

If $r<=c_j$, Select the individual j

End

g) l=l+1

End

4.2 Rank Selection

It sorts the population first according to fitness value and ranks them. Then every chromosome is allocated selection probability with respect to its rank. Individuals are selected based on their selection probability. Hence 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]. Rank-based selection uses a function to map the indices of individuals in the sorted list to their selection probabilities. Rank-based selection schemes can avoid premature convergence and eliminate the need to scale fitness values, but can be computationally expensive because of the need to sort populations. Once selection probabilities have been assigned, sampling method using roulette wheel is required to populate the mating pool. Rank-based selection scheme helps prevent premature convergence due to "super" individuals, since the best individual is always assigned the same selection probability, regardless of its objective value. However this method can lead to slower convergence, because the best chromosomes do not differ so much from other ones.

Rank Selection

Set l=1, j=1, i=nogen

While l <= mpool

Begin

a) While j<=N

Begin

Compute $rsum_i$

End

b) Set j=1

c) While j<=N

Begin

Compute $PRANK_j$

End

d) Generate random number r from interval (0,rsum)

e) Set j=1, S=0

f) While j<=N

Begin

Calculate $c_j=c_j$-1+$PRANK_j$

If r<=c_j, Select the individual j

End

g) l=l+1

End

4.3 Alpha cut Selection

Proposed Selection is a selection method, in which an alpha cut is defined between 0 and 1. As defined in Fuzzy logic, an alpha cut is determined for a Set, then a membership function is used to determine the elements fitness value, if the fitness is greater than or equal to alpha cut, that individual is a member of the set, otherwise its not [9]. Consider a fuzzy set A defined on the interval X = [0, 10] of integers by the membership Function $\mu A(x) = x / x + 2$ Then the α cut corresponding to α = 0.5 will be {2, 3, 4, 5, 6, 7, 8, 9, 10}.

Likewise, every individual is tested for the fitness function value, and if the fitness value is greater than the alpha cut, that individual is then selected, otherwise it is rejected. At some random generations this selection is useful instead of other traditional selection methods.

Alpha cut Selection

1. Set l=1, j=1, i=nogen

2. While l <= mpool

Begin

a) Generate random number r from interval (0,N)

b) Set j=1, S=0

c) While j<=N

Begin

Calculate Fitness(j)

If Fitness(j) >= alpha_cut, Select the individual j

End

End

5. COMPUTATIONAL EXPERIMENTS AND RESULTS

5.1 Experimental Setup

This section will focus on experiment that use GA selection with roulette-wheel and proposed selection with three crossover operators (Partial Matched (PMX), Order (OX) and Cycle (CX)) for TSP. The algorithms are coded in MATLAB 2011a. The performance of GA is tested at three TSP instances: the known optimal solution TSP instances taking from TSPLIB. For all experiments, the GA procedure employed a combination of three crossovers (PMX, OX & CX) and inversion mutation for producing offspring at every generation. The objective of the experiment is to investigate the performance of GA with traditional roulette-wheel and proposed alpha-cut selection strategies in terms of number of generations and iteration time to come out with the optimal solution for TSP.

One of the main difficulties in building a practical GA is in choosing suitable values for parameters such as population size, probability of crossover (Pc), and probability of mutation (Pm). In this experiment, De Jong's guidelines, which is to start with a relatively high Pc (≥ 0.6), relatively low Pm (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 [10] [11]. This experiment will use a constant population size which is approximately 10 times larger than number of instance. Noted that the larger the population size, the longer computation time it takes. In this experiment, the GA parameters were obtained from the screening experiment and trial run. For each experiment, the algorithms were run ten times and the lowest travelling distance is taken as a final result. For all experiments 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.

Three problem instances were taken for experiments from standard TSP library TSPLIB. Eil51, Eil76 and Eil101 are the names of problems where 51, 76 and 101 cities are included respectively [12].

The following parameters are used in this implementation:

- Population size (N): 50, 100 and 500
- Number of generations (ngen) : 100, 200 and 500
- Selection method: Roulette Wheel Selection (RWS) and Proposed Alpha cut Selection (AS)
- Crossover Operator: Partial Matched Crossover (PMX), Order Crossover (OX), Cycle Crossover (CX).
- Mutation: Inversion with mutation probability 0.01%
- Algorithm ending criteria: Execution stops on reaching ngen generations
- Fitness Function: Objective value of function

5.2 Results and Observations

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

TABLE 1. Experimental results with population size=50 and Generations=100

	Roulette-Wheel Selection			Alpha-cut Selection		
Crossover /Instance	PMX	OX	CX	PMX	OX	CX
Eil51	1182	1147	1295	999	824	1311
Eil76	1903	1892	2122	1679	1149	2012
Eil101	2769	2692	2873	2485	1596	2775

TABLE 2. Experimental results with population size=100 and Generations=200

	Roulette-Wheel Selection			Alpha-cut Selection		
Crossover /Instance	PMX	OX	CX	PMX	OX	CX
Eil51	1159	1087	1224	906	619	1283
Eil76	1844	1801	2100	1528	972	1941
Eil101	2618	2652	2838	1971	1331	2555

TABLE 3. Experimental results with population size=500 and Generations=500

	Roulette-Wheel Selection			Alpha-cut Selection		
Crossover /Instance	PMX	OX	CX	PMX	OX	CX
Eil51	1013	1019	1045	665	485	1068
Eil76	1667	1735	1891	1093	694	1566
Eil101	2250	2338	2566	1478	1220	2454

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. It can be seen from the graphs that the distance with proposed selection reduced towards optimal solution as the generation increased and finally converged at a certain generation.

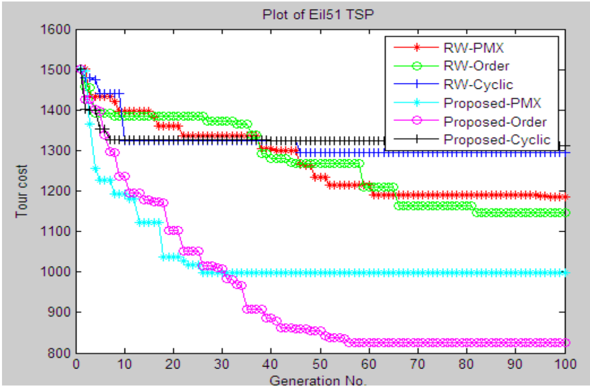

Figure 1: Eil51 results

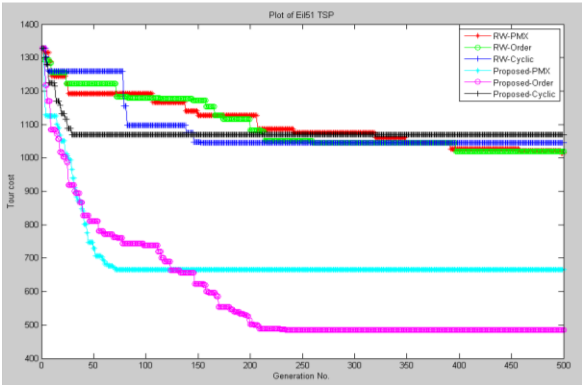

Figure 2: Eil51 results

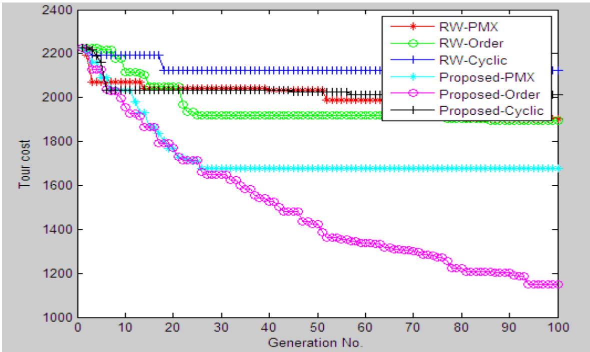

Figure 3: Eil76 results

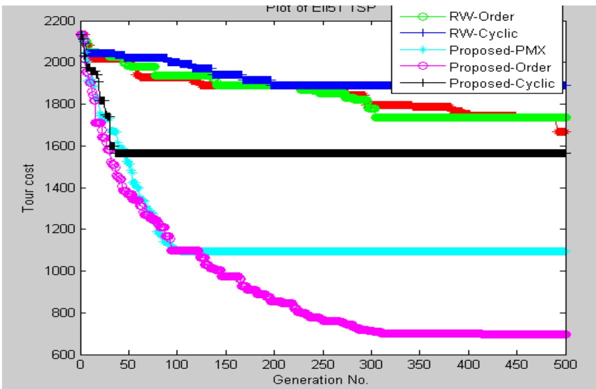

Figure 4: Eil76 results

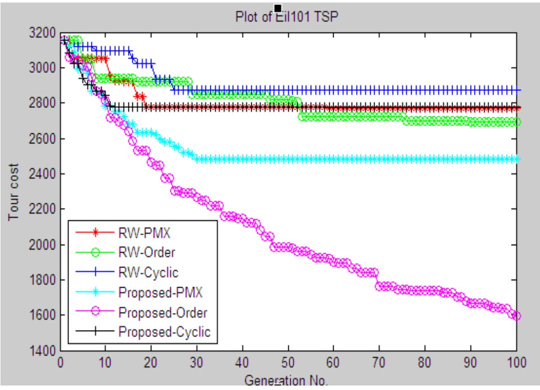

Figure 5: Eil101 results

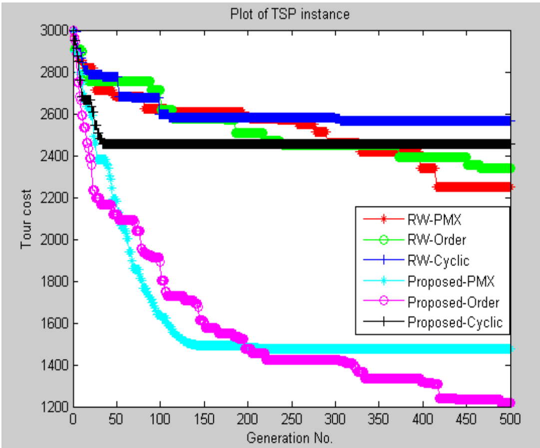

Figure 6: Eil101 results

6. CONCLUSION

In this paper, an alpha cut selection operator is proposed. The performance of alpha cut selection operator is compared with roulette-wheel selection technique on standard TSP problems. Roulette-wheel selection performed like nature selecting the more fit individuals. Alpha cut Selection performs better when combined with roulette-wheel selection with all three crossover operators in all instances of TSP. It produces better results than simple roulette-wheel method. It starts from exploitative nature and one can tune the alpha cut value as the search progress to make it explore or exploit. Further research in this is intended to make use of variable alpha cut and knowledge based operators to make GA more effective.

7. 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, 197-202.

[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] Elmer P. Dadios, "*Fuzzy Logic – Emerging Technologies and Applications", Intech Publications, 2012 pp. 184*

[10] 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

[11] 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.

[12] Reinelt G., "*TSPLIB – A Travelling Salesman Problem Library*". ORSA Journal on Computing, Vol.3, No.4, 376 – 384, 1991