Analysis of Roulette Wheel Selection and Steady state Selection using Genetic Algorithm Techniques

M.Mayilvaganan[#1], Geethamani G.S[*2]

[#] *Associate Professor: Department of Computer Science, PSG College of arts and science , Coimbatore, Tamil Nadu, India*
Assistant Professor, Research Scholar,Department of Computer Science, Karpagam University, Coimbatore, Tamil Nadu, India

Abstract— *Genetic algorithm installation with a inhabitants of folks represented by chromosomes. Each chromosome is evaluated by its fitness value as computed by the intent function of the crisis. In genetic algorithms, the roulette wheel selection operator has spirit of utilization while steady state selection is influenced by exploration. In this paper, a analysis of these two selection operators is proposed that is a ideal merge of both i.e. searching and utilization. The population undergoes conversion using three primary genetic operators – selection, crossover and mutation which form new generation of population. The proposed solution is implemented in MATLAB using DNA Nucleotide Sequence of Cancer cells and the results were compared with roulette wheel selection and steady state selection with different problem sizes.*

Keywords— *chromosome, roulette wheel, steady selection, crossover, mutation.*

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. They mimic the genetic processes of biological organisms. The population undergoes transformation using three primary genetic operators – selection, crossover and mutation which form new generation of population. The GA maintains a population of n chromosomes (solutions) with associated fitness values. Parents are selected to mate, on the basis of their fitness, producing offspring via a reproductive plan (mutation and crossover). Basic flowchart of genetic algorithm is illustrated in Fig. 1. Generally all the optimization techniques are influenced by two important issues - exploration and exploitation.

Exploration is used to investigate new and unknown areas in the search space and generate new knowledge. Exploitation makes use of the generated knowledge and propagation of the adaptations. Both techniques have their own merits and demerits. Genetic algorithm applied which has large solution search space[3]-[7]. Search space is space of all feasible solutions (the set of solutions among which the desired solution resides).

In the population of individuals, the strong ones survive longer than the weak ones "survival of the fittest". This causes a rise in the overall fitness of population. Based on the fitness value of the individuals, the week ones are terminated. The strong ones are chosen to reproduce themselves by using recombination and/ or mutation on them. Recombination is an action that is applied to two or more of the selected candidate (called parents). It will generate one or more new candidates (called children). Mutation is applied to one candidate and results in one new candidate. Execution of these two operators leads to a set of new candidates that compete with the old ones for a place in the next generation. When this process is repeated, the average fittest of the population will increase until a maximum has been reached.

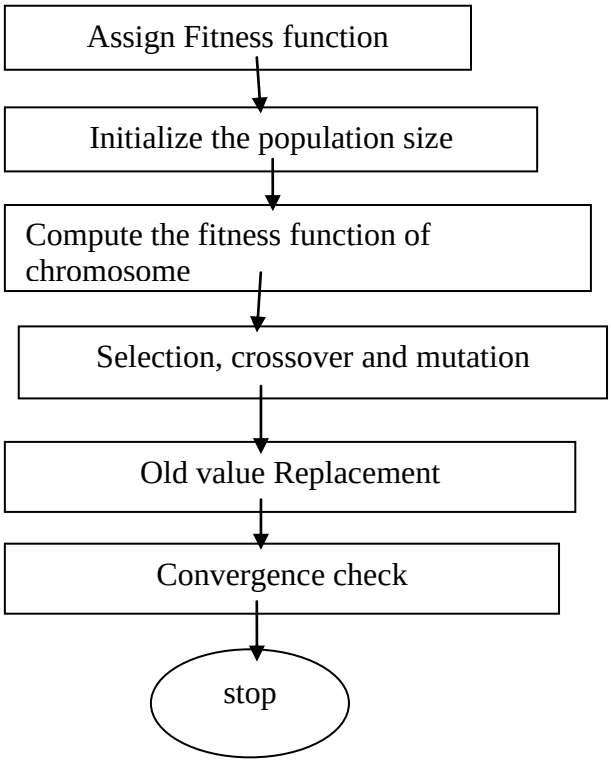

Fig 1 Genetic Algorithm Basic flow

In this paper, focus is to compare the two selection operators and generate a new selection

operator to obtain perfect mix of exploration and exploitation. The paper is organized in the following sections. The Section 2 reviews the different combination techniques related to this area. Section 3 focus on algorithm related to the proposed method. Section 4 focus on computational results. Finally, Sections 5 concludes the paper.

II. RELATED WORK

Holland showed that both exploration and exploitation are used optimally by genetic algorithm at the same time using k-armed bandit analogy [1]. This work is also described by David Goldberg [2]. It has been observed that due certain parameters, stochastic errors occur in genetic algorithms and this may lead to genetic drift [5,6]. In certain cases, selection operation gets biased towards highly fit individuals. This can be avoided by use of Rank Selection technique. Rank scaling ranks the individuals according to their raw objective value [2]. Another problem that arises with genetic algorithms is premature convergence which occurs when the population reaches a state where genetic operators can no longer produce offspring that outperforms their parents [7]. This would likely trap the search process in a region containing a non-global optimum and would further lead to loss of diversity. The same idea can be applied on the TSP problem by first finding the different solutions and then combine those, which are the fittest solutions among them, in order to create a new and healthy solution and should be optimal or near optimal according to the problem. [1] Selection is the process of choosing two parents from the population for crossing. The purpose of selection is to emphasize fitter individuals in the population in hopes that their off springs have higher fitness[8][9]. Chromosomes are selected from the initial population to be parents for reproduction. Selection is a method that randomly picks chromosomes out of the population according to their evaluation function. The higher the fitness function, the more chance an individual has to be selected. [2]

III . PROPOSED METHOD -*ROULETTE WHEEL SELECTION*

In genetic algorithm, first the population needs to be initialized with random candidate. Next step evaluate the fitness of individuals. Then they apply the selection operator which selects fittest parents for generations which are parents of next generation. The algorithm is stopped when the population converges toward the optimal solution. Roulette method selection method works similarly to a roulette wheel, where the likelihood that an individual is chosen is proportional to its fitness value[10][12]. Because the ideal fitness for individuals in this algorithm, the size of each individuals „slice" of the roulette wheel will be inversely proportional to their fitness value. Once the „slices" have been determined, a number is generated at random.

The individual with the range of numbers that contains this randomly generated number will be one parent[13][15]. This continues until the desired number of parents is found. Based on the value of fitness function, roulette wheel method selects the next best possible solution chromosome that will create a new generation and be genetic parents for the next generations. It is also allow for parents with low fitness to go to the next generation. [5]

Fitness function is the important parameter of a genetic algorithm that defines the fitness of each chromosome where the values of genetic parameters are adapted as the genetic evolution progresses. At every generation, fitness value of each chromosome is calculated using fitness function.

If fitness of two chromosomes is equal, then the mutation rate is increased, in order to help the genetic evolution get out of issues like local maxima or local minima whichever is applicable. Once there is an improvement in the overall fitness, the original mutation rate is restored to continue evolution as normal. If the evolution stabilizes, but the fitness does not seem to be improving for several generations and the search does not find any error, new set of initial population is generated using the initial default parameter values and a new randomly generated seed. [3][4] TSP is a minimization problem; we consider fitness function calculates cost (or value) of the tour represented by a chromosome in fig (2) and fig(3).

```
void rouletteParents(numParents, myPopulation,
parents){

int high = 0;
for each(chromosome : myPopulation)
if(chromosome.fitness > high)
high = chromosome.fitness;
create vector of integers = size of myPopulation;
for(0 <= i < myPopulation.size())
for(0 <= j < high – myPopulation.elementAt(i).fitness)
add i to vector of integers;
for(0 <= i < numParents)
shuffle vector of integers;
parents.add(myPopulation.elementAt(vector[i]);

}
```

Fig-2 Pseudo code for Roulette parent selection

Steady State Selection

In tournament selection, every individual in the population is paired at random with another. The

fitness values of each pair are compared. The fitter individual of the pair moves on to the next „round", while the other is disqualified. This continues until there are a number of winners equal to the desired number of parents[12]. Then this last group of winners is paired as the parents for new individuals.

```
void tournamentParents(numParents, myPopulation,
parents){
create temporary empty population;
create vector of integers = size of myPopulation;
add values 0 to index of last member of myPopulation
shuffle vector;
if(myPopulation.size <= numParents)
parents = myPopulation;
else
for(0 <= i < myPopulation.size/2){
compare myPopulation element at i and
myPopulation.size – i
add most fit to temporary population;
}
tournamentParents(numParents, temporary population,
parents)
}
```

Fig-3 shows the pseudo code for tournament selection method.

Crossover

After the completion of the selection process, the chromosomes chosen to be parents for the next generation are recombined to form children that are new chromosomes[14]. This combination can take many forms and the crossover operator can greatly affects the result of the search. [6]

Mutation

The operation of mutation allow new individual to be created[13]. It begins by selecting an individual from the population based on its fitness. A point along the string is selected at random and the character at that point is randomly changed, the alternate individual is then copied in to the next generation of the population.

IV.SIMULATION RESULTS AND DISCUSSIONS

In this paper, MATLAB code has been developed to assess the performance of genetic algorithm by using three different selection techniques on the same population . To begin with, the rank selection is applied and then the roulette wheel selection followed by the implementation of proposed blended selection operator on the same population.. Fig. 4 depicts the comparison of *FRW(Avergae fitness of population in roulette wheel) FR FX((Avergae fitness of population in steady state)* and Fig. 5 depicts the comparison of Fbest in three different selection methods.

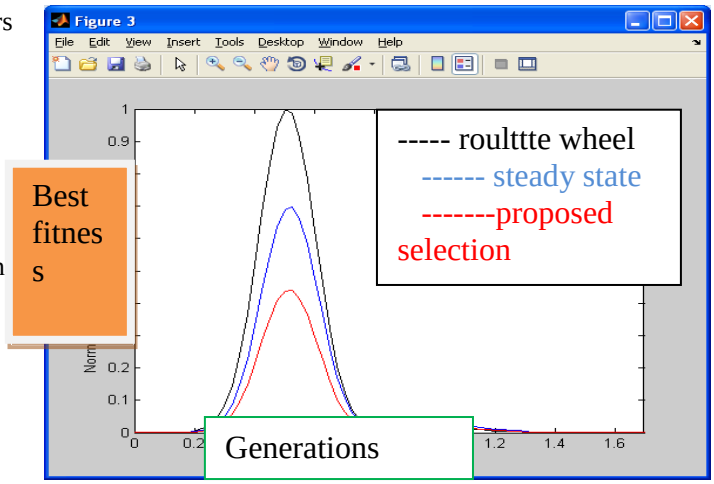

Fig 4 Generation of population size

Fig 4. best fitness of chromosome generations

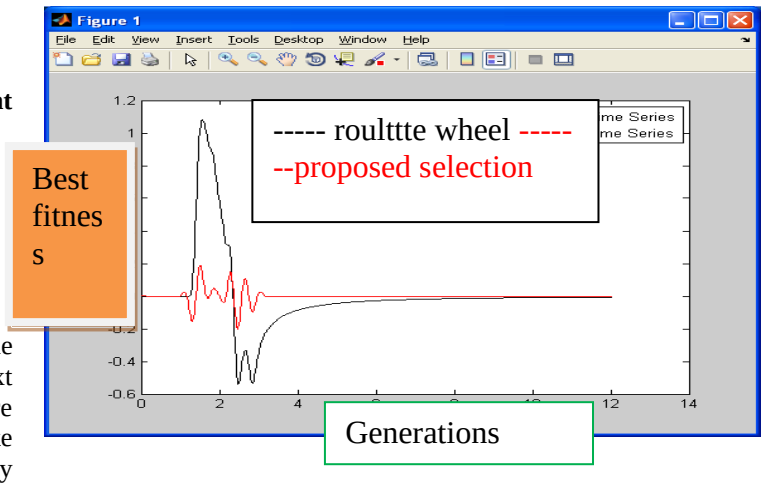

Fig 5. Comparision of minimum distance in nucleotide sequence

It was observed that the RWS had more of exploitation approach and found better chromosomes in early runs of generation and converged earlier than RS. On the contrary, RS had more of exploratory nature and kept on exploring new solutions and shown in fig(6). It has been found that with increasing problem size, problem did not converge prematurely and PBS gave better performance in each case.

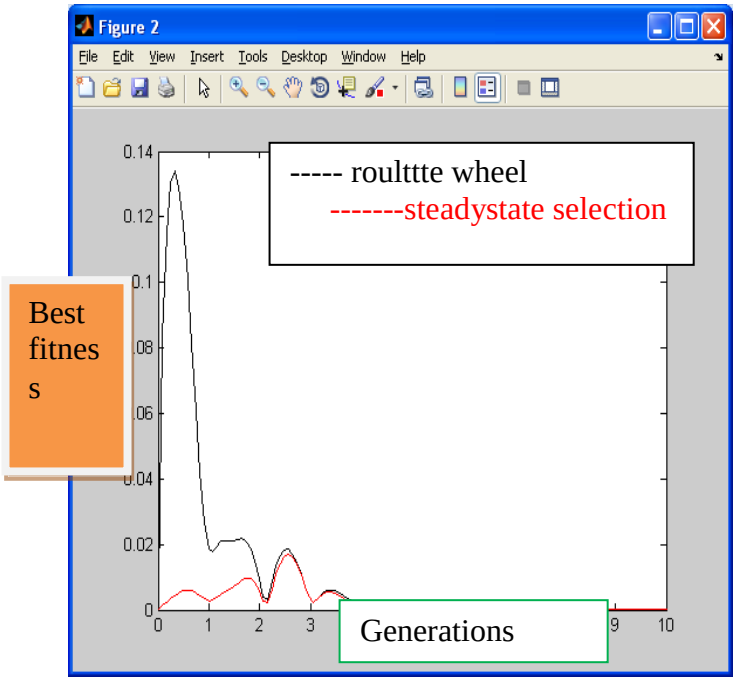

Fig 6 Comparision of minimum distance of nucleotide sequence

V. CONCLUSION

In this paper, a roultte wheel selection and steady state selection operator are analysed using the DNA nucleotide sequence of cancer cells.The performance of PBS selection operator is compared with RS and RWS technique on standard problem. RWS performed like nature selecting the most fit individuals. RS did more of exploration and maintained diversity in population. Further research in this area is analyse the various factors influencing performance of genetic algorithms.

REFERENCES

[1] J. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, 1975.

[2] D. E. Goldberg, *Genetic algorithms in search, optimisation, and machine learning*, Addison Wesley Longman, Inc., ISBN 0-201-15767-5, 1989.

[3] P. Merz and B. Freisleben, "Genetic Local Search for the TSP: New results", Proceedings of IEEE International Conference on Evolutionary Computation, IEEE Press, pp 159-164, 1977.

[4] S. Ray, S. Bandyopadhyay and S.K. Pal, "Genetic operators for combinatorial optimization in TSP and microarray gene ordering", SpringerScience + Business Media, LLC, 2007.

[5] D. E. Goldberg and P. Segrest, "Finite Markov chain analysis of genetic algorithms", Proceedings of the the Second International Conference

[6] L. Booker, *Improving search in genetic algorithms*, Genetic Algorithms and Simulated Annealing, Pitman, chapter 5, pp 61-73, 1987.

[7] D. Fogel , "An introduction to simulated evolutionary optimization", IEEE Trans. Neural Networks 5 (1), pp 3-14, 1994.

[8] Dan Adler, "Genetic Algorithms and Simulated Annealing: A Marriage Proposal", IEEE International Conference on Neural Networks 1993, San Francisco (CA), 1993.

[9] A. E. Eiben and,C.A.Schippers, *On evolutionary exploration and exploitation*, Fundamenta Informaticae, 35 IOS Press, pp 1-16, 1998.

[10] G. Van Dijck, M. M. Van Hulle and M.Wevers, "Genetic Algorithm for Feature Subset Selection with Exploitation of Feature Correlations from Continuous Wavelet Transform: a real-case Application", International Journal of Information and Mathematical Sciences 1:4 2005 pp 233-237, 2005.

[11] O. Al *jaddan*, L. Rajamani and C. R. Rao, "*Improved Selection Operator for GA",* Journal of Theoretical and Applied Information Technology, pp 269–277, 2005.

[12] A. Tsenov, "Simulated Annealing and Genetic Algorithm in Telecommunications Network Planning", International Journal of Information and Mathematical Sciences 2:4, pp 240-245, 2006.

[13] A. E. Eiben, M. E. Schut and A. R. de Wilde, "Boosting Genetic Algorithms with Self-Adaptive Selection", IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, pp 1584-1589, 2006.

[14] Z. G. Wang, M. Rahman, Y. S. Wong and K.S.Neo, "Development of Heterogeneous Parallel Genetic Simulated Annealing Using Multi-Niche Crowding", International Journal of Information and Mathematical Sciences 3:1, pp 55-62, 2007.

[15] S. B. Liu, K. M. Ng and H. L. Ong, "A New Heuristic Algorithm for the Classical Symmetric Travelling Salesman Problem", International Journal of Computational and Mathematical Sciences, Volume 1, Number 4, pp 234-238, 2007.