ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

Performance Comparison of Roulette Wheel Selection and Steady state Selection in Genetic Nucleotide Sequence

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

Associate Professor: Department of Computer Science, PSG College of Arts and Science, Coimbatore, Tamil Nadu, India[#1]

Assistant Professor, Research Scholar, Department of Computer Science, Karpagam University, Coimbatore, Tamil Nadu, India[*2]

ABSTRACT: In this paper the attempt has been made analyse the performance comparison of nucleotide sequence in DNA cells. The basic idea behind this proposed method is estimate the efficiency of roulette wheel selection and steady state selection with respect to memory and running time. Here this algorithm is applied to find the genetic operator such as mutation, crossover and selection in large dataset. In order to evaluate the proposed methodology, Comparisons are made based on the Execution time and memory efficiency in finding best fitness value in each generations and their distance estimation between each generations.. The extracted rules and analysed results are graphically demonstrated. The performance is analysed based on the different no of instances and confidence in DNA sequence data set. In genetic algorithms, the roulette wheel selection operator has spirit of utilization while steady state selection is influenced by exploration. 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

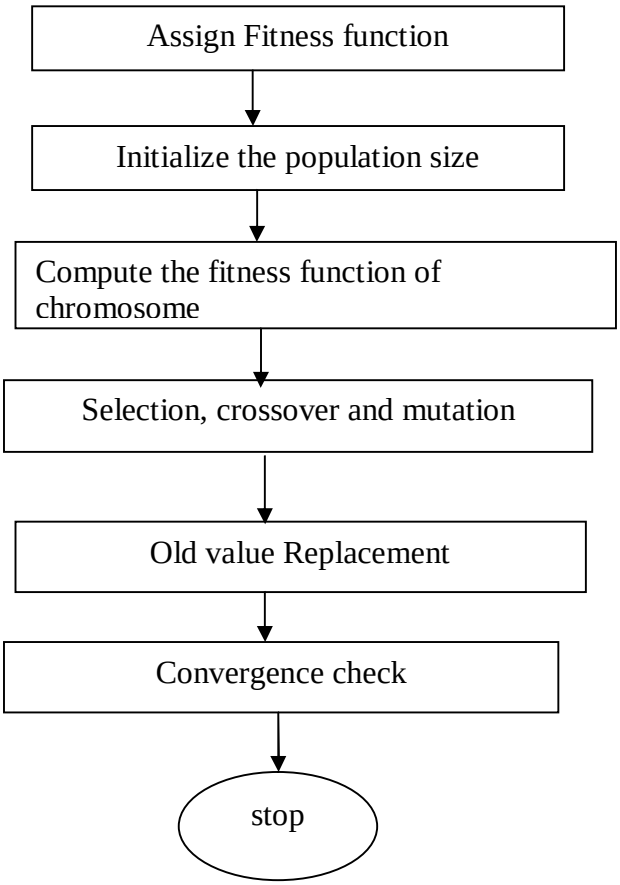

Fig 1 Genetic Algorithm Basic flow

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

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]

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

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 . Figures(5) and (6) shows that amount of used memory, efficiency and speed in the proposed algorithm that is less than the steady state selection method. This is because of the compact data structure and matrix that growths quickly in linear format. Definition of response time is the time required to produce output regardless of the output quality.

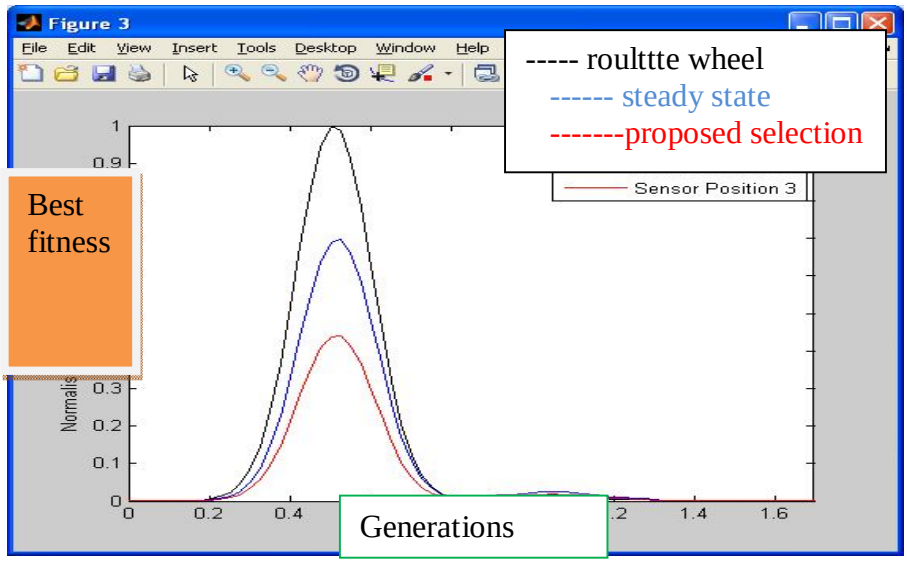

Fig 4 best fitness of chromosome generations

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

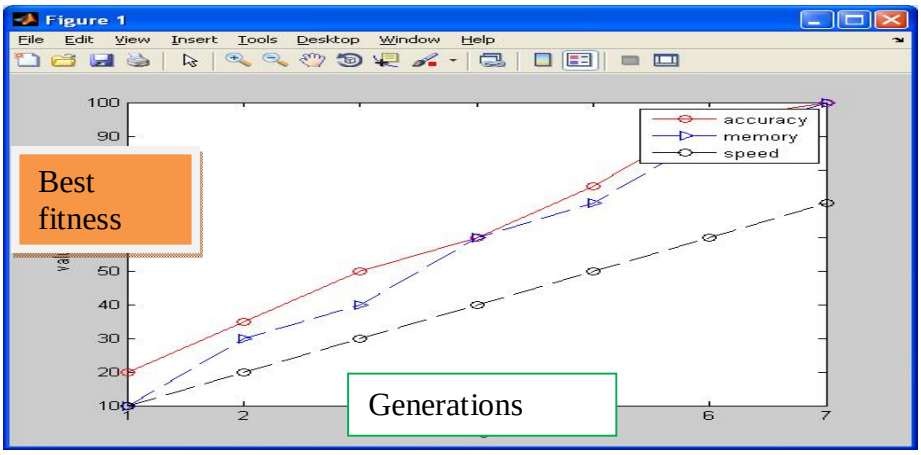

Fig 5. Performance Comparision within genetic methods

The implementation results show that the roulette wheel method can improve the quality, memory and speedup metrics. Results show that in average efficiency 81%, memory usage 94.3% and response time 0.69% as compared to steady state algorithm are improved.

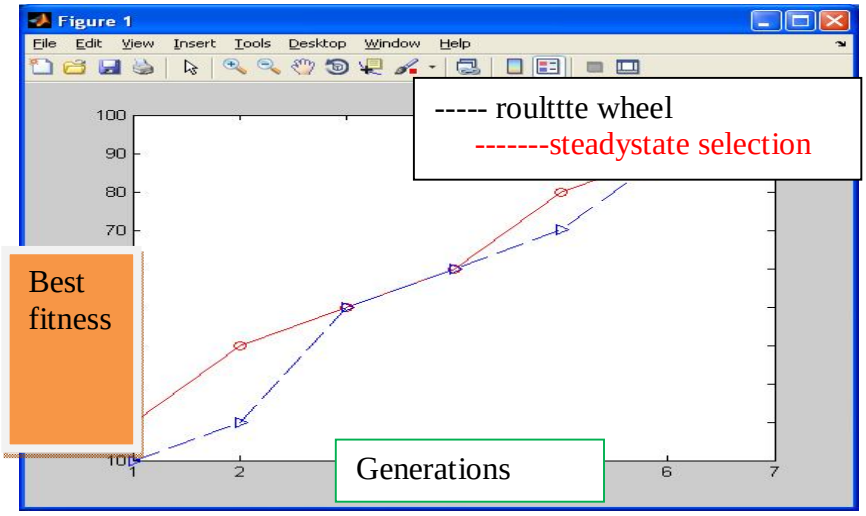

Fig 6 Comparison of roulette wheel and steady state selection

V. CONCLUSION

In this paper, the proposed roulette wheel selection and steady state selection method are used for genetic DNA nucleotide set sequence Our proposed algorithms included two phases: phase I is related to create and estimate the best fitness value for each generation. In phase II, roulette wheel selection compare the fitness value in convergence value to perform genetic operator such as selection, crossover and mutation. The implementation results show that the roulette wheel selection method should improve the quality, memory and speedup metrics. Results show that in running time memory usage and speed are better compared to steady state algorithm. . In future the work will be extended in the following areas Further research in this area is analyse the various factors influencing performance of genetic algorithms.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

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.