ISSN: 2182-2069 / E-ISSN: 2182-2077

Enhanced Fitness Proportionate Selection Algorithm for Parent Selection in Genetic Algorithms

Rawan Nassri Abulail[1*]

[1*]Associate Professor, Computer Science Department, Philadelphia University, Jordan.
rabulail@philadelphia.edu.jo, https://orcid.org/0009-0002-8168-2455

Received: November 30, 2024; Revised: January 12, 2025; Accepted: January 27, 2025; Published: February 28, 2025

Abstract

A genetic Algorithm is an evolutionary algorithm that models and simulates biological behavior, whether evolution or genetics, to reach a high-quality solution for search and optimization problems. There are many areas and applications to which genetic algorithms can be applied, like machine learning, feature selection, engineering design, and function optimization. Three leading operators must be applied to each generation's reproduction process; the first is the Selection process, which is applied to the initial population to select the candidate parents to mate and recombine to produce the next generation(offspring). The second operator is a crossover, which is applied to the selected parents from the previous operation (Selection) to make new individuals (offspring) carrying the same traits from parents by combining the parent's chromosomes; the last operator is a mutation, which is applied to the new offspring after crossover. Mutation operation aims to change the value of the chromosome gene randomly. In this research, the selection process will be demonstrated in detail. Then, fitness proportionate selection (FPS) will be presented as one of the most popular methods used in the selection process. The main problem of FPS is the candidate parent, which will mate and recombine to reproduce the next generation; in some cases, a strong individual can mate with a weak one and produce offspring with lower quality traits than the strong parents as a consequence of trait exchange, which happens between that pair. The researcher proposed an enhancement of the FPS algorithm to ensure that strong parents will mate and reproduce strong offspring and propagate their strong traits to the next generations; the proposed enhancement can be summarized as adding a step to the standards FPS to sort the selected individual in ascending or descending order after selection process and before applying cross over and mutation phases. The researcher conducted three experiments to prove the improvements in the fitness value as a consequence of applying that additional step in the selection algorithm; the experiments were performed with three different population sizes and reproduced 100 generations. The fitness score was measured in each generation, and the researcher presented the fitness score evolution over the GA iterations. The results were precise, proving that the sorted individuals after Selection gave better fitness scores than those obtained by applying the standard FPS.

Keywords: Genetic Algorithm, Reproduction Operators, Selection, Fitness Proportional Selection (FPS).

1 Introduction

Genetic Algorithm (GA): is an evolutionary algorithm that models and simulates biological behavior, whether evolution or genetics, to reach a high-quality solution for search and optimization problems (Rowe,

Journal of Internet Services and Information Security (JISIS), volume: 15, number: 1 (February), pp. 257-270.
DOI: *10.58346/JISIS.2025.I1.016*

*Corresponding author: Associate Professor, Computer Science Department, Philadelphia University, Jordan.

2012). It can also be defined as an adaptive heuristic search algorithm based on the principles of genetics and natural Selection (Dimitrov & Baumann, 2011). The natural selection process means the survival ability of strong species that can adapt to any environmental changes and reproduce for generations (Tinós et al., 2023). In other words, "survival of the fittest". The natural life cycle of reproduction is presented in GA as follows in figure 1 (Yu et al., 2009): Real-life environment: optimization problem in GA, species living in that environment: feasible solutions, adaptation degree of species in their surrounding environment: quality of solutions (score of fitness function), A population of species: A set of feasible solutions, evolutionary process in nature selection, recombination and mutation: GA stochastic operators (Selection, crossover and mutation), population evolution to adapt their environment: Applying the GA stochastic operators on the set of feasible solutions (Kawulok & Kawulok, 2022).

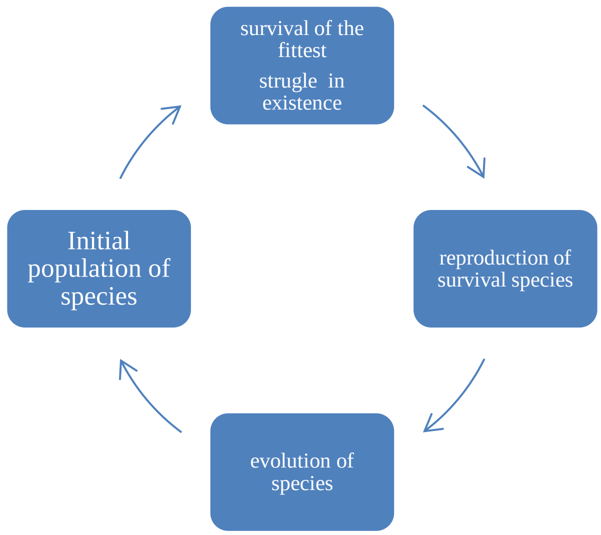

Figure 1: Natural Life Cycle Reproduction

Problem Statement

In this research, the selection process will be demonstrated in detail. Then, fitness proportionate selection (FPS) will be presented as one of the most popular methods used in the selection process (Al-Omari & Al-Haija, 2024). The main problem of FPS is the candidate parent, which will mate and recombine to reproduce the next generation. In some cases, a strong individual can mate with a weak one and produce offspring with lower-quality traits than the strong parents due to the characteristics exchanged between that pair.

Research Objectives

The main objective of this research is to propose an enhancement of the FPS algorithm to ensure that strong parents will mate and reproduce strong offspring and propagate their strong traits to the next generations. The proposed enhancement can be summarized as adding a step to the standard FPS to sort the selected individual in ascending or descending order after the selection process and before applying cross-over and mutation phases (Prakash & Prakash, 2023).

Research Structure

Five sections will be presented in this research. The first will introduce the genetic algorithm's meaning and importance. In contrast, the second section will cover the background and the literature reviews related to genetic algorithms, a flowchart of genetic algorithms, genetic operators, the selection process, and the different selection algorithms in GA. The third section will show the research methodology, followed by the fourth section, which will display the results of the experiment which the researcher conducted, and finally, the fifth section with the discussion and conclusion of the research (Flores-Fernandez et al., 2024)

2 Background and Literature Review

The following section will present the main key concepts related to the research area, such as genetic algorithms, genetic operators, Selection genetic operators, and the different algorithms used by genetic algorithms for the selection phase (Jiménez-Carrión et al., 2023).

Genetic Algorithm

A genetic algorithm is one of the optimization algorithms that simulates the natural genetic inheritance and Selection. It implements the evolutionary process of an initial population of species (organisms); individuals with desirable traits have a higher chance to survive and reproduce, and the new offspring of the species have the same desirable traits (Gliesch et al., 2017). In GA, the species are presented as chromosomes, each composed of several genes that determine the variable values to solve a specific problem (Hasegawa et al., 2017). The algorithm starts by generating a random initial population of chromosomes and evaluating the fitness value for each chromosome to find the selection probability; after finding the selection probability values, these values will be used in applying the genetic operator's Selection, crossover, and mutation to produce a new generation (offspring). (Hasegawa et al., 2017)

In the crossover process, each pair of chromosomes (parents) are combined to produce a new pair of chromosomes (offspring), inheriting all or some traits from their parents (Kapoor & Pillay, 2023). The mutation randomly changes the chromosome's genes to introduce new traits and features into the new generations (Doan et al., 2020). These reproduction processes are repeated for several generations or iterations; in each iteration, the new population is evaluated by calculating the fitness function to find fitness values; the reproduction process is stopped if the fitness scores in a generation converge to the target fitness scores (Oliveto & Witt, 2012). While in some cases the algorithm will stop if the maximum number of iterations is reached. There are many areas and applications to which genetic algorithms can be applied, like machine learning, feature selection, engineering design, and function optimization (Rowe, 2012; Onorato et al., 2024). Genetic algorithm basic operators: Three leading stochastic operators must be applied for the reproduction of each generation and can be summarized as follows:

- Selection: The first operation applied to the initial population to select the candidate parents to produce the next generation. The selected parents depend on their value of fitness function (Bahramvash & Nikaeen, 2016). If the value of fitness function for a specific individual is high, then the opportunity of choosing that individual is also high; the selected parent can pass their strong traits to the next generation (offspring). Many methods are used for selection in GA, like rank-based selection, fitness proportionate selection (roulette wheel), and tournament selection. All of those selection methods rely on the probability of Selection (Tahera et al., 2007).
- Crossover: This process is applied to the selected parents from the previous operation (Selection) to produce new individuals (offspring) carrying the same traits from parents by combining the parents' chromosomes. A crossover points or points in the chromosomes are chosen randomly to swap the values of genes in those points. Consequently, the offspring will carry new trait combinations, increasing population diversity (Iqbal & Hoque, 2016).

- Mutation: This is the last operation applied to the new offspring after crossover; this operation aims to change the value of the chromosomes gene randomly depending on the comparison between the value of mutation probability and a random number; in most cases, the probability of mutation is very low, but it is essential mainly that it can prevent the new generation from prematurely converging to a local (value) optimal solution (Ahn et al., 2010).

These three operators are the fundamental operators in GA, while many other advanced operators, such as elitism, transposition, inversion, and gene transfer, can improve the genetic algorithm's performance (Torchinskii et al., 2018).

Genetic Algorithm Flowchart: Genetic algorithm steps can be visualized using flowcharts. They are a valuable tool for developers to simplify the implementation of GA and provide a structured and precise representation of the algorithms. Here is the flowchart of the GA structure shows in figure 2 (Kapoor & Pillay, 2023):

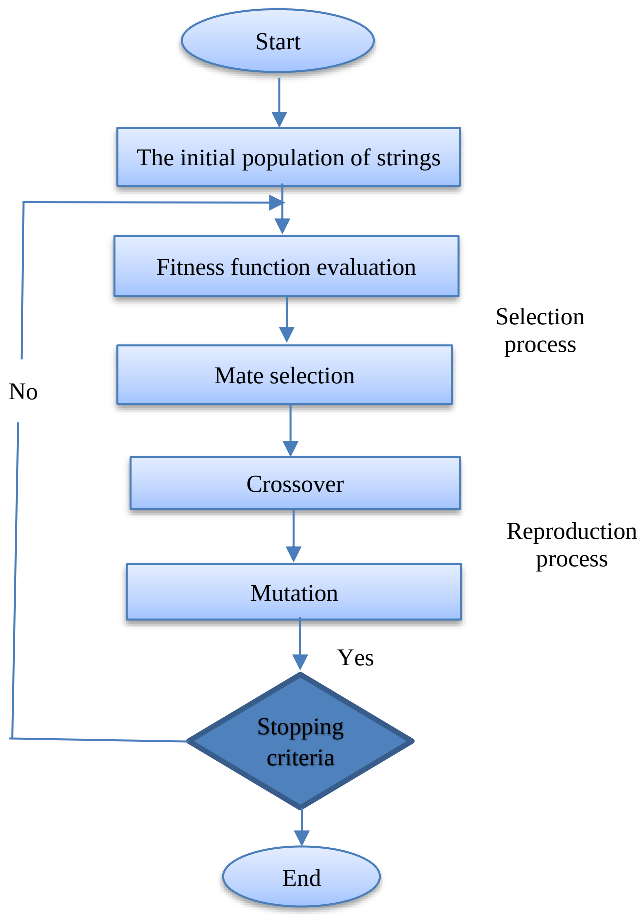

Figure 2: Structure of GA

The Selection Process in Genetic Algorithms

The most important phase in GA programming is the parent selection phase, which determines the candidate chromosomes from the initial population to be parents for the reproduction process to produce new offspring in the next generations (Tatsumi et al., 2017). also, the number of created children must be determined in the cross-over phase; many methods were presented in GA research papers dealing with parent selection; the common goal of all those methods is how to improve the performance of problem-solving with addition to improving the chromosome characteristics in the evolved generations (Marrero et al., 2021). Parent selection can be defined as the parent's selection process in which they mate and recombine to produce the next generation of offspring; it is a very crucial step because it directly affects the GA convergence rate (Helmuth & Abdelhady, 2020); parents with good characteristics can quickly drive chromosomes to a fitter and better solutions, this can be measured by fitting scores for the offspring or the produced generations (Marrero et al., 2021). One of the issues that must be considered in the selection process is avoiding the fit solutions extremely (Smith, 2007; Maltare et al., 2023). That means avoiding the power of one dominant chromosome in the reproduction process in many sequential generations. The consequences will lead to the closeness of solutions in the solution space, which makes the diversity of solutions very low, and this is an undesirable issue in GA (Helmuth & Abdelhady, 2020). The GA's success strongly depends on the diversity of the candidate solutions, so the GA must maintain an acceptable degree of diversity of candidate solutions (chromosomes) in the solution space (Nazmul & Chetty, 2013). One of the issues related to the dominant chromosomes is called premature convergence, which all GA developers try not to reach (Acampora et al., 2021).

Parent selection in GA: These methods can be used in the selection phase to select candidate parents in the next generation. The most popular method is Fitness Proportional selection; in this method, any chromosomes from the initial population have a chance and are eligible to be selected as parents in the reproduction process depending on their selection probability, which can be defined as proportional to their fitness. Therefore, chromosomes with desirable characteristics (fitter) have a higher chance of being chosen as parents to be mated and reproduce offspring that propagate their features to the next generations (Nazmul & Chetty, 2013). The Importance of the propagated features in the reproduced generation requires a selection strategy that guarantees choosing more fit chromosomes from the population to produce better-evolved chromosomes over time (Helmuth & Abdelhady, 2020). Fitness Proportional selection can be implemented in roulette wheel selection, tournament selection, Rank Selection, and Stochastic Universal Sampling (Acampora et al., 2021).

Genetic Algorithm Parent Selection Methods

Selection can be defined as a stage in evolutionary algorithms, especially in genetic algorithms, where several chromosomes are selected from an initial population of chromosomes as candidate parents to be mate and exchange their features with the produced offspring; these selected chromosomes are potentially high-quality solutions to recombine and produce new generations, the process of features exchange is the output of the crossover and mutation phases (Liu & Shang, 2014). Many algorithms can be applied in the selection phase in the evolutionary algorithms and will be listed in this research.

- Fitness proportionate selection (FPS), also known as Roulette Wheel Selection: In this method, the chance for choosing any chromosome to be selected as a candidate parent depends on the proportional of chromosome fitness to the summation of all fitness in the initial population (Gupta, 2009). The most important feature of the FPS algorithm is that there is always a chance, even for weak chromosomes, to survive in the next generations, unlike the truncation selection algorithm, which refers to the probability of selection technique used by FPS. Also, this feature can help offer a diversity of solutions, which can lead to improvements in the new offspring after the recombination process (Hart et al., 2015). FPS will be presented in detail later in this research.

- Rank selection: the rank selection algorithm similar to some extant to FPS, the main idea in this algorithm is to sort the initial population upon their fitness function and the probability of Selection is then calculated upon the chromosome rank in the population (Nuhu et al., 2021). Also, this algorithm gives a chance for weak chromosomes to survive and evolve in the next generations (Corus et al., 2021).

- Steady State Selection: This selection algorithm relies on two stages. The first stage selects a certain percentage of high-quality chromosomes from the initial population and lets them recombine and produce new offspring. The second step is to replace the old low-quality chromosomes from the initial population with the new high-quality offspring (Valem & Pedronette, 2019).
- Reward-based Selection: This selection algorithm depends on the principle of rewarding; the rewards are granted to the chromosome upon their actual Selection in the new generations; in simple words, if an offspring is selected in a particular generation, then it is rewarded and also his parents get rewards, the added value of the rewards improve the probability of Selection of that offspring in the next generations, the most important advantage of reward-based Selection is the speedy process for the identifications of the most fruitful chromosomes in the recombination process, by cumulative rewards for that chromosome (He et al., 2010).
- Tournament Selection: The tournament selection algorithm depends on the idea of conducting several tournaments among groups of chromosomes selected randomly; the process of deciding the winner depends on the fitness score for the chromosomes in the tournament after finishing all tournaments and after achieving the desired number of selected chromosomes, these chromosomes go to the next step of recombining and reproduce new offspring by applying the crossover and mutation processes. The number of tournaments set by the algorithm has an effect on the quality of the selected parents; that means if there is a large number of tournaments, the checking which each candidate's parents will be stricter while evaluating the high quality and the fitness score value (Pantridge et al., 2018).
- Stochastic universal sampling: This algorithm is extended from Fitness proportionate selection (FPS) with a main modification of setting multiple points for the selection process; the fitness value and the fitness proportion are calculated in the same way used in FPS, just the selection point is changed in FPS is just one point while in SUS multiple points (Tallón-Ballesteros et al., 2020).
- Truncation selection: This method simulates the natural breeding process, which evaluates the fitness score of the initial population of chromosomes. Then, a predefined portion of the weak chromosomes is replaced by high-quality chromosomes, recombined, and strong offspring reproduce, carrying the same features from the selected parents (Motoki, 2002).
- Elitist Selection: This method is carried out by partial recombination of strong chromosomes while keeping some chromosomes that carry very high-quality features (elite chromosomes) without any change or modification to the next generation (Du et al., 2009).

Problem statement: According to Darwin's evolution theory 1875, the best and strongest individuals or species survive to participate in the generation's reproduction. Evolution comes from changing organisms over generations by exchanging the features of parents and offspring. So, the most important issue in the reproduction process is selecting the parents; the selected parent will propagate their traits to their offspring and the next generations. Fit and strong species in the population evolve better species over time. Selection is one of the leading GA operators; there are two main Approaches to Selection: the first one is (Parent Selection) and the second one is (Survivor Selection). Parent selection means selecting parents from the current generation and applying the reproduction process to generate offspring.

In contrast, survivor selection means selecting parents from the parents and their offspring to reproduce new generations. In this research, the researcher will demonstrate the fitness Proportionate selection (FPS) algorithm as one of the most common algorithms applied in the selection stage in the GA reproduction process. After that, the researcher will propose a new modification to improve the fitness function scores of the reproduced generation.

Fitness Proportionate Selection

Fitness Proportionate Selection (FPS): Fitness Proportionate Selection is an algorithm used to select parents from the initial population to reproduce offspring of that generation. This algorithm is also known as Roulette Wheel Selection; each individual or chromosome in the population is assigned a fitness score from the optimized function, and then each fitness score is used to calculate the probability of Selection for each chromosome. The selection probability is calculated by dividing the chromosome fitness score by the summation of all fitness scores in the population. Probability to select x_i= fitness score of xi/sum of fitness scores, this algorithm is called fitness proportionate selection because the probability of Selection depends on proportional of the summation of fitness in the population. FPS is considered one of the most popular selection algorithm because it gives the chromosomes even with low probability of Selection a chance to survive in the selection process.

FPS algorithm steps

- For each chromosome in the population calculate the fitness score by applying the function being optimized to the chromosome value.
- Calculate the fitness scores summation for all chromosomes in the population.
- For each chromosome calculate the selection probability by dividing the chromosome fitness score by summation of fitness score for the whole population 4- Generate random numbers between 0 and 1.
- For each chromosome, make a comparison between its probability of Selection with the generated random number; if the random number is less or equal to the probability of Selection, then select that chromosome as a candidate parent for recombination.
- Step five is repeated until the same number of the initial population is reached.

Fitness Proportionate Selection algorithm enhancement: The researcher has proposed an enhancement to the standard Fitness Proportionate Selection (FPS) algorithm by adding an additional step to the standard algorithm. The main idea is to rearrange the selected chromosome in an ascending or descending order after completing the selection process, this step will ensure that the strongest chromosomes will mate together and reproduce the offspring and as mentioned before in this research Fit and strong species in the population evolve better species over time. And to insure that fit and strong species will mate together we must force them to mate to each other. To simplify the idea, FPS gives chromosomes with low probability of Selection a chance to survive in the selection process. That means that the selected chromosomes may have weak chromosomes with low fitness scores, the problem in this case that when the parents are selected as pairs to reproduce new offspring, there is a chance that strong chromosome is mate with weak chromosome leading to decrease the strong traits in their offspring and next generation as a consequence of traits swapping, this case can be avoided by force the algorithm to aggregate the strong chromosomes to gather in order to mate and reproduce new strong off spring , this can happened by reordering the selected chromosomes upon their fitness score then choose parents in a sequential way. For example, suppose that after completing the selection process for a given maximization optimization problem, the selected chromosomes and their fitness scores are as follows:

Table 1: Calculating Fitness Scores

Selected chromosome	Fitness score (maximum optimization)
X1 11111	961
X2 00001	1
X3 11110	900
X4 00010	4
X5 11101	841
X6 11110	900

The standard FPS algorithm will divide them into three pairs (x1, x2), (x3, x4), and (x5, x6), then apply the crossover and mutation phases for them. The above table 1 shows that x1 is the fittest chromosome will mate to the weakest chromosome x2, which means if any swapping of values (traits) happened in any digit except the first one it will destroy x1's strength and the offspring of that pair will not have the same strong traits, x3 and x4 will face the same problem. But if the proposed enhancement is applied to the selected chromosomes, the parent's pairs will be as follows

Table 2: Reordering Population Regarding Fitness Scores

Selected chromosome	Fitness score (maximum optimization)
X1 11111	961
X3 11110	900
X6 11110	900
X5 11101	841
X4 00010	4
X2 00001	1

In table 2 shows the result of applying the reordering step to selected chromosomes is ensuring that strong chromosomes will mate to each other as parents and go through to reproduce strong offspring by maintaining the strong traits to their off spring, the parent's pairs in this case will be (x1, x3), (x6, x5), and (x4, x2). That means that there will be a guarantee that the offspring will carry strong traits (high fitness score), which means higher probability to be selected in next generations.

FPS enhancement algorithm

- For each chromosome in the population calculate the fitness score by applying the function being optimized to the chromosome value.
- Calculate the fitness scores summation for all chromosomes in the population
- For each chromosome, calculate the selection probability by dividing the chromosome fitness score by the summation of the fitness score for the whole population
- Generate random numbers between 0 and 1
- For each chromosome, make a comparison between its probability of Selection with the generated random number; if the random number is less or equal to the probability of Selection, then select that chromosome as a candidate parent for recombination
- Step five is repeated until the same number of the initial population is reached
- Reorder (sort) the selected chromosome in a descending order then choose the pairs of parents for the next stages in the reproduction process.

3 Research Methodology

The researcher had conducted an experiment to show how the growth of fitness function scores was affected by random Selection of parents which used by the basic FPS and the enhanced FPS which was proposed in this research

The researcher had conducted three different experiments using Genetic algorithm to optimize a function f(X)= $\mathbf{x^2}$ for maximization, the first one with population size= 4 and the second one with population size = 10 while the third one with population size = 20, for each population size the researcher had implemented 10,000 iteration to produce 100 new generations, the researcher use a chromosome length =16 bit, cross over probability was set to 0.3 while the mutation probability was set to .03, the random seed was used to generate random numbers to make the comparisons needed in the Selection, cross over and mutation phases, the experiments details can be summarized as flow in table 3:

Table 3: The Conducted Experiments Details

Population size	No. of iterations	Function being optimized for maximization	FPS	Enhanced FBS
4	10000	$F(x)= x^2$	Fitness scores average	Fitness scores average
10	10000	$F(x)= x^2$	Fitness scores average	Fitness scores average
20	10000	$F(x)= x^2$	Fitness scores average	Fitness scores average

In each iteration the average of fitness scores was calculated for the new population (offspring), then the evolution of fitness scores average was visualized and represented using a chart to show clearly how the fitness scores average was increasing in both FPS and Enhanced FPS.

4 Experiments Results

The following charts represent the fitness value evolution for one hundred generation reproduced using the standard FPS and the proposed enhanced FPS, each chart shows the result for certain number of populations, population size in (figures 3 and 4) is 4, in (figures 5 and 6) is 10, and in (figures 7 and 8) is 20.

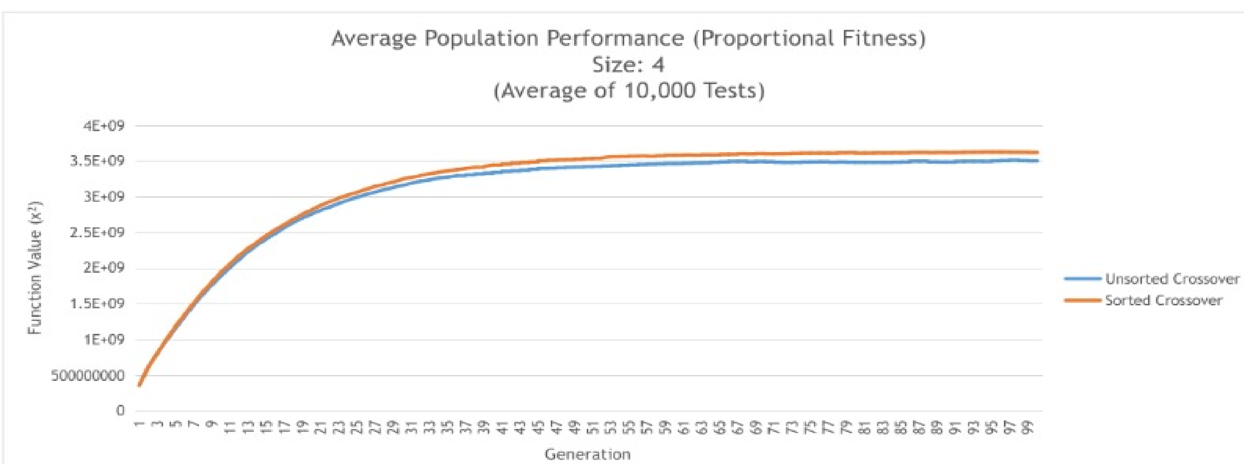

Figure 3: Generations Fitness Average for Standard FPS and Enhanced FPS, Population Size = 4

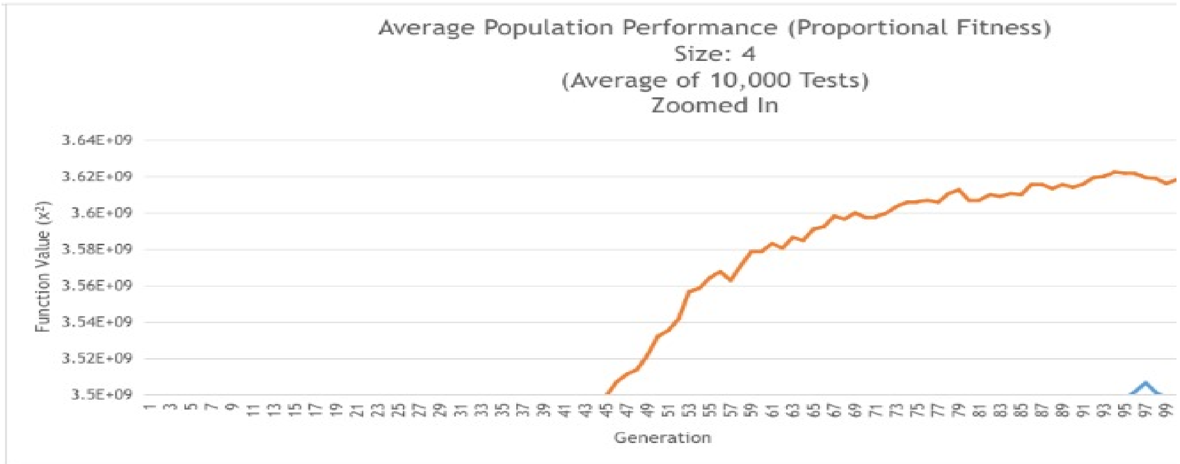

Figure 4: Generations Fitness Average for Standard FPS and Enhanced FPS, Population Size = 4 Zoomed in

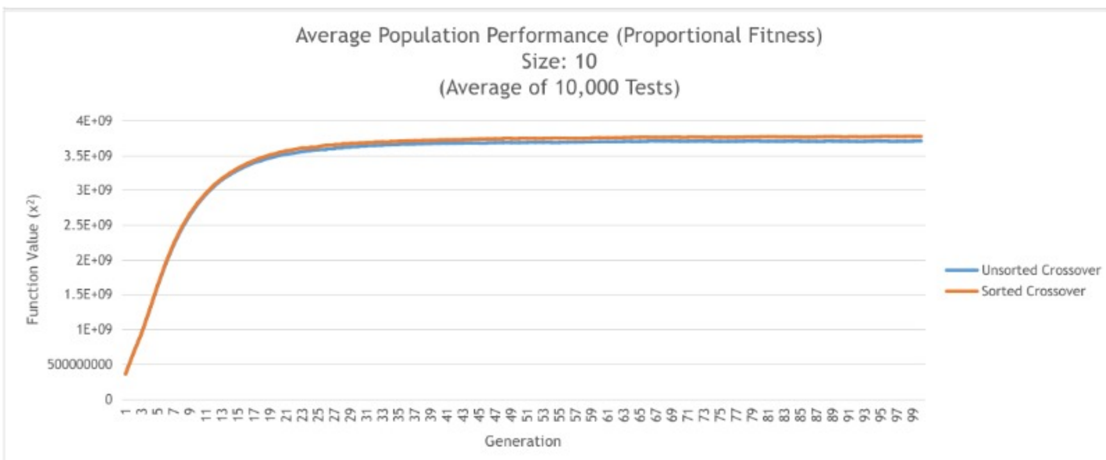

Figure 5: Generations Fitness Average for Standard FPS and Enhanced FPS, Population Size = 10

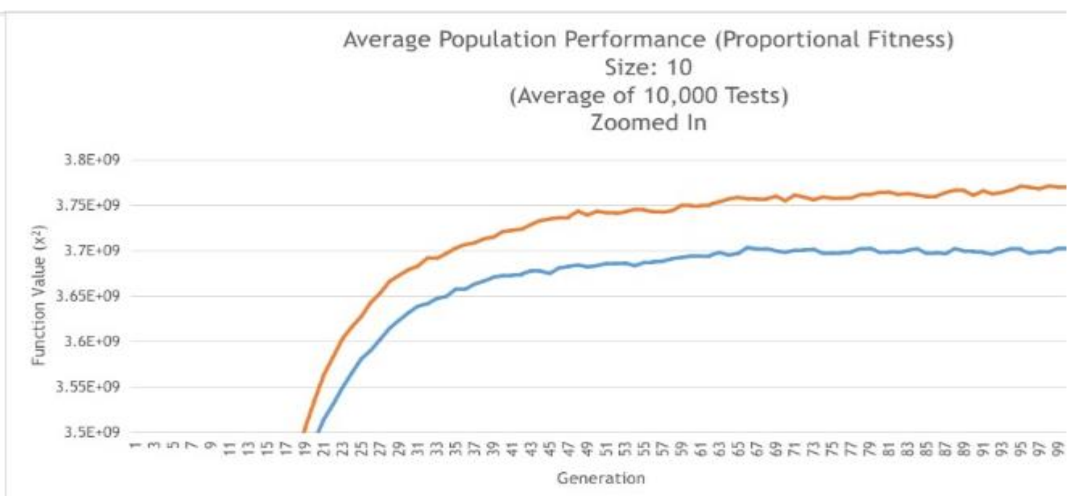

Figure 6: Generations Fitness Average for Standard FPS and Enhanced FPS, Population Size = 10 Zoomed in

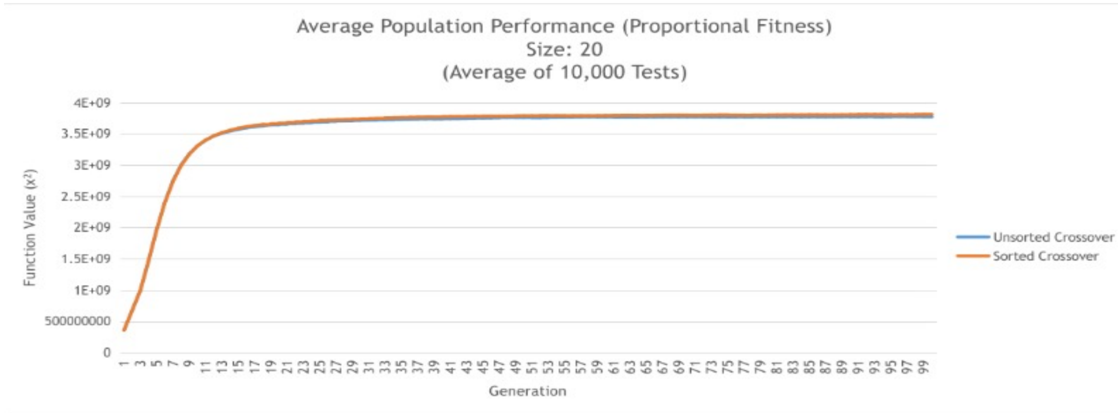

Figure 7: Generations Fitness Average for Standard FPS and Enhanced FPS, Population Size = 20

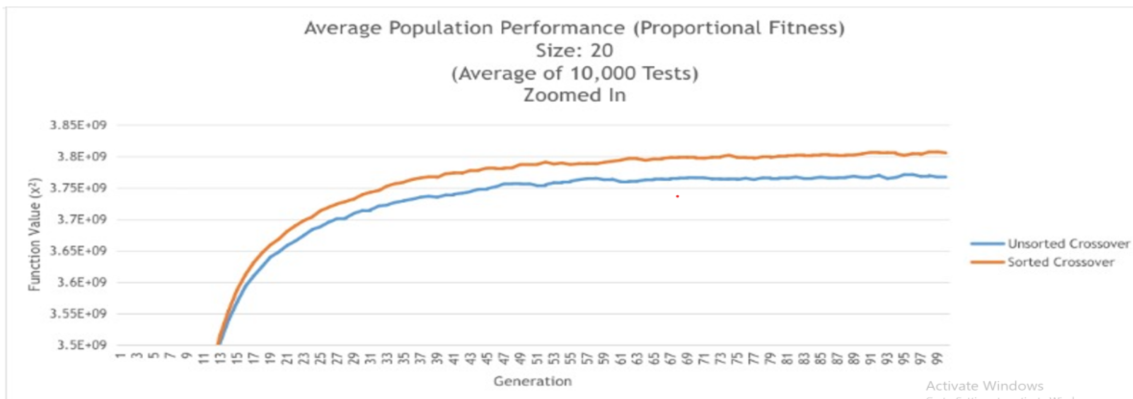

Figure 8: Generations Fitness Average for Standard FPS and Enhanced FPS,
Population Size = 20 zoomed in

5 Results Discussion and Conclusion

The obtained results from applying FPS and Enhanced FPS as displayed in the previous section show clearly the positive effect of applying the new enhanced FPS to the selection process in Genetic algorithms and prove that the main problem of the standard FPS is the parent pair selection after finishing the selecting process, that’s refer to the probability of strong chromosomes can mate and recombine with weak chromosomes which lead to produce off spring not strong enough as the strong parent and in some situation it leads to lose some of strong traits which carried by parents, while the proposed enhanced FPS algorithm will sort the chromosomes after selection process upon their fitness values and after that select the pairs of parents sequentially, this sorting guarantee that strong chromosomes will mate together and recombine to reproduce strong offspring as a consequence of reproduction process . The results obtained from the conducted three experiments show that the result of applying the enhanced FPS is always better than the standard FPS. It also important to mention that the proposed enhanced FPS algorithm maintain the most important feature of Genetic Algorithm which is the ability of chromosomes even with low probability of Selection a chance to survive in the selection process but with elimination of the weak chromosomes damages for the reproduced generations.

References

[1] Acampora, G., Schiattarella, R., & Vitiello, A. (2021, July). Quantum genetic selection: Using a quantum computer to select individuals in genetic algorithms. In *Proceedings of the genetic and evolutionary computation conference companion*, 219-220. https://doi.org/10.1145/3449726.3459505

[2] Ahn, E. Y., Mullen, T., & Yen, J. (2010, July). Finding feature transformation functions using genetic algorithm. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, 2061-2062. https://doi.org/10.1145/1830761.1830862

[3] Al-Omari, M., & Al-Haija, Q.A. (2024). Towards Robust IDSs: An Integrated Approach of Hybrid Feature Selection and Machine Learning. *Journal of Internet Services and Information Security*, *14*(2), 47-67. https://doi.org/10.58346/JISIS.2024.I2.004

[4] Bahramvash, S., & Nikaeen, Z. (2016). Designing an Ethics Workbook for Fitness Trainers of the City of Tehran. *International Academic Journal of Business Management*, *3*(1), 118–123.

[5] Corus, D., Lissovoi, A., Oliveto, P. S., & Witt, C. (2021). On steady-state evolutionary algorithms and selective pressure: Why inverse rank-based allocation of reproductive trials is best. *ACM Transactions on Evolutionary Learning and Optimization*, *1*(1), 1-38. https://doi.org/10.1145/3427474

[6] Dimitrov, T., & Baumann, M. (2011, July). Genetic algorithm with genetic engineering technology for multi-objective dynamic job shop scheduling problems. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, 833-834. https://doi.org/10.1145/2001858.2002112

[7] Doan, D. B., Pham, M. T., & Dang, D. L. (2020, November). Predicting RNA secondary structure based on machine learning and genetic algorithm. In *Proceedings of the 4th International Conference on Future Networks and Distributed Systems*, 1-12. https://doi.org/10.1145/3440749.3442659

[8] Du, X., Ding, L. X., Xie, C. W., Xu, X., Wang, S. W., & Chen, L. (2009). Convergence analysis of gene expression programming based on maintaining elitist. In *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 823-826. https://doi.org/10.1145/1543834.1543952

[9] Flores-Fernandez, G. A., Jimenez-Carrion, M., Gutierrez, F., & Sanchez-Ancajima, R. A. (2024). Genetic Algorithm and LSTM Artificial Neural Network for Investment Portfolio Optimization. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *15*(2), 27-46. https://doi.org/10.58346/JOWUA.2024.I2.003

[10] Gliesch, A., Ritt, M., & Moreira, M. C. (2017, July). A genetic algorithm for fair land allocation. In *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 793-800). https://doi.org/10.1145/3071178.3071313

[11] Gupta, S. (2009, July). Relative fitness scaling for improving efficiency of proportionate selection in genetic algorithms. In *Proceedings of the 11th annual conference companion on genetic and evolutionary computation conference: late breaking papers*, 2741-2744. https://doi.org/10.1145/1570256.1570399

[12] Hart, E., Steyven, A., & Paechter, B. (2015, July). Improving survivability in environment-driven distributed evolutionary algorithms through explicit relative fitness and fitness proportionate communication. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 169-176. https://doi.org/10.1145/2739480.2754688

[13] Hasegawa, T., Mori, N., & Matsumoto, K. (2017, July). Genetic programming with multi-layered population structure. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 229-230). https://doi.org/10.1145/3067695.3076048

[14] He, Q., Ma, J., & Wang, S. (2010, October). Directly optimizing evaluation measures in learning to rank based on the clonal selection algorithm. In *Proceedings of the 19th ACM international*

conference on Information and knowledge management (pp. 1449-1452). https://doi.org/10.1145/1871437.1871644

[15] Helmuth, T., & Abdelhady, A. (2020, July). Benchmarking parent selection for program synthesis by genetic programming. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, 237-238. https://doi.org/10.1145/3377929.3389987

[16] Iqbal, S., & Hoque, M. T. (2016, July). A homologous gene replacement based genetic algorithm. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, 91-92. https://doi.org/10.1145/2908961.2909020

[17] Jiménez-Carrión, M., Flores-Fernandez, G. A., & Jiménez-Panta, A. B. (2023). Efficient Transit Network Design, Frequency Adjustment, and Fleet Calculation Using Genetic Algorithms. *Journal of Internet Services and Information Security*, *13*(4), 26-49. https://doi.org/10.58346/JISIS.2023.I4.003

[18] Kapoor, R., & Pillay, N. (2023, July). Iterative structure-based genetic programming for neural architecture search. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, 595-598. https://doi.org/10.1145/3583133.3590759

[19] Kawulok, J., & Kawulok, M. (2022, July). A genetic algorithm for classifying metagenomic data. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 59-60. https://doi.org/10.1145/3520304.3533944

[20] Liu, X., & Shang, L. (2014, July). Fitness proportionate selection based binary particle swarm optimization. In *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 11-12. https://doi.org/10.1145/2598394.2598417

[21] Maltare, N. N., Sharma, D., & Patel, S. (2023). An exploration and prediction of rainfall and groundwater level for the District of Banaskantha, Gujrat, India. *International Journal of Environmental Sciences*, *9*(1), 1-17.

[22] Marrero, A., Segredo, E., & Leon, C. (2021, July). A parallel genetic algorithm to speed up the resolution of the algorithm selection problem. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 1978-1981). https://doi.org/10.1145/3449726.3463160

[23] Motoki, T. (2002). Calculating the expected loss of diversity of selection schemes. *Evolutionary Computation*, *10*(4), 397-422. https://doi.org/10.1162/106365602760972776

[24] Nazmul, R., & Chetty, M. (2013, July). A priority based parental selection method for genetic algorithm. In *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*, 125-126. https://doi.org/10.1145/2464576.2464641

[25] Nuhu, A. R., Yan, X., Opoku, D., & Homaifar, A. (2021, July). A niching framework based on fitness proportionate sharing for multi-objective genetic algorithm (moga-fps). In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 191-192. https://doi.org/10.1145/3449726.3459566

[26] Oliveto, P. S., & Witt, C. (2012, July). On the analysis of the simple genetic algorithm. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, 1341-1348. https://doi.org/10.1145/2330163.2330349

[27] Onorato Jr, S., Amiruzzaman, M., Nor, R. M., Islam, M. R., & You, I. (2024). Machine Learning Side Effect Trend Predictions and the SIDER Database. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 15*(4), 90-108. https://doi.org/10.58346/JOWUA.2024.I4.007

[28] Pantridge, E., Helmuth, T., McPhee, N. F., & Spector, L. (2018, July). Specialization and elitism in lexicase and tournament selection. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 1914-1917. https://doi.org/10.1145/3205651.3208220

[29] Prakash, M., & Prakash, A. (2023). Cluster Head Selection and Secured Routing Using Glowworm Swarm Algorithm and Hybrid Security Algorithm for Over IoT-WSNs. *International Academic Journal of Innovative Research*, *10*(2), 01–09. https://doi.org/10.9756/IAJIR/V10I2/IAJIR1004

[30] Rowe, J. E. (2012). *Genetic algorithm theory*. In *GECCO '12: Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, 917–940. https://doi.org/10.1145/2330784.2330923
[31] Smith, J. (2007). On replacement strategies in steady-state evolutionary algorithms. *Evolutionary Computation (EVOL)*, *15*(1), 29–59. https://doi.org/10.1162/evco.2007.15.1.29
[32] Tahera, K., Ibrahim, R. N., & Lochert, P. B. (2007, July). Adopting dynamic operators in a genetic algorithm. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 1533-1533. https://doi.org/10.1145/1276958.1277262
[33] Tallón-Ballesteros, A. J., Li, T., & Fong, S. (2020, July). Quality enhancement of stochastic feature subset selection via genetic algorithms. Assessment in bioinformatics data sets. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, 55-56. https://doi.org/10.1145/3377929.3398168
[34] Tatsumi, T., Sato, H., & Takadama, K. (2017, July). Automatic adjustment of selection pressure based on range of reward in learning classifier system. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 505-512. https://doi.org/10.1145/3071178.3080531
[35] Tinós, R., Przewozniczek, M., Whitley, D., & Chicano, F. (2023, July). Genetic algorithm with linkage learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 981-989. https://doi.org/10.1145/3583131.3590349
[36] Torchinskii, V. E., Logunova, O. S., Sibileva, N. S., & Romanov, P. Y. (2018, April). Genetic algorithm modification: Addition of the population improvement stage. In *Proceedings of the 1st International Conference on Information Science and Systems*, 286-290. https://doi.org/10.1145/3209914.3209928
[37] Valem, L. P., & Pedronette, D. C. G. (2019, June). An unsupervised genetic algorithm framework for rank selection and fusion on image retrieval. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, 58-62. https://doi.org/10.1145/3323873.3325022
[38] Yu, Y. M., Zhao, G. Q., & Liu, J. D. (2009). Hyperchaotic genetic algorithm theory and functions optimization. In *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 1041-1044. https://doi.org/10.1145/1543834.1544003

Author Biography

Rawan Nassri Abulail is an associate professor in Philadelphia University and currently working as the head of computer science department in the information technology faculty. She received her B.Sc. degree in computer science and computer information systems from Philadelphia University, Amman, Jordan, in 2002, her B.Sc. degree in accounting from Philadelphia University, Amman, Jordan, in 2004, M.Sc. and Ph.D. degrees from The Arab Academy for Banking and Financial Sciences, Amman, Jordan, in 2004, and 2009 respectively, all in Computer Information Systems.