Exploring whether rank-based selection is better than elitist selection at defeating an enemy quickly

EvoMan Task 1: Specialist Agent - Group 90

Date: September 29, 2024

Aniruddha Bettedpur
a.s.bettedpur@student.vu.nl
2836885

Joella Galas
j.galas@student.vu.nl
2637004

Dora Medgyesy
d.medgyesy@student.vu.nl
2722256

Philip Vacca
p.p.p.vacca@student.vu.nl
2712798

1 INTRODUCTION

Genetic algorithms, a key subset of evolutionary computing, amounts to any search or optimization algorithm which uses Darwinian principles of natural selection [5].

These algorithms iteratively evolve a population of individuals over multiple generations, with the goal of identifying the fittest solutions within a given search space [2]. Central to the functioning of genetic algorithms is the selection mechanism. The selection mechanism decides which individuals are selected for the next population [4].

In this research, we focus on the EvoMan framework, which is a platform for testing competitive game-playing agents using evolutionary strategies. EvoMan is widely used to optimize the behaviour and actions of agents in a dynamic game environment, with one of its core challenges being the evolution of agents to efficiently defeat an enemy [1]. The success of this optimization task depends on the evolutionary algorithm (EA) employed. Several parameters, including the selection mechanism can influence the algorithm's performance and behaviour.

Two well-known selection mechanisms are rank-based selection and elitist selection. rank-based selection selects individuals based on their rank in the population, rather than their absolute fitness. This ensures a balanced probability of selection and significantly promotes diversity [3]. This method uses a more explorative search, which can lead to slower but more robust convergence, and avoids premature convergence by maintaining population diversity [2].

In contrast, elitist selection guarantees that the best-performing individuals from generation *t* are preserved in generation *t + 1*. This approach ensures faster convergence by preserving high-quality solutions. However, it can reduce diversity and risk premature convergence due to the fact that they focus mostly on the top individuals [3]. While elitism prevents the loss of the best solution, it can sometimes stagnate the evolutionary process by limiting exploration of the search space [2].

This research report will analyse the effectiveness of each of these selection mechanisms in terms of speed within the EvoMan framework. We will answer the following research question: "is rank-based selection faster at defeating enemies than elitist selection?". How *fast* an algorithm is at defeating enemies is defined as the time it takes the average individual from a population generated from the evolutionary algorithm to defeat an enemy. Since rank-based selection preserves population diversity and maintains balanced selection pressure, as opposed to elitist selection, which can overly invest in top individuals early on and therefore miss out on other possible solutions, it is hypothesized that rank-based selection defeats the enemy faster as compared to elitist selection.

This research aims to test this hypothesis by analyzing and comparing the performance of both selection mechanisms within the EvoMan framework.

2 METHODOLOGY

We created two EAs that differ in their selection methods. Both algorithms have hyperparameters, which are defined before training and remain constant throughout the runs. Our EA is based on a neural network which controls the player when fighting the enemies. The weights of this neural network are updated throughout the run of the algorithm. The number of hidden neurons and the upper and lower bounds for the weight values in the neural network are defined as static parameters. The algorithm starts by initialising a population of random neural network weights. We define the size of the population to be one hundred.

Each individual in the population is then evaluated based on the fitness function. Since we are trying to decrease the amount of time it takes to defeat the enemy, our fitness function is solely based on the runtime and whether the player has won. An individual will get a high fitness value if they defeat the enemy quickly, and player and enemy life are not taken into account. While our player is alive, the fitness functions is given by $fitness = \frac{1}{t+\epsilon}$ where time is denoted by t and a small value $\epsilon = 10^{-5}$ is added to the time to avoid division by zero errors. In this formula, the fitness value increases as time decreases. If our player dies then we define the fitness function as $fitness = -\frac{1}{t+\epsilon}$. We take a negative value here so the individuals that finish the game quickly by dying are punished. Then individuals are selected for crossover based on a tournament, where two random individuals are selected, their fitness values are compared and the better one is chosen to be the parent. The crossover method we use is whole arithmetic recombination, where the offspring genes are determined by a weighted sum of the two parents' genes based on a randomly generated proportion. Then these offspring undergo Gaussian mutation where a small random value, drawn from a normal distribution, is added to each gene of the offspring. The offspring undergo mutation with probability 0.1 in the elitist selection algorithm and 0.05 in the rank based selection algorithm. The fitness of these offspring are found and the offspring are combined with the existing population. The population is now ready for the selection of a new generation. This selection method differs in our two algorithms. One uses elitist selection and the other uses rank-based selection, both of these methods are discussed in detail below. If the population does not improve after fifteen generations, then the the worst quarter of individuals are replaced, in a scenario called "doomsday", either with random individuals or, with some probability, it copies the DNA from the best individual. This resulting final algorithm is then run for twenty generations and then the best individual is found.

2.1 EA1: elitist selection

We chose elitist selection for one of our algorithms as it ensures that the best performing individuals are always carried to the next generation, hence the highest performing individual is never lost. This way the algorithm can focus on improving around this strong solution. This prevents discarding good solutions, allowing for a faster convergence towards the optimal solution.

Our function for elitist selection defines that the top five percent of the population should be considered as elites and these individuals are carried directly to the next generation. The individuals are sorted in order based on their fitness so this top five percent can be selected. The rest of the population is then given a probability of being selected based on their fitness value, higher fitness means a higher chance of being selected. The remaining ninety five percent of the new population is chosen from these individuals.

2.2 EA2: Rank-based Selection

Rank-based selection ranks all individuals by their fitness value and assigns selection probabilities based on their rank, not their absolute fitness value. This ensures that high-performing individuals have a higher chance of being selected for the next generation. However, some weaker individuals are also occasionally get selected, which promotes diversity in the population and can help avoid getting stuck in local optima. This balance of exploration of new strategies and exploitation of known successful strategies leads to faster convergence to the optimal solution.

The function for rank-based selection sorts the individuals based on fitness value and assigns highest rank to fittest individual, second highest rank to second fittest individual and so on. It then assigns probabilities in such a way that the higher the rank of an individual, the higher probability that individual has of being selected for the next generation. One hundred individuals are then randomly selected with the associated probability and these individuals form the next generation.

To train our evolutionary algorithms we first optimised the hyper parameters for defeating enemy 5. Then using these hyper parameters we trained each algorithm on enemy 3, 5 and 8 ten times. This way we had sixty sets of training results. For each enemy first we calculated the average of the best fitness values of the ten training runs for each generation. We then also calculated the average of the mean fitness values of the ten training runs for each generation. This was done for both elitist selection and rank based selection. Figures (1), (2) and (3) show the results of these calculations. Furthermore, we run the best individuals of each algorithm 5 more times against each enemy, and compute the gain, to see how these fast individuals perform in a more general setting. Figure (4) shows the resulting boxplots.

We then performed a paired t-test on the fitness values from these results to test whether there is a significant difference between the fitness values of each algorithm, answering our research question. Then we did the same for the results of the gains to test for significance. The results of these tests are shown in Table (1)

3 RESULTS

3.1 Fitness

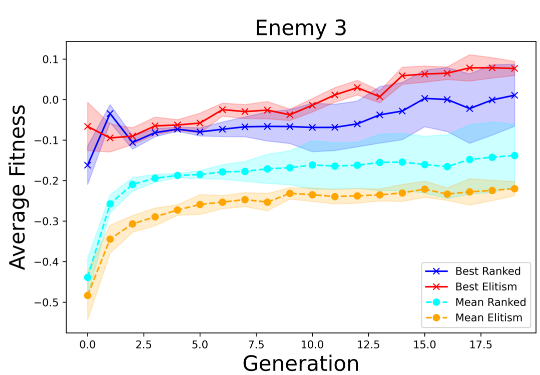

Figure 1: Average best and mean value of the ten runs with standard deviation

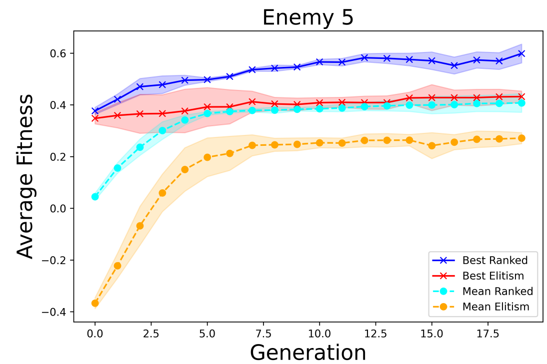

Figure 2: Average best and mean value of the ten runs with standard deviation

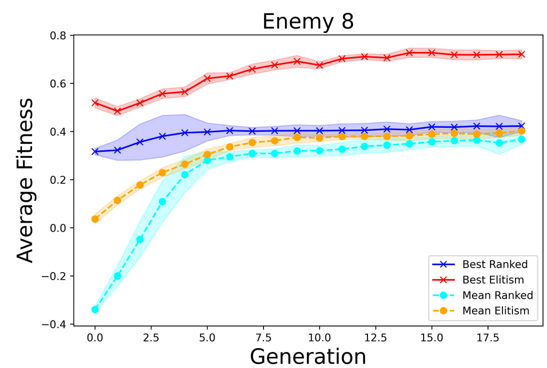

Figure 3: Average best and mean value of the ten runs with standard deviation

3.2 Gain

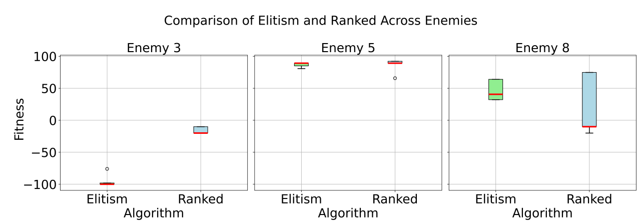

Figure 4: Difference in health (Gain) between the enemy and player

Enemy	p-value of best rank and elitism	p-value of mean rank and elitism	p-value of boxplots
3	0.00001	0	0.0055
5	0	0	0.099
8	0	0.00142	0.0081

Table 1: p-values

4 DISCUSSION

4.1 Fitness

Table (1) shows the results of the paired t-test when testing whether there is a significant difference between the fitness values for best rank-based and best elitism. Table (1) shows the results of the above test for the values of mean ranked and mean elitism. We test for a significance level of 0.05. Table (1) shows that $p < 0.05$ for each enemy. Therefore we can conclude that there is a significant difference between each pair of groups tested.

Figure (1) shows that the mean values of the rank-based selection algorithm performed better on average compared to the elitism algorithm. However, the average best run of elitist selection turned out to be superior to rank-based selection. An important observation is that the standard deviation of the fitness of all algorithms is much higher when fighting against enemy 3 than against any of the other enemies. This suggests that there is more randomness involved in the behavior of enemy 3, leading to an inconsistent performance between individuals.

From Figure (2) we can see that while the mean fitness values of rank-based selection still outperform elitist selection, there is a key difference in the performance of the best individuals against enemy 5. The average best value for the rank-based selection is now higher than that of elitist selection. Additionally, the fitness values for enemy 5 have a smaller standard deviation compared to enemy three, both for rank-based selection and elitist selection reflecting more stable performance.

Lastly Figure (3) shows that the average best individuals of the elitism algorithm perform better than those of the rank-based algorithm against enemy 8, while the mean values of both algorithms perform similarly. The standard deviation of the elitist selection algorithms lines were quite small, reflecting a steady training result for all generations for various different runs.

4.2 Gain

Figure (4) shows that the ranked-based algorithm, achieves the highest gain of the two algorithms against each enemy. Reflecting that in the best case scenario, it ends the game with a higher health score than the algorithm with elitist selection - suggestive of a better peak performance ability.

However, when comparing the median gains of the algorithm, there is no clear trend visible. Against enemy 3, the ranked-based algorithm has a significantly higher score Table (1). Against enemy 5, both algorithms achieve a similar score Table (1). And against enemy 8, the elitist algorithm has a significantly higher score Table (1). The lack of clear trend in the median gains suggests that the performance of each algorithm is highly context dependent. Each enemy behaves differently and each algorithm may shine in a different environment.

It is important to note that neither algorithm has been trained to keep their health in mind when fighting enemies. The fitness function used was based on time and created a strong selection pressure for individuals that defeated their enemy quickly, not for those that kept their health up. This explains why the gain of the individuals in Figure (4) does not correspond directly with the fitness of the individuals shown in Figures (1), (2) and (3).

4.3 Further Discussion

The best individuals from elitist selection consistently outperform those of rank-based selection in terms of fitness, likely because it is able to exploit individuals with high fitness quickly by maintaining the top performing individuals in each subsequent generation [3].

On average, elitism scores lower than rank-based selection. This can be explained by a lack of diversity within the elitist population which can reduce its ability to converge to the optimal solution in some runs, leading to a lower average fitness over multiple runs. The rank-based algorithm is better at preserving diversity and therefore is able to achieve a more consistent average performance [2].

This clearly demonstrates the trade off between the swift exploitation of the elitist algorithm and the steady exploration of the rank-based algorithm indicating that while elitist selection

4.4 Limitations

One limitation of this study is the low number of test runs in comparing the gains in Figure (4). Doing only five test runs could affect the reliability of the results. The variability in the performance of agents might not be fully taken into account. Consequently, the statistical power of the analysis is reduced, making the drawn conclusions less reliable. For a future study, a higher sample size could be used to avoid this.

Additionally, the fitness function applied solely focuses on the time it takes to defeat the enemy, without considering other factors e.g. health or the attack of a player. This could potentially skew the outcomes, as the EA only focuses on speed, rather than overall game performance. Future research could include a more complete fitness function.

5 CONCLUSION

This study indicates that rank-based selection is a better option for consistently achieving a lower time in defeating enemies than elitist selection. The findings support the hypothesis that rank-based selection, through its promotion of diversity, leads to more consistent outcomes compared to elitist selection. These results highlight the value of diversity in evolutionary algorithms for optimizing performance.

REFERENCES

[1] Karine da Silva Miras de Araújo and Fabrício Olivetti de França. “An electronic-game framework for evaluating coevolutionary algorithms”. In: *arXiv preprint arXiv:1604.00644* (2016).

[2] Lingaraj Haldurai, T Madhubala, and R Rajalakshmi. “A study on genetic algorithm and its applications”. In: *Int. J. Comput. Sci. Eng* 4.10 (2016), pp. 139–143.

[3] Zbigniew Michalewicz and Marc Schoenauer. “Evolutionary algorithms for constrained parameter optimization problems”. In: *Evolutionary computation* 4.1 (1996), pp. 1–32.

[4] Noraini Mohd Razali, John Geraghty, et al. “Genetic algorithm performance with different selection strategies in solving TSP”. In: *Proceedings of the world congress on engineering*. Vol. 2. 1. International Association of Engineers Hong Kong, China. 2011, pp. 1–6.

[5] Saneh Lata Yadav and Asha Sohal. “Comparative study of different selection techniques in genetic algorithm”. In: *International Journal of Engineering, Science and Mathematics* 6.3 (2017), pp. 174–180.