Diversity Preservation via Fitness Sharing in Evolution of EvoMan Agents

Vrije Universiteit Amsterdam

Arina Fayzulina (Student Number: 2803544)
Frank Le (Student Number: 2669711)
Joaquín Cardona Ruiz (Student Number: 2819713)
Mateusz Kedzia (Student Number: 2666752)

Course: Evolutionary Computing
Team 18: Four Robots
Assignment: Task 1: specialist agent
Date: 01.10.23

1 INTRODUCTION

Fitness Sharing, drawing inspiration from the principles of biological diversity, promotes diversity in the context of an Evolutionary Algorithm (EA) by having those individuals that are genotipically similar "share" their fitness, reducing it according to their number, in the same way that organisms in nature can exhaust niches if they become too numerous.[7] As a technique, it shows promise in its capacity of finding optimal solutions by leveraging increased diversity to prevent the population from converging to local optima. The central research question we have posed is: "Does the incorporation of fitness sharing influence the performance of EvoMan agents evolved through evolutionary algorithms?" We seek to examine this question because, in an environment like EvoMan, where perhaps performant strategies that are easy for an EA to converge to can be far from the optimal way of playing against a particular enemy, we expect the introduction of a diversity preservation mechanism like Fitness Sharing to lead to a higher level of success by encouraging the evolution of strategies that would otherwise be discarded as substandard at first, but can prove to be the optimal solution to the challenge once allowed to evolve. The experiments are conducted with two Evolutionary Algorithms ("models"), with only one of them utilizing Fitness Sharing. The metric of interest is the highest fitness values achieved by the evolved agents. The models are run against three enemies each, several times over. For each run, statistics are collected on individuals and their fitnesses across generations. We analyze the collected data and conduct two-way ANOVA tests to assess the impact of fitness sharing on maximum fitness.

2 ALGORITHMS DESCRIPTION

2.1 DEAP framework

We utilize the DEAP (Distributed Evolutionary Algorithms in Python)[8] to simplify the development of evolutionary algorithms, making use of its built-in functions for population management, genetic operators, and selection strategies.

2.2 Baseline Algorithm

Our individuals' genotype consists of an array of weights that get mapped into the neural network provided by the default player controller. Individuals are evaluated according to the fitness function given, expressed by:

$$f = 0.9(100 - e) + 0.1p - \log t$$

where p represents the player's current energy level, e denotes the enemy's current energy level, and t signifies the total number of timesteps required to conclude the fight [2]. We implement a Generational algorithm, in which the entire population gets replaced each generation by offspring produced through Two-Point Crossover, which was selected due to its ability to balance exploration and exploitation. The population first undergoes a parent selection to determine which individuals get to crossover. Those selected then produce a number of offspring larger than the population size, which undergo mutation using the gaussian mutation method, chosen for its frequent small changes to genes, and less frequent but possible large, radical changes that help with exploration of new solutions. A survivor selection stage is then performed, which establishes the population for the next generation. Both selection stages are performed using non-replacing tournament selection. Tournament selection contributes to a low selection pressure environment, allowing for a low fitness individuals to maintain an attainable chance of getting selected, which, along with implementing non-replacement to avoid identical individuals, encourage a high diversity of solutions in each population.

2.3 Diversity Preservation through Fitness Sharing

Fitness sharing is employed to maintain population diversity and explore various peaks in the search space. This approach helps uncover multiple optima and prevents premature convergence to suboptimal solutions, making it valuable for tackling multimodal optimization challenges [3]. The hamming distance function was chosen to compute distances between individuals, due to its applicability for the case of neural networks, in which other measures such as euclidean distance fail to assign high distance values to individuals with vastly different phenotypic characteristics. Once distances between all individuals have been computed, each individual's fitness score gets divided by the following term:

$$\sum_{i=1}^{N} 1 - \left(\frac{D_i}{R_f}\right)^{S_f}$$

Where N is the number of other individuals within a specified distance radius R_f, D_i represents the distance to individual i, and S_f is a factor that determines the strength (influence) of the Fitness Sharing. This reduced fitness value is used for the purpose of selection methods, but the actual evaluated fitness is also stored for data collection.

3 EXPERIMENTAL SETUP

In this section we detail our experiment setup, including the parameters chosen for our models as well as the characteristics of the evolutionary algorithms used, and of the experiments that were run. The parameters utilized were found through a dual annealing parameter tuning algorithm, detailed below. These parameters were applied identically to both models developed, so that the only discrepancy between them is the implementation of fitness sharing.

3.1 Parameter tuning

The model parameters were fine-tuned using the dual annealing global optimization method[9], which is a stochastic algorithm that is able to effectively explore a complex search space with numerous local minima. We aimed to maximize the fitness value, which was determined by how well the model fit the data and met the constraints. The optimization process minimized the negative of this value, resulting in a set of parameters that achieved high fitness.

The parameters listed in Table 1 were determined through the use of parameter tuning, whereas the parameters listed in Table 2

Table 1: Parameters determined through the use of Dual Annealing parameter tuning.

Parameter	Value
Hidden neurons	30
Population size	250
Parents Selected	126
Children per parent	4
Mutation probability	0.05
Mutation parameters (μ, σ, p)	(0,10,1)
Parent selection Tournament size	24
Survivor selection Tournament size	24

Table 2: Fixed Parameters at the team's discretion.

Parameter	Value
Type of Algorithm	Generational
Generations	50
Fitness sharing radius (model 2)	100
Fitness sharing strength (model 2)	1

remained fixed. Reasonable constraints were applied to the parameter tuning algorithm to keep values within acceptable bounds, both in terms of time (e.g., keeping the population size below a certain limit), and for algorithmic constraints (e.g. keeping the number of parents selected and tournament sizes be below the population size). To minimize resource expenditure on parameter tuning, the algorithm was run without fitness sharing enabled. The fitness sharing strength and radius were determined by subsequent experimentation, keeping the other parameters fixed.

3.2 Enemy Selection

Informed by the enemy behavior descriptions presented in the paper by Da Silva Miras De Araújo and De França (2016)[2] and guided by our own gameplay experiences, Enemies 2 (Airman), 5 (Metalman), and 7 (Bubbleman) were chosen for the evolutionary algorithms due to their distinct characteristics, in hopes of enabling the models' behaviors to be examined in diverse circumstances. These enemies prioritize effective jumping strategies over complex sideways movement, setting them apart from others. Enemy 7 in particular was selected to assess the algorithm's adaptability in diverse scenarios, given its unique environment and physics. Parameter tuning was run solely on Enemy 2, Airman, due to its simple behavior without any special features, as to provide a solid baseline that might produce parameters generalizable to all other enemies.

3.3 Experimental Design and Data Collection

The experiment involves two distinct Evolutionary Algorithms (models). The first one involves a fitness sharing stage, while the second one does not. Both models will be subjected to training and testing across three different enemy scenarios 2,5,7, each replicated 10 times. This results in a total of 60 observations. The objective is to gather data and perform a detailed analysis to address our research question.

We monitored and logged several critical metrics for each of the 10 independent runs for every enemy and algorithm combination. "Mean Fitness" tracks the average fitness value of the population across generations during each run. "Max Fitness" measures the highest fitness value achieved by any individual in the last population of each run. We also stored the weights for the individual with the highest fitness (Best Individual) in each run, so that they can be re-evaluated for further statistical power. Mean Fitness and Max Fitness data were aggregated by averaging the means and maximums of each generation for all 10 runs, which provides insights into the overall performance and progress of the algorithm. Additionally, the weights for all 60 Best Individuals were utilized to re-evaluate them 5 times each, this time measuring the individual gain of each evaluation, a measure of the player's ability to outperform the enemy, calculated as the difference between the player's life and the enemies' life.

4 RESULTS AND DISCUSSION

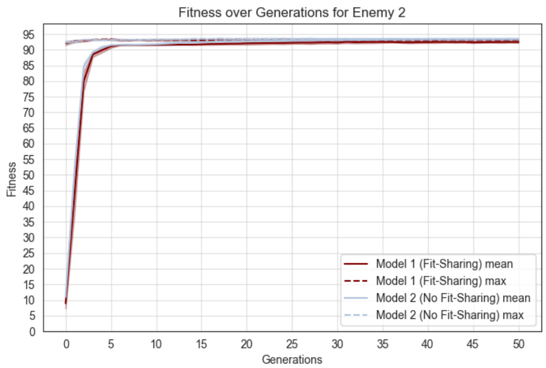
In this section, we use the aggregated figures discussed in the previous section to provide a visual representation of the experimental outcomes through line plots and boxplots, employ statistical tests to examine the significance of our findings and analyze the obtained results.

Figure 1(a) compares the mean and maximum fitness across generations for both models. Both measures are averaged across all 10 runs of the experiment, with shaded areas representing the standard deviation of the measures. Similarly, Figure 1(b) and Figure 1(c). show the same data for enemies 5 and 7 respectively. These visualizations make it clear that both models behave extremely similarly, specially towards latter generations, where they converge at almost identical fitnesses, with Model 2 tending to a slightly higher fitness than Model 1. The significance of this difference will be analyzed below.

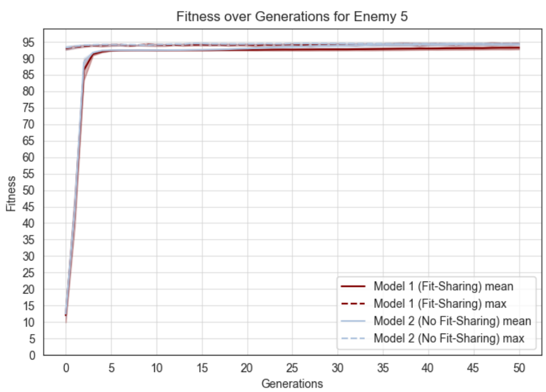
Figure 2 displays the gains of the re-evaluated Best Individuals, averaged across 5 evaluations each. Each box in the plot contains 10 data points, one per Best Individual for that model and enemy. Mean performance of the models against each other differ depending on the enemy. The statistical significance of this difference will be analyzed below. Figure 3 displays the diversity, measured as the mean Shannon Entropy of the population, across all runs in each generation. It clearly illustrates that model 1, which uses fitness sharing as a diversity preservation mechanism, consistently achieves much higher diversity levels, whereas model 2 tends to approach a point in which most individuals are almost identical. Shannon entropy is considered a superior measure of diversity according to existing research [6], as compared to the more common Mean Hamming Distance. It is notable, however, that we've found the latter to behave nearly identical to the former in our case.

4.1 ANOVA test

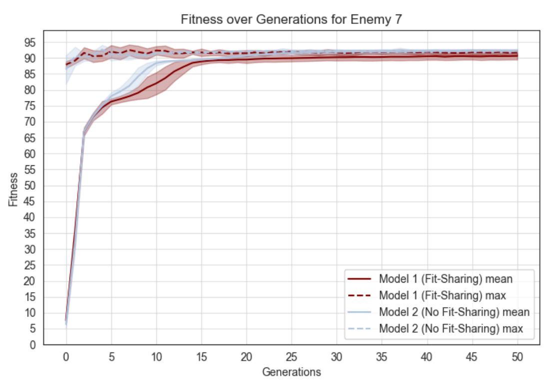
We use the two-way-ANOVA model to analyze the difference between our experimental groups. After determining that there's on interaction between independent variables (p-value of 0.1750), we can run the test on the averages of our data to determine that we can reject our null hypothesis,with a p-value of 0.0447 (< 0.05), having found a significant difference of -2.646667, shown in Figure



(a) Enemy 2



(b) Enemy 5



(c) Enemy 7

Figure 1: Mean and maximum fitness averages with standard deviations across generations for models with and without fitness sharing for different enemies.

between fitness sharing and no fitness sharing groups. Importantly, this difference is negative, suggesting a negative effect of fitness sharing on the averages. When performing similar analysis at the enemy level, it was found that the difference in mean gain for each model was only significant for enemy 2 (p-value = 0.0069).

To address the research question mentioned in the introduction, a similar test was conducted on the dataset considering maximum

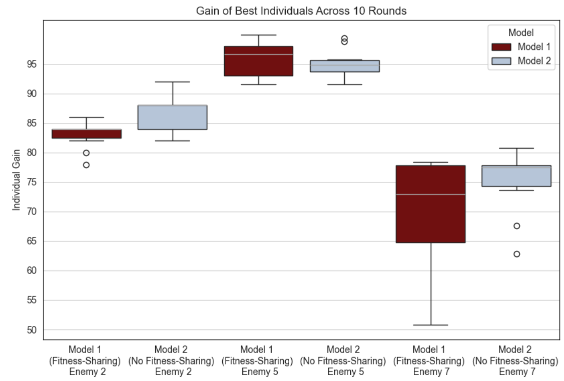


Figure 2: Box plots depicting gain distribution: 5 repetitions of best individual solutions per 10 independent runs; gains represent the difference between player's and enemies' life

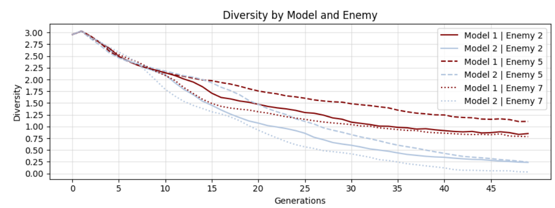


Figure 3: Diversity evolution over generations for different combinations of models and enemies

fitness of the individuals (max fitness). Once interaction between independent variables was ruled out (p-value 0.1849 > 0.05), the effect of the model on max fitness was tested. The p-value, 0.0529 for the fitness sharing variable is > 0.05, indicating that the null hypothesis (H0) cannot be rejected. This suggests that fitness sharing does not have a significant effect on the maximum fitness of individuals.

Our baseline paper[10] found that their fitness landscape was devoid of local optima, which is the reason the implementation of fitness sharing did not improve their results. In the case of our study, although our research question theorized that fitness sharing might impact our results, our findings proved the opposite, which might, with further research required, lead to the conclusion that this environment, too, lacks such local optima.

5 CONCLUSION

While fitness sharing proved to be effective in preserving diversity in the experiments performed, its application in the experiments run in this study did not significantly affect the performance (maximum fitness) of evolved individuals. This could be attributed to several potential reasons. The fitness landscape explored may lack numerous local optima, or the base algorithm may be sufficiently effective at avoiding them, making diversity preservation less essential. In addition, fitness sharing might prove more useful on a wider challenge, such as training individuals against multiple enemies at once. Future research could explore other problem spaces to find where diversity preservation techniques are most applicable.

6 REFERENCES

[1]Da Silva Miras De Araújo, K., De França, F. O. (2016). An electronic-game framework for evaluating coevolutionary algorithms. arXiv (Cornell University). https://arxiv.org/pdf/1604.00644.pdf

[2]Da Silva Miras De Araújo, K., De França, F. O. (2016). Evolving a generalized strategy for an action-platformer video game framework. https://doi.org/10.1109/cec.2016.7743938

[3]Sareni, B., Krähenbühl, L. (1998). Fitness sharing and niching methods revisited. IEEE Transactions on Evolutionary Computation, 2(3), 97–106. https://doi.org/10.1109/4235.735432

[4]Optimization (scipy.optimize) — SciPy v1.11.3 Manual. (n.d.). https://docs.scipy.org/doc/scipy/tutorial/optimize.htmlglobal-optimization

[5]Dr Gulab M. Nibrad (2019), Methodology and Application of Two-way ANOVA(https://www.ijmra.us/project

[6]High-Order Entropy-Based Population Diversity Measures in the Traveling Salesman Problem | Evolutionary Computation | MIT Press

[7] David A. Van Veldhuizen, Gary B. Lamont(2000),Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art

[8] DEAP documentation, https://deap.readthedocs.io/en/master/

[9] S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi(1983), Optimization by Simulated Annealing

[10] Santos, J., Monteagudo, Á. (2017). Inclusion of the fitness sharing technique in an evolutionary algorithm to analyze the fitness landscape of the genetic code adaptability. BMC Bioinformatics, 18(1). https://doi.org/10.1186/s12859-017-1608-x