# Evolutionary Computing, Task 2: Generalist Agent

Standard assignment group 51


Sander van Oostveen
2584302
s.van.oostveen@student.vu.nl

Nick van Santen
2628020
n.van.santen@student.vu.nl

Ryan Saeta
2739015
r.saeta@student.vu.nl

Rick van Slobbe
2601787
r.van.slobbe@student.vu.nl

## 1 INTRODUCTION

Evolutionary algorithms (EAs) are well suited for solving complex optimization problems. However, often these EAs are optimised for only a single given problem, and show worse general performance when compared to other, non-evolutionary, algorithms. One way for EAs to address this problem is through the use of Game environments, as these allow EAs to train agents to solve single problems (specialists), but also agents to solve multiple problems (generalists).

A generalist EA, like a specialist EA, is trained on a subset of the problem space, but with the aim of showing the best overall performance over the entire problem space. A key factor in the training of both types of EA is the diversity of the population, derived from the genetic differences between the individuals of a given generation. A typical measure of the diversity of a population is the pair-wise Hamming distance.

$$H = \sum_{i \in P} \sum_{j \in P} f(i, j) \tag{1}$$

Equation 1 shows the pair-wise Hamming Distance for the population of an EA. Here, $H$ is the distance, $P$ the population and is $f(i, j)$ the euclidean distance of the genotype of two individuals $i$ and $j$.

The diversity of the population is linked to the overall performance of EAs [4, 5]. Therefore, maintaining the diversity may be vital to the performance of generalist agents. One noted impact on the diversity is the fast spread of genes throughout the population due to the mating strategy. Employing different mating strategies could therefore improve the diversity.

While biological evolutionary processes are constraint by nature, these constrains do not hold for EAs. While in nature mating is typically conducted with only one or two parents, the mating strategy of EAs can involve even more parents (multi-parent). As the amount of parent's involved in the mating strategy increases, this would also increase the genetic distance of offspring from any one of their parents. Similarly, as the possible combinations increase, siblings would also have a larger genetic distance from each other. Therefore, multi-parent mating could be used to maintain the diversity of the population of an EA, and the training of a generalist agent.

The aim of this paper is to find what the impact is of a multi-parent crossover strategy on the training of a generalist agent as compared to a two parents crossover strategy. To answer this question, the impact of a four-parent as opposed to a two-parent crossover strategy on the mean and maximum fitness of the population, and the maximum gains of the generalist agents is investigated. Prior research [2] has shown that a multi-parent approach to crossover can enhance the efficiency of an EA solving a constrained optimization problem. However, it remains unclear how these different crossover strategies might affect evolutionary algorithms tasked with training generalist agents for optimization problems in a game environment.

As multi-parent crossover is assumed to lead to increased population diversity and prevention of premature convergence, it is expected that the 4-parent strategy for crossover will achieve higher mean and maximum fitness values and also a higher gain compared to the 2-parent strategy for crossover.

**Hypothesis 1:** It is expected that the mean and maximum fitness of the 4-parent EA increases at a similar rate but reach a higher level than the 2-parent EA.

**Hypothesis 2:** It is expected that the gain of the best performing generalist agents of the 4-parent EA is higher than the gain of the best performing generalist agents of the 2-parent EA.

## 2 METHOD

To test the hypotheses, the Distributed Evolutionary Algorithms in Python (DEAP) [3] framework is used to construct two EAs, one using two-parent crossover and the other using four-parent crossover. These EAs are tasked with the optimization of a neural network which needs to beat eight different enemies in the game environment framework 'evoman' [1]. Two subsets of enemies are used to train two groups of generalist agents for each EA. After training, the best performing agents (MVPs) of both training groups of both EAs are tested against all eight enemies 5 times. The fitness and gain of any individual agent from any given generation against a specific enemy is determined using equations 2 and 3 respectively.

$$\text{Fitness} = p - e - log(t) \tag{2}$$

$$\text{Gain} = p - e \tag{3}$$

Here, $p$ is the remaining life of the agent, $e$ is the remaining life of the specific enemy, and $t$ is the duration of the game.

The EAs are trained using two arbitrarily chosen training groups. Training group 1 includes the evoman enemies 1, 2, 4, and 8, while training group 2 includes enemies 3, 5, 6, and 7.

The weights and biases of the neural network are initialized randomly by drawing floating point values from a uniform distribution between -1 and 1. The initial population consists of 100 individuals, which are represented by arrays consisting of random floating point values drawn from the same uniform distribution. Both EAs run for a maximum of 30 generations. The mean and maximum fitness are calculated across all generations for both EAs and compared to show how the fitness values evolve for either EA.

After testing the population for fitness, the parents are selected through the tournament selection strategy. Five individuals are randomly selected and compared per round and the individual with the highest fitness is considered the winner, and selected to produce offspring with another winner.

Offspring is produced by combining genetic material from the parents in uniform crossover. This is where the difference in the two EAs resides. Recombination in the two parent crossover EA takes place through uniform crossover of two parents. For every allele a value is chosen from one of the two parents to create a new individual. Recombination in the four parent crossover also takes place through uniform crossover, but selects a value for an allele randomly from four parents instead of two.

The generated offspring have a 20% chance to be mutated. This so called mutation parameter was set based on initial testing. If selected for mutation, all values in an offspring's array have a 20% chance to have a value added to it from a Gaussian distribution.

Afterwards, the offspring is put through the game environment to determine their fitness. The newly evaluated population is then subjected to the same processes of selection, crossover and mutation for a total of 30 generations.

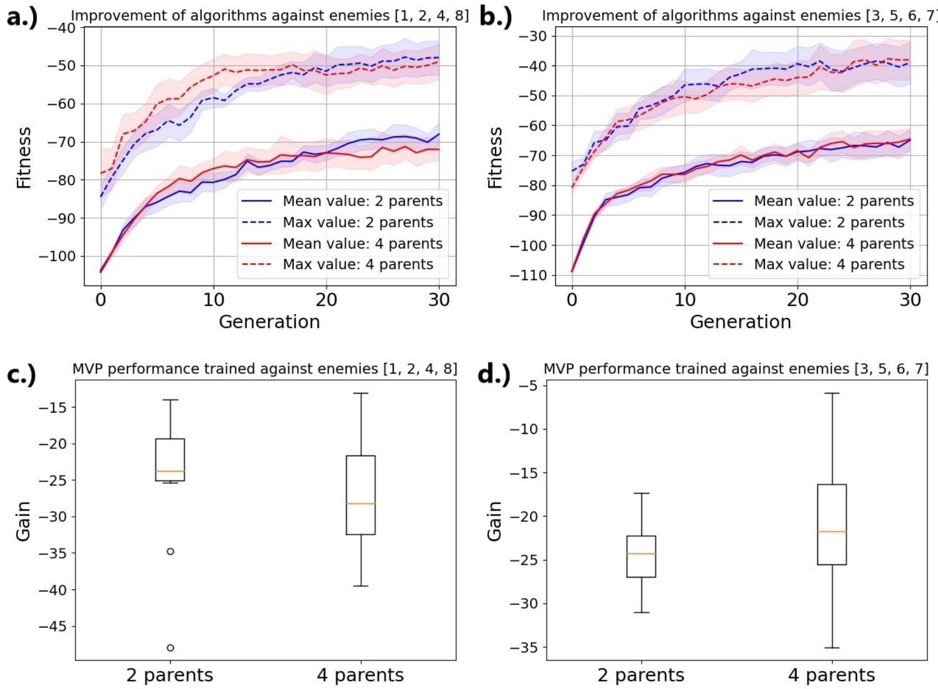


**Figure 1: a, b.) The development of the mean and maximum fitness across 30 generations for the two-parent and four-parent crossover EAs against the subset of enemies 1, 2, 4, and 8, and against enemies 3, 5, 6, and 7 respectively.**
**c, d.) Box-plots of the mean gain of the best solution over 5 runs for the two-parent and four-parent crossover EAs against all enemies, trained against enemies 1, 2, 4, and 8, and enemies 3, 5, 6, and 7 respectively.**

To test the hypothesis, the two- and four parent EAs will be run 10 times against the two subsets of enemies. The mean and maximum fitness values of each generation will be visualised in a line plot. For both EAs, the best performing individual (MVP) of each of the 10 runs is selected based on gain, and tested against every enemy 5 times. The results of these 5 tests is presented for each EA and each training group in box plots.

## 3 RESULTS

The two- and four-parent EAs are trained and compared for two training groups of enemies. Firstly, figures 1.a and 1.b show the mean and maximum fitness values over the generations of the two EAs for training groups 1 and 2 respectively.

In the case of training group 1, as shown in figure 1.a, the four-parent EA has an initially higher mean and maximum fitness, although this eventually stabilises and gets overtaken by the mean and maximum fitness of the two-parent EA. Nevertheless, by the final generation, both EA's shown similar mean and maximum fitness values, reaching a mean fitness around -70 and a maximum fitness around -50.

This similarity is also observed in the results for training group 2, as shown in figure 1.b. Besides occasional fluctuations, both the mean and maximum fitness are nearly the same for both the two- and the four-parent EAs. However, it can be noted that the mean and maximum fitness of both EAs is slightly higher for training 2 as compared to training group 1, with a mean fitness around -65 and a maximum fitness around -40 for both EAs.

Generally, it is found that beyond some initial differences, both EAs showed similar mean and and maximum fitness values to each other. It can be noted that both EAs showed higher overall mean and fitness values for training group 2 than for training group 1.

Figures 1.c and 1.d show the mean gains of the best individuals from training group 1 and training group 2 respectively. Each box represents the mean gains of the MVPs selected from each of the ten independent runs of each EA, over five runs against all eight enemies.

The MVPs from training group 1, as shown in figure 1.c, appear to have a similar gains for both EAs. However, it can be noted that the MVPs from the two-parent EA on average have a higher gain than the MVPs of the 4-parent EA. For the MVPs from training group 2, as shown in figure 1.d, it is found that both EAs on average have similar gains. Instead, the best MVPs of the four-parent EA achieved far higher gains as compared to the best MVPs of the two-parent EA, with a difference in gain of about 10 between the best MVPs of both EAs. In general, the MVPs from both training groups performed similarly, with average gains between the -20 and -30. The worst average gains are achieved by the MVPs of the four-parent EA for training set 1. However, the best average and overall best gains of an MVP are achieved by the four-parent EA for training set 2.

| Training Group | P-value |
|---|---|
| 1 | 0.54 |
| 2 | 0.31 |

**Table 1: The p-value significance of the gains of the two EAs for each subsets of enemies, based on the two-tailed t-test. Values are significant when P < 0.05**

Table 1 shows the significance of the difference in the sample distributions of the two-parent and four-parent crossover EAs for each of the two training groups. In both cases, the found p-values are larger than the $\alpha$-value of 0.05.

Lastly, table 2 shows the player and enemy life of the overall best MVP against enemies 1 to 8. Against enemies 1, 4, 6 and 7, the MVP lost all it's lives against the enemy. On the other hand, the MVP was able to win against enemies 2 and 5. While not loosing, the MVP timed out against enemies 3 and 8. The overall best MVP originated from the four-parent EA and was trained against training group 2.

| Enemy | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Player Life | 0.0 | 76.0 | 8.8 | 0.0 | 61.1 | 0.0 | 0.0 | 8.6 |
| Enemy Life | 74.0 | 0.0 | 24.0 | 86.0 | 0.0 | 30.0 | 28.0 | 24.0 |

**Table 2: The performance of the best MVP. For each enemy the average player and enemy life over 5 runs were calculated.**

## 4 DISCUSSION

Based on these results, it is found that there is no significant difference in the performance of generalist agents trained from either the two-parent crossover or four-parent crossover EAs. For the generalist MVPs trained from either training group, the found p-values in table 1 bewteen the two EAs could not reject the null-hypothesis that their samples are derived from the same distribution. As such, it cannot be conclude that four-parent crossover has a significant impact on the performance of a specialist agent as compared to an EA using two-parent crossover.

The lack of significant differences in the gains of the MVPs between both EAs may be caused by either the tournament size of 10% used in this research. As the tournament size increases, the probability that over the generations siblings are selected can increase. As more siblings are selected to mate, the diversity could decrease, regardless of the amount of parents involved, thus causing the EAs to become stuck in local optima.

Despite being unable to find a significant difference between the two EAs, it must be noted that the overall best generalist MVP was achieved from the four parent crossover EA. In general, the MVPs of the four-parent EA from training group 2 seem to score higher gains than the MVPs of the same EA from training group 1. This could imply that the selection of enemies for the training group might have a larger impact on the gains of the overall best MVP, instead of the crossover method, and by extension, the diversity of the EA. This is not to say that the crossover method does not have an impact, as the two-parent crossover based EA does not show similar differences in the MVP performance as the four-parent EA.

Upon inspecting the player and enemy life of the best MVP for each enemy in Table 2, the results can be separated into three categories: Player losses, Player wins, and Player timeouts. It can be found that the MVP loses against enemies 1, 4, 6 and 7. Of these, MVP loses particularly bad against enemies 1 and 4, with average gains of -74 and -86 respectively as opposed to average gains of -30 and -28 for enemies 6 and 7 respectively. The MVP wins against enemies 2 and 5, with average gains of 76 and 61.1 respectively. Lastly, the MVP times out against enemies 3 and 8, with average gains of -15,2 and -15.4 respectively.

While the average gains of both the MVP wins and timeouts are relatively similar for both training groups, the average gain of the MVP losses are very different for the two training groups. Its gains against enemies 6 and 7 against which the MVP was trained are greater as compared to its gains against enemies 1 and 2. As the best MVPs of the four-parent crossover EA trained against training group 1 do not show a similar average gains, this could indicate that the selection of enemies in each training set had the largest impact on the overall performance of the generalist MVP.

## 5 CONCLUSION

In this paper, the impact of two- and four-parent crossover on the performance of a generalist agent of an evolutionary algorithm are tested. Based on the gain of several best performing generalist agents, it can not be concluded that the four-parent evolutionary algorithm outperformed the two-parent crossover based evolutionary algorithm.

In regards to hypothesis 1, it is found that the mean and maximum fitness of the four-parent crossover based EA rises at a similar rate to the two-parent EA, and both reach a similar level of fitness.

In regards to hypothesis 2, it is found that there is no significant difference in the gain of the best four-parent generalist agents as compared to the gain of the best two-parent generalist agents.

In further research, the impact of tournament size in relation to the diversity of the population of an EA could be investigated. In addition, research could be conducted on the impact of incest for larger numbers of parents involved in multi-parent crossover. By adressing these two factors, multi-parent crossover may still be effectively used to generate better performing generalist agents as compared to two-parent crossover. Lastly, further research could be conducted on the impact of training group selection. Through proper understanding of the impact training groups have on the performance of generalist agents, better training groups may be selected, allowing for better performing agents.

## 6 CONTRIBUTIONS

For this paper, Sander and Rick mostly collaborated to write the document. The figures and analysis where conducted by Nick, based on the results derived from the algorithms implemented primarily by Ryan. In general, each research member equally contributed to the discussions and progress regarding the paper over the course of the project.

## REFERENCES

[1] Karine da Silva Miras de Araújo and Fabrício Olivetti de França. 2016. An electronic-game framework for evaluating coevolutionary algorithms. *arXiv preprint arXiv:1604.00644* (2016).
[2] Saber M Elsayed, Ruhul A Sarker, and Daryl L Essam. 2011. GA with a new multi-parent crossover for constrained optimization. In *2011 IEEE Congress of Evolutionary Computation (CEC)*. IEEE, 857–864.
[3] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. 2012. DEAP: Evolutionary Algorithms Made Easy. *Journal of Machine Learning Research* 13 (jul 2012), 2171–2175.
[4] Andrea Toffolo and Ernesto Benini. 2003. Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evolutionary computation* 11, 2 (2003), 151–167.
[5] Rasmus K Ursem. 2002. Diversity-guided evolutionary algorithms. In *International Conference on Parallel Problem Solving from Nature*. Springer, 462–471.