Evolutionary Computing, Task 1: Specialized 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

October 1, 2021

.

1 INTRODUCTION

Evolutionary algorithms (EAs) offer a means of solving complex real world optimization problems beyond normal machine learning (ML) algorithms. A popular method to test the efficacy of EAs is game environments. As an EA-controlled agent plays in these games, it chooses from a consistent set of actions. The agent's fitness is calculated based on the final game state.

However, even in game environments, ML algorithms can have a tendency to become trapped in local optima. This can be remedied in EAs through the mechanism of mutation. There are several different strategies through which mutation can be applied to floating-point genes, such as those based on probability distributions. During mutation, a random value from a given distribution may be added to an element of the offspring. The probability of a particular element mutating, and by extension the ability of the EA to avoid local optima, is dependent on the distribution from which the mutation is taken. While the Gaussian distribution, as shown in equation 1, is often used for this purpose, other distributions such as the Lévy distribution are likely to impact the magnitude mutation, due to the difference in the length of their tails.

$$G_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{1}$$

Specifically, a Lévy distribution, as shown in equation 2 has an infinite second moment—meaning an infinite variance—and would more likely produce offspring that are farther away than using a distribution with finite variance.

$$L_{\alpha,\gamma}(y) = \frac{1}{\pi} \int_0^\infty e^{-\gamma q^\alpha} cos(qy)dq \tag{2}$$

This report aims to answer if the Lévy-based mutation strategy is better at solving the problems in a game environment than the Gaussian- based mutation strategy. To answer this question, the impact of the Lévy- and Gaussian- based mutations on the mean and maximum fitness, and gain across generations of an evolutionary algorithm is investigated. Prior research [3] indicates that the Lévy-based mutation outperforms Gaussian-based mutation for problems with many local optima. However, it is not yet understood how these distributions might impact the mean and maximum fitness in game environments. It is expected that the Lévy-based mutation will result in higher mean fitness values compared to Gaussian-based mutation, as more local optima will be avoided. However, as Lévy would result in a higher magnitude of mutation, either positively or negatively, it is expect that the maximum fitness will fluctuate more for the Lévy-based mutation.

Hypothesis 1: It is expected that the mean fitness of the Lévy distribution to increase less rapidly but eventually reach a higher level than the Gaussian distribution.

Hypothesis 2: It is expected that the max fitness to fluctuate more using the Lévy distribution compared to the Gaussian distribution.

Hypothesis 3: It is expected that the gain of the most fit individuals to be larger for the Lévy distribution as compared to the Gaussian distribution

2 METHOD

To test these hypotheses, an experiment is set up using the Distributed Evolutionary Algorithms in Python [2] (DEAP) framework. Using DEAP, a neural network is optimised against three selected enemies in the game environment framework "Evoman" developed by Miras [1]. The goal of the neural network is to beat an enemy one at a time by shooting it. In this experiment, enemies two, five and eight of the evoman framework are used. These three enemies are selected based on preliminary results showing that they can be beaten by the neural network. The evolutionary algorithms are set with an initial population size of 50, and 20 generations are ran during the experiment.

DEAP generates the first population randomly by creating arrays with random floating point values from a uniform distribution between -1 and 1. The set weights and biases are also initialised randomly. To determine the performance of an individual, the fitness and gain are calculated with the following formulas:

$$\text{fitness} = \gamma * (100 - e) + \alpha * p - \log(t) \tag{3}$$

$$\text{Gain} = p - e \tag{4}$$

Where γ is defined as 0.99, α as 0.01, and t as the duration of the game. e and p are the enemy and the player energy levels at the end of the game, respectively. The higher the returned fitness value, the better the individual performed. Likewise, a higher gain indicates better performance.

Offspring is generated based on the parents using the tournament selection strategy. This strategy takes a third of parents randomly from the parent population and selects the individual with the highest fitness as the winner to generate offspring with another winner.

The genetic material from the parents is combined by the two-point crossover method for generating offspring. In this method two 'cuts' are made in the array of both parents, and the sequence between these two cuts are swapped in the new generation.

Mutation is then introduced by adding random values in the arrays of the offspring with values from either the Gaussian or Lévy distribution.

This results in two different EAs: One drawing values exclusively from a Gaussian distribution, and one from the Lévy distribution. During the experiment the mutation parameter is set to 0.1, giving individuals a 10% chance of being selected for mutation. If selected, each of the 265 elements of an individual has a 10% chance of mutation. That means if an individual is selected, we expect about 26 elements will be mutated.

The newly generated offspring forming the next generation are subjected to the game and evaluated using the fitness function. This process of selection, crossover and mutation then continues for a set number of generations—in this case 20.

To answer the first two hypothesis, we use the Gaussian- and Lévy- distributed EAs to perform 10 runs each against each of the three enemies. Each run sees 20 generations of a given EA trying to beat the given enemy. For each generation, the mean and maximum fitness values are determined. These values are consolidated into a line graph for each EA and per enemy.

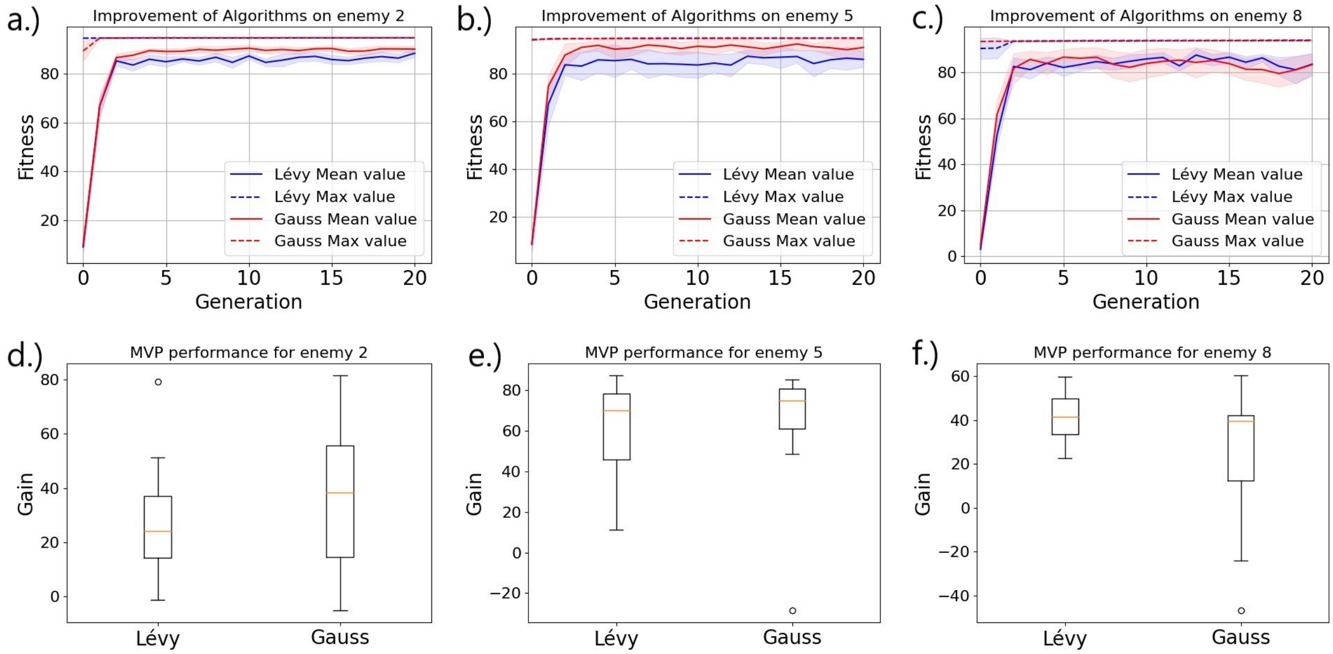

Figure 1: a,b,c) The development of the mean and maximum fitness across 20 generations for the Lévy and Gaussian EAs against enemies 2, 5, and 8 respectively.
d,e,f) Box-plots of the mean gain of the best solution over 5 runs for the Lévy and Gaussian EAs against enemies 2, 5, and 8 respectively.

To answer the third hypothesis, the best performing individual (MVP[1]) of each run is selected based on their gain. These MVPs are tested 5 times against each enemy. The mean values of these 5 tests are determined for all MVPs, and collected into a box graph of the two EAs for each enemy.

3 RESULTS

The Gaussian and Lévy distributed EAs are compared for the three enemies. Figures 1.a, 1.b and 1.c show the mean and maximum fitness values of the two EAs against enemies two, five and eight respectively.

We find for enemies two and five that the mean fitness increases at seemingly the same rate over the first couple of generation, until they both reach a stable point, with the mean of the Gaussian distribution clearly being larger than the mean fitness of the Lévy distribution. For the maximum fitness values, we find that both reach the same consistent maximum value, with the Gaussian distribution performing slightly worse only in the first generation for enemy two.

In the case of enemy 8, the rise in the mean fitness during the early generations is less consistent between the two distributions. However, both distributions appear to reach a plateau at similar mean fitness value, although periodically overtaking each other. Again, the max fitness values of both distribution seem to remain at the same point, although early on in the first two generations, the maximum of the Lévy distribution remains slightly below the maximum of the Gaussian distribution.

Generally, both EAs achieved similar mean fitness values in relatively the same amount of generations, with the Gaussian distribution outperforming the Lévy distribution in two out of three cases, and have a similar performance in the other case. On the other hand, both EAs achieved the same maximum fitness values within the first three generations.

Figures 1.d, 1.e and 1.f illustrate the gain of the best performing MVPs for both the Gaussian and Lévy distributions against enemies two, five and eight respectively. Each box represents the performance of the MVPs of ten independent runs when tested against the given enemy five times.

For enemy two, figure 1.d shows that both EAs have similar gains, although the Gaussian distribution appears to slightly outperform the Lévy distribution. This difference in the gains of each EA is even smaller against enemy 5 as shown in figure 1.e. Here, both EAs have a very similar mean and max gain, although the Lévy distribution is more spread downwards than the Gaussian distribution. However, against enemy eight, as shown in figure 1.f, the Lévy distribution has a better gain, while the Gaussian distribution is more spread downwards, even reaching negative gains.

Table 1 shows the mean and standard deviation of each EA in each of the boxplots shown in figures 1.d, 1.e and 1.f. From the table, it is found that for each enemy, both EAs perform relatively well, although in all cases the standard deviation is very high. The Gaussian achieved the highest gain against enemy 5, While the

[1]Most Valuable Player

	Enemy 2		Enemy 5		Enemy 8	
	Lévy	Gauss.	Lévy	Gauss.	Lévy	Gauss.
Mean	28.76	36.96	59.83	62.90	41.67	22.70
std.	22.32	27.30	24.87	32.44	11.55	32.63

Table 1: The mean gain and standard deviation of the best solutions from the ten independent runs per EA and enemy.

Lévy distribution achieved the highest gain and smallest standard deviation against enemy 8.

The significance of these findings is determined using the two-tailed t-test for the gains against each enemy, as shown in Table 2. In all cases, the p-values could not reject that these results were due to chance, and as such are not significant.

Enemy	P value
2	0.494
5	0.824
8	0.117

Table 2: The p-value significance of the gains of the two EAs for each enemy, based on the two-tailed t-test. Values are significant when $P < 0.05$

4 DISCUSSION

The results show a slight difference in mean and maximum achieved fitness between the Lévy- and Gaussian-based mutation EAs against all three enemies. On average, the Gaussian distribution has a slightly better mean fitness as compared to the Lévy distribution against two of the three enemies. This could be because both EAs reach the same maximum fitness fairly early on. As the EAs have reached this upper bound, most mutation at this point would decrease the fitness of the next generations. As the Lévy distributed EA would be more prone to mutation, this would decrease the mean fitness as compared to the less mutation prone Gaussian mutation.

Statistical analysis shows no significant difference in the gain between both EAs against each enemy. This suggests no difference in the avoiding of local optima when using Lévy- and Gaussian-based mutation strategies. Prior research found that Lévy-based mutation strategies performed better than Gaussian based strategies in solving problems with many local optima [3]. As such, those findings might not hold in the case of game environments, as the impact of local optima on the performance of either distribution might be lessened.

On the other hand, the relatively small difference in learning rate of both EAs may be attributed to the small size of the individuals. Each individual consisted of only 265 floating-point elements with unbounded values. Given that the Gaussian- and Lévy-based distributions most often only add values between -1 and 1, there may not have been a significant enough difference between the element value before and after mutation.

5 CONCLUSION

In this paper, the Lévy distribution based mutation strategy is tested against the more commonly used Gaussian distribution based mutation strategy in the Evoman game environment for evolutionary algorithms. From the results of the two-tailed T-Tests on the gains earned between the algorithms it was found that both evolutionary algorithms have no conclusive difference in performance given the parameters of the mutations that were used.

In regards to hypothesis 1, the mean fitness of the Lévy distributed mutation strategy was in general similar or slightly worse than the mean fitness of Gaussian distributed mutation strategy. Answering hypothesis 2, it was found found that there was no difference between the maximum fitness values for either algorithm. Both algorithms resulted in stable maximum fitness values. Lastly, with regard to hypothesis 3, there was no significant difference between the gain of either the Lévy or Gaussian distributed mutation strategies.

For further experiments, it is suggested to increase the tails of the Lévy distribution. This would in general increase the magnitude of the mutation, resulting in much more different individuals. Similarly, it is suggested to bound the elements of individuals in such a way that the magnitude of the mutation is larger compared to the parents' element values. Lastly, increasing the size of the individuals' genome so that more elements might be selected for mutation might be interesting for future experiments.

6 CONTRIBUTIONS

For this report Sander and Rick focused mostly on writing the report. Nick ran a lot of the experiments and wrote code to design the plots. Ryan wrote the code to implement the evolutionary algorithms and helped with the report. Everyone contributed equally to brainstorming for the research question, hypotheses and which parameters to manipulate.

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] 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.

[3] Chang-Yong Lee and Xin Yao. 2004. Evolutionary programming using mutations based on the Lévy probability distribution. *IEEE Transactions on Evolutionary Computation* 8, 1 (2004), 1–13.