SET10107 Computational Intelligence Coursework

40201757

ABSTRACT

The aim of this report is to investigate and analyse different approaches for evolving weights of an artificial neural network (ANN). The first section looks at what Evolutionary Algorithms (EAs) are and investigates the different EA operators. The second part of the report performs statically rigorous analysis on the performed experiments and the gathered data. It uses mean, median, standard deviation and t-tests to justify why certain operators and parameters are better than others. The last section of the report, summarises the optimal operators and parameters that were found and their mean scores on both test and training datasets.

KEYWORDS

Evolutionary Algorithms, Neural Networks, Spacecrafts simulation, Tournament selection, 1-point crossover, Tournament replacement, Gaussian distribution, Computational Intelligence, Java

1 INTRODUCTION

The results in the report are based on a spaceship simulation that has three binary actions - turn on/off left, right and/or main engines. The goal is to land a fleet of spaceships as smoothly as possible and with as little fuel as possible. The project aims to achieve this by creating and calculating the fitness scores of ANNs using no more than 20,000 evaluations. Every ANN is scored using 8 spaceships and its fitness score is equal to the average of the 8 spaceships.

2 APPROACH

This section of the report explains the background reading that was performed as part of the project. It investigates different Evolutionary Algorithm operators and explains the genetic algorithm that was used. The last part of this section describes the neural network parameters that were explored as part of the experiments.

2.1 Background reading

Background reading was required prior starting work on the project. This helped for the thorough understanding of how Evolutionary Algorithms (EAs) and the different operators work. The main literature sources include scientific papers found on Google Scholar and the book on Evolutionary Computing by [Eiben and Smith].

As can be seen in figure 1 below, EAs are a continuous loop of selecting parents, combining their chromosomes, mutating and creating offspring that is added to the population. Similarly to other machine learning algorithms, EAs require balancing between exploration and exploitation in order to find an optimal or a nearly optimal solution to a problem [8]. This appears to be achieved with different EA operators that are further analysed in the next section. It is also important to note that every new member of the population is scored using a fitness function that evaluates how "fit" a member is towards achieving a predefined goal. This project aims at solving a minimisation problem and because of this, the lower the fitness score of a member is, the better it is.

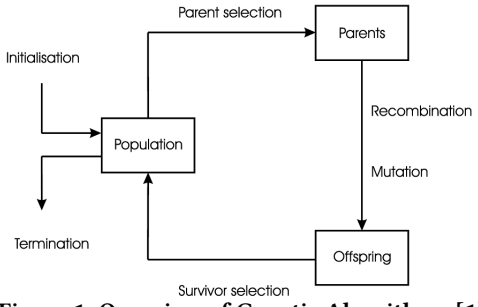

Figure 1: Overview of Genetic Algorithms [1]

2.2 Investigated EA operators

2.2.1 Selection. The selection operator identifies the members of the population that will be used to create a new individual. For the purposes of this project, the following three approaches were investigated:

- *Uniform selection* - A random member of the population is selected regardless of its fitness score. In this way, there is no selection pressure and every member of the population has an equal chance of being selected [1].
- *Roulette selection* - According to [Eiben and Smith], roulette selection (also called fitness proportionate selection) uses the same principle as spinning a roulette wheel, where the size of the segments reflects the selection probabilities. As can be seen in figure 2 below, the individuals' portion on the wheel is dependant on its fitness. As a result, more optimal members have higher chances of being selected for reproducing.

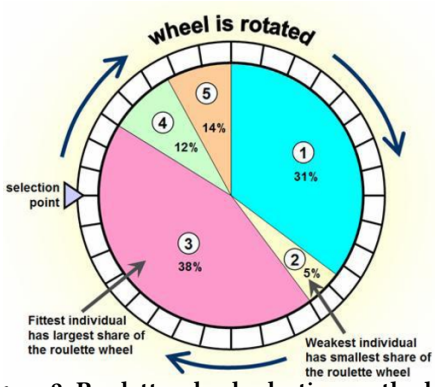

Figure 2: Roulette wheel selection method [2]

- *Tournament selection* - [Eiben and Smith] describes tournament selection as an operator that picks an individual from

a subset of the population based on its fitness score. The size of the subset is called tournament size N and it defines the selection pressure. The closer N is to the population size, the higher the selection pressure is. Figure 3 below presents a high level overview of the tournament selection operator with $N = 3$.

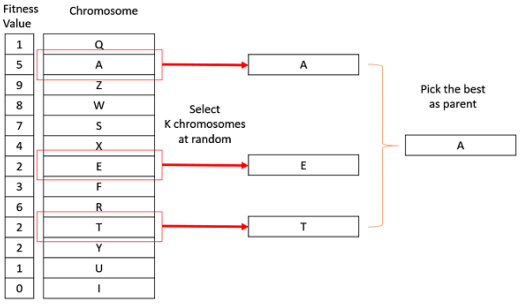

Figure 3: Tournament selection method [7]

2.2.2 Crossover. The crossover operator uses the chromosomes of the selected members and combines them to produce offspring. This way, the EA exploits good members of the population to find more optimal solutions. For the purposes of this project the following four crossover operators were investigated:

- *Exact copy* - The offspring has the exact chromosome as the parent. If this operator is used, the only way to introduce diversity in the population is through mutation.
- *One-Point* - As [Eiben and Smith] describes it, 1-point crossover chooses a random number between 1 and the length of the chromosome. Then it splits the two parents' chromosomes at this point and swaps them to create the offspring.
- *N-Point* - [Eiben and Smith] further analyse that N-Point crossover is a generalised approach to 1-point crossover. The only difference is that instead of splitting the chromosome at one point, it is split at N points.
- *Uniform Crossover* - The last crossover operator that was investigated is uniform crossover. According to [Eiben and Smith], this operator treats each gene independently and has an equal chance of being inherited.

Please refer to figure 4 below for further details on the 1-point, 2-point (N-point) and uniform crossover operators.

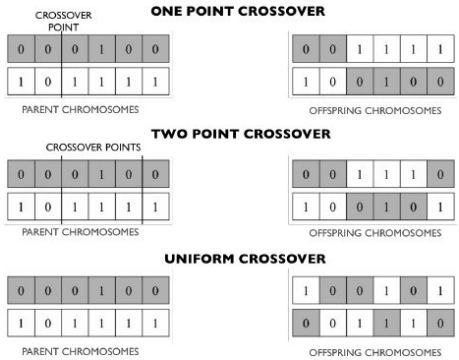

Figure 4: Crossover methods [4]

2.2.3 Mutate. The mutation operator is the part of the EA that introduces diversity in the population and allows exploration of the search space. This operator can potentially help the EA escape local optimas and find the global optima. For the purposes of the project, the following two mutate approaches were investigated:

- *Uniform mutation* - Mutates the genes using a constant mutate change and a constant mutate rate.
- *Nonuniform mutation* - Add or subtract a mutate change based on a Gaussian distribution [1].

2.2.4 Replacement. The last investigated operator is replacement. For the purposes of this project the following four approaches were analysed:

- *Replace the worst* of the population regardless of its fitness score.
- *Replace the worst* of the population, if and only if, it is a more optimal solution.
- *Tournament replacement* - Similarly to the tournament selection it uses a subset of the population and it replaces a member, if and only if, its replacement is a more optimal solution [5].
- *Tournament replacement ensuring diversity* - It is similar to tournament replacement, however, in this case an individual is added to the population if it has a lower fitness value by a predefined threshold and there are no other individuals with the same genes. This approach was partially inspired by the analysis performed by [Gupta and Ghafir] on "maintaining diversity in Genetic Algorithms".

2.3 Genetic algorithm

The used algorithm is a steady-state genetic algorithm. As [Eiben and Smith] describes the steady-state algorithm replaces only a small part of the population, contrary to Elitism that replaces the majority of the population.

2.4 Neural network parameters

The neural network used within the project has one hidden layer and the set of parameters described below. The successful completion of the project required exploration and tuning of their values.

- *Mutate rate* - It defines how often a gene is changed. Mutate rate of 0 results in no mutation and a mutate rate of 1 results in all genes being updated.
- *Mutate change* - It defines the degree of change that is applied on every gene.
- *Number of nodes* - It define the number of neurons in the hidden layer of the neural network. The higher the number of nodes is, the bigger the chromosome is.
- *Gene values* - The gene value defines the numeric range used for defining the initial random values of the chromosome.
- *Activation function* - The activation function is a non-linear function that produces the neurons output [6].

3 EXPERIMENTS & ANALYSIS

This part of the report presents the experiments and analysis performed as part of the project. The experiments focused on investigating:

- the different select, crossover, mutate and replace operators;
- parameter tuning for the neural network and the different EA operators.

Every experiment consisted of 10 consecutive trials using both test and training datasets. The results were then statistically analysed using the mean, median and standard deviation. T-tests were used for some of the experiments in order to prove that there was a significant difference between the means of the results.

3.1 Default values

The first experiment used the default settings of the project, presented in table 1 and 2 below.

Parameter	Value
Population	200
Hidden nodes	5
minGene	-3
maxGene	3
Mutate rate	0.01
Mutate change	0.05
Activation function	Tanh

Table 1: Default parameters

Type of operator	Operator
Selection	Random member
Reproduce	Copy of the parent
Mutate	Add or subtract the mutate change value
Replace	Replace worst regardless of fitness value

Table 2: Default operators

The results of this experiment are presented in table 3 below. The found figures were used as a starting benchmark and the goal was to improve them by changing the operators and tuning the EA's and NN's parameters.

	Training data	Test data
Mean	0.1463772008	0.2846437415
Median	0.1492445686	0.2924996899
Standard deviation	0.0159608531	0.04291812

Table 3: Results from default project

3.2 Selection

For the selection phase of the evolutionary algorithm, two operators were investigated - roulette and tournament selection. In both cases, all parameters from table 1 were kept the same and the only difference was the selection operator.

3.2.1 Roulette section. The results of roulette selection are presented in table 4 below. Based on the mean, median and the standard deviation, roulette selection did not improve the EA.

	Training data	Test data
Mean	0.1418105404	0.2829815891
Median	0.1475781729	0.3016731663
Standard deviation	0.0286908497	0.0789431463
Select operator	**- Roulette selection**	
Reproduce operator	- Copy of the parent	
Mutate operator	- Add or subtract the mutate change value	
Replace operator	- Replace worst regardless of fitness value	

Table 4: Results from roulette selection

3.2.2 Tournament selection. As can be seen in table 5, tournament selection appeared to be performing better than the random and roulette selection. This operator required exploration of the tournament size parameter. It was found that tournament size of 5 is the optimal value as it applied low selection pressure and produced results with lower standard deviation.

	Training data	Test data
Mean	0.1222741603	0.258188694
Median	0.1289010532	0.2587669168
Standard deviation	0.0286424055	0.0531130243
Select operator	**- Tournament selection of size 5**	
Reproduce operator	- Copy of the parent	
Mutate operator	- Add or subtract the mutate change value	
Replace operator	- Replace worst regardless of fitness value	

Table 5: Results from tournament selection

3.2.3 Comparison. As can be seem in figure 5, tournament selection outperformed roulette selection. Even though t-tests did not confirm that the operators are significantly different ($p = 0.14$ on training data and $p = 0.4$ on test data), tournament selection was the preferred approach as it had lower mean, median and standard deviation.

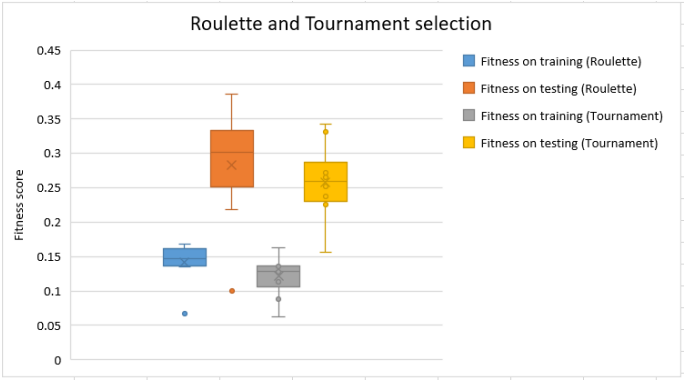

Figure 5: Selection operators

3.3 Crossover

Once the selection operator was updated to tournament selection, the experiments focused on the crossover operator. The EA's and NN's parameters were kept the same as in table 1.

3.3.1 1-point crossover. The crossover experiments started by implementing the 1-point crossover operator. As can be in table 6 below this improved the performance and decreased the mean and median values.

	Training data	Test data
Mean	0.1062115709	0.2384196868
Median	0.1122991801	0.2483131234
Standard deviation	0.0297123027	0.0509859733
Select operator	- Tournament selection of size 5	
Reproduce operator	**- 1-point crossover**	
Mutate operator	- Add or subtract the mutate change value	
Replace operator	- Replace worst regardless of fitness value	

Table 6: Results from 1-point crossover

3.3.2 N-point crossover. The experiments continued by implementing N-point crossover, where $N = 2, 3, 4$ and 5. At this stage of the experiments, it was found that 5-point crossover was the most optimal option.

	Training data	Test data
Mean	0.0928781019	0.2186619448
Median	0.0971089402	0.2161111309
Standard deviation	0.0269858286	0.0509485707
Select operator	- Tournament selection of size 5	
Reproduce operator	**- 5-point crossover**	
Mutate operator	- Add or subtract the mutate change value	
Replace operator	- Replace worst regardless of fitness value	

Table 7: Results from 5-point crossover

3.3.3 Uniform crossover. The last crossover operator that was investigated was uniform crossover. As can be seen in table 8 below, it was able to achieve similar results to the 5-point crossover.

	Training data	Test data
Mean	0.094816236	0.223113375
Median	0.100325882	0.235194144
Standard deviation	0.024583582	0.058965437
Select operator	- Tournament selection of size 5	
Reproduce operator	**- Uniform crossover**	
Mutate operator	- Add or subtract the mutate change value	
Replace operator	- Replace worst regardless of fitness value	

Table 8: Results from uniform crossover

3.3.4 Comparison. As can be seen in figure 6, 5-point crossover performed a bit better than the other two approaches. Even though t-tests did not confirm that the 5-point is significantly different from uniform, at this point of the experiments, 5-point was the preferred approach as it had lower mean, median and standard deviation.

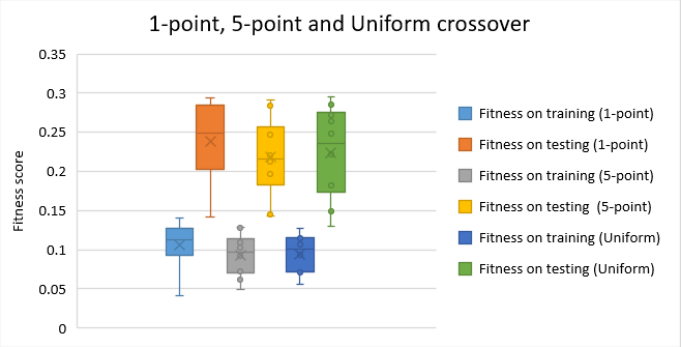

Figure 6: Crossover operators

3.4 Replacement

The next operator that was investigated was replacement. The EA's and NN's parameters were again kept the same as in table 1 and the experiments focused on the replacement operator only.

3.4.1 Replace if better. The first approach was to only replace the worst in the population, if and only if, its replacement has a lower fitness score (it is better at landing the spaceships).

	Training data	Test data
Mean	0.119619536	0.262431769
Median	0.123918718	0.276938965
Standard deviation	0.014372067	0.03635581
Select operator	- Tournament selection of size 5	
Reproduce operator	- 5-point crossover	
Mutate operator	- Add or subtract the mutate change value	
Replace operator	**- Replace worst if its replacement has a lower fitness value**	

Table 9: Results from replace if better

3.4.2 Tournament replace. The second operator that was investigated was tournament replacement. Similarly to tournament selection, tournament replacement has a tournament size parameter that is subject to exploration. It was found that having tournament size of 2 produced the most optimal results.

	Training data	Test data
Mean	0.096361685	0.225569524
Median	0.105490094	0.237754049
Standard deviation	0.024930484	0.052764828
Select operator	- Tournament selection of size 5	
Reproduce operator	- 5-point crossover	
Mutate operator	- Add or subtract the mutate change value	
Replace operator	**- Tournament replacement of size 2**	

Table 10: Results from tournament replacement

3.4.3 Tournament replace ensuring diversity. It appeared that from the two operators, tournament replace was a better option. However, in order to ensure bigger diversity in the population an additional functionality was implemented. It ensured that replacements have lower fitness scores by a predefined margin (0.001)

and that there are no members of the population that have similar chromosomes (at least 20% of the gene values should be different).

	Training data	Test data
Mean	0.056901155	0.148614233
Median	0.061553216	0.155975913
Standard deviation	0.036694515	0.073541378
Select operator	- Tournament selection of size 5	
Reproduce operator	- 5-point crossover	
Mutate operator	- Add or subtract the mutate change value	
Replace operator	**- Tournament replacement ensuring diversity of size 2**	

Table 11: Results from tournament replacement ensuring diversity

3.4.4 Comparison. As can be seen in figure 7, *tournament replace ensuring diversity* outperformed the other two approaches. This can be confirmed by the t-tests according to which *tournament replacement ensuring diversity* is significantly different from *replace if better* ($p = 0.00009$ on training data and $p = 0.00036$ on test data) and *tournament replacement* ($p = 0.01152$ on training data and $p = 0.01501$ on test data). Based on this, *tournament replacement ensuring diversity* was the preferred replacement operator.

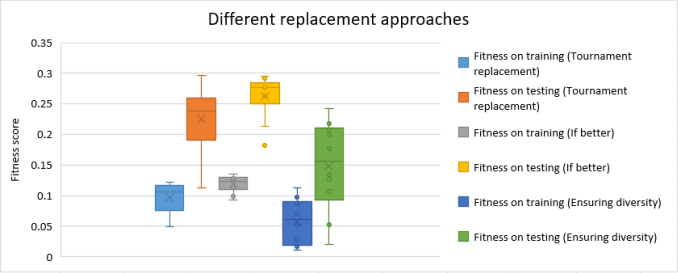

Figure 7: Replacement operators

3.5 Mutate

The last operator that was investigated was mutation. This part of the experiments, aimed at decreasing the high standard deviation that was found in the last section. In order to achieve this, parameters exploration was done for the neural network and the EA. As a result the following parameters were found to be optimal:

Parameter	Value
Population	200
Hidden nodes	**8**
minGene	**-1.0**
maxGene	**1.0**
Mutate rate	**0.2**
Mutate change	**- Gaussian distribution with mean equal to 0.5 and standard deviation of 0.1**
Activation function	Tanh

Table 12: Updated parameters

After running the experiment 10 consecutive trials, the following result were found:

	Training data	Test data
Mean	0.01842667	0.042967665
Median	0.012360074	0.028648918
Standard deviation	0.015145155	0.049628983
Select operator	- Tournament selection of size 5	
Reproduce operator	- 5-point crossover	
Mutate operator	- Add or subtract the mutate change value	
Replace operator	- Tournament replacement ensuring diversity of size 2	

Table 13: Results from changes in the mutate operator

3.5.1 Comparison. As can be seen in figure 6, the updated NN and EA parameters decreased the standard deviation and outperformed the default ones. This was also confirmed by t-test according to which the two result sets are significantly different ($p = 0.00667$ on training data and $p = 0.00142$ on test data).

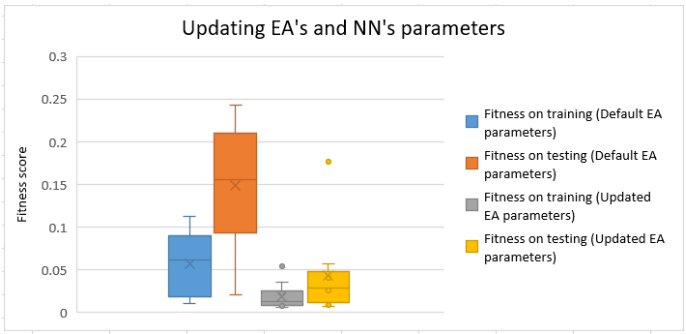

Figure 8: Updating EA's and NN's parameters

3.6 Decreasing population size and reinvestigating selection operators

At this point, it appeared that the EA is performing well on the training data, but not so well on the test data. One of the hypothesis was that the EA required more generations in order to produce a good NN. In order to check this, the number of population was reduced to 100, which decreased the amount of initially evaluated members.

Parameter	Value
Population	**100**
Hidden nodes	8
minGene	-1.0
maxGene	1.0
Mutate rate	0.2
Mutate change	- Gaussian distribution with mean equal to 0.5 and standard deviation of 0.1
Activation function	Tanh

Table 14: Updated parameters

In order to keep the selection pressure at the same level, the tournament selection size was decreased to 2 as well. Tournament replacement size was not changed, which appeared to apply higher replacement pressure. As can be seen from table 15 below, this improved the performance of the EA on the test data and decreased the standard deviation.

	Training data	Test data
Mean	0.0167311985	0.0292534296
Median	0.0170309352	0.0200260205
Standard deviation	0.0036875053	0.0204475179
Select operator	**- Tournament selection of size 2**	
Reproduce operator	- 5-point crossover	
Mutate operator	- Add or subtract the mutate change value	
Replace operator	- Tournament replacement ensuring diversity of size 2	

Table 15: Results after setting population to 100

Due to the fact that the earlier experiments on the crossover operator were not conclusive, the experiments focused again on this operator. As can be seen in table 16 below, this time, 1-point crossover appeared to be the more stable approach that avoided overfitting.

	Training data	Test data
Mean	0.0124894482	0.019152061
Median	0.011541219	0.0138892768
Standard deviation	0.0032076313	0.011330734
Select operator	- Tournament selection of size 2	
Reproduce operator	**- 1-point crossover**	
Mutate operator	- Add or subtract the mutate change value	
Replace operator	- Tournament replacement ensuring diversity of size 2	

Table 16: Results from 1-point crossover

3.6.1 Comparison. As can be seen in figure 9, the updated NN's and EA's parameters decreased the standard deviation and outperformed the default ones. Even though this was not confirmed by t-test ($p = 0.24089$ on training data and $p = 0.15632$ on test data), these were the preferred population size and crossover operator, because it produces lower mean, median and standard deviation.

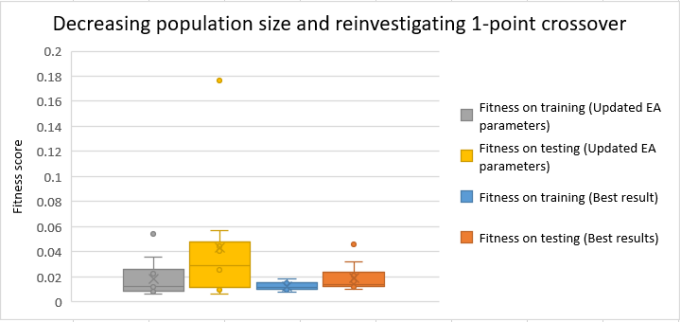

Figure 9: Best results

4 FUTURE WORK

The initial plan for the replacement operator was to investigate elitism. However, due to time constraints the implementation was not successful, hence was not included in the final version of the project and is considered as future work. In addition to this, it will also be interesting to experiment and analyse how the different activation functions (e.g. Sigmoid and ReLU) perform and to what extend they affect the final results.

5 CONCLUSION

Table 17 below shows the set of parameters and operators that were found as a result of the experiments presented in section 3 above. Considering the low mean values on test and training data, this approach can be considered as successful.

	Training data	Test data
Mean	0.0124894482	0.019152061
Select operator	- Tournament selection with tournament size of 2	
Crossover operator	1-point crossover	
Mutation operator	- Add or subtract the mutate change value	
Replacement operator	- Tournament replacement ensuring diversity of size 2	
Mutate rate	0.2	
Mutate change	- Gaussian distribution with a mean equal to 0.5 and a standard distribution of 0.1	
minGene	-1.0	
maxGene	1.0	
Population	100 members	
Hidden nodes	8 nodes	
Activation function	Tanh	

Table 17: Best results, operators and parameters

It was interesting to observe that at the beginning of the experiments 1-point crossover did not appear as an efficient operator for crossover. However, once the mutate and replace operators were implemented, this helped the 1-point crossover to produce lower mean score than 5-point.

In addition to this, it was also interesting to find that having a tournament replacement that ensures diversity in the population, helped for the more stable convergence of the ANN. Last but not least, based on what was found, it appeared that the higher the mean value of the Gaussian distribution was, the better the EA was performing.

REFERENCES

[1] Agoston E Eiben and James E Smith. 2008. Introduction to evolutionary computing (natural computing series). *Publisher: Springer-Verlag New York, LLC* (2008).

[2] M El-Shorbagy. 2016. A HYBRID GENETIC ALGORITHM FOR JOB SHOP SCHEDULING PROBLEMS. *International Journal of Advancement in Engineering, Technology and Computer Sciences (IJAETCS)* 3 (01 2016), 6–17.

[3] Deepti Gupta and Shabina Ghafir. 2012. An overview of methods maintaining diversity in genetic algorithms. *International journal of emerging technology and advanced engineering* 2, 5 (2012), 56–60.

[4] Usama Mehboob, Junaid Qadir, Salman Ali, and Athanasios Vasilakos. 2014. Genetic Algorithms in Wireless Networking: Techniques, Applications, and Issues. *Soft Computing* 20 (11 2014). https://doi.org/10.1007/s00500-016-2070-9

[5] K. Rutczyfflska-Wdowiak. 2017. Replacement strategies of genetic algorithm in parametric identification of induction motor. In *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*. 971–975. https://doi.org/10.1109/MMAR.2017.8046961

[6] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction, Second edition.* MIT Press, 224.

[7] TutorialsPoint. [n. d.]. Genetic Algorithms - Parent Selection. https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_parent_selection.htm. ([n. d.]). Accessed on 2018-04-08.

[8] F. Vafaee, G. Turn, P. C. Nelson, and T. Y. Berger-Wolf. 2014. Balancing the exploration and exploitation in an adaptive diversity guided genetic algorithm. In *2014 IEEE Congress on Evolutionary Computation (CEC)*. 2570–2577. https://doi.org/10.1109/CEC.2014.6900257