Vrije Universiteit Amsterdam

Diversity Promotion by implementation of Island Model in Evolutionary Algorithm for optimisation

Evolutionary Computing - Standard Assignment - Task 1

Group 108:
Rick Geling (2804908)
Matúš Halák (2724858)
Isis van Loenen (2676369)
Rens Koppelman (2750795)

Abstract

Diversity is crucial in evolutionary algorithms (EAs) for avoiding premature convergence in local optima. The Island Model (IM) EA aims to enhance diversity by running multiple sub-populations (islands) in parallel and exchanging a fraction of individuals after certain intervals. This paper investigates the impact of an IM-EA with a ring topology on population diversity and solution quality compared to a baseline EA. The EAs are trained and tested on the EvoMan Framework using artificial neural networks (ANN). Results demonstrate that there is no significant difference between the diversity of the baseline EA and the IM-EA, and the solution quality achieved by the IM-EA is only improved compared to the baseline EA when trained upon the environment for which the hyperparameters were tuned. Therefore we emphasize the need of further research to investigate the population diversity promotion of using an IM-EA more extensively.

Keywords: Evolutionary Algorithm, Island Model, Fitness Sharing, Diversity, EvoMan

1 Introduction

For natural evolution to occur there are three essential components: heredity, selection and variation. Since natural evolution can be approached as a fitness optimisation problem, these three components are crucial as well when using Evolutionary Algorithms (EAs) for optimisation problems in general. In this paper we will mainly focus on the importance of variation, also referred to as diversity.

Variation is mainly essential in the exploration process of the algorithm where it explores the so called adaptive landscape, which conceptualizes the potential solutions with their corresponding quality (fitness) in height [5]. One of the main challenges in solving optimisation problems is the presence of several local optima, which is referred to as multi-modality., This could cause the algorithm to converge prematurely without finding the global optimum. Maintaining a considerable amount of variation helps to escape from the local optima and stimulate the algorithm into exploring the landscape further [3].

The need for more sophisticated algorithms adjusted to multimodal problems has driven the development of, among others, the Island Model (IM) EAs. In IMs multiple populations (islands) are ran in parallel in attempt to explore different regions of the solution landscape and a migration policy controls the communication between the island in terms of exchanging individuals after a defined period, also referred to as an epoch [1]. Many variants of IMs have been designed varying in migration policies and topologies, often mainly focusing on improvements in speed and performance [1] rather than its direct influence on the population diversity, which can also be defined by several measures as described by Črepinšek et al. (2013) [12].

To further enlighten the benefits of IMs considering diversity, we aim to investigate how the population diversity on the genotype level is enhanced by using an IM-EA compared to a baseline EA and whether this impacts the fitness of generated solutions for an optimisation problem. To demonstrate this, we use the well-documented electronic-game framework called EvoMan [8] [9] to design the two EAs specialized on a specific environment. The NeuroEvolution EAs generate Artificial Neural Network (ANNs) fit to the given problem by adjusting its weights in order to optimise the fitness [7]. In view of our focus on promoting diversity during evolution, the used IM-EA has a ring topology, because empirical analysis has shown that this variant generates the highest diversity during evolution compared to the lattice and full-connected IM topologies [11]. This method will contribute to our understanding of how an IM-EA impacts the population diversity and thereby the solution quality for optimisation.

2 Methods

2.1 Representation of individuals

The representation of the individual solutions is a fixed ANN architecture. The ANN consists of 20 input nodes, sensing the locations and distances of the agent, the enemy and the 6 objects fired by the enemy. The hidden layers consists of 10 nodes and is connected to the output layer consisting of 5 nodes. Additionally, the hidden and output nodes have biases (in total: 15). Therefore, each individual represents a vector of 265 parameters, encoding for the weights to be set for the ANN generating the strategy of the agent.

2.2 Baseline EA

To evaluate the effect of the IM-EA we first designed a baseline EA specialized on the sixth enemy option called 'Crashman' [9], chosen because during initial testing with it seemed to be one of the harder enemies to achieve consistently good performance against. We reasoned that hyperparameters tuned against a harder enemy could yield an EA robust across enemies. The used hyperparameters are listed in 1. Population size, crossover rate and mutation rate are tuned upon this environment using GridSearch going over 75 value combinations. The tournament size (k) is set to be 10% of the population size. The initial mutation step size is set to be 0.05 corresponding to 2.5% of the search space. The sharing distance is set by scaling the maximum genetic Euclidian distance as suggested by Deb and Goldberg (1989) [4]. The BLX-α is set similarly as was described by Eiben and Smith (2015) [5]. The elite fraction was set to 0.8 by experimental evaluation. Using these hyperparameters the baseline EA was constructed implementing fitness sharing [10], tournament selection for parent selection, BLX-alpha recombination [6], uncorrelated mutation with one self-adaptive step size, and survivor selection with elitism as described by Algorithm 1.

Table 1: Hyperparameters tuned for baseline-EA

Hyperparameter	Value
Population size*	120
Crossover rate*	0.85
Mutation rate*	0.25
Tournament size (k)	15
Initial mutation step size ($\sigma_{mutation}$)	0.05
Sharing distance (σ_{share})	4.88
BLX-alpha (α)	0.5
Elite fraction for survivor selection	0.8

*Parameters which are tuned using GridSearch

Algorithm 1 Baseline EA

```
Initialize a population of individuals randomly.
Evaluate the fitness of each individual.
Set the best solution to the current best individual.
for each generation until the maximum is reached do
    Encourage diversity through fitness sharing.
    Select parents using a tournament-selection.
    Create offspring by BLX-α recombination of parents.
    Mutate offspring by Uncorrelated Mutation with σ_mutation.
    Select the next generation based on fitness using elitism.
    Evaluate the fitness of the new population.
    Record fitness and diversity statistics.
    Update the best solution if a fitter individual is found.
end for
return The best solution found.
```

2.3 Island Model EA

For the Island Model EA the same hyperparameters are used as in the baseline EA with the additional IM-specific hyperparameters of which the majority is tuned by GridSearch going over 27 possible value combinations, as listed in 2. Besides, the number of migrating individuals is set to 1/5 of the island population, i.e. 8. The IM-EA is described by Algorithm 2, using the same evolution loop for each individual per generation as in the baseline EA.

Table 2: Hyperparameters tuned for IM-EA

Hyperparameter	Value
Number of islands*	3
Epoch time (generations) *	20
Fraction of migrating individuals	0.2
Elite fraction*	0.65

*Parameters which are tuned using GridSearch

Algorithm 2 Island Model EA

```
Initialize the population of solutions over islands.
for each epoch until the maximum is reached do
    for each island do
        Evaluate the fitness of each individual on the island.
        Set the best solution to the best individual on the island.
        for each generation within the epoch do
            Perform the evolutionary process as in the baseline EA.
        end for
    end for
    Migrate a fraction of individuals to neighboring islands.
end for
return The best solution found.
```

2.4 Evaluation procedure

For quality evaluation of each individual we used the following fitness function which both EAs aimed to maximise:

$$fitness = 0.9 \cdot (100 - e_e) + 0.1 \cdot e_p - \log t \tag{1}$$

, where e_e is enemy energy at the end of the episode, e_p is player energy and t is the time left. Besides the fitness progress during the evolutionary training, we measured the population genetic diversity per generation defined by the gene-wise standard deviation across all individuals of the population, which was then averaged over the 265 genes. After tuning of the hyperparameters on enemy 6, as described above, both the baseline EA and the IM-EA were trained 10x on enemy 6 and two other enemies. The other chosen enemies are enemy 5 'Metalman' and 8 'Quickman', because they seemed similar in the enemy's strategy of moving and jumping a lot, although for enemy 5 the movement is caused by the particular floor [9]. For each of the combinations of EA and enemy the individual with the highest fitness was saved, which subsequently battled 5 times against each of the three enemies. From these final battles we analyzed the individual gain, defined as:

$$individual\ gain = e_p - e_e \tag{2}$$

We evaluate the differences between the baseline EA and the IM-EA by comparing the average best fitnesses, diversities and individual gains over the generations. For these comparisons we used the non-parametric Wilcoxon signed rank test, as the assumption for normality was violated.

3 Results and discussion

3.1 Comparison of best fitness

Comparing the best fitness over all the generations showed a significantly lower performance of the IM-EA compared to the baseline EA when trained upon enemy 5 [p-value = 0.00025], see Figure 1A. In contrast, for enemy 6 the best fitness taken over all generations was significantly higher [p-value = 0.0001], see 1B. For enemy 8 there was no significant difference between the EAs [p-value = 0.57], see 1C. This could suggest that specialized tuning of hyperparameters, which was done for enemy 6, is especially beneficial for IM-EA relatively to the baseline EA.

Interestingly, despite the fact that hyperparameters were tuned on enemy 6, the best obtained fitnesses and the mean fitnesses against this enemy are lower than for enemy 5 and 8 for both EAs. Moreover, both EAs need more generations to reach their maximum values against enemy 6. These results suggest that enemy 6 seems to be harder to deal with. More extensive tuning of the hyperparamters, for instance by using Bayesian optimisation for all hyperparameters, could improve the specialized performances of the EAs.

These results show a minor performance improvement by the IM-EA compared to the baseline EA. This could be improved by more extensive hyperparameter tuning and/or by adjustments such as done in the Diversity-driven Migration Strategy by Araujo and Batista (2023) [2], where the migration intervals are dynamic and dependent on the assessed loss in diversity. This could help in using the migration only when needed. Moreover, in this method the individual(s) to be migrated are not chosen randomly as in our IM-EA, but chosen depending on their dissimilarity with the target island. Furthermore, when discussing the performance of IM-EA it should be noted that the fitness sharing algorithm as used in the baseline EA was also implemented into the IM-EA. Fitness sharing is another way of promoting population diversity [10] by decreasing fitness of individuals which are more similar. When applying this in the IM-EA, the within-island diversity is pushed as well as the among islands diversity. Combining the two diversity promotion techniques could cause excessive exploration obstructing exploitation towards higher fitness levels. Whether this combination is indeed disadvantageous could be investigated by future research.

3.2 Comparison of diversity

As expected, for both EAs and for each enemy the population diversity decreases over time which is a characteristic of convergence, see 2C, D and E. However, the diversity in the IM-EA is not significantly different during the process compared to the baseline EA for all three enemies with p-values > 0.05 (although it frequently peaks much higher than baseline EA diversity). Remarkably, the population diversity for IM-EA fluctuates severely, whereas the diversity of the baseline decreases smoothly. The fluctuations are more frequent than the migration events, which is suspicious and requires further investigation in order to state whether diversity is effectively promoted.

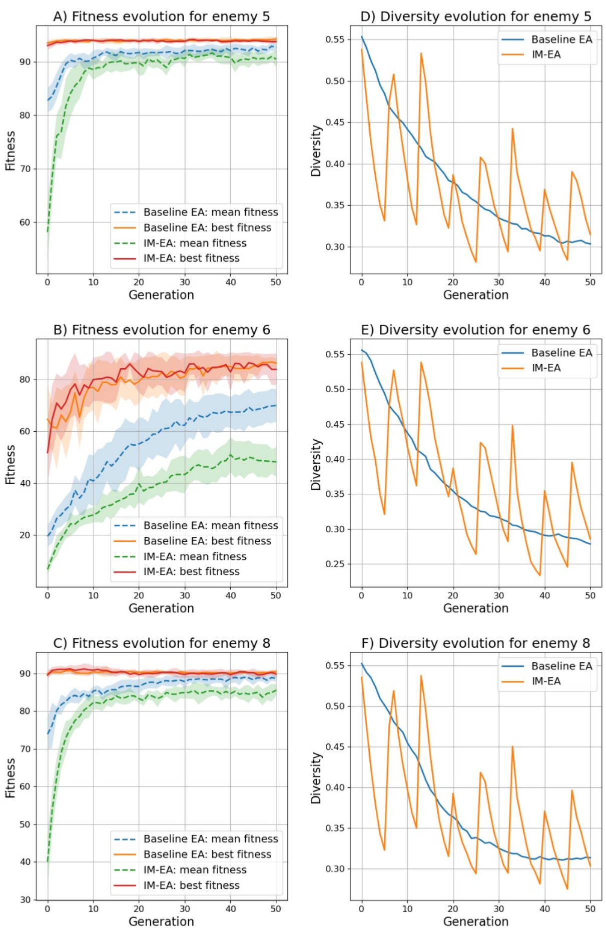

Figure 1: Fitness and diversity evolution: for the baseline EA and the IM-EA the mean fitness and the best fitness over the generations is shown for enemy 5 (A), enemy 6 (B) and enemy 8 (C). The population diversity over the generations for both EAs is shown for enemy 5 (D), enemy 6 (E) and enemy 8 (F).

3.3 Comparison of gain

The gains of the best performing individuals per EA-enemy combination, see 2, show that there are no significant differences between the baseline EA and the IM-EA with all p-values >0.05 , meaning that the differences in generated best fitness do not result in a significantly different gains of the best performing individual. Besides, these results support the difficulty of enemy 6 compared to enemy 5 and 8 as well. The median gain for both EAs applied to all enemies are positive, meaning that the agents we trained were consistently able to beat all enemies tested against. This pattern was also found in the results of Miras and Olivetti (2016) [9] where the final energy level of the agent battling against enemy 6 was equal to 0, and for enemy 8 lower than for enemy 5, when using their Genetic Algorithm.

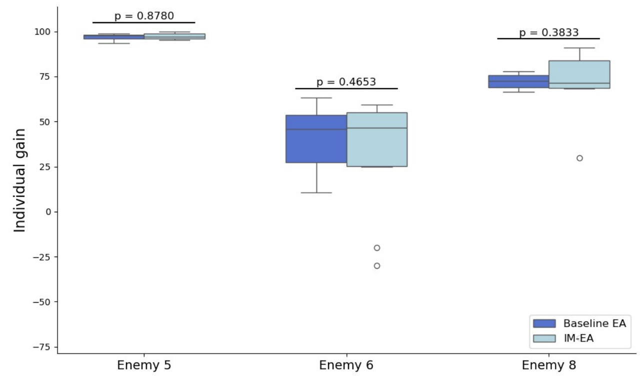

Figure 2: Differences in individual gain: for each enemy and for both EAs the best individual is saved for 10 optimizations, each of them ran 5 times of which the average is taken. The resulting individual gains are shown in boxplots.

4 Conclusion

To conclude, this paper focused on investigating the potential of IM-EA to promote population diversity on the genotype level compared to a baseline EA and its impact on the generated best fitness. Our results showed that there is no significant difference between the population diversities on the genotype level between the baseline EA and the IM-EA for any of three tested enemies. Additionally, there is only a significant improvement in the best fitness of the IM-EA compared to the baseline EA when trained upon the enemy for which the hyperparameters are tuned, emphasizing the importance of hyperparameter tuning. In addition to more extensive hyperparameter tuning and considering more specialized and advanced IM-variants, further research could focus on the implementation of fitness sharing within IM-EA. This would extend our understanding of its effect on population diversity and improve the effectiveness of IM-EA for optimisation problems.

References

[1] E. Alba and M. Tomassini. 2002. Parallelism and evolutionary algorithms. *Proceedings of the IEEE Transactions on Evolutionary Computation* 6, 5 (Jan. 2002), 397–404. https://doi.org/10.1109/TEVC.2002.800880

[2] Jean N.R. Araujo and Lucas S. Batista. 2023. A diversity-driven migration strategy for distributed evolutionary algorithms. *Swarm and Evolutionary Computation* 82 (2023), 101361. https://doi.org/10.1016/j.swevo.2023.101361

[3] E.K. Burke, S. Gustafson, and G. Kendall. 2004. Diversity in genetic programming: an analysis of measures and correlation with fitness. *IEEE Transactions on Evolutionary Computation* 8, 1 (2004), 47–62. https://doi.org/10.1109/TEVC.2003.819263

[4] Kalyanmoy Deb and David E. Goldberg. 1989. An investigation of niche and species formation in genetic function optimization. (1989), 42–50.

[5] A.E. Eiben and J.E. Smith. 2015. *Introduction to Evolutionary Computing* (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-44874-8

[6] Larry J. Eshelman and J. David Schaffer. 1992. Real-Coded Genetic Algorithms and Interval-Schemata. (1992). https://api.semanticscholar.org/CorpusID:26430737

[7] Dario Floreano and Claudio Mattiussi. 2008. Neuroevolution: From architectures to learning. *Evol Intell* 1 (03 2008). https://doi.org/10.1007/s12065-007-0002-4

[8] K. Miras and F. Olivetti. 2016. An electronic-game framework for evaluating coevolutionary algorithms. (2016). arXiv:1604.00644 [cs.NE] https://arxiv.org/abs/1604.00644

[9] K. Miras and F. Olivetti. 2016. Evolving a generalized strategy for an action-platformer video game framework. (07 2016), 1303–1310. https://doi.org/10.1109/CEC.2016.7743938

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

[11] Ting-Chen Wang, Chih-Yu Lin, Rung-Tzuo Liaw, and Chuan-Kang Ting. 2019. Empirical Analysis of Island Model on Large Scale Global Optimization. (06 2019), 342–349. https://doi.org/10.1109/CEC.2019.8790331

[12] Matej Črepinšek, Shih-hsi Liu, and Marjan Mernik. 2013. Exploration and Exploitation in Evolutionary Algorithms: A Survey. *Comput. Surveys* 45 (06 2013), Article 35. https://doi.org/10.1145/2480741.2480752