Bio-Inspired AI Project
A Quality-Diversity Co-Evolutionary Scenario

Gabriele Padovani, *gabriele.padovani@studenti.unitn.it*
Nadia Benini, *nadia.benini@studenti.unitn.it*

I. Introduction

The importance of thorough search algorithm has been a pivotal in several fields, both of computer science and applied to different external subjects. For the former, robotic is one of the first which comes to mind, especially for path-finding and routing purposes. As for the latter, some non-theoretical implementations have been developed on the matter of Bio-Informatics, for example for matching similar sequences of DNA.

It is however well known that these categories of algorithms, especially the ones in the evolutionary and genetic subsets, are extremely susceptible to getting stuck in local optima, or struggling to converge. To obviate to this issue, Quality-Diversity algorithms were developed with the intent of separating agents into clearly defined classes, which could show different behaviours based on their attributes.

II. Background

Before describing the work central to this report, it is mandatory to introduce four main topics, essential to understanding this paper.

A. Reinforcement Learning

Reinforcement Learning (RL) is a machine learning approach where an agent learns to make decisions in an environment in order to maximise a reward function. The agent interacts with the environment, receives feedback in the form of a reward or penalty and adjusts its actions accordingly.

One drawback of these algorithms is the computational complexity of having to simulate the entire environment and all agents within it. The reason is these systems need to produce a complete enough reconstruction of the problem for the agent to make an accurate decision.

B. MAgent2

MAgent2[4] is a research framework for multi-agent reinforcement learning. It provides a set of environments for training and testing multi-agent systems using RL algorithms. The environment used in this project is *combined arms*, where two armies composed of several classes of agents with different capabilities clash. Melee and ranged units are considered, with the former moving more slowly but having more Hit Points (HP) and the latter being able to fight from a distance, move faster but have less HP. Agents slowly regain HP over time, specifically they have 10 HP and are damaged 2 HP by each attack. Once safe from combat they recover 0.1 HP every turn.

Being this a reinforcement learning scenario, each agent is attributed reward points for:

- Killing an opponent: 5 points;
- Taking a movement step: -0.005 points;
- Attacking any agent: -0.1 points;
- Attacking an opponent: 0.2 points;
- Dying: -0.1 points.

The map onto which agents fight is 45x45 cells in size, each individual is able to see up to 13 cells away from where it is positioned, as shown in Fig.1.

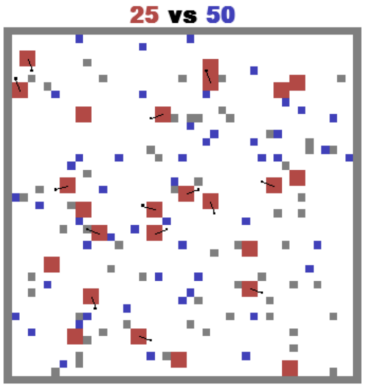

Fig. 1: A screenshot of the MAgent2 library running

The evaluation of a MAgent2 environment can be fine-tuned using three main parameters:

1) Max Cycles: describes the number of time steps each simulation runs for;
2) Max Episodes: indicates how many times each simulation is repeated without changing any parameter, results are then averaged and returned to MAP-Elites;
3) Epochs: indicated how many times MAP-Elites iterates, in other words how many times it is able to change the configuration of soldiers for new fights.

C. Quality-Diversity

Quality-Diversity is a computational approach derived from the specific need of Evolutionary Algorithms to avoid local optima[1]. This approach focuses on generating a diverse set of high-quality solutions rather than optimising a single objective. The main idea is to maintain a population of diverse individuals and aim to cover a wide range of possible outcomes,

allowing each agent, or species of agents, to develop its own behaviour, which may be very different from an individual in another class.

D. MAP-Elites

Multi-dimensional Archive of Phenotypic Elites[3] (MAP-Elites) is a quality-diversity algorithm which maintains a grid or archive of high-performing solutions across multiple traits. It explores the solution space by incrementally filling the archive with elite solutions found in different regions. This algorithm allows the discovery of a set of diverse, high quality solutions[2].

The aim of this project is to develop a MAP-Elites algorithm to find the optimal army composition to beat the adversary, who will try to do the same.

While the official paper[3] encourages leaving empty cells, to allow for quicker research in the available space, the flexibility offered by the MAgent2 library, which allows choosing the amount of iterations for each simulation, meant the implementation in this project could start by filling every tile in the grid, yielding more precise results.

III. Design Choices

The MAgent2 library allows for experimentation on a wide variety of benchmarks, from gathering simulations, where evolution can be regarded to as cooperative, to battling ones, so competitive. The scenario chosen for this project is *combined_arms*, which consists in a competitive co-evolutionary game, where different classes of agents fight against each other on teams.

Although it is possible to create fully customizable environment, as well as new agent types, since the aim of this project was to test the MAP-Elites algorithm, it was opted to keep the default environment and only modify the initial formation of agents.

At the beginning of every simulation, each teams is positioned in one of three formations:

- The default grid formation;
- Randomly on any tile;
- In a square formation.

What was noticed during testing of these, is that the formation does not seem to influence the overall outcome, as long as it is balanced for both teams. Specifically, if a formation favors the left team, for instance by placing the ranged units of the right group in front (Fig.2), results back up this bias.

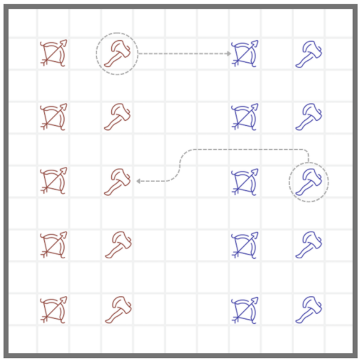

Fig. 2: An instance of an unfair starting formation

The initial version of the program made use of a single grid of elites, this however caused the two agents to always oppose each other with the same formation, resulting often in luck-based encounters. To avoid this issue, an adversarial grid was created, allowing each agent to develop its own soldier formation.

Depending on the amount of cells in each grid, the MAP-Elites algorithm can be tweaked to have more fine-grained definition of the feature space at the expense of giving up some performance. The cell grid size is set to twenty for all the conducted experiments, as this value was considered a good compromise for a feature space of a hundred fighters divided into two classes.

The stopping criterion chosen for all the runs in this report is simply if the number of generations is exceeded. Two configurations of stopping criteria have been implemented, the aforementioned one and a patience-based method, which stops execution if the fitness doesn't improve after a certain number of iterations. This latter method however, turned out to be too unstable, as many consequent runs may happen on cells which do not have an optimal soldier configuration, yielding low fitness scores. Overall, a more advanced patience method could have been implemented, keeping track or several cell state, it was however opted to follow a simpler approach.

Two mutation strategies have been implemented, namely a completely random strategy, where the algorithm selects stochastically a value inside the range of the cell, and a Gaussian method, where a random value is sampled from a normal distribution and added to the current cell, always making sure this is inside the bounds. Since the Gaussian addition often results in a very minor modification, a mutation incentive was also added, which defines the minimum distance the cell has to change by. This feature probably tilts the algorithm too much towards the exploration aspect, however, since the execution times are extremely long, it was decided to keep this change in all experiments.

On the matter of crossover, since MAP-Elites limits a lot its effect separating individuals into classes, it was opted to implement a simpler approach revolving around calculating the mean between two individuals. The sum between the two elites is also weighted by an crossover parameter, set to 0.2.

Finally, several cell selection strategies are implemented:

- Random: which selects a cell stochastically;
- Adjacent: which selects the first cell at random, then an adjacent cell (for crossover);
- Best: which selects always the best cell, if the best is already selected (for crossover), then selects the second best;

- Gaussian Best: selects the best cell probabilistically, depending on how good the elite is;
- Random Best: selects with probability p a cell at random, otherwise chooses the best cell.

Although the results for all these methods are not reported in this paper, the two better performing ones are the completely random approach and the random-best one. This is probably due to them being the best compromise between the exploration and exploitation aspects.

IV. Experiments

The long execution times for each experiment rendered infeasible a broader exploration of more possibilities, in the sense of more combination of epochs and run iterations.

To keep execution times manageable, two configuration where chosen:

1) 500 epochs, 3 iterations: which ignores the improvements happening inside the environment execution, and focuses on the MAP-Elites algorithm;
2) 100 epochs, 30 runs: which was thought to be a good compromise between the two aspects.

In all experiments, it is assumed that the grid initialization uses 100 cycles, so the number of time-steps the fight goes on, to give a baseline value for further exploration. On the other hand all consequent runs are executed with 100000 cycles instead. This expedient also reduces the grid initialization times by a non-trivial amount.

V. Results

Reporting the findings of this project, several benchmarks will be used[3]:

- **Global Reliability**: indicates how the performance of the grid improves over time, by calculating an average fitness over all the cells in feature space;
- **Global Performance**: measures how performing the algorithm is, by reporting the highest fitness obtained, as well as the cell composition.
- **Coverage**: measuring how many cells of the feature space a run of an algorithm is able to fill.

In our case, a preliminary run of the algorithm is executed, filling every cell, so coverage is 100%. It is still possible, through the activity grids in Fig.5, to see how many times each cell is selected for updating.

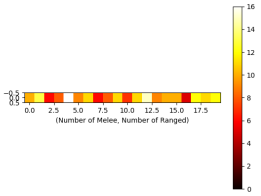

Fig. 3: Primary Agent

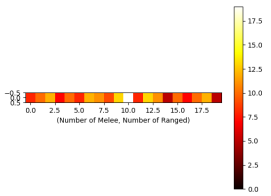

Fig. 4: Adversarial Agent

Fig. 5: Activity grid with 500 epochs

TABLE I: Results of MAP-Elites Runs

Selection	Epochs	Iterations	Composition	Fitness
Random	100	30	(95, 5)[1]	-898.72
Random	500	3	(94, 6)	-549.18
Best	100	30	(90, 10)	-574.72
Best	500	3	(95, 5)	-415.37

As for global performance, all the results have been reported in Table I, where only the final configuration of the best performing army is taken into consideration.

Finally, for reliability, all findings reported come from the execution of the program for either 30 or 3 times for each epochs, and the fitness is averaged. A visual representation of this can be seen in Fig.6 and Fig.7, where the average fitness over all elites in the primary and adversarial grids are plotted.

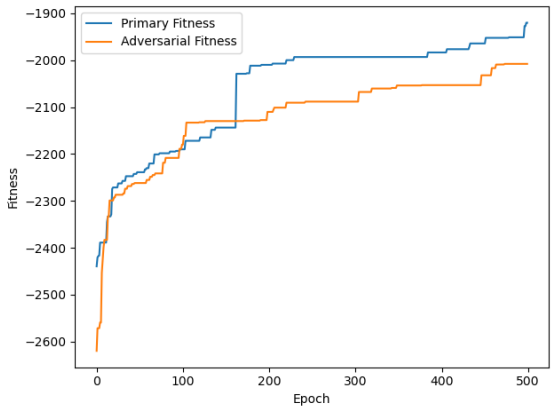

Fig. 6: Fitness function for 500 epochs and 3 runs

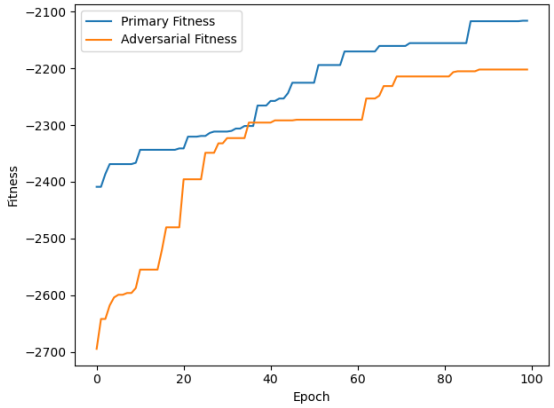

Fig. 7: Fitness function for 100 epochs and 30 runs

In Fig.7 it can be seen how the reward score raises quickly for the first epochs, then starts to plateau towards the end, as long as enough epochs are played out.

As for displaying visually the results, in Fig.12 are shown, staring from the top left: the primary performance grid at epoch 0, the same grid after 200 epochs and on the lower row the adversarial performance grid in the same settings.

Due to space constraints, only the results for the experiments using random and random best cell selection are reported,

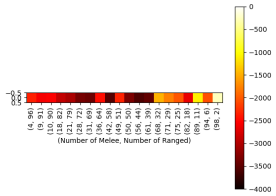

Fig. 8: Primary Agent at epoch 1

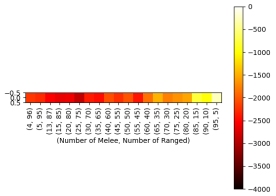

Fig. 9: Primary Agent at epoch 500

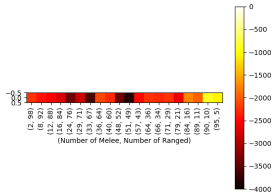

Fig. 10: Adversarial Agent at epoch 1

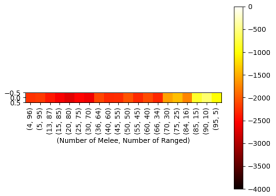

Fig. 11: Adversarial Agent at epoch 500

Fig. 12: Performance grid with 500 epochs

but several configurations of tests have been conducted, and always seem to favor melee fighters over archers.

As for a more Quality-Diversity oriented analysis, going back to the Performance Grids in Fig.12, it is clear that this scenario is not balanced enough for ranged units to succeed. The MAP-Elites algorithm clearly prefers melee fighters and it is tricky to find a balanced situation without completely render one or the other too beneficial. In a scenario of this type, it might be always optimal to reduce the number of classes and trade all diversity for performance.

VI. Conclusion

With this report, the trade-offs between the quality and the diversity in a population have been shown. It is clear that, in a very stochastic procedure like the one in question, a class of individuals may be so much more performant that it renders the second one useless.

One possible improvement to this project, could be to implement more suitable maps, where even ranged units could excel, for example by adding walls they can shoot over. Another improvement could be letting the simulations run for longer, where due to time constrains it was not able to conduct more thorough experiments.

Overall, this report shows the major trade-offs between Quality and Diversity in this specific scenario, and how in a competitive environment, one agent is able to gain an hedge over another with the aid of the evolutionary process.

References

[1] Andrea Ferigo, Leonardo Lucio Custode, and Giovanni Iacca. *Quality Diversity Evolutionary Learning of Decision Trees*. 2022. arXiv: 2208.12758 [cs.NE].

[2] Devon Fulcher. *The MAP-Elites Algorithm: Finding Optimality Through Diversity*. URL: https://medium.com/@DevonFulcher/the-map-elites-algorithm-finding-optimality-through-diversity-def6dcbc0f5b.

[3] Jean-Baptiste Mouret and Jeff Clune. *Illuminating search spaces by mapping elites*. 2015. arXiv: 1504.04909 [cs.AI].

[4] Lianmin Zheng et al. "MAgent: A many-agent reinforcement learning platform for artificial collective intelligence". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.