Humane approach to Starcraft micromanagement: not only victory, evaluate losses as well

Author 1
University of Author 1
City 1, Country 1
Author1

ABSTRACT

TODO In this paper we investigate the training of an artificial neural network that controls a squad of units in the StarCraft:Brood Wars game so that the squad not only achieves the highest proportion of victories, but does so with the smallest number of casualties. **TODO**

CCS CONCEPTS

• **Applied computing → Computer games**;

KEYWORDS

Starcraft, micromanagement, NEAT, neuroevolution, artificial neural networks, evolutionary algorithms, RTS

ACM Reference Format:
Author 1. 2018. Humane approach to Starcraft micromanagement: not only victory, evaluate losses as well. In *Proceedings of the Genetic and Evolutionary Computation Conference 2018 (GECCO '18).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

TODO

2 LITERATURE REVIEW

2.1 NEAT

NEAT is a *NeuroEvolution* method. The principle is to *start with a minimal topology and grow it* to match the problem difficulty in order to find an appropriate network. It does that by adding and removing neurons, changing weights, adding connections, … Any neuron can be connected to any neuron including itself, meaning even *recurrent neural networks* can be generated. At each generation high fitness networks get better chance of reproduction. But even so, *NEAT maintains genetic diversity* through a process called *speciation*. That is, similar networks are considered to be in the same species. Then, to encourage innovation, *explicit fitness sharing* is performed on members of the same species. Each species is assigned a number of offsprings it can produce. Then species' members compete against each others. [2]

Various variant of the method has been created to answer specific problems such as *Cascade-NEAT*.

2.2 Cascade-NEAT

Restrict the search process to topologies that have a cascade architecture, good for fractured problems.

Article: learning in fractured problems with constructive neural networks algorithms.

Article: evolving neural networks for strategic decision-making problems.

2.3 Novelty search

explanation todo [1]

2.4 Potential fields

Micromanagement potential fields multi-objectives genetic algorithm. Micromanagement using Potential Fields tuned with a Multi-Objective Optimized Evolutionary Algorithm (namely NSGA-II). Present a model to handle priority targets, focus firing, retreat spots, … Talk about how to get a behaviour optimizing multiple objectives. Use of the killscore.

Siming et al. [3]

2.5 Others

Applying reinforcement learning to small scale combat in the strategy game starcraft broodwar. Uses Q-learning and Sarsa.

Connectionist reinforcement learning for intelligent unit micro management in starcraft. Uses neural networks to evaluate Q-values for Sarsa reinforcement learning method. Results on incremental learning. Interesting part for reward units: dead units don't get a negative reward for dying. Instead the reward of the dead unit is the average reward of all living units in the next state. It introduce a sort of cooperative behaviour. See neural-fitted Q-iteration.

NeuroEvolution for micromanagement in rts game starcraft bw. About battle micromanagement and NEAT vs rtNEAT.

3 PROPOSAL

Techniques:

- Vanilla NEAT

GECCO '18, July 15–19, 2018, Kyoto, Japan
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

- Cascade-NEAT
- Novelty Search and vanilla NEAT
- 'Unified' networks with vanilla NEAT

A fresh C++ implementation of these methods was developed and is hosted on Github under the name of NEToolKit[1].

The Starcraft bot can be found on Github as well under the name of Evolubot[2].

The analysis of the data was performed with the R language. Source can, again, be found on Github[3].

4 MATCHUPS AND EXPERIMENTS

- 22 marines vs 22 marines (with stimpacks)
- 22 marines vs 44 zergs (with stimpacks)
- 22 vultures vs 22 vultures
- 22 vultures vs 22 zealots

For each matchup and for each technique, we run three times an evolution from scratch up to the 100th generation. Then, 50 games using the five best genomes produced during the evolution.

During the evolution the following data are logged:

- number of survivors
- average fitness of each generation
- best fitness of each generation
- best current fitness ever for each generation

For the best units games, we only need to log the number of survivors.

We consider the number of survivors to be a relevent criteria. Indeed, a high number of survivors means the skirmish went well and in a full game, these survivors can be reallocated to the next battle and can thus influence the long term outcome. However, this information is not showed as much as we would like to. Siming et al. [3] though also included some measurement of loss and survivors in their evaluation.

To encourage cooperation and survival rate, we integrate the number of survivors in the fitness function.

Marines with stimpack may be a deceptive problem since it's basicaly losing health to gain performance. We hope the agents learns to do so in order to win and increase survivors.

Especially, the marines vs zerg problem may be good to learn how to use stimpacks since it's very difficult for marines to win without using them. So, more stimpack usage is expected for this problem.

Novelty currently just use fitness as novelty metrics. Not best use case at all for this method.

TODO: add all parameters for experiments.

5 EXPERIMENTS AND RESULTS

5.1 Developping stimpack behaviour

Vanilla NEAT doesn't seems to be good a developing the stimpack behaviour since organisms that survives without stimpack get better reward, even though they may have survived because others used stimpacks to get in the frontline and got themselves killed first. So, organisms tends to lose this behaviour over generations.

CHECK: Cascade-NEAT and Novelty [expected to be approximately identical].

In comparison, the unified version, which is good a developing uniform behaviours was somewhat better at developing this behaviour. Indeed, since every agents use the same type of neural network, all units in the round has the behaviour encoded and don't win because others did all the hard stuff.

Sadly the marines vs marines problem can be solved efficiently just by waiting for the enemy and attacking all out mostly without stimpacks. Depending on the enemy's formation at it's arrival it can be even more effective. They can indeed shoot first with more firepower.

Novelty search got a significantly better result on the marines vs zergs experiments, meaning it was able to develop some stimpack behaviours. [need more behaviour analysis]

Conclusion: not a big success. Overall failure.

5.2 Resulting behaviours

…

5.3 Survivors and victory comparison

…

6 DISCUSSION

TODO

7 CONCLUSION

TODO

ACKNOWLEDGEMENTS

I would like to thank ACKNOWLEDGEMENT 1 for their assistance in the preparation of this manuscript. AUTHOR 1 was partially funded by ACKNOWLEDGMENT 2.

REFERENCES

[1] Joel Lehman and Kenneth O. Stanley. 2011. Abandoning Objectives: Evolution through the Search for Novelty Alone. *Evolutionary Computation* 19 (2011).

[2] Kenneth O. Stanley and Risto Miikkulainen. 2002. Evolving Neural Networks through Augmenting Topologies. (2002).

[3] Liu Siming, J. Louis Sushil, and Christopher Ballinger. 2014. Evolving Effective Micro Behaviors in RTS Game. (2014).

[1] https://github.com/CBenoit/NEToolKit

[2] https://github.com/CBenoit/Evolubot

[3] https://github.com/CBenoit/EvolubotAnalysis