Noname manuscript No.
(will be inserted by the editor)

Search, Abstractions and Learning in Real-Time Strategy Games

A Dissertation Summary

Nicolas A. Barriga

Received: date / Accepted: date

Abstract Real-Time Strategy games' large state and action spaces pose a significant hurdle to traditional AI techniques.

We propose decomposing the game into sub-problems and integrating the partial solutions into action scripts that can be used as abstract actions by a search or machine learning algorithm. The resulting high level algorithm produces sound strategic choices, and can then be combined with a low-level search algorithm to refine tactical choices.

We show strong results in SparCraft, Starcraft: Brood War and μRTS against state-of-the-art agents.

We expect advances in RTS AI can be used in commercial videogames for playtesting and game balancing, while also having possible *real-world* applications.

Keywords Real-Time Strategy Games · Game Tree Search · Deep Convolutional Neural Networks · Evolutionary Algorithms

1 Introduction

Games have long been used as Artificial Intelligence (AI) testbeds, possibly since Claude Shannon proposed the first chess playing program and argued *"that a satisfactory solution of this problem will act as a wedge in attacking other problems of a similar nature and of greater significance."* [15,14]. Great advances have been made since then, and traditional board and card games have been surrendering to AI, from Backgammon [18] and Checkers [13] to Go [16,17] and Heads-up Limit Hold'em Poker [7]. Still, videogames, with their faster rate of play and vastly larger action and state spaces have proven more resilient. AI agents for early and simple computer games have reached human level strength [9] in the last few years. However, most current game genres are still out of reach even to the most recent algorithmic advances. However, this is an area of active research, and since the publication of this dissertation, significant progress has been made [19].

Considering the possible real-life applications of advanced videogame AI, ranging from machine translation to self-driving cars, as well as the $100+ billion market, incentives are high to develop new suitable AI methods.

Real-time strategy (RTS) games simulate war campaigns in which the players gather resources, build a base and army, and engage in combat. Multiple actions can be executed at once, and the game world is only partially visible. RTS games make for particularly attractive research platforms, due to the availability of a strong professional player community—providing competent benchmarks—and their similiarities to board games. This resemblance (e.g. zero-sum, grid-worlds, independent units) gives us a beachhead, by adapting techniques successful at playing Chess or Go, to RTS games.

2 Main Contributions

In our dissertation [1] we propose decomposing the problem of playing an RTS game into sub-problems, integrating these solutions into multiple expert scripts. These scripts can then be used as abstract actions by a game tree search algorithm, which will select the ap-

N.A. Barriga
Universidad de Talca
2 Norte 685 – Talca, Chile
E-mail: nbarriga@utalca.cl

propriate one depending on the current game state. Aditionally, we show that Convolutional Neural Networks (CNNs) can be used for evaluating game states and predicting the output of the game tree search algorithm.

Finally, we combine this resulting high-level strategic algorithm, with a low-level tactical algorithm.

2.1 Sub-Problem Solutions

We start by tackling the building placement subproblem [2]. An evolutionary algorithm, using the outcome of several attack waves as a fitness function, optimizes the position of buildings in a defending base. Empirical results are presented in SparCraft [8], a *Starcraft: Brood War* combat simulator.

2.2 Search over Abstract Actions

Next, we present *Puppet Search* [3,6,4], a search algorithm that uses scripts as action abstractions. These scripts can use search or machine learning based subproblem solutions, such as pathfinding, building placement or combat, for strong tactical performance. The exploration of the search space, coupled with the long search horizon facilitated by using scripts rather than low level actions, enables Puppet Search to make reliable strategic choices.

We show that Puppet Search outperforms it's individual component scripts in *Starcraft: Brood War*, as well as surpassing other state-ofthe-art agents in μRTS [10], a simple RTS game designed for academic research.

2.3 Learning Abstract Action Selection

The next step, is to train a CNN as an evaluation function for Puppet Search. The training set is generated by playing several state-of-the-art agents against each other, and saving states from those games, labelled with the winner of the game. Thus, the network reaches a high accuracy at predicting which side is winning in any given game state, starting at 74% early in the games, and rapidly increasing as the games progress. Using Puppet Search with this network, we produce a second training set, in which game states are labelled with the outcome of the search algorithm, and a second CNN is trained to predict the output, replacing the search procedure, but only using a fraction of the time [5].

2.4 Strategy/Tactics

We then combine the strategic Puppet Search (or the CNN predicting its output), with NaïveMCTS [11], a tactical algorithm with strong performance in small scenarios. We use the strategic algorithm to assign actions to most units in the game, while the tactical algorithm focuses on small scale battles. The combined performance was strong enough to win the first μRTS AI competition at IEEE Conference on Computational Intelligence and Games (CIG) 2017 [12].

3 Conclusions

This dissertation presented the use of scripts as an action abstraction mechanism. These scripts can include sub-problem solutions, such as pathfinding, build-order search, combat resolution or our proposed evolutionary algorithm for building placement optimization.

The abstract action selection can be performed by a search algorithm, such as Puppet Search, or a supervised classifier, such as a convolutional neural network. We finally show how to combine tactical low-level search algorithms with strategic high-level decision making. All of our claims are backed by experimental results in standard RTS test enviroments (SparCraft, Starcraft: Brood War and μRTS) and competitions.

3.1 Future Work

Since the publication of this dissertation, the team behind AlphaStar [19] has mostly solved the problem of reaching a high level of play in RTS games. However, the computing power, time and effort needed to implement their solution for other games is out of reach for most commercial game developers looking to automate playtesting and game balancing. There is room for further research in the area, exploring simpler—although maybe partial—solutions.

References

1. Barriga, N.A.: Search, abstractions and learning in real-time strategy games. Ph.D. thesis, University of Alberta (2017)
2. Barriga, N.A., Stanescu, M., Buro, M.: Building placement optimization in real-time strategy games. In: Workshop on Artificial Intelligence in Adversarial Real-Time Games, AIIDE (2014)
3. Barriga, N.A., Stanescu, M., Buro, M.: Puppet Search: Enhancing scripted behaviour by look-ahead search with applications to Real-Time Strategy games. In: Eleventh Annual AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE), pp. 9–15 (2015)

4. Barriga, N.A., Stanescu, M., Buro, M.: Combining scripted behavior with game tree search for stronger, more robust game AI. In: Game AI Pro 3: Collected Wisdom of Game AI Professionals, chap. 14. CRC Press (2017)
5. Barriga, N.A., Stanescu, M., Buro, M.: Combining strategic learning and tactical search in real-time Strategy games. In: Accepted for presentation at the Thirteenth Annual AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE) (2017)
6. Barriga, N.A., Stanescu, M., Buro, M.: Game tree search based on non-deterministic action scripts in real-time strategy games. IEEE Transactions on Computational Intelligence and AI in Games (TCIAIG) (2017)
7. Bowling, M., Burch, N., Johanson, M., Tammelin, O.: Heads-up limit hold'em poker is solved. Science **347**(6218), 145–149 (2015). DOI 10.1126/science.1259433. URL http://www.sciencemag.org/content/347/6218/145.abstract
8. Churchill, D.: SparCraft: open source StarCraft combat simulation. http://code.google.com/p/sparcraft/ (2013). URL http://code.google.com/p/sparcraft/
9. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
10. Ontañón, S.: The combinatorial multi-armed bandit problem and its application to real-time strategy games. In: AIIDE (2013)
11. Ontañón, S.: Combinatorial multi-armed bandits for real-time strategy games. Journal of Artificial Intelligence Research **58**, 665–702 (2017)
12. Ontañón, S., Barriga, N.A., Silva, C.R., Moraes, R.O., Lelis, L.H.: The first microrts artificial intelligence competition. AI Magazine **39**(1) (2018)
13. Schaeffer, J., Lake, R., Lu, P., Bryant, M.: CHINOOK: The world man-machine Checkers champion. AI Magazine **17(1)**, 21–29 (1996)
14. Shannon, C.: A Chess-playing machine. Scientific American **182**, 48–51 (1950)
15. Shannon, C.E.: Programming a computer for playing chess. Philosophical Magazine (First presented at the National IRE Convention, March 9, 1949, New York, U.S.A.) **Ser.7, Vol. 41, No. 314** (1950)
16. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of Go with deep neural networks and tree search. Nature **529**(7587), 484–489 (2016)
17. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al.: Mastering the game of Go without human knowledge. Nature **550**(7676), 354 (2017)
18. Tesauro, G.: TD-Gammon, a self-teaching Backgammon program, reaches master-level play. Neural Computation **6**(2), 215–219 (1994)
19. Vinyals, O., Babuschkin, I., Chung, J., Mathieu, M., Jaderberg, M., Czarnecki, W.M., Dudzik, A., Huang, A., Georgiev, P., Powell, R., Ewalds, T., Horgan, D., Kroiss, M., Danihelka, I., Agapiou, J., Oh, J., Dalibard, V., Choi, D., Sifre, L., Sulsky, Y., Vezhnevets, S., Molloy, J., Cai, T., Budden, D., Paine, T., Gulcehre, C., Wang, Z., Pfaff, T., Pohlen, T., Wu, Y., Yogatama, D., Cohen, J., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Apps, C., Kavukcuoglu, K., Hassabis, D., Silver, D.: AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/ (2019)