Searching by learning: Exploring artificial general intelligence on small board games by deep reinforcement learning

Wang, H.

Chapter 1

Introduction

1.1 Background

Modern Artificial Intelligence (AI) research began in the mid-1950s [1]. Mainstream AI scientists focused on the narrow AI research which usually aims to solve a single sub-problem (specific task). In 1990s, some researchers started to try to develop artificial general intelligence (AGI), also known as strong AI [2, 3], by combining the programs that solve various sub-problems (different tasks). AGI is the hypothetical intelligence of a computer program that has the capacity to understand or learn any intellectual task that a human being can [4]. Although current AI techniques have achieved impressive performance in mastering specific tasks, AGI it is still speculated to be decades away [5]. Therefore, it is useful to further study how far the current techniques (especially in reinforcement learning landscape [6]) can bring us to AGI, which I will do in this thesis. I will now list a few challenges in AGI to frame the contributions in this thesis.

Reinforcement Learning

There are many of AI techniques, such as searching, reasoning, pattern recognition and learning that achieve impressive successes [7, 8, 9, 10]. In reinforcement learning, as this thesis will emphasize, searching and learning are both important. Well-known techniques are Monte Carlo Search (MCS) [11], Monte Carlo Tree Search (MCTS) [7, 12] and table based (or neural network based) Q-learning [13, 14, 15]. These techniques have shown impressive capability in

mastering practical problems, especially the recent success of training the AlphaGo series of programs playing Go, Chess and Shogi [10, 16, 17]. The AlphaGo programs use an architecture that combines MCTS and neural network training, which has become a highly successful paradigm of deep reinforcement learning for high-dimensional problems. Before neural networks were used, table-based Q-learning has been employed in combination with MCTS, resulting to the same structure combining online search and offline learning [18]. It is interesting to study this approach.

General Game Playing

General game playing (GGP) is a well-known testbed for AGI. The goal of GGP is to play previously unknown board games, where the rules are not known in advance. The program must play the game without human intervention. The games are described using a standard game description language; legal moves can be automatically generated. Thus, in writing the program, the GGP-author can use search and learning techniques. For example, MCTS and its variants achieve quite good performance on the GGP system [19, 20, 21, 22]. However, there are few works that apply deep reinforcement learning to play GGP games. Therefore, a table based Q-learning should be investigated for GGP to enter the deep reinforcement learning era.

AlphaZero

AlphaZero [10] can also be regarded as an AGI framework. AlphaZero provides a general framework to play Go, Chess and Shogi. In fact, it is implemented to play a class of two-player zero sum games. Therefore, it is a promising testbed for AGI with deep reinforcement learning. The landmark achievements of the AlphaGo series of programs have created a large research interest into self-play in reinforcement learning. In self-play, MCTS is used to train a deep neural network, that is then used in tree searches. Training itself is governed by many hyper-parameters. There has been surprisingly little research on design choices for hyper-parameter values and loss functions, presumably because of the prohibitive computational cost to explore the parameter space.

Expert Data

In addition, we note that the creators of AlphaGo use data from expert games for AlphaGo, but not for AlphaGo Zero nor for AlphaZero, which are so-called tabula rasa approaches. However, a further program, in this series, AlphaStar,

for StarCraft, does use expert data again [23]. There is little research into the necessity of expert data. Well studied MCTS enhancements, such as Rapid Action Value Estimation (RAVE) [24, 25, 26] can improve the performance of table-based Q-learning, but there is no research reported on applying such enhancements to the AlphaZero framework.

Single Agent Combinatorial Optimization

Last but not the least, AlphaZero-like deep reinforcement learning is initially developed for two-player games, where it is highly successful. Therefore, it is interesting to see how it could be transferred to deal with single agent combinatorial optimization games and to benefit the solution of combinatorial problems in computer science.

Overview

Overall, in this dissertation, we focus on GGP and the AlphaZero framework to test promising and novel ideas to explore AGI. The computational demands of these approaches are high. By using small board games we are able to perform many experiments while retaining many aspects of larger games.

Specifically, in Chapter 2, we assess the potential of classical Q-learning in GGP, together with dynamic ϵ and MCS enhancements. Then, in Chapter 3, we investigate 12 hyper-parameters in an AlphaZero-like self-play algorithm and evaluate how these parameters contribute to training. Next, Chapter 4 evaluates the alternative loss functions of AlphaZero-like self-play to study the importance of policy function and value function. Subsequently, we propose a warm-start search enhancement method to boost training at the start phase of self-play training in Chapter 5, which evidences the necessity of expert data and we show how these data can be generated by MCTS enhancements. In the following Chapter 6, we further propose an adaptive warm-start method to dynamically control the warm-start length during training. In Chapter 7, we embed a ranked reward algorithm within AlphaZero-like self-play to challenge a well-studied single player combinatorial game, Morpion Solitaire, and obtain a near human level result as a first attempt.

In Fig 1.1, the structure of this dissertation is depicted as a diagram. The diagram shows that the main work of our thesis is based on two frameworks (GGP and AlphaZero-like self-play). For each framework, different learning and searching techniques are investigated in different chapters.

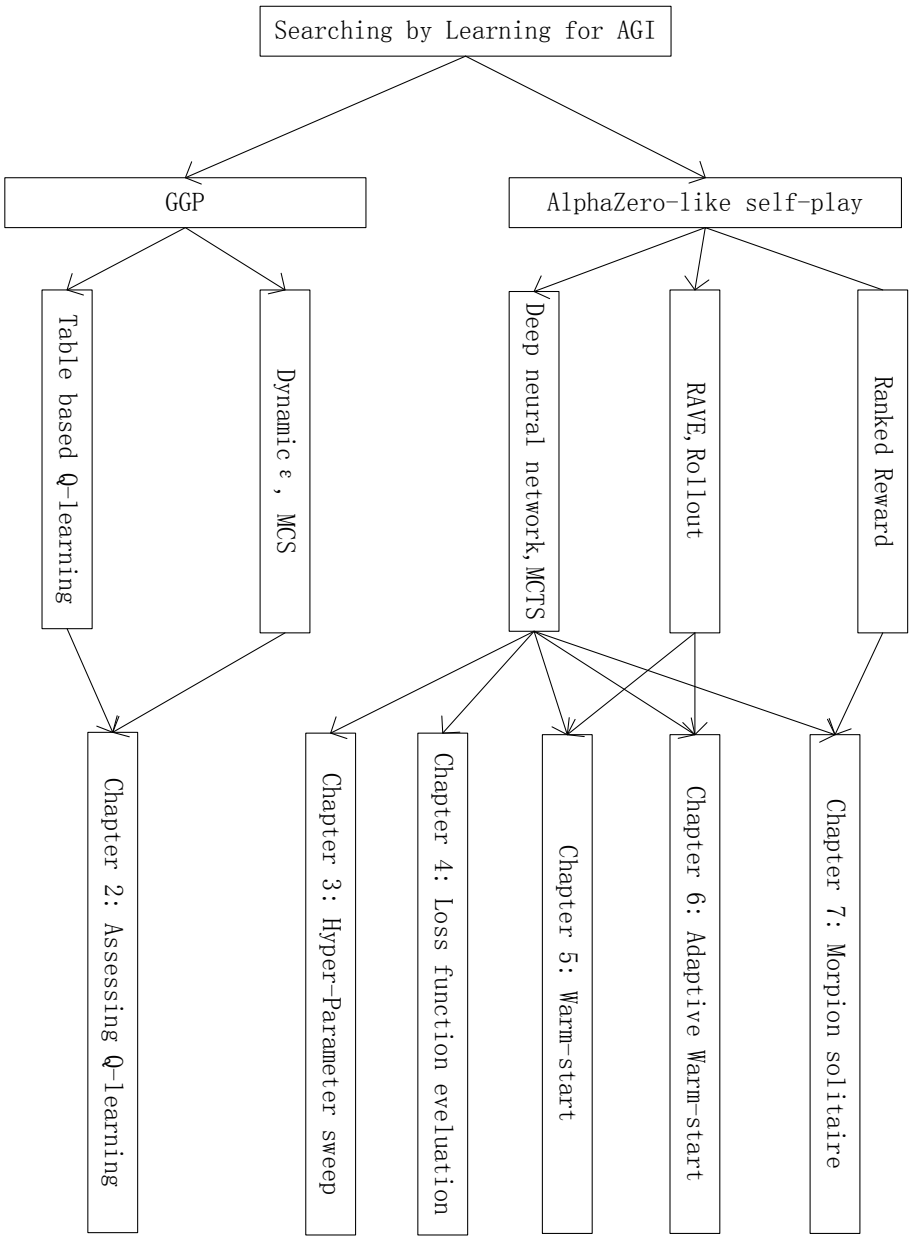

Figure 1.1: A general structure of this dissertation

1.2 Research Questions

In the following, we list the research questions of this thesis and sketch the approaches used to find answers to them.

RQ1 (Chapter 2) How to assess the potential of classical Q-learning in GGP? In order to build a bridge between GGP and deep reinforcement learning, the initial step is to assess the potential of table-based Q-learning on GGP system. We test different values for fixed ϵ and then propose a dynamic ϵ enhancement, where the ϵ is used to balance the exploration and exploitation of Q-learning. Furthermore, we introduce another enhancement with MCS which is to generate better training examples at the beginning when the Q-table has no record of the state. All experiments are tested on small board games, such as Tic-Tac-Toe, 3×3 Hex and 4×4 Connect Four. In order to assess convergence with different board size, 3×3, 4×4 and 5×5 Tic-Tac-Toe are also investigated.

RQ2 (Chapter 3) How hyper-parameters contribute to AlphaZero-like self-play? A hyper-parameter sweep is computationally expensive; Little is known on how to set these hyper-parameters, there is a need to provide insight into how the hyper-parameters contribute to training efficiency. AlphaZero-like self-play can be divided into three stages, namely *self-play*, *neural network training* and *arena comparison*. We identify 12 main hyper-parameters for this framework. The explanation of each hyper-parameter is given, and a light hyper-parameter sweep is performed. This sweep provides an overview of the hyper-parameter contributions. Besides, a further study of the interaction between important hyper-parameters is also performed in this dissertation, which helps to understand the trade off between searching and learning.

RQ3 (Chapter 4) How alternative loss functions work in AlphaZero-like self-play? Players in AlphaZero consist of a combination of MCTS and a deep neural network, that is trained using self-play. A unified deep neural network is used, which has a policy-head and a value-head. During training, the optimizer minimizes the **sum** of policy loss and value loss. However, it is not clear if and under which circumstances other formulations of the loss function are better. Therefore, we perform experiments with different combinations of these two minimization targets. In contrast to many recent papers who adopt single run experiments and use the whole history Elo ratings from self-play, we propose to use repeated runs. The results show that this method can describe the training performance quite

well within each training run. Because of a high self-play bias a final best player Elo rating is adopted to evaluate the playing strength in a direct competition between the evolved players, inspired by the approach reported by the AlphaZero team.

RQ4 (Chapter 5) Can MCTS enhancements be used to replace human experts to improve the AlphaZero-like self-play? AlphaZero's design is purely based on self-play and makes no use of labeled expert data or domain specific enhancements; it is designed to learn from scratch. We propose a novel approach to deal with this cold-start problem by employing simple search enhancements at the beginning phase of self-play training. We use Rollout, RAVE and dynamically weighted combinations of these with the neural network, and Rolling Horizon Evolutionary Algorithms (RHEA).

RQ5 (Chapter 6) How to control the warm-start length? While tuning warm-start length for different enhancements, the results show that it is costly and usually unstable. Therefore we propose an adaptive method to control the warm-start length of using MCTS enhancements by employing an arena to determine whether the enhancement is not better any more. The experimental results show that our approach works better than the fixed I', especially for deep, tactical, games (Othello and Connect Four). We conjecture that the adaptive value for I' is also influenced by the size of the game, and that on average I' will increase with game size. We conclude that AlphaZero-like deep reinforcement learning benefits from adaptive rollout based warm-start, as RAVE did for rollout-based reinforcement learning 15 years ago.

RQ6 (Chapter 7) Can AlphaZero-like self-play be used to master complex single player combinatorial optimization games? Morpion Solitaire is a popular single player complex combinatorial optimization game, performed with paper and pencil [27, 28]. Due to its large state space (on the order of the game of Go) traditional search algorithms, such as MCTS, have not been able to find good solutions. A new algorithm, Nested Rollout Policy Adaptation, was able to find a new record of 82 steps, albeit with large computational resources [29]. Morpion Solitaire has never been studied in a deep reinforcement learning framework. A challenge of Morpion Solitaire is that the state space is sparse, there are few win/loss signals. Therefore, we use an approach known as ranked reward to create a reinforcement learning self-play framework for Morpion Solitaire. This enables us to find medium-quality solutions with reasonable computational effort.

Our record is a 67 steps solution, which is very close to the human best (68) without any other adaptation to the problem than using ranked reward.

1.3 Dissertation Outline

The dissertation outline is described in this section. For each main chapter of this dissertation, there is at least one publication by the author. A brief outline of this work is presented as follows.

★ Chapter 2 introduces the GGP system and the definition of classical Q-learning. In addition, two enhancements (dynamic ϵ and QM-learning) of classical Q-learning are proposed. The classical Q-learning and proposed enhancements are assessed in a GGP system to play several different games (different size of Tic-Tac-Toe, 3×3 Hex and 4×4 ConnectFour). The contents of this chapter are published in a preprint [30] and a conference paper [31] (**best regular paper award**).

> **Wang H.**, Emmerich M., Plaat A. (2018) Monte Carlo Q-learning for General Game Playing. arXiv preprint 1802.05944.
>
> **Wang H.**, Emmerich M., Plaat A. (2019) Assessing the Potential of Classical Q-learning in General Game Playing. In: Atzmueller M., Duivesteijn W. (eds) Artificial Intelligence. BNAIC 2018. Communications in Computer and Information Science, vol 1021. Springer.

★ Chapter 3 introduces the AlphaZero-like self-play framework with three stages in a single iterative loop, and identities 12 potentially important hyper-parameters. A hyper-parameter sweep is performed for every hyper-parameter and moreover, further experiments of four selected more interesting hyper-parameters are also performed and evaluated. Parts of this chapter are published in preprints [32, 33].

> **Wang H.**, Emmerich M., Preuss M., Plaat A. (2019) Hyper-Parameter Sweep on AlphaZero General. arXiv preprint 1903.08129.
>
> **Wang H.**, Emmerich M., Preuss M., Plaat A. (2020) Analysis of Hyper-Parameters for Small Games: Iterations or Epochs in Self-Play?. arXiv preprint 2003.05988, submitted to journal.

★ Chapter 4 introduces the default loss function of AlphaZero-like self-play, and three alternative loss functions. A running Elo is computed and a full

tournament Elo is also employed. Parts of this chapter are published in a conference paper [34] and a preprint [33].

> **Wang H.**, Emmerich M., Preuss M., Plaat A. (2019) Alternative loss functions in AlphaZero-like self-play. 2019 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, pp. 155-162.
>
> **Wang H.**, Emmerich M., Preuss M., Plaat A. (2020) Analysis of Hyper-Parameters for Small Games: Iterations or Epochs in Self-Play? arXiv preprint 2003.05988, submitted to journal.

★ Chapter 5 investigates the AlphaZero-like self-play start phase by employing MCTS enhancements to improve training performance. The description and analysis of RAVE and RHEA is provided. The work of the chapter is published in a conference paper [35].

> **Wang H.**, Preuss M., Plaat A. (2020) Warm-Start AlphaZero Self-play Search Enhancements. In: Bäck T. et al. (eds) Parallel Problem Solving from Nature – PPSN XVI. PPSN 2020. Lecture Notes in Computer Science, vol 12270. Springer.

★ Chapter 6 introduces the adaptive warm-start method, and a parameter tuning for fixed I' is also provided. The work of the chapter is published as a preprint [36].

> **Wang H.**, Preuss M., Plaat A. (2021) Adaptive Warm-Start MCTS in AlphaZero-like Deep Reinforcement Learning. arXiv preprint 2105.06136, submitted to conference.

★ Chapter 7 introduces how to embed the ranked reward mechanism into AlphaZero-like self-play. A description of Morpion Solitair game is provided. And a near-human level solution with 67 steps is presented. The work of the chapter is published in a conference paper [37].

> **Wang, H.**, Preuss, M., Emmerich, M. and Plaat, A. (2020) Tackling Morpion Solitaire with AlphaZero-like Ranked Reward Reinforcement Learning. 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). IEEE, pp. 149-152

★ Chapter 8 summarizes the contributions of this dissertation and highlights points to some interesting directions for future work.

★ In addition to the aforementioned publications, another publication of the author is [38]. This publication is related to the research, but was not part of this thesis.

> **Wang H.**, Tang Y., Liu J., Chen W. (2018) A Search Optimization Method for Rule Learning in Board Games. In: Geng X., Kang BH. (eds) PRICAI 2018: Trends in Artificial Intelligence. PRICAI 2018. Lecture Notes in Computer Science, vol 11013. Springer.