Solving *Legends of the Three Kingdoms* based on Hierarchical Macro Strategy Model

Youjie Zhang
Nanjing University of Posts and Telecommunications
Nanjing, Jiangsu
Q17010217@njupt.edu.cn

Xiang Ding
Nanjing University of Aeronautics and Astronautics
Nanjing, Jiangsu
dxokay@nuaa.edu.cn

Zhi Chen
Nanjing University of Posts and Telecommunications
Nanjing, Jiangsu
chenz@njupt.edu.cn

ABSTRACT

Legends of the Three Kingdoms is a Chinese card game based on the novel *Romance of the Three Kingdoms*. As a turn-based strategy game, *Legends of the Three Kingdoms* is an imperfect information game where agents interact with one another to deduce the unknown information. The most famous solution for an imperfect information game is to find a Nash equilibrium where no agent can benefit from changing strategies. Tencent AI Lab has recently developed a novel learning-based Hierarchical Macro Strategy model for AI to master MOBA games, a sub-genre of RTS games. In this paper, we adopt this method on our AI to solve *Legends of the Three Kingdoms* in a shrunken size of 4 players.

CCS CONCEPTS

• **Computing methodologies** → **Abstraction and micro-operators**; *Search with partial observations*; • **Theory of computation** → Solution concepts in game theory;

KEYWORDS

Legends of the three kingdoms, Imperfect information, Nash equilibrium, Hierarchical Macro Strategy Model

ACM Reference Format:
Youjie Zhang, Xiang Ding, and Zhi Chen. 2019. Solving *Legends of the Three Kingdoms* based on Hierarchical Macro Strategy Model. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3326748

1 INTRODUCTION

The idea of getting a computer to play a game has been around since the earliest days of computing. Game theory defines a game as a domain of conflict or cooperation between several entities [4]. For perfect information games, the agent normally would build a game tree starting at its own position, and then evaluate the outcomes of all its leaves by an evaluation function. Basically, the idea was to let the computer determine the optimized move based on the current sub game. [3] This method has been long used to solve perfect information games such as Go or Chess. But for real-world imperfect information games, this method cannot quite work, as the amount of leaves on the game trees is tremendous. The most famous solution for an imperfect game is to find a Nash equilibrium, where no player can benefit by changing strategies given that the other players keep theirs unchanged. According to John Nash, every game with a finite number of players has at least one Nash equilibrium, as long as the set of strategy choices of each player is finite. In order to achieve the most optimal solution, we need to find an abstraction algorithm to create the abstracted game first and then write a special algorithm for solving it, thus find a Nash equilibrium. In the next part I am going to discuss the abstraction algorithm first, and then our solving algorithm second.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07.
https://doi.org/10.1145/3319619.3326748

2 ABSTRACTION MODEL

As is mentioned above, like most poker games, *Legends of the Three Kingdoms* has a game tree of tremendous amount of leaves so that it is infeasible to scale to the size of that in practice. An integrated description of the game is fed to abstraction algorithm to yield a smaller but strategically similar, or even equivalent, game. According to Gilpin and Sandholm, any Nash equilibrium of the shrunken game corresponds to a Nash equilibrium of the original game. Admittedly, it is possible to do lossless information abstraction, which runs in $O(n^2)$ time, where n is the number of nodes in the signal tree. In this algorithm, two sibling nodes are considered ordered game isomorphic if (1) if the nodes are leaves, the payoff vectors of the players are the same at both nodes, and (2) if the nodes are interior nodes, there is a bipartite matching of the nodes' children so that only *ordered game isomorphic* children get matched [5] [1]. Take into account the size of the game, it is possible that after lossless abstraction, the resulting model is still too large to solve. To address this problem, a more aggressive lossy information abstraction is feasible, despite the loss in solution quality. In this case, siblings are considered ordered game isomorphic if their children can be approximately matched in the bipartite matching part of the algorithm [2].

3 HIERARCHICAL MACRO STRATEGY MODEL

3.1 Overview

A two-person zero-sum game can be formulated as a saddle-point problem, which can later find an equilibrium. However, in many real-world games such as poker, these linear programs are tremendous and unsolvable via standard algorithms such as interior-point methods. Certain algorithms, like gradient-based algorithms are applied to those circumstances. But for games involving more than

2 players, or with team cooperation, that would be a lot harder. So we propose a Hierarchical Macro Strategy (HMS) model, based on recent findings by Tencent AI Lab on their own MOBA games solving [6].

3.2 Phase Layer

Phase layer aims to determine the current phase of game. Human player may have different major tasks in different phases, for example in the opening phase player in the Monarch's side tends to proof it's loyalty and find the Rebel and in the later phase it may change it's task to, like, kill the Rebel and protect the Monarch. Basically the phases' change are correlated to time. To solve that, we propose to use the identity revelation of the whole game as a landmark of phase. With a identity identification algorithm based on the actions taken by players, we can predict player's side and when it comes to a high certainty, we could move forward to the next phase.

Currently, The standard mode of *Legends of the Three Kingdoms* can be played by up to 10 people divided into four parties: the Monarch, the Minister, the Rebel, and the Traitor. Different roles have different victory conditions. For example, the Minister's goal is to protect the Monarch and the Rebel is to kill the Monarch without leaving one Traitor as the sole survivor. Those objectives are all related to one player, the Monarch. So actually finding a player's identity is to find out the player's interaction with Monarch. In our model, we will mainly discuss a 4 player scenario. With data provided by human playing records, we could form a basic classifier for each parties' action toward the Monarch and then if some player in a new round shows similar behavior, their probability of some role increase. For example, if some player attacks the Monarch we could identify that player as a Rebel or Traitor, and if someone helps the Monarch then that player is a Minister.

Each time with a new identity confirmed, the phase moves forward. For a simplified 4 player scenario, there would be at most 3 phases. Take the Monarch as an example, At first, we need to determine who's the Rebel and who's the Traitor, then we could determine players' rough identity, forming an alliance confronting another, then with one side down, we could move to the final phase and we should at that time know all identities.

3.3 Attention Layer

Similar to how players playing cards and making decisions during the whole game, our attention layer determines the best move player can play. *Legends of the Three Kingdoms* is a sophisticated game, so we build a trigger system corresponding with a skill-card system for our AI to better comprehend the game.

The skill-card system plays a role in merging skills and card-playing together by making all skills invisible cards. For there is no essential difference between the use of some commanders' skills and the concept of playing a card. All cards can be defined by the four dimensions, the timing of use, the conditions of use, the price and the effect. Take *Kill Card* as an example, the timing of use is commander's own stage of playing, the conditions of use are that the other party is within the scope of the commander's attacking range, and they do not use the *Kill Card* multiple times in this round, the price is after the card is used, the card is discarded, and the effect is to ask the other party to play a *Dodge Card*, otherwise it will drop HP.

Some commanders' skills obviously have the same characteristics. For example, *checks and balances* is a skill of Sun Quan. The timing of use is his own stage of playing cards. The conditions of use are that he has a card and he has not launched it many times in the same round. The price is to discard a number of cards and the effect is to get the same number of cards from the card pile. Those skills could be seen as invisible, stack-freed cards.

The trigger system is a series of minor events to fulling describe a macro event. The so-called triggering chain in the *Legends of the Three Kingdoms* is that when the current process of the game meets certain conditions, a result will be triggered, and the new result may be another condition triggered by the result, thus forming a triggering chain. In other words, *trigger skills* are condition-oriented. If all the trigger conditions are extracted and treated as an event, this could constitute an event chain, and the relationship between the events can finally be perfectly handled after we put certain *trigger skills* on the corresponding event. In case one event corresponds to multiple skills, we use a priority value to distinguish the processing order.

With the major tasks determined by the phase layer, attention layer aimed at finding the best way to accomplish that task. If the major task is to kill the remaining Rebel on the ground, Attention layer is to use the combination of cards to achieve that.

4 CONCLUSION AND FUTURE WORK

In this paper, we designed a Hierarchical Macro Strategy model based on Tencent's work to solve *Legends of the Three Kingdoms*. We would also like to move forward and trying to realize that model to practical use and generalize the scenario to 5 or 6 player rounds. In that case, we need to consider not only player's interaction with Monarch, but interaction with each other as well.

ACKNOWLEDGMENTS

This work was funded by the Jiangsu Provincial Key R & D Program (Social Development) (Grant No. BE2016778), and Scientific Research Foundation of Nanjing University of Posts and Telecommunications (Grant No. NY217054).

REFERENCES

[1] Noam Brown and Tuomas Sandholm. 2017. Safe and nested endgame solving for imperfect-information games. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*.
[2] Andrew Gilpin. 2009. *Algorithms for abstracting and solving imperfect information games.* Ph.D. Dissertation. PhD thesis, Carnegie Mellon University.
[3] Johannes Heinrich and David Silver. 2016. Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121* (2016).
[4] Roger B Myerson. 2013. Game Theory: Analysis of Conflict. (2013).
[5] Tuomas Sandholm. 2010. The state of solving large incomplete-information games, and application to poker. *AI Magazine* 31, 4 (2010), 13–32.
[6] Bin Wu, Qiang Fu, Jing Liang, Peng Qu, Xiaoqian Li, Liang Wang, Wei Liu, Wei Yang, and Yongsheng Liu. 2018. Hierarchical macro strategy model for moba game ai. *arXiv preprint arXiv:1812.07887* (2018).