Mastering Terra Mystica: Super-human Strategy in AI

Abstract

In this paper, we explore and compare multiple algorithms for solving the complex strategy game of Terra Mystica, thereafter abbreviated as TM. Previous work in the area of super-human game-play using AI has proven effective, with recent break-throughs for generic algorithms in games such as Go, Chess, and Shogi [4]. We directly apply these breakthroughs to a novel state-representation of TM by combining multiple techniques to create an AI capable of rivaling human players. Specifically, we present initial results of applying AlphaZero to this state-representation, and analyze the strategies developed, their strengths and weakness. We call this modified algorithm with our novel state-representaiton AlphaTM. In the end, we discuss the success and short-comings of this method by comparing against multiple baselines and typical human scores.

1. Task Definition

Our proposal involves developing the infrastructure, framework, and models required to achieve super-human level game-play in the game of Terra Mystica (TM). The game of TM involves very-little luck, and is entirely based on strategy (similar to Chess, Go, and other games which have recently broken to novel Reinforcement Learning such Deep Q-Learning and Monte-Carlo Tree Search as a form of Policy Improvement [3] [5]). In fact, any random component to the game comes only from the initial configuration of the map and the initial reward structure, which is selected at random during typical game-play.

TM is a game played between 2-5 players. For our research, we focus mostly on the adversarial 2-player version of the game. We do this mostly for computational efficiency, though for each algorithm we present, we discuss briefly strategies for generalizing them to multiple players.

TM is a fully-deterministic game whose complexity arises from the large branching factor and a large number of possible actions A_t from a given state, S_t. There is further complexity caused by the way in which actions can interact, discussed further in Section 1.1.

1.1. Input/Output Of System

In order to understand the inputs and outputs of our system, the game of TM must be fully understood. We lay out the important aspects of a given state below, according to the standard rules [2]..

The game of TM consists of a terrain board that's split into 9×13 terrain tiles. The board is fixed, but each terrain tile can be terra-formed (changed) into any of the 7 distinct terrains (plus water, which cannot be modified). Players can only expand onto terrain which belongs to them. Furthermore, TM also has a mini-game which consists of a Cult-Board where individual players can choose to move up in each of the cult-tracks throughout game-play.

The initial state of the game consists of players selecting initial starting positions for their original dwellings.

At each timestep, the player has a certain amount of resources, consisting of Workers, Priests, Power, and Coin. The player also has an associated number of VPs.

Throughout the game, the goal of each player is to accumulate as many VPs as possible. The player with the highest number of VPs at the end of the game is the winner.

From the definition above, the main emphasis of our system is the task of taking a singe state representation S_t at a particular time-step, and outputting an action to take to continue gameplay for the current player. As such, the input of our system consists of the following information, fully representative of the state of the game:

1. Current terrain configuration. The terrain configuration consists of multiple pieces of information. For each terrain tile, we will receive as input:
   (a) The current color of the tile. This gives us information not only about which player currently controls the terrain, but also which terrains can be expanded into.
   (b) The current level of development for the terrain. For the state of development, we note that each terrain tile can be one of (1) UNDEVELOPED, (2) DWELLING, (3) TRADING_POST, (4) SANCTUARY, or (5) STRONGHOLD.
   (c) The current end-of-round bonus as well as future end-of-round bonus tiles.

    (d) Which special actions are currently available for use.

2. For each player, we also receive the following information.

    (a) Current level of shipping ability, Current level of spade ability, the current number of VPs that the player has.
    (b) The current number of towns the player has (as well as which town is owned), The current number of worker available to the player, the current number of coins available to the player, the current number of LV1, LV2, and LV3 power tokens.
    (c) The current number of priests available to the player.
    (d) Which bonus tiles are currently owned by the player.
    (e) The amount of income the player currently produces. This is simply the power, coins, priests, and worker income for the player.

The above is an incomplete enumeration of the input to our algorithm, and has not yet been fully formalized. However, in general, the input to the algorithm is a complete definition of the game state at a particular turn. Note that Terra Mystica *does not* have any dependencies in previous moves, and is completely Markovian. As such, modeling the game as an MDP is fully realizable, and is simply a question of incorporating all the features of the state.

1.2. Output and Evaluation Metrics

For a given state, the output of our algorithm will consist of an action which the player can take to continue to the next state of the game. Actions in TM are quite varied, and we do not fully enumerate them here. In general, however, there are eight possible actions:

1. Convert and build.
2. Advance on the shipping ability.
3. Advance on the spade ability.
4. Upgrade a building.
5. Sacrifice a priest and move up on the cult track.
6. Claim a special action from the board.
7. Some other special ability (varies by class)
8. Pass and end the current turn.

We will evaluate our agents using the standard simulator. The main metric for evaluation will be the maximum score achieved by our agent in self-play when winning, as well as the maximum score achieved against a set of human competitors.

2. Infrastructure

An open-source simulator for the game of TM already exists and is written in TM [1]. We've currently converted this open-source simulator into Python and can run locally, simulating gameplay. For our baselines, we've taken the built-in AI agents as baselines (which tend to perform relatively poorly). For our oracles, we've instead taken the approach of analyzing a large database of game data available online.

2.1. Model and Approaches

Developing an AI agent that can play well will be extremely challenging. Even current heuristic-based agents have difficulty scoring positions. The state-action space for TM is extremely large. Games typically have trees that are ¿50 moves deep (per player) and which have a branching factor of > 10.

We can approach this game as a typical minmax search-problem. Simple approaches would simply be depth-limited alpha-beta pruning similar to what we used in PacMan. These approaches can be tweaked for efficiency, and are essentially what the current AIs use.

Further improvement can be made to these approaches by attempting to improve on the $Eval$ functions.

However, the main contribution of this paper will be to apply more novel approaches to a custom state-space representation of the game. In fact, we will be attempting to apply Q-Learning – specifically DQNs (as per [3], [5], and [4]).

3. Baselines

For the base-lines, we have quite a few options. Using the simulator that's been converted into Python, we already have multiple AI agents that have been developed and can be played against. We can use these as the lowest bound for the baselines, since even the strongest-AI agents so far struggle to complete against average TM players. We summarize the results for these agents in Table 1.

4. Oracle

For an Oracle, we have two options. Even the current state-of the art AI in TM does not come close to beating average TM players. However, there is a large collection of online games that have been played by humans [6]. From these statistics, we can derive the following-upper bounds.

4.1. Competitive Player Oracle

Our oracle will therefore consists of the scores achieved by expert players. We've summarized this data in Table 2, broken down by faction. We look only at the two-player scenario, as this is the one on which our algorithm focuses.

5. Appendices

Average Human Score (2p)		
Faction	**Average Score**	**Sampled Games**
Darkling	137.10	2496
Swarmling	132.93	2302
Chaos Magician	130.06	2621
Halfing	133.32	2227
Auren	122.74	964
Cultists	123.25	702
Alchemist	127.48	1443
Witches	127.44	2124
Engineers	127.72	1543
Mermaids	125.11	1700
Giants	122.16	975
Nomads	121.15	2291
Dwarves	123.58	1054
Fakirs	119.16	629

Table 1. Average human scores by faction for a two-player TM games online.

Simulated Self-Play Average Scores		
Faction	**Average Score**	**Sampled Games**
Darkling	90.10	1000
Swarmling	93.12	1000
Chaos Magician	61.44	1000
Halfing	92.21	1000
Auren	91.74	1000
Cultists	93.11	1000
Alchemist	72.12	1000
Witches	79.89	1000
Engineers	77.12	1000
Mermaids	85.11	1000
Giants	85.11	1000
Nomads	99.9	1000
Dwarves	90.11	1000
Fakirs	82.19	1000

Table 2. Self-play easy AI agent: AI_Level5 from [7]

References

[1] jsnell. Open source terra mystica implementation, 2018. 2

[2] M. LLC. Terra mystica: Rule book, 2010. 1

[3] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484 EP –, Jan 2016. Article. 1, 2

[4] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. P. Lillicrap, K. Simonyan, and D. Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *CoRR*, abs/1712.01815, 2017. 1, 2

[5] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354 EP –, Oct 2017. Article. 1, 2

[6] Snellman. Terra mystica: Statistics, 2011. 2

[7] tmai. Terra mystic ai implementations, 2018. 3