Desert Fox: an AI for a Simple Wargame

Parker C. Michaelson

November 1, 2013

1 INTRODUCTION

The classic game of Chess is said to have originated in India sometime in the 6$^{\text{th}}$ century, and might be said to be a wargame due to its military theme and movement rules which superficially represent the strengths of the pieces. Chess is also a classic problem in AI, and more importantly for this project, has become something of a dead horse as far as basic research is concerned. Unless someone has a Chess-specific technique or is testing a new idea in AI, research on Chess is becoming less and less relevant to the field. Additionally, the AI effect continues to cause Chess-related AI research to have a diminishing influence on the public conciousness. This is bad not only for the reputation of the field of Artificial Intelligence altogether, but is also a negative influence of funding for further AI research.

This project, called Desert Fox, seeks to remedy this situation by applying AI techniques such as the Monte Carlo Tree Search and Reinforcement Learning to the problem of winning a simple wargame. This wargame is based on the commercial wargame Memoir '44, albeit this version is greatly simplified to expedite the process on the basic problem of implementing an AI for the game, as well as to reduce the solution space to a reasonable size given the resource constraints of this project.

2 BACKGROUND

2.1 Memoir '44

Memoir '44 is a turn-based tabletop wargame set at a tactical scale where action takes place on a hexagonal playing board, the map, and features combat between Allied and Axis forces in the Second World War. Individual units of infantry within the game are roughly representative of a single squad of soldiers (8-13 men) or a platoon strength unit (26-64 men). The game's map covers only a small area of battlefield, just large enough to show, for instance, the landing at Pegasus at a squad scale, or the Pointe du Hoc landing at platoon scale. The map might also be covered with tiles representing terrain, each with rules regarding movement and line of sight for units moving about the map, the simplified game features only three terrain types, ordinary terrain, forest and hill.

The map is divided into three sections or flanks, the left, center and right flanks. These sections are used by "command cards", cards drawn at random from a deck by the player, which enable orders to be issued to between one and three units on a section, on between one and all of the sections. The command cards also come in varieties which are independent of section, or which associate special rules with units ordered in that turn. Almost all of these "special" command cards will be ignored in the Desert Fox version of Memoir '44. Ordering a unit consists of moving a unit any distance, and then optionally ordering it to fight an enemy unit, given that the enemy unit is within range and line-of-sight exists. Combat between units is decided by using a set of dice to determine casualty types (infantry, armor, wildcard or retreat), and then by applying relevant rules such as range modifiers, terrain modifiers and modifiers which may be in place due to a command card.

The commercial version of the game features three distinct unit types (infantry, armor and artillery), along with special forces or resistance versions of infantry and armor units, while the simplified version used for Desert Fox has only infantry. The simplified version also features only the sectional command cards, which take the form "Order (1 — 2 — 3) units on (the left — the center — the right — the left and right — all) sections." and the "Direct from HQ" command card, allowing any four units to be ordered.

The goal of Memoir '44 is to score a pre-determined number of "medals", gained either by killing an enemy unit or by taking an objective tile and holding it. The simplified version of the game only counts kills as medals. The goal of the Desert Fox AI is to attempt to gain its medal count before the enemy player, thus winning the game.

2.2 Monte Carlo Tree Search [1]

The Monty Carlo Tree Search (MCTS) is a technique for quickly searching the successors of a state in a search tree and determining the optimal action. The basic principle of the MCTS is the use of stochastic or "Monte Carlo" methods to create a random sampling of possible successor states in a search tree based on a *tree policy*. MCTS then expands nodes in the sample and uses a *default policy* to reach the bottom of the search tree from the children of the sample nodes. These paths to the bottom are used to compute an expected reward for each sample node, this reward is back-propagated through the tree to the root node, updating average reward values on the way. After this, a number of heuristics can be used for determining the best path to follow, including picking the node with maximum reward or the node which has the best lower bound on losses.

Desert Fox is expected to use this technique to find the best move for a single turn.

2.3 High-Level Reinforcement Learning [2]

Reinforcement Learning is a machine learning technique combining the representational power of Markov Decision Trees with one of several methods allowing Markov Trees to be updated to match observations, *Q-learning* and *Model-based Q-learning* being mentioned by Amato and Shanai. A Markov Decision Tree is a tree where each node represents some state of the system, where each node has a set of actions as edges and table of probabilities for reaching successor states as a result of an action. After taking an action and reaching a successor, some reward, not necessarily positive, is given based on the successor reached. Given the tree, it is possible to find the *policy* which maximizes the expected value of the action taken, as given by the Bellman Equation.

The learning method is an algorithm used to update the Markov Trees as experience with a problem is gained. Q-learning updates the Markov Tree in real time, and strengthens the reward given by taking some action based on the reward resulting from the successor state of the action.

Desert Fox is expected to use this technique to switch between offensive and defensive strategies, each of which adjusts the rewards given for states in the Monte Carlo Tree Search. This controls the strategy used in a single turn.

3 IMPLEMENTATION

The project is to proceed as follows:

Initial Planning and Proposal In this phase of the project, the problem to be solved by Desert Fox will be determined and Desert Fox will enter the first stages of planning. After this, a proposal for the project will be drafted and submitted for review.

Problem Implementation This phase of the project calls for the implementation of the simplified Memoir '44, and the creation of a human interface for the game as well as an API for Desert Fox.

AI Implementation At this point, Desert Fox itself will be implemented using the API for the game.

AI Training and Fine Tuning In this phase, the AI will be trained against a human opponent, and fine tuning will be made to optimize performance.

Data Collection and Consolidation With the completion of Desert Fox, performance data can be collected, analysed and then consolidated into poster form for presentation.

Final Submission At the completion of the project, all code, resources, data and written material will be organized and submitted for final judgement.

4 CONCLUSION

Concluding, the project is accomplish the following objectives prior to its termination:

- A version of Desert Fox capable of beating a human opponent in simplified Memoir '44.
- A collection of data regarding performance improvement as training is given to Desert Fox.
- A report explaining the workings of Desert Fox and the relevance of Monty Carlo Tree Search and High-Level Reinforcement to its success.

References

[1] High-level Reinforcement Learning in Strategy Games. Christopher Amato and Guy Shanai. *Proc. of 9^{th} Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010).*

[2] A Survey of Mente Carlo Tree Search Methods. Browne, Powley, *et. al.*. *IEEE Transactions of Computational Intelligence and AI in Games, Vol. 4, No. 1, MARCH 2012.*