**CITS3001 Algorithms, Agents & Artificial Intelligence**

**Project – Love Letter**

Jaydeep Gajera – 22228003

Parthvi Sheladia – 22241757

**Literature Review**

Love Letter is a game of risk, deduction, and luck and can be played between 2–4 players. A player’s goal is to get their love letter to the Princess. Two ways through which the letter can be delivered are: giving the letter to an intermediary or directly to the princess. From a deck of sixteen cards, each player starts with only one card in hand; with each card representing an intermediary. Each intermediary corresponds to an action that can affect all players in the game in different ways. Actions consists of swapping cards with another player to eliminating a player completely from the round. The game comprises of 13 rounds and each player draws a card on their turn leading to make a choice between two cards depending on what action increases their chance of winning the round and essentially the game. Powerful cards lead to early gains, but make you a target. A player can rely on weaker cards, however, their letter may be tossed in the fire!(intext reference) (mention something about incomplete information..). The game is full of uncertainty and no player have a clue of other player’s cards, hence can be referred to as a imperfect information game.

Monte Carlo Tree Search(MCTS) is a decision making algorithm that helps deal with such games. It is probabilistic and heuristic driven search which works even better when combined with other useful implementation like the classic tree search implementation. It uses the idea of building a game tree with each node representing a game state that potentially helps with making optimal decisions. The algorithm periodically exploits the best action and strategy found while also continuing to explore other alternative to see if they could replace the current best. The nodes as previously mentioned are a result of a number of simulations. The whole process can be broken down into four distinct steps: selection, expansion, simulation and backpropagation. The MCTS algorithm relies on two fundamental concepts: • The expected reward of an action can be estimated doing many random simulations. • These rewards can be used to adjust the search toward a best-first strategy. (intext)

Perfect Information Monte Carlo Search with the Determinized Upper Confidence Bounds Applied to Trees is based on the idea to build a tree in an incremental and asymmetric manner by doing many random simulated games. It’s main purpose is to find a balance between the exploration of the less frequently simulated nodes and the exploitation of the already chosen best ones. To explain the whole process in terms of the four steps of MCTS, once a node is selected, expansion occurs by choosing an available move and adding it to the tree. Once that is done, by running a simulation on the child node and backpropagating, a result is achieved. A simulation is based on a default policy in the algorithm and the result is calculated using the UCB formula. By continuously repeating these steps, the result values are updated. This algorithm is beneficial as no prior knowledge is required for the agent(knowledge of game rules is enough). Along with this, a determinization is a conversion of a stochastic game with imperfect information to a deterministic game with perfect information, in which the hidden information and the outcomes of all future chance events are fixed and known.

Another useful algorithm that can deal with imperfect information games is an extension of MCTS, Information Set MCTS(ISMCTS). In this algorithm, the node is now represented as the information set rather than the state. The information set is from the root player’s point of view and corresponds to moves player by the corresponding player.

**Selected Techniques**

* **Knowledge Based Algorithm**
* **Monte Carlo Tree Search (MCTS) Algorithm**

**Implementation**

**Validation**

Firstly, to test the performance of our knowledge based agent, we ran a tournament between three RandomAgents and our knowledge based agent. The tournament included 100,000 games of Love Letter with knowledge based agent playing from different position in each tournament.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Pos 1 Wins | Pos 1 Win Rate | Pos 2 Wins | Pos 2 Win Rate | Pos 3 Wins | Pos 3 Win Rate(%) | Pos 4 Wins | Pos 4 Win rate(%) |
| KBAgent | 50660 | 50.660004 | 49260 | 49.26 | 57159 | 57.159 | 51182 | 51.182003 |

**References**

* Brown, J. (2018). *Monte Carlo Tree Search for Love Letter*. [online] ResearchGate. Available at: <https://www.researchgate.net/publication/327679828_Monte_Carlo_Tree_Search_for_Love_Letter> [Accessed 16 October 2019].
* GitHub. (2018). *eugenp/tutorials*. [online] Available at: <https://github.com/eugenp/tutorials/tree/master/algorithms-miscellaneous-1/src/main/java/com/baeldung/algorithms/mcts/montecarlo> [Accessed 18 October 2019].
* GitHub. (2019). *eugenp/tutorials*. [online] Available at: <https://github.com/eugenp/tutorials/tree/master/algorithms-miscellaneous-1/src/main/java/com/baeldung/algorithms/mcts/tree> [Accessed 18 October 2019].
* Lanzi, P. and Palma, S. (2014). *Monte Carlo Tree Search algorithms applied to the card game Scopone*. [ebook] The University of Western Australia. Available at: <http://teaching.csse.uwa.edu.au/units/CITS3001/project/2017/paper1.pdf> [Accessed 15 October 2019].
* Mańdziuk, J. (2018). *MCTS/UCT in solving real-life problems*. [online] ResearchGate. Available at: <https://www.researchgate.net/publication/320003615_MCTSUCT_in_solving_real-life_problems> [Accessed 12 October 2019].
* Monte Carlo Tree Search. (2016). [ebook] Swarthmore College. Available at: <https://www.cs.swarthmore.edu/~bryce/cs63/s16/slides/2-15_MCTS.pdf> [Accessed 17 October 2019].
* Baeldung. (2019). *Monte Carlo Tree Search for Tic-Tac-Toe Game | Baeldung*. [online] Available at: <https://www.baeldung.com/java-monte-carlo-tree-search> [Accessed 18 October 2019].
* Roy, R. (2019). *ML | Monte Carlo Tree Search (MCTS) - GeeksforGeeks*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/ml-monte-carlo-tree-search-mcts/> [Accessed 16 Oct. 2019].