Wargaming with Monte-Carlo Tree Search

Presentation for the 14th NATO ORA Conference - Erik Kalmér and Christoffer Limér

Project background

- Started as our Bachelor’s degree project at The Royal Institute of Technology (KTH) in collaboration with the Swedish Defence Research Agency (FOI)

- Within the context of artificial intelligence, there was an interest in applications within military strategy.

- Specifically, wargaming was chosen for exploration.

Strategy games as training and evaluation of military strategies

- Even the ancient Greeks played strategy board games in the fifth-century B.C

- The Germans (Prussians) further developed the game of Chess into more realistic strategy games that were used heavily during 19^{th} century.

- *"It's not a game at all! It's training for war. I shall recommend it enthusiastically to the whole army."* - General Karl von Müffling of Prussia (1824)

Prussian officers playing Kriegsspiel (illustr. August 1872).

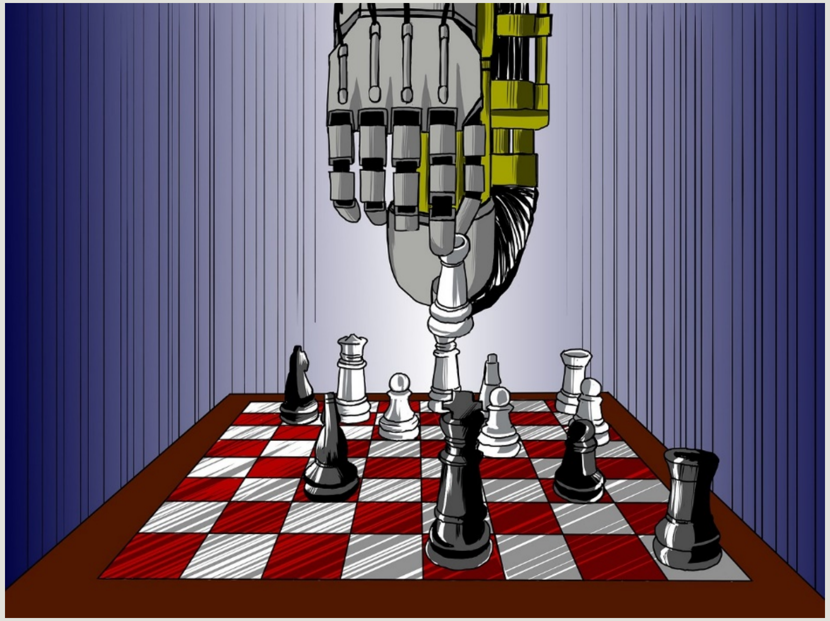
Strategy games as training and evaluation of military strategies

- Many aspects of wargaming have stayed the same.
- Wargaming is still relevant today, could modern AI techniques enhance this further?
- Could AI-agents be developed to give decision support for Course of Action (COA) within wargaming?

A wargame at the US Marine Corps War College (April 2019)

Artificial intelligence for strategy games

- In the early 50s when artificial intelligence was a somewhat new field, attempts were being made to get computers to play chess.
- In 1997 the Chess program Deep Blue officially beat the world champion, Garry Kasparov
- Since then, scientists have been looking for new challenges



Artificial intelligence for strategy games

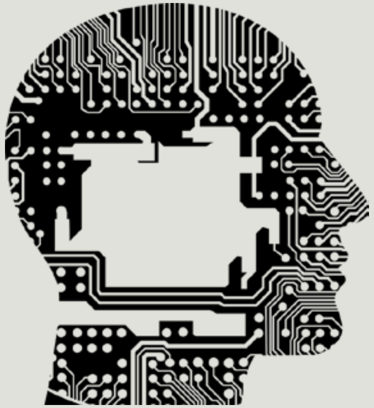
- In March of 2016, the program AlphaGo defeats world champion Lee Sedol in the game of Go
- This was unexpected, some believed such a program would take at least another decade to develop
- A vital part of the AlphaGo and its successors is its use of an effective search algorithm called Monte Carlo Tree Search (MCTS)
- Could MCTS be used to play a less abstract strategic board game?

The AlphaGo logo

A Go game in progress

Our research project in one sentence:

Can we create a program that can play the strategic board game Risk at a high skill level, using Monte Carlo Tree Search?

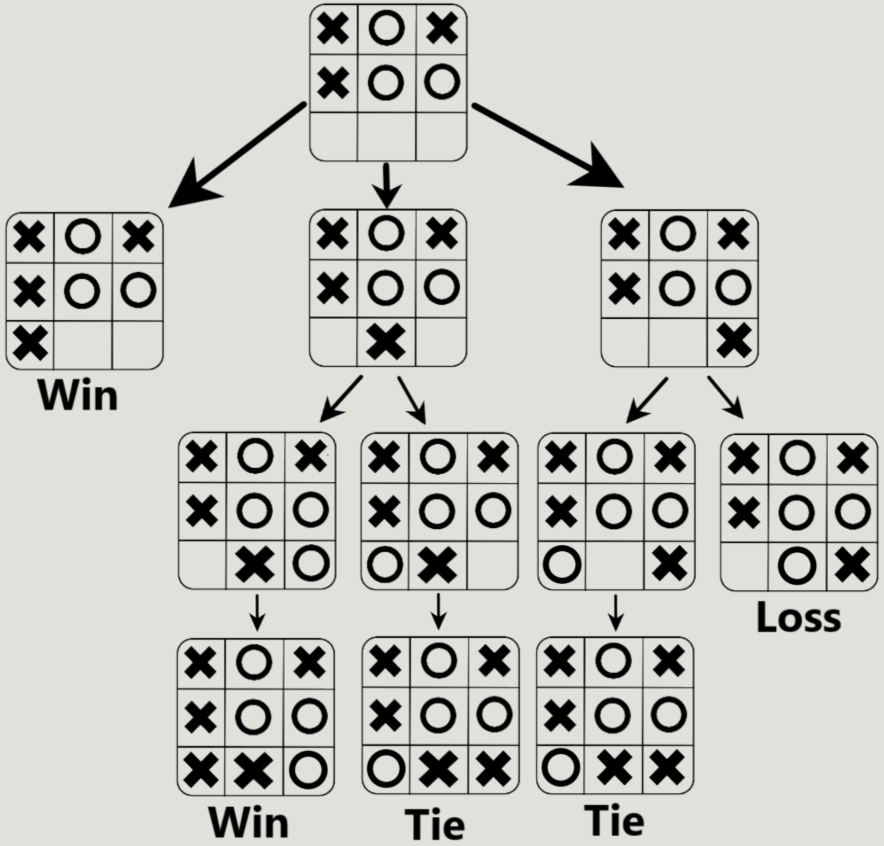


- A commercial strategic board game that is less abstract then Chess but with reasonable complexity

- Our first step towards more complex strategy environments

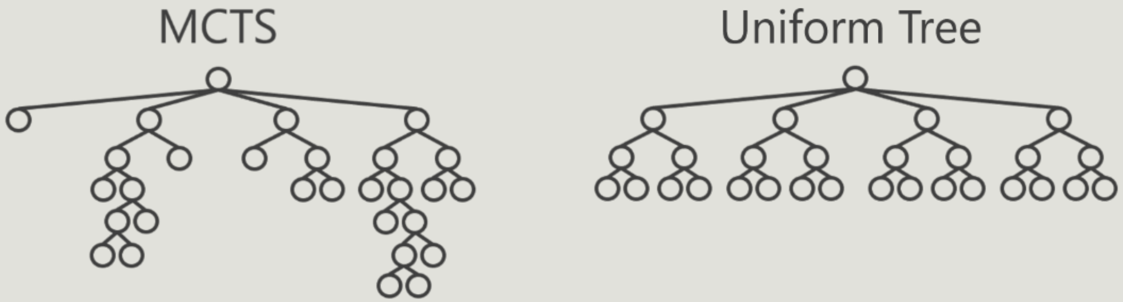
What makes the Monte Carlo Tree Search effective?

- If we use an easy example, Tic-tac-toe



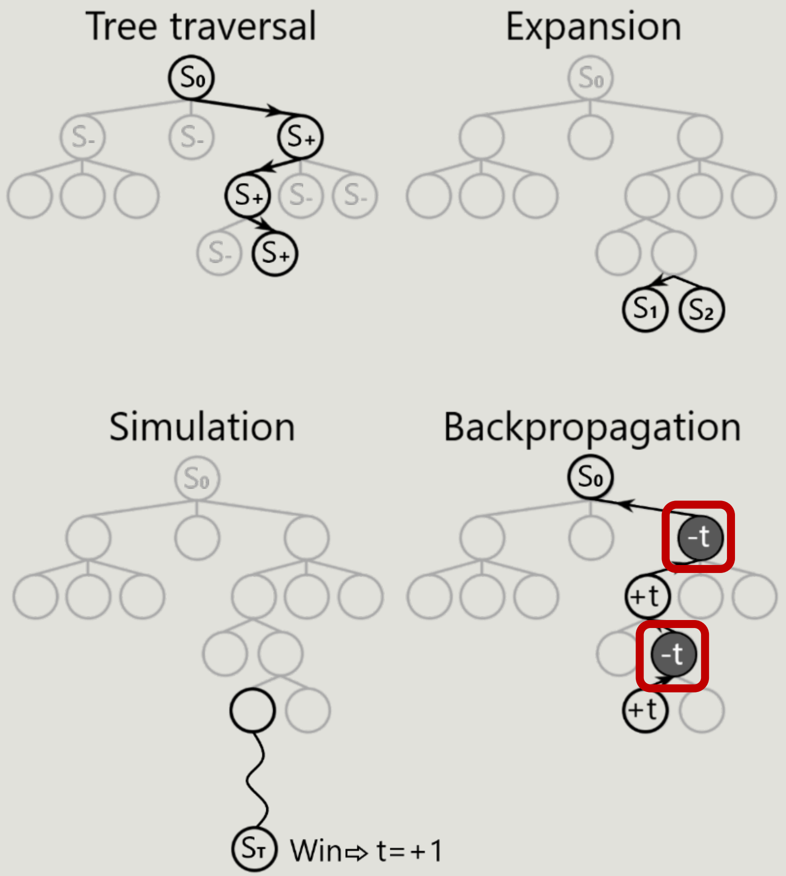
Example of a tree structure for Tic-tac-toe, played by X

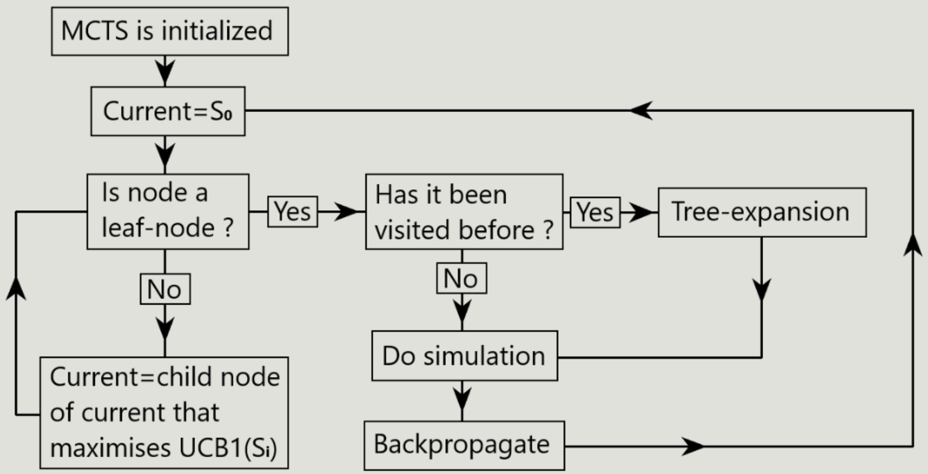
- Monte Carlo Tree Search or Uniform Tree Search



How the MCTS algorithm works at a glance

- The four phases of MCTS





Flowchart for MCTS

Looking under the hood of MCTS using UCB1

- The UCB1 formula is used by MCTS to select the next move/node in the traversal phase

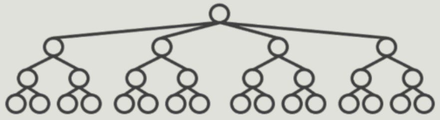
$$UCB1_i = \boxed{\overline{V}_i} + \boxed{C * \sqrt{\frac{\ln N}{n_i}}}$$

Where:

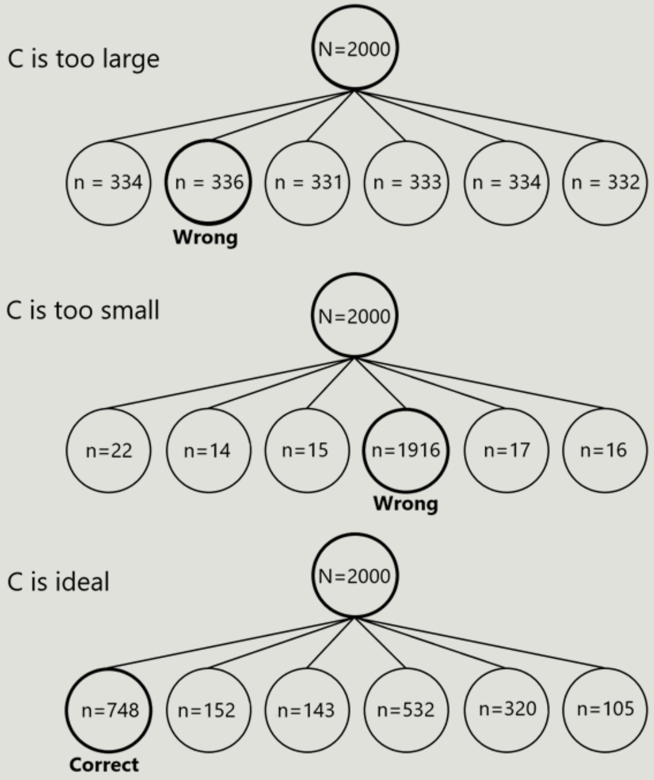
- $\overline{V}_i$ = Empirical mean-valuation (t_i/n_i)
- n_i = Number of simulations for node$_i$
- t_i = Sum of all valuations of node$_i$
- N = Total number of simulations for parent-node
- C = Constant (exploration parameter)

MCTS

Uniform Tree

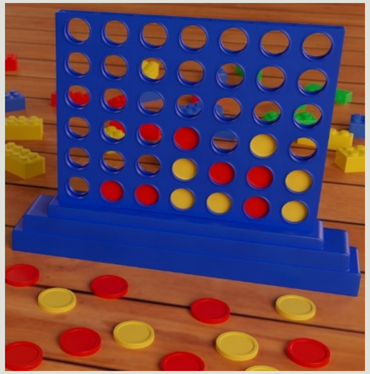


- The constant "C" is used to balance the exploration-exploitation trade-off, and it is essential to choose wisely

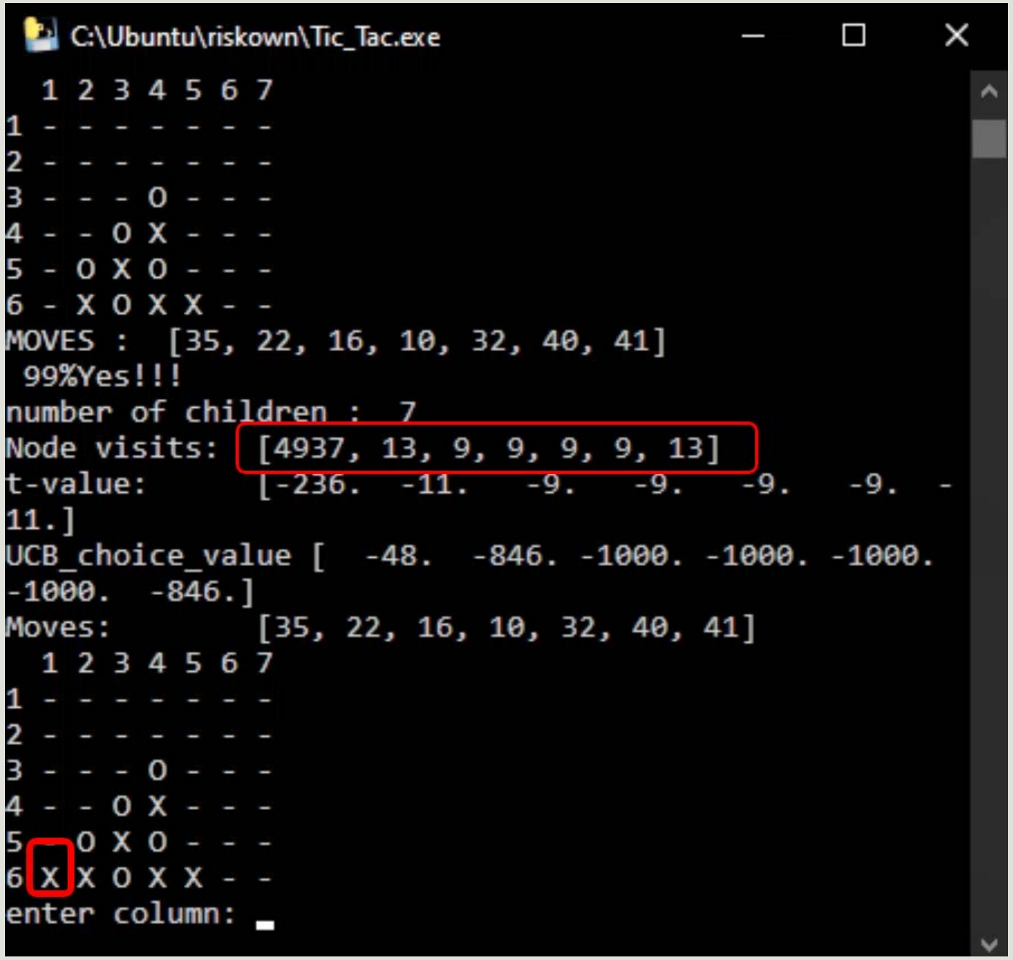


Our first implementation with Connect 4

- We used the programming language Python to build our client from scratch
- An initial program was made to serve as a trial for our MCTS algorithm. The game of choice was Connect 4
- It was important to get a successful MCTS-agent on a simple game first, before moving on to a more complex game

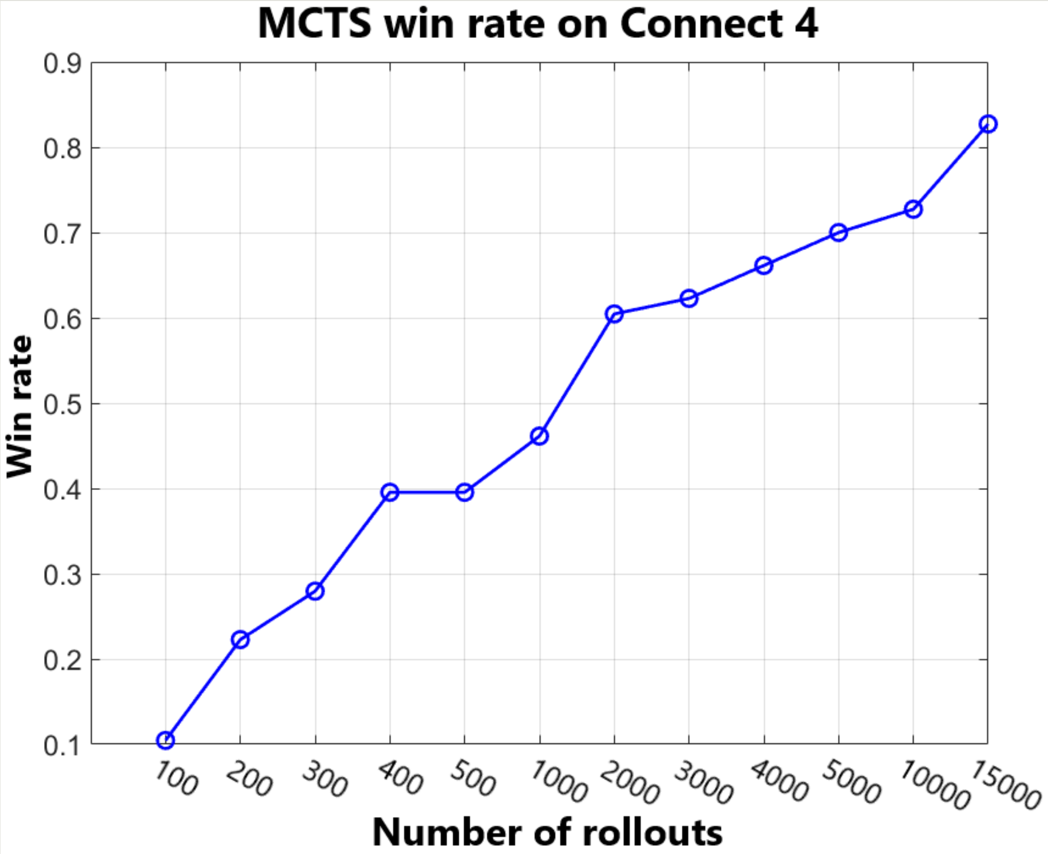


The game Connect 4



Our first implementation with Connect 4

- The term **Rollout** means the number of evaluated moves/nodes
- If the MCTS-agent works correctly, there should be a correlation between the number of rollouts and performance
- In the implementation of Connect 4, we see a distinct correlation between the win-rate and the number of rollouts
- The agent with 100 rollouts wins only about 10% of its matches, while the agent with 15000 wins almost 83%



MCTS for Risk

Risk has some challenges that must be solved when implementing it on the MCTS.

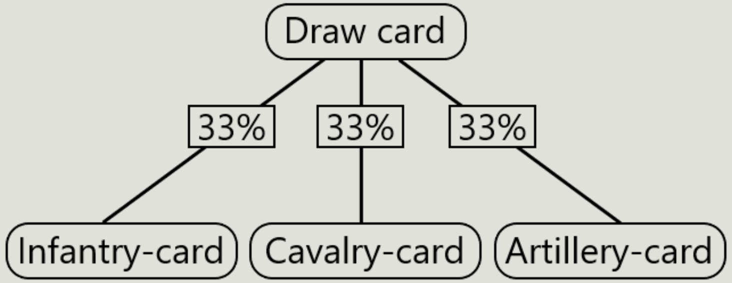
- For example, it is not a game with perfect information such as chess or connect-4 when army cards are kept secret by the opponent.

- There are some elements of uncertainty that our first MCTS-algorithm can’t handle.

- To handle these and more, we have used various techniques to facilitate and improve our implementation.

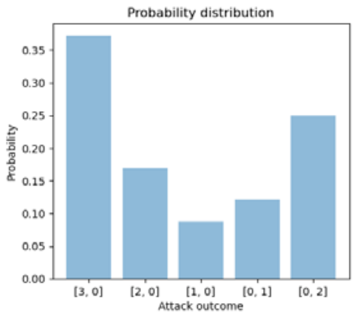
Chance-nodes

Cards

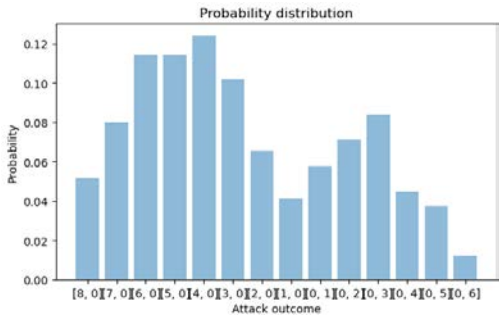


Attack

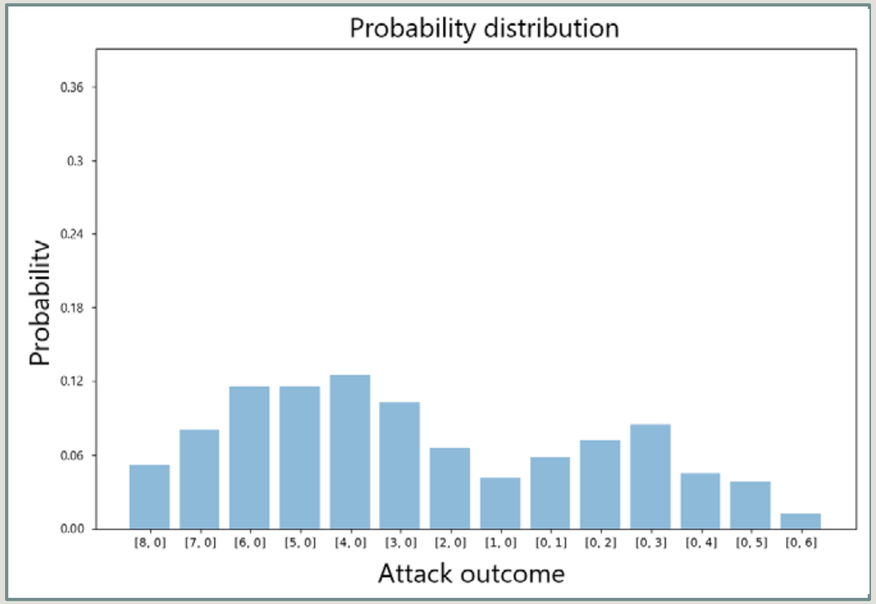
Small attack
(3 vs 2)

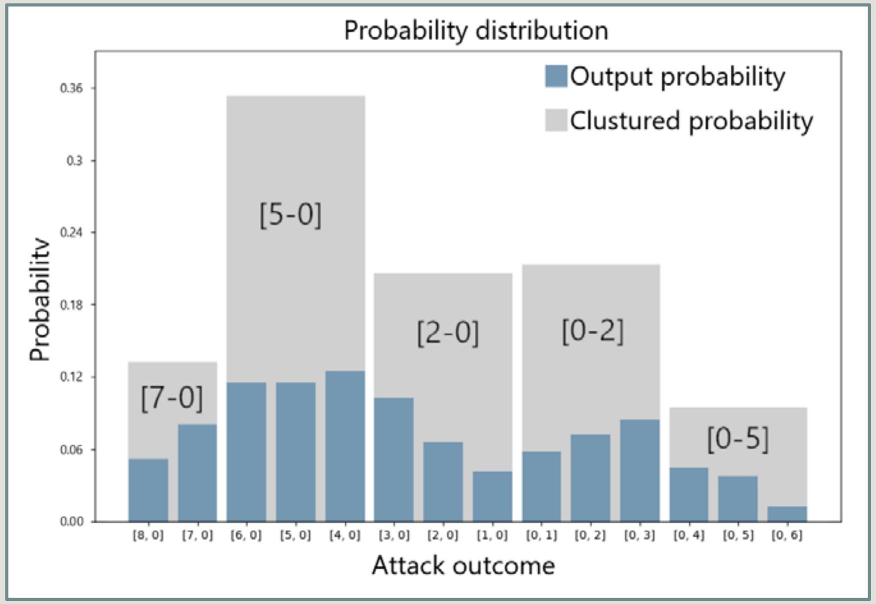


Large attack
(8 vs 6)

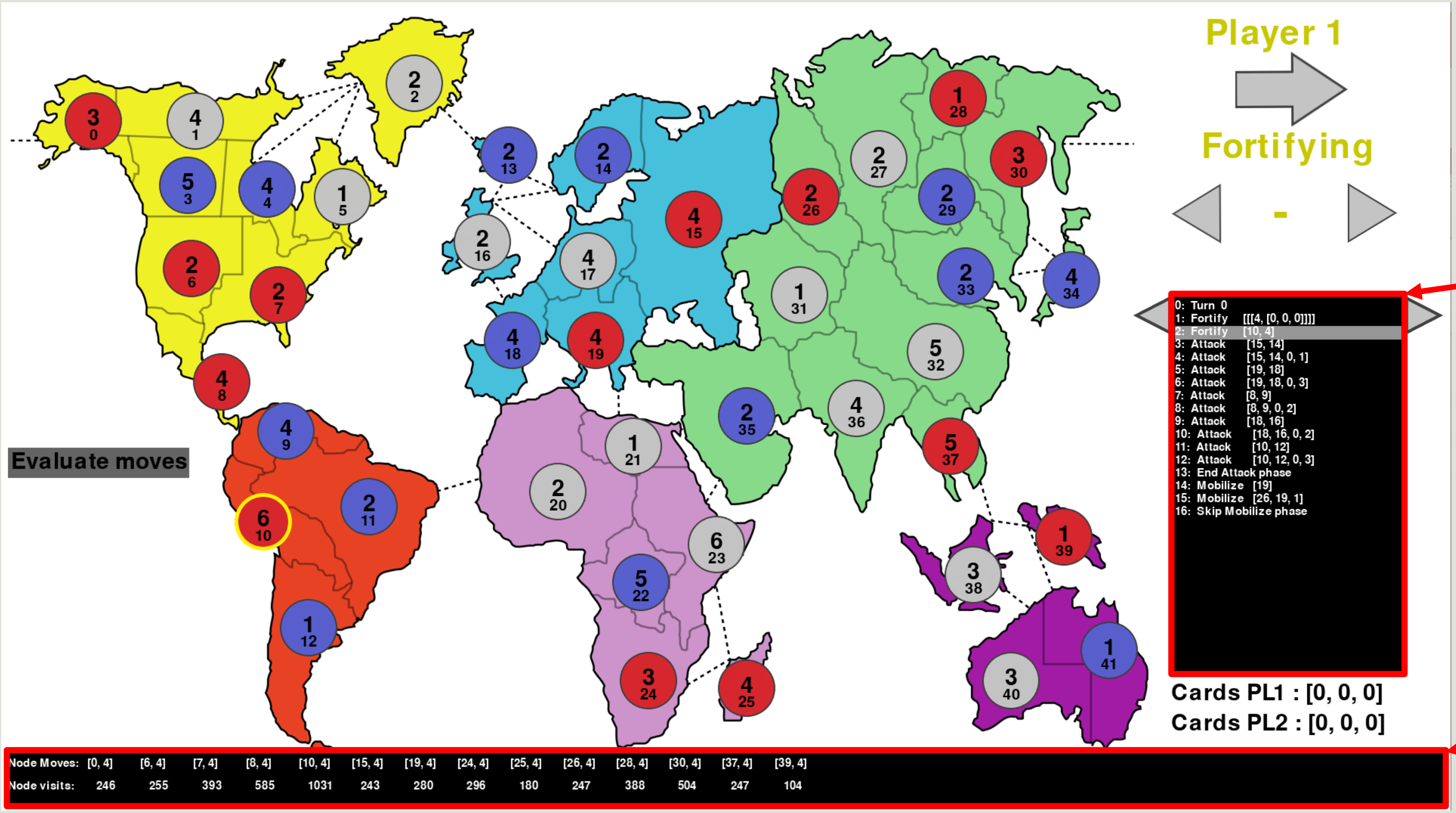


Chance-node Clustering





User Interface (UI)

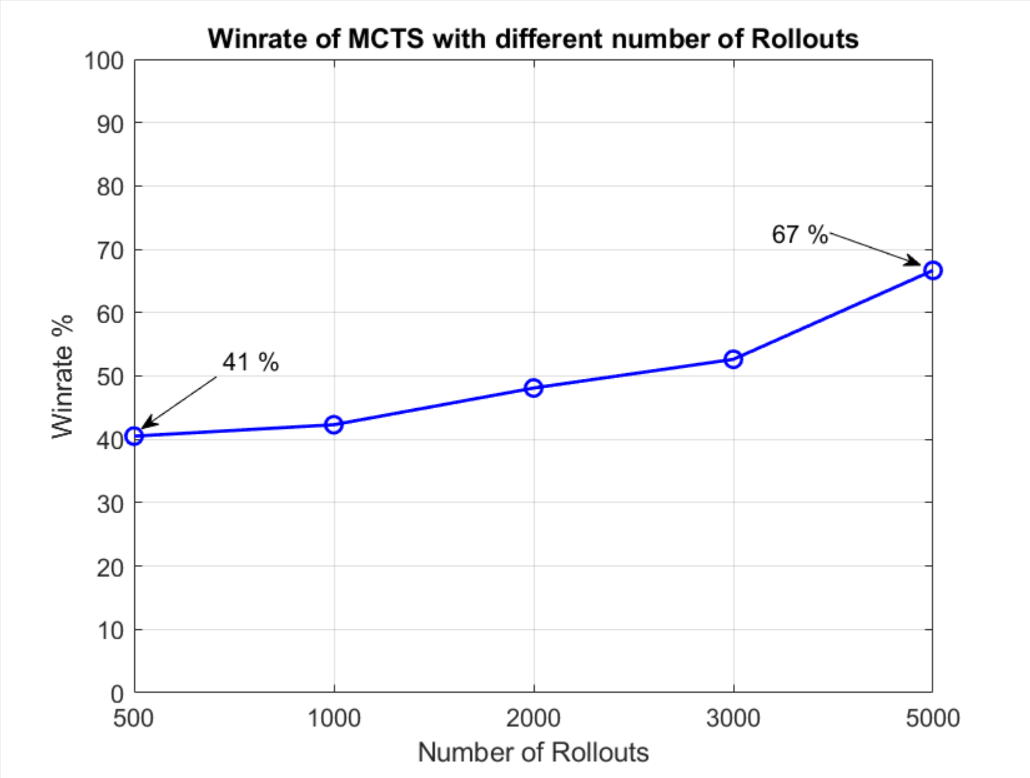


History list of previous Actions

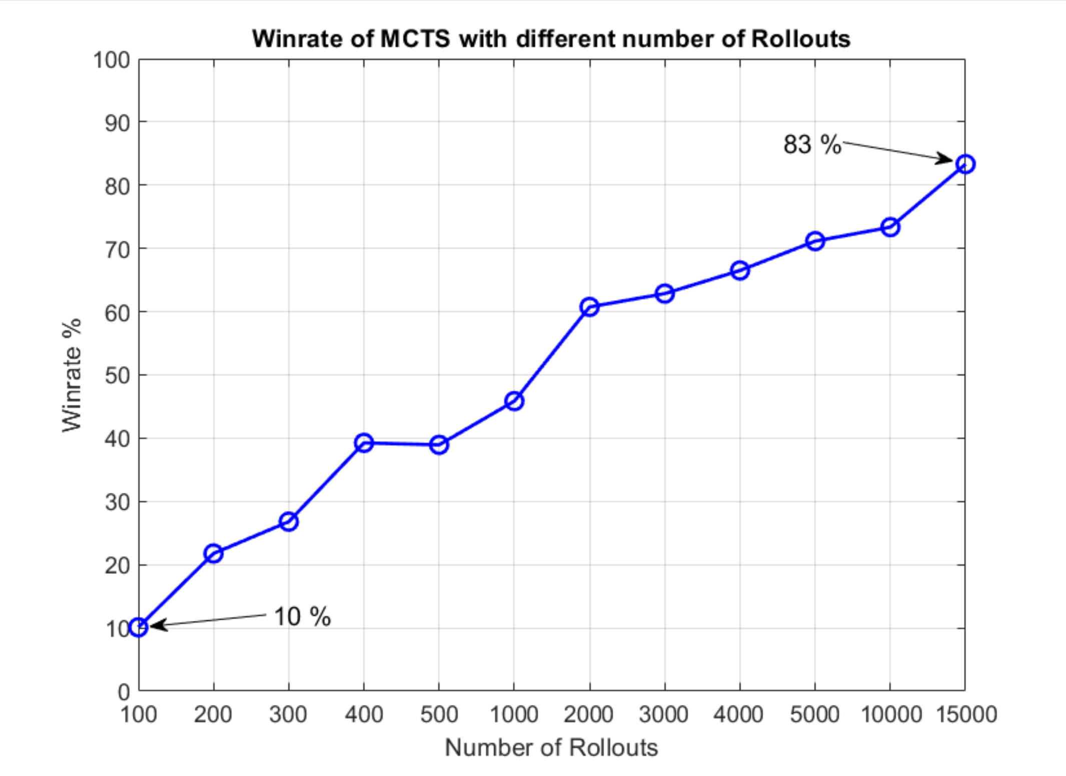
The MCTS evaluation of all other possible actions

Testing the MCTS

Risk

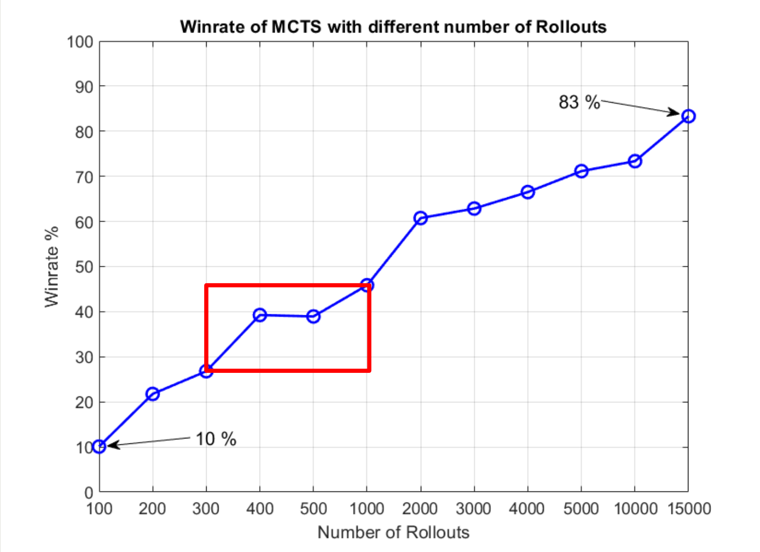


Connect-4



How does the MCTS stack up against human players?

Connect-4



	Rollouts
Casual players	300-1000

Risk

MCTS-agents with Cut-off 10	Win rate against human players	
500 Rollouts	10 %	Almost no wins
5 000 Rollouts	40 %	Under half
15 000 Rollouts	60 %	Just over half

Despite the game of Risk having a large amount of luck, which we first thought would disrupt the convergence, our MCTS has shown itself capable of playing at a human level, possibly even outperforming us.

Applications in wargaming

Lessons learned when working with Risk

What are the limitations?

- Large sets of possible actons may need ”action pruning”
- Elements of chance/uncertanty prohibit the MCTS from converging

How much work is needed for an implementation?

- Relatively little work and time required to implement on new environments

Applications in wargaming

Possibilities for decision support

Synthetic players in educational wargaming

- Relieve people from roles and tasks in wargaming scenarios. (Collaborator or/and Enemy) [Instead of a whole team of people playing different roles, the AI can play those roles with an equal skill/performance]
- Train people in different scenarios and learn from possible mistakes.

Evaluations in analytical wargaming

- Evaluate action alternatives
- Scenario analysis

For further interest please read our paper or contact us: limer@kth.se or ekalmer@kth.se

Thanks for listening!