

# Monte Carlo Search Tree and Its Applications

Max Magnuson

Senior Seminar  
Division of Science and Mathematics  
University of Minnesota, Morris  
Morris, Minnesota, USA

April 25, 2015

# Kasparov vs Deep Blue



# Kasparov vs Deep Blue

Great display of artificial intelligence (AI)

Techniques employed by IBM

- ▶ Brute force deterministic approach
- ▶ Human knowledge

Limitation

- ▶ Scalability into larger search spaces

Monte Carlo tree search (MCTS) is an alternative method

# Outline

Introduction

Naive MCTS Implementation

Applying MCTS to Go

Applying MCTS to Narrative Generation

Conclusion

# Monte Carlo Tree Search (MCTS)

- ▶ Combines random sampling and game trees
- ▶ Lightweight random simulations
- ▶ Probabilistic not deterministic
- ▶ Useful for problems with larger search spaces

# Applying MCTS to Go

## Go

- ▶ Board game about positional advantage
- ▶ Game board for Chess:
  - ▶ 8x8
- ▶ Average possible configurations for a game of Chess:
  - ▶  $10^{120}$
- ▶ Game board for Go:
  - ▶ 19x19
- ▶ Average possible configurations for a game of Go:
  - ▶  $10^{761}$

# Applying MCTS to Narrative Generation

- ▶ Useful Applications
  - ▶ Video game replay value
  - ▶ Educational applications
- ▶ The search space scales with the number of characters, items, locations, and actions

# Outline

Introduction

**Naive MCTS Implementation**

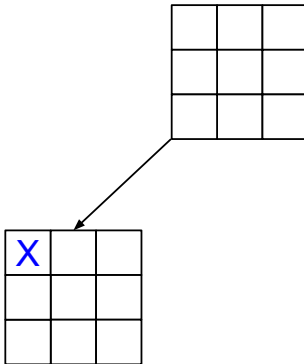
Applying MCTS to Go

Applying MCTS to Narrative Generation

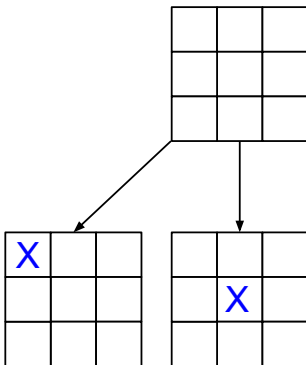
Conclusion



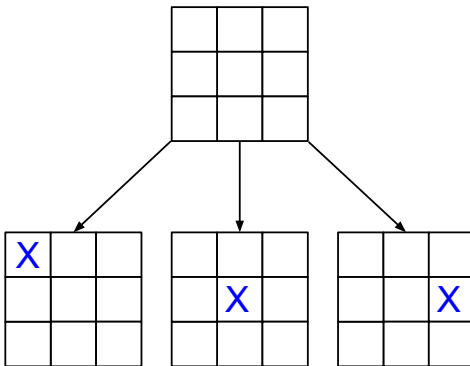
# TicTacToe Diagram



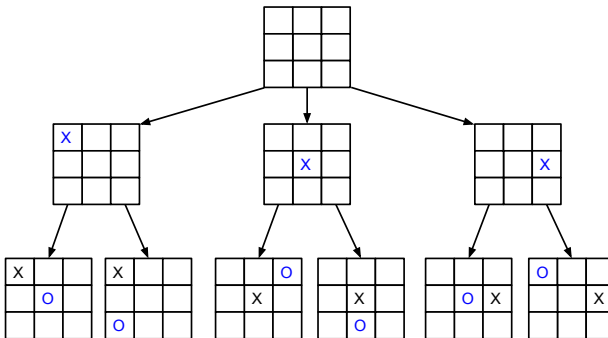
# TicTacToe Diagram



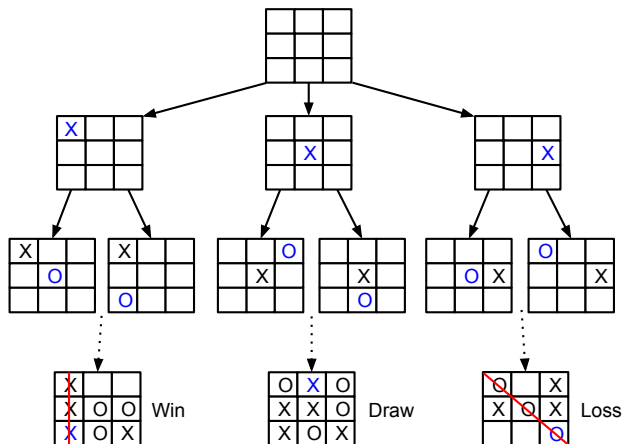
# TicTacToe Diagram



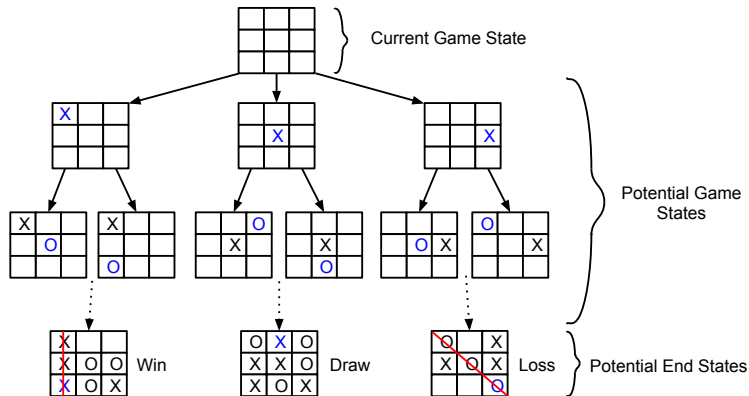
# TicTacToe Diagram More Levels



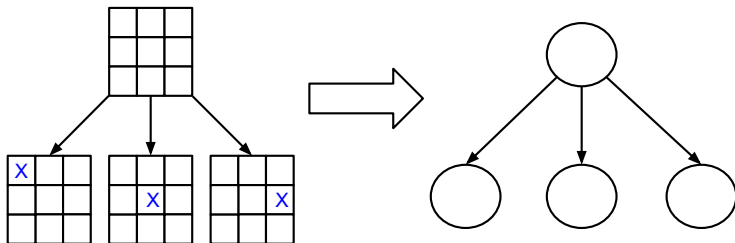
# TicTacToe Diagram



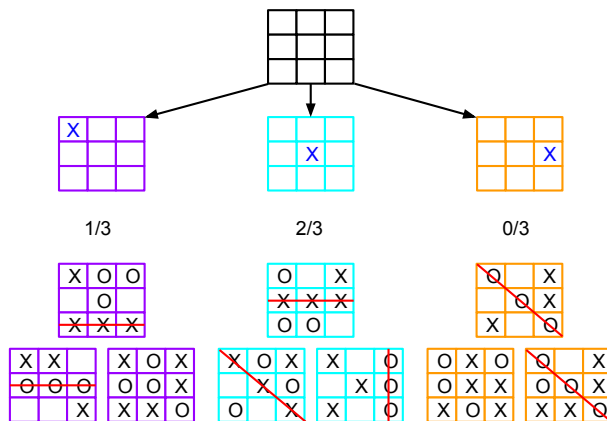
# TicTacToe Diagram



# Tree Structure

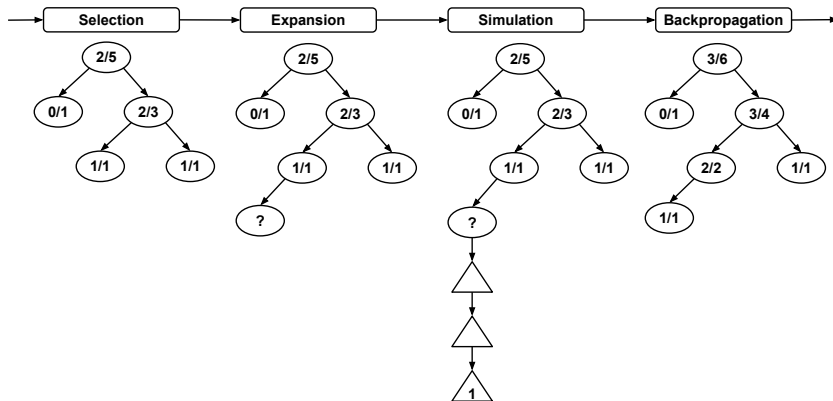


# Sampling

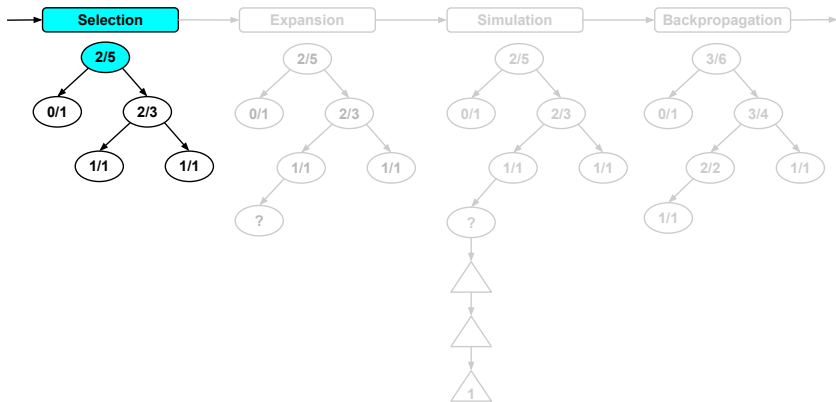




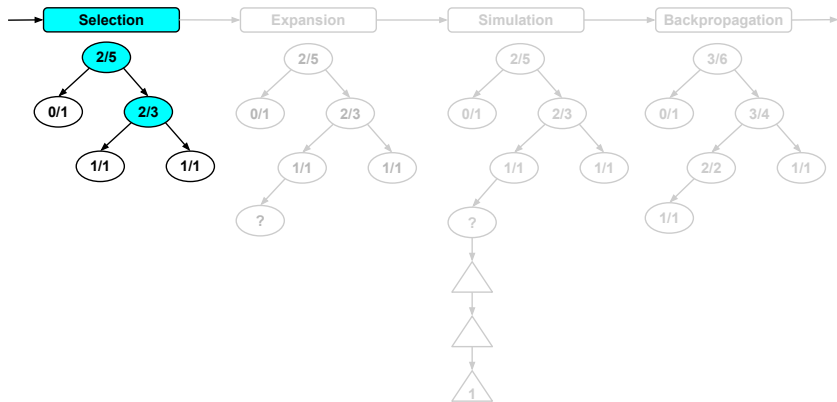
# Four Steps Diagram



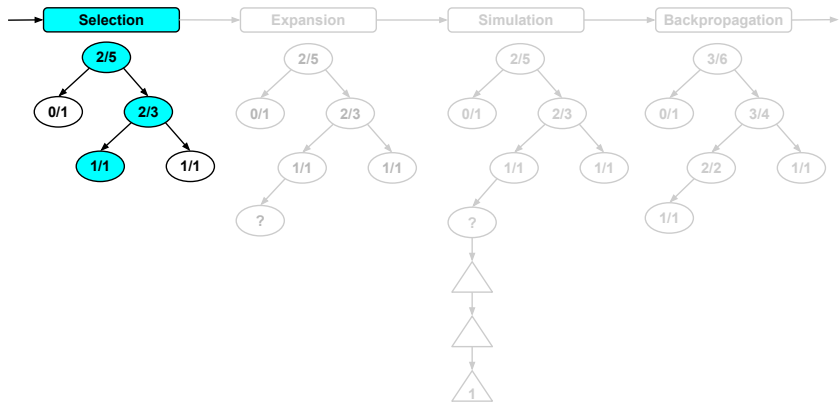
# Four Steps Diagram



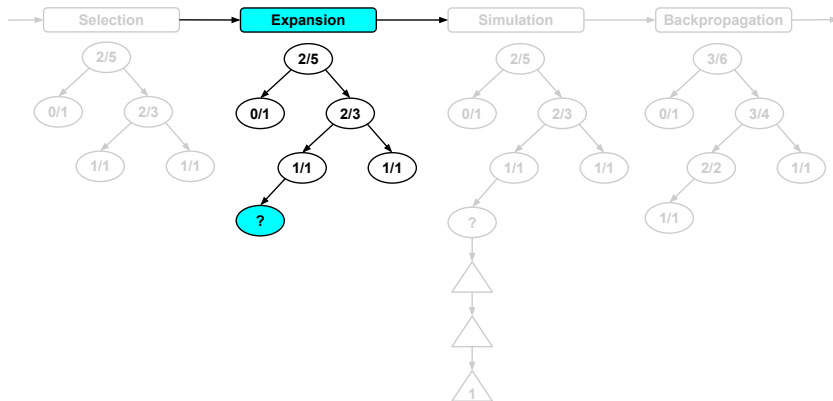
# Four Steps Diagram



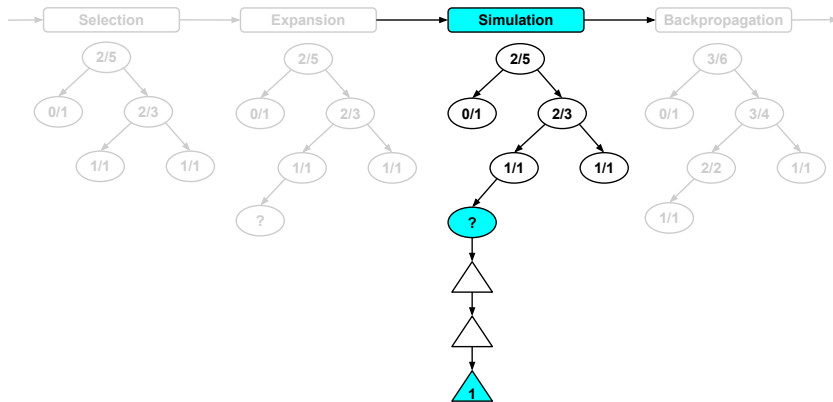
# Four Steps Diagram



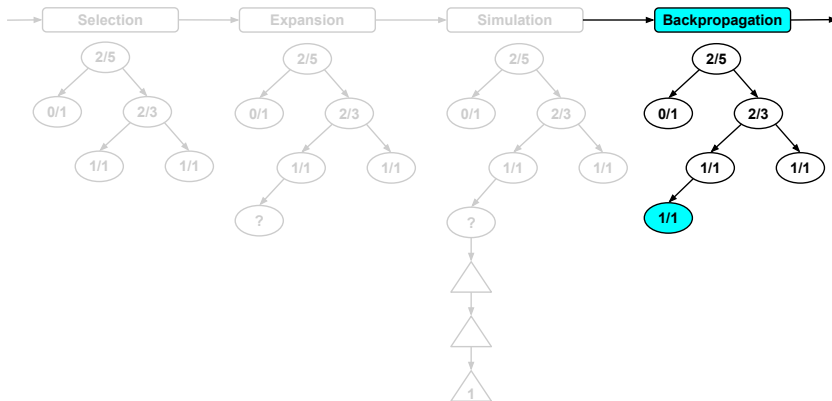
# Four Steps Diagram



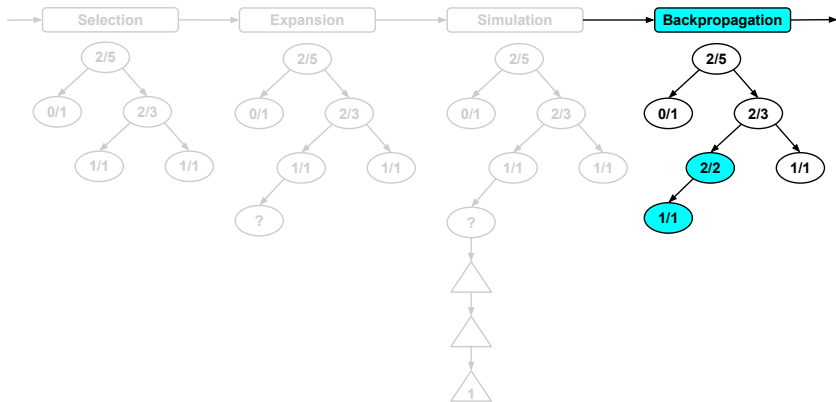
# Four Steps Diagram



# Four Steps Diagram

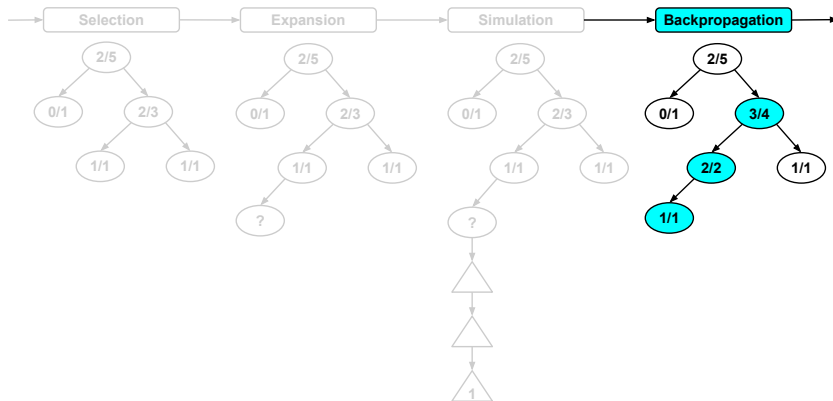


# Four Steps Diagram

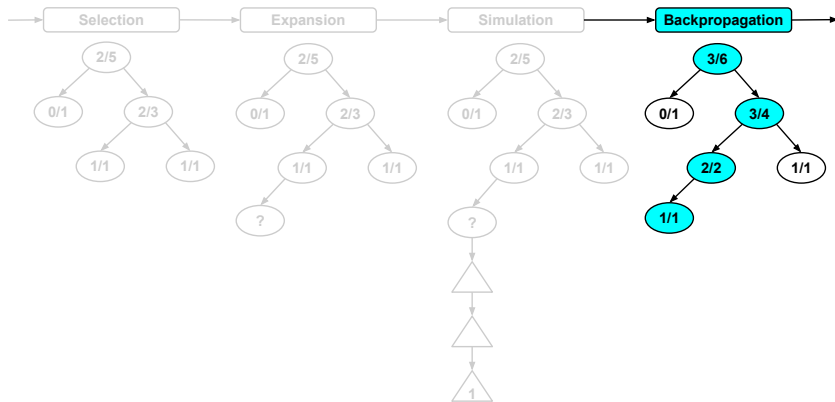




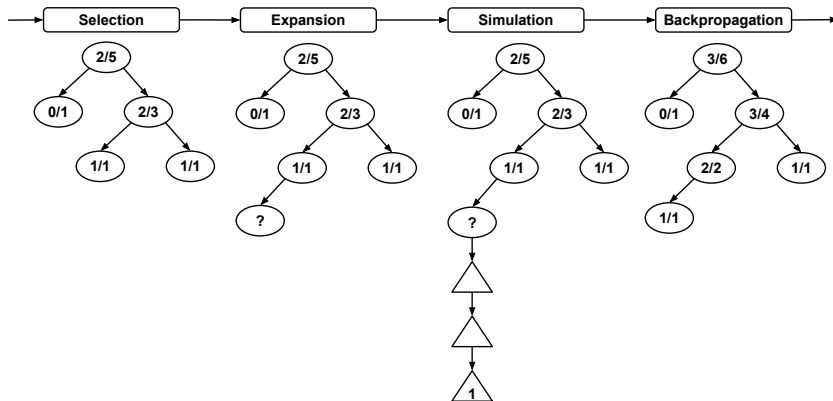
# Four Steps Diagram



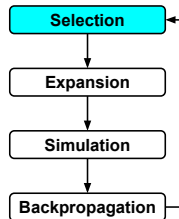
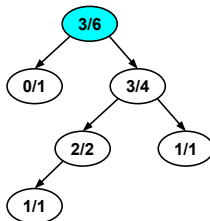
# Four Steps Diagram



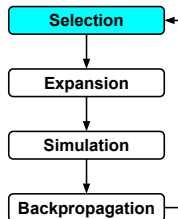
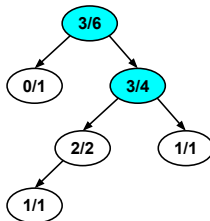
# Four Steps Diagram



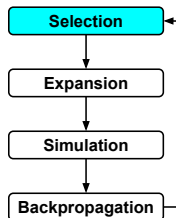
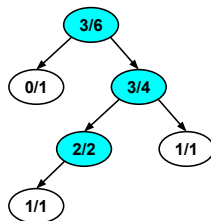
# Four Steps Diagram



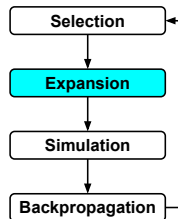
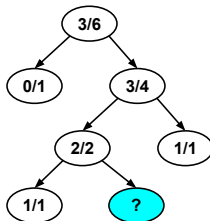
# Four Steps Diagram



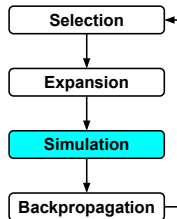
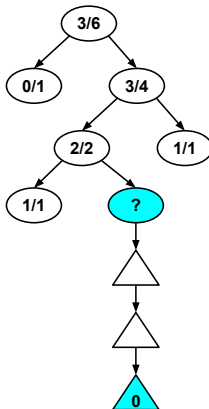
# Four Steps Diagram



# Four Steps Diagram

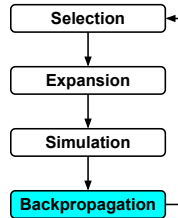
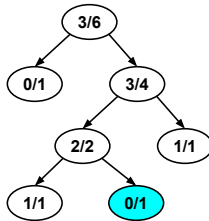


# Four Steps Diagram

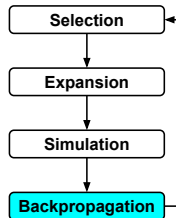
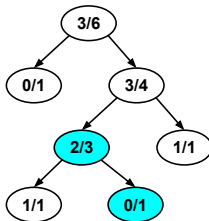




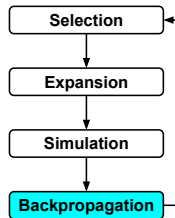
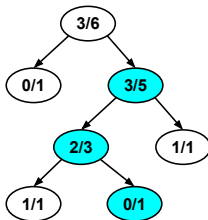
# Four Steps Diagram



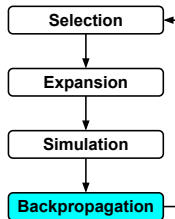
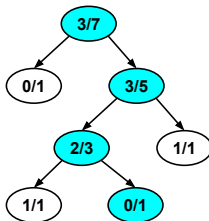
# Four Steps Diagram



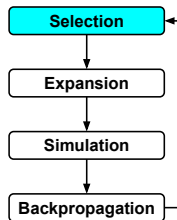
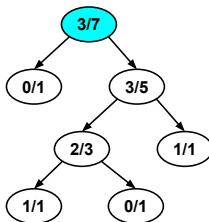
# Four Steps Diagram



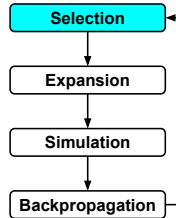
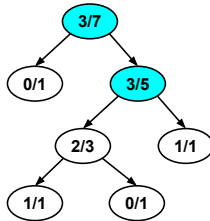
# Four Steps Diagram



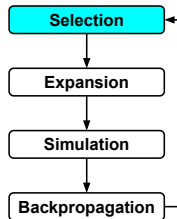
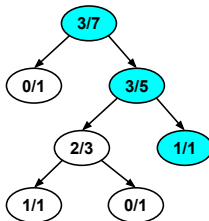
# Four Steps Diagram



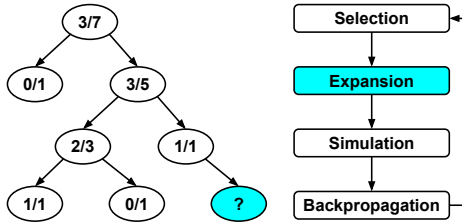
# Four Steps Diagram



# Four Steps Diagram

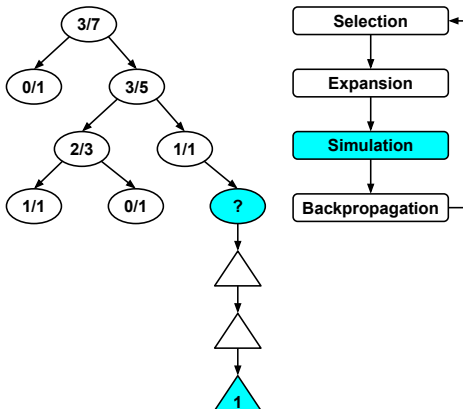


# Four Steps Diagram

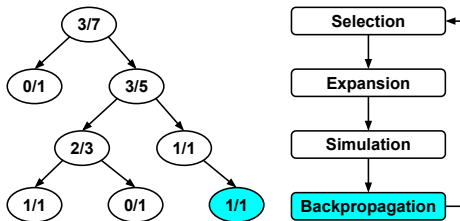




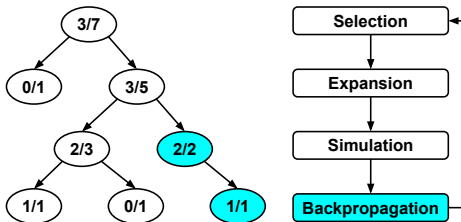
# Four Steps Diagram



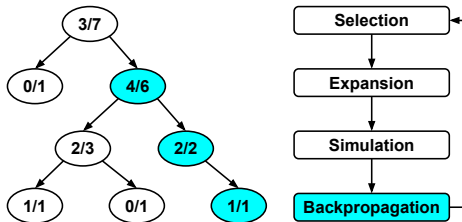
# Four Steps Diagram



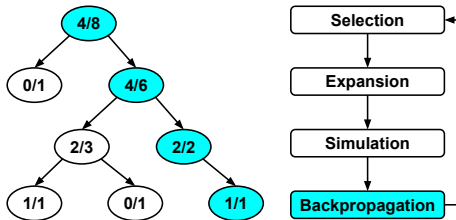
# Four Steps Diagram



# Four Steps Diagram



# Four Steps Diagram



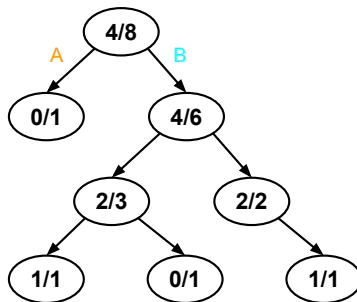
# What Happens When We Choose a Move?

Now we have:

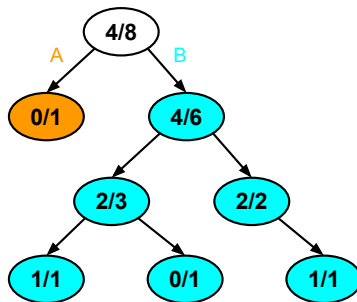
- ▶ A tree structure
- ▶ A method of generating the tree

What happens when we need to choose a move?

# Choosing a Move

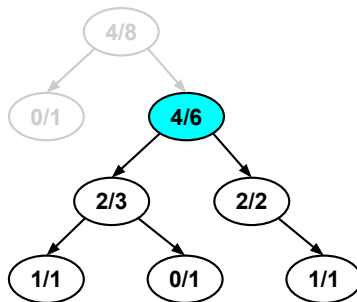


# Choosing a Move

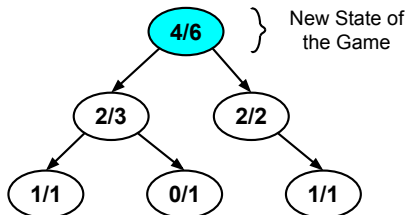




# Choosing a Move



# Choosing a Move



# Exploration vs Exploitation

- ▶ We might overlook better paths
- ▶ Exploration vs Exploitation
  - ▶ Exploration looks at more options
  - ▶ Exploitation focuses on the most promising path
- ▶ Must find a balance between the two

# Upper Confidence Bound Applied to Trees (UCT)

$$UCT(node) = \underbrace{\frac{W(node)}{N(node)}}_{\text{Value of the Node}} + \underbrace{c \sqrt{\frac{\ln(N(\text{parentNode}))}{N(node)}}}_{\text{Exploration Bonus}}$$

- ▶ W represents the number of simulated wins
- ▶ N represents the total number of simulations
- ▶ C is an experimental constant
- ▶ Used during tree traversal
- ▶ Balances exploration vs exploitation

# Outline

Introduction

Naive MCTS Implementation

**Applying MCTS to Go**

Applying MCTS to Narrative Generation

Conclusion

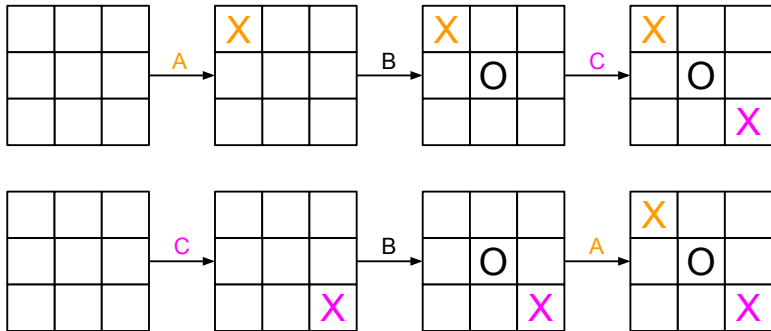
# MCTS applied to Go

What variations can we make specific to Go?

In Go each player takes turn placing pieces on a game board

- ▶ How much does the order of these moves matter?
- ▶ Can we use this to improve MCTS in the context of Go?

# Tree Redundancy

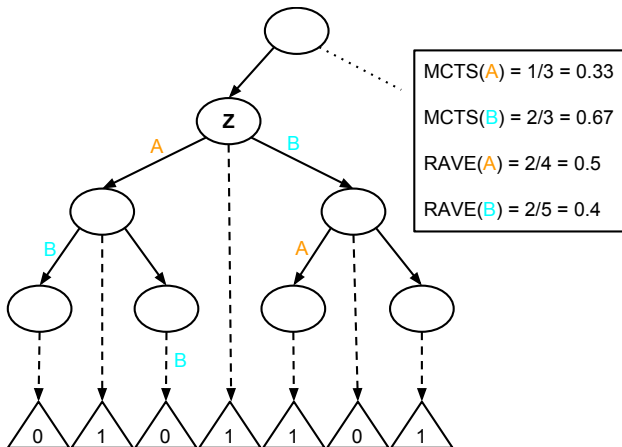


# Rapid Action Value Estimate (RAVE)

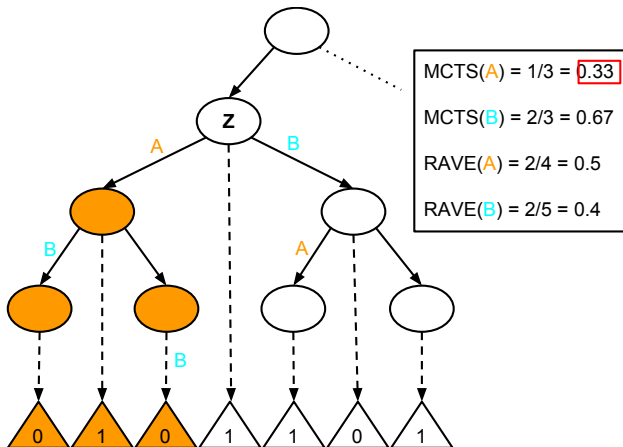
- ▶ Takes advantage of tree redundancy
- ▶ Moves have no contextual dependencies
- ▶ Stores the value of a move within a subtree at each node



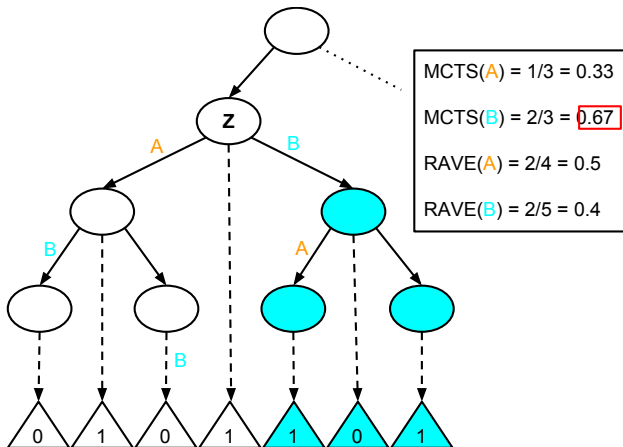
# RAVE Diagram



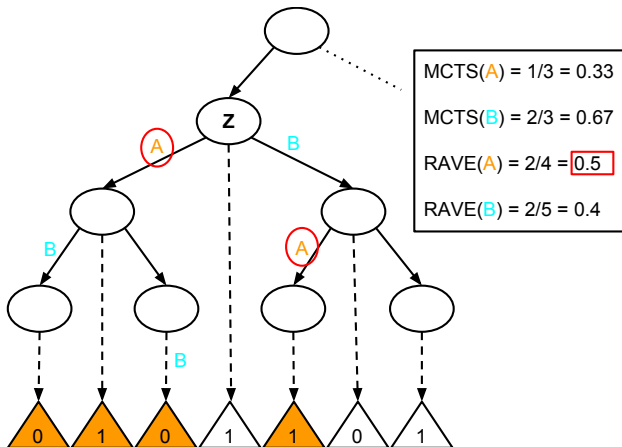
# MCTS Values



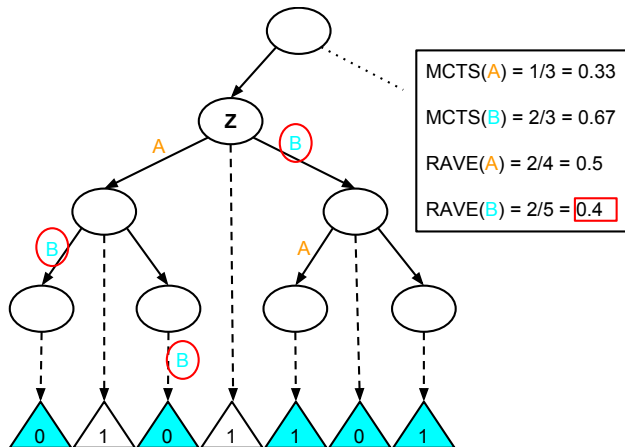
# MCTS Values



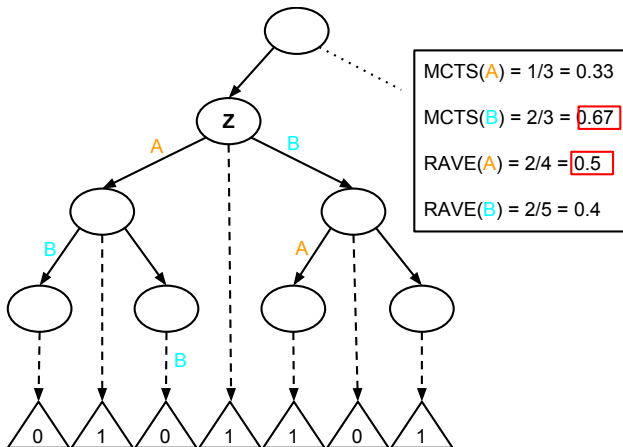
# RAVE Values



# RAVE Values



# MCTS RAVE Comparison



# RAVE

- ▶ Very powerful approach
- ▶ Each simulation provides us with more information
- ▶ Sometimes we do need contextual dependencies
  - ▶ Example: Close tactical battles

# MC RAVE

- ▶ Combines MCTS values with RAVE values
- ▶ Uses a weighted average
- ▶ Favors RAVE values when fewer simulations have been performed
  - ▶ Contextual dependencies are unknown
- ▶ Favors MCTS values when more simulations have been performed
  - ▶ Contextual dependencies are more developed



# Go Results

- ▶ Deterministic approaches could hardly defeat low level amateurs
- ▶ Computer Go programs use MC RAVE
  - ▶ MoGo
  - ▶ Crazy Stone
- ▶ Can compete against top pros in 9x9 Go
- ▶ Can compete against top pros in handicapped 19x19 Go

# Outline

Introduction

Naive MCTS Implementation

Applying MCTS to Go

**Applying MCTS to Narrative Generation**

Conclusion

# Narrative Generation

Kartal et al. applied MCTS to Narrative Generation

- ▶ Crime story
- ▶ User defines the set up and goals for the story
  - ▶ Example Setup: The detective starts in his office
  - ▶ Example Goal: The killer must be arrested

Unlike Go and other games

- ▶ Slightly different tree structure
- ▶ Evaluation function needed

# Actions

- ▶ Actions drive the story
- ▶ Actions are believable based on context
  - ▶ Example: Inspector searches for clues
  - ▶ Example: Character A kills Character B

# Example Actions

**Move(A, P):** A moves to place P.

**Kill(A, B):** B's health to zero(dead).

**Earthquake(P):** An earthquake strikes at place P.

- ▶ Actions take the place of moves as nodes
- ▶ No clear end state
- ▶ The researchers used a set threshold during simulation

# Evaluation function

- ▶ Method of giving nodes value
- ▶ Incorporates believability and goal completion
- ▶ Ensures stories are interesting

$$\text{Value}(\text{story}) = \text{Believability}(\text{story}) * \text{GoalCompletion}(\text{story})$$

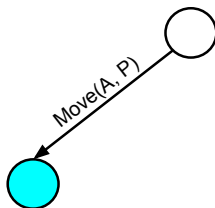
- ▶ Believability is the mathematical product of every action in a story
- ▶ The value is between 0 and 1

# Narrative Generation Test

MCTS compared against three deterministic algorithms

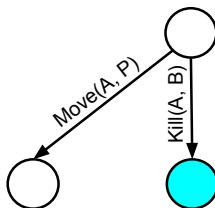
- ▶ Breadth-first search
- ▶ Depth-first search
- ▶ Best-first search

# Breadth-First Search

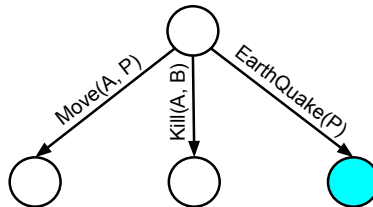




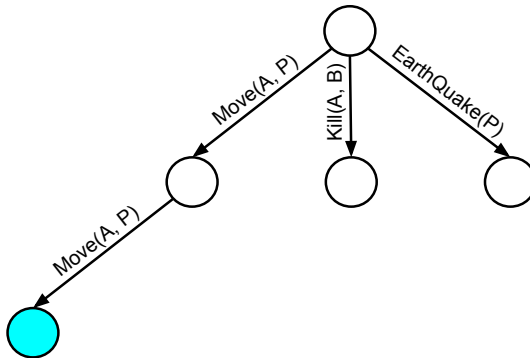
# Breadth-First Search



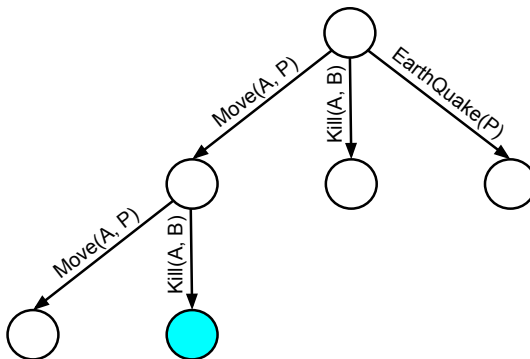
# Breadth-First Search



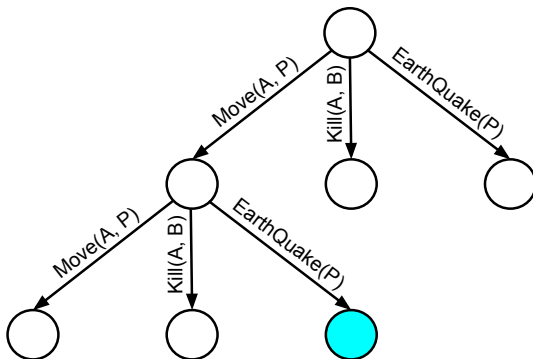
# Breadth-First Search



# Breadth-First Search



# Breadth-First Search



# Depth-First Search



# Depth-First Search



# Depth-First Search





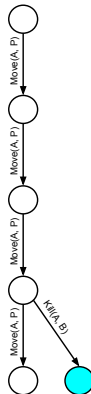
# Depth-First Search



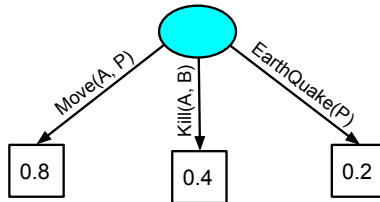
# Depth-First Search



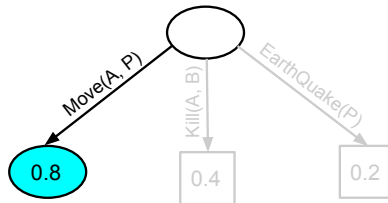
# Depth-First Search



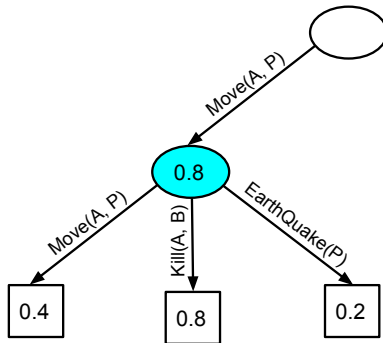
# Best-First Search



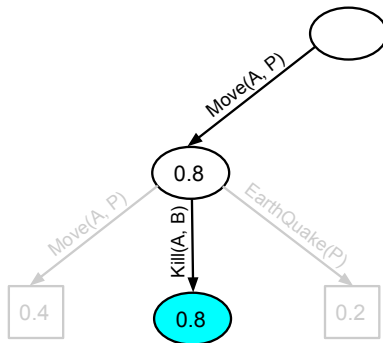
# Best-First Search



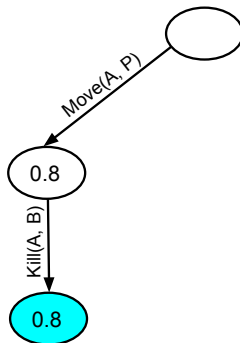
# Best-First Search



# Best-First Search



# Best-First Search





# Test Conditions

Goals for the narrative:

- ▶ At least two people are killed
- ▶ The killer is arrested

Each algorithm was given two budgets

- ▶ 100,000 nodes
- ▶ 3 million nodes

Each algorithm ran three times

The score of the narratives were averaged

# Results

	MCTS	Breadth-first	Depth-first	Best-first
Low Budget	0.07	0.05	<0.001	0.005
High Budget	0.9	0.06	<0.01	<0.01

- ▶ MCTS performed the best in both
- ▶ Breadth-first came the closest out of the deterministic algorithms

# Stories Produced by MCTS

- ▶ Stories from MCTS tended to be believable
- ▶ Completed both user defined goals
- ▶ Some Problems
- ▶ Overall reasonable narratives

# Low Scoring Example from Breadth-First

Sherlock moved to Alice's House. An Earthquake occurred at Alice's House! Sherlock and Alice both died due to the earthquake.

# Outline

Introduction

Naive MCTS Implementation

Applying MCTS to Go

Applying MCTS to Narrative Generation

Conclusion

# Conclusion

- ▶ MCTS successful in extending AI capabilities
- ▶ Tackles problems with larger search spaces
- ▶ Effective in Go and narrative generation
- ▶ Applicable to other problems
  - ▶ Can outperform humans in many puzzles
  - ▶ Real time games
  - ▶ Super Mario Brothers

# Any Questions?

