Faculty of Mathematics and Information Science
Warsaw University of Technology

Multigame Playing - A Challenge for Computational Intelligence

Jacek Mańdziuk

Grand challenges – MULTIGAME PLAYING

- Deep Blue II is unable to play simple tic-tac-toe
- Meta-level approach (autonomous game analysis, learning and problem solving)
- Focus on universal, generic AI/CI algorithms rather than solutions and metods optimized for single (particular) game:
  - Abstract reasoning
  - Knowledge representation
  - Game-independent learning
  - Planning
  - Knowledge transfer
  - Life-long learning
  - ...
- **Return to the idea of Strong AI (AGI)**

Multigame playing - SAL

- Search And Learning (M. Gherrity, 1993)

- Deterministic, two-player, perfect-information board games
- Game independent kernel with the general knowledge about this game genre – pre-defined, unchanged
- Game related knowledge (move-generator, constants defining the board, the pieces, etc.) provided by the user in a separate game-specific module (written in C)
- Two evaluation functions - 1hl MLP (➔ non-zero-sum games)
- Each represented as a NN with TD-learning
- Positional features (pieces, quantities), Dynamic features (moves, captures) and rule-based features (threathened pieces, controlled squares, etc)
- Shallow search (2-ply only)

- EXTREMELY SLOW LEARNING
- EXPLOSIVE NUMBER OF FEATURES

Multigame playing - HOYLE

- Hoyle (Susan Epstein, 1991-2004)
- Two-player, perfect-information, deterministic, finite board games
- Only the rules of the game are provided
- Three tiers of advisors commenting on particular aspects of a game position
- Tier 1: binary assessment of particular moves from the point of view of a given Advisor, e.g. Victory Advisor – winning moves, Wiser – winning in 1 ply, Sadder Advisor – losing, Dont' Lose – not losing in 1 ply, etc.)
- Tier 2: advocating certain plans of play (achieving particular goals)
- Tier 3: *non-binary* assessment of moves (in Advisor's expertise area) – Fork, Freedom (Mobility), Material, etc.
- Tiers 1 and 2 – sequential decision. If not conclusive then weighted decision made in Tier 3

- REASONABLE LEVEL OF PLAY in 18 two-player games (incl. t-t-t).
- WEAK CHESS playing (novice level)

Multigame playing - METAGAMER

- METAGAMER (Barney Pell, 1992-1996)
- Any „symmetric chess-like" (SCL) game, i.e. chess, checkers, shogi
- Universal evaluation function – linear combination of simple pre-defined features (partial goals)
- Similarly to Hoyle, some number of Advisors was employed, which voted on potential usefulness of particular features (in the form of numerical estimation of the worth of particular features)
- Among 23 Advisors there were 4 related to *mobility*, 4 to *threats and capturing*, 4 to *goal functions* and 11 to *material values*
- Shallow search – 1 ply with 1 ply non-quiescence search

- Novice level in chess (GNU Chess level 1 with a Knight handicap) and intermediate level in checkers
- No positive results in ad-hoc defined SCL games

PART 2

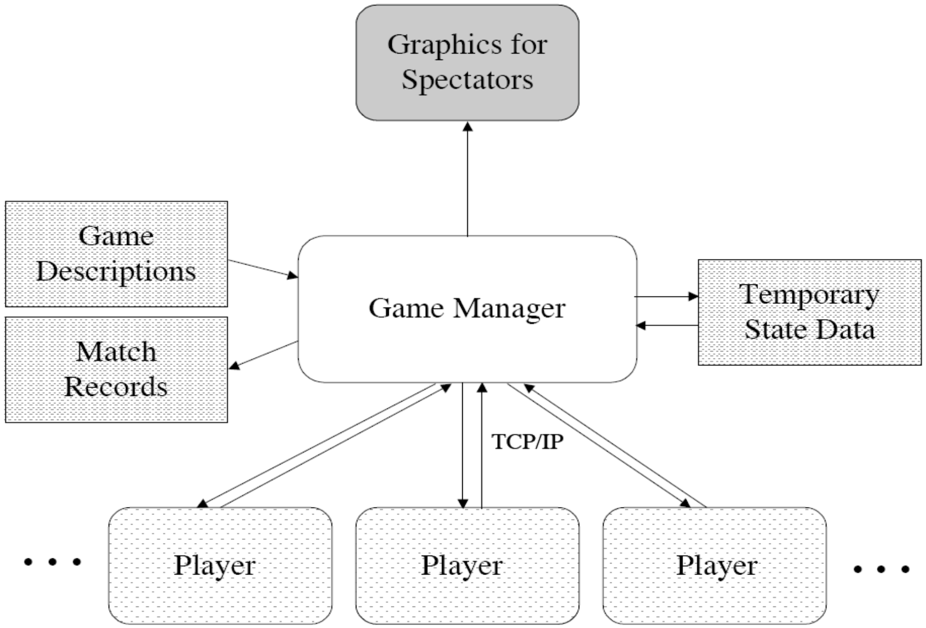
General Game Playing
UCT search

General Game Playing

- Building programs (agents) able to play any game (within a certain class of games) without human intervention.
- Annual GGP tournament (since 2005)
- Stanford online course (2014)
- Game controlled by a GameMaster
- Communication via HTTP

General Game Playing

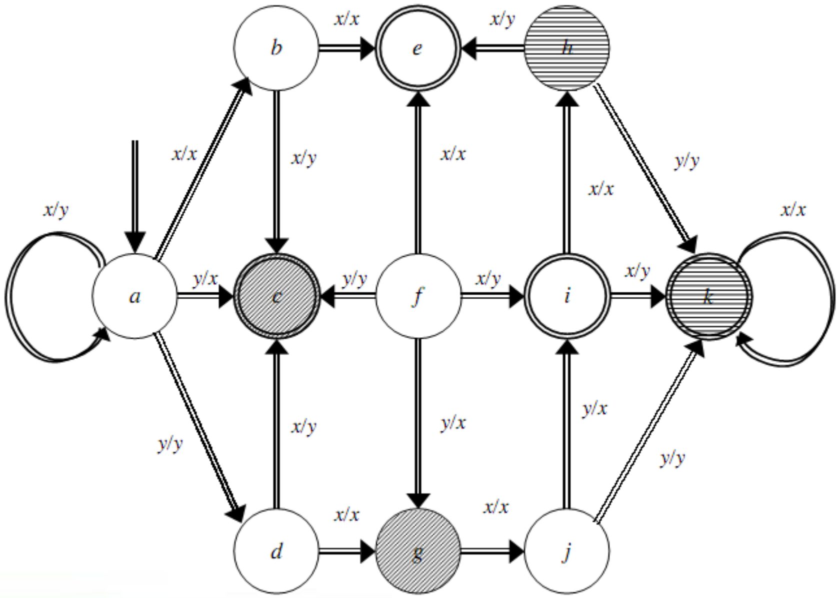
- GameMaster
  - Provides game description to each player
  - Verifies legality of proposed actions
  - Sends out info about (other) players’ actions
  - Monitors time limits
- Time limits
  - StartClock
  - PlayClock



[http://games.stanford.edu/readings/aaai.pdf]

Genre of games

- Multiplayer
- Finite
- Synchronous
- Deterministic



M. Genesereth, N. Love, and B. Pell. General Game Playing: Overview of the AAAI Competition. AI Magazine, 26(2):62-72, 2005.

Game Definition Language (GDL)

- Allows more consise description of game rules
- Uses a version of Datalog – a subset of Prolog
- Specific notation of variables, relations, logic operators
- A set of keywords

Main GDL keywords/relations


```
(role xplayer)
(role oplayer)

(init (cell 1 1 b))
(init (cell 1 2 b))
...
(init (cell 3 3 b))
(init (control xplayer))

;; Cell
(<= (next (cell ?m ?n x))
   (does xplayer (mark ?m ?n))
   (true (cell ?m ?n b)))

(<= (next (cell ?m ?n o))
   (does oplayer (mark ?m ?n))
   (true (cell ?m ?n b)))

...

(<= (next (control xplayer))
   (true (control oplayer)))

(<= (next (control oplayer))
   (true (control xplayer)))

(<= (row ?m ?x)
   (true (cell ?m 1 ?x))
   (true (cell ?m 2 ?x))
   (true (cell ?m 3 ?x)))
```

```
(<= (column ?n ?x)
   (true (cell 1 ?n ?x))
   (true (cell 2 ?n ?x))
   (true (cell 3 ?n ?x)))

(<= (diagonal ?x)
   (true (cell 1 1 ?x))
   (true (cell 2 2 ?x))
   (true (cell 3 3 ?x)))

(<= (diagonal ?x)
   (true (cell 1 3 ?x))
   (true (cell 2 2 ?x))
   (true (cell 3 1 ?x)))

(<= (line ?x) (row ?m ?x))
(<= (line ?x) (column ?m ?x))
(<= (line ?x) (diagonal ?x))

(<= open
   (true (cell ?m ?n b)))

(<= (legal ?w (mark ?x ?y))
   (true (cell ?x ?y b))
   (true (control ?w)))

(<= (legal xplayer noop)
   (true (control oplayer)))

(<= (legal oplayer noop)
   (true (control xplayer)))
```

```
(<= (goal xplayer 100)
   (line x))

(<= (goal xplayer 50)
   (not (line x))
   (not (line o))
   (not open))

(<= (goal xplayer 0)
   (line o))

(<= (goal oplayer 100)
   (line o))

(<= (goal oplayer 50)
   (not (line x))
   (not (line o))
   (not open))

(<= (goal oplayer 0)
   (line x))

(<= terminal
   (line x))

(<= terminal
   (line o))

(<= terminal
   (not open))
```

role(r)
init(p)
true(p)
legal (r,a)
does(r,a)
next(p)
terminal
goal(r,value)

Closed world assumption

GGP simulation scheme

- Calculate legal actions **[legal]**
- For each role $i, i = 1, \dots, N$ **[does]**
  - move[i] **(select an action)**
  - perform(move[i]) **(send out all moves)**
- Calculate the next game step **[next]**
- Check if terminal **[terminal]**
  - YES → goal(role,value) and STOP **[goal]**
  - No → goto [legal]

Games Played

Chess

IBM Deep Blue II

GGP Player

Chess
Checkers
Chinese Checkers (3,4,6)
Connect Four
Connect Five
Connect Four Suicide
Othello
Quarto
Pentago
Pilgrimige
Blocker
Tic-Tac-Toe
9-Board Tic-Tac-Toe
Numeric Tic-Tac-Toe
Tic-Tac-Chess (3,4)
Counterstrike (Simplified)
Eight Puzzle (solving)

Sudoku Puzzle (solving)
Breakthrough
Knight-through
Free For All (2,3,4)
Rock-Paper-Scissors
Lights Out
Cephalopod
Chess and Othello (simultaneously)
Farmers (3)
Zhadu
Nim
Cylinder Checkers
Nine Men's Morris
Finding a Knight's Tour
Flipping Pancakes
...

GGP approaches - summary

- 2005 – 2006
  - heuristic-based syntactical game analysis
  - construction of the evaluation function

- 2007-2016 → **MCTS** (cont. GVGP)
  - simulation-based methods
  - Cadia Player (2007, 2008, 2012) – Reykiavik University, Iceland
  - Ary (2009, 2010) – Paris 8, France
  - TurboTurtle (2011, 2013) – USA
  - Sancho (2014) – USA & UK
  - Galvanise (2015)
  - WoodStock (2016) - France

MCTS – MOTIVATION

- MCTS combines the focused and precisely-defined tree search with the power of massive random sampling of the problem space

- MCTS can be applied to any problem which relies on sequential decisions implemented as a tree-search process (whose solutions can be represented as paths in a tree)

- MCTS is a well suited method for problems intractable by min-max / alpha -beta search (deep tree, large branching factor, lack of compact and reliable assessment function)

- DAG

MONTE CARLO METHODS – BASICS

Rooted in statistical physisc – numerical approximation of complex integrals.

Used for estimating the quality of actions in certain domains (games, planning) based on **uniform sampling** (now called ***flat Monte Carlo***)

Flat MC in many domains / problems is ineffective as it does not assume modeling the opponent in any way

Significant improvement in 2006 → UCB1 action selection policy → UCT search tree method

UCB1 (K-ARMED BANDIT)

- UCB = Upper Confidence Bounds (UCT = UCB applied to Trees)
- k-Armed Bandit Problem or k (one-arm) Bandits Problem
- Distributions of pay-offs $\{X_{j,t}\}_{t=1,2,\ldots}$ $j=1,2,\ldots,k$ are fixed, but unkno wn
- How to maximize the total (long-term) reward?

- Exploration vs. exploitation balance
- First, play each arm once
- Then, use the following rule:

$$A^* = \arg\max_{i=1,\ldots,k} \left\{ \overline{X}_i + C\sqrt{\frac{\ln n}{n_i}} \right\}, \quad C = \sqrt{2}$$

UCT - MULTIPLE SIMULATIONS

Perform multiple simulations according to the following pattern:

- Choose an action not yet selected (if exists)
- If all had already been tried select action a^*

$$a^* = argmax_{a \in A(s)} \left\{ Q(s,a) + C\sqrt{\frac{\ln N(s)}{N(s,a)}} \right\}$$

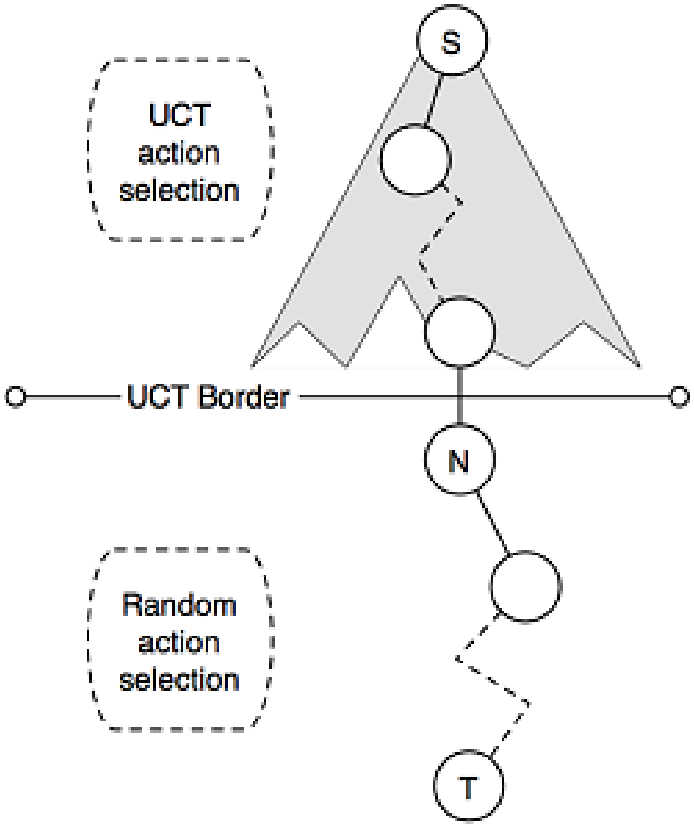
- *Q(s,a)* – the average result for a pair (*state*, *action*)
- *N(s)* – the number of visits in state *s*
- *N(s,a)* – the number of times action *a* was selected in state *s*

Once the simulation is completed propagate the result back in the tree

The saliency of C parameter

The UCT-based GGP player performs as many such simulations as possible (within allotted time), thus estimating the Q(s,a)

MCTS WITH UCT

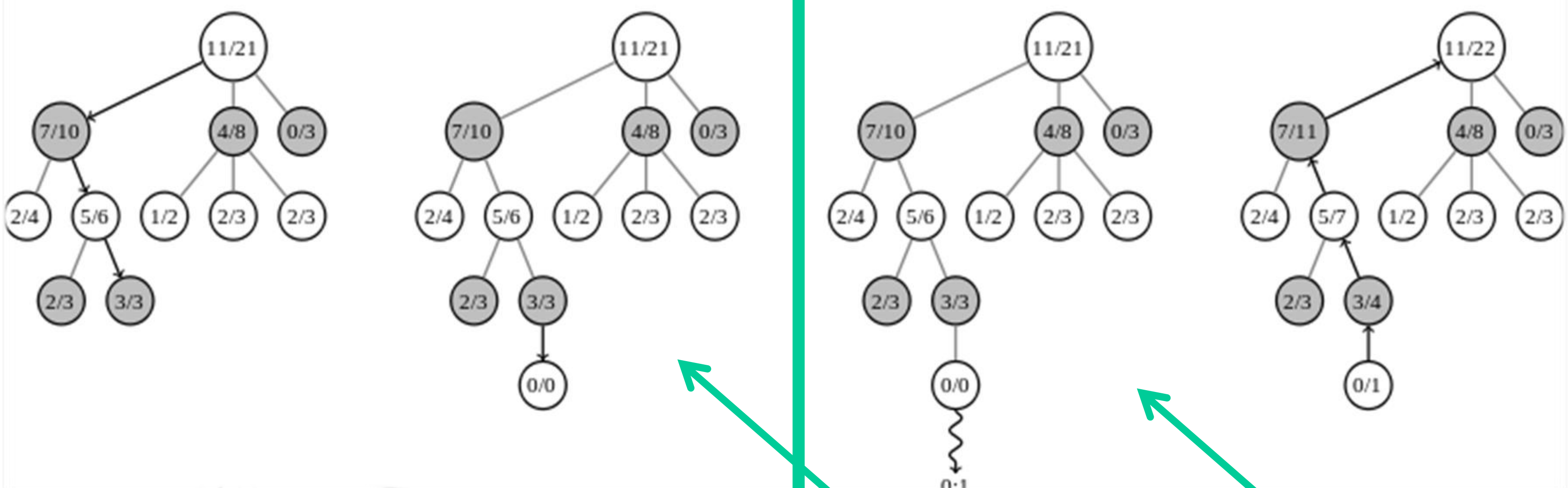


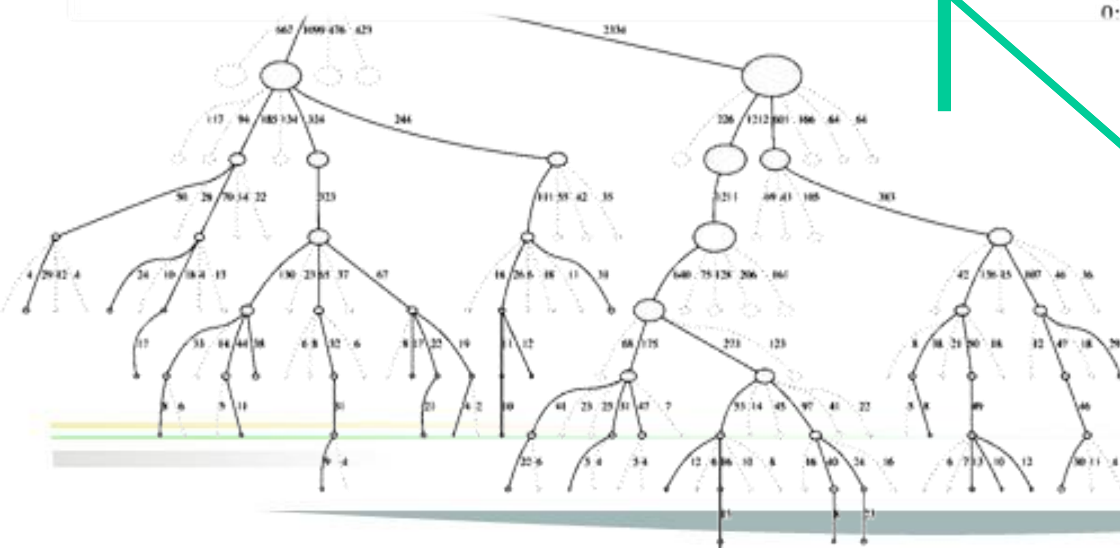
- Due to memory limitations:
  - A game tree is built gradually – one new node is added in each simulation.
  - Once the real (not simulated) action is selected by the system part of the tree above a given node is deleted from memory
- Random simulation to the leaf node.
- Node's assessment is equal to the average payoff of the simulations made so-far.

Bjornsson, Finnsson, IEEE TCIAI G, 1(1), 2009

MCTS/UCT

selection | expansion | simulation | backpropagation





Tree policy (UCB1)

Default policy (random sampling)

MCTS/UCT

- In practice there are some constraints:
  - Computational budget
  - Time constraints
  - Memory limitations

- The system performs as many simulations as possible estimating the Q(s,a), in particular **Q(root, a)**

- Afterwards the action to be taken (in real NOT simulated mode) is selected

MCTS – ACTION SELECTION CRITERIA

$$\arg\max_a Q(root, a)$$ – ***Max child policy***

$$\arg\max_a N(root, a)$$ – ***Robust child policy***

Continue sampling until both measures point the same node - ***Max Robust child policy***

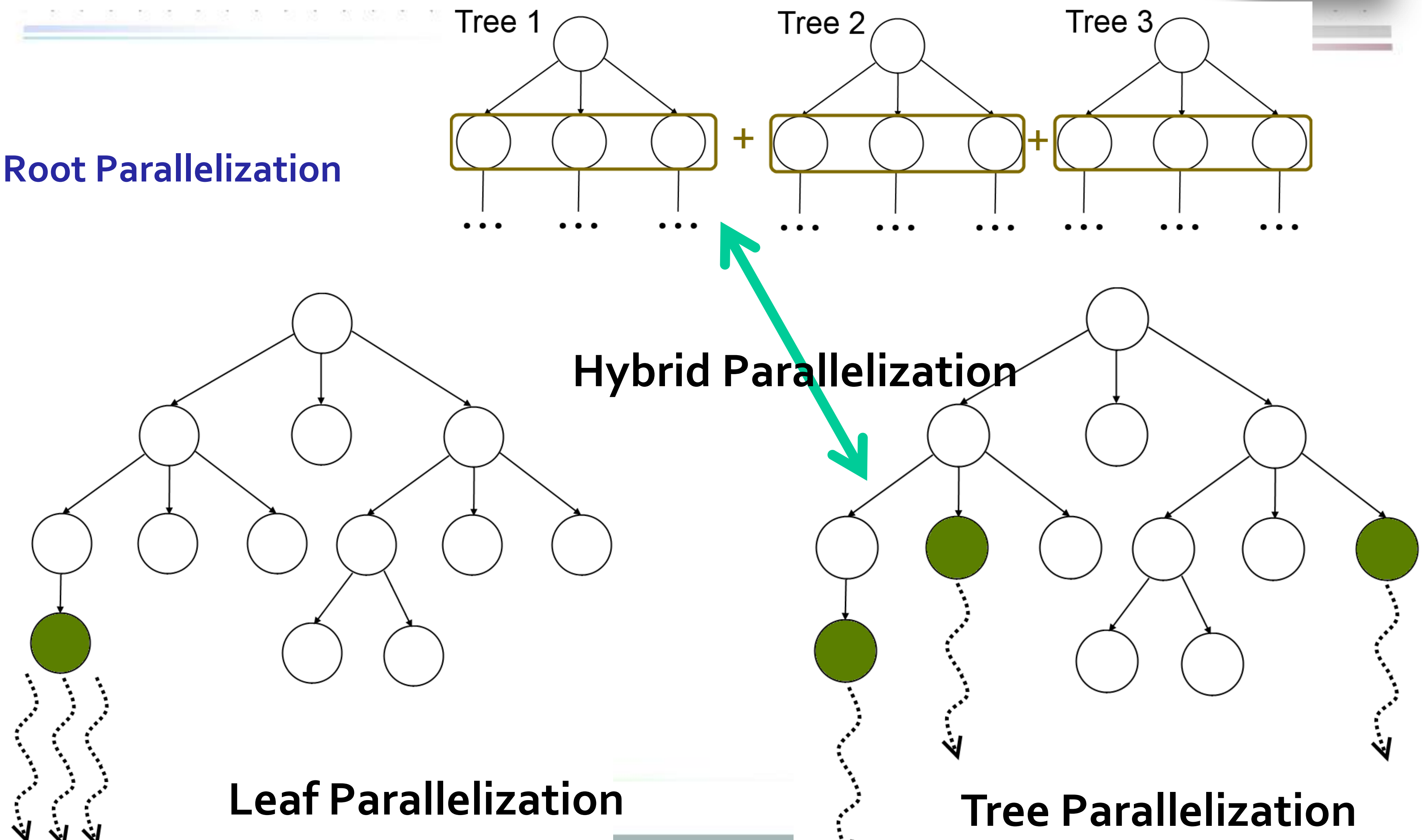
A combination of *Max* and *Robust* – ***Secure child policy***:

$$\arg\max_a \left[Q(root, a) + \frac{k}{\sqrt{N(root, a)}} \right]$$

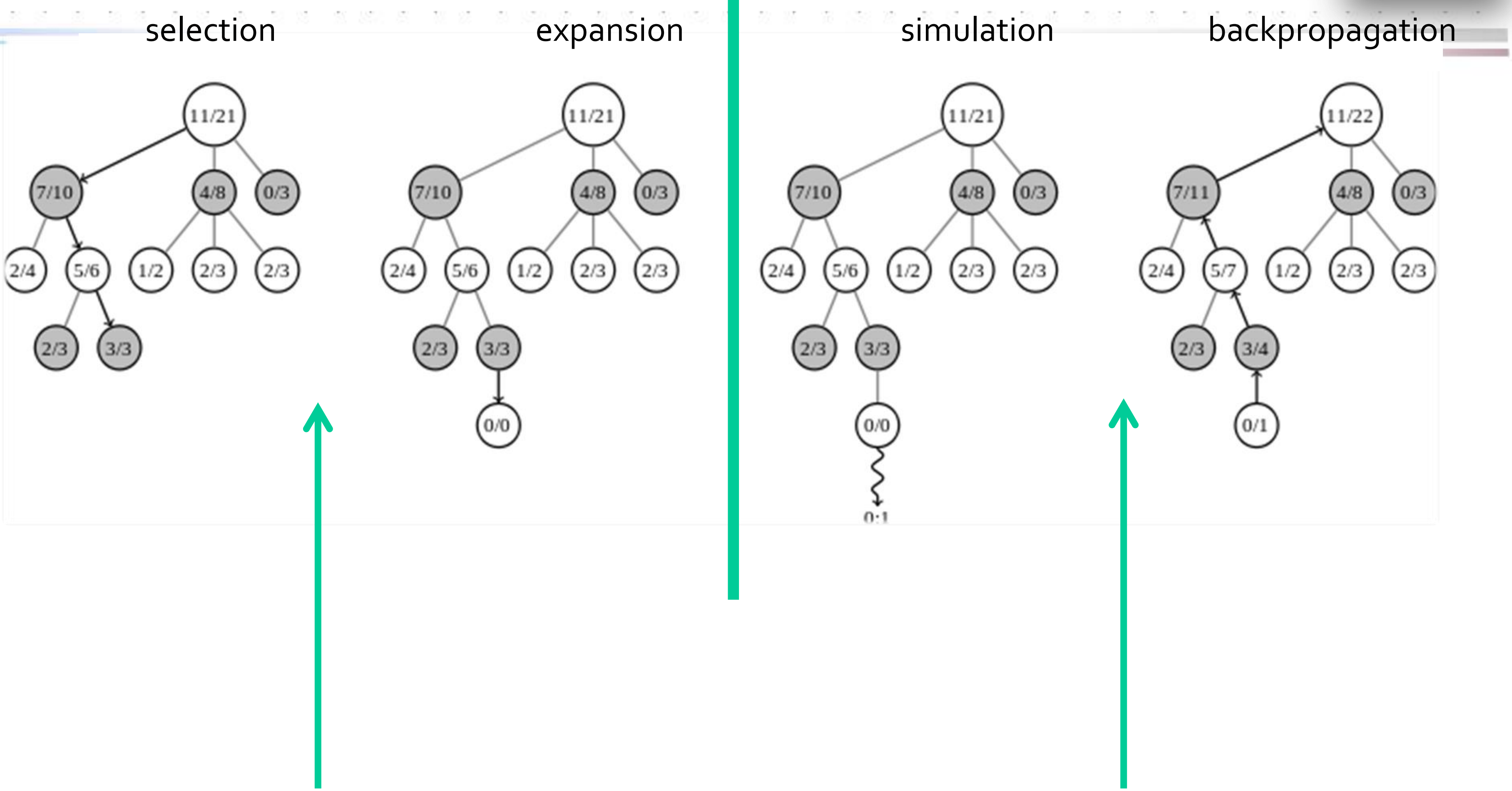
MCTS - HIGHLIGHTS

- Estimates true min-max value
- Anytime
- Aheuristic (Knowledge-free)
- Asymmetric (Best-First-like search)
- Well scaling and easily parallelized

MCTS – EASY PARALLELIZATION



MCTS/UCT



Tree policy (UCB1)

Default policy (random sampling)

TREE POLICY ENHANCEMENTS

- Three types of approaches:
  - Replacing UCB1 by another exploration-exploitation formula → bayesian inference
  - Tuning UCB1 (variations)
  - UCB1 supported with an evaluation function

UCB1-TUNED

- Distributions of pay-offs $\{X_{j,t}\}_{t=1,2,\ldots}$ $j=1,2,\ldots,k$ are fixed, but unknown
- First, play each arm once
- Then, use the following rule: $$A^* = \arg\max_{i=1,\ldots,k}\left\{\overline{X}_i + C\sqrt{\frac{\ln n}{n_i}}\right\}, \quad C=\sqrt{2}$$

$$C = \sqrt{\min\left\{\frac{1}{4}, V_i(n_i)\right\}}$$

where
$$V_i(r) = \left(\frac{1}{r}\sum_{m=1}^{r} X_{i,m}^2\right) - \overline{X_{i,r}^2} + \sqrt{\frac{2\ln n}{r}}$$

UCB1 AND EVALUATION FUNCTION

- Suppose we have a „light"evaluation function:
  - Initial sorting of not-yet-selected actions
  - Initial estimation of actions' strengths based on some number of simulations

EXPANSION ENHANCEMENTS

- Rarely considered in practice
- One may try to optimize the number of new nodes added
  - Hard to estimate
  - Problem dependent
- One may optimize the number of concurrent simulations
  - The information is not updated (accurate)

BACKPROPAGATION ENHANCEMENTS

- Weighting simulation results
  - Some simulations are more important than the others
  - Newer simulations preferred over the later ones (esp. in the case of changing environment)
  - Shorter simulations are more accurate than the longer ones
  - Each simulation is assigned a positive integer weight ***w*** and is treated as performed ***w*** times

- Decaying Reward
  - Preference for shorter wins over the longer ones
  - A reward value is decayed by a value of $\gamma, 0 < \gamma \leq 1$ between each two nodes on the backpropagation path

DEFAULT POLICY ENHANCEMENTS

The biggest room for potential improvement

Motivation:

- The true min-max estimation, but …
- In the case of insufficient number of simulations: there is too much randomness in actions’ assessment
- The sequences of actions are often „not realistic”

HISTORY HEURISTICS - GIBBS DISTRIBUTION

- An action which was successful in the past (regardless of the state) is treated as a potential candidate also in a given state.
- After each simulation statistics for each action are updated (the average performance – expected reward)

- Gibbs distribution:

$$\pi_s(a) = \frac{e^{\frac{Q(a)}{\tau}}}{\sum_{b \in A(s)} e^{\frac{Q(b)}{\tau}}} \qquad \tau > 0$$

Roulette wheel or **Epsilon-greedy**

LAST GOOD REPLY POLICY

- Used specifically in games
- Each move is considered a reply to the opponent’s last move and treated as successful if the player won that game
- For each move the last successful reply is stored (frequently overridden)
- In the simulation, if the LGR move is legal it will be played, otherwise the default policy will be used

DEFAULT POLICY ENHANCEMENTS

- Except for the HH (and its variants) it is hard to show a generally improving method unless some domain knowledge is considered

- The use of the „ad-hoc" constructed (**simple and automatically devised**) evaluation function

  - Replacing the whole playout phase (with some predefined probability)

  - Finishing the playout phase before reaching the leaf node (with some predefined probability in each node → MAGICIAN

GGP Competition 2012

- Two phases of the tournament

- PHASE 1: Seven games played against selected opponents
- The top 8 advanced to the final round (MP was the 5th, 4 wins and 3 losses)

- PHASE 2 – until two losses
- Finally MP placed 7th

- Connect-4
- Cephalopod Micro
- Free for All 2P
- Pentago
- 9 Board Tic-Tac-Toe
- Connect-4 Suicide
- Checkers
- Farming Quandries
- Pilgrimage

GGP Competition 2013

- Two phases of the tournament

- PHASE 1: many games including 1,2,4-player ones; 5 advanced to phase 2 (but not MP ☹, and not M ☹)

- MP – a bug in single-player games ☹

- Finally MP placed 11th (out of 16)

GGP Competition 2014

- Three phases of the tournament
- Massive Stanford online course

- PHASE 1: eliminations (various types of games)
- The top 17 advanced to the second PHASE
- PHASE 2 – top 12 advanced to the final phase

- PHASE3 – we ended up at the 7-8th place (out of 49 participants)

Conclusions

- **Multigame playing is besides intuitive playing and creativity one of the hallmarks of human intelligence (in game domain)**
- GGP directly addresses this issue and stimulates research towards accomplishing this goal
- The are some inefficiencies (especially slowness of analysis) which stem from logic description → how to obey/alleviate them?
- GDL (GGP) is potentially extendable to non-deterministic games
- **If we leave out time constraints more established CI learning approaches (neural nets, genetic/memetic methods) can come into play**
- **Applicability of UCT extends beyond game domain (problems with pay-off defined only in terminal states and/or varying in time)**
- Strategy switching mechanism allows for greater flexibility of the method, its self-improvement and adaptability