The Monte-Carlo Revolution in Go

Rémi Coulom

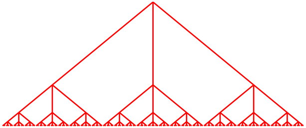
December, 2015

Game Complexity

Game	Complexity*	Status
Tic-tac-toe	10^3	Solved manually
Connect 4	10^{14}	Solved in 1988
Checkers	10^{20}	Solved in 2007
Chess	10^{50}	Programs $>$ best humans
Go	10^{171}	Programs $\ll$ best humans

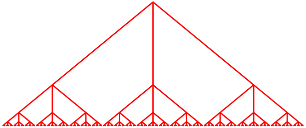
*Complexity: number of board configurations

Dealing with Huge Trees

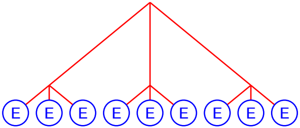


Full tree

Dealing with Huge Trees

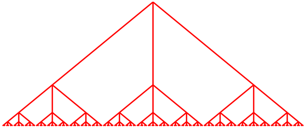


Full tree

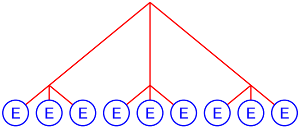


Classical approach =
depth limit + pos. evaluation (E)
(chess, shogi, . . .)

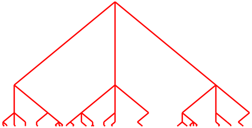
Dealing with Huge Trees



Full tree

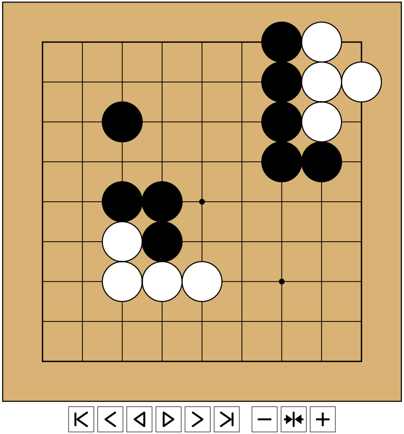


Classical approach =
depth limit + pos. evaluation (E)
(chess, shogi, . . .)

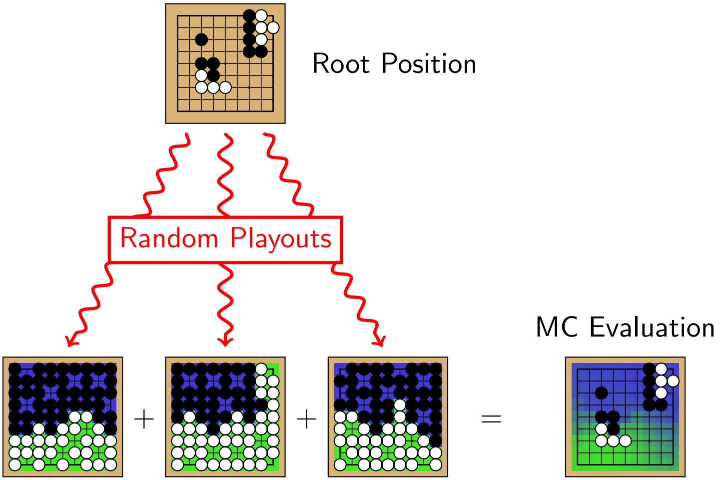


Monte-Carlo approach =
random playouts

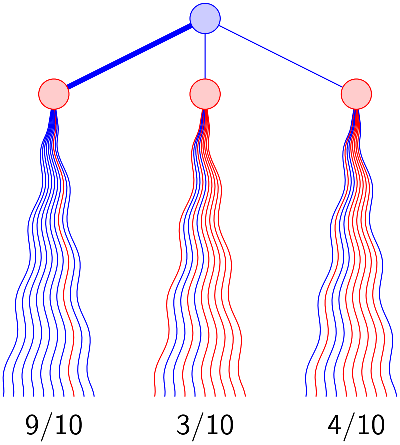
A Random Playout



Principle of Monte-Carlo Evaluation



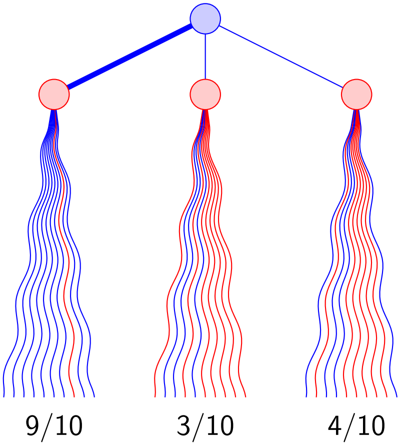
Basic Monte-Carlo Move Selection



Algorithm

- N playouts for every move
- Pick the best winning rate
- 5,000 playouts/s on 19x19

Basic Monte-Carlo Move Selection



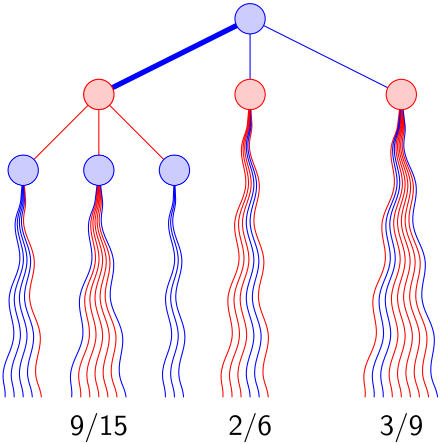
Algorithm

- N playouts for every move
- Pick the best winning rate
- 5,000 playouts/s on 19x19

Problems

- Evaluation may be wrong
- For instance, if all moves lose immediately, except one that wins immediately.

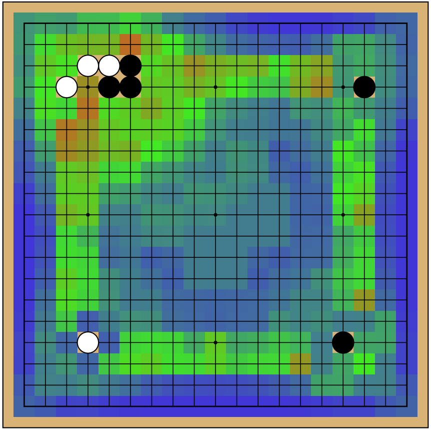
Monte-Carlo Tree Search



Principle

- More playouts to best moves
- Apply recursively
- Under some simple conditions: proven convergence to optimal move when #playouts $\rightarrow \infty$
- UCT: $\mathrm{UCB}_i = \frac{W_i}{N_i} + c\sqrt{\frac{\log t}{N_i}}$

Supervised Learning of Stone Patterns

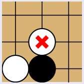


◯ to move

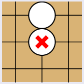
Patterns

- Library of local shapes
- Automatically generated
- Used for playouts
- Cut branches in the tree

Examples (out of ~30k)



Good

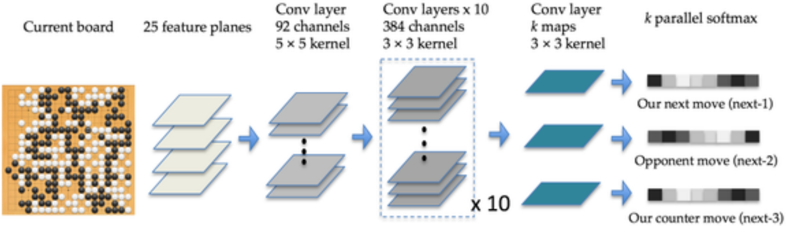


Bad

Deep Convolutional Neural Networks

Google and Facebook play Go

- Christopher Clark, Amos Storkey, 2014
- Chris J. Maddison *et al.*, ICLR 2015
- Yuandong Tian and Yan Zhu, ICLR 2016. KGS 3 dan!



History (1/2)

Pioneers

- 1993: Brügmann: first MC program, not taken seriously
- 2000: The Paris School: Bouzy, Cazenave, Helmstetter

History (1/2)

Pioneers

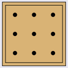
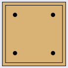
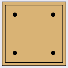
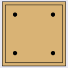
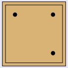
- 1993: Brügmann: first MC program, not taken seriously
- 2000: The Paris School: Bouzy, Cazenave, Helmstetter

Victories against classical programs

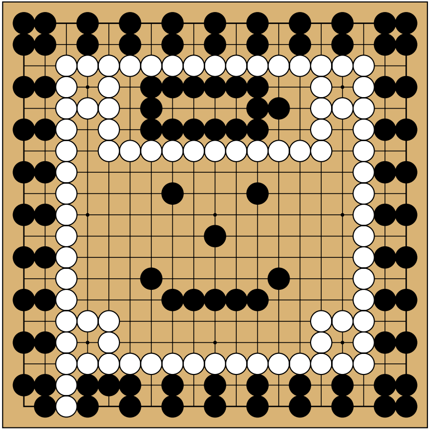
- 2006: Crazy Stone (Coulom) wins 9×9 Computer Olympiad
- 2007: MoGo (Wang, Gelly, Munos, ...) wins 19×19

History (2/2)

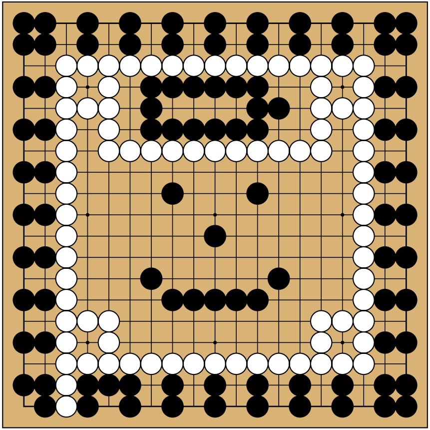
Games Against Strong Professionals

- 2008-08:  MoGo beats Myungwan Kim (9p), H9
- 2012-03:  Zen beats Masaki Takemiya (9p), H4
- 2013-03:  CrazyStone beats Yoshio Ishida (9p), H4
- 2014-03:  CrazyStone beats Norimoto Yoda (9p), H4
- 2015-03:  CrazyStone loses to Chikun Cho (9p), H3

Limits of the Current MC Programs



Limits of the Current MC Programs



Difficulties

- Tree search can't handle all the threats.
- Must decompose into local problems.
- Patterns help, but are sometimes wrong

A Promising Approach

- Policy Gradient (Tobias Graf, Abakus)

Conclusion

Summary of Monte-Carlo Tree Search

- A major breakthrough for computer Go
- Works for similar games (Hex, Amazons) and automated planning

Conclusion

Summary of Monte-Carlo Tree Search

- A major breakthrough for computer Go
- Works for similar games (Hex, Amazons) and automated planning

Perspectives

- Deep convolutional neural networks for clever patterns
- Policy gradient for adaptive playouts