Monte-Carlo Tree Search in Crazy Stone

Rémi Coulom

Université Charles de Gaulle, INRIA, CNRS, Lille, France

November 8th-10th, 2007

UEC and 12th Game Programming Workshop, Japan

Talk Outline

A New Approach to Go

The Challenge of Go

- strongest programs weaker than amateur humans

Difficulty of Position Evaluation

- has to be dynamic
- unlike quiescence search + static evaluation of western chess
- local search lacks global understanding

The Monte-Carlo Approach

- random playouts
- dynamic evaluation with global understanding

The Monte-Carlo Revolution: Pioneers

1993: Bernd Brügmann (Gobble)

- Not considered seriously

2000-2005: The Paris School

- Bernard Helmstetter (Oleg)
- Tristan Cazenave (Golois)
- Bruno Bouzy (Indigo)
- Guillaume Chaslot (Mango), joined in 2005

The Monte-Carlo Revolution: Success

2006: Success on small boards

- Crazy Stone wins 9×9 Computer Olympiad
- Viking (Magnus Persson), then Crazy Stone, then MoGo (Yizao Wang and Sylvain Gelly) lead 9×9 CGOS

2007: Success on all boards

- MoGo wins 19×19 Computer Olympiad
- Steenvreter (Erik van der Werf) wins 9×9
- Crazy Stone beats KCC Igo with a score of 15-4 on 19×19

Principle: Random Playouts

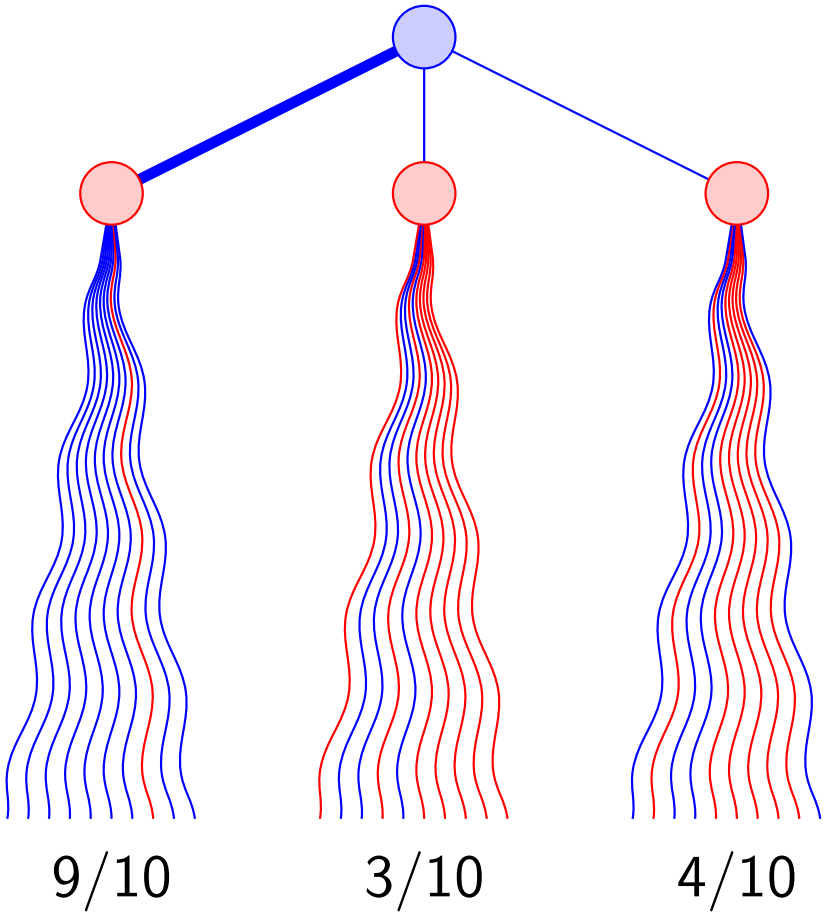
One Playout

- Play at random
- Don't fill-up eyes

Position Evaluation

- Run many playouts
- Average them

Move-Selection Method



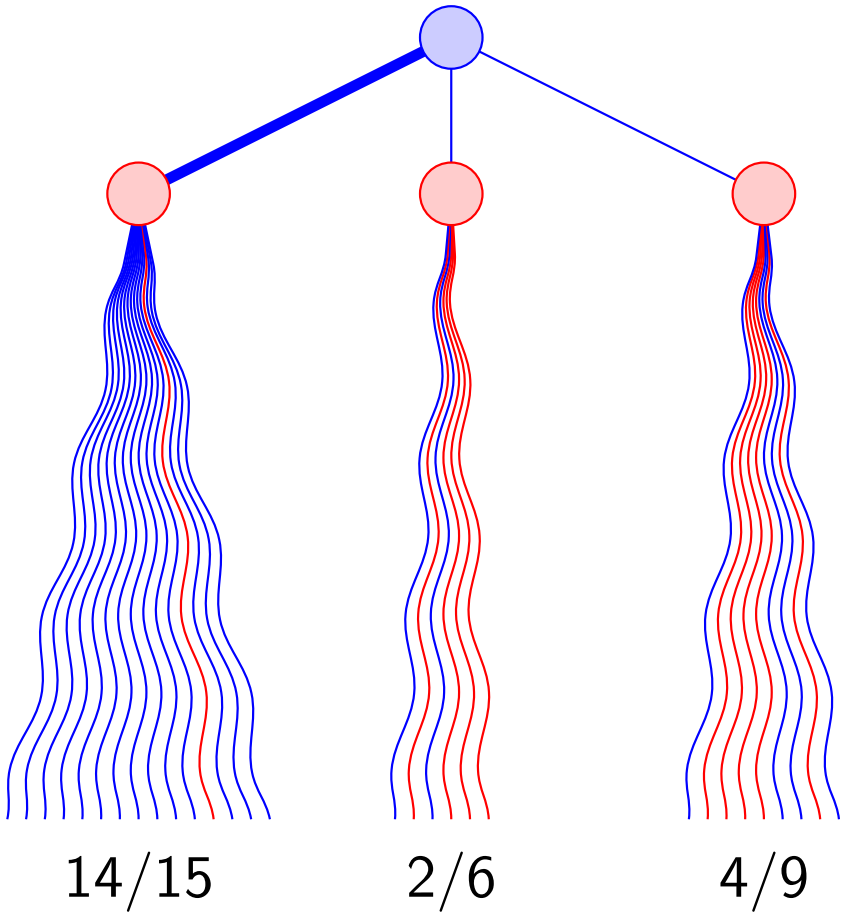
Algorithm

- N playouts for every move
- pick the best winning rate

Cost

- accurate like $1/\sqrt{N}$
- 0.01 precision requires $\sim 10,000$ playouts

Efficient Playout Allocation



Idea

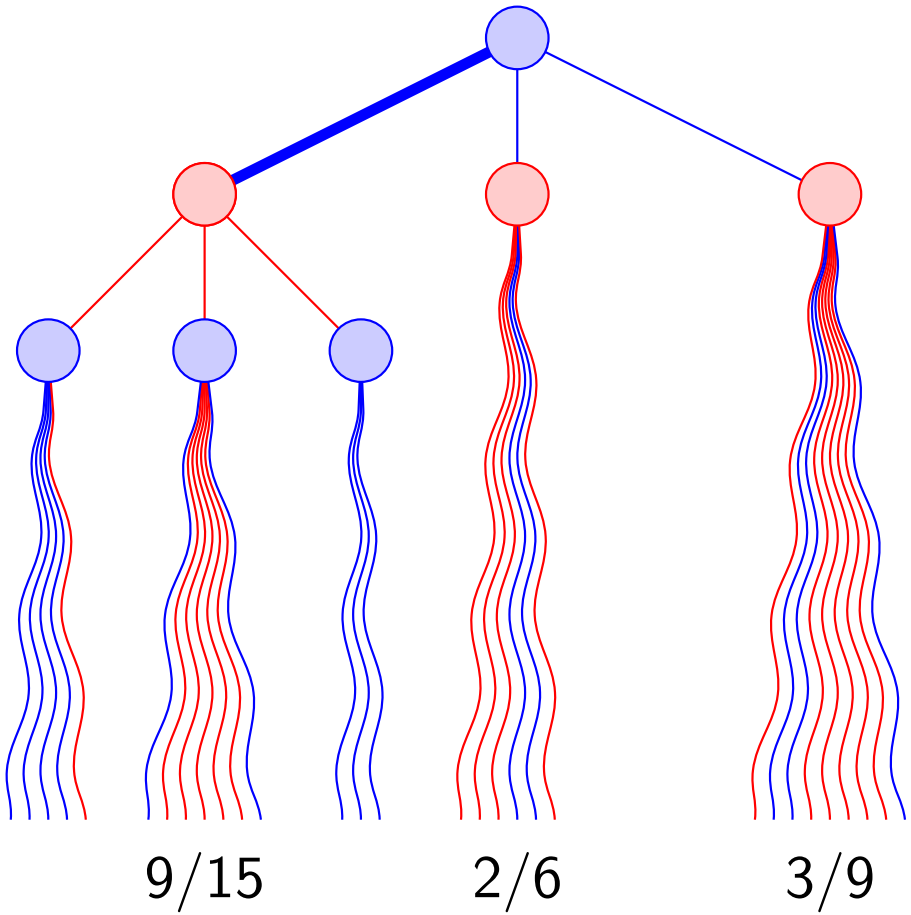
- more playouts to best moves

UCB: Upper Confidence Bound

$$\mathrm{UCB}_i = \frac{W_i}{N_i} + c\sqrt{\frac{\log t}{N_i}}$$

- W_i: wins (move i)
- N_i: playouts (move i)
- c: exploration parameter
- t: playouts (all moves)

Recursive Tree Search: UCT



- Apply UCB to every position visited more than N_0 times
- No min-max backup: backup average outcome
- Proved convergence to min-max value
- Best-first tree growth

Efficiency of Tree Search

Successes

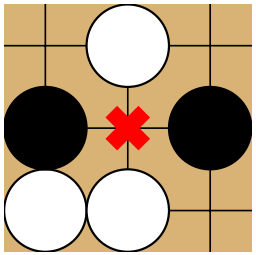
- gold in Turin Olympiad on 9×9
- 9×9 level on KGS: about 10k
- strength scales with thinking time
- only domain knowledge: don't fill eyes, and in atari, extend

Limits

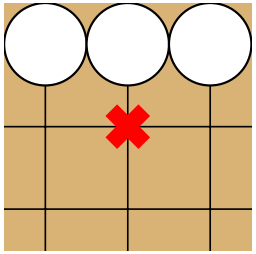
- Not deep enough, even on 9×9
- Too many moves on 19×19
- 19×19 level on KGS: about 30k

Patterns

- learnt from human games
- Combine several features:
  - shape (surrounding stones)
  - distance to previous move
  - capture, extension
  - ...
- Probability distribution over moves
- Used in playouts



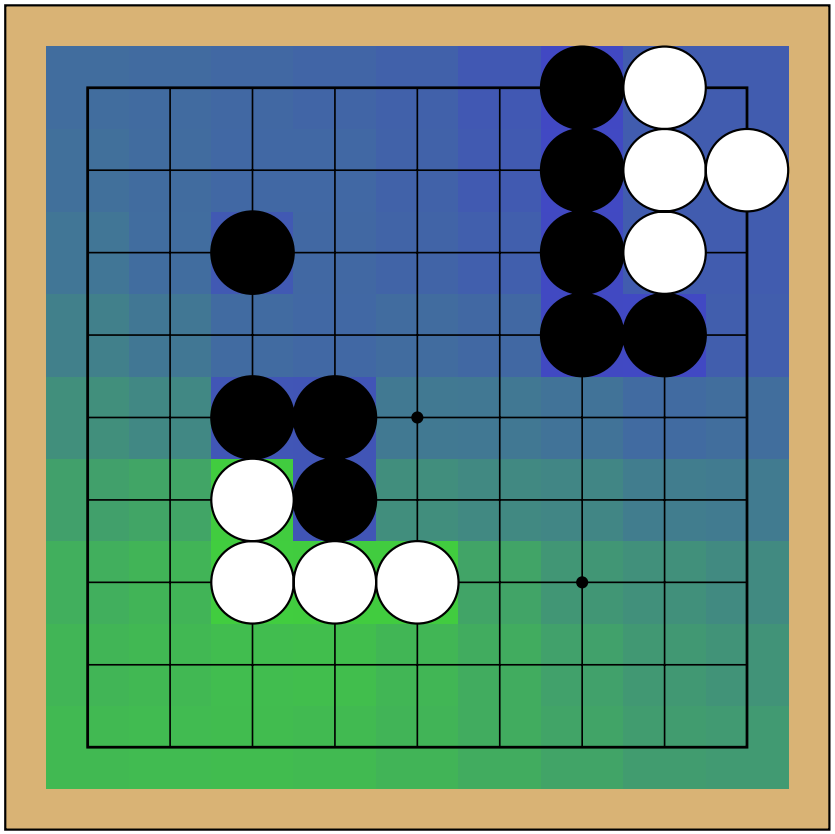
High probability



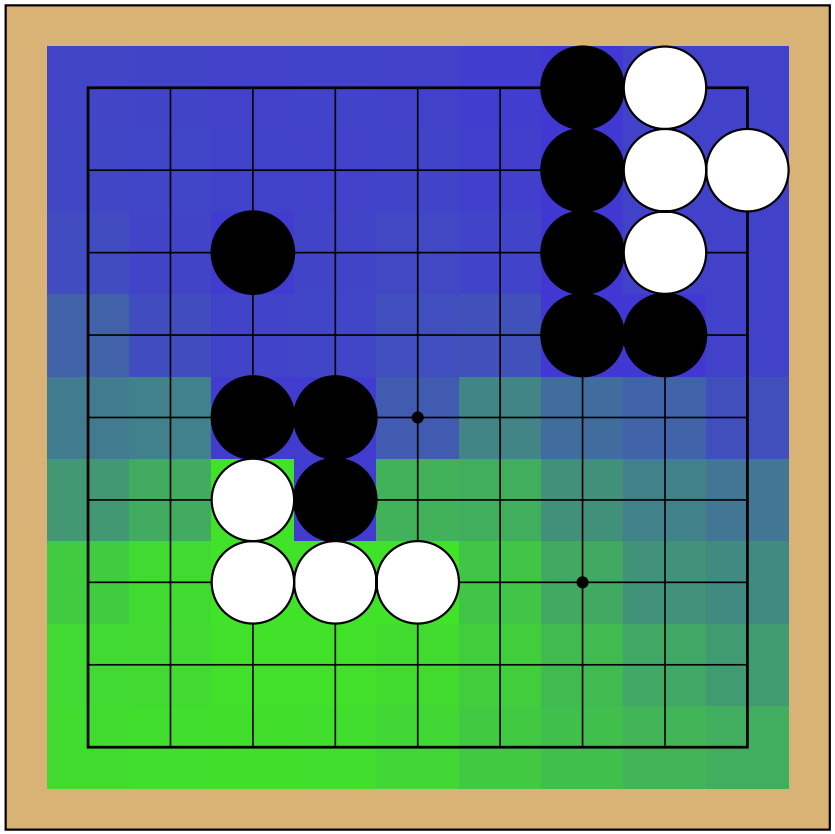
Low probability

Random playout with patterns

Comparison 1

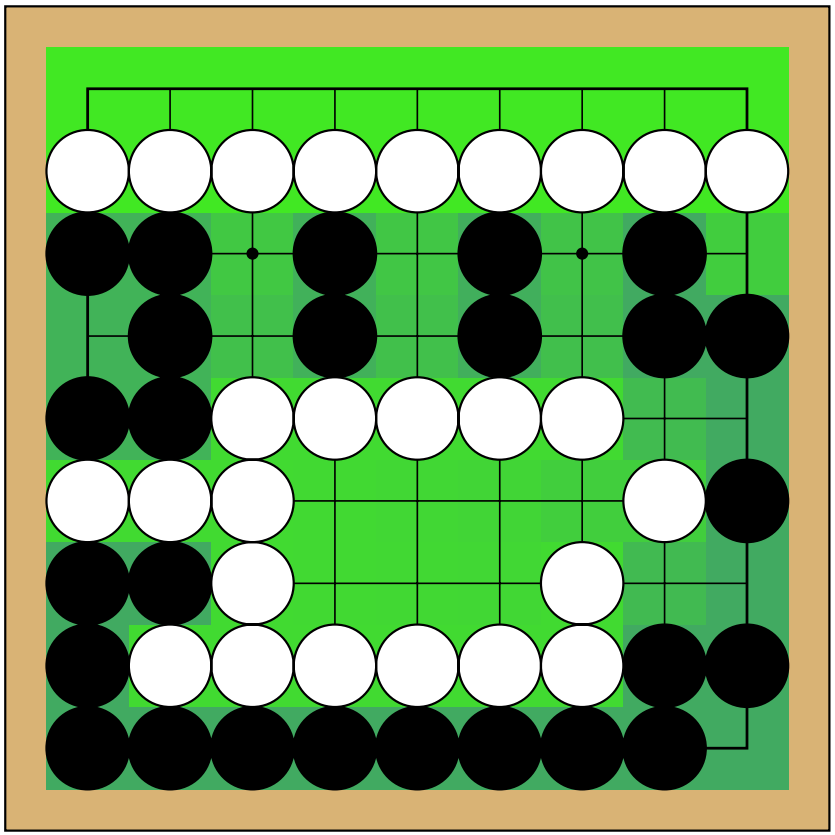


no patterns

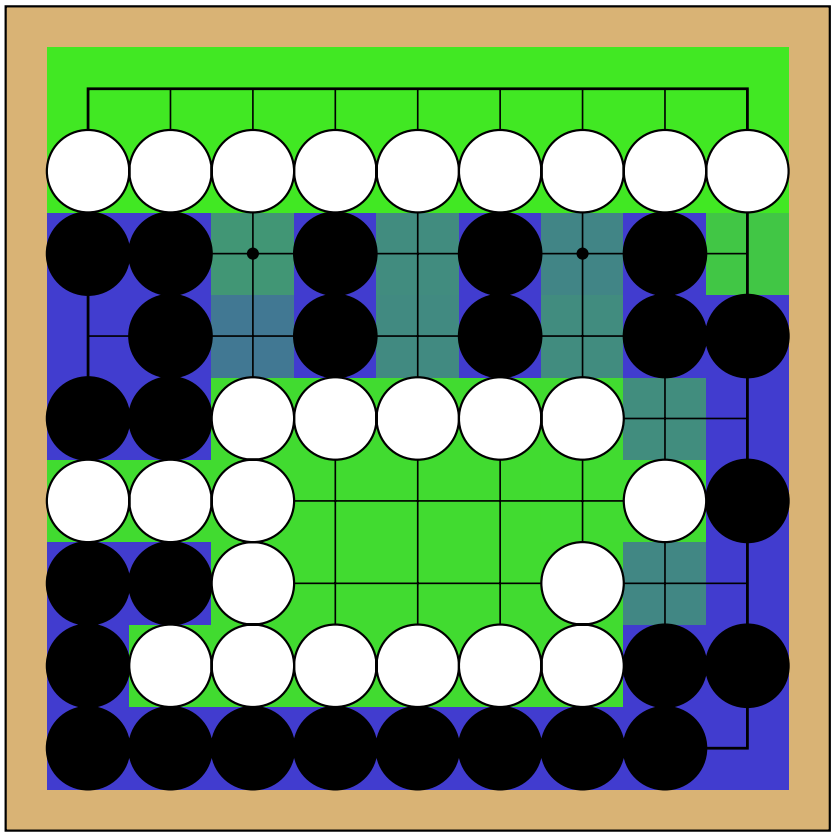


patterns

Comparison 2



no patterns



patterns

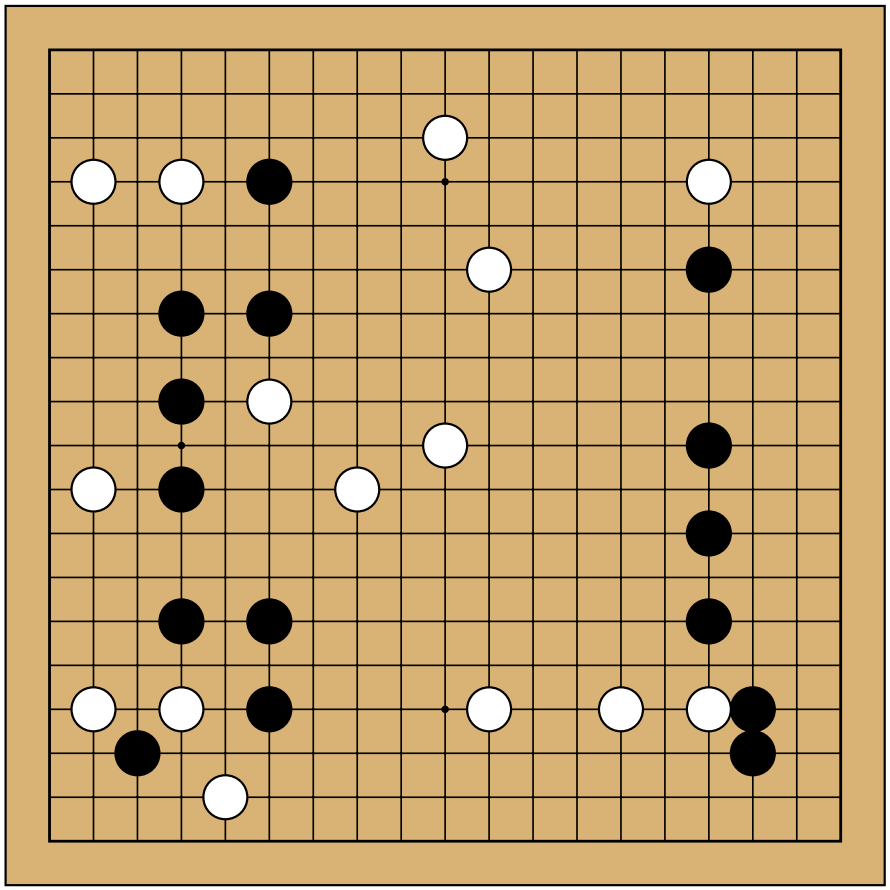
Progressive Widening

- Sort moves with patterns
- Keep best moves only
- Progressively add more

Playing Strength

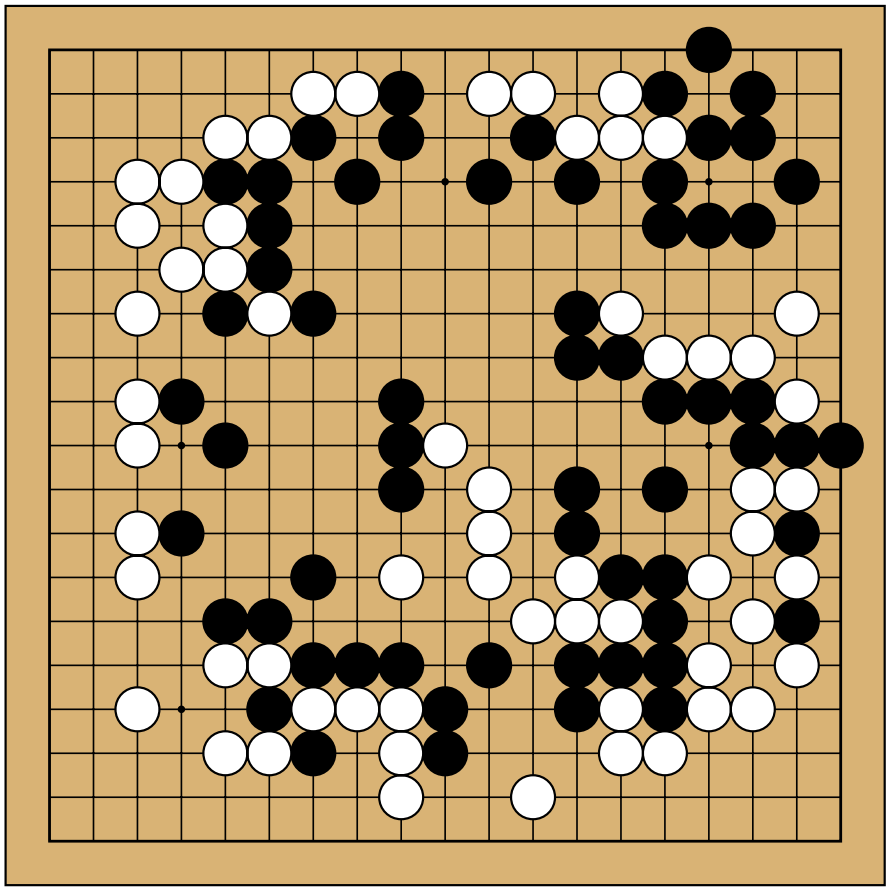
- Stronger than classical programs on 19×19
- Ranked 2k on KGS

Crazy Fuseki



○ MoGo
● Crazy Stone

Play in the Center



○ GNU Go
● Crazy Stone

Win by 0.5, Lose by a lot

○ Crazy Stone
● Jimmy

Speculative Attacks: Provoke Opponent Blunder

○ Go Intellect
● Crazy Stone

Speculative Attacks: Another Tricky Move

○ Miel (human)
● Crazy Stone

Ugly Blunder

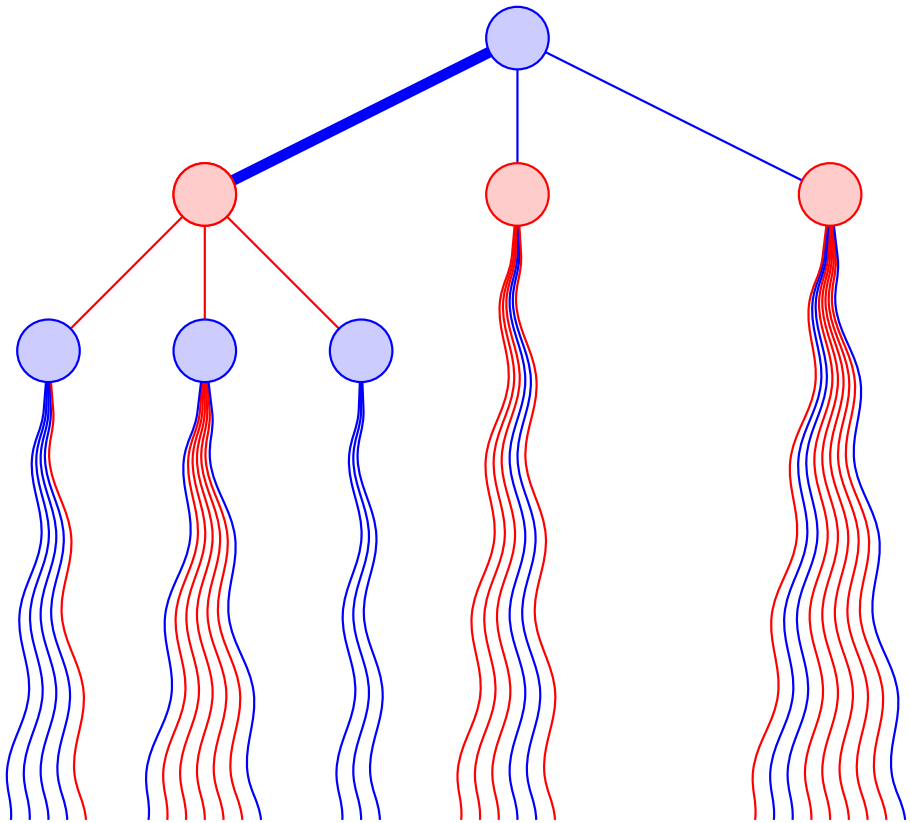
○ Crazy Stone
● Human

Future of Monte-Carlo Search

Improving Crazy Stone further

- More knowledge: playouts + progressive widening
- Adaptive playouts

Adaptive playouts



adaptive UCB policy

~~static~~ playout policy

Interesting ideas in RLGO (David Silver)

Future of Monte-Carlo Search

Application to Other Domains

- Other games (Hex, Clobber)
- Automated book learning (for chess?)
- Automated Planning in general

If You Wish to Know More

`http://remi.coulom.free.fr/Hakone2007/`

- Download these slides
- Download papers
- Connect to KGS and play against Crazy Stone