Computer-Go...
...is not only for Go.

Vincent Berthier, Amine Bourki, Mathieu Coulm, Guillaume Chaslot, Christophe Fiter, Sylvain Gelly, Jean-Baptiste Hooock, Rémi Munos, Julien Pérez, Arpad Rimmel, Philippe Rolet, Olivier.Teytaud@inria.fr, Paul Vayssière, Yizao Wang, Ziqin Yu (et al :-)

TAO, Inria-Futurs, Cnrs 8623, Lri, Univ. Paris-Sud, Digiteo Labs, Pascal Network of Excellence; with the help of NUTN (Taiwan).

Korea,
August 2009.
Outline

- Progress in computer-Go
- Algorithms
- Just other bricks in the wall?
- Beyond Go
Computers vs Humans in Go

1998: ManyFaces looses with 29 stones against M. Mueller

2007: win against a pro (5p) 9x9 en blitz MoGo
2008: win against a pro (5p) 9x9 MoGo
2009: win against a pro (5p) 9x9 noir MoGo

2008: win against a pro (8p) 19x19, H9 MoGo
2008: win against a pro (4p) 19x19, H8 CrazyStone
2008: win against a pro (4p) 19x19, H7 CrazyStone
2009: win against a pro (9p) 19x19, H7 MoGo
2009: win against a pro (1p) 19x19, H6 MoGo

==> very clear progress, not only hardware improvement, but still 6 stones at least!
Outline

- Progress in computer-Go
- **Algorithms**
- Just other bricks in the wall?
- Beyond Go
The race: many improvements

- The main idea: Monte-Carlo with growing tree
  - UCT
  - Bandit-Based Monte-Carlo Planning
  - Monte-Carlo Tree Search
- Many improvements
The main ideas

- Random simulations
  (Bruegman, 93)

- A tree of possible futures, increasing along simulations
  (several simultaneous papers, 2006)
UCT (Upper Confidence Trees)

Current state → Tree Policy

Coulom (06)
Chaslot, Saito & Bouzy (06)
Kocsis Szepesvari (06)

Default Policy
UCT

Current state

3/5

2/3

0/1

0/1

1/1

Tree Policy

Default Policy

1
UCT

Current state

0/1 → 1/1 → 4/5 → 5/7 → 6/10 → 0/2 → 0/1

Kocsis & Szepesvari (06)
Exploitation ...

Current state → 6/10

Diagram with nodes labeled 0/1, 1/1, 4/5, 5/7, 0/2, 2/3, and 0/1.
... or exploration?

Current state →

6/10

5/7  4/5  2/3

0/1  1/1  0/1  1/1  0/1  0/2

choose the move maximizing:

\[
\log(\text{nb simulations of all moves}) \quad \text{empirical probability of win + squareRoot} \left( \text{nb simulations of this move} \right)
\]
choose the move maximizing:

\[ \log(\text{nb simulations of all moves}) + \sqrt{\text{nb simulations of this move}} \]
choose the move maximizing:

\[
\log\left(\frac{2+7}{0}\right) + \sqrt{\text{nb simulations of this move}}
\]
choose the move maximizing:

\[
\log \left( \frac{2+7}{0} \right) + \text{squareRoot} \left( \frac{\text{--------------------------}}{2} \right)
\]
Improvements

- Much better default policy (in MoGo, default policy by Y. Wang ==> “sequence-like simulations” now more or less in all efficient programs)
- Multi-core Parallelization
- Message passing Parallelization
- Bias in the tree (patterns, rules)
- RAVE values
- Opening book by meta-UCT
- ...

ubuntu
Outline

- Progress in computer-Go
- Algorithms
- Just other bricks in the wall?
- Beyond Go
Just other bricks in the wall?

- Recent improvements = big improvement in self-play
  - Parallelization
  - Opening books
- Does it lead to a huge progress against humans?
- No - some situations are very poorly handled by all programs.
For sure, MoGo won one game with H7 against a top pro (9p, winner of LG Cup).

But MoGo then lost 3 games with H7.

Zen and ManyFaces recently also lost their H7 games against 9p players.

Stupid mistakes in specific situations (life&death in corners, semeai).

In 9x9, some wins 3 years ago, but still many losses in spite of improvements.

Strong programs very close to each other.
Just other bricks in the wall?

===> many improvements, closer and closer to the wall \(\rightarrow\) but no "big" improvement;

===> we need some massive change for "long term effects" (life&death, semeai);

===> many trials, no success.

(conditional Monte-Carlo, statistical bias...
Outline

- Progress in computer-Go
- Algorithms
- Just other bricks in the wall?
- Beyond Go
Beyond Go

- After all, is it worth all this work?
Beyond Go

- After all, is it worth all this work?
- No, it's only a game.

No.
Beyond Go

- After all, is it worth all this work?
- No, it's only a game.
- Yes, it's for many other applications: all discrete-time control problems.

Yes.
Beyond Go

- After all, is it worth all this work?
- No, it's only a game.
- Yes, it's for many other applications: all discrete-time control problems
- But perhaps we need more experiments beyond Go (already notable successes!) and extensions of UCT to other frameworks.

Yes, but ...
High-dimensional discrete time control

A main application: the management of many energy stocks in front of randomness

- At each time step we see random outcomes
- We have to take decisions
- After H time steps, we observe a cost
Biblio

- Bandits: Lai, Robbins, Auer, Cesa-Bianchi...
- UCT: Kocsis, Szepesvari, Coquelin, Munos...
- MCTS (Go): Coulom, Chaslot, Fiter, Gelly, Hooock, Silver, Muller, Pérez, Rimmel, Wang...
- Tree + DP for industrial application: Péret, Garcia...
- Bandits with infinitely many arms: Audibert, Coulom, Munos, Wang..
- Applications far from Go: Rolet, Teytaud (F), Rimmel, De Mesmay ...
- Links with “macro-actions”