Nouvelles propositions pour un système cognitif émergent

André Fabbri, Frédéric Armetta, Éric Duchêne and Salima Hassas

LIRIS - Équipe GrAMA

Réunion d'équipe - 19 octobre 2012

Outline

Problem description

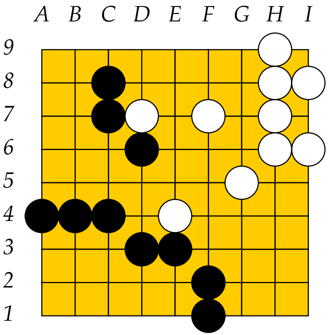


Figure: Go 9x9 game state

Combinatorial Games:

- *Zero-sum*
- *Perfect information*
- *Sequential*
- *Determinist*
- *Discrete*

Problem description

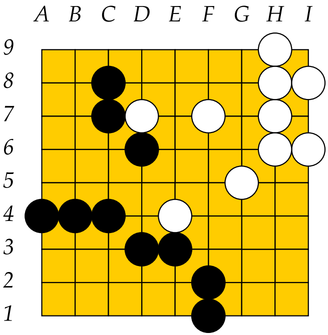


Figure: Go 9x9 game state

Combinatorial Games:

- *Zero-sum*
- *Perfect information*
- *Sequential*
- *Determinist*
- *Discrete*

The game of Go:

- Huge space state *[Bouzy and Cazenave, 2001]*
  19x19 $\simeq 10^{160}$, *9x9* $\simeq 10^{40}$
- No simple in-game evaluation
  [Wang and Gelly, 2007]

Problem description

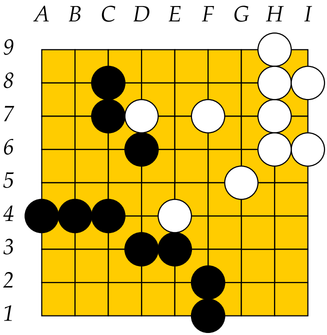


Figure: Go 9x9 game state

Combinatorial Games:

- *Zero-sum*
- *Perfect information*
- *Sequential*
- *Determinist*
- *Discrete*

The game of Go:

- Huge space state *[Bouzy and Cazenave, 2001]*
  19x19 $\simeq 10^{160}$, *9x9* $\simeq 10^{40}$
- No simple in-game evaluation
  [Wang and Gelly, 2007]

High volume & Complex data

Domain expert knowledge

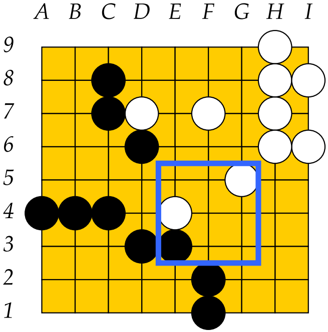


Figure: Go domain knowledge

"Low level" knowledge ≃ Moves

- Spatial stone disposition *(patterns)*
- Automatic reply *(opening book)*

Domain expert knowledge

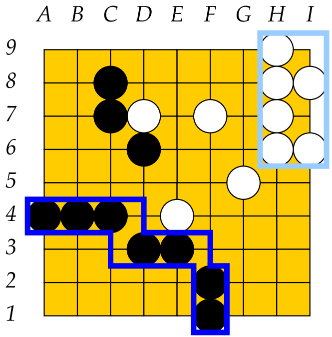


Figure: Go domain knowledge

"Low level" knowledge ≃ Moves

- Spatial stone disposition *(patterns)*
- Automatic reply *(opening book)*

"High level" knowledge ≃ Concepts

- Complex stone structure *(group,eye)*
- Tactical objective *(atari rule, semeai)*

Domain expert knowledge

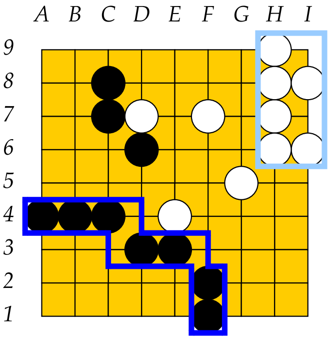


Figure: Go domain knowledge

"Low level" knowledge $\simeq$ Moves

- Spatial stone disposition *(patterns)*
- Automatic reply *(opening book)*

"High level" knowledge $\simeq$ Concepts

- Complex stone structure *(group,eye)*
- Tactical objective *(atari rule, semeai)*

More knowledge $\Rightarrow$ less effective

" 'improving a program' by adding knowledge makes it weaker in practice" [Müller, 2002]

Self-acquired knowledge

Learning methods and data

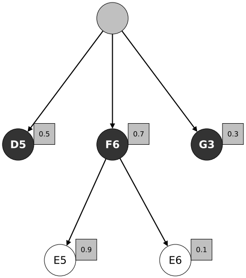
- Off-line : game databases, simulations
  TD(λ), Bayesian learn [Stern et al., 2006]
- On-line : current game, simulations
  TD(λ), MCTS [Coulom, 2007b]

Self-acquired knowledge

Learning methods and data

- Off-line : game databases, simulations
  TD(λ), Bayesian learn [Stern et al., 2006]
- On-line : current game, simulations
  TD(λ), MCTS [Coulom, 2007b]

Monte Carlo Tree Search



Self-acquired knowledge

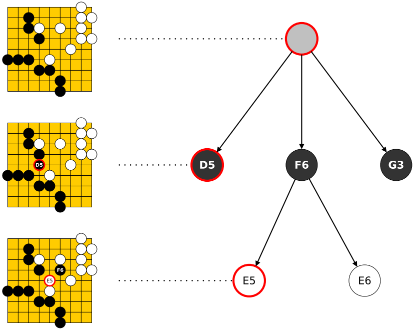
Learning methods and data

- Off-line : game databases, simulations
  TD(λ), Bayesian learn [Stern et al., 2006]
- On-line : current game, simulations
  TD(λ), MCTS [Coulom, 2007b]

Monte Carlo Tree Search

Tree Search 1 node = 1 future state

- *relative to the current state*
- *fine grained*
- *low level*



Self-acquired knowledge

Learning methods and data

- Off-line : game databases, simulations
  TD(λ), Bayesian learn [Stern et al., 2006]
- On-line : current game, simulations
  TD(λ), MCTS [Coulom, 2007b]

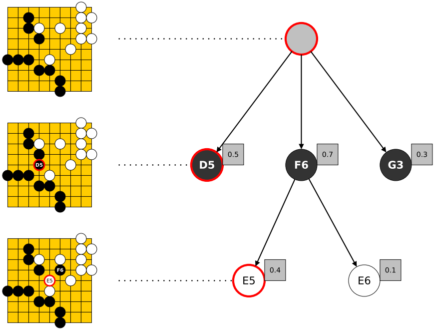
Monte Carlo Tree Search

Tree Search 1 node = 1 future state

- *relative to the current state*
- *fine grained*
- *low level*

Monte Carlo = node evaluation

- *Mean (standard estimate)*
- *UCB (balance EvE)*
  [Kocsis and Szepesvári, 2006]
- *RAVE (fast convergence)*
  [Gelly and Silver, 2011]



Learning policy iteration

MCTS iteration

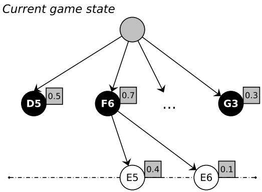


Figure: MCTS procedure

Learning policy iteration

MCTS iteration

1. *Descent*

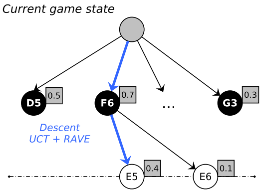


Figure: MCTS procedure

Learning policy iteration

MCTS iteration

1. *Descent*
2. *Roll-out*

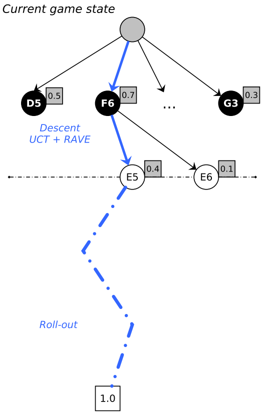


Figure: MCTS procedure

Learning policy iteration

MCTS iteration

1. *Descent*
2. *Roll-out*
3. *Update*

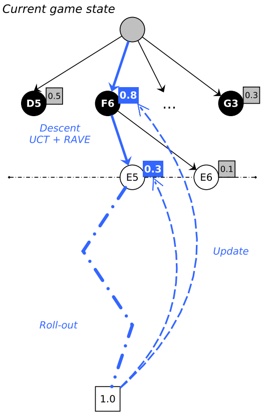


Figure: MCTS procedure

Learning policy iteration

MCTS iteration

1. *Descent*
2. *Roll-out*
3. *Update*
4. *Growth*

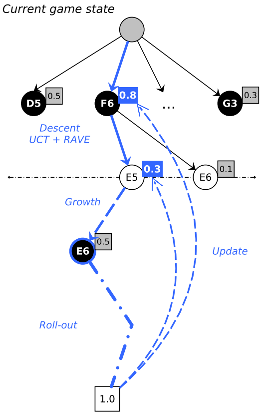


Figure: MCTS procedure

Learning policy iteration

MCTS iteration

1. *Descent*
2. *Roll-out*
3. *Update*
4. *Growth*

Interest

- Consistency
- Aheuristic
- Anytime
- Asymmetric
- Parallel

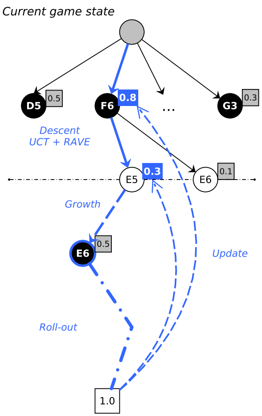


Figure: MCTS procedure

Learning policy iteration

MCTS iteration

1. *Descent*
2. *Roll-out*
3. *Update*
4. *Growth*

Interest

- Consistency
- Aheuristic
- Anytime
- Asymmetric
- Parallel

Limitations

- No capitalization
- "Horyzon effect"
- Irrelevant roll-out
- Scale limited

[Browne et al., 2012, Drake, 2009, Lee et al., 2010]

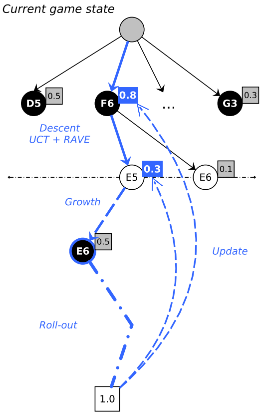


Figure: MCTS procedure

Expert *Roll-out* policy

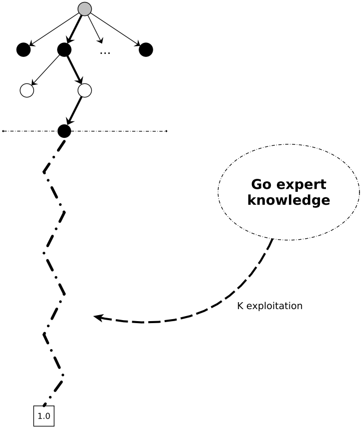


Figure: MCTS with expert K

Go expert knowledge

- Stone patterns *[Wang and Gelly, 2007]*
- Tactical rules *[Enzenberger et al., 2009]*

Amateur level $\longrightarrow$ Professional level

Expert *Roll-out* policy

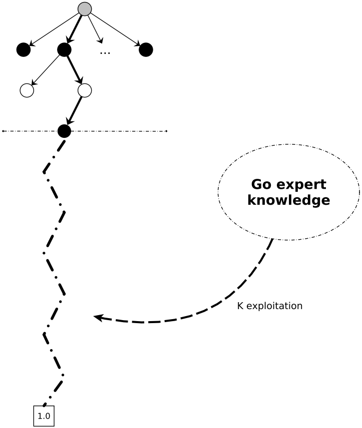


Figure: MCTS with expert K

Go expert knowledge

- Stone patterns [Wang and Gelly, 2007]
- Tactical rules [Enzenberger et al., 2009]

Amateur level ⟶ Professional level

Limitations

- Heavy *Roll-out* ⇒ bad learning
- Static knowledge
- Difficult to tune

Expert *Roll-out* policy

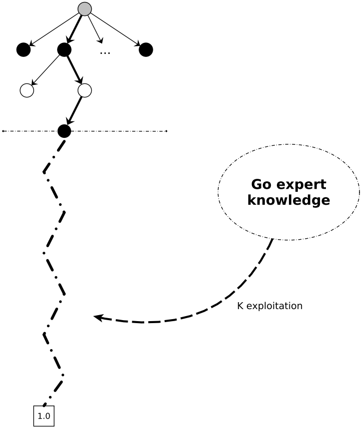


Figure: MCTS with expert K

Go expert knowledge

- Stone patterns [Wang and Gelly, 2007]
- Tactical rules [Enzenberger et al., 2009]

Amateur level ⟶ Professional level

Limitations

- Heavy *Roll-out* ⇒ bad learning
- Static knowledge
- Difficult to tune

"Adding further Go knowledge [. . .] has proven to be surprisingly difficult."

[Silver and Tesauro, 2009]

Enhanced *Roll-out* policy

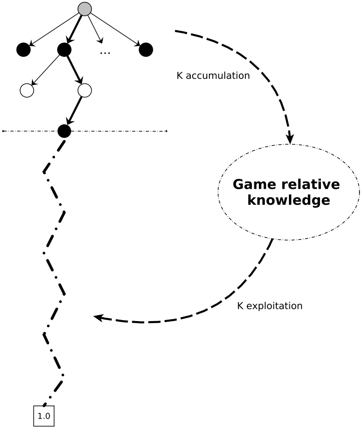


Figure: MCTS with game relative K

Automatic tuning methods

Go expert K Simulation Balancing

[Huang et al., 2011]

Enhanced *Roll-out* policy

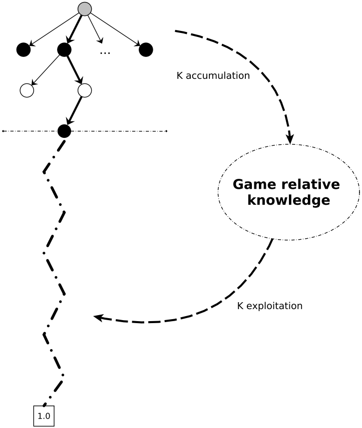


Figure: MCTS with game relative K

Automatic tuning methods

Go expert K Simulation Balancing

[Huang et al., 2011]

Game relative knowledge methods

Tree search K Pool RAVE *[Rimmel et al., 2011]*

New K forms CMC *[Rimmel and Teytaud, 2010]*

LGRF *[Baier and Drake, 2010]*

Enhanced *Roll-out* policy

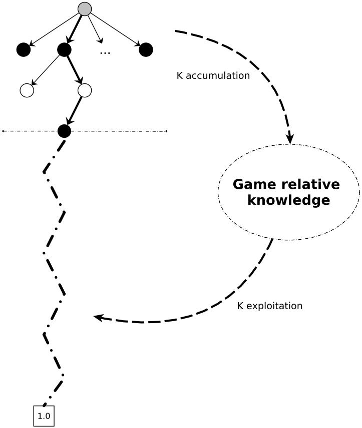


Figure: MCTS with game relative K

Automatic tuning methods

Go expert K Simulation Balancing
[Huang et al., 2011]

Game relative knowledge methods

Tree search K Pool RAVE *[Rimmel et al., 2011]*

New K forms CMC *[Rimmel and Teytaud, 2010]*
LGRF *[Baier and Drake, 2010]*

Limitations

- Low level K
- Limited scope K structure

Reverse tree search structure

future game state $\rightarrow$ *sequence of action preceeding a reply*

Reverse tree search structure

future game state $\rightarrow$ *sequence of action preceeding a reply*

Background History Reply tree

Reply tree 1 root = 1 reply

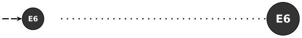


Figure: Background History Reply tree

Reverse tree search structure

future game state $\rightarrow$ *sequence of action preceeding a reply*

Background History Reply tree

Reply tree 1 root = 1 reply

Background History node = sequence

- *low level*
- *heavy grained*

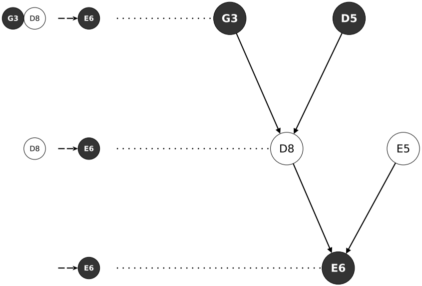


Figure: Background History Reply tree

Reverse tree search structure

future game state $\rightarrow$ sequence of action preceeding a reply

Background History Reply tree

Reply tree 1 root = 1 reply

Background History node = sequence

- *low level*
- *heavy grained*

Evaluation Mean, UCB

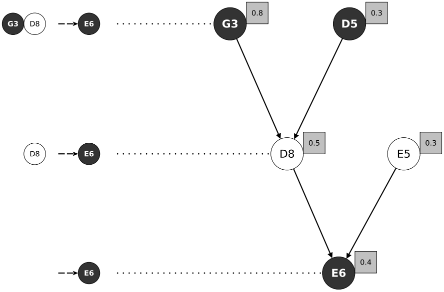


Figure: Background History Reply tree

Reverse tree search structure

future game state → sequence of action preceeding a reply

Background History Reply tree

Reply tree 1 root = 1 reply

Background History node = sequence
- *low level*
- *heavy grained*

Evaluation Mean, UCB

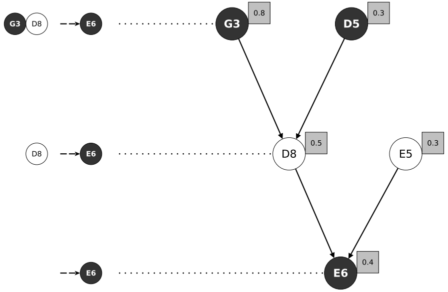


Figure: Background History Reply tree

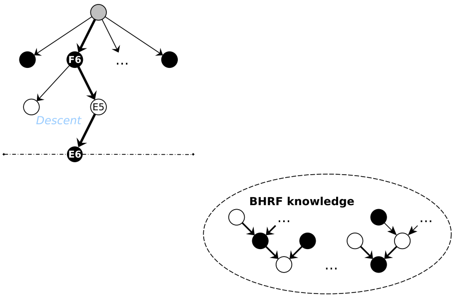
Background History Reply Forest

n **actions possible = forest**

Knowledge exploitation

MCTS iteration

1. *Descent*

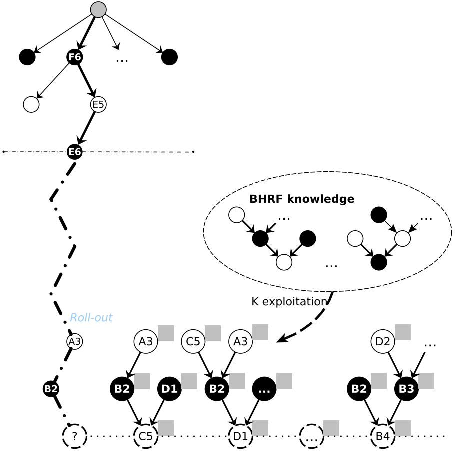


Knowledge exploitation

MCTS iteration

1. *Descent*
2. *Roll-out*

Roll-out policy



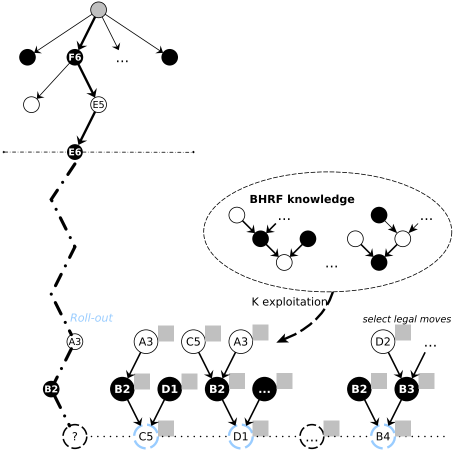
Knowledge exploitation

MCTS iteration

1. *Descent*
2. *Roll-out*

Roll-out policy

1. Whole board legal moves



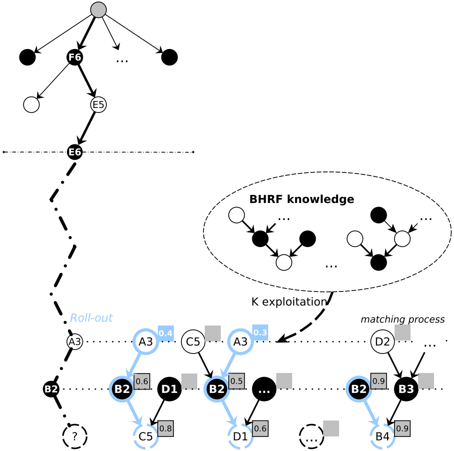
Knowledge exploitation

MCTS iteration

1. *Descent*
2. *Roll-out*

Roll-out policy

1. Whole board legal moves
2. Longest matching sequence



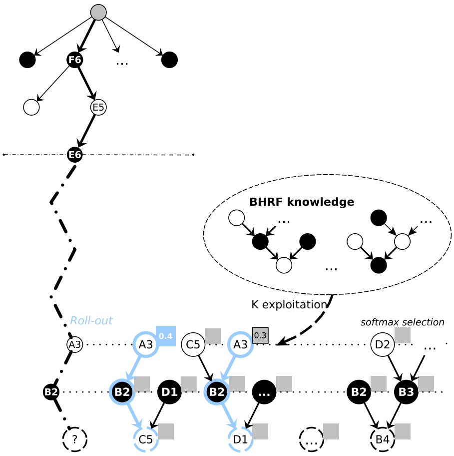
Knowledge exploitation

MCTS iteration

1. *Descent*
2. *Roll-out*

Roll-out policy

1. Whole board legal moves
2. Longest matching sequence
3. UCB based selection



Knowledge exploitation

MCTS iteration

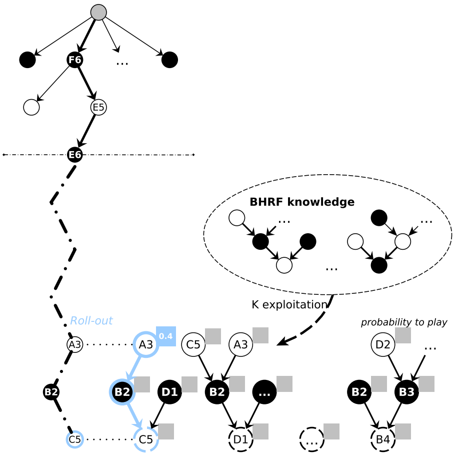
1. *Descent*
2. *Roll-out*

Roll-out policy

1. Whole board legal moves
2. Longest matching sequence
3. UCB based selection

Interest

- Global search - context
- Diversity and exploration

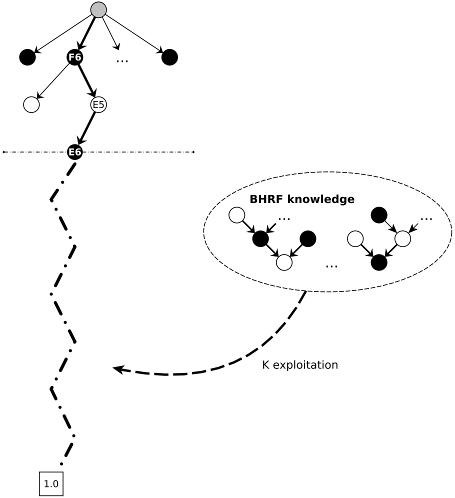


Knowledge accumulation

MCTS iteration

1. *Descent*
2. *Roll-out*

Update and *Growth* policy



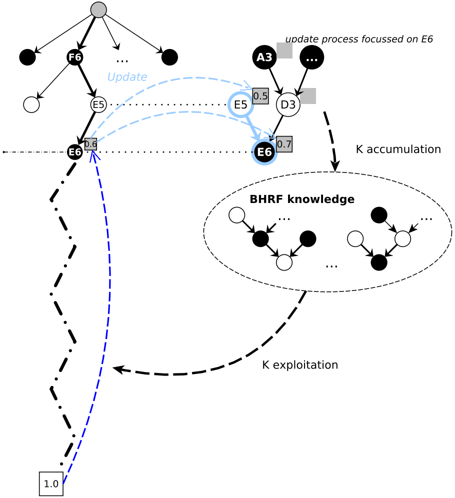
Knowledge accumulation

MCTS iteration

1. *Descent*
2. *Roll-out*
3. *Update*

Update and *Growth* policy

- *Descent* sequence



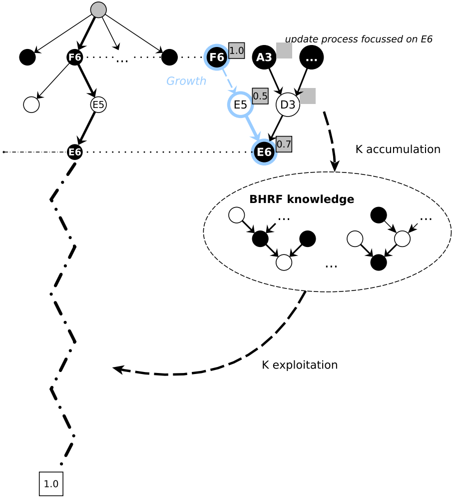
Knowledge accumulation

MCTS iteration

1. *Descent*
2. *Roll-out*
3. *Update*
4. *Growth*

Update and *Growth* policy

- *Descent* sequence
- Progressive growth



Knowledge accumulation

MCTS iteration

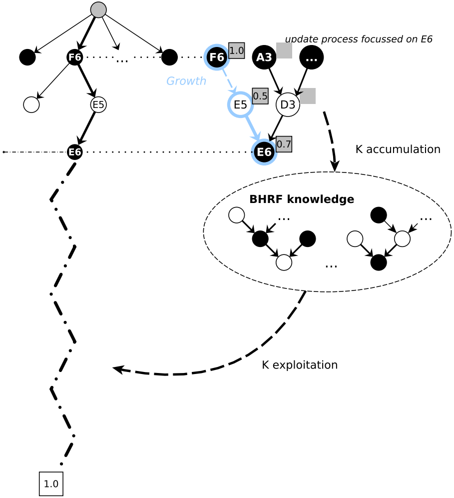
1. *Descent*
2. *Roll-out*
3. *Update*
4. *Growth*

Update and *Growth* policy

- *Descent* sequence
- Progressive growth

Interest

- Extract tree K incrementally
- Capitalization



Model validation

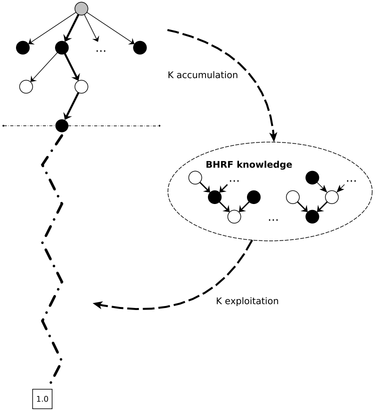


Figure: MCTS with BHRF

Model parameters

MCTS : $MC \rightarrow$ *K consistence*

BHRF : $\alpha \rightarrow$ *K exploitation*

depth $\rightarrow$ *K pertinence*

Model validation

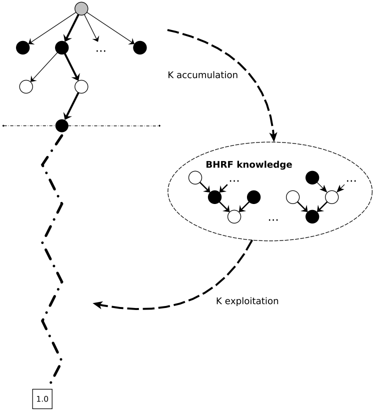


Figure: MCTS with BHRF

Model parameters

MCTS : *MC* $\rightarrow$ *K consistence*

BHRF : $\alpha \rightarrow$ *K exploitation*

depth $\rightarrow$ *K pertinence*

Experimental setup

- Fuego *[Enzenberger et al., 2009]*
- MCTS + BHRF vs MCTS

  no expert knowledge
- 9x9 Goban - *1 thread*

Model validation

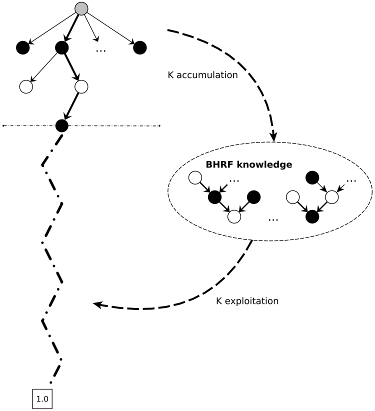


Figure: MCTS with BHRF

Model parameters

MCTS : *MC* $\rightarrow$ *K consistence*

BHRF : $\alpha \rightarrow$ *K exploitation*

depth $\rightarrow$ *K pertinence*

Experimental setup

- Fuego *[Enzenberger et al., 2009]*
- MCTS + BHRF vs MCTS

  no expert knowledge
- 9x9 Goban - *1 thread*

Results

- Outperform plain UCT *(High MC)*
- Time execution *(10 $\times$ more)*

Future works

- **Temporal** *(history sequences)*
- Spatial *(emergent pattern)*
- Rule-based *(life and death)*

Future works

- **Temporal** *(history sequences)*
- Spatial *(emergent pattern)*
- Rule-based *(life and death)*

Super-structure

Future works

- **Temporal** *(history sequences)*
- Spatial *(emergent pattern)*
- Rule-based *(life and death)*

Super-structure

MCTS $\Rightarrow$ generates its own traces

Future works

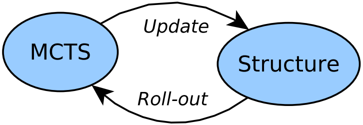
- **Temporal** *(history sequences)*
- Spatial *(emergent pattern)*
- Rule-based *(life and death)*

⇒ **Super-structure**

MCTS $\Rightarrow$ generates its own traces

Aims and directions

- Autocatalytic learning
- Towards high level K *(emerging K ?)*



Future works

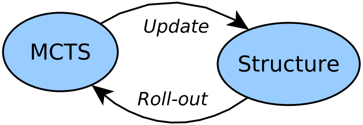
- **Temporal** *(history sequences)*
- Spatial *(emergent pattern)*
- Rule-based *(life and death)*

Super-structure

MCTS ⇒ generates its own traces

Aims and directions

- Autocatalytic learning
- Towards high level K *(emerging K ?)*



Questions ?

References I

Baier, H. and Drake, P. (2010).
The Power of Forgetting: Improving the Last-Good-Reply Policy in Monte-Carlo go.
IEEE Transactions on Computational Intelligence and AI in Games, 2(4):303–309.

Bouzy, B. and Cazenave, T. (2001).
Computer Go: an AI oriented survey.
Artificial Intelligence, 132(1):39–103.

Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012).
A Survey of Monte Carlo Tree Search Methods.
Computational Intelligence and AI in Games, IEEE Transactions on, 4(1):1–43.

Coulom, R. (2007a).
Computing Elo Ratings of Move Patterns in the Game of Go.
In *Computer Games Workshop, Amsterdam, The Netherlands*.

Coulom, R. (2007b).
Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search.
Computers and Games, pages 72–83.

Drake, P. (2009).
The Last-Good-Reply Policy for Monte-Carlo Go.
International Computer Games Association Journal, 32(4):221–227.

Enzenberger, M. (2003).
Evaluation in Go by a Neural Network using Soft Segmentation.
Advances in Computer Games: Many Games, Many Challenges, pages 97–108.

References II

Enzenberger, M., Muller, M., Arneson, B., and Segal, R. (2009).
FUEGO - an Open-Source Framework for Board Games and Go Engine Based on Monte-Carlo Tree Search.
IEEE Transactions on Computational Intelligence and AI in Games, 2(4):259–270.

Gelly, S., Kocsis, L., Schoenauer, M., Sebag, M., Silver, D., Szepesvári, C., and Teytaud, O. (2012).
The Grand Challenge of Computer Go: Monte Carlo Tree Search and Extensions.
Communications of the ACM, 55(3):106–113.

Gelly, S. and Silver, D. (2011).
Monte-Carlo Tree Search and Rapid Action Value Estimation in Computer Go.
Artificial Intelligence.

Huang, S., Coulom, R., and Lin, S. (2011).
Monte-Carlo Simulation Balancing in Practice.
Computers and Games, pages 81–92.

Kocsis, L. and Szepesvári, C. (2006).
Bandit based Monte-Carlo Planning.
Machine Learning: ECML 2006, pages 282–293.

Lee, C., Rimmel, A., Teytaud, O., Tsai, S., Wang, M., and Yen, S. (2010).
Current Frontiers in Computer Go.
Computational Intelligence and AI in Games, IEEE Transactions on, 2(4):229–238.

Lee, C., Wang, M., Chaslot, G., Hoock, J., Rimmel, A., Teytaud, O., Tsai, S., Hsu, S., and Hong, T. (2009).
The Computational Intelligence of MoGo Revealed in Taiwan's Computer Go Tournaments.
Computational Intelligence and AI in Games, IEEE Transactions on, pages 73–89.

References III

Müller, M. (2002).
Computer go.
Artificial Intelligence, 134(1):145–179.

Rimmel, A. and Teytaud, F. (2010).
Multiple Overlapping Tiles for Contextual Monte Carlo Tree Search.
Applications of Evolutionary Computation, pages 201–210.

Rimmel, A., Teytaud, F., and Teytaud, O. (2011).
Biasing Monte-Carlo Simulations through RAVE Values.
Computers and Games, pages 59–68.

Silver, D., Sutton, R., and Müller, M. (2012).
Temporal-difference search in computer Go.
Machine Learning, pages 1–37.

Silver, D. and Tesauro, G. (2009).
Monte-Carlo Simulation Balancing.
In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 945–952. ACM.

Stern, D., Herbrich, R., and Graepel, T. (2006).
Bayesian Pattern Ranking for Move Prediction in the Game of Go.
In *Proceedings of the 23rd international conference on Machine learning*, pages 873–880. ACM.

Wang, Y. and Gelly, S. (2007).
Modifications of UCT and sequence-like simulations for Monte-Carlo Go.
In *IEEE Symposium on Computational Intelligence and Games, Honolulu, Hawaii*, pages 175–182.

Experimental results

MC / α	1000	10000
1	58.7 ± 2.49	63.4 ± 2.99
100	58.5 ± 2.49	68.8 ± 2.87

MCTS parameter *(depth = 2)*

- MC=1000 : no improvement
- MC=10000 : improvement over α

Experimental results

MC / α	1000	10000
1	58.7 ± 2.49	63.4 ± 2.99
100	58.5 ± 2.49	68.8 ± 2.87

MCTS parameter *(depth = 2)*

- MC=1000 : no improvement
- MC=10000 : improvement over α

BHRF scales with consistent K

Experimental results

MC \ α	1000	10000
1	58.7 ± 2.49	63.4 ± 2.99
100	58.5 ± 2.49	68.8 ± 2.87

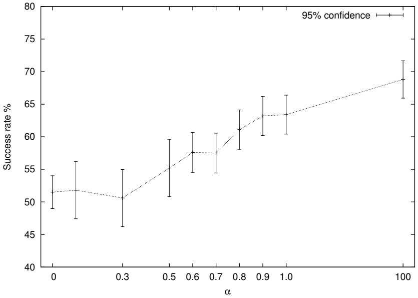


Figure: BHRF success rate over α
MC =10000, depth=2

MCTS parameter *(depth = 2)*

- MC=1000 : no improvement
- MC=10000 : improvement over α

BHRF scales with consistent K

BHRF parameters *(MC = 10000)*

- α : softmax policy $\longrightarrow$ exploration
- *depth* : not significant *(α=100)*

Experimental results

α \ MC	1000	10000
1	58.7 ± 2.49	63.4 ± 2.99
100	58.5 ± 2.49	68.8 ± 2.87

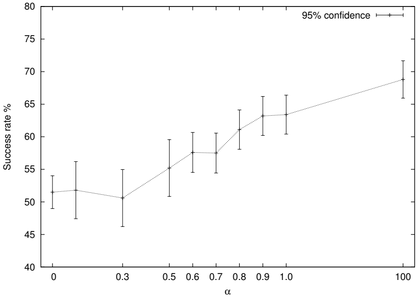


Figure: BHRF success rate over α
MC =10000, depth=2

MCTS parameter *(depth = 2)*

- MC=1000 : no improvement
- MC=10000 : improvement over α

BHRF scales with consistent K

BHRF parameters *(MC = 10000)*

- α : softmax policy $\longrightarrow$ exploration
- *depth* : not significant *(α=100)*

BHRF K improves baseline MCTS

Experimental results

MC / α	1000	10000
1	58.7 ± 2.49	63.4 ± 2.99
100	58.5 ± 2.49	68.8 ± 2.87

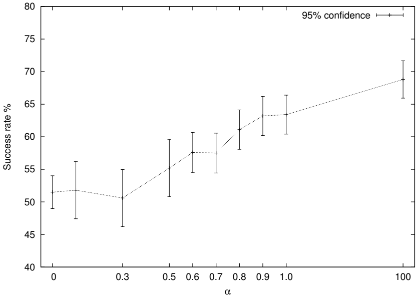


Figure: BHRF success rate over α
MC =10000, depth=2

MCTS parameter *(depth = 2)*

- MC=1000 : no improvement
- MC=10000 : improvement over α

BHRF scales with consistent K

BHRF parameters *(MC = 10000)*

- α : softmax policy $\longrightarrow$ exploration
- *depth* : not significant *(α=100)*

BHRF K improves baseline MCTS

Limitations

- Time execution *(up to 10 times slower)*
- BHRF *depth* $\longrightarrow$ policy ?