

モンテカルロ木探索

Monte-Carlo Tree Search

の理論と実践

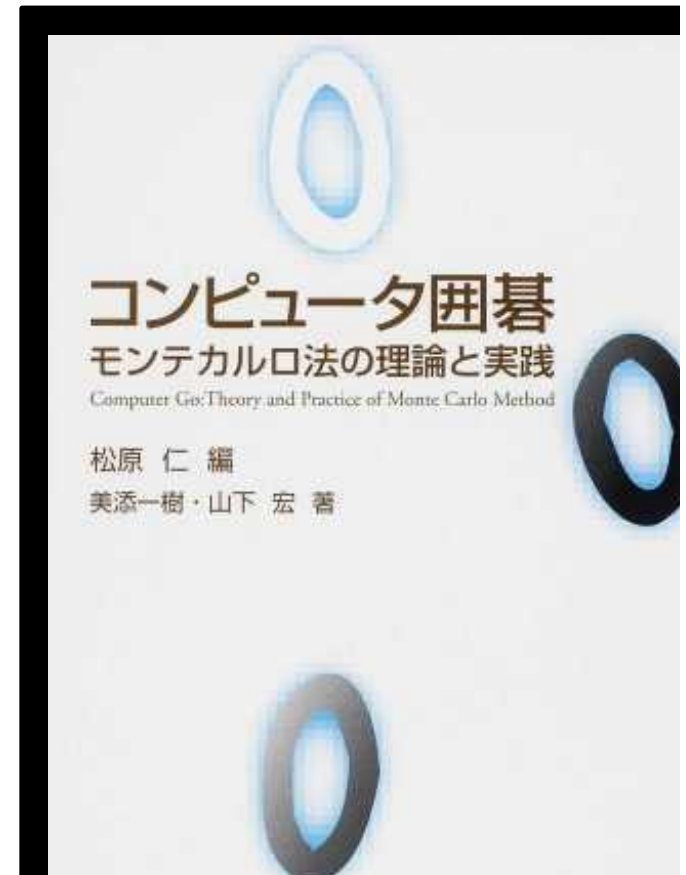
美添 一樹 Kazuki Yoshizoe

JST ERATO 湊離散構造処理系プロジェクト / 東工大

IBIS2014 企画セッション
離散アルゴリズムの機械学習応用

自己紹介と概要

- 最近の研究テーマ（あるいは世を忍ぶ仮の姿）
 - 探索アルゴリズムの大規模並列化
- 本当のテーマ
 - コンピュータでプロ囲碁棋士に勝つ
- 今日はモンテカルロ木探索の
 - 成立の経緯
 - 最初しばらく囲碁の話
 - 理論と実践
 - どちらも深くは立ち入らない
 - モンテカルロ木探索を
 - 使いたい人、理論を知りたい人
 - 両方に役立つと良いと思っています



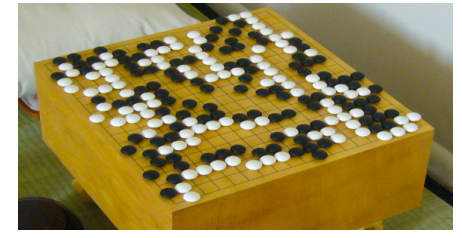
IBM 1997
DeepBlue Kasparov



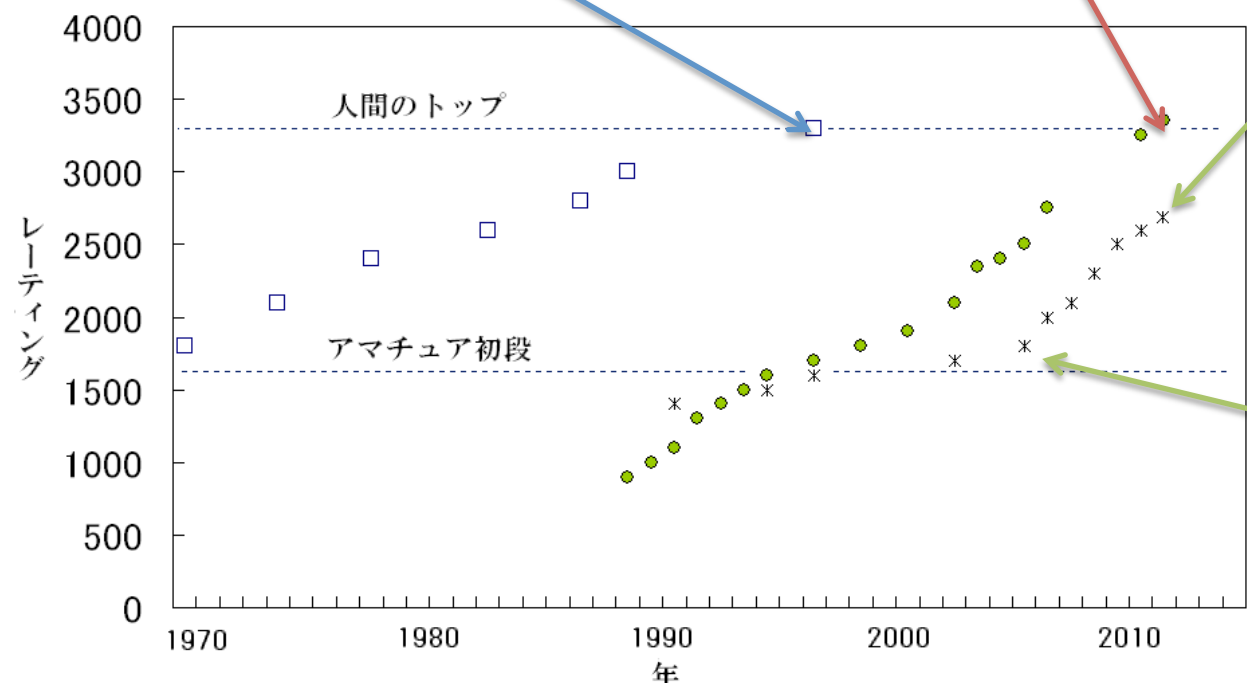
2012~14 電王戦



2013~14 電聖戦
4子でプロと2勝2敗



□ チェス ● 将棋 * 囲碁



モンテカルロ
木探索の発明

FPGAで将棋プログラムを作ってみるブログ
http://blog.livedoor.jp/yss_fpga/archives/53897129.html

2013年～14年 電聖戦

東進ハイスクール 東進衛星予備校

第1回 電聖戦

～日本棋院プロ棋士 vs. コンピュータ囲碁 公式定期戦～

公益財団法人日本棋院は、2012年6月に国立大学法人電気通信大学とコンピュータ囲碁の進歩に関する5年間の提携を結び、プロ棋士とコンピュータ囲碁との公式定期戦(『電聖戦(でんせいせん)』)を開催し、コンピュータ囲碁の棋力を測ることにしました。本イベントは、この提携を受け開催される第1回のイベントとなります。この対戦は今後少なくとも5年間は、定期的に行われる予定です。

※本イベント関連・掲載放映情報※

<イベント前>

- 11月23日朝日新聞 [「囲碁ソフト、プロと公式定期戦 3月、石田元名人に挑む」](#)
- 12月31日産経新聞 [「囲碁公式戦、来年3月開催 棋士VSコンピューター、白黒つける」](#)
- 3月19日朝日新聞 [「仏プログラムが優勝」](#)

<イベント後>

- 3月20日産経新聞 [「『アマ六段の力、天才かも』 囲碁棋士、コンピュータに敗れる 初の公式戦」](#)
- 3月21日西日本新聞 [「1勝1敗 白黒つかず 電聖戦、公開対局」](#)
- 3月21日毎日新聞 [「囲碁ソフト 初の対局 棋士に1勝1敗」](#)
- 3月21日毎日新聞 [「ルボ:第1回電聖戦 囲碁ソフトが3目勝ち 仏大学教員開発の「クレージー・ストーン」](#)

シーンII

- 4月 1日号週刊星 [「第1回電聖戦」](#)
- 4月 2日朝日新聞夕刊 [「電聖戦 強さともろさ 第1回電聖戦 プロに1勝1敗」](#)

<今後放映、掲載予定>

- 4月21日放映予定:「囲碁フォーカス」(NHK Eテレ)
- 放映未定:「Youは何しに日本へ?」(TV東京)

<対戦プロ棋士紹介>



石田芳夫九段
(24世本因坊秀芳)

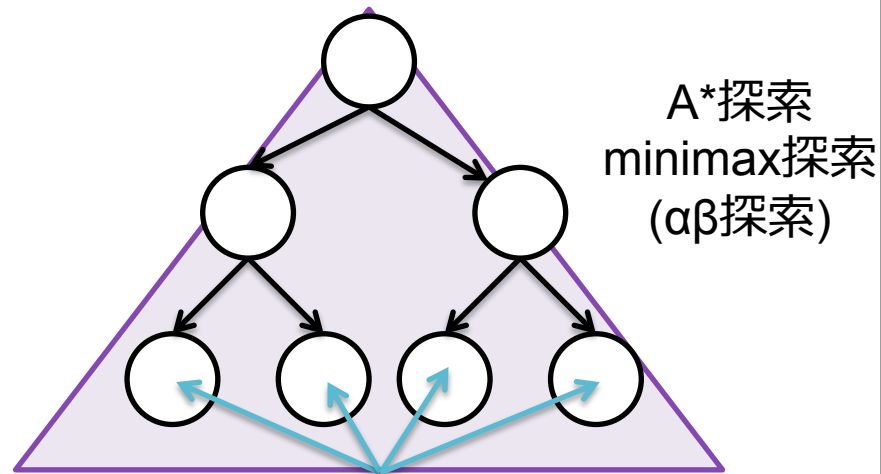
4子局は、将棋の角落ちより少し差があるハンデ

プロと4子局で対戦
2年連続で1勝1敗

参加プログラムは2年連続で
Zen と CrazyStone

電聖戦ウェブサイト
<http://entcog.c.ooco.jp/entcog/densei/>

従来の探索アルゴリズム



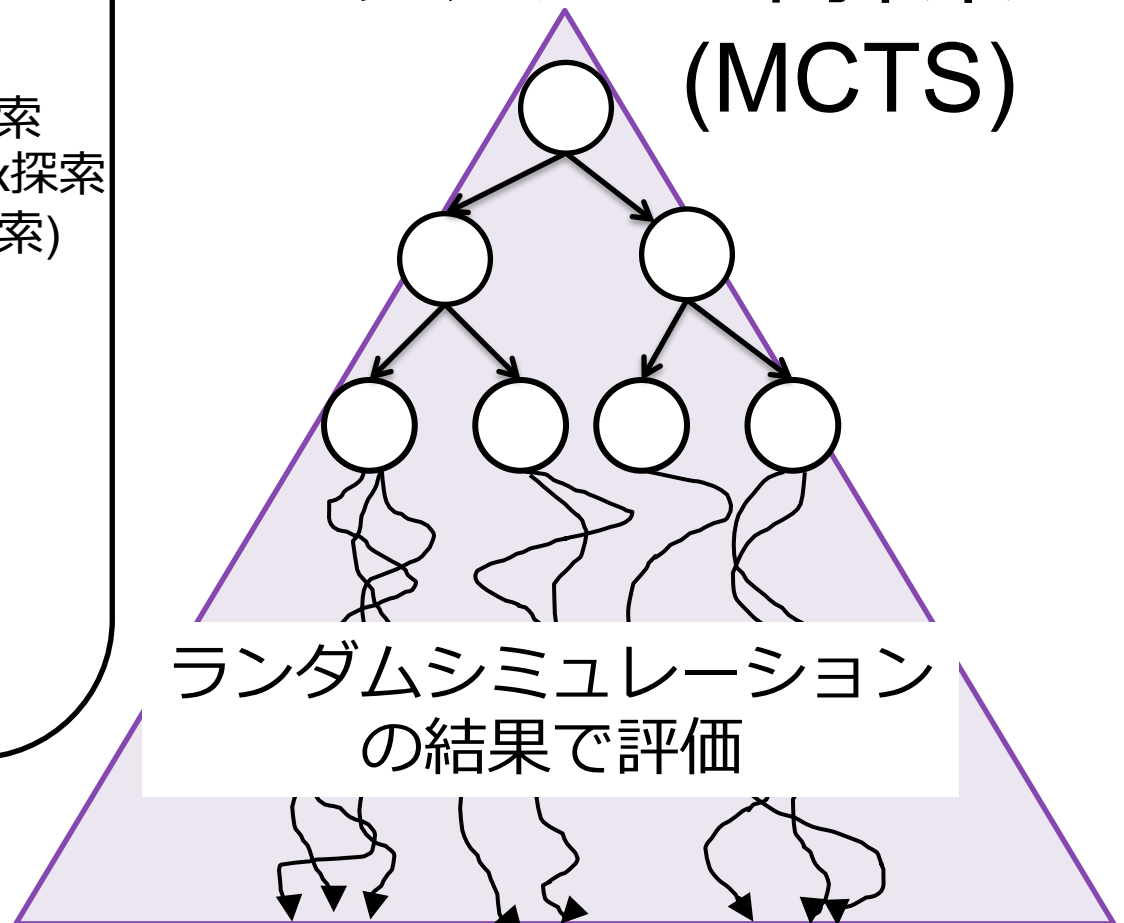
評価関数で評価

各節点の
数値の比較

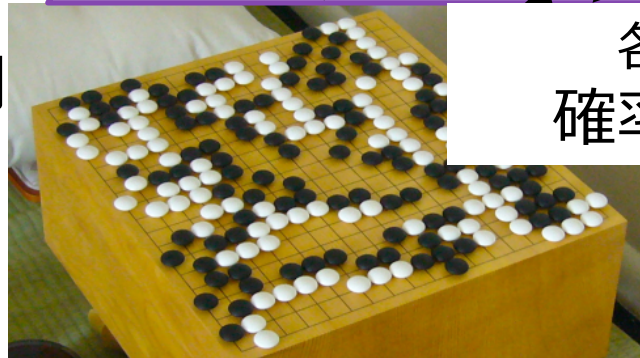
[Coulom 2006] CrazyStone

評価関数無しでも
探索できる画期的な発明
囲碁のために開発

モンテカルロ木探索 (MCTS)



各節点の持つ
確率分布の比較



囲碁の評価関数の難しさ

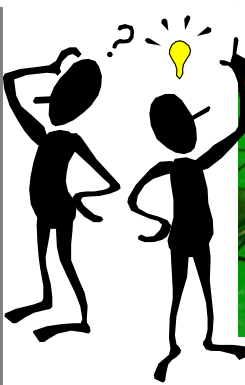
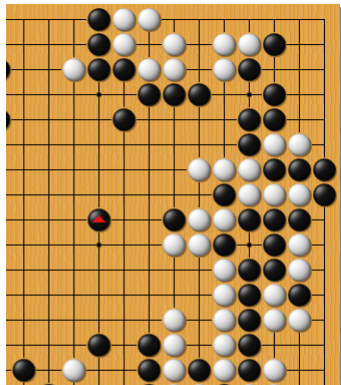


駒の価値のような
手がかりがない

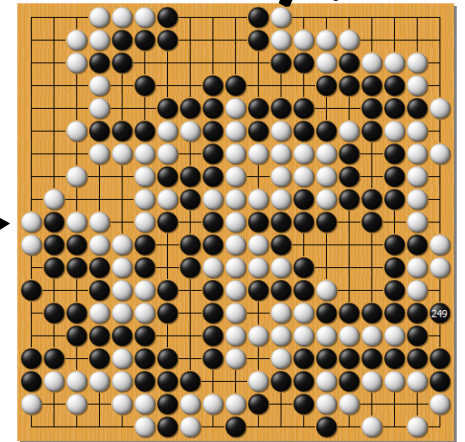
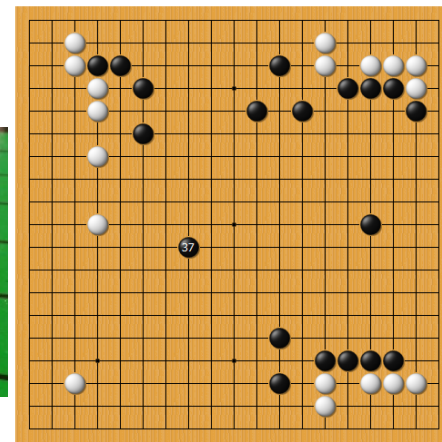
良い評価関数は、
正確で高速でないといけない

囲碁の評価関数を
作れた人は居ない

オセロの隅ほど
特徴のある場所がない



途中で終局図を
予想するのも困難





乱数を使って囲碁を打つ

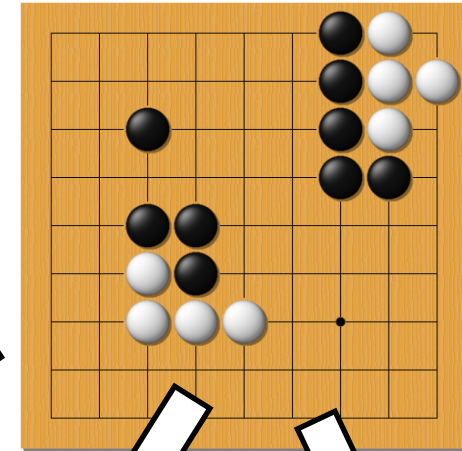
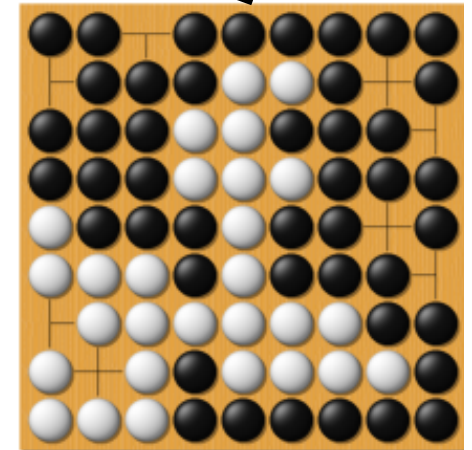
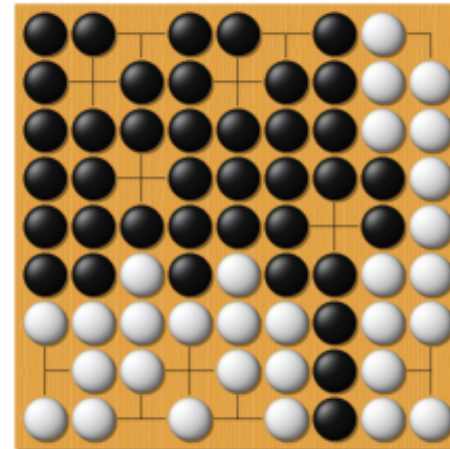
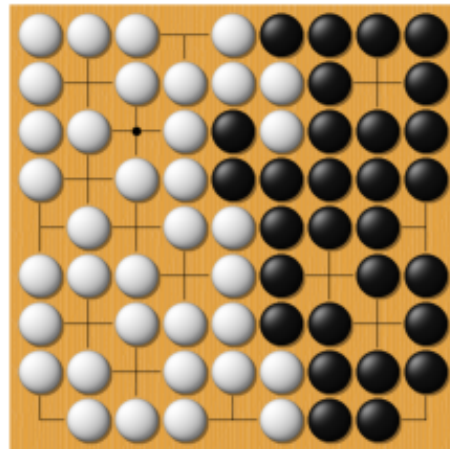
乱数を使って、多数の
終局図を作り、勝敗を記録

これを**プレイアウト**と呼ぶ

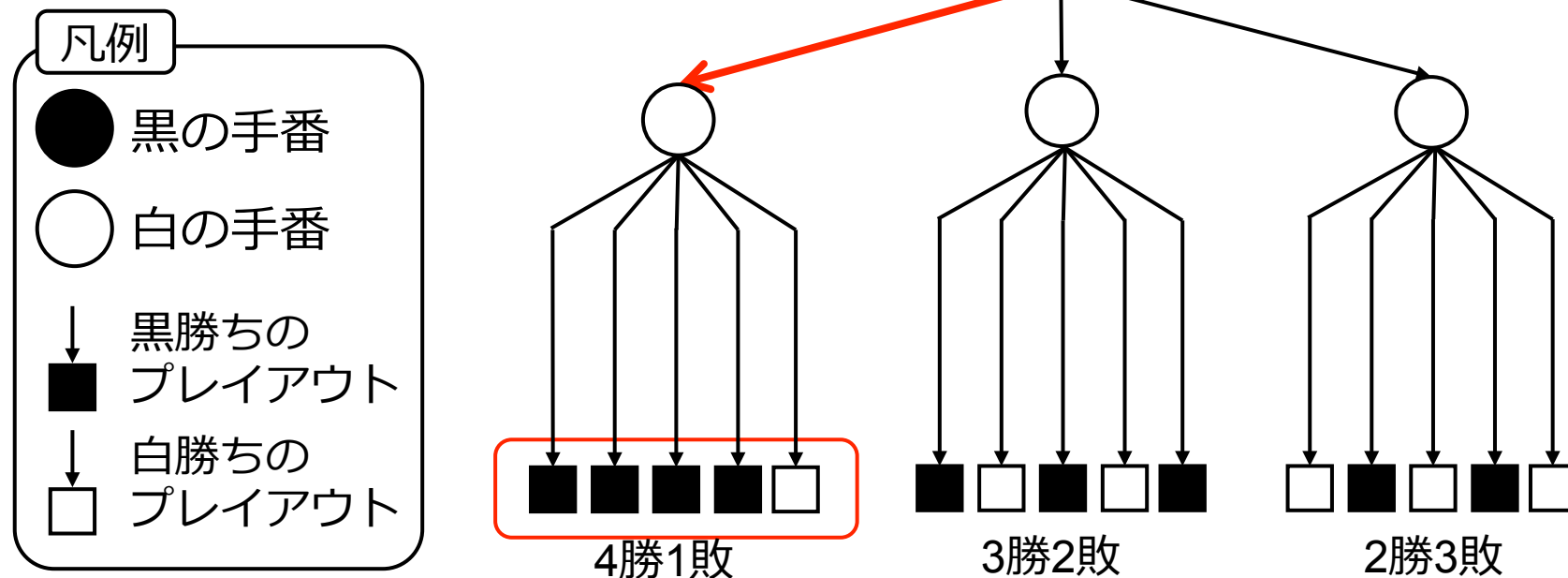
便宜上この方法を
原始モンテカルロ囲碁
と呼ぶことにする

黒
2勝1敗

注：眼を残す



原始モンテカルロ囲碁



一番勝てそうな手を選ぶ

注：この方法は弱い

注：いくら時間を掛けても最善手は分からない

原始モンテカルロ囲碁を改良

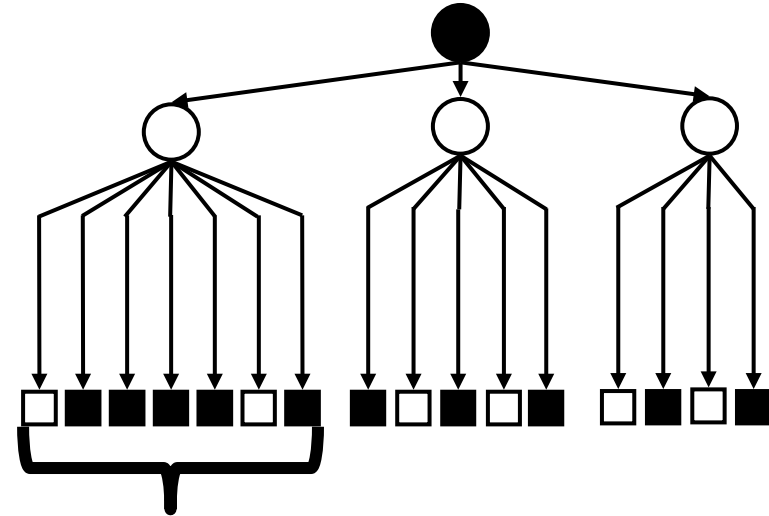
原始モンテカルロ囲碁は
そのままだと**弱い**

相手のミスに期待した手を打つ
飛車の頭に歩を打つような手

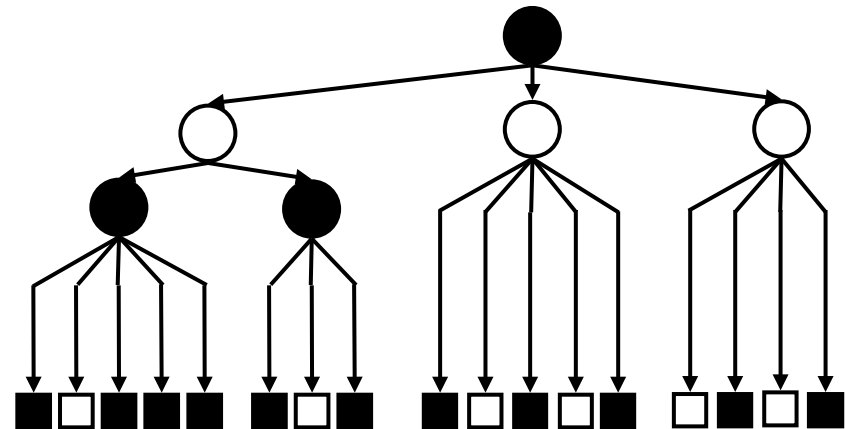
そこで、
良さそうな所を
深く読むように改良

これが
モンテカルロ木探索

プログラムCrazyStone [Coulom 2006]



プレイアウト数が閾値以上なら展開



MCTS の応用

囲碁の研究が
様々な応用に貢献

single player game
(real time game)

multi player game

two player
game

[H. Nakhost and M. Müller 2009]
“Monte-Carlo Exploration for
Deterministic Planning”, IJCAI’09

Planning

biometric security

[Tanabe, Yoshizoe and Imai 2009]
“A study on security evaluation methodology for
image-based biometrics authentication systems”

[Cazenave, Balbo, Pinson 2009]
“Monte-Carlo bus regulation”

自然言語処理

scheduling

[Chevelu, Putois, Lepage 2010]
“The true score of statistical paraphrase generation”

あらためて発表の概要

モンテカルロ木探索 Monte-Carlo Tree Search (MCTS)

囲碁のために開発

評価関数不要

汎用性が高く
様々な問題に応用

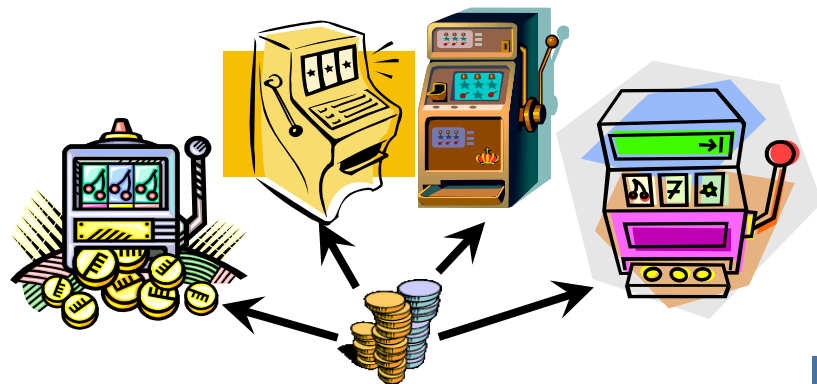
MCTSの理論: Multi-Armed Bandit と木探索

MCTSの実践: 機械学習や職人芸で強化

MCTSの得意、不得意

MCTS の並列化

Multi-Armed Bandit と囲碁



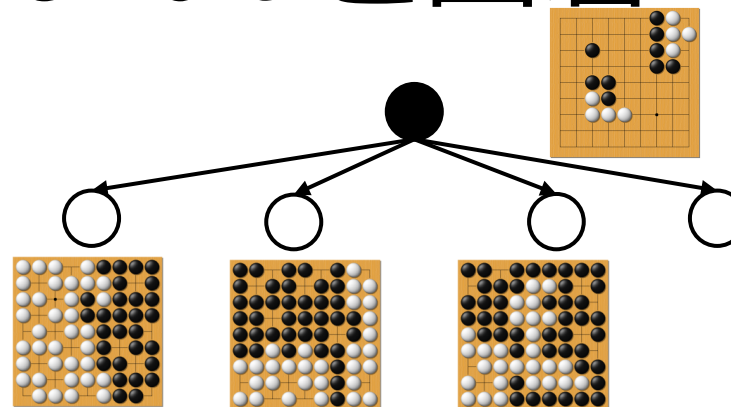
[Auer, Cesa-Bianchi, Fischer 2002]

UCB1 アルゴリズム

cumulative regret を最小化する $\frac{w}{s} + \sqrt{\frac{2 \ln t}{s}}$

mean項 + bias項

w : その腕の報酬
 s : 投入したコイン
 t : s の合計



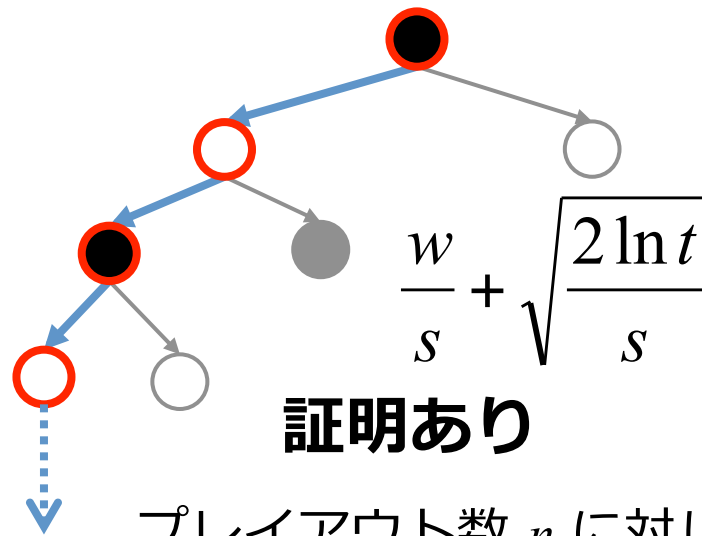
MAB と 原始モンテカルロ囲碁

腕	↔	合法手
コイン	↔	プレイアウト
コインの枚数	↔	考慮時間
regret 最小化 が目的	?	最善手が目的

UCT algorithm

Upper Confidence bound applied to Trees

節点の比較に**UCB1**を用いる
木探索アルゴリズム



証明あり

プレイアウト数 n に対し、

$$\frac{w}{s} + O\left(\frac{\log n}{n}\right)$$

また、**1手目**が間違える確率は 0 に収束.

つまり(いずれは)最善手が分かる

値が**最大の枝**をたどる

(末端の節点を展開)

末端で**プレイアウト**

経路上の節点を更新

繰り返し

UCB1, UCT: 証明関連の補足

報酬が $[0, 1]$ なら $c=1$ で証明成立

[Auer, Cesa-Bianchi and Fischer 2002]

UCB1の証明

Chernoff bound を用いる

[Kocsis, Szepesvári 2006]

UCTの証明

各部分木を確率過程とみなす
Hoeffding-Azuma 不等式を用いる

値域が $[0, 1]$ より広がると
($c=1$ のままでは) 証明は不成立

1回プレイアウトするごとに
更新しないと証明は不成立

exploration constant

$$\frac{w}{s} + c \sqrt{\frac{2 \ln t}{s}}$$


実践的には問題に合わせて
 c を調整する

困碁なら、強くすると
大抵 c の値は小さくなる

実際には、収束するかどうか
気にする開発者は少ない
(というかいらない?)

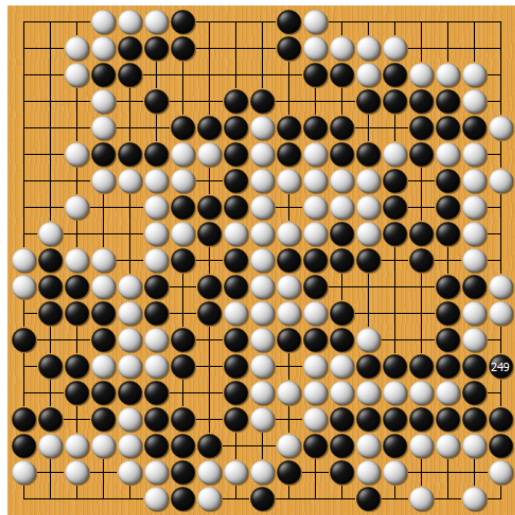
MCTS の実践 (囲碁)

9路盤ならそこまで弱くないが
19路盤だと壊滅的に弱い

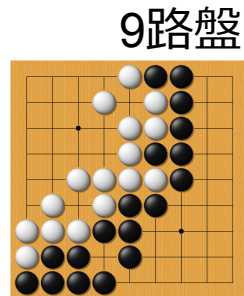
プレイアウトが
pure random では弱い



機械学習や職人芸で強化



19路盤
(正式)



9路盤

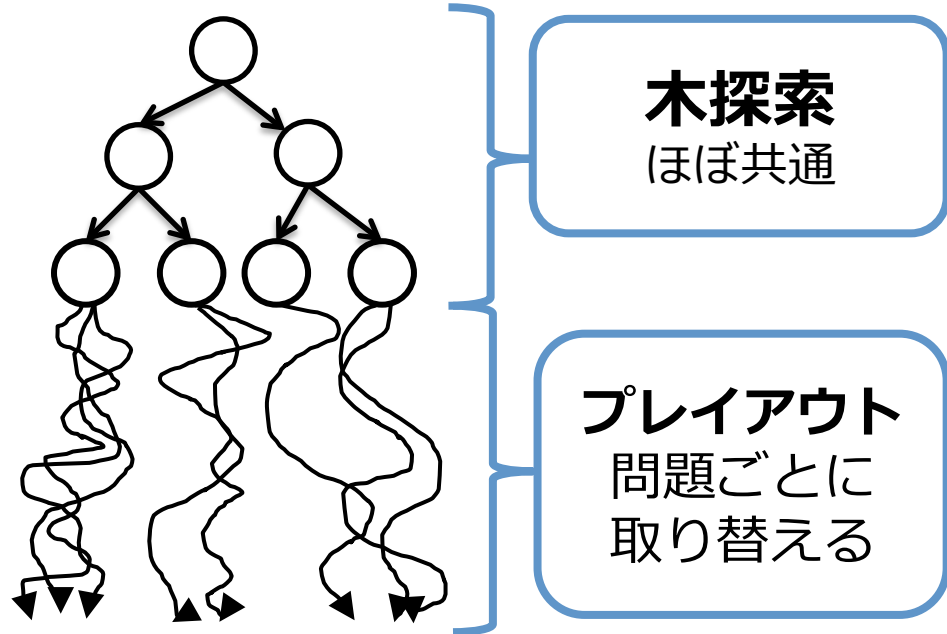
$$\frac{w}{s} + c \sqrt{\frac{2 \ln t}{s}} \quad \text{信頼がおけない腕を} \\ \text{bias項が優遇}$$

この効果があればある程度うまく行く

UCB1 は別に最善じゃない
強いプログラムは使わない

囲碁の知識や謎のヒューリスティック
でbias項をおきかえる。実は c は 0

証明？何それおいしいの？



MCTS の汎用性

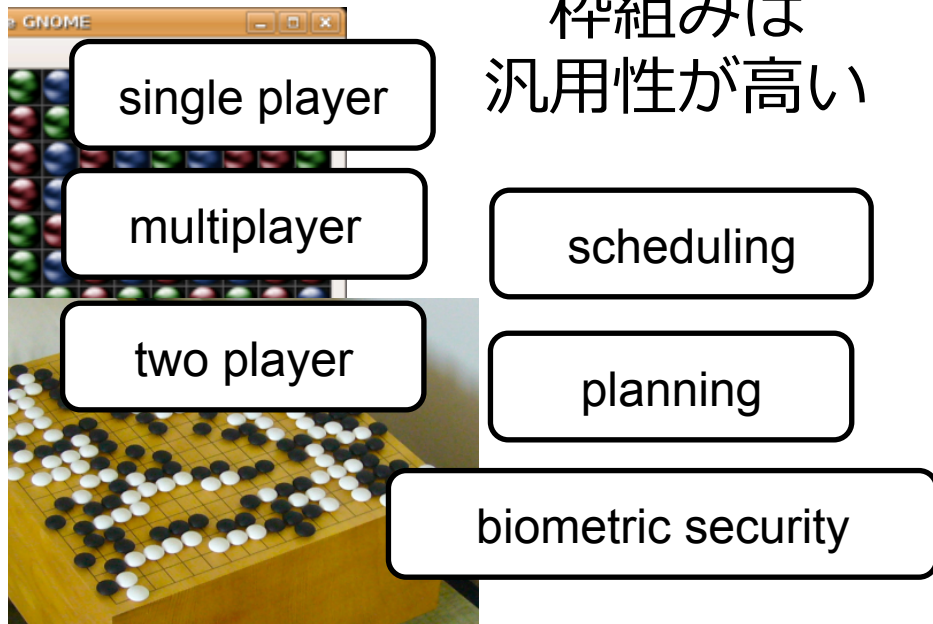
評価関数を

- ・作れない
 - ・**作るのがめんどくさい**
- 等の時にも効率よく探索

もちろん、何でもできる
万能のツールではない

MCTS の 得意、不得意は？

例えば、
囲碁は得意
将棋は苦手



プレイアウトの作りやすさ

囲碁

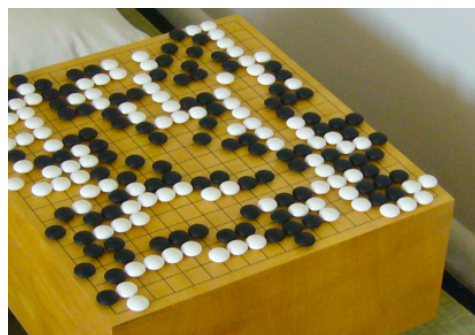
候補手が減っていくので
自然と終局する
(簡単な制約は必要)

将棋

ランダムなプレイで
自然な終局図を
迎えることは困難

プレイアウトで「**自然な**」報酬が
得られるかどうかの問題

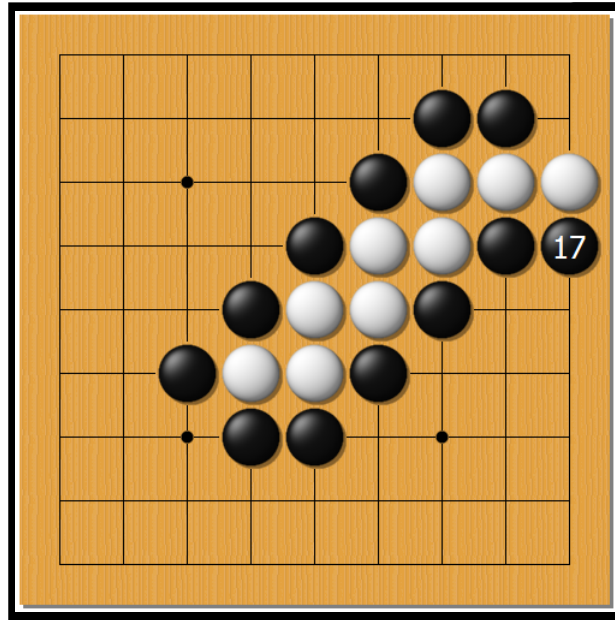
何が「自然」か考えると難しいが...



注意：MCTS 将棋は
初段よりは強い

初段あるのに苦手とは生意気な...

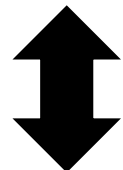




シチョウ

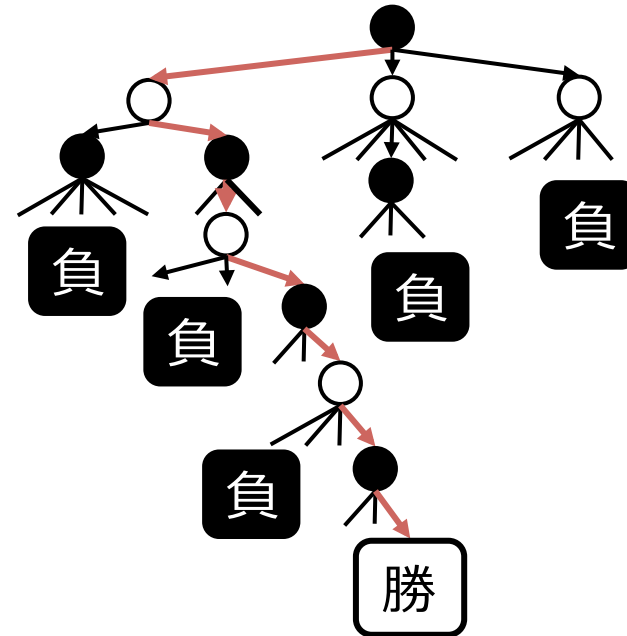
分かりやすい一本道

木探索は一般には**深い手順が苦手**
(選択枝の多い局面は得意)



プレイアウトは**深い手順もOK**
(選択枝の多い局面は苦手)

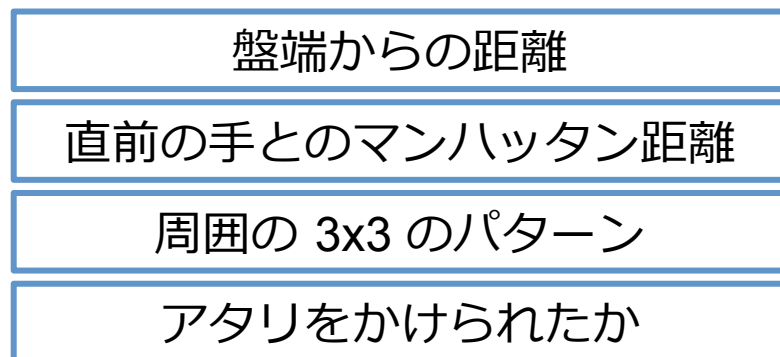
「シチョウ」



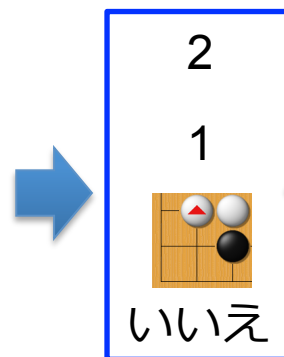
プレイアウトが木探索の
弱点を補うのが理想

人間の棋譜から学習

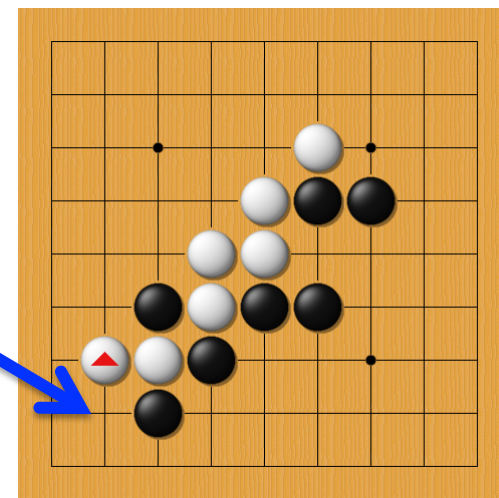
数千～数十万の feature の重みを調整



この特徴がシチョウ対策



候補手の重みは
特徴の重みの積

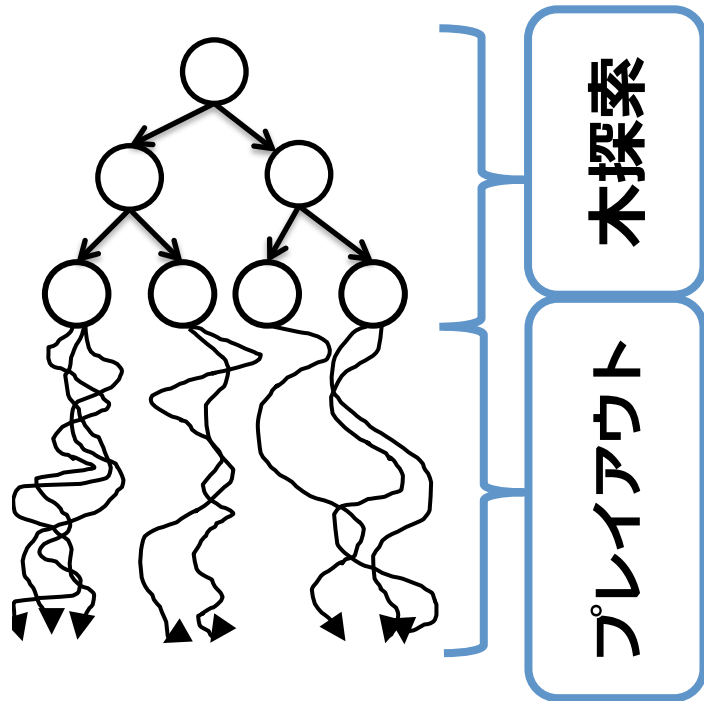


一番重みの高い手がプロの手と
一致する回数を最大化

プレイアウト強化
探索の枝刈り

参考: 将棋の bonanza メソッドも
人間の棋譜との一致率を上げるように学習

プレイアウトの難しさ



木探索と組み合わせるため
挙動の予測が困難
(強さ以外の) 評価も難しい

プレイアウト単体では「弱い」が
木探索と組み合わせると強いという例あり

cf. Simulation Balancing [Silver, Tesauro 2009]
[Huang, Coulom, Lin 2011]

プレイアウトの
質が強さに直結
一番重要な部分

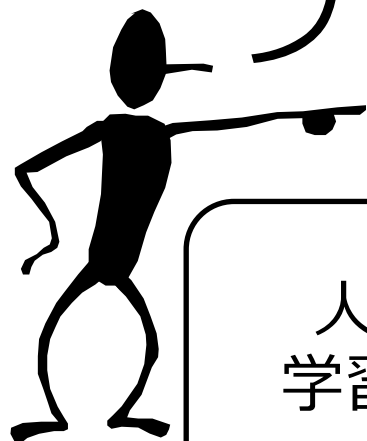
完全なランダムだと弱い

しかし一番良く
分からない部分

しかしランダム性は必要

プレイアウトの作り方？

囲碁でも二つの派閥がある



人間の棋譜に
学習で似せる派

CrazyStone 等

職人芸で
手作りする派

Zen 等



今のところ... { それなりのランダム性
それなりの速さ (1ミリ秒なら十分)
プラス職人芸

要するに、それなりにうまくいけば良いのでは
(投げやりですみませんが本当に分かりません)

UCT 以外のMAB → 木探索

[Li, Chu, Langford, Schapire 2010]

LinUCB → LinUCT

[万代, 金子 2014]

[Garivier, Cappé 2010] [??? 2012ごろ]

KL-UCB → KL-UCT

[Karnin, Koren, Somekh 2013] [Cazenave 2014]

Seq. Halving → SHOT
simple regret 最小化

[Kocsis, Szepesvári 2007]

Discounted UCB

[Hashimoto, Kishimoto, Yoshizoe, Ikeda 2012]

Accelerated-UCT

二人ゲームに特化

木探索アルゴリズムになって
かつ収束の証明がある物はUCTのみ

いろいろあるが、
**強いプログラム
は強くない**

弱いプログラムなら
UCT より強くなる例あり

モンテカルロ木探索の性質 最近の囲碁プログラム

形勢判断が正確

絶対に数え間違えない

「地」ではなく勝率を
最大化するように打つ。よって

**リードしていると安全に
不利だと攻撃的に打つ**

負けると非常につまらない
(勝つと面白い)

局所的な読みが不正確
詰碁は解けない

重要

プレイアウトの報酬を
スコアでは無く
勝ち=1, 負け=0 とする

他のアルゴリズム ($\alpha\beta$ 探索)
だとこれが非常に難しい

考慮時間が短い

探索には厳しい

モンテカルロ木探索の性質

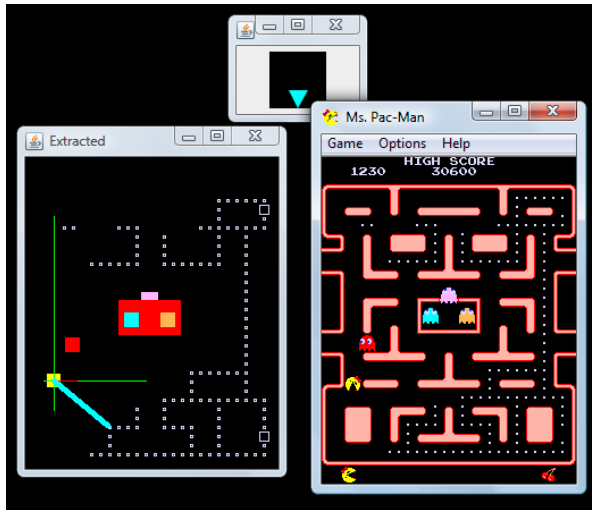
リアルタイムゲームの例

Ms. Pac-Man AI Competition

MCTS を用いたプログラムが
世界記録を更新 [Ikehata, Itoh 2011]

挟み撃ちをさけるように作られた
プレイアウトを使って MCTS

Ms. Pac-Man 1/15秒ごとに行動



<http://cswww.essex.ac.uk/staff/sml/pacman/PacManContest.html>

○○○AI Competition

某有名アクションゲーム
のクローンを使ったAI大会

「人間のプレイに似せる部門」
(Turing Track)

A*探索と MCTS を比較 (動画)

[Baumgarten 2009?] [中野, 美添, 脇田 2013]



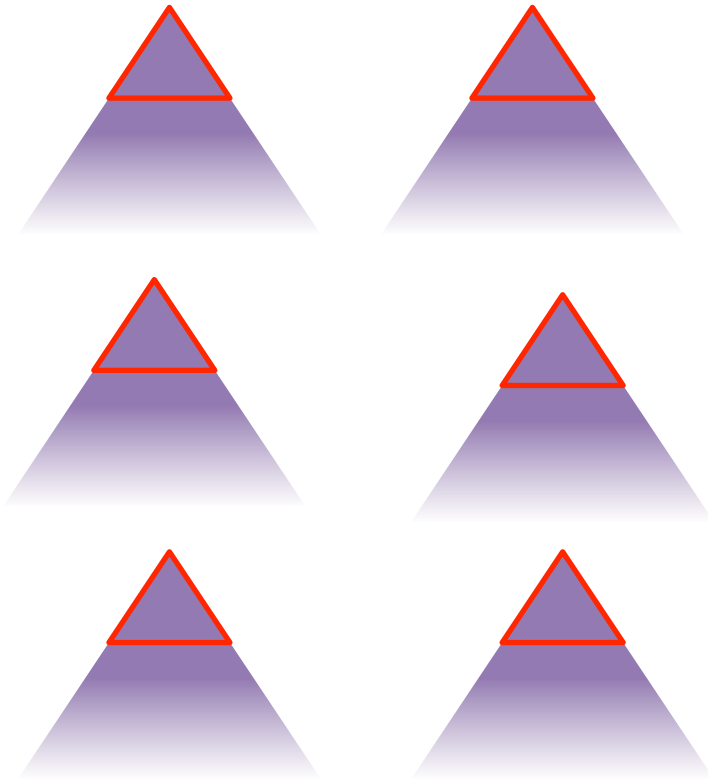
プライバシーに
配慮しています

1/24秒ごとに行動



<http://www.platformersai.com/>

MCTS の並列化: Root 並列化



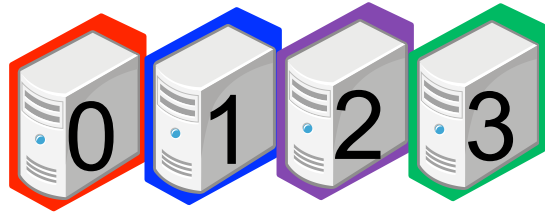
異なる乱数シードで
複数の探索を行う

深さ 1~3 程度の
部分を共有する

少し工夫すると16ノード
くらいまでは有効

探索の並列化は難しいので
これでもそれなりに良い

参考: SAT solver の portfolio 並列化

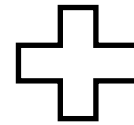


均等な負荷分散が可能

[Romein et al. 1999]

TDS

分散ハッシュテーブル
による並列探索手法

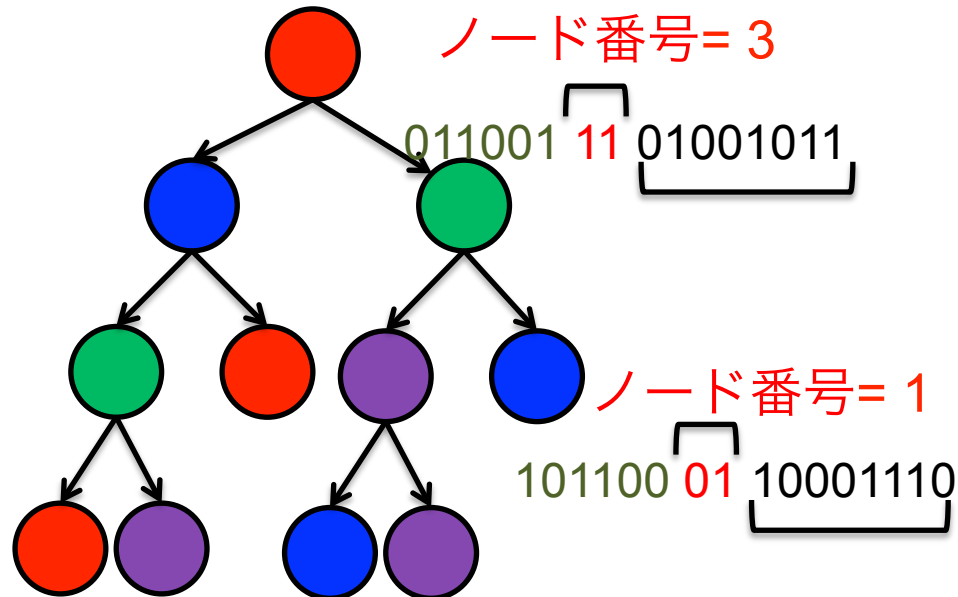


[吉本, 岸本, 金子, 美添, 田浦 2007]

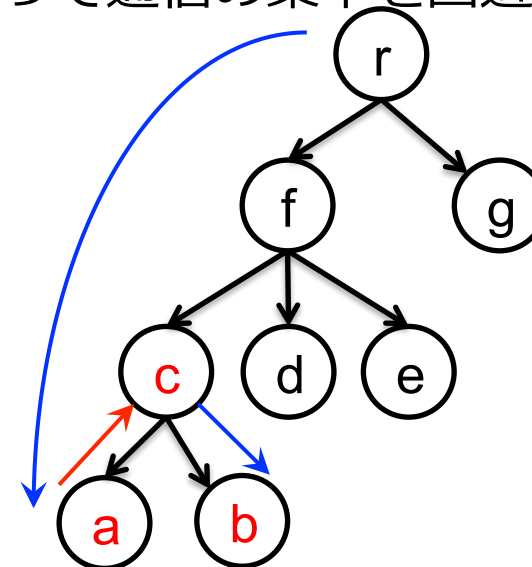
df-UCT

depth-first reformulation
(深さ優先変形)

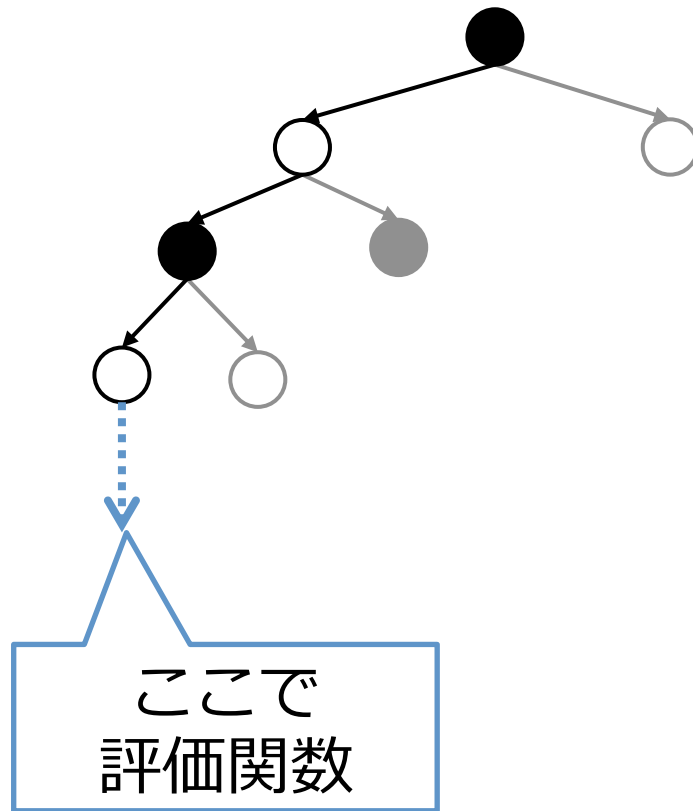
グラフの各節点を、異なる
計算ノードに割り当てる



アルゴリズムの深さ優先変形に
よって通信の集中を回避する



補足1: 評価関数 + プレイアウト

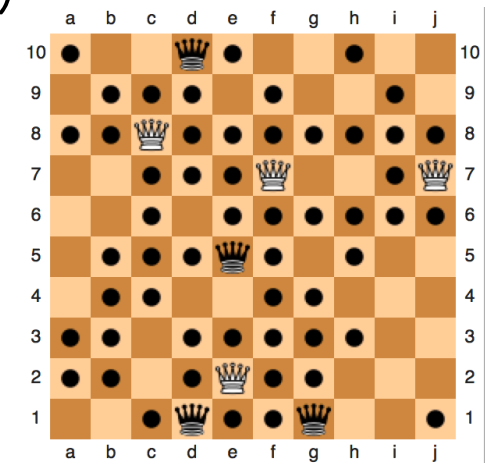


プレイアウトは
終局する必要はない

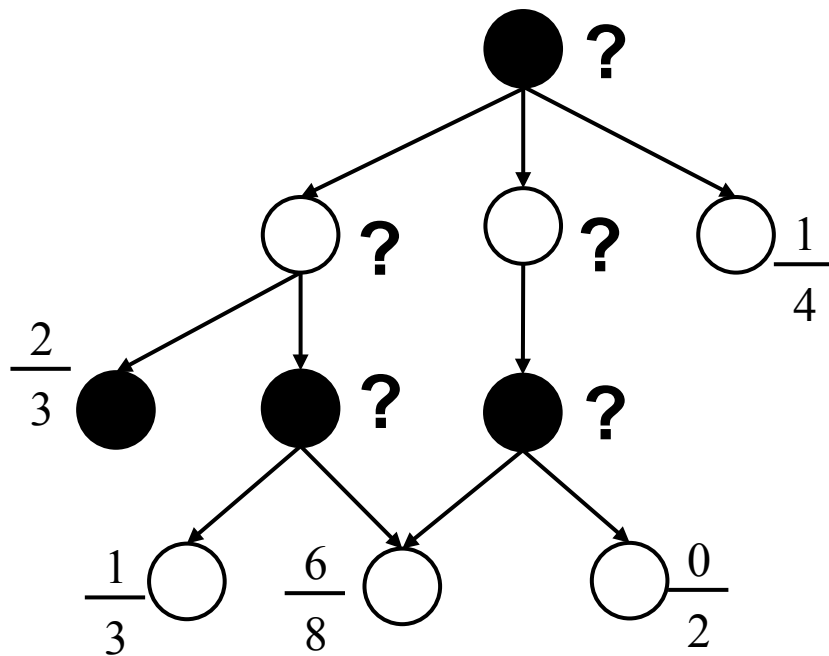
数手進めて評価関数を
呼ぶ方法もありうる

評価関数が不正確な場合に
強化することが可能
(かもしれない)

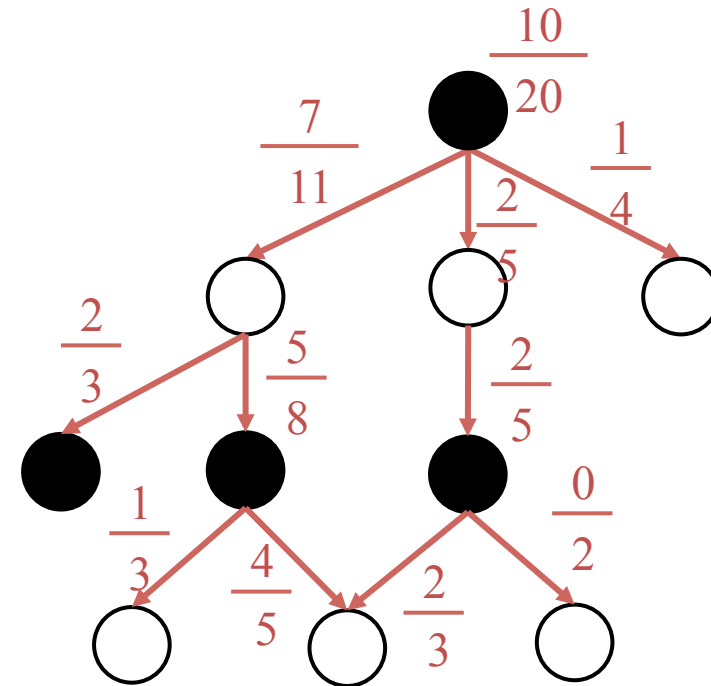
Amazons
(という検索しにくいゲーム)
などで成功例あり



補足2：合流への対処



合流があるとmean項
の定義が難しい



節点ではなく**枝が報酬を持つ**

簡単だし、収束もする

[Childs, Brodeur, Kocsis 2008] ただし無駄はある

もっと良い方法、求む

モンテカルロ木探索 Monte-Carlo Tree Search (MCTS)

囲碁のために開発

評価関数不要

理論: MAB と木探索

得意、不得意がある

囲碁を強くしたい

汎用性が高く
様々な問題に応用

実践: 機械学習と職人芸

並列化は有望

さらに多くの研究者の
参入求む