コンピュータ囲碁における モンテカルロ法 ~理論編~

美添 一樹

コンピュータ囲碁に起きた革命

- 2008年3月末、パリ囲碁トーナメントのエキシビションでプロ対コンピュータの対戦が実現 (http://paris2008.jeudego.org/)
  - プロ:タラヌ・カタリン五段(日本棋院中部総本部所属)
  - コンピュータ:MoGo
- 9路盤はハンデなしで3局対戦
  - MoGoの1勝2敗
- 19路盤はMoGoが9子のハンデをもらい、1局対戦
  - カタリン五段の勝利

ここが革命です、念のため

コンピュータの強さ

囲碁だけが弱かった

- 主な二人零和完全情報ゲームの中で囲碁だけが他と比較して際立って人間優勢だった
  - コンピュータが勝利したのは、正式に用いられる19路盤ではなく、コンピュータ有利と思われる9路盤だが、公の場でプロにコンピュータが勝利するというのは3年前の状況からは想像できない快挙である

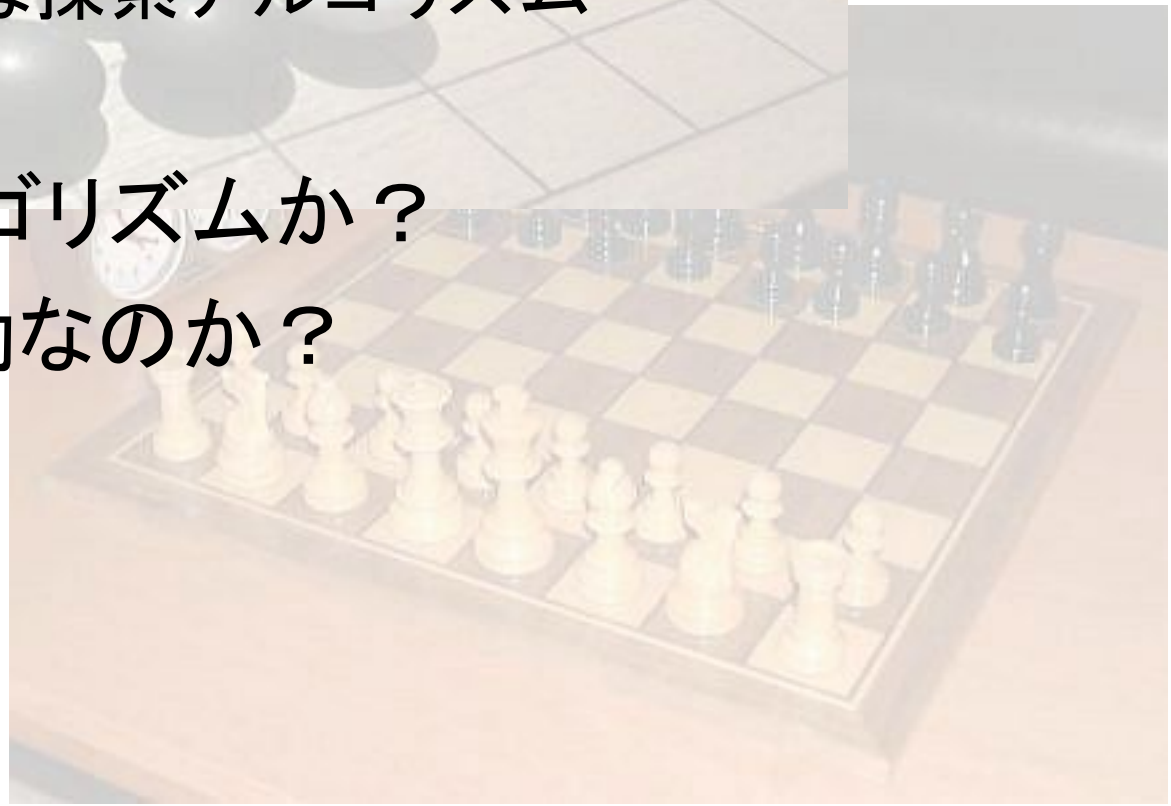
チェッカー	1994年に世界チャンピオンに勝利 (2007年に初期配置の引き分け証明)
オセロ	1996年に世界チャンピオンに完勝
チェス	1997年にIBMのDeepBlueが当時世界チャンピオンのKasparovを破る
将棋	アマトップレベルの強さと言われている
囲碁	アマ初段をようやく超えた程度

主なゲームにおけるコンピュータの強さ

快挙の原動力は？

- 2006年に登場した画期的なアルゴリズム
  - 通称「モンテカルロ木探索」
  - 評価関数不要な探索アルゴリズム

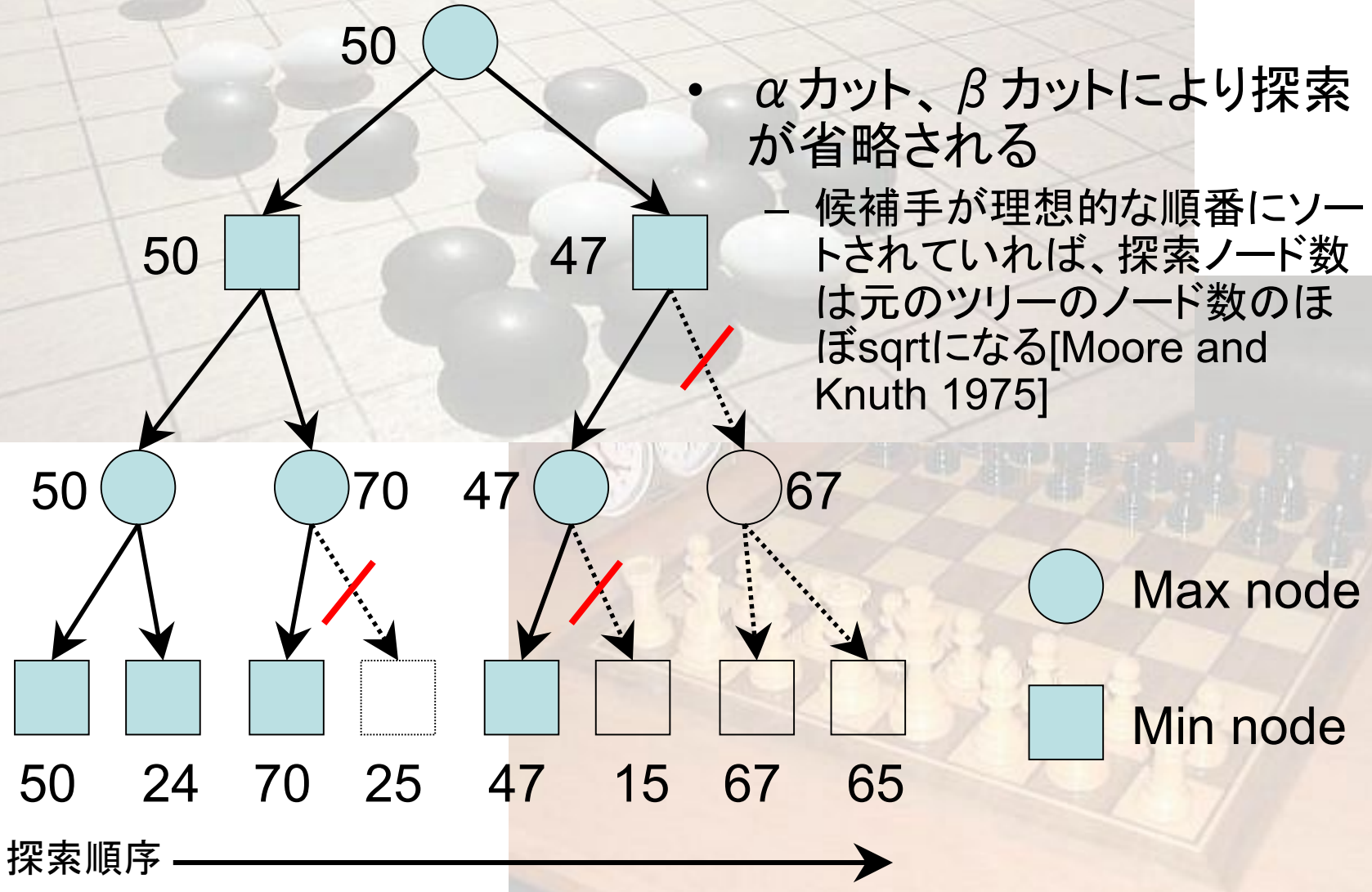
- どのようなアルゴリズムか？
- なぜ囲碁に有効なのか？



コンピュータプレイヤーの進歩

- 囲碁だけが難しかった
  - つまり、他のゲームで有効だった手法が囲碁には通用しなかった
  - なぜか？
- まず、他のゲームのコンピュータプレイヤーのアルゴリズムについて説明する

mini-max探索＋α β 枝刈り



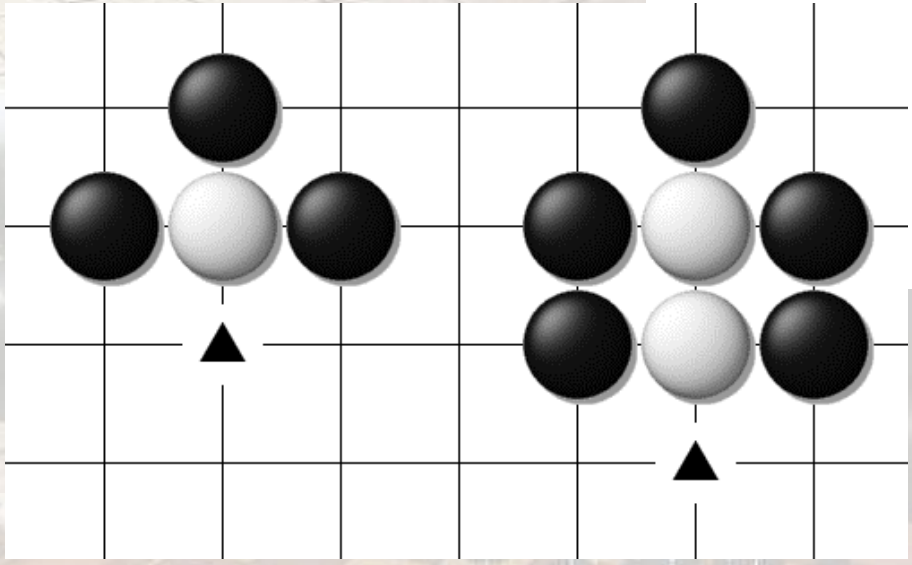
- α カット、β カットにより探索が省略される
  - 候補手が理想的な順番にソートされていれば、探索ノード数は元のツリーのノード数のほぼsqrtになる[Moore and Knuth 1975]

囲碁のルール

- 黒、白交互に交点に石を置いていく
  - 19x19の盤が普通
- 最終的に「地」が大きいほうが勝ち
  - 「地」とは一方の色の石だけで囲われた範囲のこと

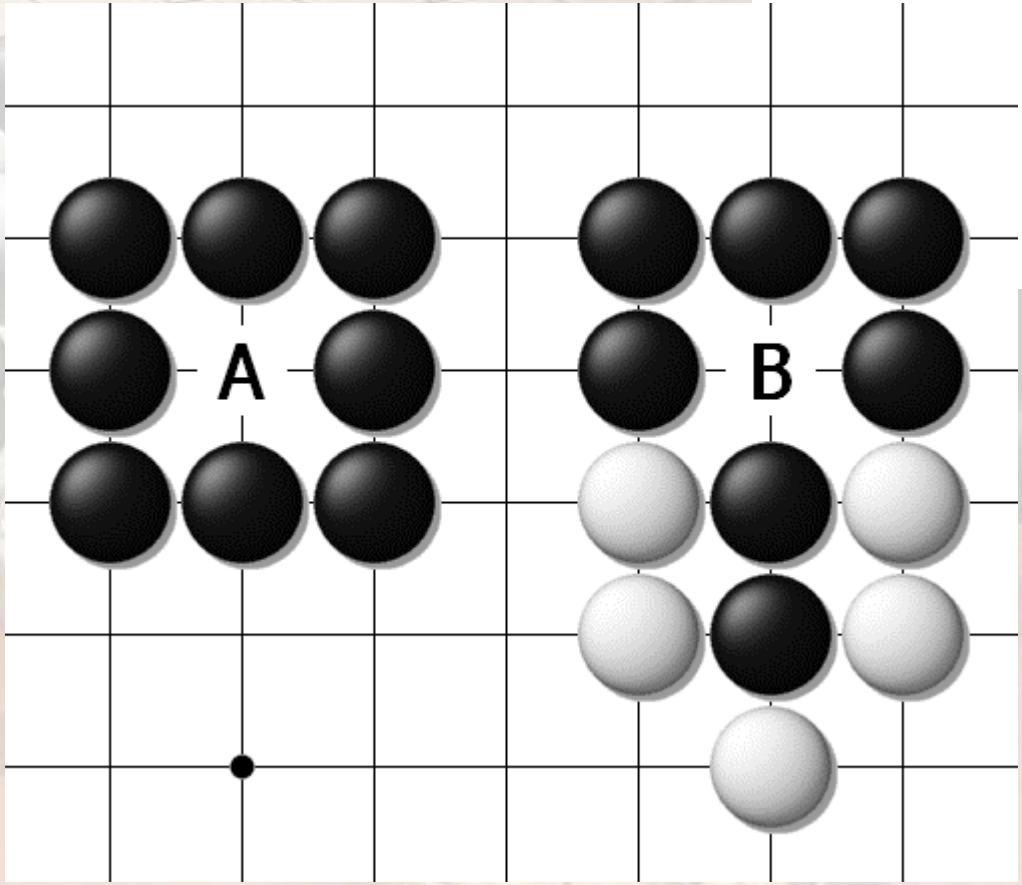
囲碁のルール：囲んだら取れる

- ▲のところに黒が打つと、白石を取れる
  - 空点が無くなると取られる
    - 空点のことを、「呼吸点」「ダメ」などと言う
  - つながっている石は一蓮托生になっている
    - 取られるときはまとめて取られる
    - つながっている石の集合を「連」という



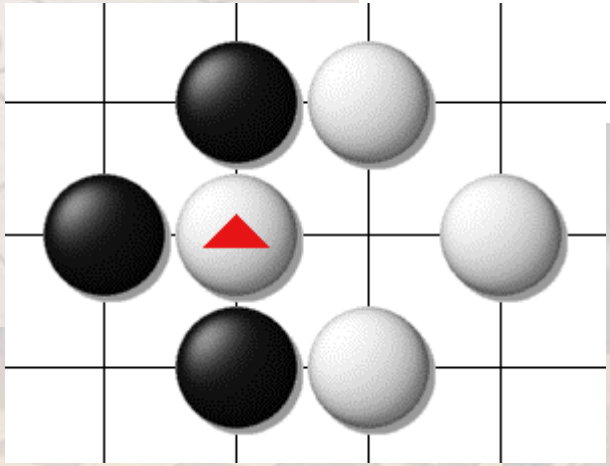
囲碁のルール：着手禁止点と例外

- Aに打つと反則
  - そのまま取られる場所には打てない
- Bには打って良い
  - 打った瞬間に黒石を取れるから



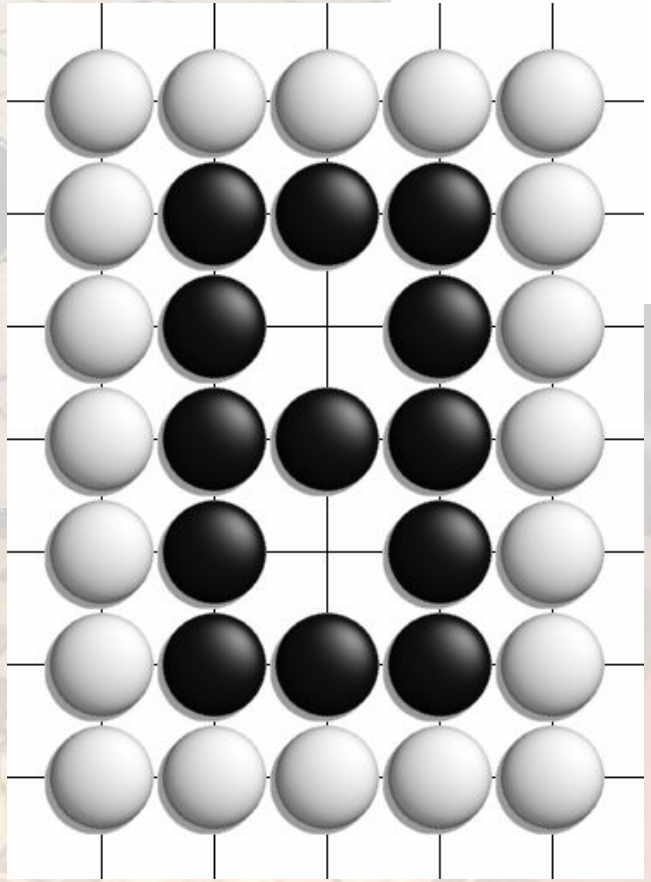
囲碁のルール：同型反復禁止

- 右図の形になったら、簡単に無限反復が生じる
  - 取られてもすぐに取り返してはいけない
  - 取り返すと反則



生き、死に、という概念

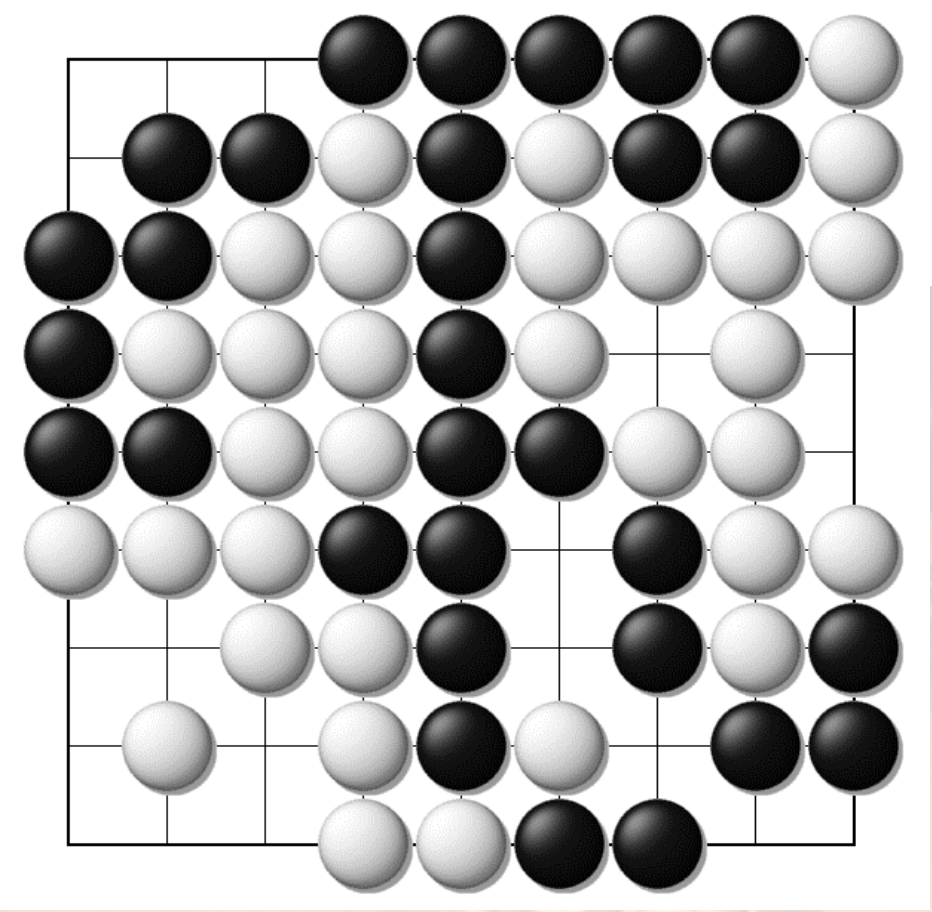
- 着手禁止点が二つある石は、絶対に取られる事はない
  - 絶対に取られない石を「生きている」と言う
  - 着手禁止点のことを「眼」と言う



- 二眼あると「生き」

実戦例

- とある商用ソフトと私が打った例
  - これは終局図
  - 先手の黒が有利なため、それを是正するために黒にハンデを負わせるのが普通
  - それをコミという
    - 19路盤でも9路盤でも6.5目か7.5目が普通



囲碁の難しさ その1
探索空間が大きい

- 19路盤囲碁は探索空間が巨大
  - チェッカーは初期局面が引き分けになることが解明された(2007年)
  - 同様に、5路盤の囲碁は最善手順が完全解明されている
- ところで、9路盤の探索空間はチェス以下
  - それでも2005年までは弱かった
  - どっちもアマ初段くらい
- おかしい、だって他のゲームだと・・・
  - 性質の似たゲームなら探索空間が小さい方がコンピュータ有利
    - 将棋、チェス、中国将棋などの比較
    - チェッカー(8路)とドラフト(10路)の比較
  - なぜ、19路盤と9路盤の強さに差が無いの？

チェッカー	10^{20}
オセロ	10^{28}
チェス	10^{50}
将棋	10^{71}
囲碁(9路盤)	10^{38}
囲碁(19路盤)	10^{171}

探索空間(可能な局面数)

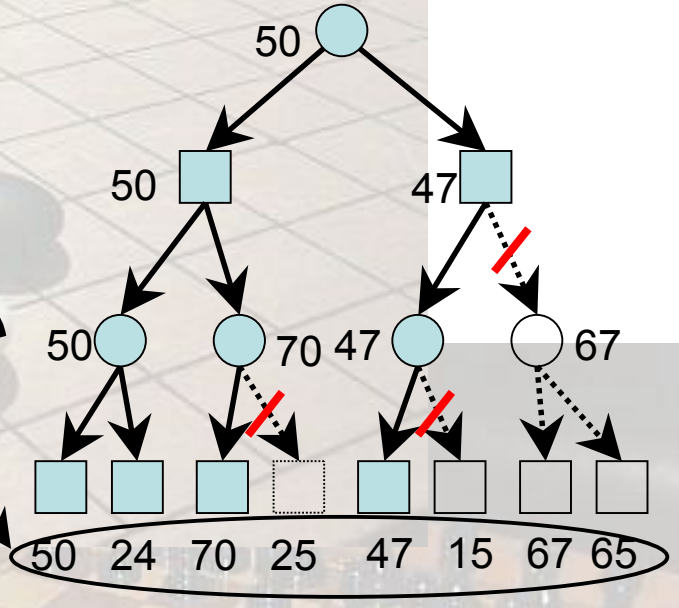
囲碁の難しさ その2

評価関数が作れない

この数値はゲームのスコア

しかし、実際のスコアは勝敗がつくまで深く探索しなければ分からない

よって、探索を途中で打ち切り、その時点でのスコアを近似する評価関数を用意する



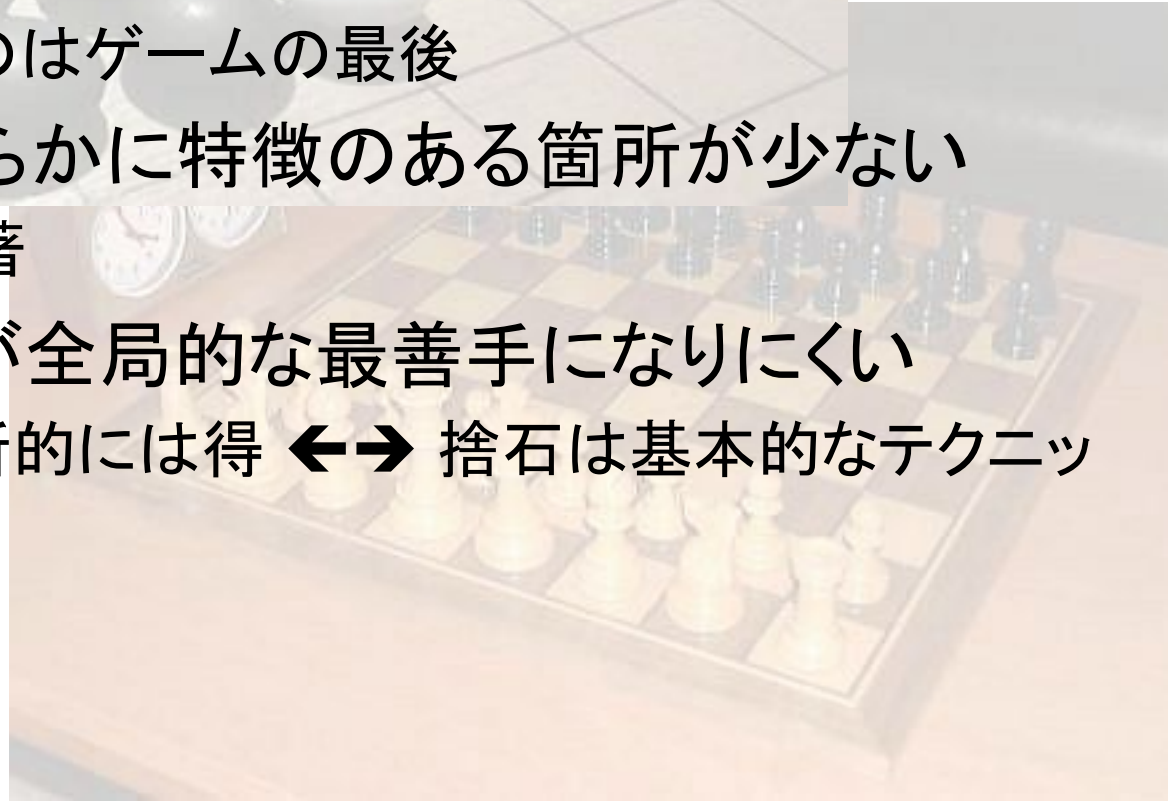
評価関数はどうやって作るもの？

評価関数の例
囲碁以外のゲーム

- オセロ
  - 隅や辺の重要な箇所のパターンを学習して評価関数を作成
- チェスや将棋
  - 駒の価値、玉の安全度、駒が自由に動けるか等
    - チェスの例：ポーン1点、ビショップとナイト3点、ルーク5点、クイーン9点、キング∞点
  - ボナンザメソッドなどもあり

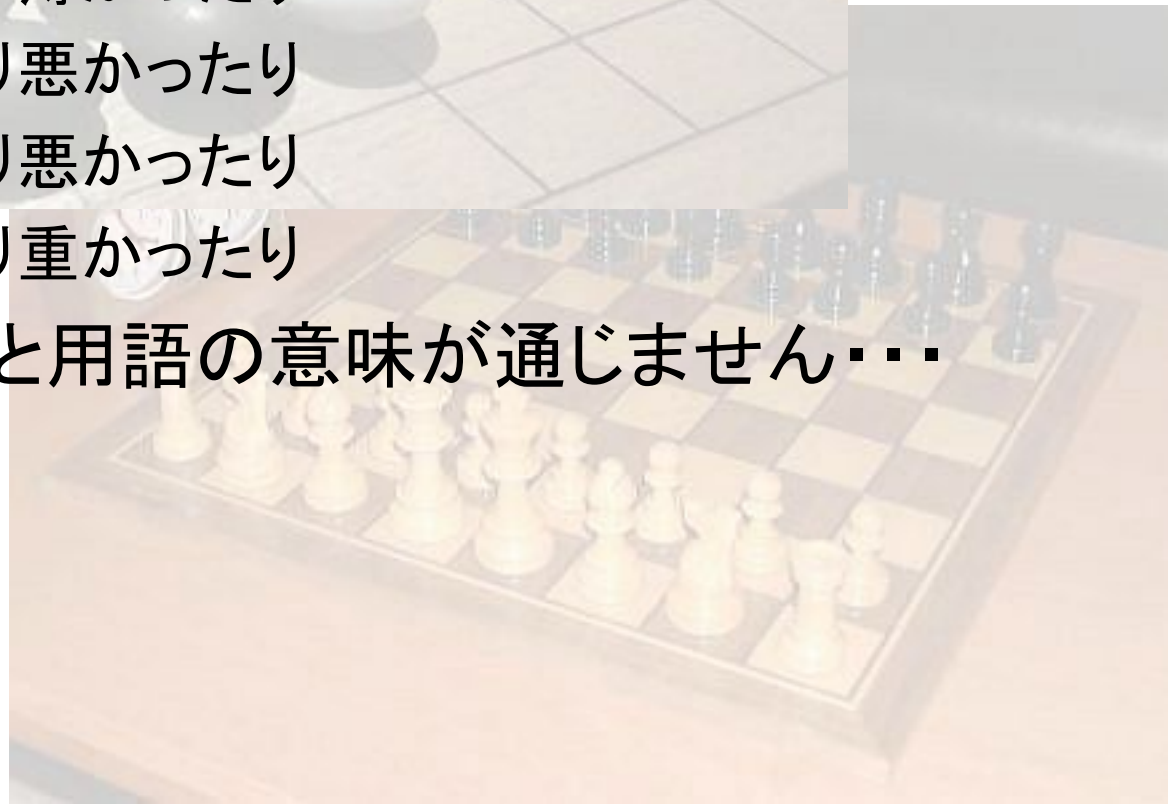
囲碁の評価関数の難しさ

- 石の価値は平等
  - 駒の価値などは用いることができない
- 領域の広さを競うなら、広さを基準にする？
  - 領域が確定するのはゲームの最後
- オセロのような明らかに特徴のある箇所が少ない
  - 特に19路盤で顕著
- 局所的な最善手が全局的な最善手になりにくい
  - 石を取るのは局所的には得 ←→ 捨石は基本的なテクニック



人間はどうやってプレイしてるの？

- ・・・説明不能です。
  - 特に中盤は難しいです
    - 石が厚かったり薄かったり
    - 形が良かったり悪かったり
    - 味が良かったり悪かったり
    - 石が軽かったり重かったり
  - 初段くらい無いと用語の意味が通じません・・・



つまり囲碁は難しい

- チェスや将棋の駒得のような明らかな評価基準がない
- 何かの要素の足し算で局面の優劣を評価するのは難しい

- 評価関数は速く、正確である必要がある
  - 囲碁の評価関数は、遅いか、不正確である
    - 両方という説も・・・

従来の囲碁プログラムの例 GNU Go

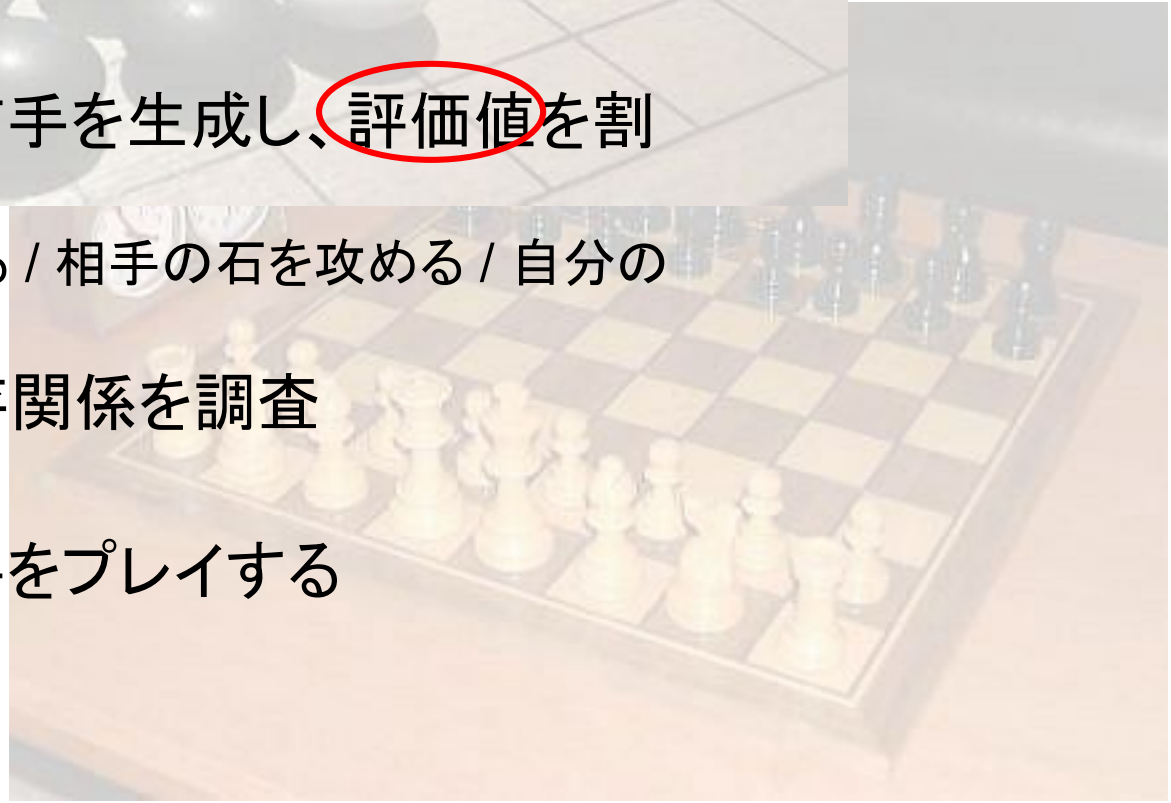
- 商用ソフトの中身は分からないので、オープンソースの囲碁プログラム GNU Goについて説明
  - GNU Goは最強の商用プログラムよりも少し弱い
  - 多数の複雑な評価関数を用いている
  - コードはCで約80,000行
  - パターンデータベースがテキストで約52,000行
- 棋力はアマ初段より少し弱い
  - 19路でも9路でも

GNU Goの着手選択
職人芸の結晶(?)

- 盤面の状況を分析する
  - 連絡・切断をある程度調査
  - それから石の安全度を調査
- パターンデータベースにマッチする手を発見し、評価値を割当てる
- 着手の目的別に候補手を生成し、評価値を割当てる
  - 目的: 自分の石を守る / 相手の石を攻める / 自分の領域を広げる など
- 複数の評価値の依存関係を調査

⇩

- 一番評価値の高い手をプレイする



モンテカルロ木探索によるプログラム

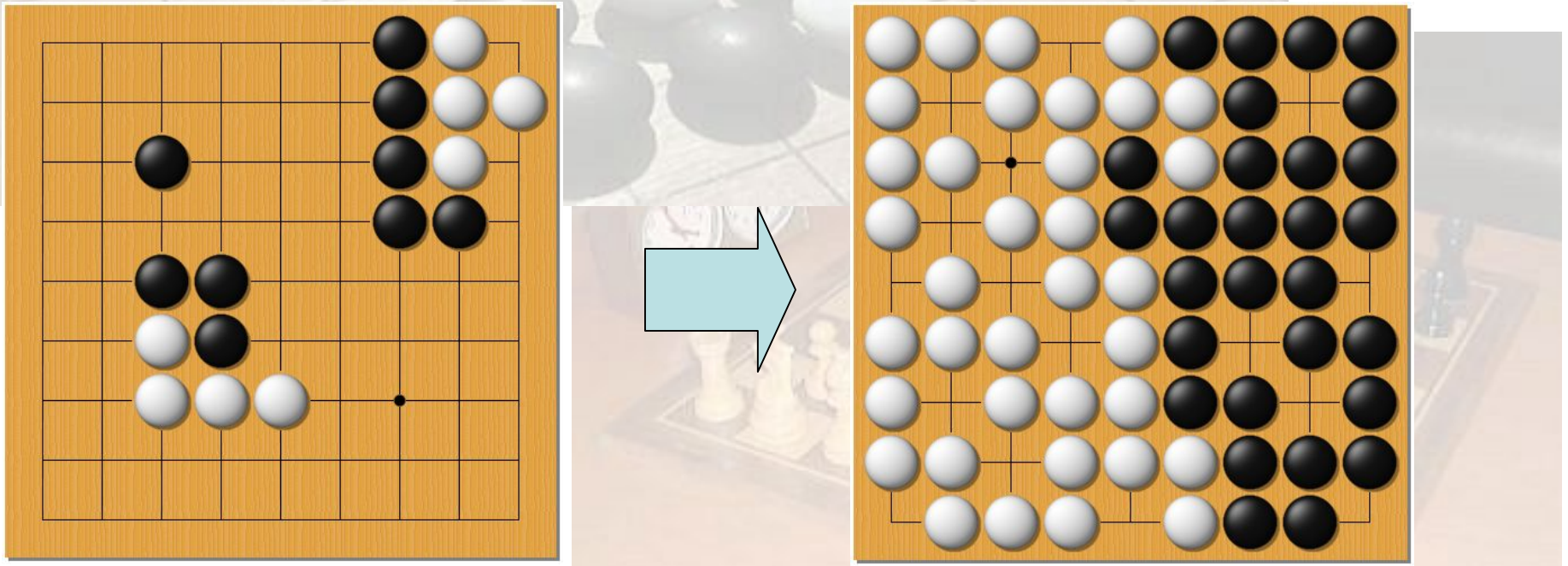
- 囲碁の評価関数は難しい←これは本当である
- しかし、囲碁でも終局した状態なら簡単に勝敗の判定が可能
- この性質をうまく利用したプログラムが2006年に登場した CrazyStone

原始モンテカルロ囲碁

- 乱数を用いて囲碁をプレイする
  [Brügmann][Bouzy][Cazenave]
  - 囲碁は終盤に近づくに連れて合法手が減少する
  - 合法手の中からランダムに選んで打つだけのプレイヤーでも終局可能
  - ただし、少し制約が必要
    - 自分の「眼」には打たないようにする
    - 二つ「眼」を持つ石は取られない

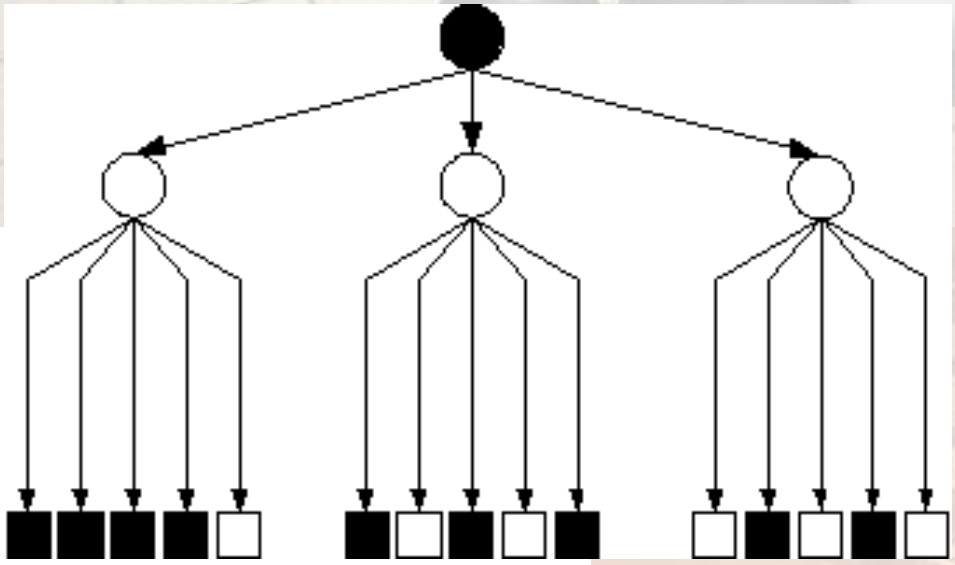
プレイアウトとは

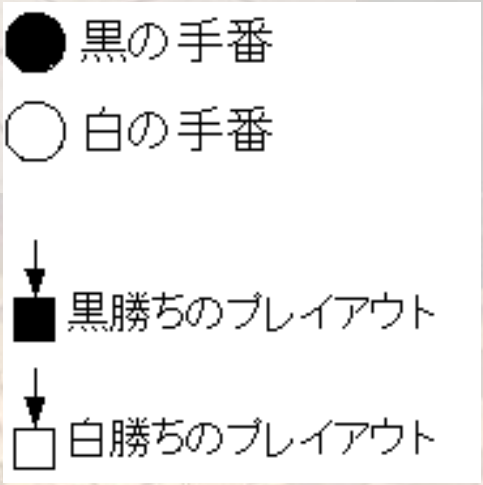
- 乱数を用いて、終局までプレイすることをプレイアウトと呼ぶ



プレイアウトによる局面評価

- 要するに、たくさんプレイアウトを行って、勝てそうな手を選ぶ





もちろん原始モンテカルロは弱い

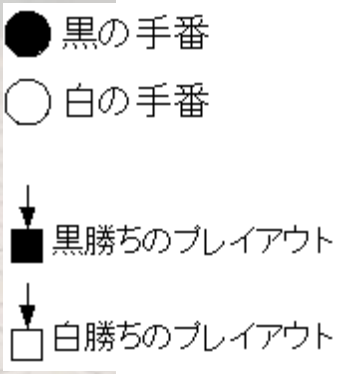
- 深さが2段以上の木に対しては、最善手を返す保証は無い
  - 相手がミスをしたら得だが、正しく応じられると損をする手があるとする
  - 正解の手が少なければプレイアウト中には正解を打つ確率は低い
  - 相手がミスをすることに期待して、その手を打つ
- どれくらい弱いのか調べた論文あり
  - GNU Go相手の勝率は1割くらいでした
  - *H. Yoshimoto, K. Yoshizoe, T. Kaneko, A. Kishimoto and K. Taura*, "Monte Carlo Go Has a Way to Go," AAAI-06, pp 1070-1075

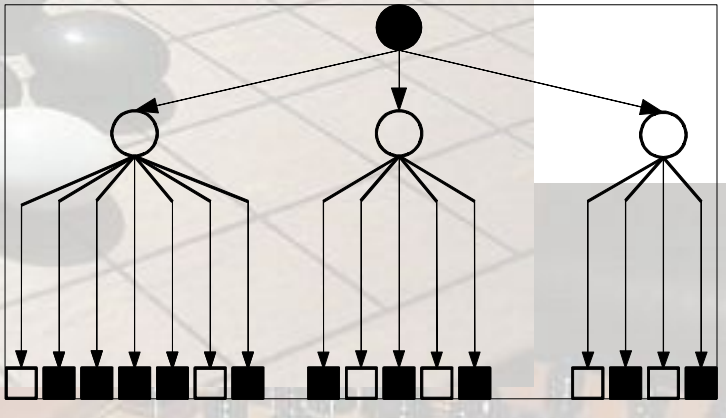
CrazyStone

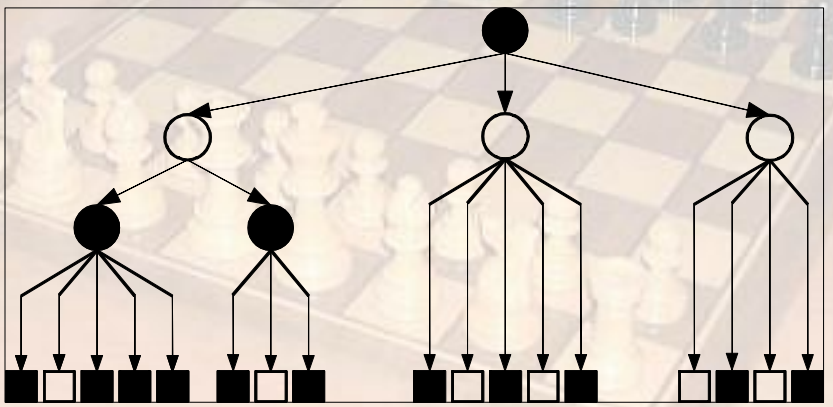
- 2006年のComputer Olympiad 囲碁9路盤部門 優勝プログラム [Rémi Coulom 2006]
  - 原始モンテカルロ囲碁を改良したアルゴリズムを用いた
  - それがモンテカルロ木探索

モンテカルロ木探索
Monte Carlo Tree Search

- 変更点は2つ
  - 有利な手に多くのプレイアウトを割当てる
  - プレイアウトの回数が閾値を超えたら木が生長する
- さらに以下の工夫が重要
  - プレイアウトが返す値は、スコアでなく、勝ち/負け
  - スコア差ではなく、勝率を最大化するようにプレイする
    - リードしているときは安全に
    - 負けている時は無理な手も
  - 勝率最大化により、対GNU Go勝率が3割台から6割以上に跳ね上がった



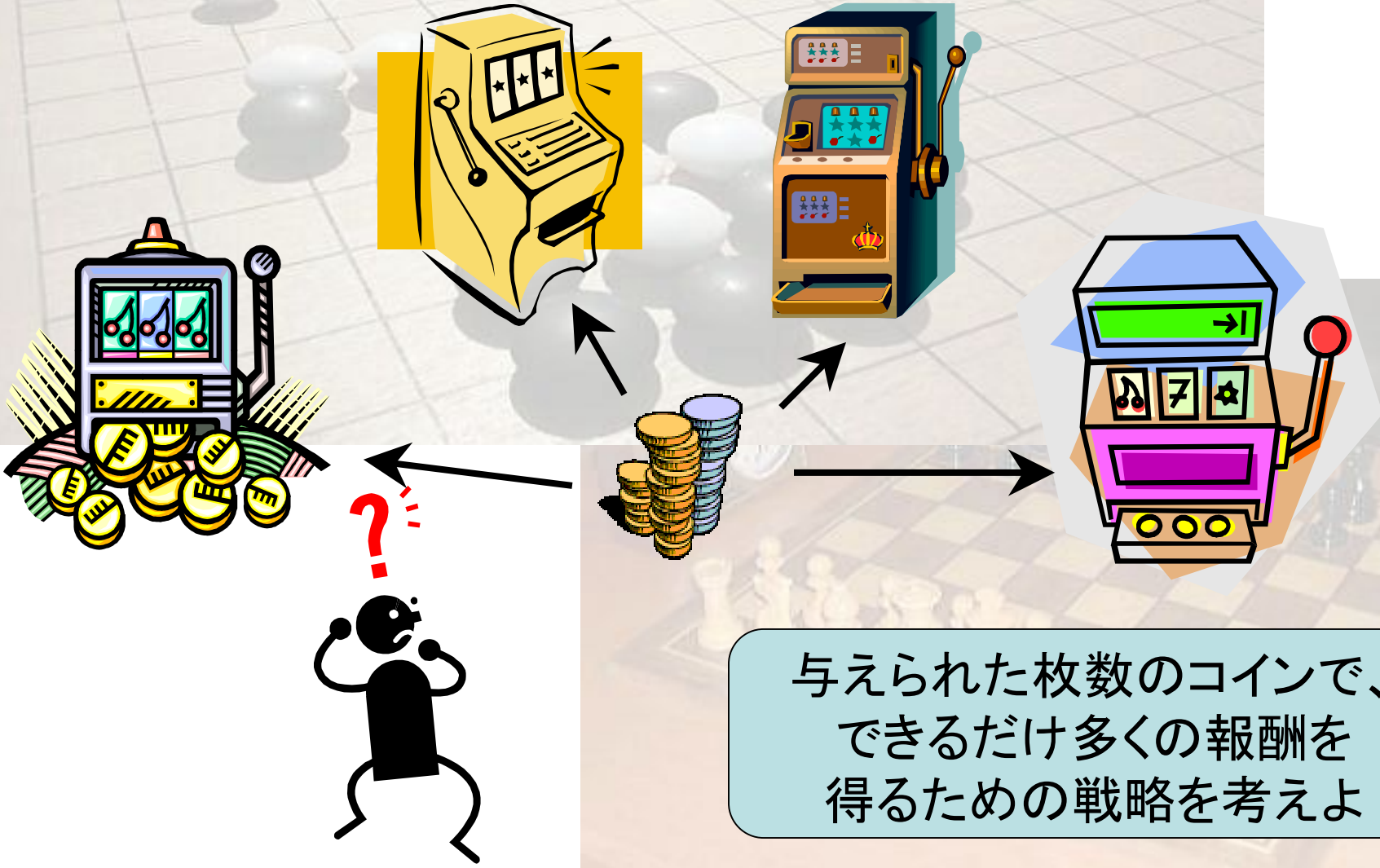




理論的背景

- Multi-Armed Bandit 問題
  - 統計学や機械学習の分野で研究されてきた
  - Multi-Armed Bandit とは？
    - 腕が複数あるスロットマシンのこと(空想上の存在)
    - One-Armed Bandit とはスロットマシンの俗称

Multi-Armed Bandit 問題

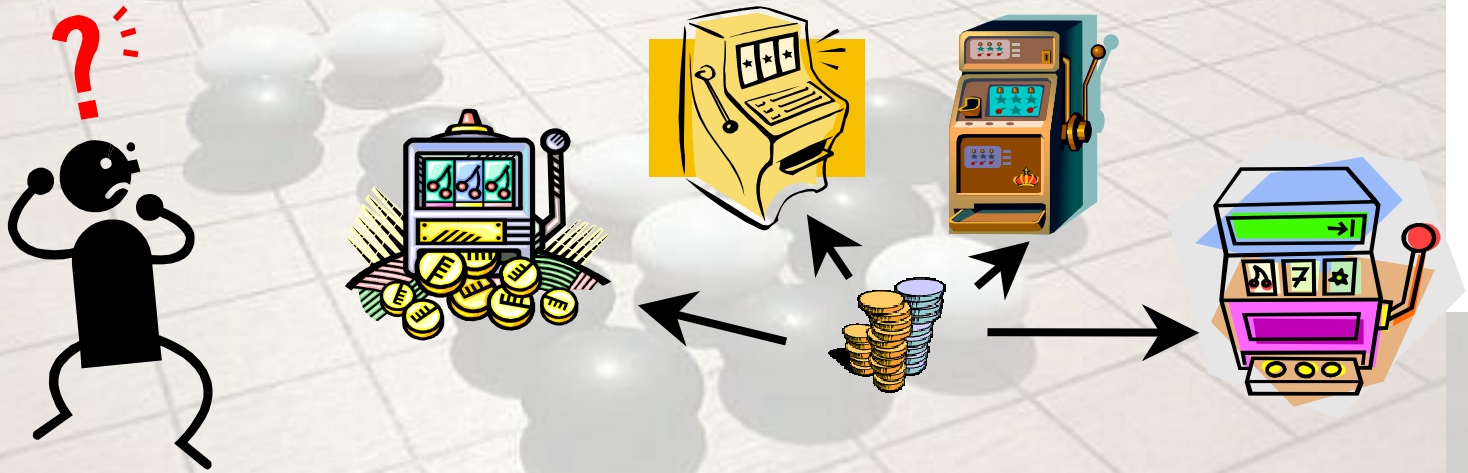


与えられた枚数のコインで、
できるだけ多くの報酬を
得るための戦略を考えよ

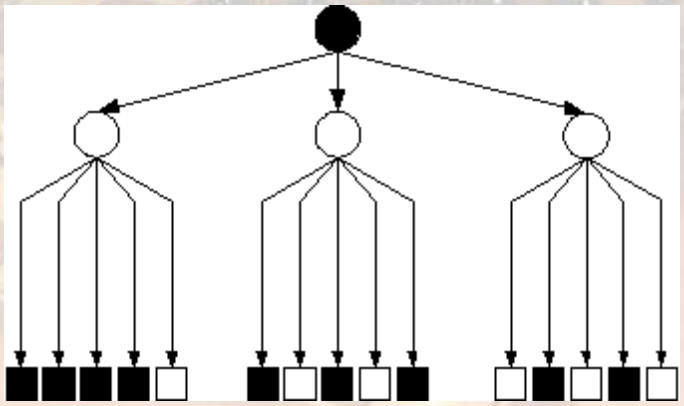
最善の戦略は？

- Multi-Armed Bandit 問題の最善の戦略は知られている [Lai and Robbins 1985]
  - しかし、計算量、メモリ消費ともに大きいために実際にはあまり用いられない
    - 各確率分布同士のKL情報量を計算する必要がある
- よって、計算量が小さく、かつ性能もそれほど悪くない戦略が求められる

全部に同じ枚数を投入しよう！
そして平均を比べればいい？



原始モンテカルロ囲碁と
同様の戦略
つまり全然ダメ



UCB1という戦略

- 各マシンについてUCB1値という値(Upper Confidence Bound)を計算 [Auer, Cesa-Bianchi, Fischer 2002]
  - UCB1値が最大になるマシンにコインを投入

$$\overline{X}_j + c\sqrt{\frac{2\log n}{n_j}}$$

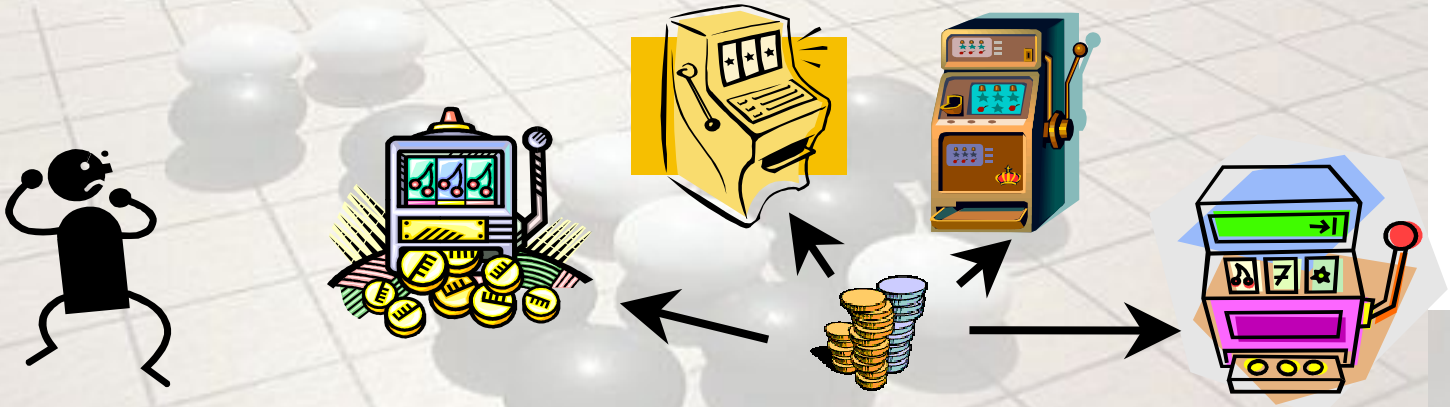
$\overline{X}_j$: j番目のマシンの報酬の期待値

n : それまでに投入したコイン数の合計

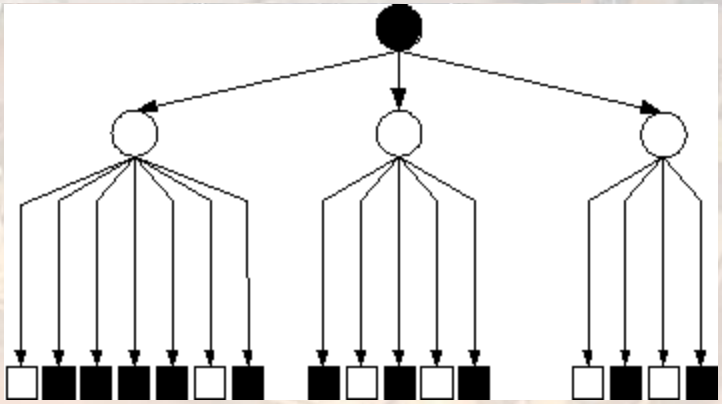
n_j : j番目のマシンに投入したコインの数

c : 期待値の値域によって決まる定数

有望なマシンにたくさんコインを投入しよう！
それがつまりUCB1



有望な手に多くの
プレイアウトを割当てる

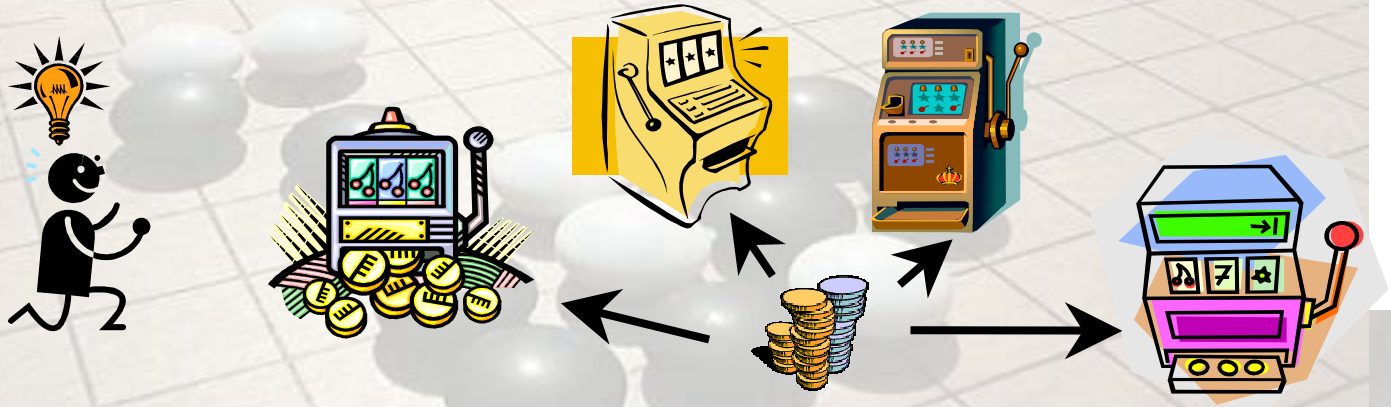


UCT (UCB applied to Trees)

- CrazyStoneの成功を受けて提案された木探索アルゴリズム [Kocsis and Szepesvári 2006]
  - UCB1を木探索に応用
  - UCB1値の高い候補手を辿って探索を行う
  - 末端の候補手でプレイアウトの回数が閾値を超えると、その手を展開する
  - 探索回数nが大きくなると、UCB1値が以下のように、期待値に収束することが証明されている

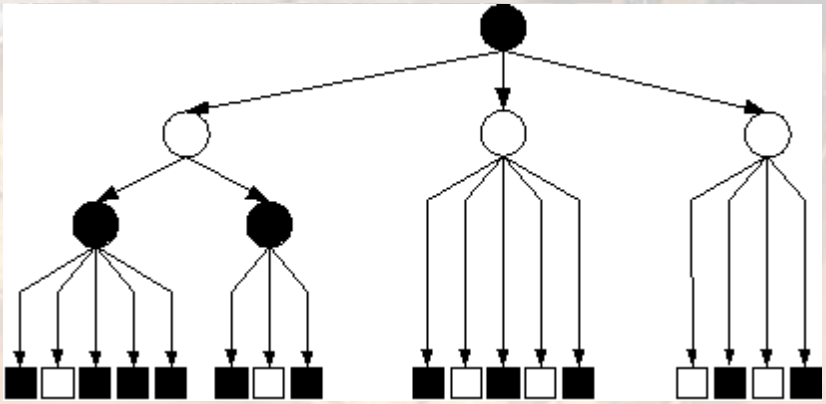
$$\overline{X}_j + c\sqrt{\frac{2\log n}{n_j}} \quad \Rightarrow \quad \overline{X}_j + O\left(\frac{\log n}{n}\right)$$

UCTを使えば深さ2以上の木でも最善手に到達する！



最初にUCTを取り入れた
囲碁プログラムが
MoGo (冒頭でプロと対戦)

[Gelly et al. 2006]



その後の進歩

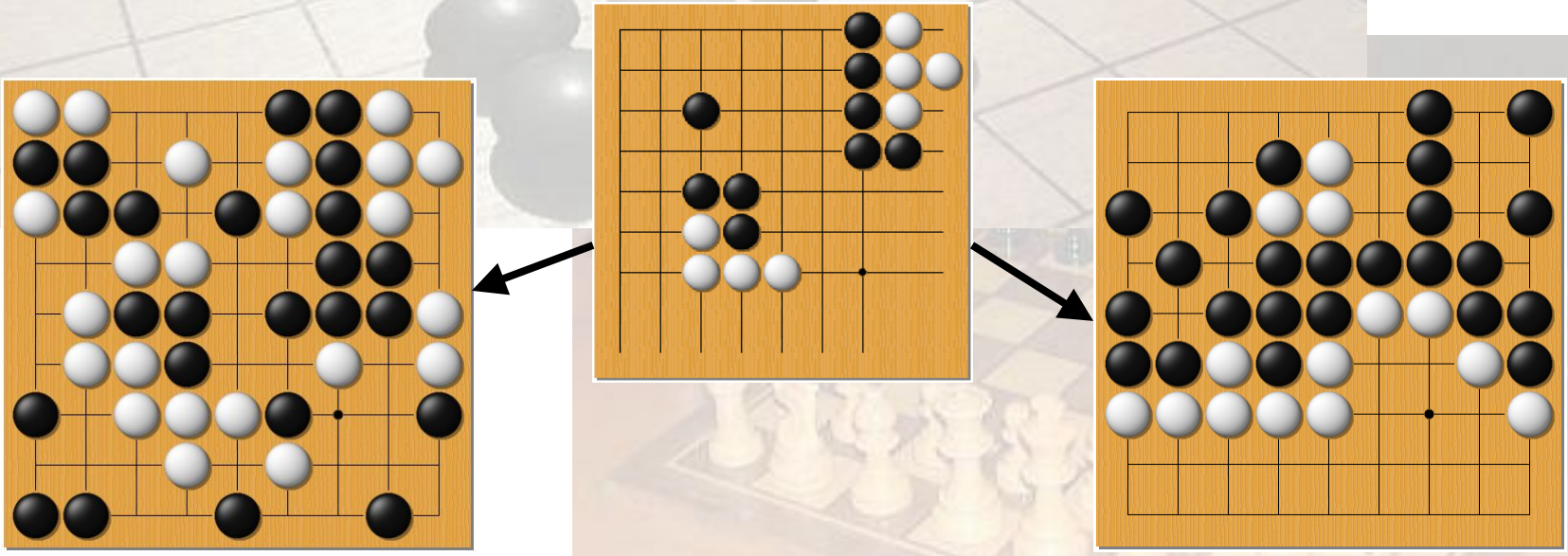
- MoGoがUCTを採用して猛威を奮って以降、CrazyStoneを含め、多くのプログラムがUCTを採用
  - Computer Olympiad、電通大で開催されたUEC杯コンピュータ囲碁大会などでモンテカルロ木探索を用いたプログラムが上位を独占
  - 全て、UCTか又は同様に木が成長するモンテカルロ木探索を用いている
- 19路盤でも強くなった
  - 当初は9路盤はアマ3級程度、19路盤では非常に弱かった
  - 現在では19路盤でもアマ有段者並み(CrazyStoneはKGSという囲碁サイトで2級＝普通の碁会所なら二段?)
- 何が改良されたのか説明したい

探索部分の改良

- Progressive Widening
  - 囲碁の知識を用い、良さそうな手から順に候補手をソートしておく
    - それを徐々に探索木に加えていく
- AMAF (All Moves As First)
  - プレイアウト中に打たれた初手のみを用いるのが通常の考え方だが、AMAFでは、全ての手を初手に打ったとみなす
- UCTのパラメータの調整
- UCTよりも最善手を優遇する探索手法

プレイアウトの改良

- 初期のCrazyStoneのプレイアウトは単純
  - 19路盤では非常に弱かった
- パターンを用いてプレイアウトを改良
  - 速度は数分の1になったが、棋力は大幅に向上



初期のCrazyStone
(秒間4万回程度?)

強化版CrazyStone
(秒間1万回程度?)

なぜ囲碁に有効なの？

- プレイアウトで普通に終局するゲームだから
  - チェスや将棋では普通に終局を迎えるのは難しい
    - 現在、プレイアウトと探索を組み合わせる研究などが行われている
  - オセロや五目並べは終局に至る
    - 囲碁同様に有効であると思われるがまだ研究途上
- 囲碁では、
  - 最善手と次善手の価値の差が小さい(ことが多い)
  - 手順に関係なくある位置を占めておけば有利という点が多い

モンテカルロ木探索の弱点

- 細く長い正解手順がある場合
  - 最善手が1手だけある、という局面が長手順連続すると、確率的に正解にたどり着かない
- 例
  - シチョウ：プレイアウトをパターンで強化して回避
  - 死活、攻め合い：まだ対処法は不明
    - 山下さんは、探索との組合せなどを試しているらしい

今後の展望

- モンテカルロ木探索の利点
  - 単純に強い
  - プログラミングの労力が少ない
    - 探索部分とプレイアウトの実装だけ
    - プレイアウトの強化には機械学習も有効
- 多くの研究者が参入
  - 機械学習のプロなど
- 並列化の研究も行われている
  - 冒頭のMoGoは256コアのクラスタを用いていた
- 現在も日々、強化されている

- 今後が非常に楽しみです

参考文献

- *P. Auer, N. Cesa-Bianchi and P. Fischer*, Finite-time analysis of the multi-armed bandit problem, *Machine Learning*, vol. 47, pp 235-256, 2002.
- *R. Coulom*, Computing Elo Ratings of Move Patterns in the Game of Go, *Computer Games Workshop*, 2007.
- *S. Gelly, Y. Wang, R. Munos and O. Teytaud*, Modification of UCT with patterns in Monte-Carlo Go, Technical Report No.6062, INRIA, 2006.
- *L. Kocsis and C. Szepesvári*, Bandit Based Monte-Carlo Planning, *LNCS vol.4212 (ECML 2006)*, pp. 282-293, 2006.
- *T. L. Lai and H. Robbins*, Asymptotically efficient adaptive allocation rules, *Advances in Applied Mathematics*, vol. 6, pp. 4-22, 1985.
- *H. Yoshimoto, K. Yoshizoe, T. Kaneko, A. Kishimoto and K. Taura*, Monte Carlo Go Has a Way to Go, *AAAI-06*, pp. 1070-1075, 2006.