モンテカルロ法による ゲームAIの可能性

美添　一樹

yoshizoe@acm.org

CEDEC

CESA Developers Conference 2009

スライドの最後に、当日
説明しきれなかった
内容の補足があります。

自己紹介

- (最初の)大学院生時代には並列計算を研究
- その後、某研究所に就職、携帯関係の研究開発
- なぜか大学院に戻って、人工知能の研究
- 今はいわゆるポスドクで、量子計算機の研究中
- コンピュータ囲碁の研究も続けている

- 専門はたぶん「探索アルゴリズム」
- 囲碁は自称三段
- 強い囲碁プログラムを作って自慢したい…けど時間がない
- あとゲームは好きです、いろいろと

あらすじ：モンテカルロ木探索(MCTS)

- コンピュータ囲碁を革命的に強くするアルゴリズムが発見された(2006年)
  - なぜ、2005年までは囲碁が弱かったのか
  - ついでに他のゲームAI研究についてちょっと紹介
- MCTSはどういうアルゴリズムか
  - 理論的に奥が深い (でも今日は理論は省略)
  - 見た目の性質も面白い
  - 長所と短所を紹介
- 特に、汎用性が高い
  - 現在の応用例について紹介
    - 二人ゲーム　：囲碁, 将棋, アマゾン, Hex, Lines of Action 等
    - その他ゲーム：ハーツ, カタン, Magic the Gathering, さめがめ 等
    - ゲーム以外　：最適化、プランニング、バイオメトリクスなど
- ゲームAI開発者に新しい選択肢を提供できれば嬉しい

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

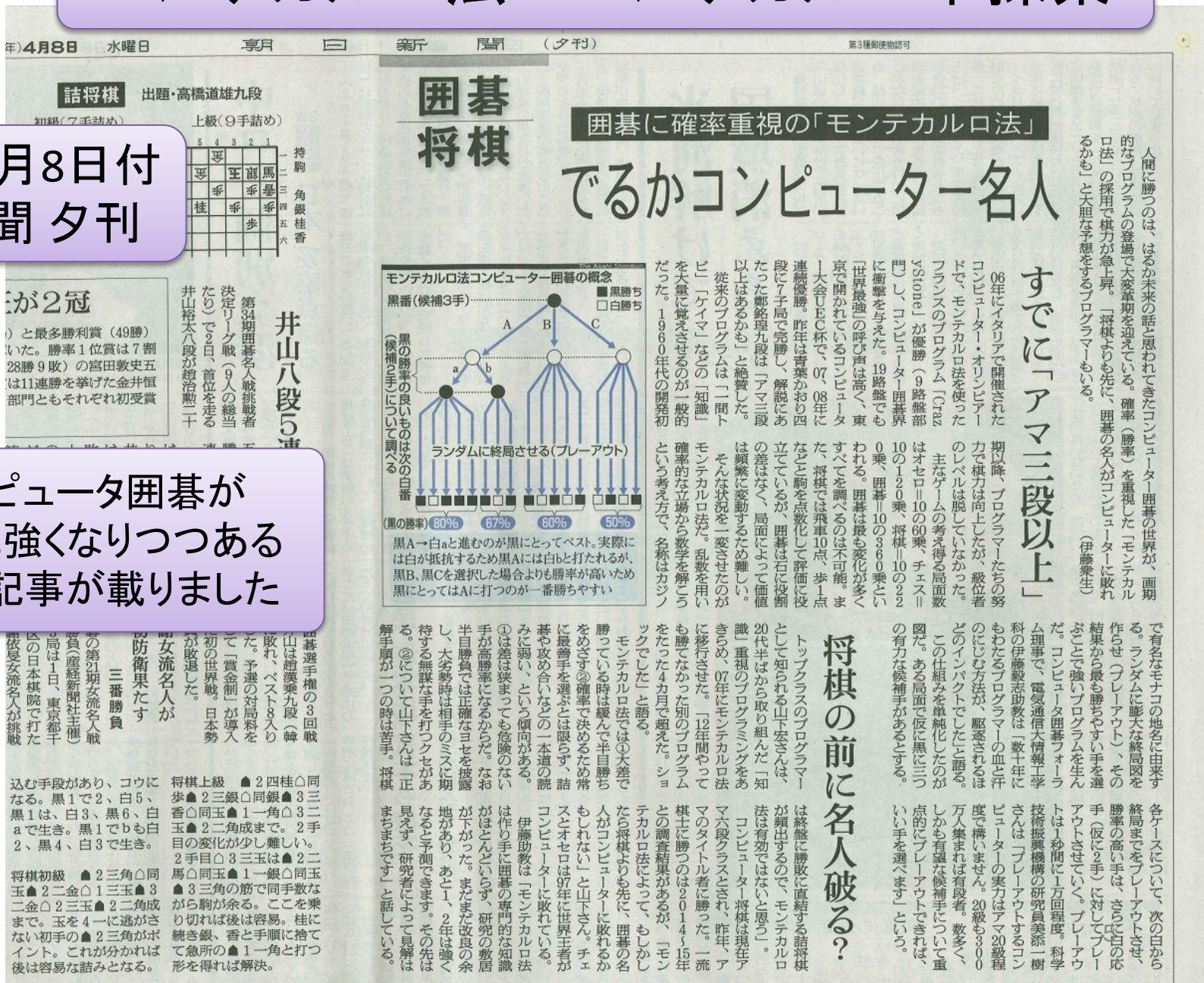
ここが革命です、念のため

- 囲碁が弱かったから新しいアルゴリズムが産まれた
  - それが他の問題にも波及しつつある

モンテカルロ法＝モンテカルロ木探索

2009年4月8日付
朝日新聞 夕刊

コンピュータ囲碁が
急激に強くなりつつある
という記事が載りました



年)4月8日 水曜日 朝日新聞 (夕刊) 第3種郵便物認可

詰将棋 出題・高橋道雄九段

初級(7手詰め) 上級(9手詰め)

持駒 角銀桂香

井山八段5連

井山八段が2冠

第34期囲碁名人戦挑戦者決定リーグ戦(9人の総当たり)で2日、首位を走る井山裕太八段が趙治勲二十

囲碁選手権の3回戦 井山は趙漢乗九段(韓 敗れ、ベスト8入り た。予選の対局料を て「賞金制」が導入 初の世界戦。日本勢 が敗退した。

囲碁 将棋

囲碁に確率重視の「モンテカルロ法」

でるかコンピューター名人

すでに「アマ三段以上」

人間に勝つのは、はるか未来の話と思われてきたコンピューター囲碁の世界が、画期的なプログラムの登場で大変革期を迎えている。確率(勝率)を重視した「モンテカルロ法」の採用で棋力が急上昇。「将棋よりも先に、囲碁の名人がコンピューターに敗れるかも」と大胆な予想をするプログラマーもいる。(伊藤寮生)

06年にイタリアで開催されたコンピューター・オリンピードで、モンテカルロ法を使ったフランスのプログラム「Crazy Stone」が優勝(9路盤部門)し、コンピューター囲碁界に衝撃を与えた。19路盤でも「世界最強」の呼び声は高く、東京で開かれているコンピュータ大会UEC杯で、07、08年に連続優勝。昨年は青葉かおり四段に7子局で完勝し、解説にあたった鄭銘瑝九段は「アマ三段以上はあるかも」と絶賛した。

従来のプログラムは「一間トビ」「ケイマ」などの「知識」を大量に覚えさせるのが一般的だった。1960年代の開発初期以降、プログラマーたちの努力で棋力は向上したが、級位者のレベルは脱していなかった。

主なゲームの考え得る局面数はオセロ=10の60乗、チェス=10の120乗、将棋=10の220乗、囲碁=10の360乗といわれる。囲碁は最も変化が多くすべてを調べるのは不可能。また、将棋では飛車10点、歩1点などと駒を点数化して評価に役立てているが、囲碁は石に役割の差はなく、局面によって価値は頻繁に変動するため難しい。

そんな状況を一変させたのがモンテカルロ法だ。乱数を用い確率的な立場から数学を解こうという考え方で、名称はカジノで有名なモナコの地名に由来する。ランダムに膨大な終局図を作らせ(プレーアウト)、その結果から最も勝ちやすい手を選ぶことで強いプログラムを生んだ。コンピュータ囲碁フォーラム理事で、電気通信大情報工学科の伊藤毅志助教は「数十年にもわたるプログラマーの血と汗のにじむ方法が、駆逐されるほどのインパクトでした」と語る。

この仕組みを単純化したのが図だ。ある局面で仮に黒に三つの有力な候補手があるとする。

黒A→白aと進むのが黒にとってベスト。実際には白が抵抗するため黒Aには白bと打たれるが、黒B、黒Cを選択した場合よりも勝率が高いため黒にとってはAに打つのが一番勝ちやすい

将棋の前に名人破る?

トップクラスのプログラマーとして知られる山下宏さんは、20代半ばから取り組んだ「知識」重視のプログラミングをあきらめ、07年にモンテカルロ法に移行させた。「12年間やっても勝てなかった別のプログラムをたった4カ月で超えた。ショックでした」と語る。

モンテカルロ法では①大差で勝っている時は緩んで半目勝ちをめざす②確率で決めるため常に最善手を選ぶとは限らず、詰碁や攻め合いなどの一本道の読みに弱い、という傾向がある。①は差は狭まっても危険のない手が高勝率になるからだ。なお半目勝負では正確なヨセを披露し、大劣勢時は相手のミスに期待する無謀な手を打つクセがある。②について山下さんは「正解手順が一つの時は苦手。将棋は終盤に勝敗に直結する詰将棋が頻出するので、モンテカルロ法は有効ではないと思う」。

コンピューター将棋は現在アマ六段クラスとされ、昨年、アマのタイトル者に勝った。一流棋士に勝つのは2014~15年との調査結果があるが、「モンテカルロ法によって、もしかしたら将棋よりも先に、囲碁の名人がコンピューターに敗れるかもしれない」と山下さん。チェスとオセロは97年に世界王者がコンピューターに敗れている。

伊藤助教は「モンテカルロ法は作り手に囲碁の専門的な知識がほとんどいらず、研究の敷居が下がった。まだまだ改良の余地があり、あと1、2年は強くなると予測できます。その先は見えず、研究者によって見解はまちまちです」と話している。

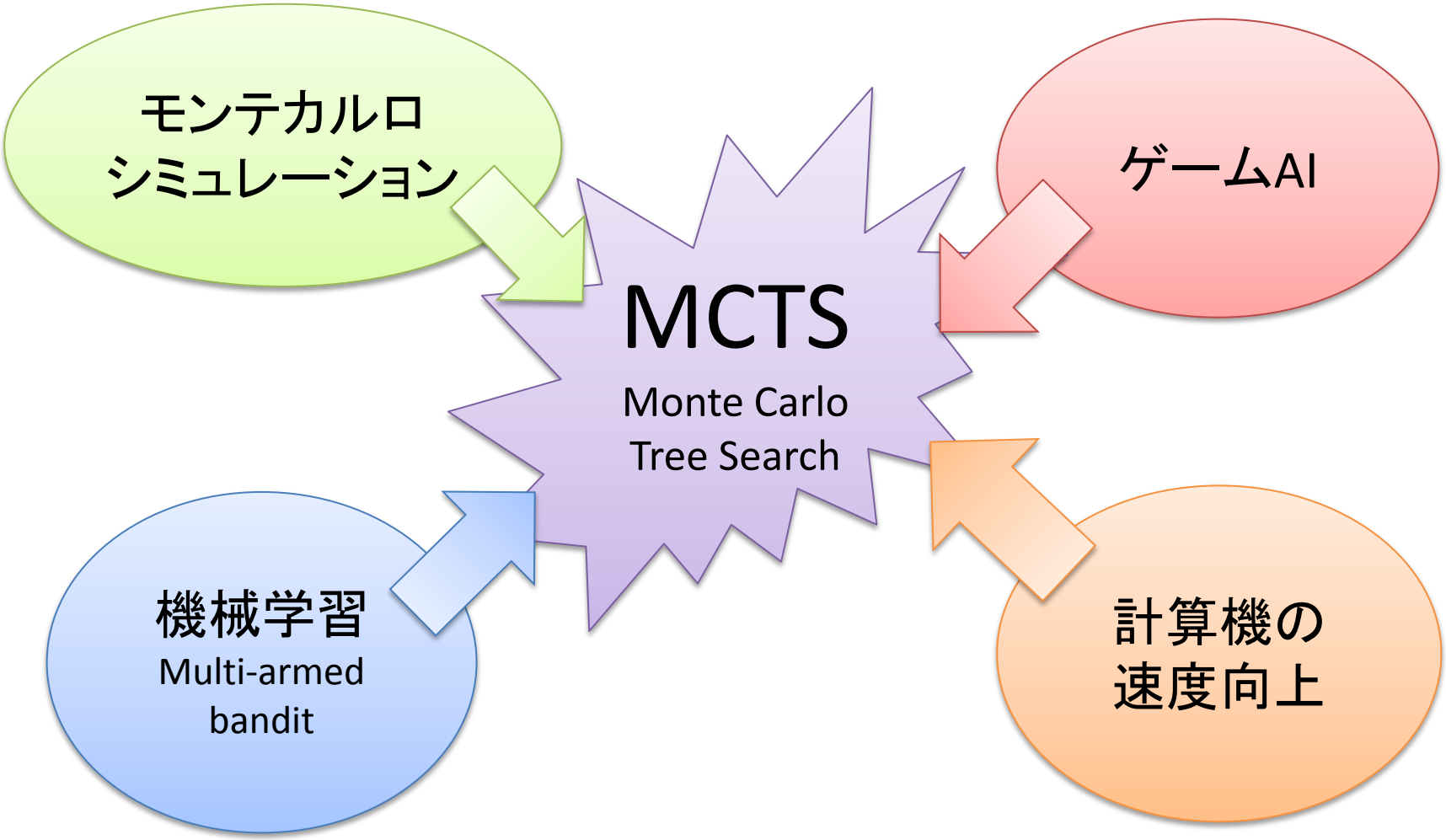
各ケースについて、次の白から終局までをプレーアウトさせ、勝率の高い手は、さらに白の応手(仮に2手)に対してプレーアウトさせていく。プレーアウトは1秒間に1万回程度。科学技術振興機構の研究員美添一樹さんは「プレーアウトするコンピューターの実力はアマ20級程度で構いません。20級も300万人集まれば有段者。数多く、しかも有望な候補手について重点的にプレーアウトできれば、いい手を選べます」という。

込む手段があり、コウになる。黒1で2、白5、黒1は、白3、黒6、白aで生き。黒1でbも白2、黒4、白3で生き。

将棋初級 ▲2三角△同玉▲2二金△1三玉▲3二金△2三玉▲2二角成まで。玉を4一に逃がさない初手の▲2三角がポイント。これが分かれば後は容易な詰みとなる。

将棋上級 ▲2四桂△同歩▲2三銀△同銀▲3二香△同玉▲1一角△3二玉▲2二角成まで。2手目の変化が少し難しい。2手目△3三玉は▲2二馬△同玉▲1一銀△同玉▲3三角の筋で同手数ながら駒が余る。ここを乗り切れば後は容易。桂に絡き銀、香と手順に捨てて急所の▲1一角と打つ形を得れば解決。

4つの背景


モンテカルロ
シミュレーション
ゲームAI
MCTS
Monte Carlo
Tree Search
機械学習
Multi-armed
bandit
計算機の
速度向上

ゲームAI研究

- 人工知能の分野では、まじめな研究対象
  - 商業ゲームもよく研究対象になる
    - 学会によってはFPSや戦略シミュレーションなども
  - AIのトップレベルの学会でもゲームが題材になる
    - 主にチェスや囲碁だが、freeciv に関する論文が出たことも
- 様々な技術のテストベッドとして有用である
  - 研究のための研究はすみやかに淘汰される傾向がある
    - 理屈倒れの研究は見向きされない
  - min-max探索(alpha-beta探索)の発展
    - チェス、オセロ、チェッカーなど
      - 二人ゼロサム完全確定情報ゲーム
  - 知識ベースのアプローチ
    - 初期のチェスなど
  - 機械学習(ニューラルネットなど)
    - バックギャモンなど
  - モンテカルロシミュレーションの活用
    - バックギャモン、Scrabble、ポーカー など

二人零和完全確定情報ゲーム

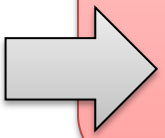
二人(1対1)ゲーム

零和である

全て情報が見えている

不確定要素がない
(サイコロを振らない)

ここに大きなギャップがある
これはなぜか？



2005年時点でのコンピュータの強さ

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	1997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3級くらい? (深い突っ込み禁止)

二人零和完全確定情報ゲームは、ゲーム理論的には一番簡単

しかし人間との比較では…？

ポーカー(テキサスホールデム)は人間のチャンピオンにプログラムが勝利(2008年)

バックギャモン、Scrabbleなどは、かなり前から人間より強い

余談

- ポーカー (テキサスホールデム) はコンピュータが強い
  - 人間ペア 対 プログラム 2対2の対戦
  - AAAIで開催 (全米人工知能会議 : AIでは2番目に格が高い)
  - 2008年にPolaris(プログラム)が人間のチャンピオンペアに勝利
    - The Second Man-Machine Poker Competition
    - http://www.cs.ualberta.ca/~games/poker/man-machine/
  - オンラインポーカーのサイトで、リアルマネーを使えるところもあるので、かなり問題

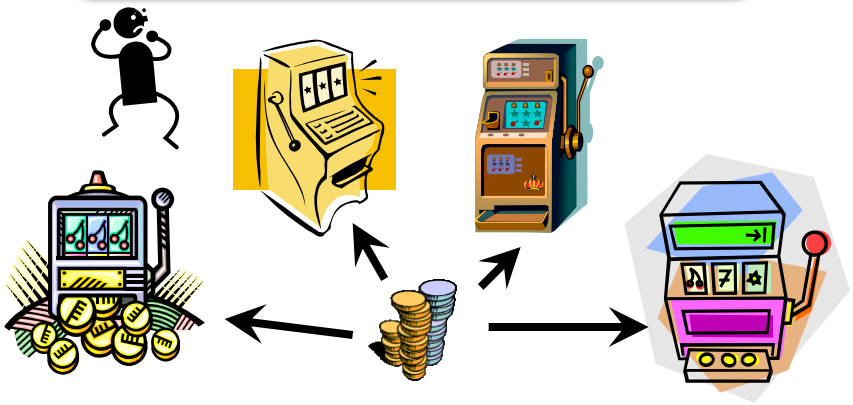
- コンピュータはいろいろなゲームで強いが例外もある
  - 一人ゲーム : 倉庫番
    - 実は人間にしか解けない問題が多数ある
  - 二人ゲーム : 囲碁
    - ただし、今は急激に強くなっている
  - 多人数ゲーム : これはたくさん弱い物がある

機械学習

- 機械学習の理論
  - 長年研究されてきた強固な理論
  - あとでちょっと説明

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

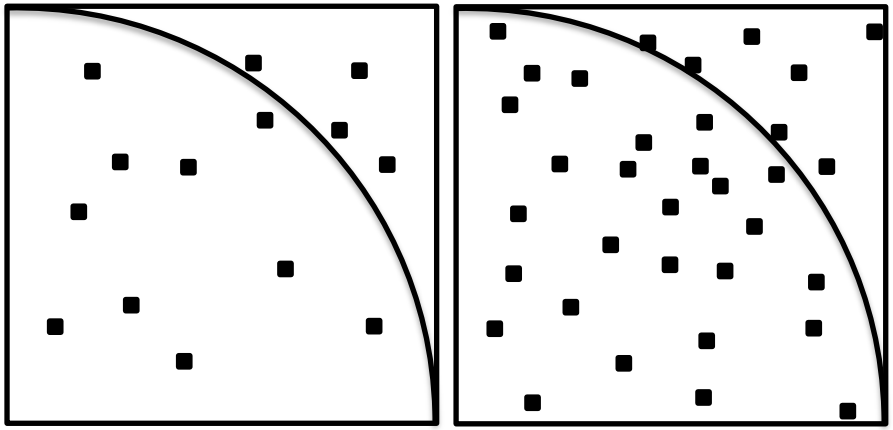


コンピュータの速度向上

- 速いコンピュータを必要とするアルゴリズム
  - alpha-beta探索
    - 知識ベースより強くなった
  - LDPC符号 (1950年代提案)
    - 地上デジタル放送など
  - トマスロのアルゴリズム (1960年代提案)
    - superscalarプロセッサ
  - ボナンザメソッド
    - 将棋で大きな成功
  - モンテカルロ木探索
    - これが本題
- コンピュータが十分速いことに最初に気づく人は偉い

モンテカルロシミュレーションとは？

- 一番簡単な例
  - (よく説明に使われる例)
  - 円周率を求める
- 乱数がたくさん必須
  - モンテカルロはカジノで有名
- 主な応用例
  - 物理シミュレーション、経済シミュレーション
- 様々な分野で使われている歴史のあるアルゴリズム
  - 本当は凄く奥が深い



ランダムに
たくさん点を打つ

数える

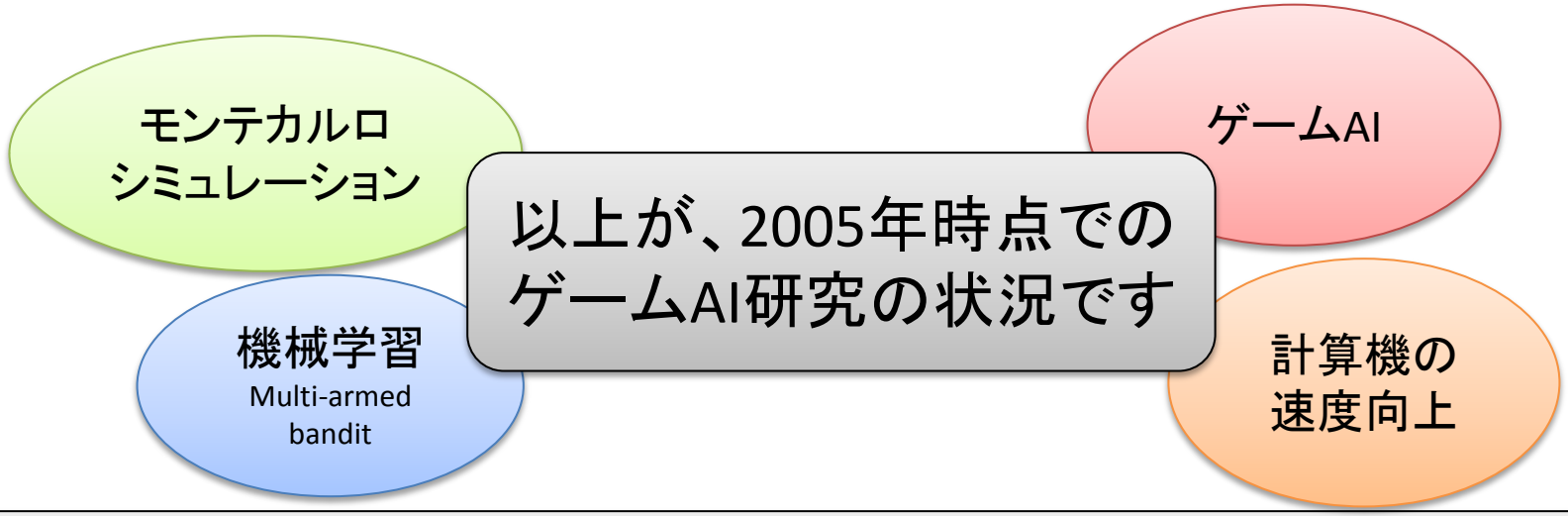
割り算する

点が多いほど正確

投げやりです

モンテカルロシミュレーションとゲーム

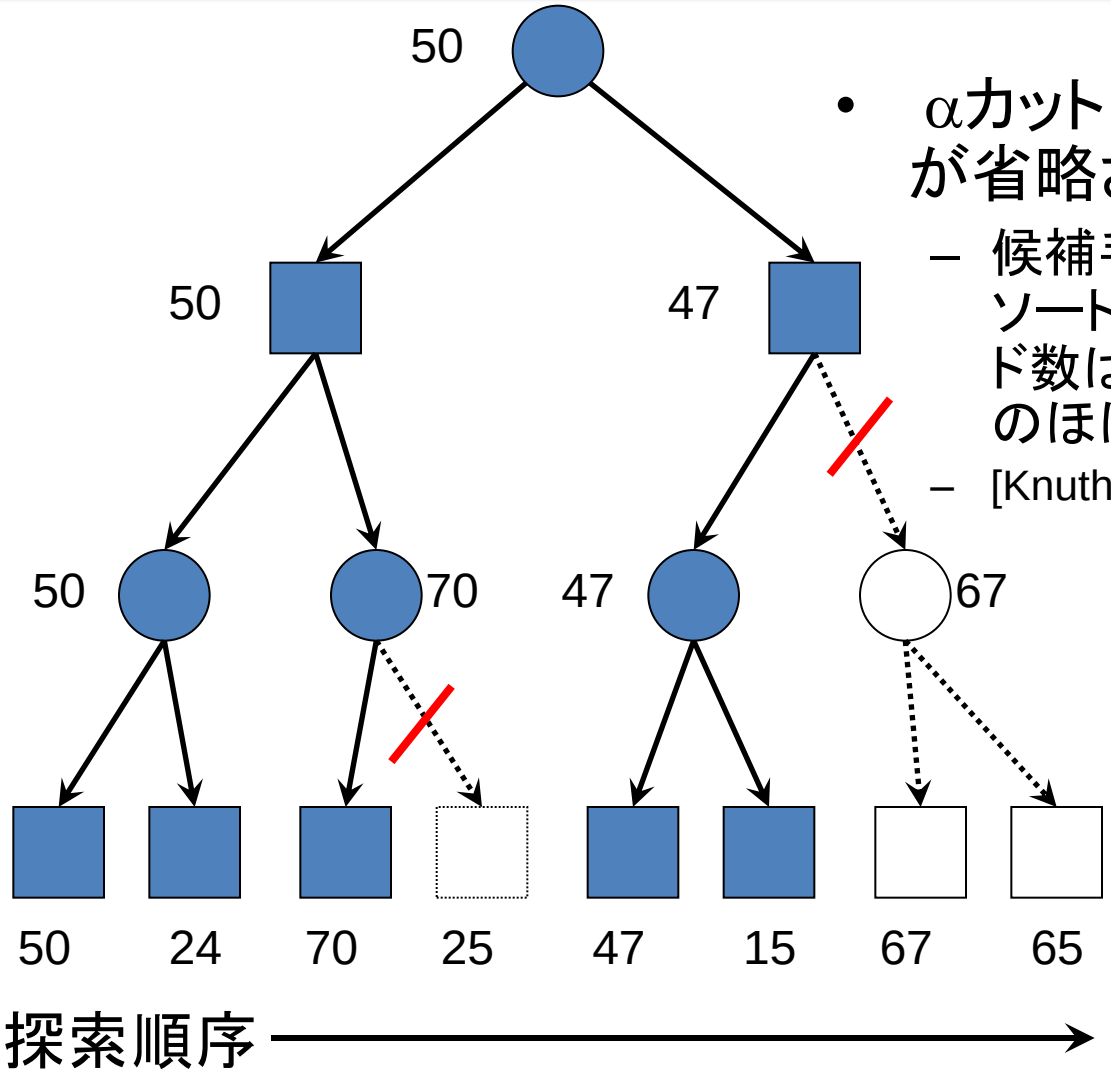
- 不確定要素があるゲームにモンテカルロシミュレーションを使うのは自然なアイデア
  - バックギャモン
  - Scrabble
  - ポーカー(テキサスホールデム)
- このアイデアはかなり成功した
  - 人間のチャンピオンレベルに近いものも
- 完全確定情報ゲームにモンテカルロシミュレーションを使うアイデアが実はあった
  - 原始モンテカルロ囲碁



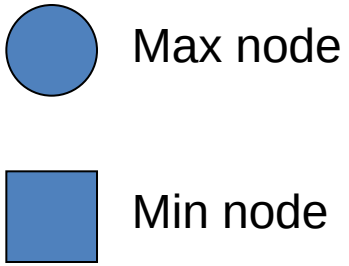
なぜ囲碁だけこんなに弱いのか？

- 二人零和完全情報ゲームでは完全な仲間外れ
  - チェスもオセロも将棋も強いのに・・・
- 他のゲームだって強いものが多い
  - ポーカーとかバックギャモンとか
- メジャーなゲームなのに弱いのは囲碁だけ

普通の二人零和完全情報ゲーム
min-max探索＋αβ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
  - [Knuth and Moore 1975]



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
- ところが、9路盤の探索空間はチェス以下
  - それでも2005年までは19路同様に弱かった
  - どっちも(建前は)アマ初段くらい
- これはおかしい、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か？

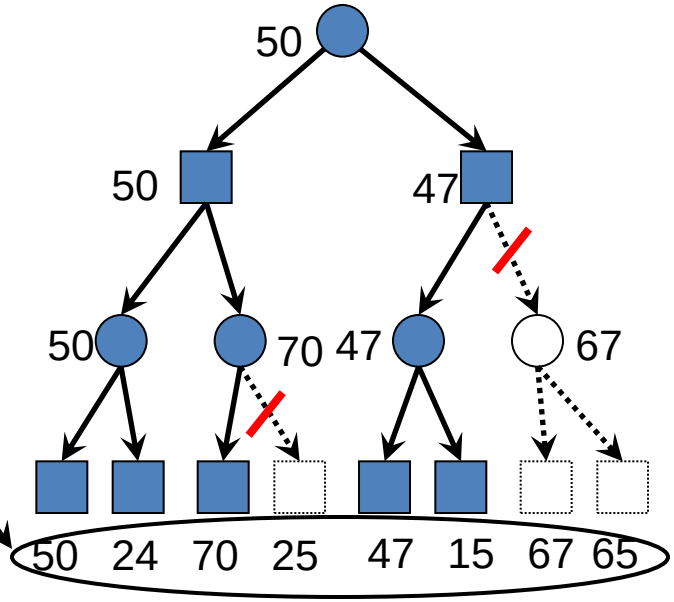
	探索空間 (可能な局面数)
チェッカー	10^{20}
オセロ	10^{28}
チェス	10^{50}
将棋	10^{71}
囲碁(9路盤)	10^{38}
囲碁(19路盤)	10^{171}

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を示す

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

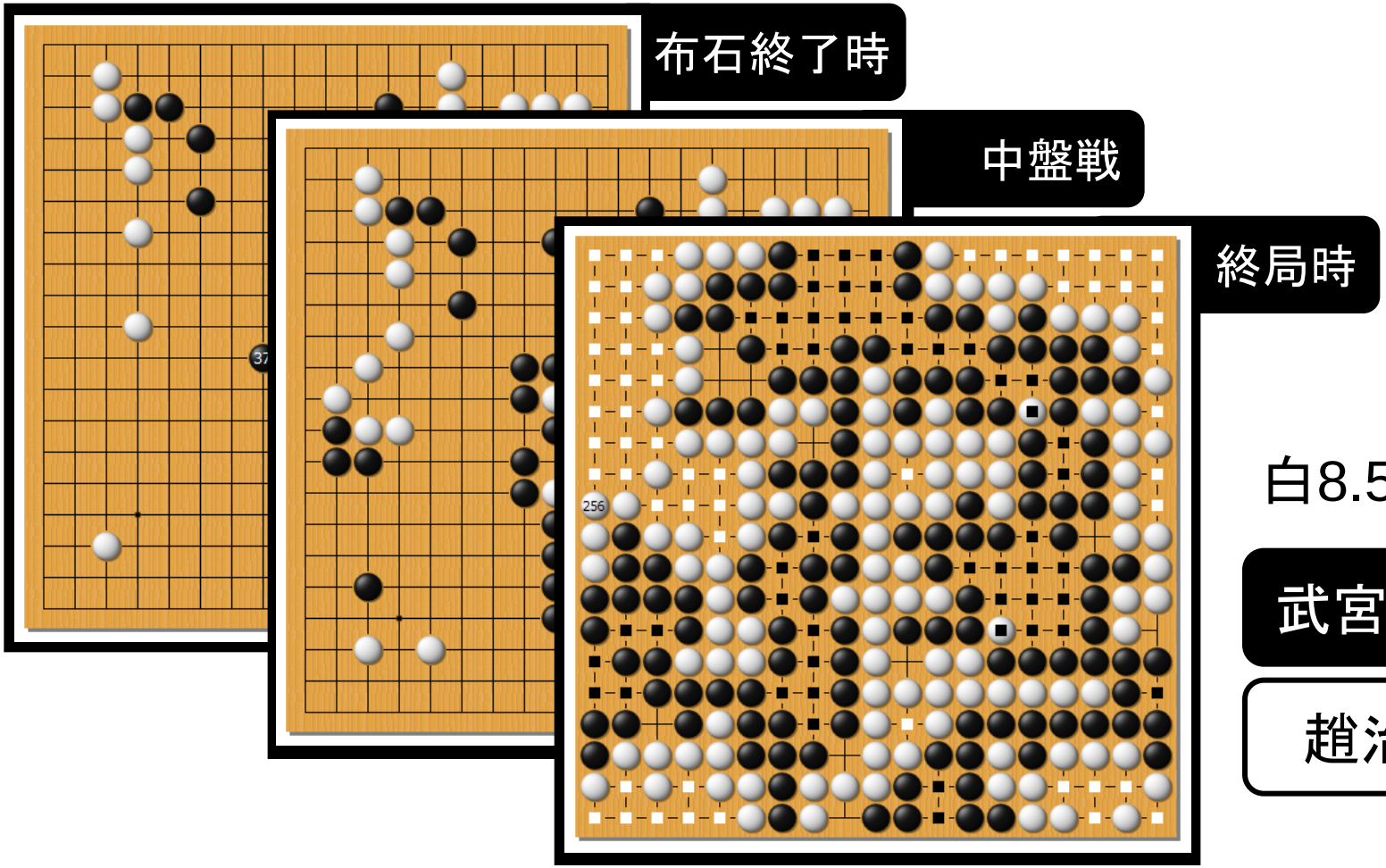
- オセロ
  - 隅や辺の重要な箇所のパターンを学習して評価関数を作成
  - オセロでの学習は簡単にうまくいく
    - logistelloやZebraが有名
- チェスや将棋
  - 駒の価値、玉の安全度、駒が自由に動けるか等
    - チェスの例：ポーン1点、ビショップとナイト3点、ルーク5点、クイーン9点、キング∞点
  - ボナンザメソッドなどもあり
    - 人間の棋譜から自動的に評価関数を作成

囲碁の評価関数の難しさ

- 石の価値は平等
  - なので、駒の価値などは用いることができない
- オセロのような明らかに特徴のある箇所が少ない
  - これは特に19路盤で顕著

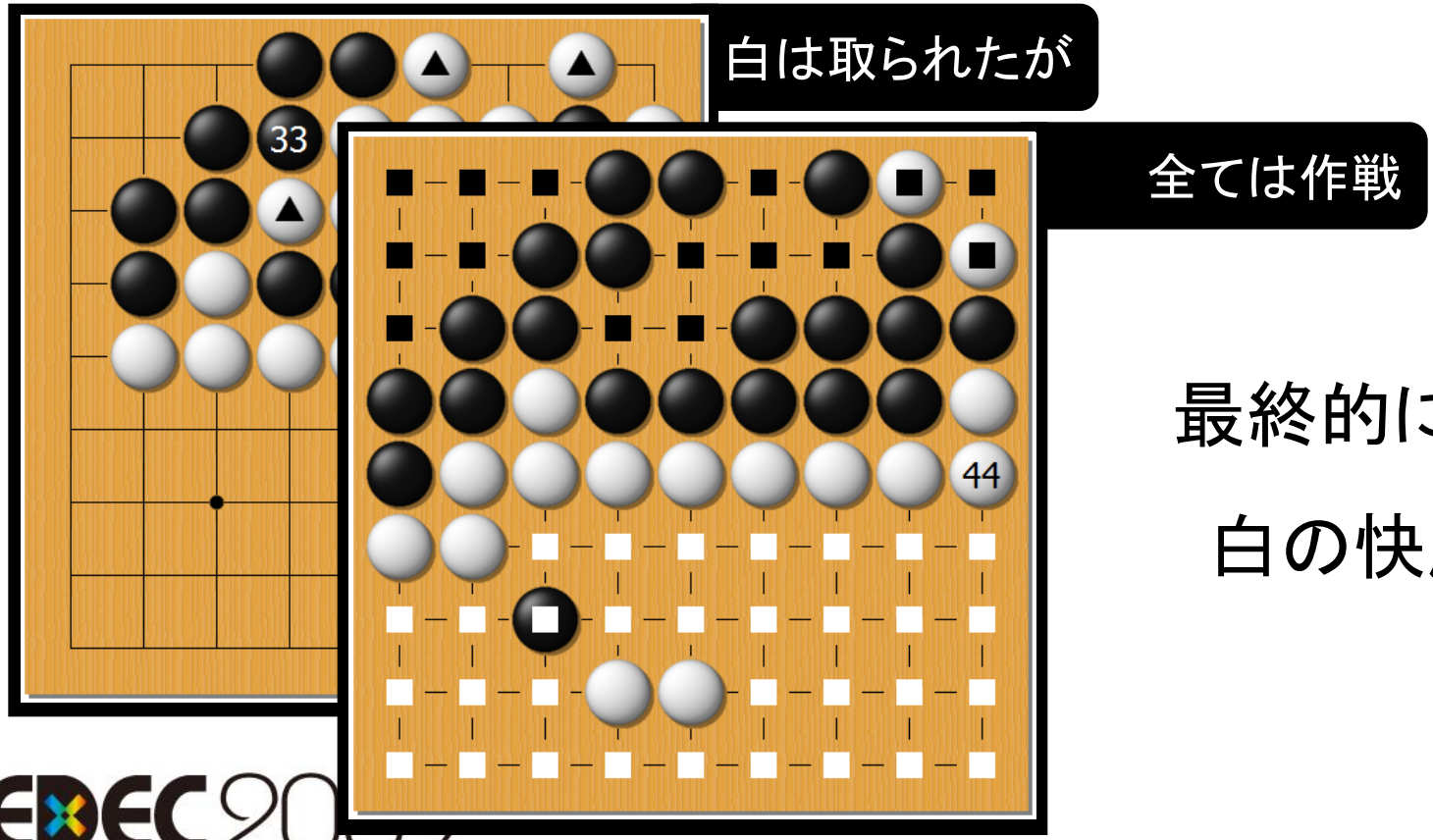
囲碁の評価関数の難しさ

- 領域の広さを競うなら、広さを基準にする？
  - しかし領域が確定するのはゲームの最後



囲碁の評価関数の難しさ

- 局所的な最善手≠全局的な最善手
  - 石を取るのは局所的には得
  - しかし捨石は基本的なテクニック



最終的には

白の快勝

囲碁の評価関数の難しさ

- 石の価値は平等
  - なので、駒の価値などは用いることができない
- オセロのような明らかに特徴のある箇所が少ない
  - これは特に19路盤で顕著
- 領域の広さを競うなら、広さを基準にする？
  - しかし領域が確定するのはゲームの最後
- 局所的な最善手≠全局的な最善手
  - 石を取るのは局所的には得
    - しかし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？

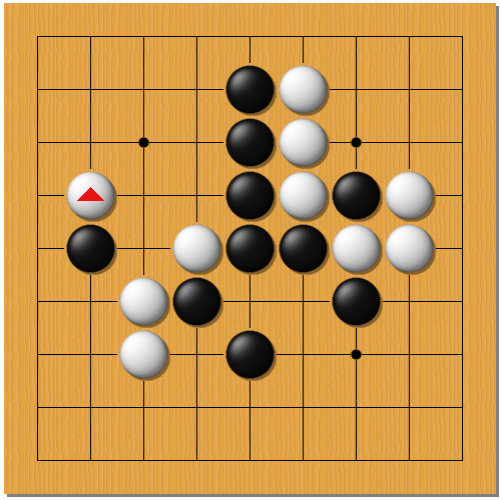
- ・・・説明不能です。
  - 特に中盤は難しいです
    - 石が厚かったり薄かったり
    - 形が良かったり悪かったり
    - 味が良かったり悪かったり
    - 地に辛かったり甘かったり
    - 石が軽かったり重かったり
  - 初段くらい無いと用語の意味が通じません・・・

つまり囲碁は難しい（難しかった）

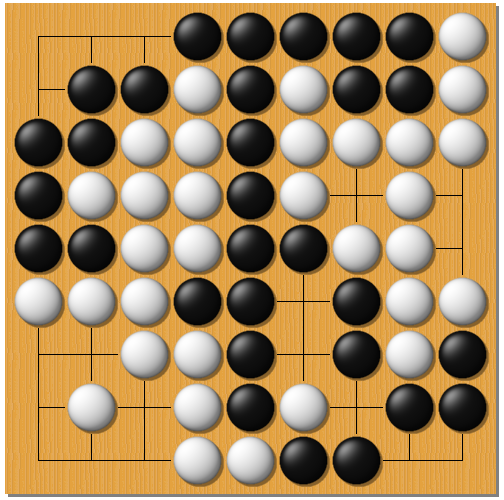
- チェスや将棋の駒得のような明らかな評価基準がない
- 何かの要素の足し算で局面の優劣を評価するのは難しい

- 評価関数は速く、正確である必要がある
  - 最低でも1秒に1万回くらいは計算できないとダメ
  - 囲碁の評価関数は、遅いか、不正確である
    - 遅い上に不正確、というと怒られるかな・・・

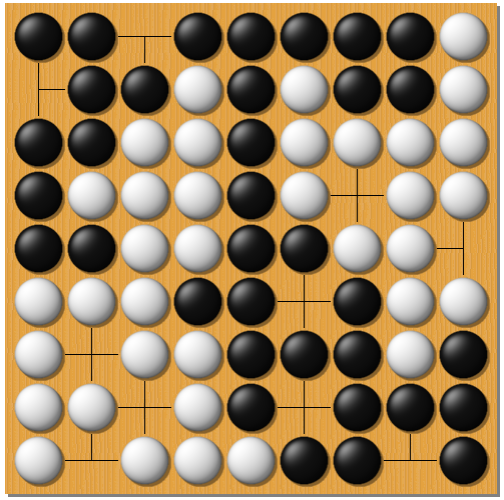
囲碁の評価関数は難しいが・・・



中盤の評価関数は
非常に難しい



しかし終局後なら
スコア判定は簡単



中国ルールの
終局図なら
もっと簡単

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（当然、ほぼ全て思考ルーチン）
  - 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路でも同じくらいの強さ

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

全部意味の違う値

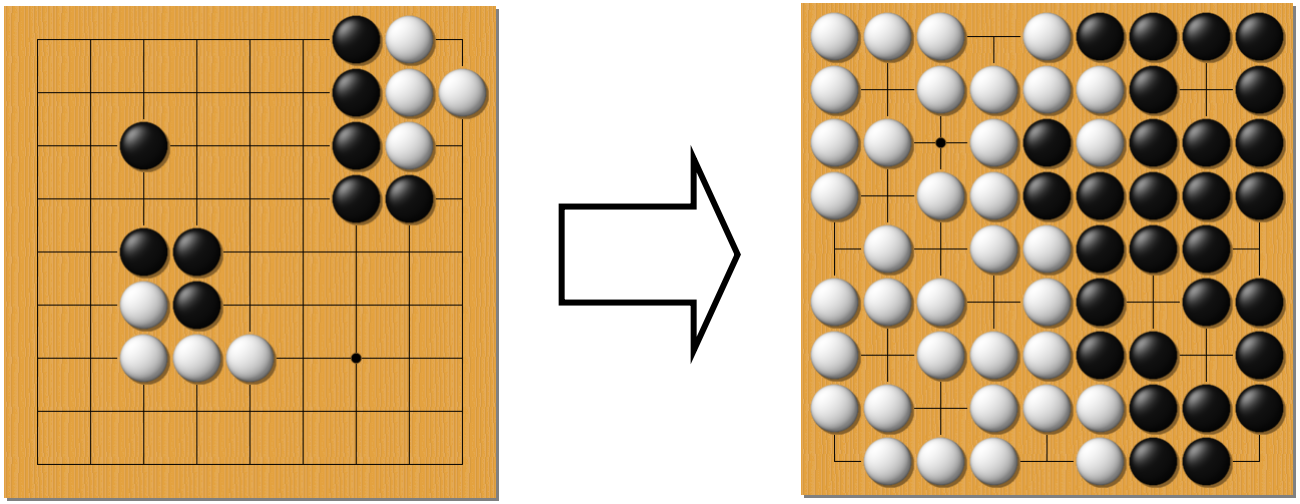
原始モンテカルロ囲碁

- 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
  - 「原始モンテカルロ囲碁」は説明の都合上つけた名前
- 変わったアイデアだと思われていた
  - 不確定な情報がないゲームにモンテカルロシミュレーションを使う？

プレイアウトとは

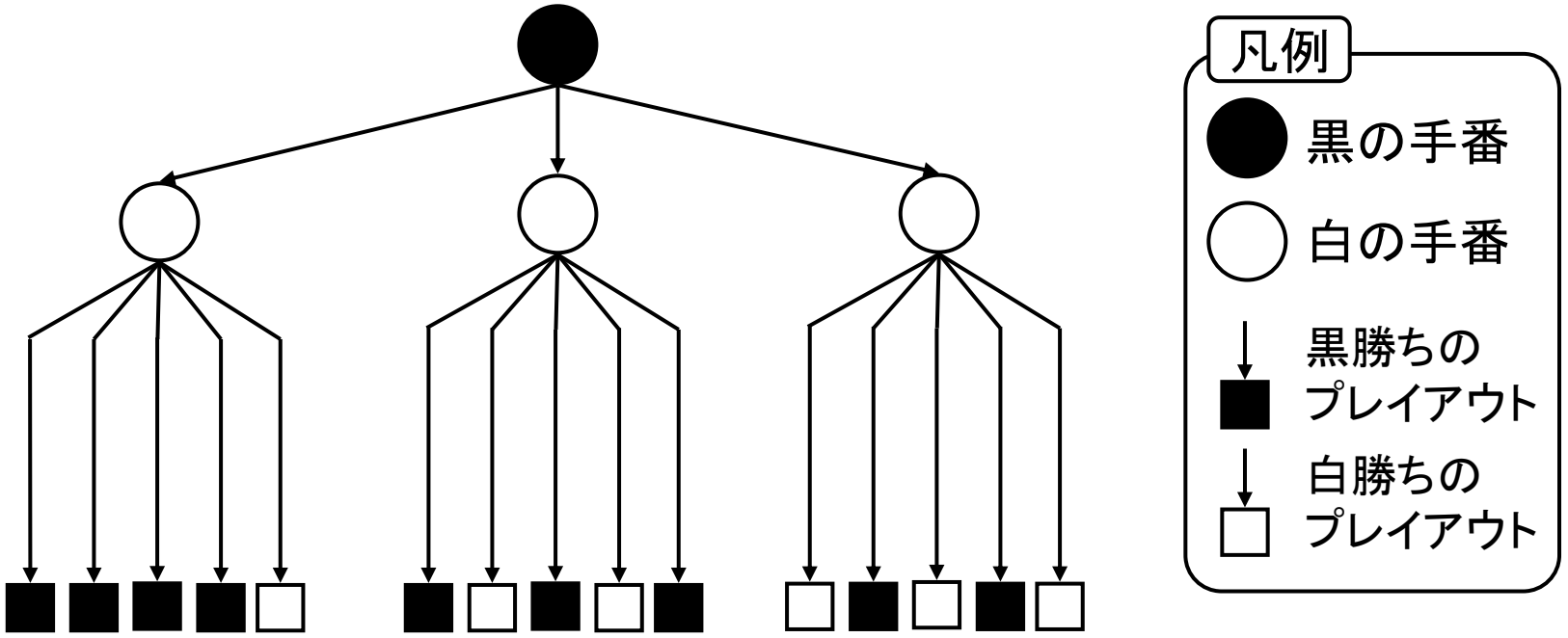
覚えてください

- 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 (新しい用語)
  - 普通の用語はシミュレーション
  - 機械学習だとエピソードとも



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

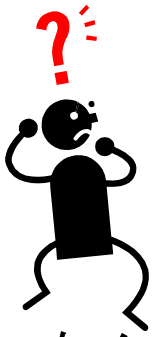


もちろん原始モンテカルロ囲碁は弱い

- 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(私も共著者)
  - 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の登場

- 2006年のComputer Olympiad 囲碁9路盤部門 優勝プログラム [Rémi Coulom 2006]
  - モンテカルロを使っているらしい
  - しかも打ち方が他のプログラムと全然違う…
    - 優勢だと手加減してきっちり僅差で勝つ
    - 負けていると無理な手を打ってくる



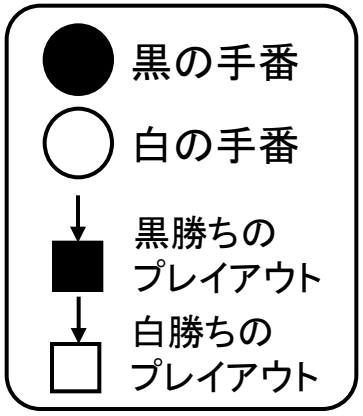
- 単純なモンテカルロ囲碁は弱いはず
  - 自分たちでそういう論文も書いたところなのに・・・なんで？

- CrazyStone は原始モンテカルロ囲碁を改良したアルゴリズムを用いていた
  - それがモンテカルロ木探索
  - コンピュータ囲碁界だけでなく、ゲームAI研究に革命を起こした

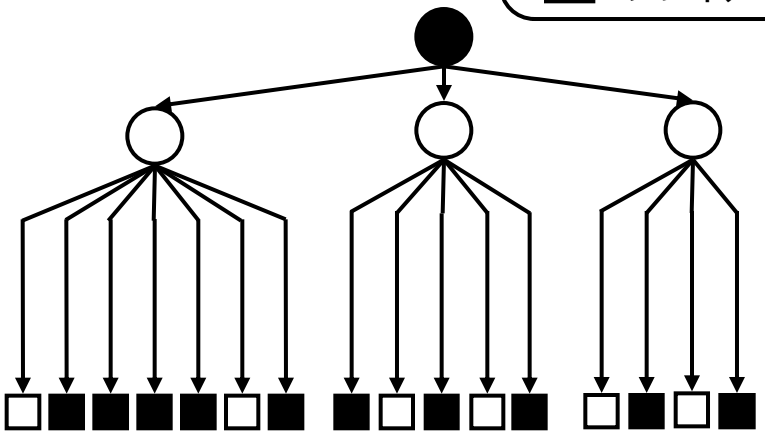
モンテカルロ木探索によるプログラム

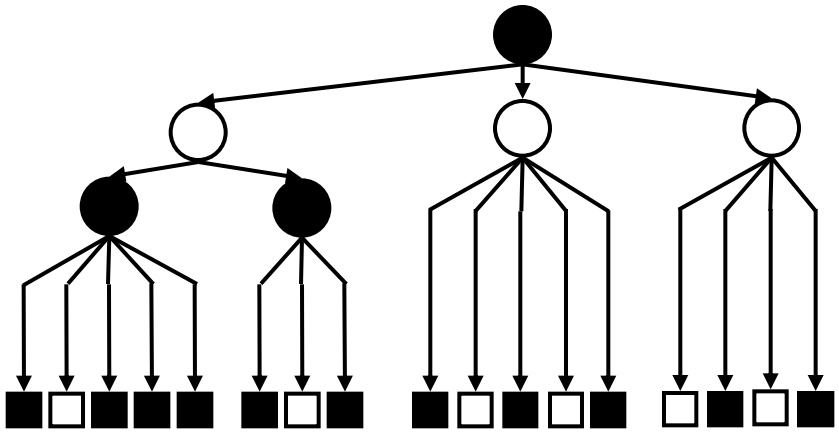
- 囲碁の評価関数は難しい
  - これは今でも本当だとみんな思っている
- しかし、囲碁でも終局した状態なら簡単に勝敗の判定が可能
  - 終局してるよ、と教えてくれれば、計算は簡単
- この性質をうまく利用したプログラムが CrazyStone

モンテカルロ木探索
Monte Carlo Tree Search



- 原始モンテカルロからの変更点は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成長する
- 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



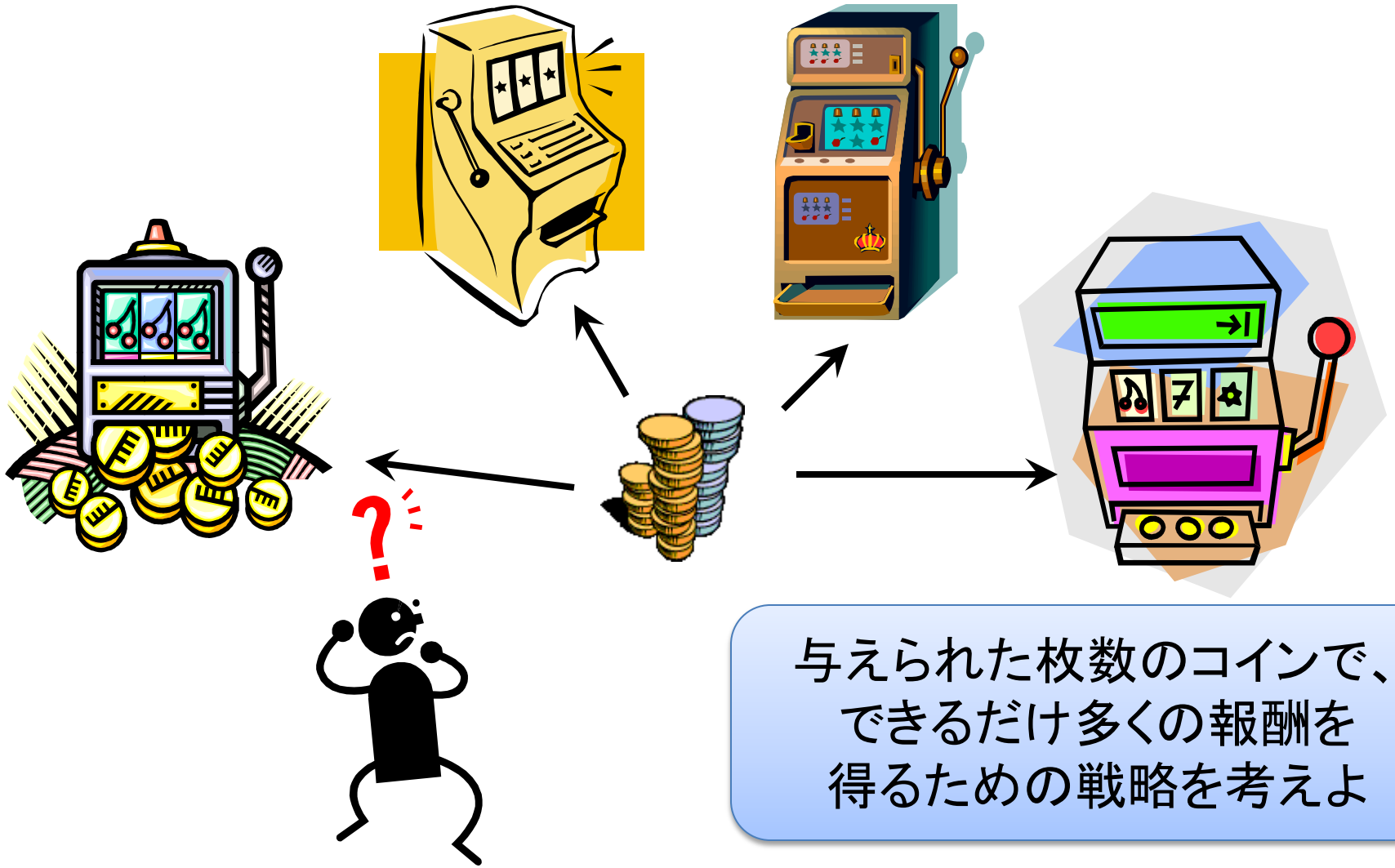


理論的背景

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 ＝「一本腕の山賊」＝「スロットマシーン」
    - 善良な人から金を盗んでしまう一本腕の悪いヤツ

Multi-Armed Bandit 問題



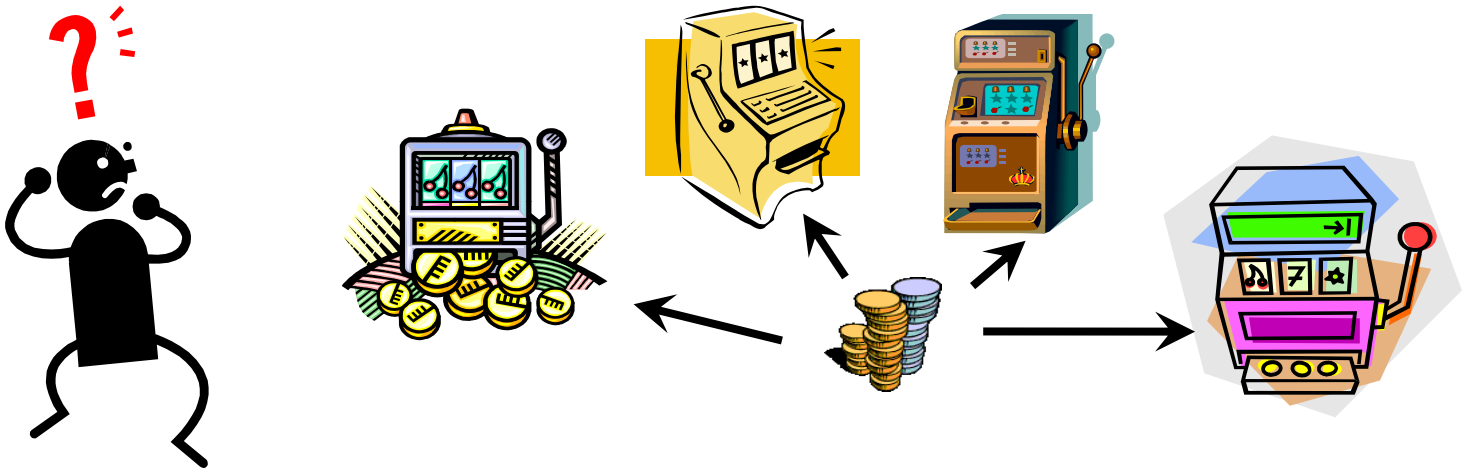
与えられた枚数のコインで、
できるだけ多くの報酬を
得るための戦略を考えよ

最善の戦略は？

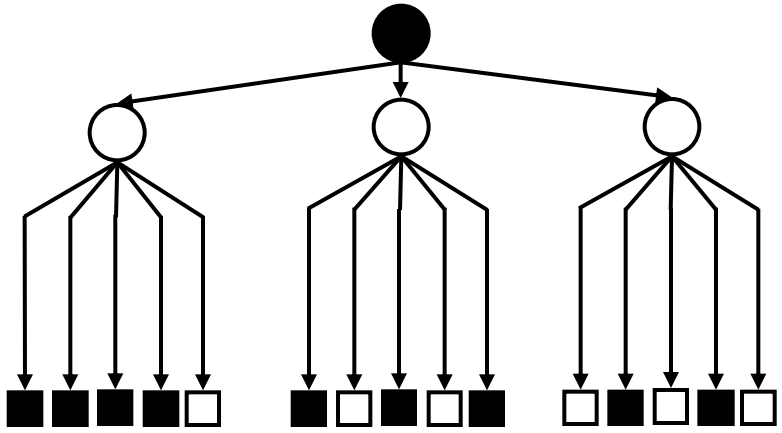
- Multi-Armed Bandit 問題の最善の戦略は知られている [Lai and Robbins 1985]
  - しかし、最善の戦略の性質が知られているだけで、実際に計算するのは大変

- よって、計算量が簡単で、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番目のマシーンの報酬の期待値

n ：それまでに投入したコイン数の合計

n_j ：j番目のマシーンに投入したコインの数

c ：アルゴリズムの性格を決める定数

定数c は、実際には実験して決めるべき

UCB1値の意味

期待値

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

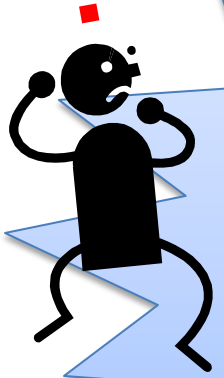
バイアスと呼ばれる値
コインが少ないほど多い

$\overline{X}_j$ ：j番目のマシーンの報酬の期待値

n ：それまでに投入したコイン数の合計

n_j ：j番目のマシーンに投入したコインの数

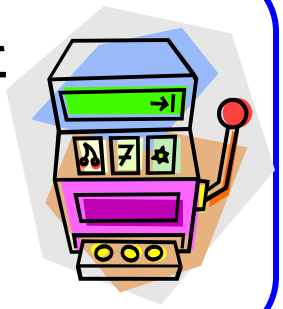
c ：アルゴリズムの性格を決める定数



コインが少ないマシーンほど、優遇するようにする！

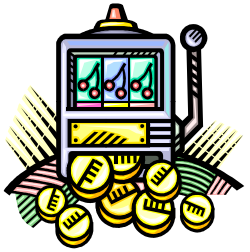
コインをちょっと投入した

ハズレばっかりだけど、
タダ運が悪いのかも

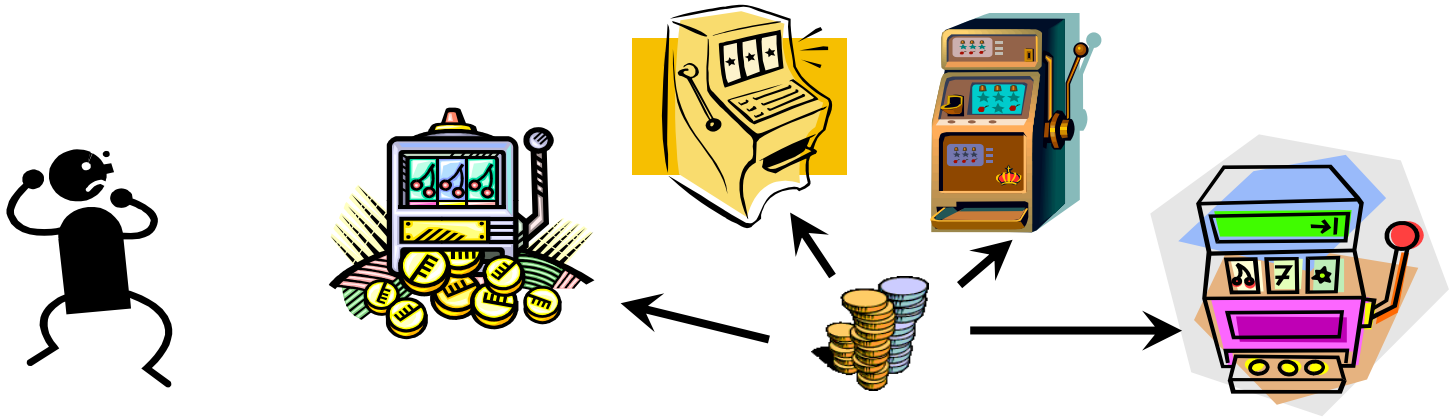


コインをいっぱい投入したけどハズレばっかり

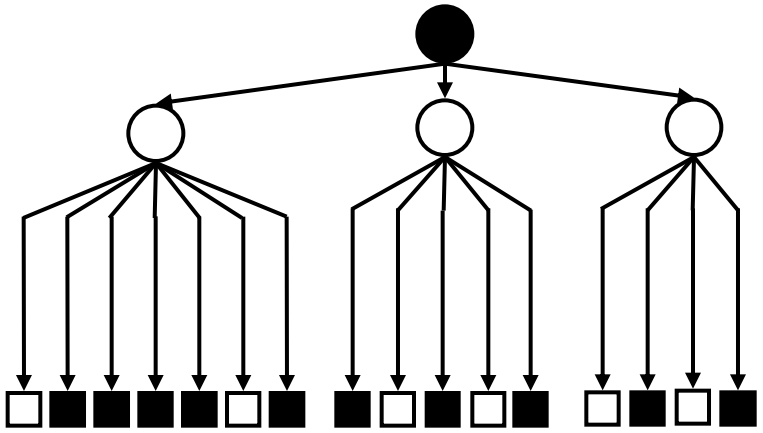
たぶん本当にダメ



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



理論的背景

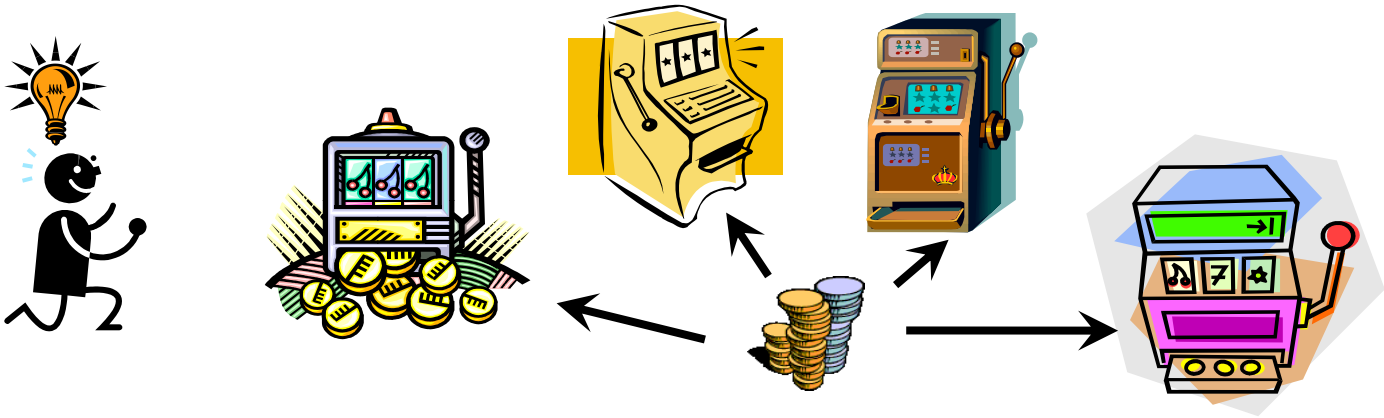
UCT (UCB applied to Trees)

- 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

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

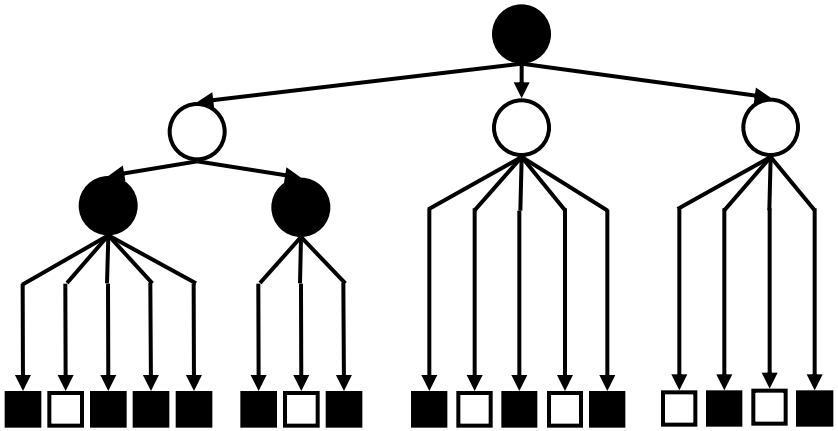
  - UCTはCrazyStoneの方法を改良し、さらに理論的な基盤を与えた

UCTを使えば深さ2以上の木でも（いつかは）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 [Gelly et al. 2006]



2006年に一気に成立

CrazyStone [2006 Rémi Coulom]

2006Computer Olympiad
囲碁9路盤部門で優勝

重要な概念
をほぼ網羅

勝率最大化
リードしているときは安全に、
負けているときは冒険をする

5月

UCT Algorithm [2006 Kocsis & Szepesvári]

最善解に収束する証明

9月

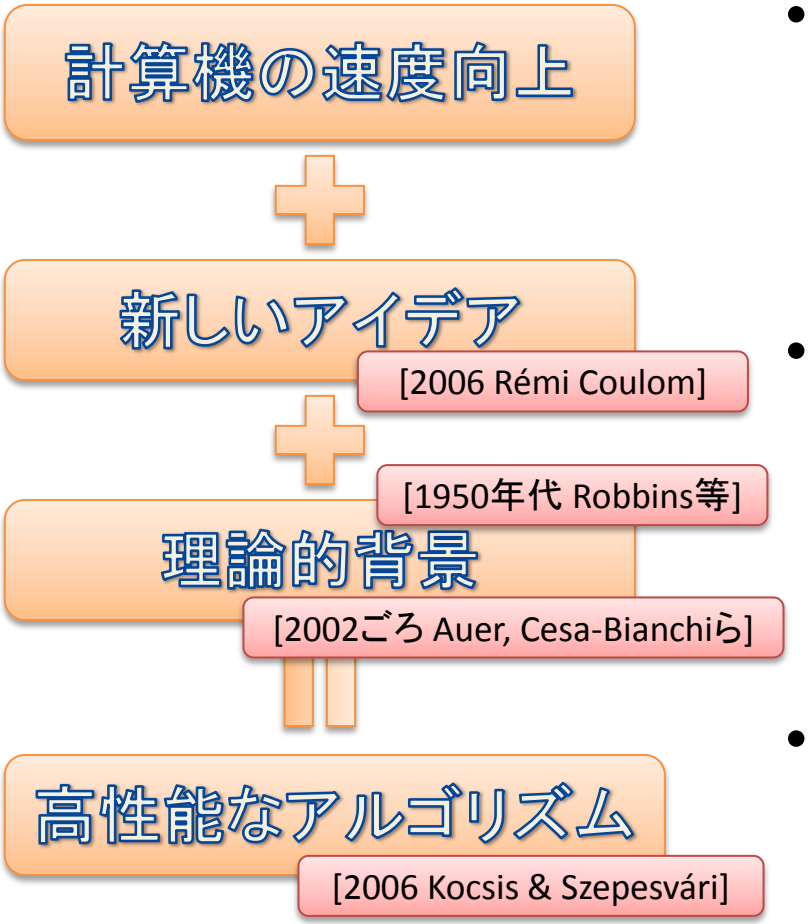
MoGo [2006 Gelly, Wang, Munos & Teytaud]

UCTを用いた初のプログラム
19路盤でアマ初段程度に到達

11月

全部2006年の出来事！

複数の背景からブレイクスルー



- コンピュータ囲碁研究の歴史は長い
  - 始まりは1960年代
  - しかし全然うまくいってなかった
- 山下さん(彩作者 兼 AI将棋作者)
  - 12年かけて作ったプログラムをMCTSで作ったプログラムが2ヶ月で逆転
    - 「暗黒面に墜ちた」
- 当初の私の感想
  - こんなアルゴリズムでうまくいくはずがない！
  - 私が間違ってました

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

MCTSの強化、Mogo、Zenの登場

- CrazyStoneは19路盤では弱かった
  - 9路盤はアマ3級程度、19路盤では非常に弱かった
- しかしMoGoが登場 (UCTを初めて使用)
  - 手生成のパターンを使って強化し、19路盤でも強くなった
  - 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
- 現在最強はZenというプログラム
  - 19路盤でアマ三段以上 (クセを見抜かれると微妙・・・)
  - 9月18日発売予定
    - 商品名「天頂の囲碁」http://soft.mycom.co.jp/pcigo/tencho/
  - 開発者は尾島陽児氏(当初はYamatoという仮名で活動)
    - RPGツクール、アスキーエンターテイメントソフトウェアコンテスト

コンピュータ囲碁の革命、かつ探索の革命

古典的な囲碁プログラム
(古典=2005年以前)

19路盤 2級から3級

9路盤 2級から3級

MCTS

Monte Carlo
Tree Search

近代的なプログラム
(近代的=2006年以後)

19路盤 2段以上

9路盤 アマ高段並

囲碁だけが弱かった

チェス、将棋、ポーカーなどはコンピュータが非常に強い

今までのアルゴリズムは、評価関数が無いとお手上げだった

2006年5月に発表されたMCTSによって状況が一変
[2006 Rémi Coulom]

9路盤では既に複数のプログラムがプロに勝利した

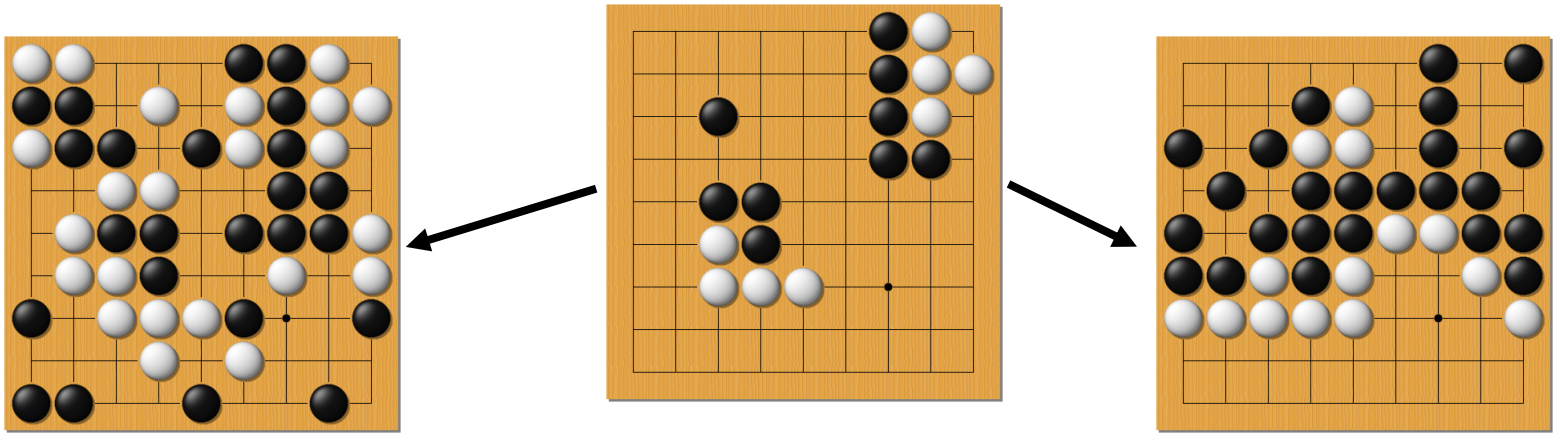
19路盤では、公開対局で、7子のハンデでプロに勝利している
・CrazyStone対 青葉かおり
・MoGo 対 周俊勲

木探索部分の改良

- 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
    - それを徐々に探索木に加えていく
    - 要するに前向き枝刈り
- All Moves As First (AMAF)
  - 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
    - Rapid Action Value Estimation (RAVE) とも呼ぶ
    - 手順を無視して近似する
    - 勝ったプレイアウトで打たれた手は全部良い手
- UCTのパラメータの調整
  - 定数の部分を増減させると、性格が変わる
- 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
  - プレイアウトの回数は数分の1になった
  - しかし全体としての棋力は大幅に向上



初期のCrazyStone
(秒間4万プレイアウト程度)

強化版CrazyStone
(秒間1万プレイアウト程度)

強さのためには
プレイアウトの強化が大事

- 必要な性質は？
  - 完全に決定的なプレイアウトは意味がない
  - 完全にランダムなプレイアウトを使うと弱い
- それらしいプレイアウトを使えば、回数が少なくてもそれなりに強い
  - たとえば fuego（囲碁プログラム）は100～1000回程度のプレイアウトでもそれなりに強い
- それらしいけど、ランダムなプレイアウトが必要
  - あと、速さもそれなりに必要
  - 囲碁だと、秒間1万回くらい実行している

プレイアウトに必要な性質は？

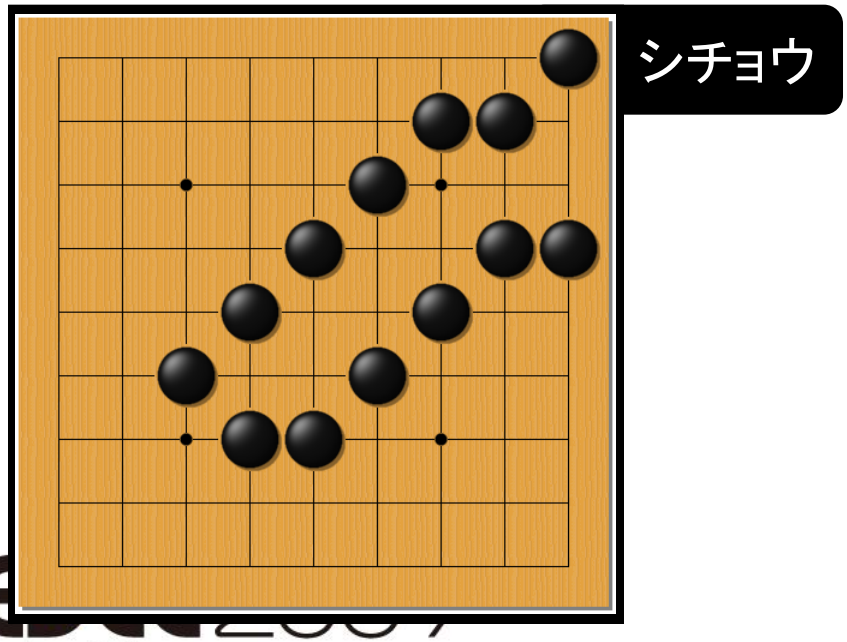
- 理論的にはまだよく分かっていない
  - ICML2009 に新しい論文あり (International Conference on Machine Learning)
    - 機械学習のトップカンファレンス
    - “Monte Carlo Simulation Balancing” [Silver and Tesauro 2009]
- 必ずしも、プレイアウト単独で強い必要はない
  - 実際に、強いプログラムの部品をプレイアウトに使ったが、手で作ったパターンの方が強かった
    - [Gelly and Silver 2007, 2008]
  - 棋譜からの学習と手生成のパターンの両方が効果がある
    - やってみたら強かった、という側面が強い
- MCTSの弱点をカバーできることが重要
  - ありがちな一本道を高い確率で通るのが良い
  - 強さを競わないなら適当でもそれなりに良い
    - 「世界 信長の野望 AI選手権」があるならがんばらないと駄目

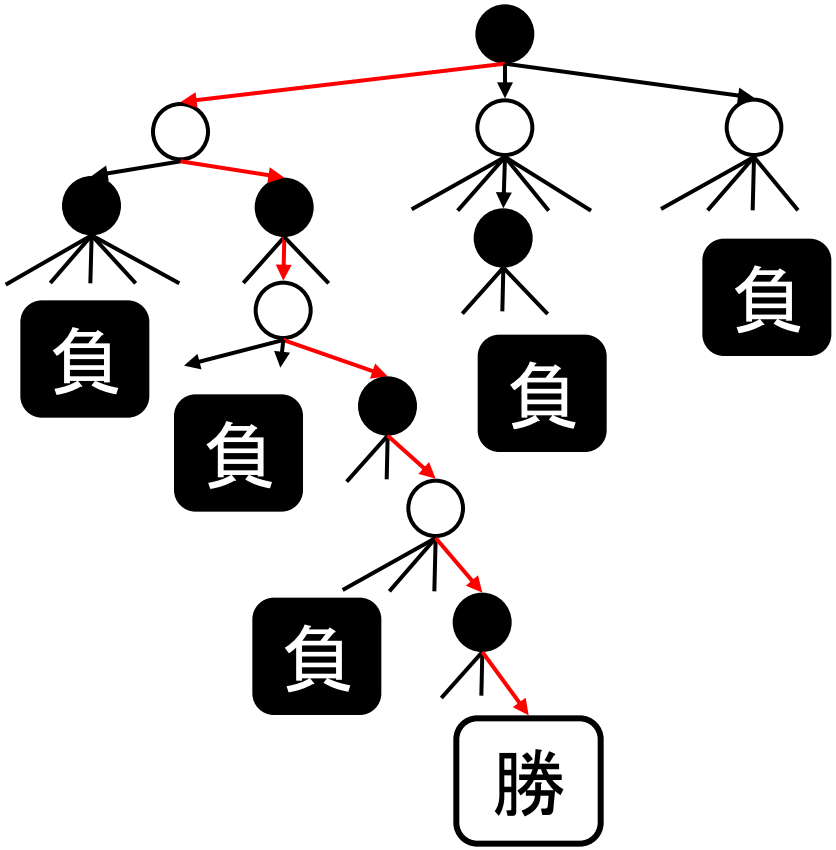
MCTSはなぜ囲碁に有効なのか？

- 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
    - しかし将棋では初段レベルの物が開発された
  - 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、誰もやってない（たぶん、もう十分強いから）
- 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

- 確率的探索だから、勝率の高い手を調べる
- 勝てる手順が一本だけあって、他は全部負け、という場合を苦手とする





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
- 現在の対処法
  - プレイアウト中には、ありがちな一本道はたどるようにする
    - 囲碁のシチョウ（アタリを逃げるようにして回避）
    - 良くあるナカデ、セキ（CrazyStoneはパターンで回避)
  - ありがちでない一本道・・・は、まだ弱い
    - 囲碁の死活、攻め合いはまだ間違える
      - 他の探索アルゴリズムとの組合せなどが研究されている

コンピュータ囲碁の現状

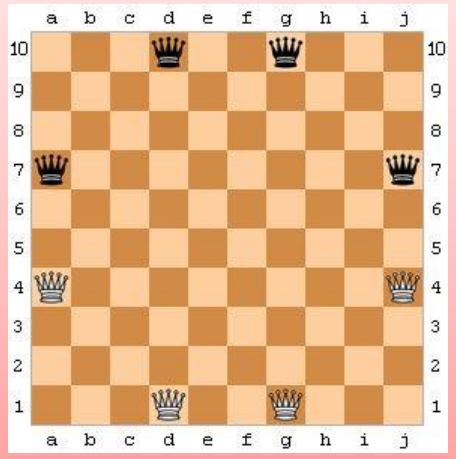
- モンテカルロ木探索の利点
  - 単純に強い
  - プログラミングの労力が少ない
    - 探索部分とプレイアウトの実装だけ
    - プレイアウトの強化には機械学習も有効
- 多くの研究者が参入
  - 機械学習のプロなど
- 並列化の研究も行われている
  - 1000コア以上のクラスターを使ったプロとの対戦も実現
- 進歩が非常に速いので、来年のことも分からない

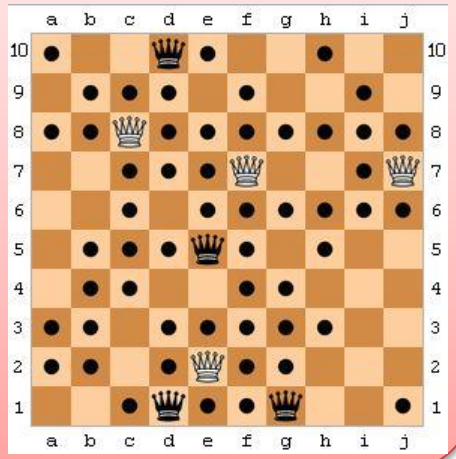
alpha-beta探索にMCTSが追いついた例

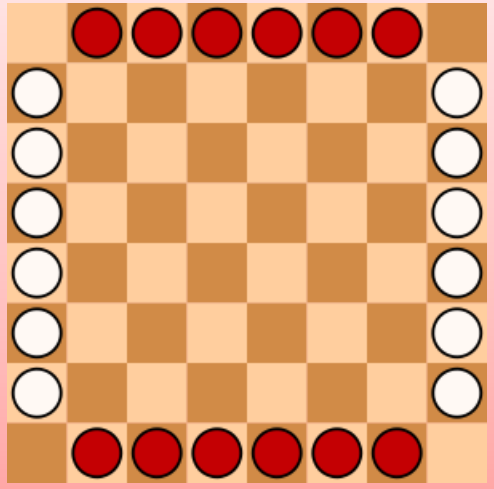
アマゾン（非常にググりにくい）

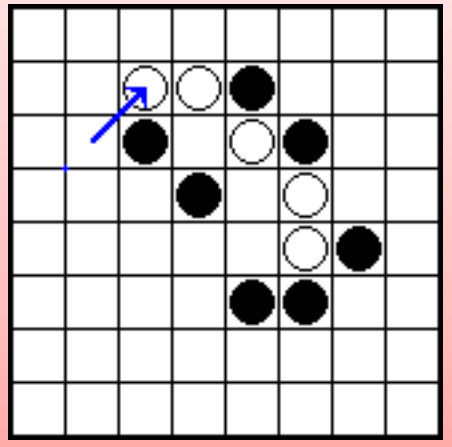
乱数を使うと自然な終局になりにくいゲームだが、プレイアウトを打ち切って評価関数を呼ぶ手法により強くなった

[Lorentz2008] [Kloetzer,Iida,Bouzy2007]









Lines of Action

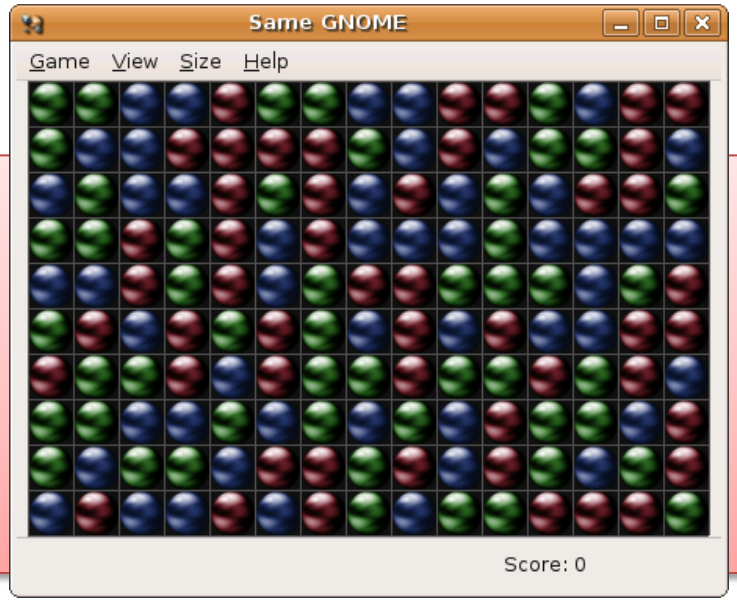
モンテカルロ木探索とαβ探索ベースのプログラムが互角くらいという研究

[Winands,Björnsson,Saito2008]

将棋 (alpha-betaには劣る)

プレイアウトで自然に終局しにくいため、MCTSに不向きなゲームと思われていたが、パターンを学習して初段くらいまで強くなった。

[佐藤,高橋.2008.]



さめがめ

問題集を解かせて、スコアを競う。
モンテカルロ木探索ベースのプログラムが記録を更新

[Schadd,Winands,Herik,Cahslot,Uiterwijk2008]

多人数ゲームでも有効

カタンの開拓者たち

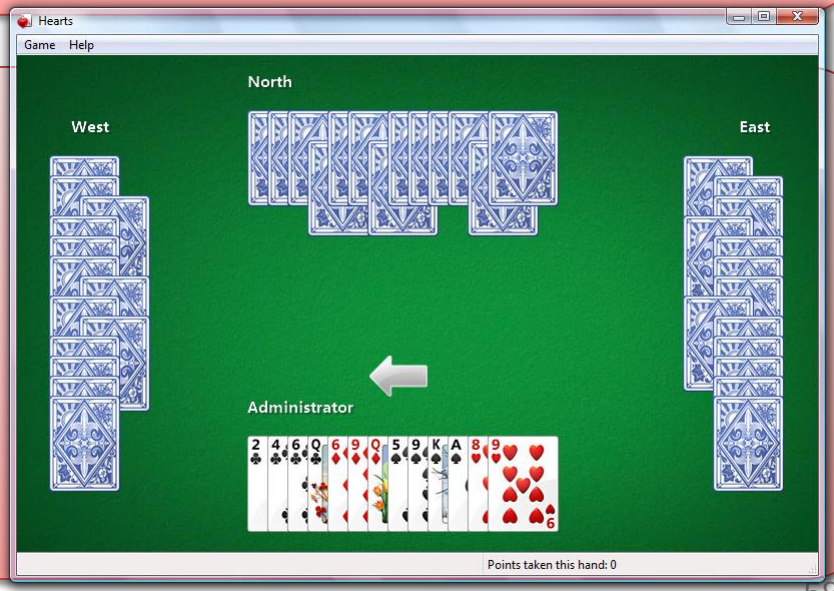
ドイツ製の有名なカードゲーム。MCTSにより強いプログラムが作成された。関係ないがグーグルにはカタン部があるそうである。

[Szita, Chaslot and Spronck 2008]

ハーツ（Windowsに付いてくる）

モンテカルロ木探索を用いたプログラムが、既存のプログラム以上の強さを示す研究あり

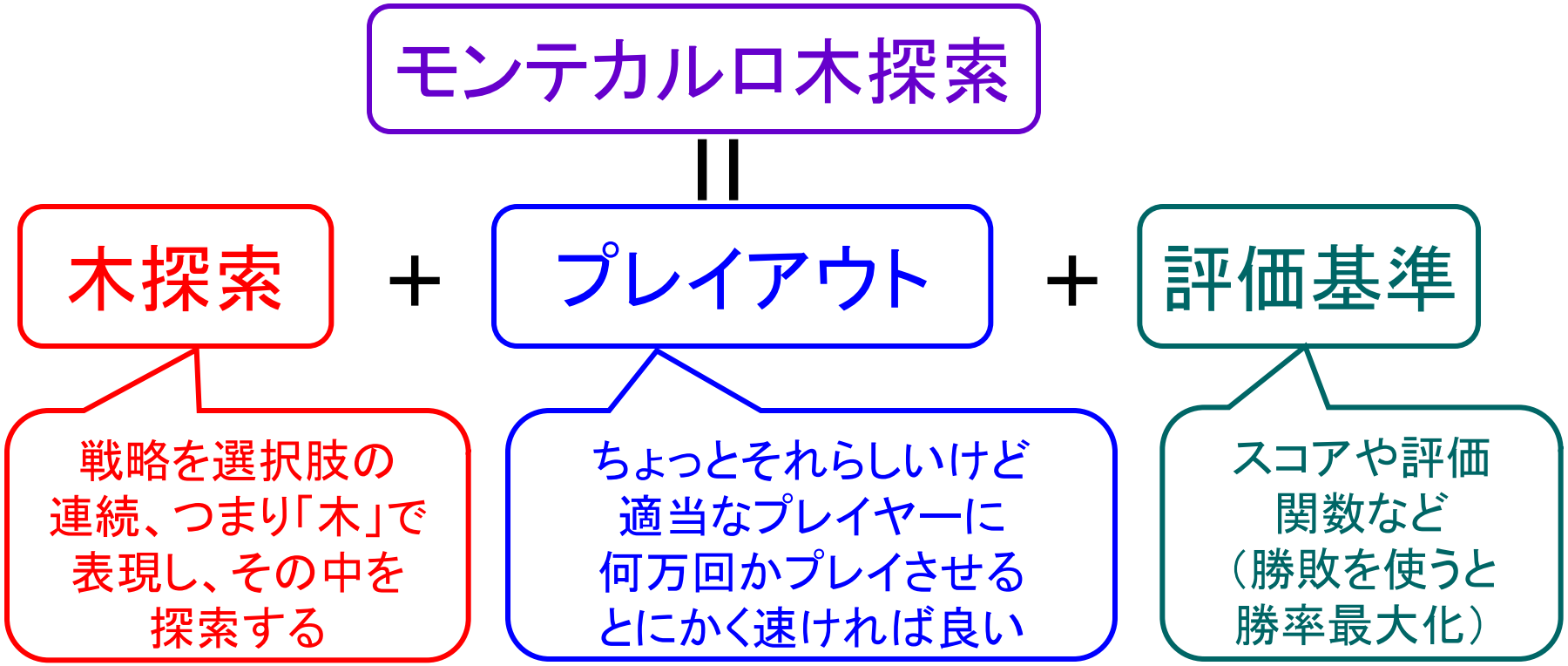
[Sturtevant2008]



汎用性が高い (囲碁以外でも高性能)

- 一人用ゲーム(パズル)
  - SameGame(さめがめ)
- 二人用ゲーム
  - Amazons、Lines of Action、Hex、(将棋)
- 多人数ゲーム
  - ハーツ(Windowsに付いてくるヤツです)、カタンの開拓者たち、Magic the Gathering
- General Game Player Competition(汎用ゲームプレイヤー大会)
  - 大会の場で架空のゲームのルールが提示される
  - その場でプログラムがルールを分析し、直後に対戦する
  - 一人ゲーム、二人ゲーム、多人数ゲームなどごちゃまぜ
  - 総合点が高かったら優勝
  - CADIA Player (UCTをベース)が2年連続で優勝[Finnsson,Björnsson2008]
- ゲーム以外
  - 最適化、プランニング、バイオメトリクス

MCTSを実装するには・・・



- 20級も三百万人集まれば有段者の知恵
  - ただし、うまく集めれば
  - ただし、頭のいい20級ならば（ランダム性必須）
  - ただし、良い評価基準があれば

「・・・のAIを作れ」と言われたら？ MCTSの実装、応用について想像

- 今までのアプローチで普通に作る
  - そのノウハウをプレイアウトと枝刈りに使ってMCTSも試す
  - うまくいけばラッキー
- 木探索は最初の一回の実装は大変
  - しかしノウハウは各ゲームで共通
- プレイアウトは各ゲームの知識を利用する
  - 既存のノウハウがほぼ使える
  - しかしちょっと難しい
    - 決定的なものは駄目だけど、完全なランダムも駄目
- 評価基準
  - 評価関数をそのまま使っても良い

木探索部分の実装

- UCTとかUCBとかは、理論はともかく結果の式だけ使っても大丈夫
  - たぶん適当な近似でも十分良い
- データ構造はツリーでOK (DAGでなくていい)
  - まじめに探索をすると、合流を考えるからツリーでは無くなる
    - オセロはDAGだし、将棋や囲碁はサイクルもある
  - 合流を無視してツリーにしてもほとんど問題無いことが囲碁では知られている
  - ハッシュテーブルなどは必要ないのでその分簡単
- ヒューリスティックな枝刈りが有効
  - ここは既存のアルゴリズムと変わらない
- 例えば、A*と比べてそれほど実装が難しいとは思わない
  - 公開されているサンプルコードもある
    - Fuego： オープンソース最強の囲碁プログラム、ちょっと難しい
    - 彩 : MCTSのコア部分だけ (cf. YSSと彩のページ)

プレイアウトの実装

- プレイアウトは、決定的でなく、完全なランダムでもなく、しかも速い必要がある
  - でも、実は適当でもそれなりにうまくいく
  - 強さを競うのならば、やっぱり大変
    - 「世界いたストAI選手権」とかがあれば大変
- プレイアウトで自然に終局しないゲームも何とかなる
  - アマゾンではプレイアウトを途中で打ち切って評価関数と組み合わせる手法が提案されている
    - αβ探索+評価関数よりも強かった
    - 評価関数の不具合をMCTSがカバーしてくれる
  - 将棋でもそれなりの強さのものはできている
    - 評価関数や詰探索と組み合わせた例 [橋本,橋本,長嶋2006]
    - うまくプレイアウトを作ったらちゃんと終局した [佐藤,高橋2008]

MCTSによるゲームAIの実例 (論文)

- カタンの開拓者たち [Szita, Chaslot and Spronck 2008]
  - (Chaslot は MoGoプロジェクトにも参加)
  - 木探索部分はたぶん普通
  - プレイアウトは開拓者を置く確率を高めるなどの工夫あり
  - 他のプログラム(JSettlers)と対戦
    - (1手当たり)1000プレイアウトで互角、10000プレイアウトで大きく勝ち越し
- Magic the Gathering [Ward and Cowling 2009]
  - Magic the Gathering の一部についてUCBを適用
  - 既存のプログラムをそのままプレイアウトに利用
    - 少ない回数のプレイアウトでも元より有意に強くなることを示した
  - この論文では一段読みまで

- 今後はもっと実例が増えると思われる

MCTSの長所と短所

- 強さが思考時間次第
  - PS3やCore2Quadに向いているが、DSは無理ではないか
  - いつ止めてもその時点で最善の結果を返す(anytime性がある)
- (ある意味で)強い、しかし制御が難しい
  - MCTSに限らず、探索を使ったAIに共通の性質
- 細く長い一本道の手順は弱い
  - しかしある意味で、自然な弱さが生まれる (うっかりミスをする)
- 勝率最大化だと
  - 負かされると非常につまらない
    - 強さを犠牲にすれば、つまらなさを解消可能と思われる
  - 勝つときは面白い
    - じり貧を嫌うので、負ける前に勝負に出てくるプログラムになる
- 汎用性が高い
  - 戦略が木構造で現せるなら使える

Civilization 4 でMCTSという妄想

ご存じとは思いますが

- 中毒性の高いゲーム
- ターン性戦略シミュレーションの最高傑作
  - 僕の主観です
- 文明を一番繁栄させると勝利
- 注：僕は Civilization 4は嫌いです
  - その証拠に、もう7回ほどアンインストールしてます

Civ4で妄想
長所：ある程度強くなる

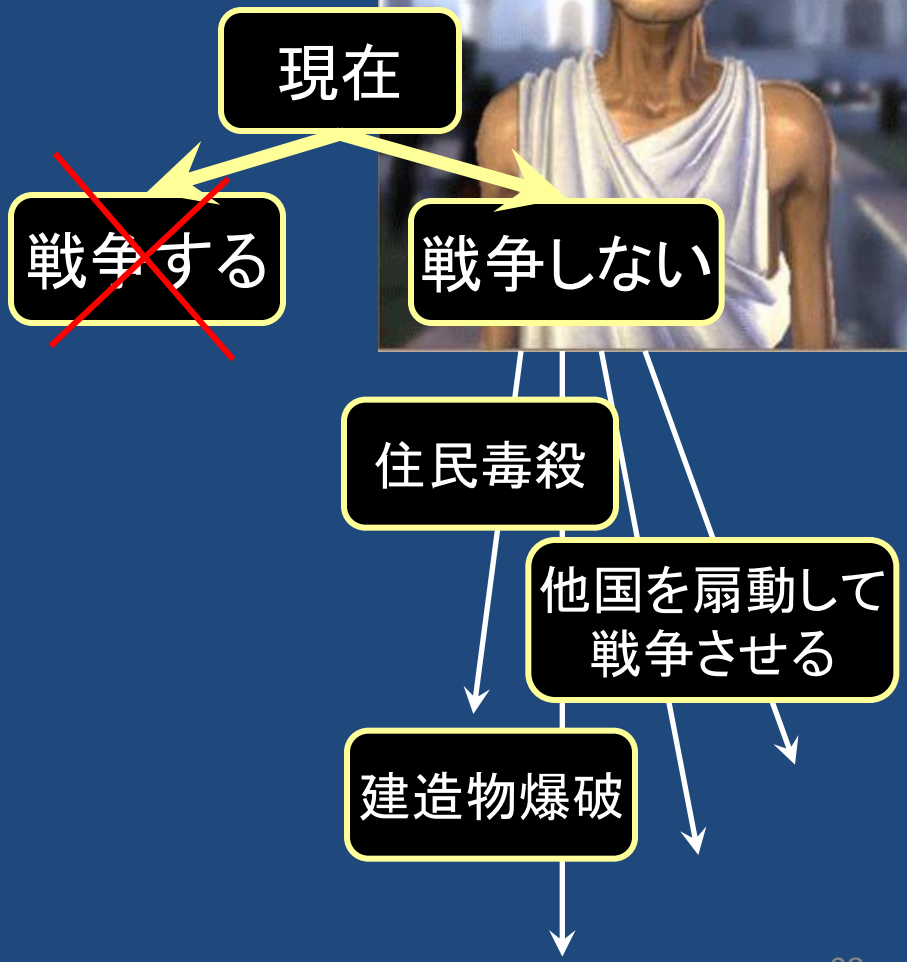
- このように戦略を木で表現することを考える
  - プランニング＋MCTS

- モンテカルロ木探索によるAIの場合、
  - 戦争した方が勝率が高い
  - （しないとほぼ負け）
  - だから戦争する
  - 最後の勝負を仕掛けてくる
  - 対人戦の練習によさそう
  - でも面白いかどうかは・・・？
  - 上級者向け?

Civ4で妄想
短所：思い通りに動かない

- 戦争を禁じ手にして平和主義者にしたつもりのAI
  - 戦争しない範囲でできるだけのことをする
  - 平和主義者のはずが悪の組織の黒幕に！

- つまり、頭が良いので言うことを聞かせにくい
  - 狙い通りの挙動をさせるのは難しいかも？



最後に

- MCTSは、
  - 理論的には奥が深いが実装は簡単
  - 手強いAIを作れる
  - 既存のアプローチとは違った個性を持ったAIを作れる
  - リアルタイムゲームに全く使えないということは無いように思います
  - しかし、戦略を木構造で表せない物には使えない

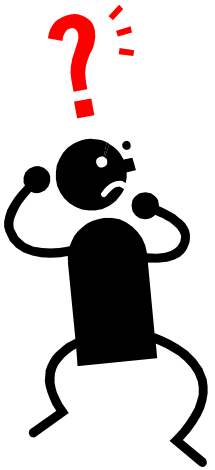
- 今までとは個性の違うAIが、おもしろいゲームにつながったらうれしく思います
  - ゲームAI開発の手助けになれば嬉しいです
  - 個人的には、Civilization 4 のAIが強くなったら嬉しいです

- [Auer,Cesa-Bianchi,Fischer2002] *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.
- [Coulom2006] *R. Coulom*, "Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search Export", 5th International Conference on Computer and Games (CG2006), pp. 72-83, 2006.
- [Coulom2007] *R. Coulom*, Computing Elo Ratings of Move Patterns in the Game of Go, *Computer Games Workshop*, 2007.
- [Gelly,Wang,Munos,Teytaud2006] *S. Gelly, Y. Wang, R. Munos and O. Teytaud*, Modification of UCT with patterns in Monte-Carlo Go, Technical Report No.6062, INRIA, 2006.
- [Kocsis,Szepesvári2006] *L. Kocsis and C. Szepesvári*, Bandit Based Monte-Carlo Planning, *LNCS vol.4212 (ECML 2006)*, pp. 282-293, 2006.
- [Lai,Robbins1985] *T. L. Lai and H. Robbins*, Asymptotically efficient adaptive allocation rules, *Advances in Applied Mathematics*, vol. 6, pp. 4-22, 1985.
- [Yoshimoto,Yoshizoe,Kaneko,Kishimoto,Taura2006] *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.
- [小泉,石井,美添,三好,菅原,稲葉,平木2009] *小泉賢一, 石井康雄, 美添一樹, 三好健文, 菅原豊, 稲葉真理, 平木敬*, "FPGA基板を用いたモンテカルロ碁の高速化", 信学技報, vol. 109, no. 168, CPSY2009-19, pp. 55-60, 2009年.

- [Sturtevant2008] *N. R. Sturtevant,* "An Analysis of UCT in Multi-player Games", Computers and Games (CG2008), pp.37-49, 2008.
- [Schadd,Winands,Herik,Cahslot,Uiterwijk2008] *M. P. D. Schadd, M. H. M. Winands, H. Jaap van den Herik, G. M. J. -B. Chaslot and J. W. H. M. Uiterwijk*, "Single-Player Monte-Carlo Tree Search", Computers and Games (CG2008), pp.1-12, 2008.
- [Lorentz2008] *R. J. Lorentz*, "Amazons Discover Monte-Carlo", Computers and Games (CG2008), pp.13-24, 2008.
- [Kloetzer,Iida,Bouzy2007] *J. Kloetzer, H. Iida, and B. Bouzy*, "The Monte-Carlo Approach in Amazons", In Proc. Computer Games Workshop, 2008.
- [Winands,Björnsson,Saito2008] *M. H. M. Winands, Y. Björnsson and J.-T. Saito*, "Monte-Carlo Tree Search Solver", Computers and Games (CG2008), pp.25-26, 2008.
- [橋本, 橋本, 長嶋2006] *橋本隼一, 橋本剛, 長嶋淳*, "コンピュータ将棋におけるモンテカルロ法の可能性", In Proc. 11th Game Programming Workshop, 2006
- [佐藤, 高橋2008] *佐藤佳州, 高橋大介*, "モンテカルロ木探索によるコンピュータ将棋", In Proc. 13th Game Programming Workshop, 2008.
- [Finnsson,Björnsson2008] *H. Finnsson and Y. Björnsson*, "Simulation-based Approach to General Game Playing", In 23rd AAAI Conference on Artificial Intelligence, pp. 259–264, 2008.

- [Gelly and Silver 2007] *S. Gelly and D. Silver*. "Combining Online and Offline Knowledge in UCT", ICML 2007, 2007
- [Szita, Chaslot, Spronck 2009] *S. Szita, G. Chaslot, and P. Spronck*. "Monte-Carlo Tree Search in Settlers of Catan." 12th Advances in Computer Games (ACG2009), 2009.
- [Ward Cowling 2009] *C. D. Ward and P. I. Cowling*. "Monte Carlo Search Applied to Card Selection in Magic: The Gathering", IEEE Conference on Computational Intelligence in Games (CIG 2009), 2009.
- [Silver and Tesauro 2009] *D. Silver and G. Tesauro*. "Monte-Carlo Simulation Balancing", ICML 2009, 2009.
- [Nakhost an Mueller 2009] *H. Nakhost and M. Müller*. "Monte-Carlo exploration for deterministic planning", IJCAI 2009, 2009.
- [Tesauro and Galperin 1996] *G. Tesauro and G. Galperin*. "On-line policy improvement using Monte-Carlo search", Advances in Neural Information Processing 9 (NIPS), pp. 1068-1074, 1996.
- [Sheppard2002] *B. Sheppard*. "World-championship-caliber Scrabble" Artificial Intelligence vol. 134, pp.241-275, 2002.
- [Billings, Castillo, Schaeffer, Szafron 1999] *D. Billings, L. P. Castillo, J. Schaeffer and D. Szafron*. "Using Probabilistic Knowledge and Simulation to Play Poker", AAAI-99, pp. 697-703, 1999.

MCTSの残る課題



- Simulation (Playout) の性質
- 合流への対処
- 並列化
- 乱数の初期値に鋭敏

まだ分からないことだらけ

強さのためには
プレイアウトの強化が大事

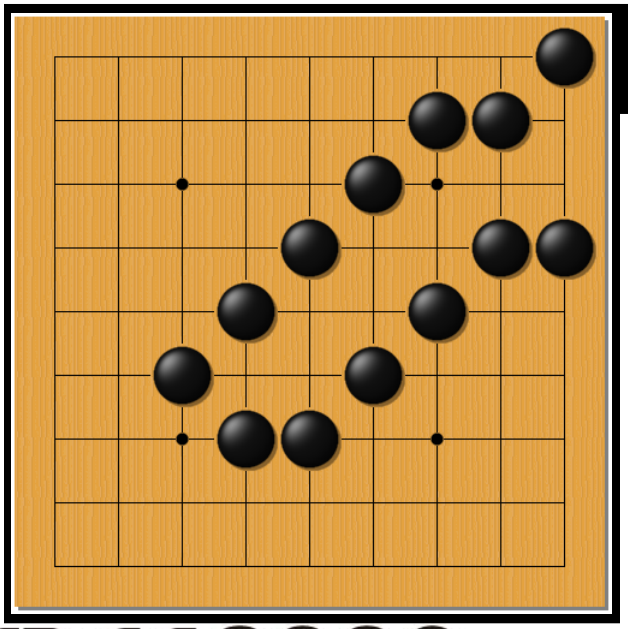
- 必要な性質は？
  - 完全に決定的なプレイアウトは意味がない
  - 完全にランダムなプレイアウトを使うと弱い
- それらしいプレイアウトを使えば、回数が少なくてもそれなりに強い
  - たとえば強い囲碁プログラムは 100～1000回程度のプレイアウトでもそれなりに強い
- それらしいけど、ランダムなプレイアウトが必要
  - あと、速さもそれなりに必要
  - 囲碁だと、秒間1万回くらい実行している
- 「それらしい」って何？

プレイアウトに必要な性質は？

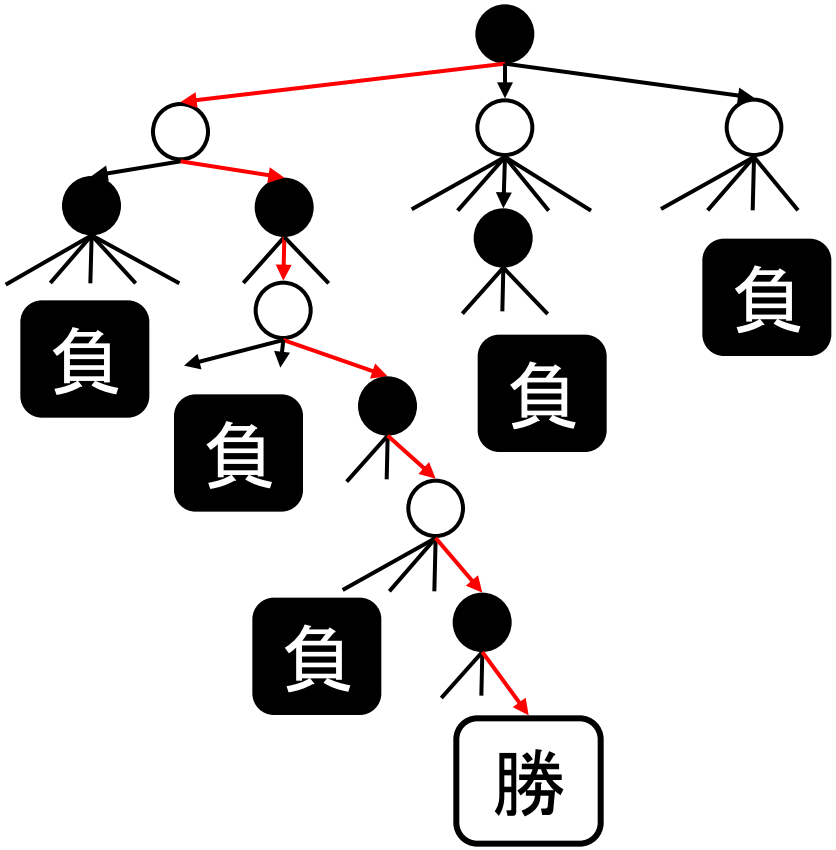
- 理論的にはまだよく分かっていない
  - ICML2009 に新しい論文あり
    - [Silver and Tesauro 2009] “Monte Carlo Simulation Balancing”
- 必ずしも、プレイアウト単独で強い必要はない
  - 実際に、強いプログラムの部品をプレイアウトに使ったが、手で作ったパターンの方が強かった (囲碁の例)
    - RLGo と MoGo の組合せの研究 [Gelly and Silver 2007, 2008]
  - 棋譜からの学習と手生成のパターン、両方が効果がある
    - やってみたら強かった、という側面が強い
- MCTSの弱点をカバーできることが重要
  - ありがちな一本道を高い確率で通るのが良い？

モンテカルロ木探索の弱点

- 確率的探索だから、勝率の高い手を調べる
- 勝てる手順が一本だけあって、他は全部負け、という場合を苦手とする



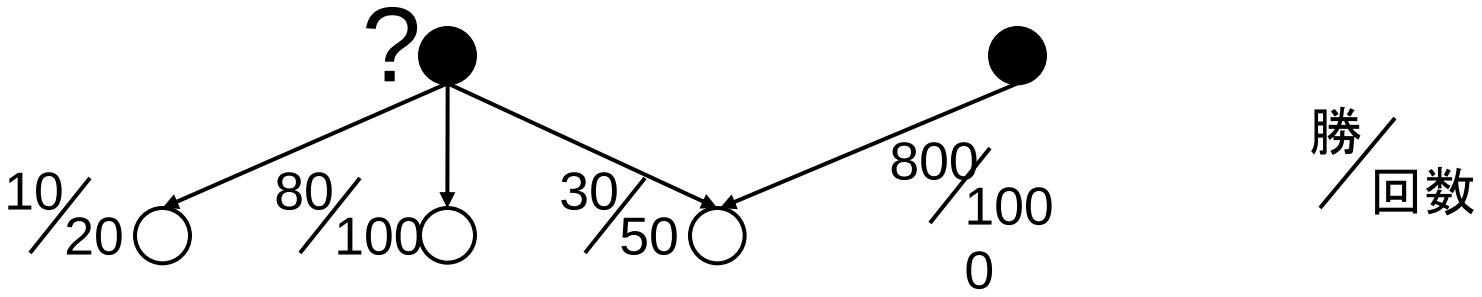
シチョウ



探索を playout で補う必要

- 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
- 現在の囲碁での対処法
  - プレイアウト中に、ありがちな一本道をたどるようにする
    - 囲碁のシチョウ（アタリを逃げるようにして回避）
    - 良くあるナカデ、セキ（CrazyStoneはパターンで回避)
  - ありがちでない一本道・・・は、まだ弱い
    - 囲碁の死活、攻め合いはまだ間違える
      - 他の探索アルゴリズムとの組合せなどが研究されている
- 弱点を補うことが重要
  - 必ずしも playout 単独で強い必要は無い
  - 必ずしも playout と棋譜との一致率が高い必要もない(?)

合流時のUCB値計算



- 合流がある場合の計算は自明でない
  - 理論を示した論文あり [CBK2008]
  - まだ決定版かどうか分からない
- 難しいので(?)トップレベルの囲碁プログラムでも合流を無視する派が多い
  - hash table 派 (CrazyStone等) vs tree 派 (MoGo等)
  - treeは無駄なはずだが、実際は十分強い
  - treeなら実装が簡単
    - 特に並列化も簡単
- つまり、よほど強さを目指すので無い限り、ツリーで十分
  - 合流は無視してもたぶん問題無い

並列化について

- どのような並列化手法が良いのかまだ試行錯誤中
  - ルート並列化という非常に単純な手法がかなり有効
    - 乱数のシードを変えて並列に探索を行い、最後に合計する
- 特にクラスタではルート並列化がそれなりに良い
  - MoGo, Fuego など
  - 理由は不明だが、乱数の初期値に敏感な性質のせいか？
- 共有メモリマシンでは virtual loss という手法がある
  - 並列実行中の playout は全部負けると仮定する
  - 多数のコインを同時に投入するケースのmulti armed banditとの関連
- FPGA による試みもある
  - [小泉、石井、美添、三好、菅原、稲葉、平木2009]
- GPGPUはまだ論文無し
  - 今自分でやってます
- しかしAIが複数存在するなら、それぞれに1 CPU 割り当てれば十分

初期の乱数に鋭敏

- 初期の乱数への敏感さ
  - 初期の乱数に長期間影響されるらしい
    - 対称性を考慮すると同じ価値のはずの手が、大きく異なる評価をされることが多い
  - root 並列化が有効な理由の一つではないか
  - First Move Urgency が有効であることと関係するか
    - 勝てる手がある場合はその1手に集中して探索する手法
- これはそもそも解決する必要がある問題ではなく、そういう性質のあるアルゴリズムだということ
  - 最初に運良く高い評価をされた手が、長いこと優遇される

探索時間、探索木のサイズ、強さ

- 考慮時間と強さ
  - 囲碁では、4倍の考慮時間で一段ちょっと強くなる
    - 16倍なら二段ちょっと、64倍で三段ちょっと
  - 逆に言えば、64倍速くしても、三段強しか変わらない
    - 競争相手がいなければ、そんなにがんばらなくていい
- 探索木の目安
  - 囲碁9路盤では、初手の分岐数が81でプレイアウトの手数は100程度
    - この探索木はかなり大きい
    - ランダムに近いプレイアウトで10級程度
  - 囲碁19路盤は、初手の分岐数は361で、プレイアウトは400手程度
    - ランダムに近いプレイアウトだと非常に弱い
    - このサイズだと、プレイアウトをかなり工夫しないと駄目
  - ゲームの場合も、探索木のサイズに注意
    - 分岐が多くて手数が長ければ、プレイアウトを工夫する必要がある
      - 途中で打ち切って評価関数を呼ぶ
      - あるいは、本当にがんばる(囲碁の場合のように)
    - 時間の制約の許す範囲で、木が大きい方が良い(強い) (ここは職人技)
      - ゲームによって許される考慮時間は違うと思われるので、そこは調整

ありがとうございました