

計算機システム (番外編)

教授 石川 裕

助教 美添 一樹

連絡先: ishikawa@is.s.u-tokyo.ac.jp

yoshizoe@is.s.u-tokyo.ac.jp

本郷 理学部7号館505、507

計算機システム 番外編

2010/11/15

1

内容

- はじめに
 - 探索アルゴリズムとゲームAIという分野について
- 探索アルゴリズムについて
 - A*探索について
 - ゲーム分野:ルービックキューブの20手証明など
 - 一人ゲームだけでなく様々な問題に役立つ
 - alpha-beta 探索と過去のゲームAIの強さ
 - 将棋: 清水女流王将 対 あから2010
- 以下ずっと、モンテカルロ木探索の説明
 - モンテカルロ木探索の概要
 - 自分の今の研究(並列化)について

2010/11/15

計算機システム 番外編

2

はじめに ゲームAIという分野について

2010/11/15

計算機システム 番外編

3

人工知能 (AI) という分野

- 人間にはできるが、コンピュータにはできないことを研究する (自分の定義)
 - 犬と猫を見分ける、英語を日本語に翻訳する、面白い冗談を言う、空気を読む、...
- コンピュータができるようになると、AIを卒業する
 - 過去にAIの研究と見なされていたことがある物
 - 漢字変換、コンパイラ、...
- 自分の興味は「ゲームAI」「探索」
 - 技術要素は「探索アルゴリズム」 目的は「ゲームAI」
 - 「チェスをプレイさせる」欧米ではチェスは知性の象徴だった
 - 1997年にIBMの DeepBlue が当時の世界チャンピオン Kasparov を破る
 - もしかするとそろそろAIを卒業する？

2010/11/15

計算機システム 番外編

4

ゲームAI研究

- 人工知能の分野では、まじめな研究対象
 - 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 (たまたに将棋も)
 - 技術の進歩を分かりやすく示すことができる
 - 尺度:強さ=進歩, 目的:熟練した人間に勝つ
- 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
 - **モンテカルロ木探索**

2010/11/15

計算機システム 番外編

5

(AIにおける)探索アルゴリズム

- 最短経路問題
 - A*探索
 - 例, ルービックキューブ (最長でも20手で解ける証明)
 - [Rokicki, Kociemba, Davidson, and Dethridge, 2010]
 - <http://cube20.org/>
 - その他, カーナビ, DNAの解析の一部など, 様々な応用
- 二人ゼロ和完全情報ゲーム
 - min-max木探索 (alpha-beta探索)
 - 囲碁以外の主な2人ゼロ和完全情報ゲーム
 - 例, チェス, オセロ, チェッカー, 将棋, ...
- 確率的ゲーム木
 - ここから先はまだ本が無い世界
 - モンテカルロ木探索
 - 囲碁を含む多くのゲーム
 - その他いくつかの問題

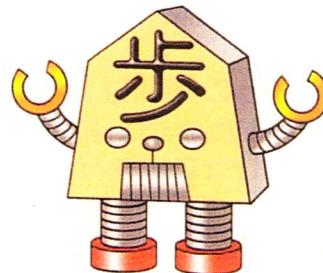
2010/11/15

計算機システム 番外編

6

清水市代女流王将vs.あから2010

- 女流プロ棋士 対 コンピュータ
 - 日本将棋連盟, 情報処理学会共催の公式イベント
 - 情報処理学会-コンピュータ将棋プロジェクト
 - <http://www.ipsj.or.jp/50anv/shogi/index2.html>
- 合議制を用いた4つのプログラム
 - 激指, GPS将棋, Bonanza, YSS
- コンピュータの勝利に終わる



2010/11/15

計算機システム 番外編

7

あから2010の体制

- 強さが認められた4つのプログラム
 - 世界コンピュータ将棋選手権などの実績
 - 激指, GPS将棋, Bonanza, YSS
- 「合議制」による並列化
 - 4つのプログラムの多数決による着手選択
 - それぞれのプログラムが, クラスタ (分散メモリ型並列計算機) およびバックアップマシンを利用
 - 実はバックアップマシンが本体

多数決の票の重み

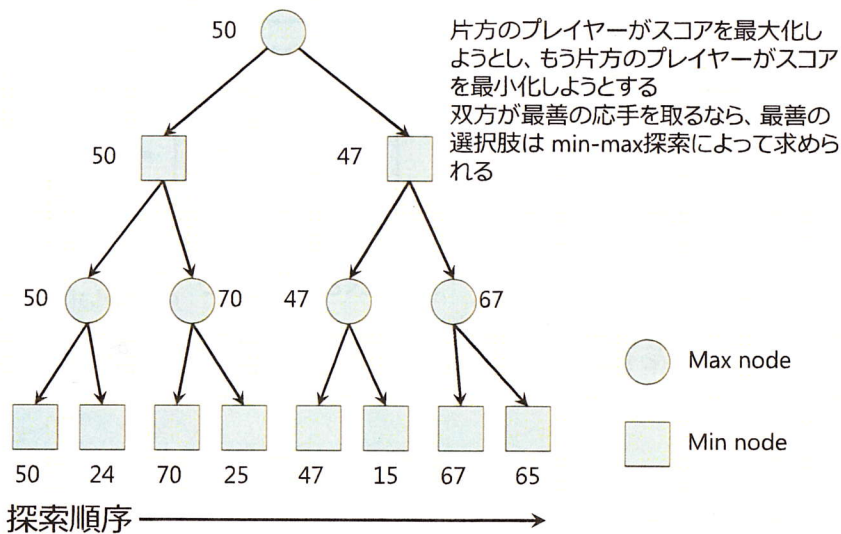
	激指	GPS将棋	Bonanza	YSS
クラスタ	0.1	1.0	0.1	0.1
バックアップマシン	2.9	1.0	1.9	1.9

2010/11/15

計算機システム 番外編

8

min-max探索 (二人零和完全情報ゲーム)

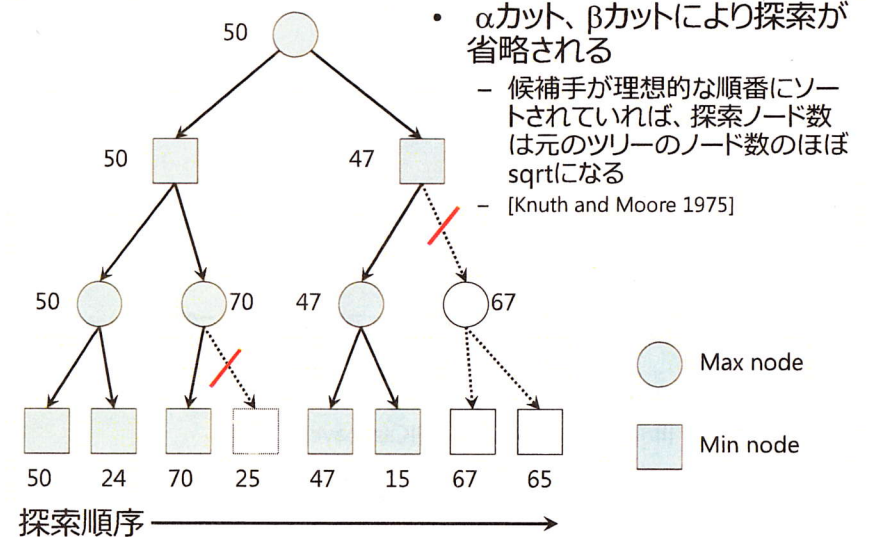


2010/11/15

計算機システム 番外編

9

$\alpha\beta$ 探索 (min-max探索 + $\alpha\beta$ 枝刈り)



2010/11/15

計算機システム 番外編

10

alpha-beta探索の有効性

- ほとんどのメジャーな二人ゼロ和完全確定情報ゲームでは alpha-beta探索が有効
 - チェッカー、オセロ、チェス、将棋、...
 - 非常に効率的
 - (並列化が難しい)

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	プロに挑戦中
囲碁	アマ三~五段?

- 囲碁には有効でない!
 - 主な二人ゲームで alpha-beta探索が有効でないのは囲碁だけ

余談

- ポーカー はコンピュータが強い
 - ただし、1対1のテキサスホールデム
 - 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/>
- 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

2010/11/15

計算機システム 番外編

11

2010/11/15

計算機システム 番外編

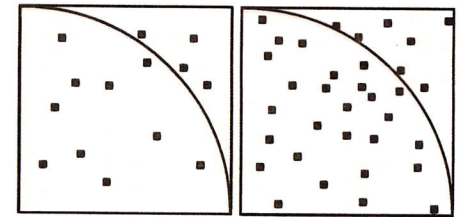
12

背景：モンテカルロとゲーム

- いわゆるモンテカルロシミュレーションでスコアの高い選択肢を選ぶ
 - 不完全情報ゲームに適用するのは自然なアイデア
 - Backgammon
 - 参考文献 [TG1996]
 - Scrabble
 - 参考文献 [Sheppard2002]
 - Poker (テキサスホールデム)
 - 参考文献 [BCSS1999]
 - 完全確定情報ゲームに使う？面白いアイデアですね(他人事)
 - 原始モンテカルロ囲碁
 - [Brugmann 1993][Bouzy][Cazenave]

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

コンピュータ 囲碁の革命



2005年以前の強さ

19路盤で3級
9路盤でも3級

主なゲームの中で、
囲碁だけが際立って
人間が優勢だった

2006年にMCTS
が提案される
[Rémi Coulom 2006]

MCTS
Monte Carlo
Tree Search

2010年現在

19路盤ならアマ四段？
9路盤ならそろそろ
プロに近い

現在のプログラムは、9路盤
ならプロに勝つことも珍しくない

4年間の棋力向上は、
19路盤なら7段程度、
9路盤では10段以上

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
- 汎用性が高い
 - 現在の応用例
 - 二人ゲーム、多人数ゲーム、一人ゲーム(パズル)
 - 最適化、プランニング、バイオメトリクスなど
- MCTSはどういうアルゴリズムか
 - 一応の理論的背景を説明
 - 見た目の性質も面白い
 - 長所と短所を紹介
- 現在は、
 - やってみたいらうまくいった、という研究が多い
 - 理論研究の余地が多く残されている(と思われる)

囲碁の難しさ

2010/11/15

計算機システム 番外編

17

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
- ルールは単純だが奥が深いと言われる

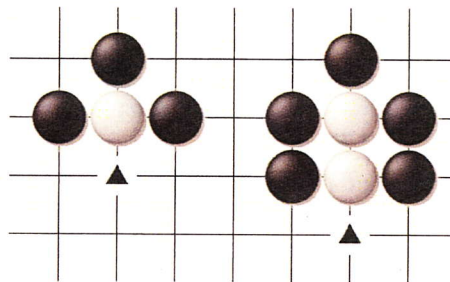
2010/11/15

計算機システム 番外編

18

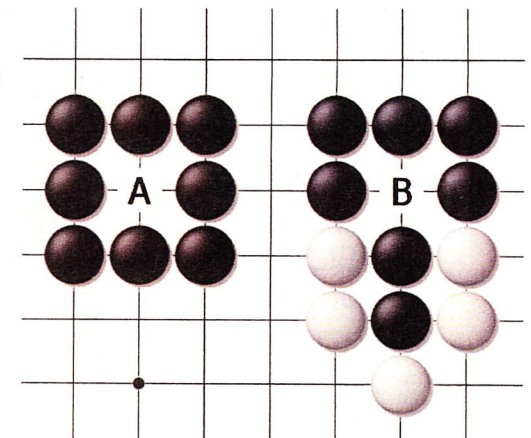
囲碁のルール：囲んだら取れる

- ▲のところ黒が打つと、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連托生になっている
 - 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



2010/11/15

計算機システム 番外編

19

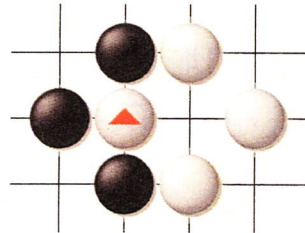
2010/11/15

計算機システム 番外編

20

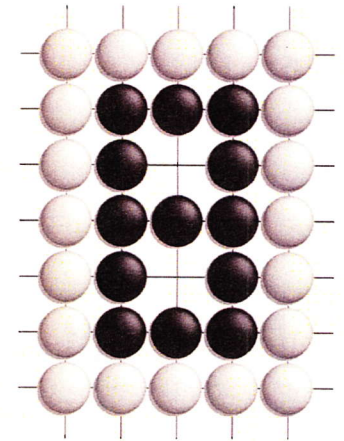
囲碁のルール：同型反復禁止

- 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



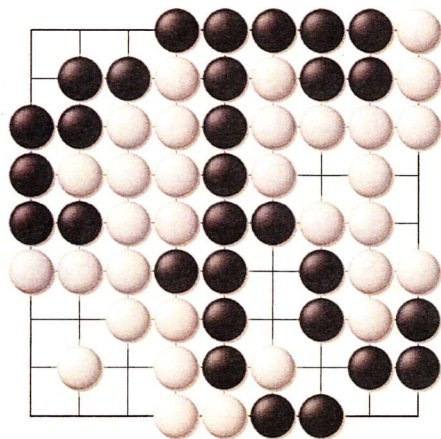
生き、死に、という概念

- 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「生きていう
 - 着手禁止点のことを「眼」という
- 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 (日本ルール)
 - 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)
 - サイエンスに珍しくコンピュータサイエンスの論文
 - 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
 - どちらも(建前(は)アマ初段くらい)
- 他のゲームと比較して、これはおかしい
 - 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？

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}

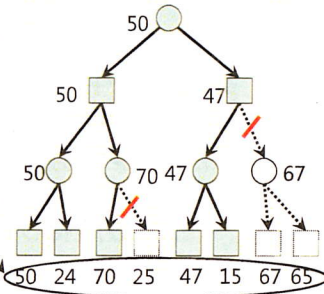
探索空間
(可能な局面数)

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

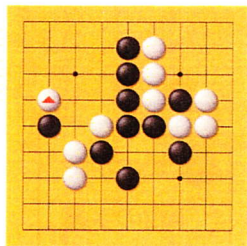


囲碁の良い評価関数は未だに作られていない

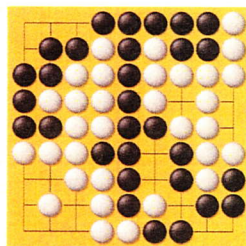
評価関数の例 囲碁以外のゲーム

- 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
 - 多数 (数万~1億) の特徴の重み付けを自動的にを行う
 - 人間の棋譜から自動的に評価関数を作成

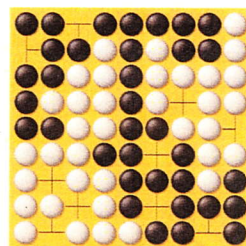
囲碁の評価関数は難しいが...



中盤の評価関数は
非常に難しい



終局後なら
スコア判定は簡単だが、
日本ルールの終局判定は難しい



中国ルールの
終局図なら簡単

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

2010/11/15

計算機システム 番外編

29

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?

2010/11/15

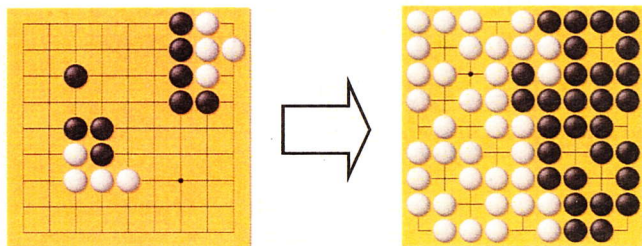
計算機システム 番外編

30

プレイアウトとは

ゲームAI用語

- 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



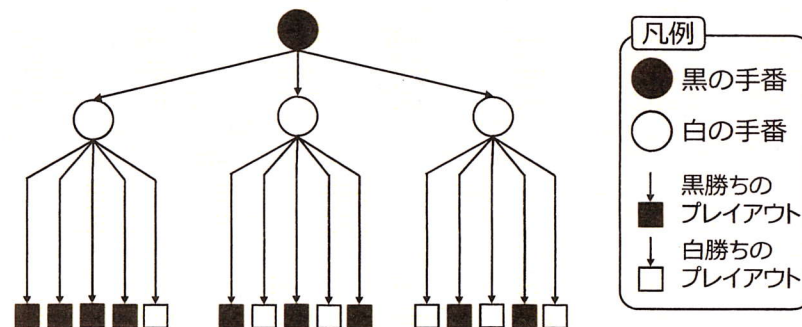
2010/11/15

計算機システム 番外編

31

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



2010/11/15

計算機システム 番外編

32

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
 - 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

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

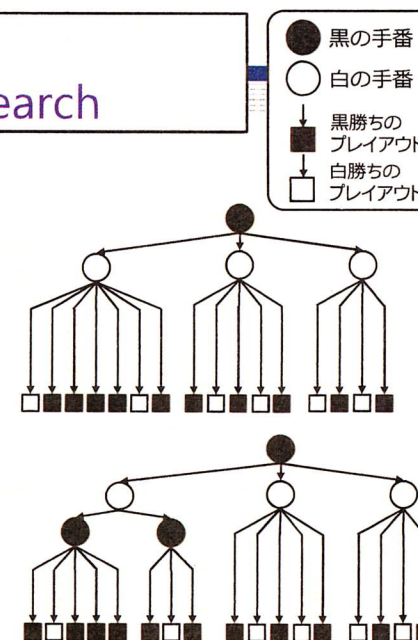
CrazyStoneの登場

- 2006年のComputer Olympiad (@Torino) 囲碁9路盤部門 優勝プログラム [Rémi Coulom 2006]
 - モンテカルロ法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



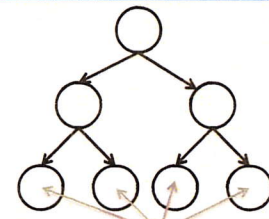
モンテカルロ木探索 Monte Carlo Tree Search

- 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増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



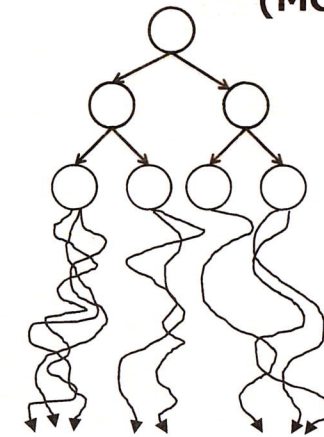
MCTSの理論的背景

通常の探索アルゴリズム



末端で評価関数を呼ぶ

モンテカルロ木探索 (MCTS)



末端で
モンテカルロシミュレーション
(だからモンテカルロ木探索)

コンピュータ囲碁の
棋力を飛躍的に向上させ、
注目を集める
[Coulom2006] [Kocsis & Szepesvari 2006]
特に、UCTアルゴリズムが有名

初期のCrazyStoneの方式

候補手が $0 \dots N$ までとする。

i 番目の候補手は評価値 μ_i を持ち、分散が σ_i^2 であるとする。

また、 $\mu_0 > \mu_1 > \dots > \mu_N$ であるとする。

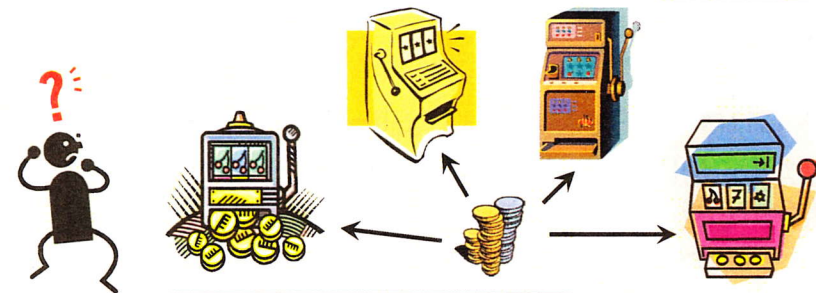
そのとき、各候補手を、以下の値に比例する確率で選択する

$$u_i = \exp\left(-2.4 \frac{\mu_0 - \mu_i}{\sqrt{2(\sigma_0^2 + \sigma_i^2)}}\right) + \epsilon_i$$

囲碁の性質を考慮に
入れた定数

今では歴史的価値しか無いが、当初の
CrazyStoneはこの方式を用いていた。

Multi-Armed Bandit 問題



最善の選択肢を選んだ場合の
報酬との差 (regret) の期待値を
最小化したい

T. L. Lai and H. Robbins, *Asymptotically efficient adaptive allocation rules*,
Advances in Applied Mathematics, vol. 6, pp. 4-22, 1985.

UCB1という戦略

- 各マシンについてUCB1値という値 (Upper Confidence Bound)を計算する

- UCB1値が最大になるマシンにコインを投入

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

\bar{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 : アルゴリズムの性格を決める定数

Upper Confidence Bound

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.

UCT (UCB applied to Trees)

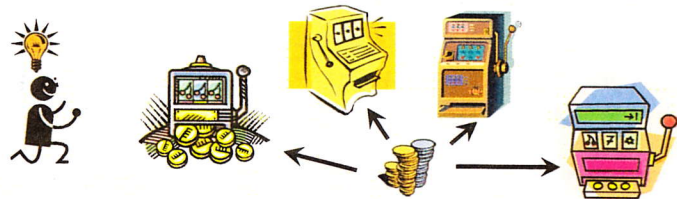
- CrazyStoneの成功を受けて提案された木探索アルゴリズム
 - UCB1を木探索に応用
 - UCB1値の高い候補手を辿って探索を行う
 - 末端の候補手でプレイアウトの回数が閾値を超えると、その手を展開する
 - Simulationの回数nが大きくなると、UCB1値が以下のように、期待値に収束することが証明されている

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

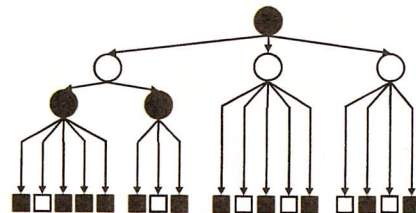
- UCTはCrazyStoneの方法を改良し、さらに理論的な基盤を与えた

L. Kocsis and C. Szepesvári, *Bandit Based Monte-Carlo Planning*, *LNCS vol.4212 (ECML 2006)*, pp. 282-293, 2006.

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]



UCB1-TUNED

$$V_j(s) = \left(\frac{1}{s} \sum_{r=1}^s X_{j,r}^2 \right) - \bar{X}_{j,s}^2 + \sqrt{\frac{2 \ln t}{s}} \quad \text{と定義する}$$

$$\text{UCB1の式を変更} \quad \bar{X}_j + p \sqrt{\frac{\ln n}{n_j} \min\left\{\frac{1}{4}, V_j(n_j)\right\}}$$

調整のためのパラメータ

2項分布の分散の上限

Regretのboundに関する証明はされていないが、性能はUCB1, UCB2 よりも良い

おそらく、現在もっとも一般的に使われている方式。実際に、囲碁プログラムにおいても、UCB1よりも強くなる。

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.
S. Gelly, Y. Wang, R. Munos and O. Teytaud, *Modification of UCT with patterns in Monte-Carlo Go*, Technical Report No.6062, INRIA, 2006.

2006年に一気に成立

CrazyStone [2006 Rémi Coulom]

2006 Computer Olympiad
囲碁9路盤部門で優勝

重要な概念
をほぼ網羅

5月

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

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！

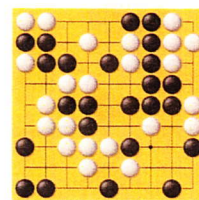
その後の進歩

- 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
 - ZenはKGSという囲碁サイトで2段 = 普通の碁会所なら四段？
- 何が改良されたのか説明したい

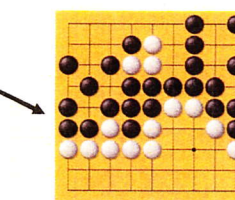
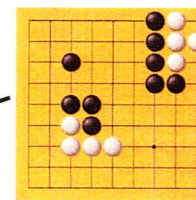
モンテカルロ木探索(MCTS)の性能向上

プレイアウトの改良 (機械学習又は手調整)

- 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
 - 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

2010/11/15

計算機システム 番外編

49

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 に論文あり
 - [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
 - 今のところは、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動く

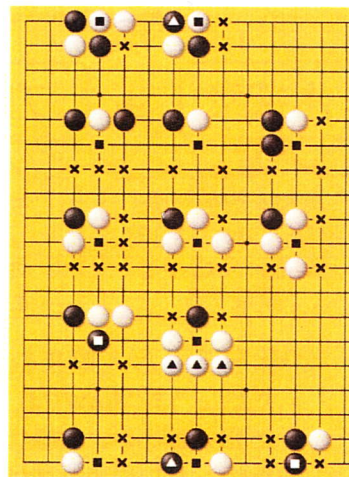
2010/11/15

計算機システム 番外編

50

例、手調整のパターンによる強化 (意味不明な職人芸)

- パターンにマッチする点に高確率で打つようなプレイアウトを実装する
- 実装の都合上、3x3のパターンがよく用いられる
 - プレイアウトの速度を落とさないため
- 右図は、初期のMoGoが利用していたパターン
 - アマ高段者が作成
- 理論ではなく、職人芸の世界
 - しかしこれより良いパターンを作ることは意外と難しい



- 手番にかかわらず着手候補
- 石の上の■はその手番のみ候補
- x don't care
- ▲ その色の石では無い

2010/11/15

計算機システム 番外編

51

CrazyStoneの学習 (機械学習による方法)

- 色々な特徴 (feature) がそれぞれ強さを持つと想定し、勝利に貢献する特徴を強い特徴とする
 - 例えば、特定のパターンがあると勝率が上がる
 - 特徴の例
 - 3x3のパターン、石を取る手、盤端からの距離、直前の着手からの距離、2手前の着手からの距離、...
- ある候補手の強さは、その候補手が持つ特徴の強さを掛け合わせたもの
 - 「強い」候補手を高い確率で選択するプレイアウトを実装
- 学習対象は人間の棋譜
 - 人間の棋譜っぽく見えるようにする

R. Coulom, Computing Elo Ratings of Move Patterns in the Game of Go, Computer Games Workshop, 2007.

2010/11/15

計算機システム 番外編

52

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
 - 探索の専門家、機械学習の専門家、が主
- 並列化の研究も行われている
 - 1000コア以上の並列計算機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

2010/11/15

計算機システム 番外編

53

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 [佐藤高橋2008]
 - 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
- 同様の性質のある問題であれば、有効であると期待される

2010/11/15

計算機システム 番外編

54

汎用性が高い (囲碁以外でも高性能)

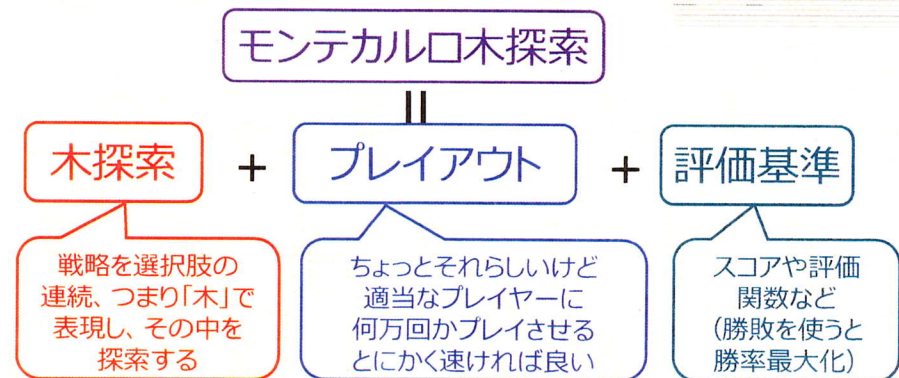
- 一人用ゲーム(パズル)
 - SameGame(さめがめ)
- 二人用ゲーム
 - Amazons、Lines of Action、Hex、(将棋)
- 多人数ゲーム
 - ハーツ(Windowsに付いてくる)、カタンの開拓者たち、Magic the Gathering
- General Game Player Competition (汎用ゲームプレイヤー大会)
 - 大会の場で架空のゲームのルールが提示される
 - その場でプログラムがルールを分析し、直後に対戦する
 - 一人ゲーム、二人ゲーム、多人数ゲームなどごちゃまぜ
 - 総合点が高かったら優勝
 - CADIA Player (UCTをベース)が2年連続で優勝[Finnsson,Björnsson2008]
- ゲーム以外
 - プランニング[NM2009]、最適化、バイオメトリクスセキュリティ

2010/11/15

計算機システム 番外編

55

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

2010/11/15

計算機システム 番外編

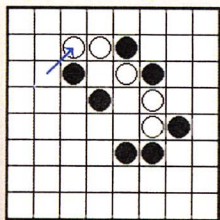
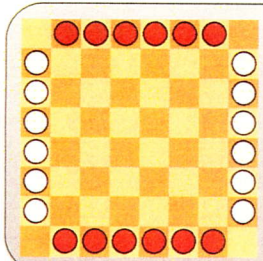
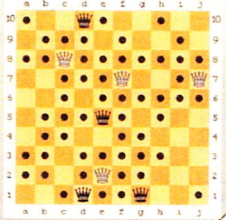
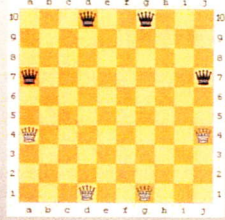
56

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

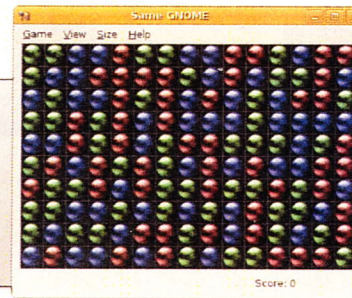
モンテカルロ木探索と $\alpha\beta$ 探索ベースのプログラムが互角くらいという研究

[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いため、MCTSに不向きなゲームとされていたが、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

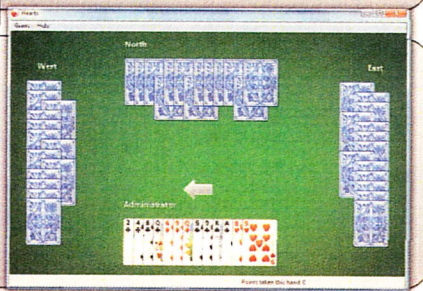
ドイツ製の有名なカードゲーム。MCTSにより強いプログラムが作成された。関係ないがグーグルにはカタン部があるそうである。

[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]



自分の研究

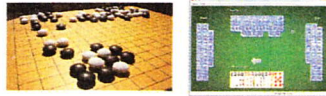
MCTSの並列化

スーパーコンピュータによる

汎用、高性能探索の実現

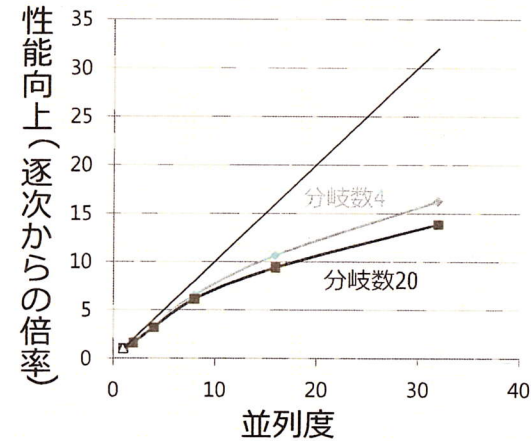
現在の自分の研究

- モンテカルロ木探索 (MCTS) が期待される理由
 - 評価関数不要というアドバンテージによる、汎用性
 - 証明されつつある、高い性能
 - 二人ゲーム: α β 探索以上の性能
 - 囲碁、将棋、アマゾン...
 - 多人数ゲーム: ゲーム理論的なアプローチに迫る性能
 - ハーツ(windows付属), カタンの開拓者
 - プランニング: A*に迫る可能性あり
 - その他: 最適化など
- モンテカルロ法の応用であることの利点
 - スケラビリティの高い探索
 - (耐故障性も (他の探索アルゴリズムより) 高い)



性能向上 32並列まで (1プロセス / node)

1,000,000 シミュレーション
実行の所要時間による

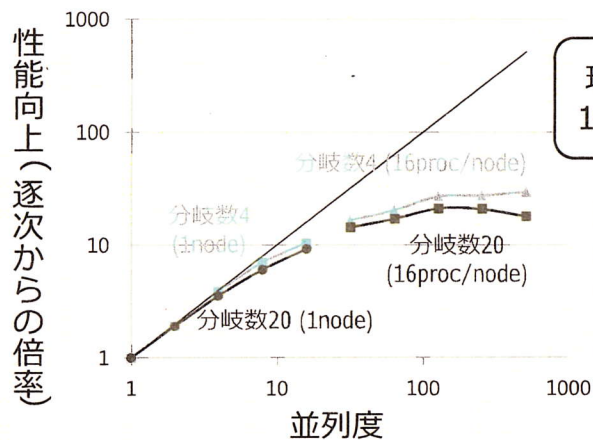


仮想的なゲーム木(P-Game)
分岐数 4と20で探索

分岐数4の時の実行時間

並列度	実行時間 (秒)
1	188.3
2	105.3
4	58.2
8	28.9
16	17.6
32	11.6

512並列までの性能向上(両対数グラフ)



現在の性能向上は
128並列で30倍程度

現在研究中
(α β 探索と比較すると、
既に良い結果だが、今
はもっと良い結果が出
ている)

質問 ?