コンピュータ将棋の技術と GPS 将棋について

JST ERATO 湊離散構造処理系プロジェクト

竹内 聖悟

概要

- GPS 将棋の紹介
- コンピュータ将棋で使われる技術
  - 形勢判断と先読み
  - GPS将棋の技術
- 今後の将棋AI と研究
  - コンピュータ将棋と可視化

近年のコンピュータ将棋

* 2007年: 渡辺明竜王-Bonanza
  * 渡辺竜王の勝利
* 2010年: あから2010-清水市代女流王将
  * あからの勝利
* 2012年: ボンクラーズ-米長邦雄永世棋聖
  * ボンクラーズの勝利
* 2013年: プロ棋士5人対5プログラム
  * 三勝一敗一分でコンピュータ側の勝越
  * 現役プロ棋士に初勝利

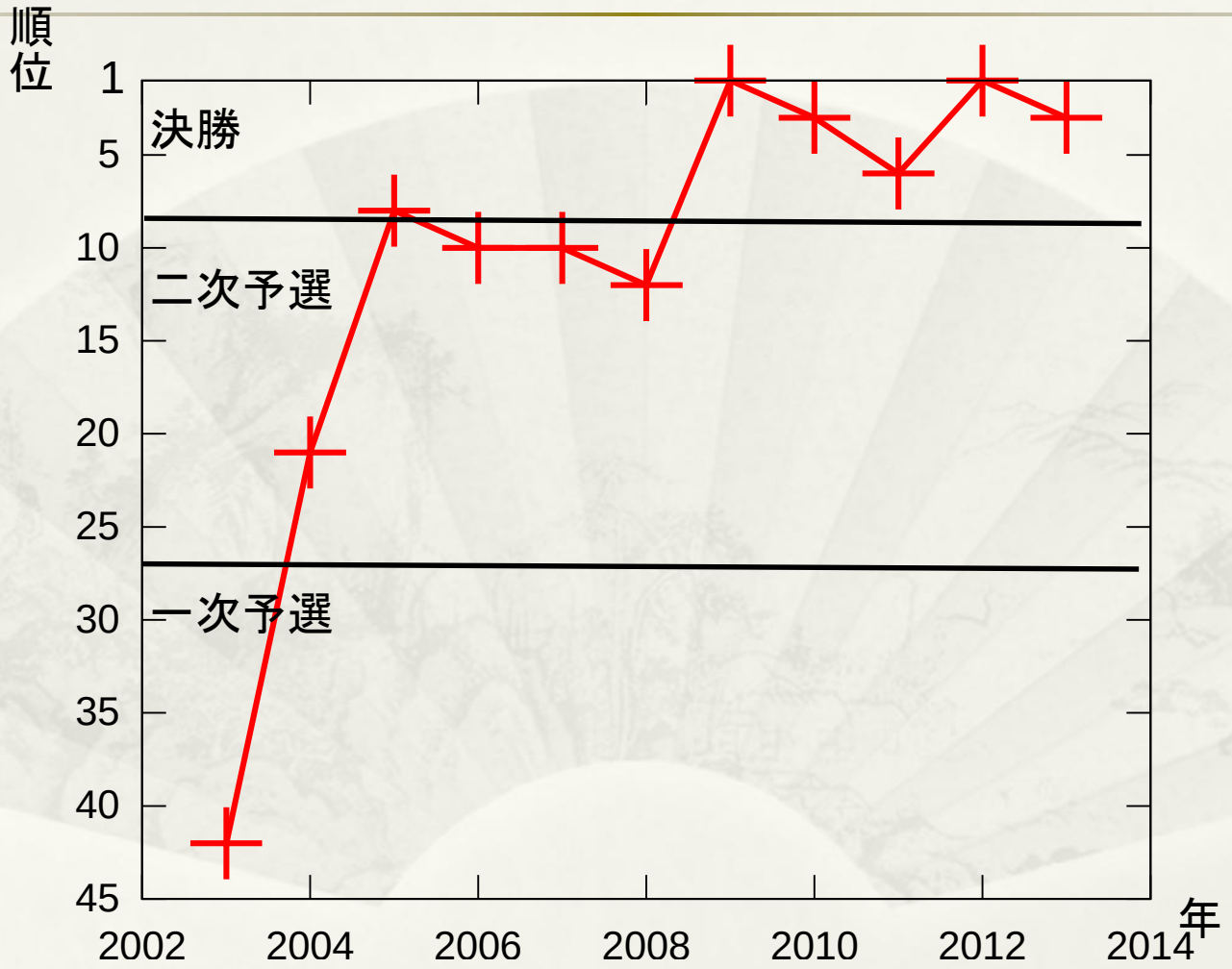
GPS将棋とは

* 将棋プログラム
  * 選手権優勝２回
  * あから2010，電王戦(2013)参加
* GPS = Game Programming Seminar
  * 東京大学大学院総合文化研究科の教員・学生が開催しているセミナー
* 開発チーム：6人
  * セミナーのメンバーだけではない

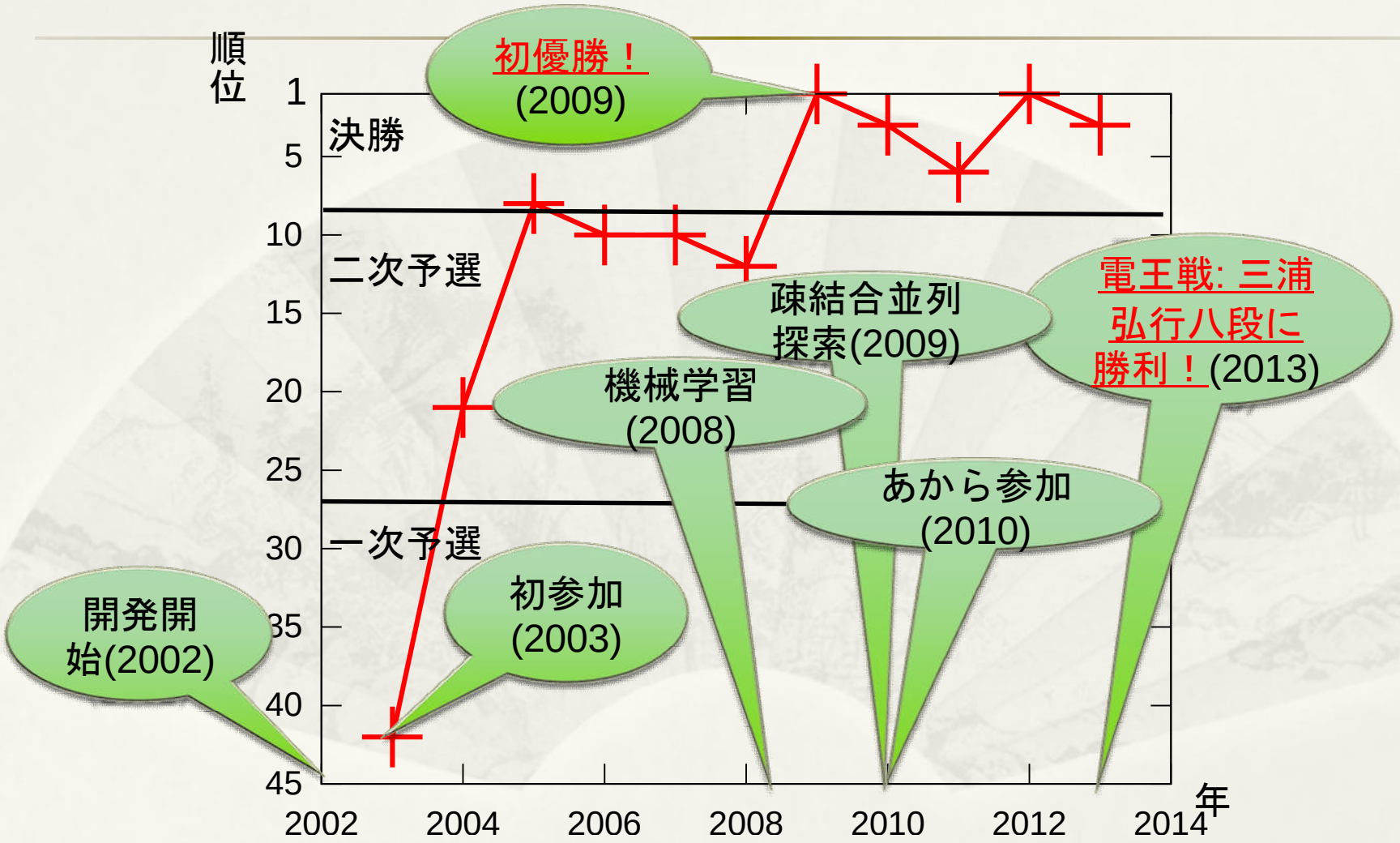
GPS 将棋の特色

* コンピュータチェスやコンピュータ将棋の最新の研究を取り入れている
  * 実現確率を用いた探索
  * 評価関数の機械学習
  * (並列)df-pn による詰将棋探索
  * 疎結合並列探索
* オープンソース
* 将棋ライブラリの公開

コンピュータ将棋選手権の成績



GPS将棋の成績と出来事



Twitter

* タイトル戦の形勢判断
  * 無料中継のタイトル戦
  * 最近はお休み中
* 各局面に対し
  * 評価値,読み筋,探索時間
  * 詰みや狙いなど

指手，評価値, 最善応手手順, 秒数

[(100) △９九と] -4473 ▲９五香△７七歩▲同角△同金▲８六玉△６六角▲７六銀△８五香▲同銀△同桂▲３二角成△同玉▲７二飛△４二金▲６四歩△７五角▲同飛成△同銀▲同玉△８四銀▲７四玉△７三飛

ゲーム研究

- 情報科学/人工知能
  - 完全解析
  - 強いプログラム
- 認知科学
  - 強いプレーヤーはどう考えるか
- 教育
  - どう教えたら強くなりやすいか

強いプログラムを作るには

* 評価関数（形勢判断）
* 探索（先読み）

* どちらかが完全 ⇒ 解析できる
  * 現実的でない

* トレードオフ: 評価関数の**正確さ**と探索の**速さ**
  * 現在: そこそこ正確で軽い評価関数 + 速い探索

1手読み

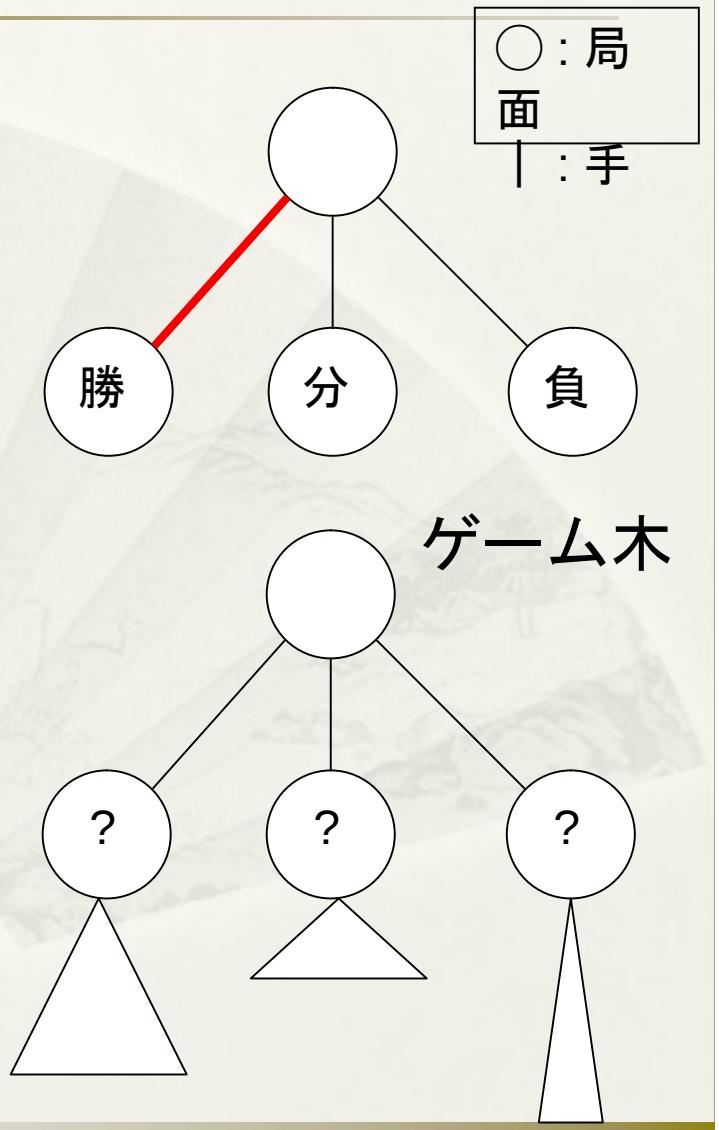
* 1手進めてから選ぶ
  * 1手で終わるゲームなら解析

実際のゲーム:

  * 1手では終わらない

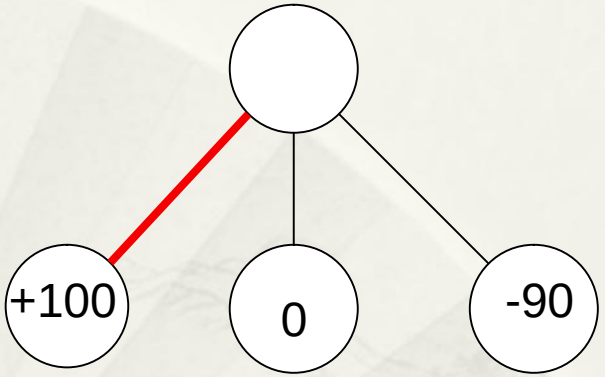
1手先の勝ち負けを知りたい

⇒ 形勢判断



1手読み + 形勢判断

* 1手進め、形勢が良い手を選ぶ
  * 形勢判断が完璧なら解析

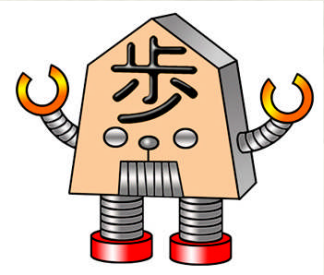


実際のゲームの形勢判断：
  * 不正確

⇒ 深い探索，正確な形勢判断が必要

ゲーム木サイズ

ゲーム	サイズ	コンピュータの強さ
チェッカー	10^{30}	解析済み（引分）
オセロ	10^{60}	チャンピオンを超えた
チェス	10^{120}	チャンピオンを超えた
将棋	10^{220}	プロ棋士レベル？
囲碁(19路)	10^{360}	アマチュアレベル



阿伽羅(あから) = 10^{224}

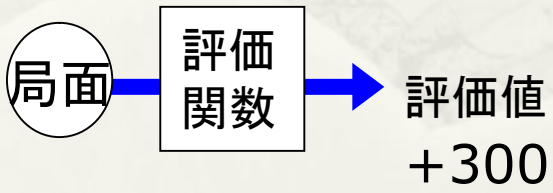
現実的には解けない

強くするために

* 正確な形勢判断
  * 評価関数の重みを機械学習により調整
* 効率的な探索
  * 枝刈, 延長, 短縮
  * 実現確率探索
  * 詰将棋探索
  * 並列探索
    * 疎結合並列探索

評価関数

* 局面の良し悪しを数値化
* 評価項目/特徴とその重みからなる
  * 例：　(駒得) × 5 + (危険度) × 10 + . . .
* 重みは機械学習で自動調整
* 特徴は人間が考える



評価関数, ひとむかし

- 特徴を考える
  - 人間が考える, 将棋の知識が必要
  - 駒の点数, 王の危険度…
- 重みをつける
  - 人間が考える, 将棋の知識が必要
  - 歩が100点として、香車は200?400?
- パラメータ数に限界
  - せいぜい数百数千?

評価関数, 現在

* 特徴をたくさん考える
  * 人間が考える, 将棋の知識が必要
  * 駒の点数, 王の危険度, 3駒間の関係...
* 重みをつける
  * 機械学習による自動処理
  * 棋譜の指し手を選ぶように重みを調整
* パラメータ数は数百万, 数千万, 億?

GPS将棋の評価関数

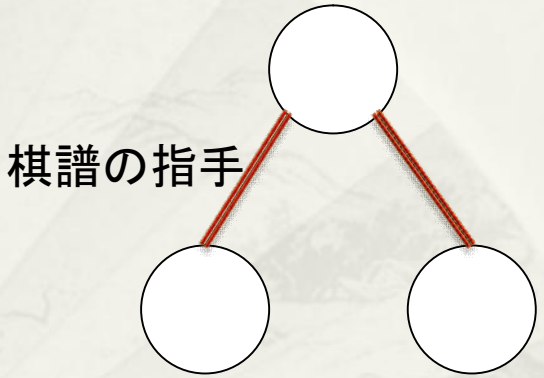
* 序盤，中盤，中盤2，終盤 の4種類
  * 8,952,491 項目（重み0の項目も含める）
  * 局面の進行度に基づき内分を取る
* 人間の知識で項目を選択
* 重みは棋譜から調整
* 調整後強くなったか対戦で確認，採否

評価関数の学習

- 評価関数のパラメータ調整
  - 強化学習や進化的アルゴリズム
  - あまり成功していなかった
- Bonanza が成功(2006)
  - GPW 2006 にて機械学習の発表
  - 2008年ソースコードの公開
- 現在、将棋プログラムの大半が利用
  - オンライン学習化など研究も進んでいる

学習のイメージ

- 棋譜の指手を真似られるように調整
  - プログラムの指手との一致率を高くする
  - 棋譜以外の手を選ばない
    - 兄弟局面の比較
  - 探索を行う
    - 探索末端局面同士の比較



歩兵	7	11
金将	3	1
f(x)	19	15

簡単な例

f(x) = 3 × 歩 + 3 × 金

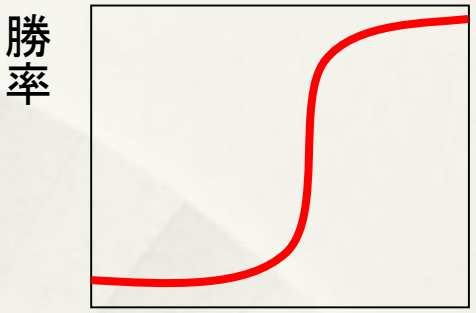
評価関数の項目

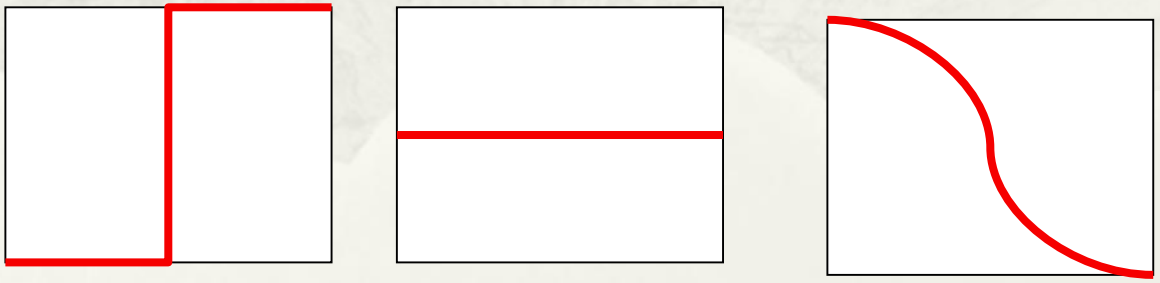
- $f(x) = \sum w_i x_i$
  - x_i: 駒の枚数，３駒の関係
- どうやって項目を選ぶか
  - 人間の知識が必要
- 自動生成の研究: あまり成功していない

Evaluation Curve

手法:

- 評価値に対し勝率をプロット
- 有効な特徴の発見を目的とする
  - X軸: 評価値
  - Y軸: 勝率
- 強いプログラムのカーブ:
  - 単調性（評価値と勝率の大小関係が単調増加）
  - 一貫性（異なる棋譜でも同じカーブになる）

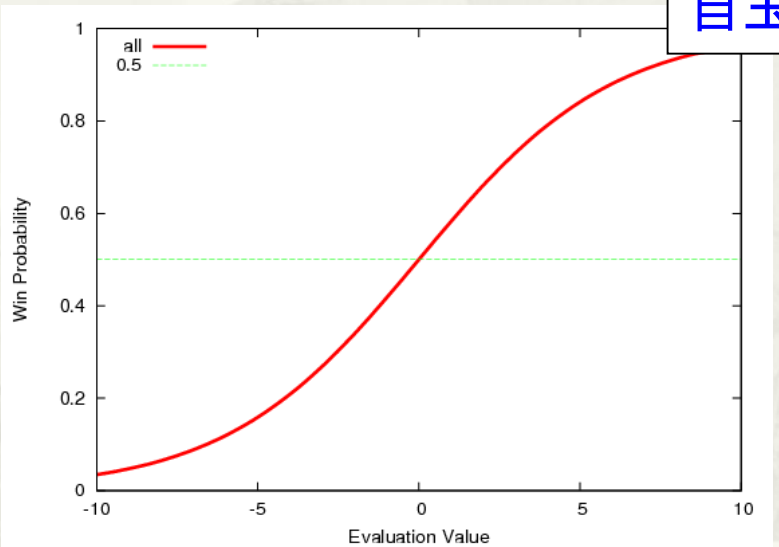


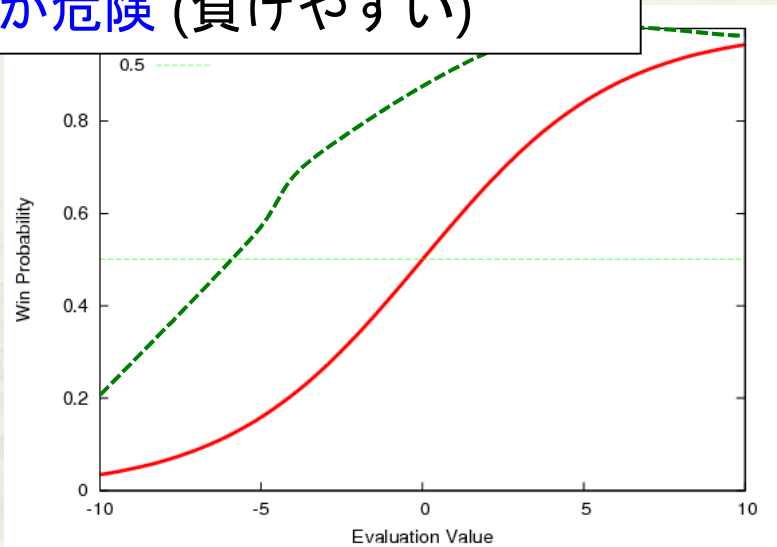


E.C. の分離

例) 駒の価値だけの評価関数
全局面
自玉が安全 (勝ちやすい)
自玉が危険 (負けやすい)

棋譜セット A, B, C



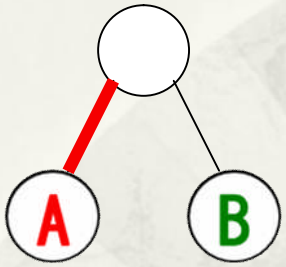


一本 ➡ 一貫
(良いプログラム)

分離 ➡ 非一貫
(良くないプログラム)

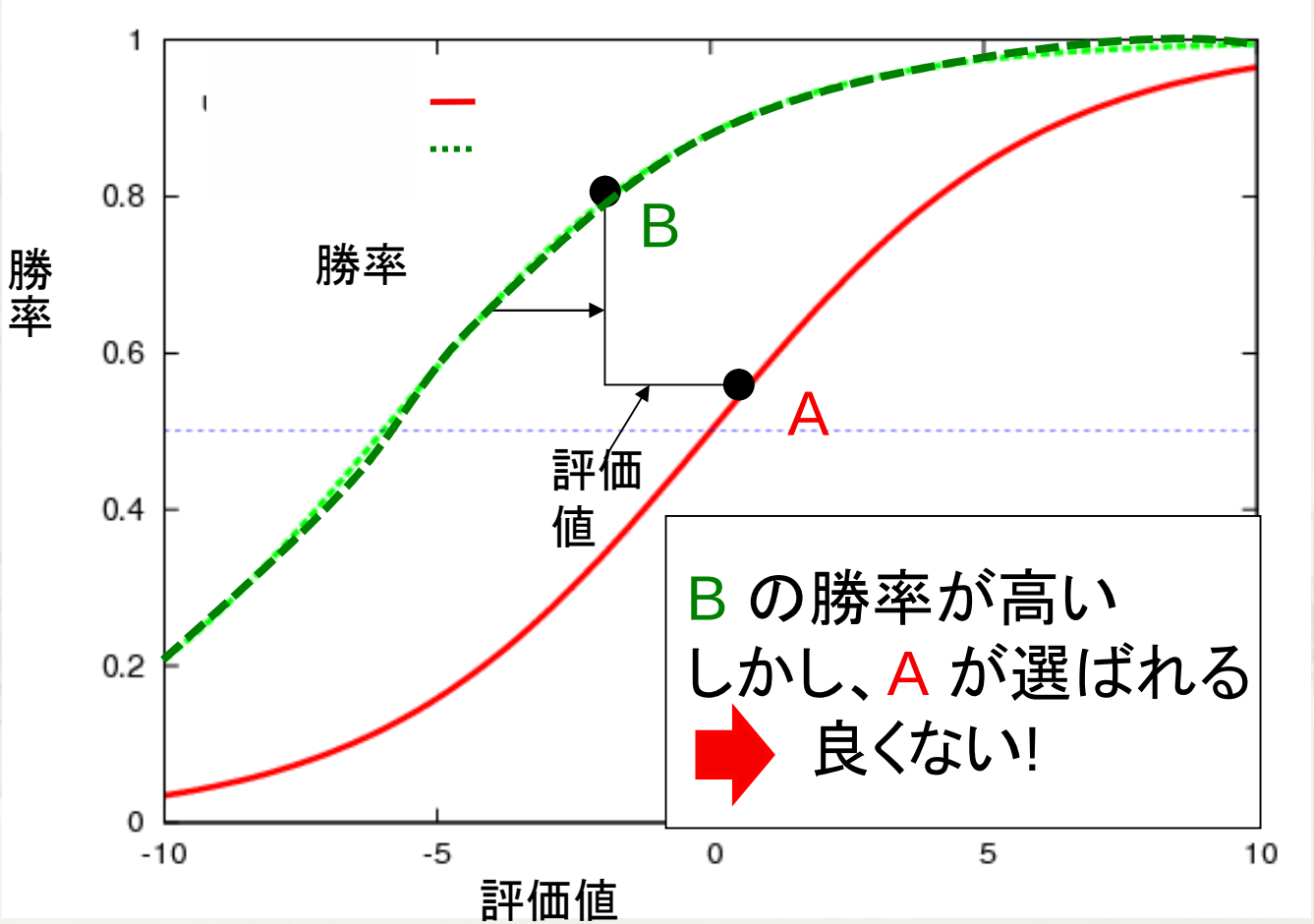
分離したE.C. の問題点

ゲーム木



+1
0.5

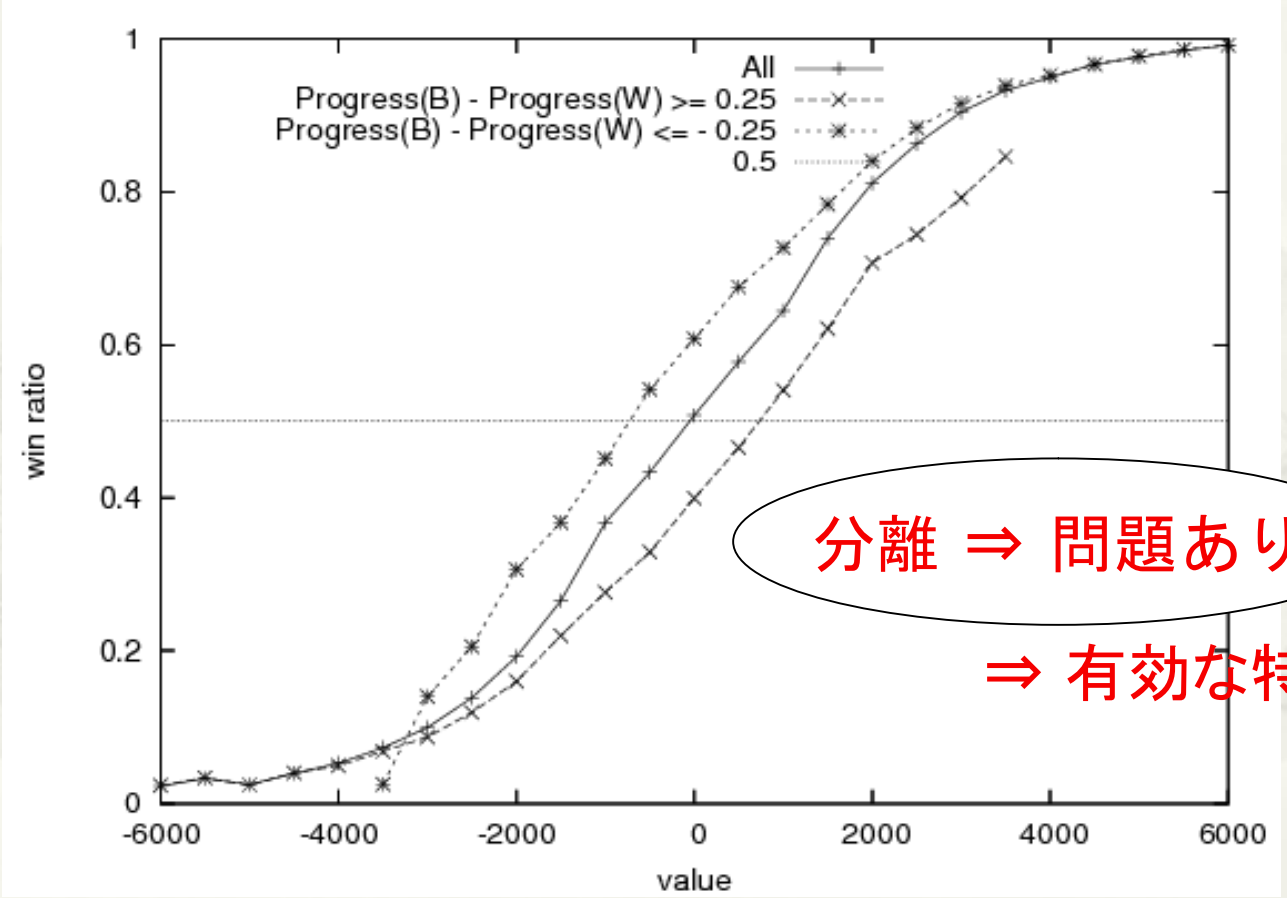
-3
0.8



実例

- 2006年前後のGPS 将棋の評価関数
  - ※ 機械学習以前のもの
- 王の危険度: 王周辺にある敵の利きの数
  - 多いほど負けやすい
  - 利き: 駒が動ける範囲
- 評価関数が上記を正しく評価できているなら、勝率に影響はない
  - 正しく評価: Evaluation Curve が1本化される

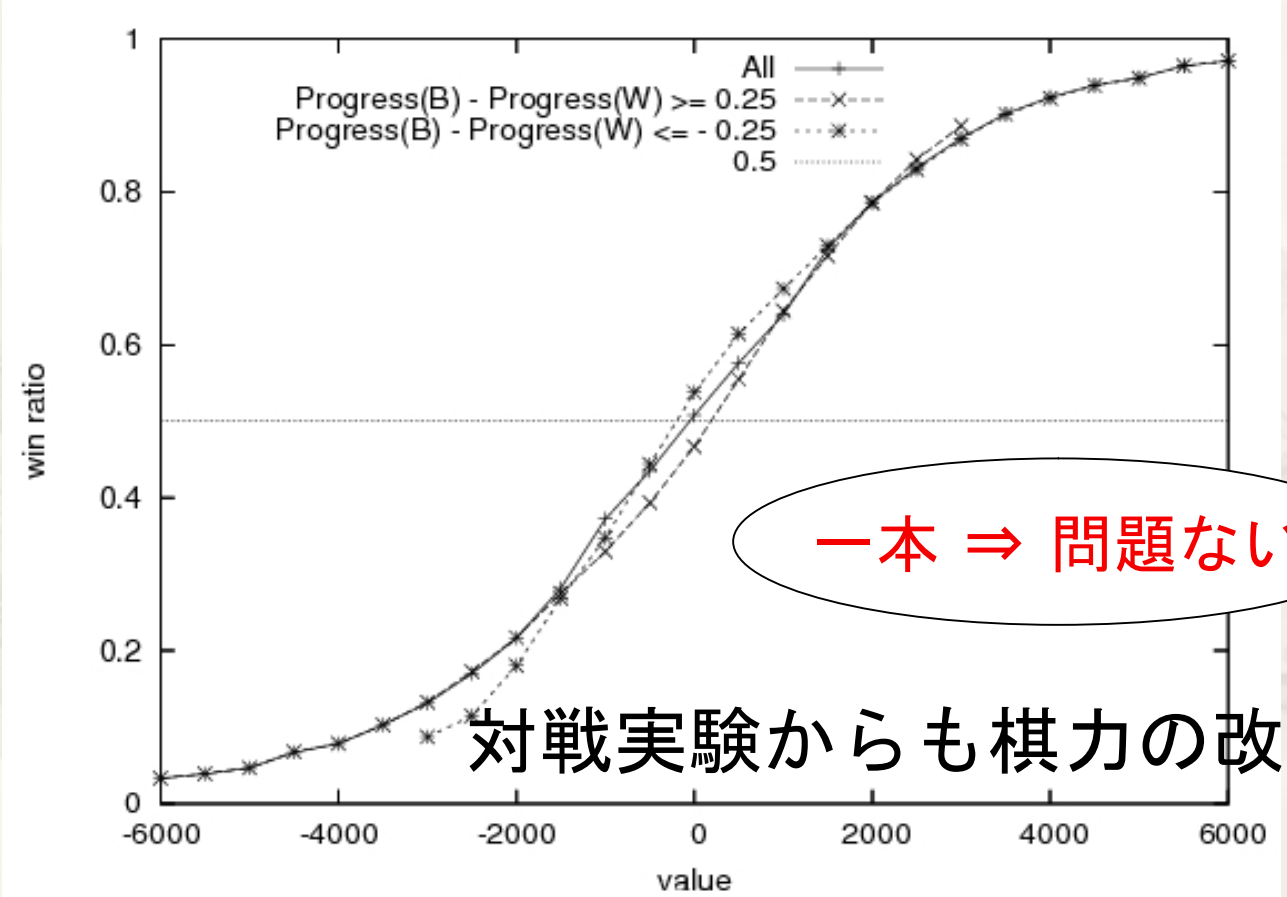
改良前



分離 ⇒ 問題あり

⇒ 有効な特徴！

改良後



一本 ⇒ 問題ない

対戦実験からも棋力の改善を確認

E.C.

* 現在は使われていない

* 前述の通り、項目はもはや1千万近い

⇛ 全項目のグラフを見るのは非現実的

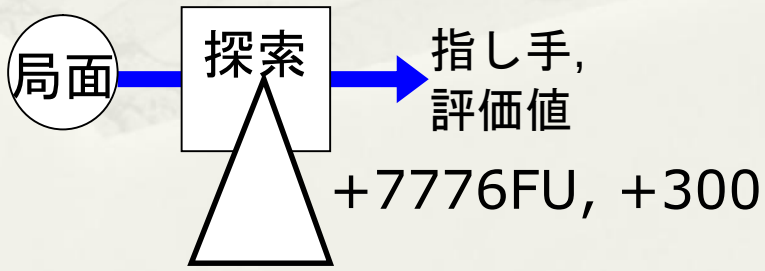
  * 数値化するなど工夫が必要だが、未着手

評価関数

- 線形の評価関数が主流
  - ニューラルネットワークなども一部あり
  - 基本的に速度が優先される
- 重さは機械学習による自動調整
  - 数百万の重みを調整
- 特徴は人間が知識を使って見つける
  - サポートする手法が必要

探索

* 評価関数 + $\alpha\beta$ 探索
  * 互いに最善を尽くす前提
  * 深さ打ち切り探索
  * 葉ノードで、評価関数による評価値を得る
* 一般に, 深く探索するほど強い
* 速度を上げる工夫

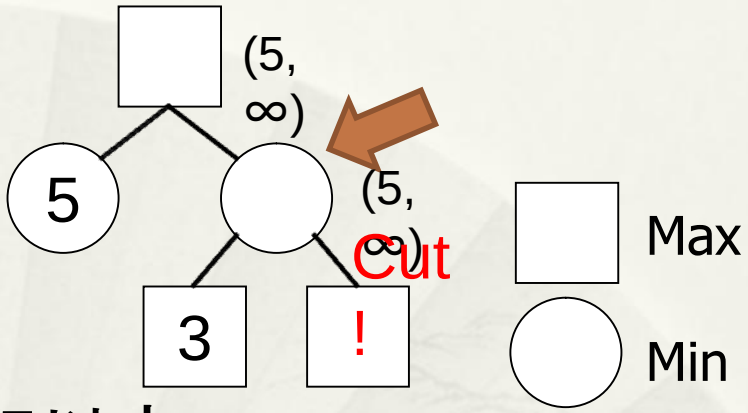


αβ探索

- Min-Max を効率的に行い、同じ結果を得る
- 不要な探索を行わない : 枝刈
- 探索窓, alpha-beta window の導入
  - 興味のある評価値の範囲
  - (alpha, beta) として表記
  - 返り値V で更新
    - Max : If (V > alpha) alpha = V
    - Min : if (V < beta)   beta = V

枝刈

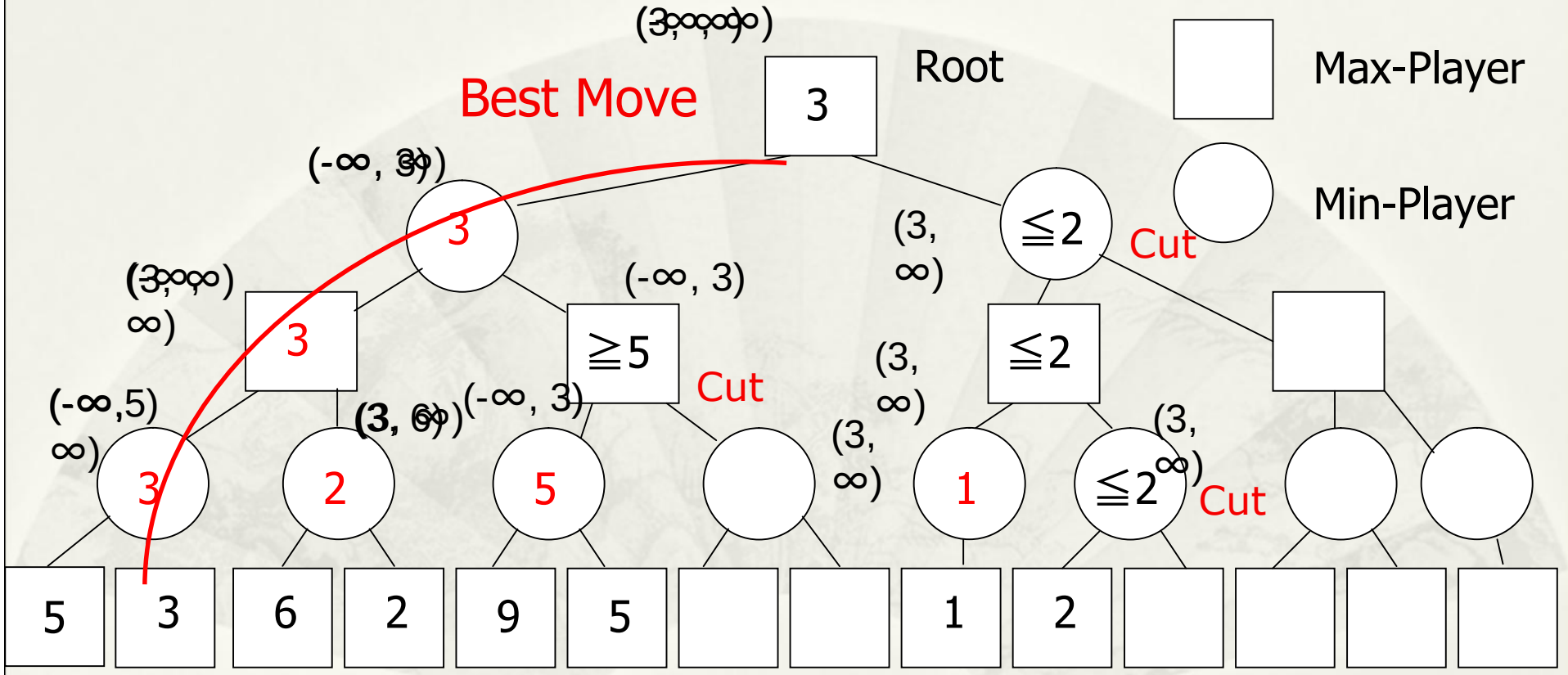
* 枝刈条件
  * Max : V >= beta
  * Min : V <= alpha



例:

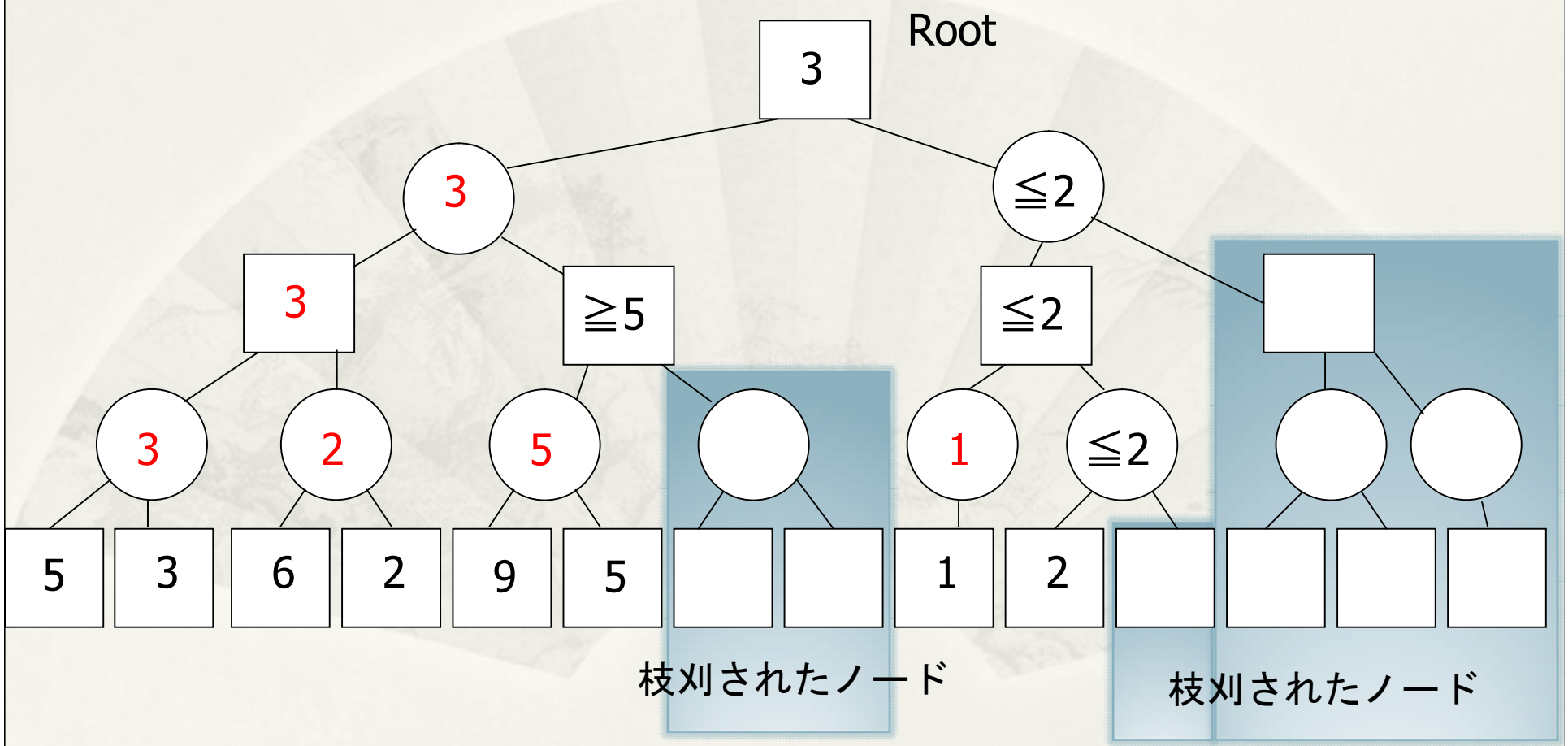
  * ルートのMax ノードは5以上
  * 矢印のノードに左の子ノードから3が返った
    ⇛ 値は3以下になる (∵ Min ノード)
  * ルートには3以下しか返らない ⇛ 選ばれない
  * それ以上探索するのは無駄 ⇛ 枝刈

αβ探索の挙動



数字は”評価値”
点数が高いほどMax-Player が有利

αβ探索の結果



探索の効率化に重要な情報

* 探索順序
  * 最善を先に探索できると効率的
  * 最悪の場合、枝刈が起こらないことも
* 探索窓の広さ
  * 狭いほど枝刈は起こりやすい
* ハッシュ表
  * 探索結果の保持 : 同一局面の探索を行わない
  * 手の並び替え: 浅い探索結果を元に

探索の工夫

* 枝刈・探索延長
* 探索順序
* 探索窓
* ハードウェア
  * 専用ハードウェア (例: Deep Blue)
  * CPUのオーバークロック
  * マルチコア
  * クラスタ/疎結合

並列化の難しさ

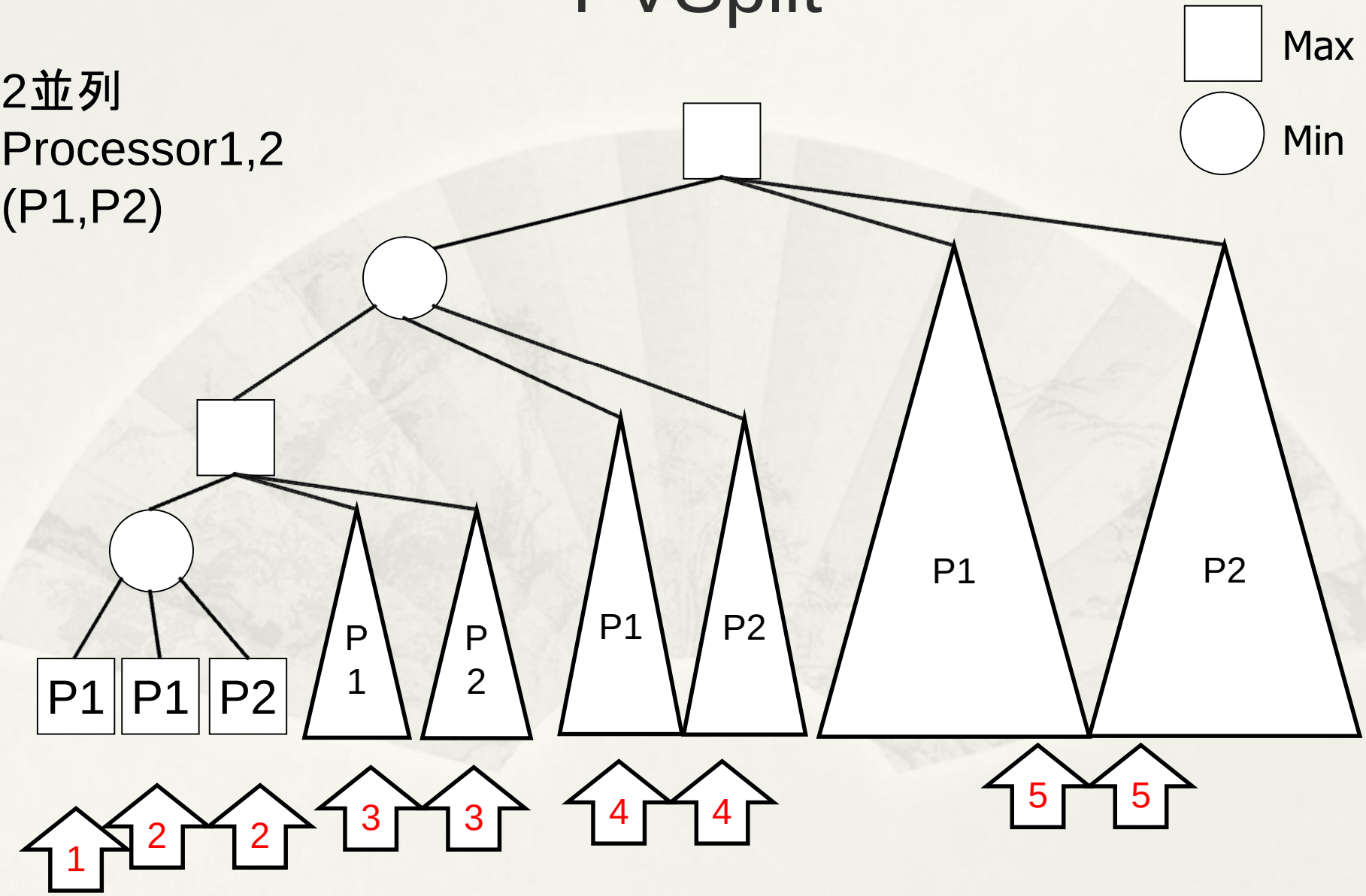
- 並列処理が可能か?
  - 処理に順序依存性があると難しい
- オーバヘッド
  - 探索 : 逐次なら枝刈されるノードの探索
  - 同期 : 他のプロセッサの結果を待つ
  - 通信 : 仕事の分割, 仕事を通信, 通信遅延

メモリ共有環境

- プロセッサ間の通信は十分速い
  - 通信オーバヘッドはあまりない
- PV Split
  - 左端を1人で展開
  - 残りのノードを並列にnull window search
    - ハッシュ表を共有
    - ロックレスハッシュ

PVSplit

2並列
Processor1,2
(P1,P2)



GPS将棋の疎結合並列探索

* 2010年の第20回大会からクラスタ参加
* 順位: 3位 -> 6位 -> 1位! -> 3位
* コア数: 666, 800, 3200, 3200
  * 情報教育棟のiMac, Amazon EC2 (2011)

* ネットワークで緩く接続されたマシン群
  * 通信速度はそんなに速くない
  * 情報のやり取りがあまり出来ない
  * 協調的に動かすのが難しい

計算機群

* 情報基盤センター教育用計算機を利用
  * 東京大学駒場キャンハ゜ス情報教育棟
  * 平日と土曜日は学生が利用
    * 土曜日は一部演習室は閉鎖されている
* 日曜祝日しか利用できない
* 利用申請が必要
  * 申請者は離れられない

http://gps.tanaka.ecc.u-tokyo.ac.jp/gpsshogi

クラスタ構成（選手権）

	iMac	他	合計 台/コア数	備考
2010	307	7	314 / 666	Intel Core 2 Duo 2.0GHz
2011	208	55	263 / 832	Amazon EC2 40台
2012	788	9	797 / 3224	iMac 入れ替え Intel Core i5 2.5GHz
2013	791	13	804 / 3318	

単純なアイデア

* 木を展開していき、台数分ノードができたら全ノードに1台ずつ割り振る

⇒ 無駄な探索がかなり多い

* 台数効果が出ない
  * 将棋の平均合法手数は80
  * 1手深く探索するには80台
  * 2手深く探索するには6,400台必要！

従来手法

* YBWC, APHID, TDSAB
  * YBWC はPVSplit に近い手法
* 合議 (2010)

* 実際に成功した例があまりない
* チェスでもRybka がクラスタ探索をしているが、詳細は不明

http://cluster.rybkachess.com/

GPS将棋の疎結合並列探索

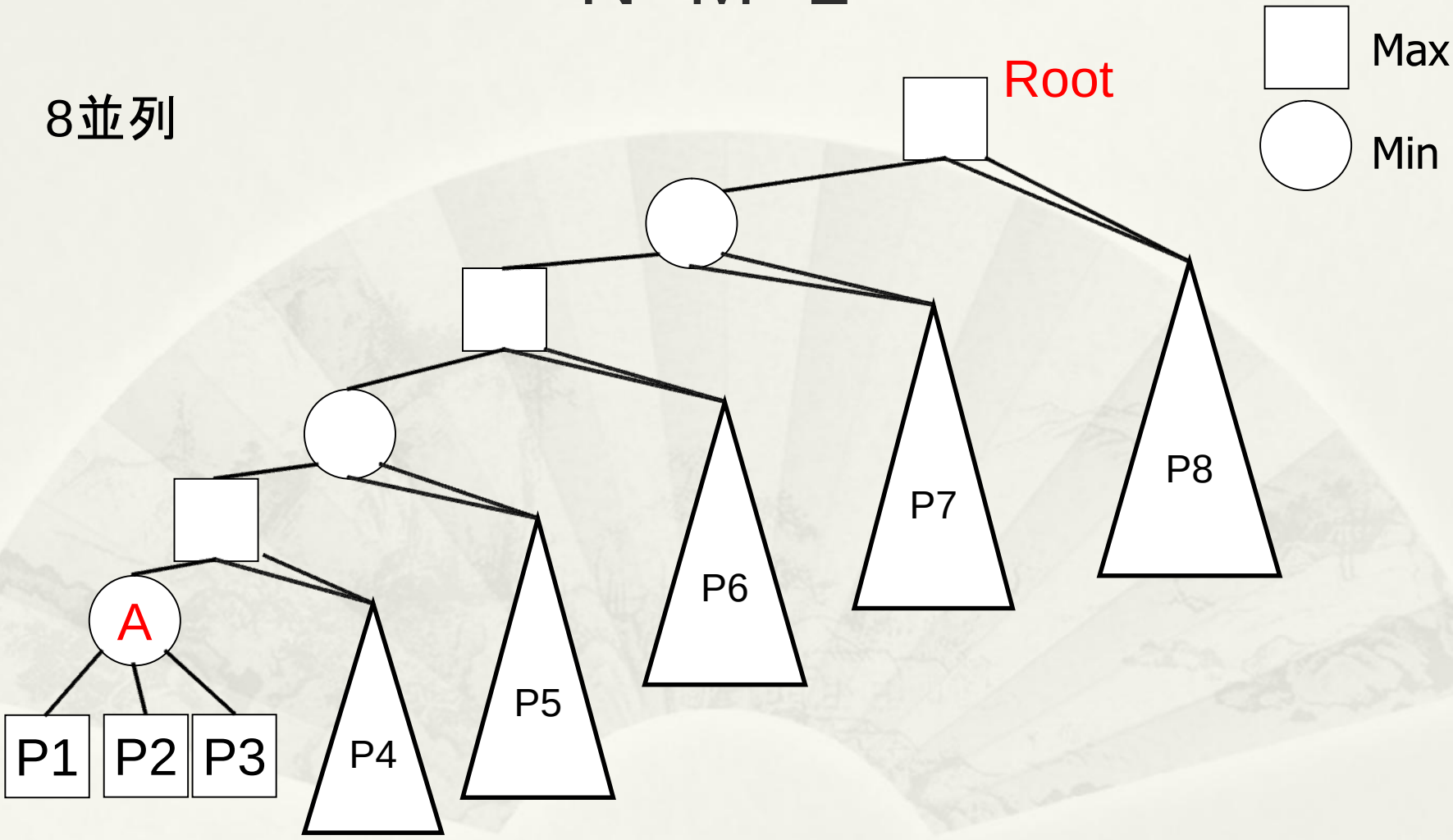
* 探索窓を共有しない
  * 同期オーバヘッドと効率のトレードオフ
* ハッシュ表は各自で持つ
  * 割当て時に前回担当分に割当てられる
  * ここでは、通信オーバヘッドはない
* 並び替えは探索など
  * 探索オーバヘッドはあるが、並び替えがうまくいけば、少なく抑えられる

GPS将棋のアプローチ（概要）

* ルートで手生成
* 上位N 手にマシンを割当
  * 順位に応じて台数を変化
  * 残りの手は1台で通常探索
  * 前回担当した局面は同じマシンが担当
* それぞれ, 手を進めた局面で手生成
  * 各手の台数が1台なら1台で通常探索
  * 上位M手にマシンを割当, 残りは1台で探索
  * 以下、繰り返し
* 「残りの手」が最善となったら探索時間延長

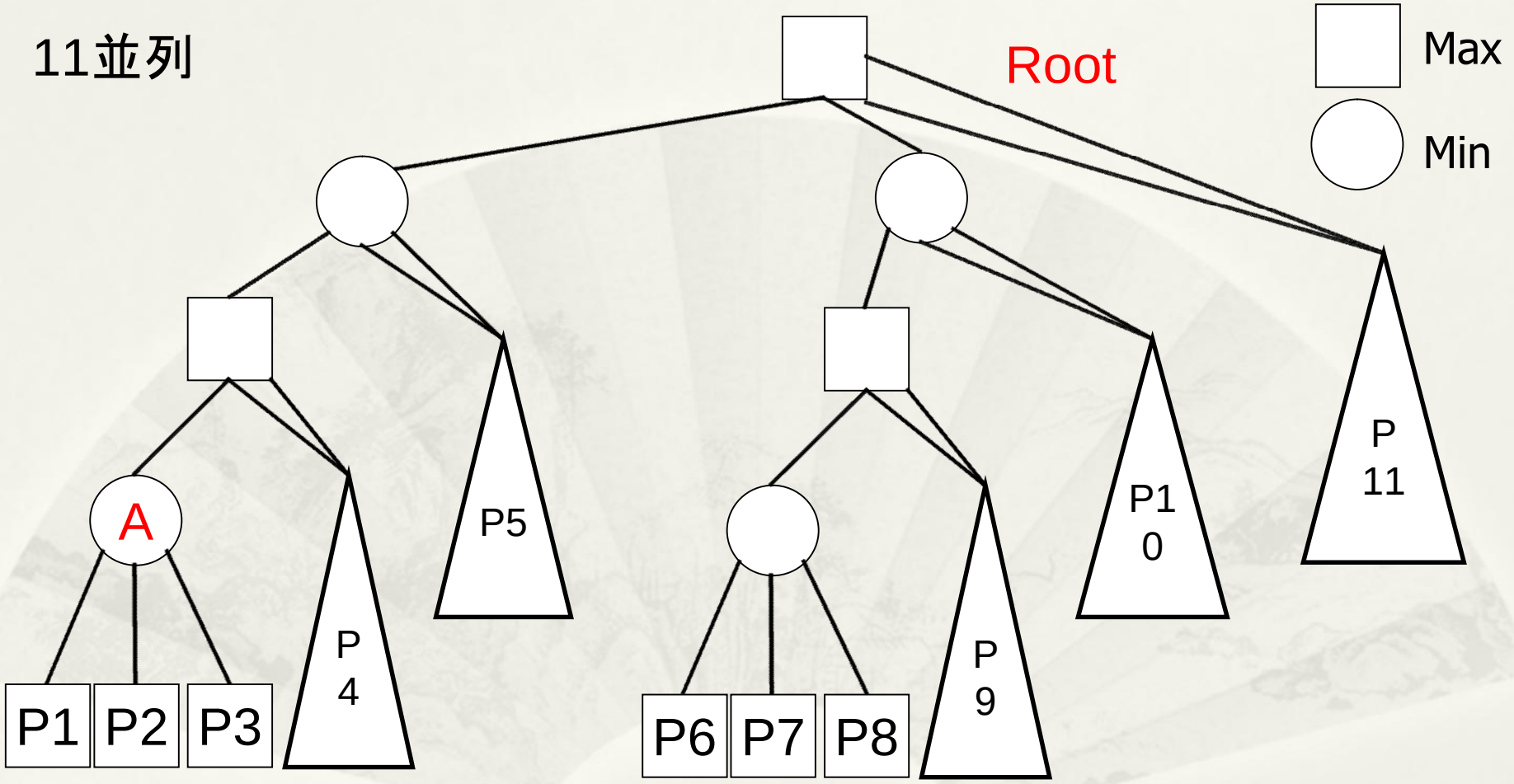
N=M=1

8並列



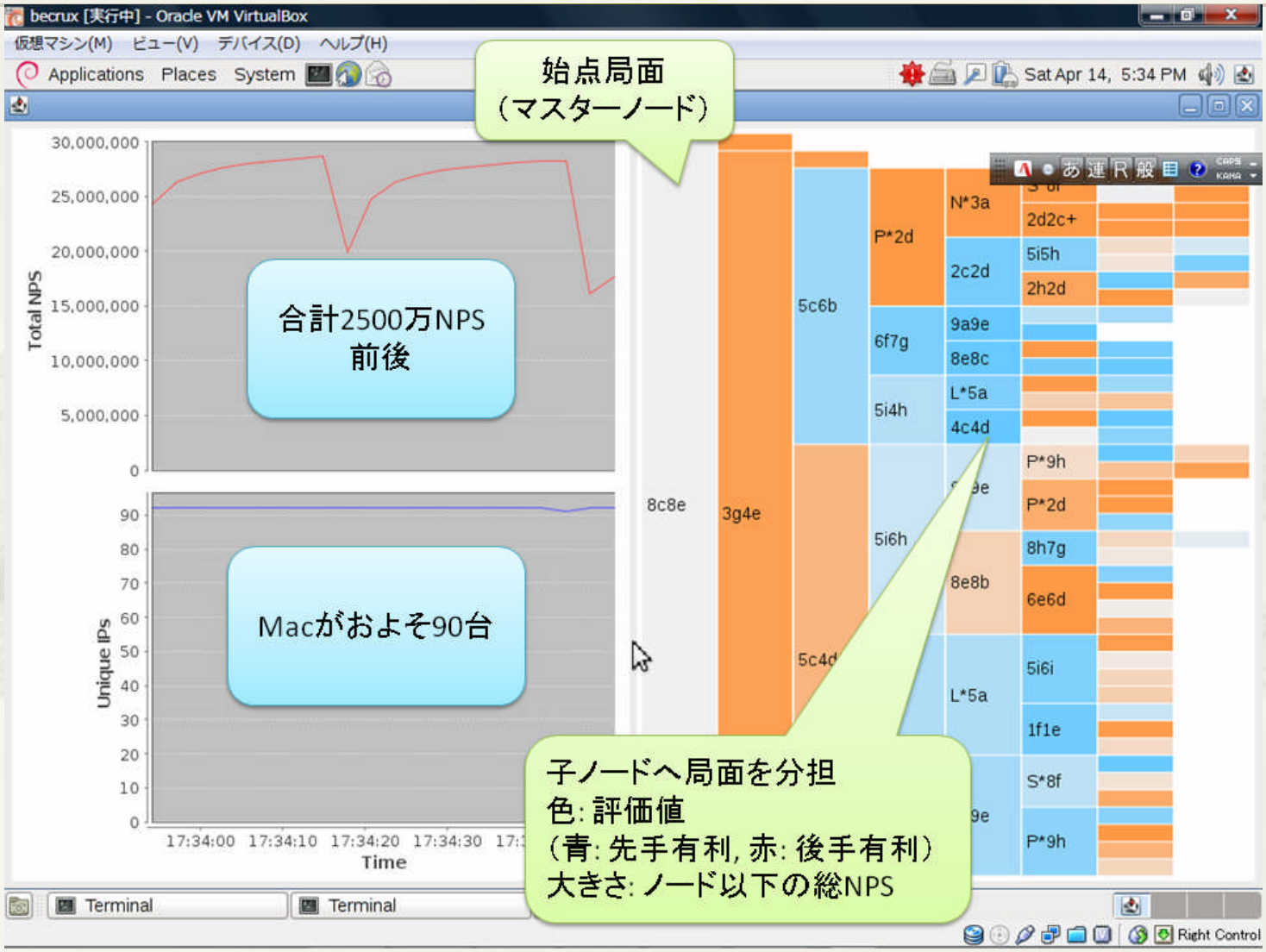
P8 と比較し, Aの子ノードは5手深く探索

N=2, M=1



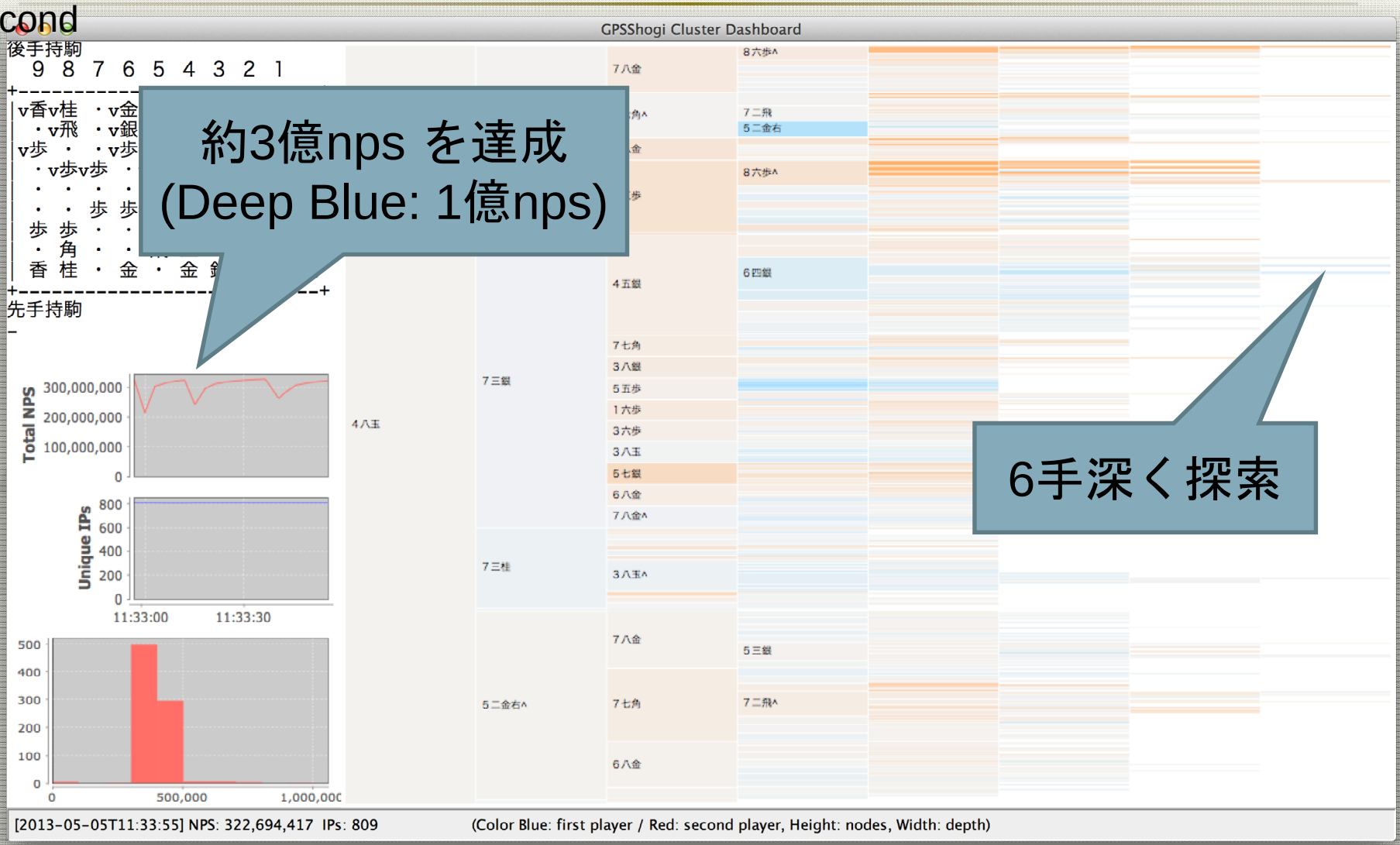
P11 と比較し, Aの子ノードは3手深く探索

クラスタの思考の可視化



2013年5月の例

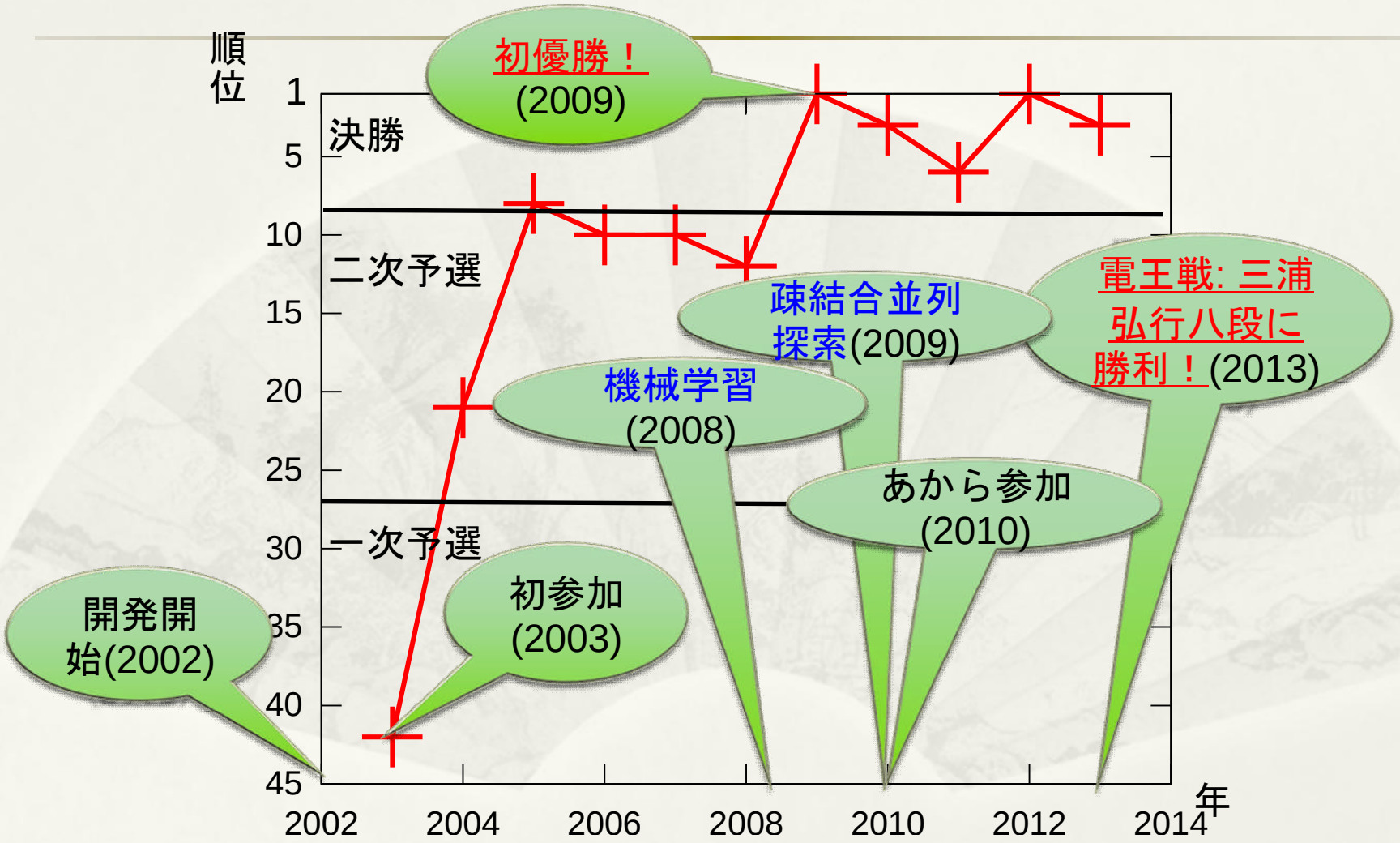
nps = nodes per second



探索のまとめ

- アルファベータ探索と評価関数
- 効率的な探索の工夫
  - 枝刈や探索延長など
- 並列探索
  - SMP 環境での並列化
  - 疎結合並列探索

GPS将棋の成績と出来事



今後の展望など

* 人間のトップに勝つことは、１つの目標
  * 段々近付いてきた？
* さらに強いプレイヤの作成
  * チェスは現在も強くなり続けている
* 強いプレイヤがいないと出来ない研究
  * プレイヤの強さの解析
  * 時代の違うプレイヤの比較

これからの研究

* 人間らしい指手
  * 昔から研究されている
  * 「人間らしい」ことの評価の難しさ
* 人間のサポート
  * 感想戦支援など
* 思考の可視化
  * 言語化の研究など

コンピュータ将棋と可視化

* 「何を考えているか分からない」
  * 変な手に見えても正解かもしれない

思考の可視化

* デバグや性能強化
* 将棋の解説，棋譜の検討

* 開発に役立つ可視化
  * クラスタの可視化，Evaluation Curve