「芝浦将棋 Jr.」のチーム紹介

2016 年 3 月 1 日
芝浦工業大学情報工学科
五十嵐治一，原 悠一，和田悠介，古根村 光，桐井杏樹

1. はじめに

本稿は，第 26 回世界コンピュータ将棋選手権（2016 年 5 月開催）に出場予定の「芝浦将棋 Jr.」（シバウラショウギ　ジュニア）の紹介文です．本チームは芝浦工業大学工学部情報工学科の学生と教員により構成されており，教育と研究の一環として活動しています．これまで 2010 年～2012 年の間,「芝浦将棋」[1]のチーム名で本大会へ出場しておりました．この芝浦将棋は，保木邦仁さんが開発し，インターネット上でソースコードを公開されている“Bonanza”[2]を探索エンジンとして使用しておりました．これに対し，2013 年の大会からは「芝浦将棋 Jr.」というチーム名で参加しております．芝浦将棋 Jr.では，盤面表現のデータ構造として Bonanza でも用いられている bitboard 構造を将棋向きの独自の形式に変更し，かつ，探索エンジン自体も完全にゼロから独自開発しております．以下では，本チームの開発コンセプト，開発メンバー，特徴，今年の追加修正点，将来計画などについて述べます．

2. 開発コンセプト

現在，Bonanza を初めとする評価関数の学習にはプロ棋士同士の棋譜データベースなどを教師データとする教師付学習が用いられています．また，序盤の定跡を利用することも常識とされています．それに対して，芝浦将棋 Jr.やその前身となった芝浦将棋は，

「人間の持っている将棋に関する専門知識に頼ることなく，コンピュータが対局を通して試行錯誤しながら，自ら将棋を学び，棋力を向上させ，人間のレベルを超えること」

を基本的な目標としています．これによって，人間の定跡や戦法が再現されるのか，あるいは，人間も考えつかなかった新しい定跡や戦法が創発されるのか，その点に興味があります．もちろん，すぐにこの目標が達成できるとは考えておりません．長期的に開発を進めて行くという考えでおります．

これまでの芝浦将棋でも，評価関数の学習においては，教師付き学習ではなく，対局結果を報酬として反映させる強化学習を適用することや，対局や学習時に序盤定跡を一切使

用しないことを世界選手権大会で試みて参りました．芝浦将棋 Jr.でも強化学習で評価関数を学習するなど，上記の開発コンセプトに沿って開発を進めていくつもりです.

3. 開発メンバー

開発を統括している五十嵐は芝浦工業大学工学部情報工学科に所属する教員です．残りの開発メンバーは五十嵐研究室の学生です．芝浦将棋 Jr.は，2012 年 4 月から 4 年生の川内博世が最初に開発に着手し，2013 年～2014 年に 4 年生大串明と修士 2 年生谷川俊策がプログラムのバグ修正と探索エンジンの大幅な機能強化を行いました．2014 年 4 月からは 4 年生の小川俊樹と原 悠一とが加わり，それぞれ評価関数の差分計算や前向き枝刈りの強化を図り, 処理速度を向上させました. さらに, 2015 年 4 月からは 4 年生の和田悠介, 古根村 光, 桐井杏樹が加わり，探索処理の並列化を可能とし，また，評価関数の学習も試みました.

2016 年 4 月からは，修士 2 年になった原，修士課程へ進学した和田，古根村，桐井の 4 人体制で開発と研究に取り組んでいます.

4.「芝浦将棋 Jr.」の特徴

芝浦将棋 Jr.の特徴を，以下の１）～６）のようにまとめました．その中で，<u>今年のプログラムに新たに追加された機能は，６）の探索処理の並列化です</u>.

１）盤面表現のデータ構造

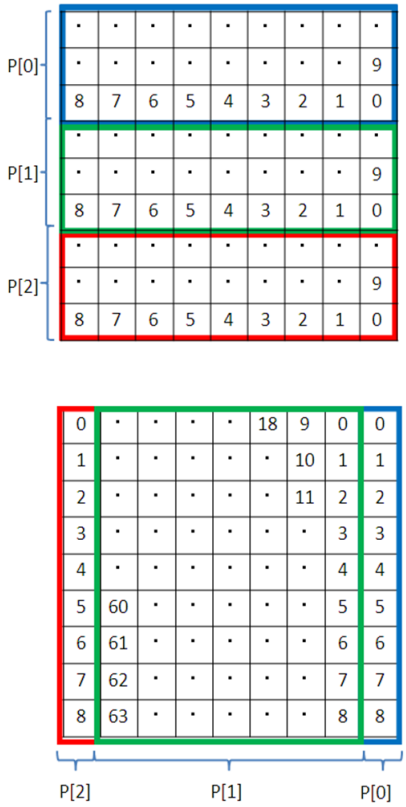
Bonanza と同じように bitboard により盤面上の駒の配置や利きを表現し，評価値計算や合法手の生成などの処理に使用しています. ただし, bitboard のデータ格納構造が異なります. 図１に Bonanza と本チームとが用いている bitboard のデータ構造を示します.

図１b のようなデータ構造を用いるメリットは次の①～③の３つです.

①１つの変数内で盤面更新や飛び駒の利き算出が計算できた方が高速である.
②Magic bitboard（後述）を用いる際には，中央の変数 P[1]に対する処理だけで済む.
③使用コンピュータ CPU が 64bit マシンであるので, 64bit の変数の処理に適している.

①は，将棋の駒は横よりも縦に動くものの方が多く，盤面の更新や飛び駒（飛車，角，香車）の利き算出などを行う際には，図１b のように１つの変数内で処理できた方が高速です．なお，この利き算出には，両端の駒の有無に関する情報は必要ないことを利用しています.

②については，Bonanza では，飛び駒の利きは図１a からわかるように P[0],P[1],P[2]の３つの変数をまたいでいるので，このまま連続する bit 列として取り出すことができません.



(a) Bonanza における bitboard 構造：
盤面は三段ずつ横方向に 3 つの領域に分割されており，P[0]～P[2]の 32bit の整数 3 つで表現されている．また，マス内の番号は bit の位置を表している．

(b) 芝浦将棋 Jr. における bitboard 構造：
盤面は縦方向に 3 つの領域に分割されており，P[0]～P[2]の 64bit 整数 3 つで表現されている．また，マス内の数字は bit 位置を表している．

図 1　bitboard のデータ構造：(a) Bonanza，(b) 芝浦将棋 Jr.

そのため，Bonanza では盤面を回転させる事によって関係する bit 列を一つの変数に収めてから取り出す”Rotated bitboard”という手法を使っていました．しかし，Rotated bitboard では，同時に一方向ずつしか利きを算出できません．また，複数枚の回転盤面を管理する必要がありました．

２）Magic bitboard の使用

Magic bitboard は Rotated bitBoard とは異なり，盤面の回転を行わずに bit 列を取り出すことができます．もともとはコンピュータチェスで使用されていた技術です．芝浦将棋 Jr.でもこの方法を使って飛び駒の利きを算出しています．この方法では盤面を表現した bit 列に Magic number と言われる整数を掛けることにより，利きの算出に必要なビットパターンを最上位に集めて，一度の bit 演算で処理を済ませることができます．そのため一度に 2 方向の利きを求めることができます．例として，図 2 に角の利き場所での駒の配置情報の計算法を示しました．

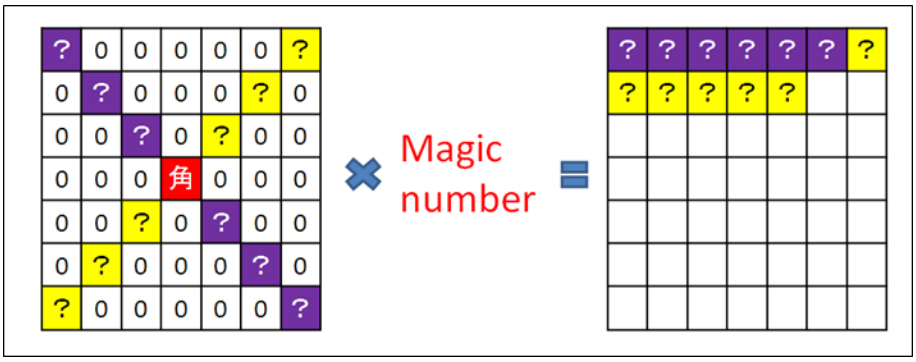


図２　Magic bitboard を用いた角の利き場所での駒の配置情報の計算例：bit 列に Magic number を掛けて, 参照対象の bit 列を上位に集める. その後は, 右シフトによって必要な bit 列を取り出す.

将棋でも山本一成さんの Ponanza がこの Magic bitboard の手法を採用しています[3]. しかし, そこでは盤面を上 6 段, 下 3 段に分けて２つの 64bit 変数で表現しています. そのため, Magic number も２つ用意し，それぞれに Magic number を掛けた後，その結果を XOR 演算により統合する必要がありました. 芝浦将棋 Jr.では図 1 b のような盤面の表現を採用していますから，殆どの場合，中央の変数 P[1]だけに Magic number を掛けて shift 処理をするだけで済んでしまいます．ただし，飛車が盤面の両端の列に居るときだけは別の処理が加わります．つまり，図 1b のような bitnoard 構造を採用することにより，盤面サイズが 9×9 の将棋であっても 8×8 のチェスの場合と同じように，1 変数についての処理で殆ど済ませることができます.

３）序盤定跡の不使用

大会対局時には定跡データベースを使用しません．これは**2**. で述べた開発コンセプトによるものです．もし，従来定跡になかった新手らしきものが得られるようであれば，コンピュータ将棋の一つの成果と言えるのではないかと期待しています.

４）評価関数について

現在のところ，評価関数の特徴量は，大会公式ライブラリである Bonanza (Ver. 6.0.0) のものをそのまま使用しています．しかし，特徴量の重み係数については，2012 年の「芝浦将棋」の値を使用しています．すなわち，駒の重みについては Bonanza (Ver. 6.0.0)のものをそのまま使用していますが，それ以外の重み係数については，強化学習の一種である最急降下 TDleaf(λ)法で強化してあります．詳細については 2012 年の芝浦将棋のアピール文書[1]をご参照願います.

昨年(2015 年)の芝浦将棋 Jr.は，この評価関数の計算処理を局面間の駒配置の差分から評価値の差分計算を行うことにより高速化しました．この差分計算プログラムを独自の性能

評価実験を行って評価したところ，評価関数の計算時間は差分計算を行わない従来の芝浦将棋 Jr.と比べて約 10 分の 1 にまで減少しました．この結果，全体の探索速度は約 2.3 倍に高速化されました．勝率も従来プログラムに対して約 72％と大きく勝ち越しました．

現在は，方策勾配法と呼ばれる強化学習をベースに考案された PGLeaf 法[8]や，深い探索結果を教師として学習する Tree Strap 法などの学習法をいろいろ試しているところです．

５）探索法について

2013 年のバージョンでは単純な *αβ* 探索しか実装していませんでした．しかし，2014 年のバージョンからは以下の処理を追加しています．

①*αβ* 探索の高速化：

・局面表(トランスポジションテーブル)の利用

・キラー手の利用

・上記両者を利用したムーブオーダリング（候補手の並べ替え）の実行

②前向き枝刈り機能の追加：

・Late Move Reductions(LMR)の実装

2015 年は，LMR に加えて，

・Futility Pruning の実装

を行いました．この実装には Bonanza の実装法を参考にしましたが，そこで用いられていたマージンの定数を手動で調整しました．その結果，約 1.2 倍探索が高速化され，勝率も実装前のプログラムに対して約 58％と向上しました．

③静止探索による深さ拡張：

・静止探索の実装･･･20 手まで

・Static Exchange Evaluation(SEE)の実装による静止探索の高速化

④時間制御機能：

・時間制御処理･･･探索時間は 1 手 20 秒前後に固定してあります．

・相手手番中の先読み処理の強化

2015 年は，相手の手番中にも探索を行う先読み機能を強化しました．従来の芝浦将棋 Jr.では，先読み中の探索に時間や深さの制限を設けていました．今年のプログラムはそれを撤廃し，相手の考慮時間を最大限に利用できるようにしました．また，先読みがヒットした場合にもすぐに指すのではなく，一定の時間，さらに探索を行うようにしてあります．

６）探索処理の並列化について

今年(2016 年)の芝浦将棋 Jr.では，探索処理に YBWC(Young Brothers Wait Concept)アルゴリズム[9]を実装し，探索の高速化を図りました．この並列化にあたり，チェスの公開プロ

グラムである Stockfish[10]を参考にしました．スレッドの制御方法やソースコードの可読性などを考慮するとともに，評価関数の差分計算や前向き枝刈り手法をスレッド間の競合が起こらないように調整した結果，探索プログラムの約 40%を書き換えることになりました．

現在，従来プログラムにあったすべての枝刈り処理を並列化プログラムへ実装することはまだ出来ていませんが，従来版と比べて約 2 倍の探索速度を得て，さらに 2 手ぐらい深く読めるようになっています．大会までには実装を完成させ，昨年のプログラムの 4 倍ぐらいの探索速度を出したいと考えています．

5. 今大会での目標

昨年(2015 年)の選手権大会では 21 位でしたので，今年は 10 位台への復帰を目標にしております．本年の大会には強い公開ソフトをベースにしたチームが 1 次予選からも多数出場しそうなので，なかなか難しいとは思いますが，精一杯頑張るつもりです．

6. 将来計画

現在の延長で，探索の並列処理化や，さらなる前向き枝刈り処理の実装を行う予定です．一方，局面評価関数については独自の機械学習により構築したいと考えています．この学習には強化学習を適用したいと考えています．ただし，コンピュータチェスや芝浦将棋で行われていた TD(λ)法や TDLeaf(λ)法ではなく，方策勾配法と呼ばれている強化学習の適用を試みたいと考えています．このアイデアは，すでに 2012 年のゲームプログラミングワークショップ(GPW2012) [4]で研究発表しています．

また，局面評価関数以外にも，探索時に用いるシミュレーション方策の学習もこの方策勾配法を用いて学習することができます．したがって，モンテカルロ探索への応用や，探索木中のノードへの遷移確率値の計算を通して読みの深さの制御に利用することができます．さらには，ある局面で特定の正解手を指すような方策の教師付き学習に対しても，同様な方策勾配の計算により行うことができます．これらのアイデアについても，2013 年の情報処理学会のゲーム情報学研究発表会[5]や 2014 年のワークショップ(GPW2014)[6]で発表しております．

昨年(2015 年)は，上記のような強化学習も含めて，プロ棋士の棋譜データベースを使用することなく，プログラムが自らの対局を通して局面評価関数を学習する方法についての考察をゲーム情報学研究発表会で報告しました[11]．探索法についても，定番となっている min-max 法(α β法)と反復深化に基づく方式ではなく，soft-max 法と累積遷移確率とをベースにした並列処理向きのマルチエージェントシステム的な探索法も検討中です．今後は，これらの手法の実装や学習実験を行っていく予定です．

7. おわりに

本チームの前身である「芝浦将棋」の生い立ちや，2010 年の選手権大会への初参加の様

子について，コンピュータ将棋協会の会誌の中でまとめさせて頂きました[7]．ご興味がある方はそちらも併せてご覧頂ければ参考になるかと思います．

最後になりましたが，「芝浦将棋 Jr.」の開発コンセプト，方向性にご賛同頂ける方であれば，学生，社会人の如何を問わず，どなたでも歓迎いたします．実際にメーリングリストを開設したり，本学豊洲キャンパスにおいて討論を行うなど，他チームの開発者の方とも活発に交流しています．共同研究やチーム開発に参加，協力をご希望の方，あるいは討論や意見交換をご希望の方は， arashi50@sic.shibaura-it.ac.jp までご連絡下さい．よろしくお願いします．

参考文献

[1] 五十嵐 治一，山本一将，川内博世，濱村綾，“「芝浦将棋」のチーム紹介”，第 22 回世界コンピュータ選手権向け「芝浦将棋」アピール文書，http://www.computer-shogi.org/wcsc22/appeal/Shibaura_Shougi/appeal.pdf

[2] Bonanza のホームページ，http://www.geocities.jp/bonanza_shogi/

[3] 山本一成，”コンピュータ将棋における Magic Bitboard の提案と実装”，第 15 回ゲームプログラミングワークショップ (GPW2010)予稿集, pp.42-48 (2010).

[4] 五十嵐治一，森岡祐一，山本一将，“方策勾配法による静的局面評価関数の強化学習についての一考察”，第 17 回ゲームプログラミングワークショップ(GPW2012)予稿集, pp.118-121 (2012).

[5] 五十嵐治一，森岡祐一，山本一将，“方策勾配法による局面評価関数とシミュレーション方策の学習”, 第 30 回ゲーム情報学研究発表会, 情報処理学会研究報告, Vol. 2013-GI-30, No. 6, pp.1-8 (2013).

[6] 五十嵐治一，森岡祐一，山本一将，“方策勾配法による探索制御の一考察”，第 19 回ゲーム・プログラミングワークショップ(GPW2014)予稿集，pp.90-94 (2014),

[7] 五十嵐治一，”教育・研究プロジェクト「芝浦将棋」の展望“，コンピュータ将棋協会誌，Vol.22, pp.35-47（2011 年 4 月発行）.

[8] 森岡祐一，五十嵐治一，“方策勾配法とαβ探索を組み合わせた強化学習アルゴリズムの提案”，第 17 回ゲーム・プログラミング・ワークショップ(GPW2012) 予稿集，pp.122-125(2012)

[9] 岸本章宏、柴原一友、鈴木豪，”ゲーム計算メカニズム―将棋・囲碁・オセロ・チェスのプログラムはどう動く―“, 小谷善行 編著,「コンピュータ数学シリーズ⑦」, P81-P85, コロナ社, 2010.

[10] Stockfish のホームページ，https://stockfishchess.org/

[11] 五十嵐治一，森岡祐一，山本一将，”プロ棋士の棋譜データベースを用いない局面評価関数の学習法についての考察”， 第 34 回ゲーム情報学研究発表会，情報処理学会研究報告，Vol. 2015-GI-34, No. 4, pp.1-8 (2015)