

実現確率探索のゲーム全般への応用 – Lines of Action を題材にして –

橋本 剛¹, 長嶋 淳², 作田 誠², Jos Uiterwijk³, 飯田弘之^{2,4}

¹ 静岡大学工学部 ² 静岡大学情報学部

³ Department of Computer Science, Universiteit Maastricht

⁴ 科学技術振興事業団さきがけ研究 21 「機能と構成」領域

E-mail: {hasimoto,cs8066,sakuta}@cs.inf.shizuoka.ac.jp,

uiterwijk@cs.unimaas.nl, iida@cs.inf.shizuoka.ac.jp

概要

鶴岡は実現確率探索を考案し、プログラム「激指」に実装して2002年世界コンピュータ将棋選手権を見事に制した。実現確率探索は局面の実現確率を閾値として探索することで、より「有りそう」な手の場合ほどどんどん深く読み、「無さそう」な手の場合はほとんど読まない。これまで前向き枝刈りと探索延長によって行ってきたことをシステムティックに非常に簡単なアルゴリズムで行う優れた探索法であるといえる。ところが、激指の手法ではプロの棋譜から指し手の確率を計算しており、参考となる棋譜が十分に存在しない多くのゲームではこの手法は通用しない。本稿ではコンピュータの自動学習により指し手の確率を自動で計算し、実現確率探索をあらゆるゲームに応用させる手法、ARPS（自動実現確率探索）を提案する。実験には連結型の二人ゲーム Lines of Action (LOA) を選び13種類の分類を与えてARPSを実装し分類の確率を計算した。一般的な反復深化探索との対戦結果は、ARPSが反復深化に比べてより深くよりより絞り込んだ読みで大きく勝ち越しその優位性を示した。

Application of Realization Probability Search for Any Games – a case study using Lines of Action –

Tsuyoshi Hashimoto¹, Jun Nagashima², Makoto Sakuta²,
Jos Uiterwijk³ and Hiroyuki Iida^{2,4}

¹ Faculty of Engineering, ² Faculty of Information, Shizuoka University

³ Department of Computer Science, Universiteit Maastricht

⁴ "Information and Systems", PRESTO, Japan Science and Technology Corporation

abstract

Recently, Realization-Probability Search was proposed and implemented in the shogi program GEKISASHI, which won the World Computer Shogi Championship in 2002. It searches deeper after "probable" moves and shallower after "improbable" moves. It can efficiently perform a selective search similar to forward pruning and search extensions, while using a considerably simpler algorithm in a systematic way. However, since the method in GEKISASHI calculates move-category probabilities from many games by professional players, it cannot be used for games that lack the availability of a sufficient quantity of game records. This paper proposes a technique called Automatic Realization-Probability Search (ARPS), which can be applied to any game. We have selected Lines of Action (LOA), which is a connection-based two-player game and is considered very different from chess-like games. ARPS has been implemented and examined in a LOA program. The results of our experiments have shown that ARPS is superior to normal iterative deepening in strength and performs narrower and deeper searches.

1 はじめに

将棋では合法手が非常に多いため、これまでは各プログラマーが独自にさまざまな前向き枝刈りや探索延長の手法を用いていた。だがそのやり方はプログラマーの手作業に依存する事が多いのが現状で、複雑でわかりにくいものになりやすいという大きな欠点がある。これに対して、鶴岡はプロの棋譜から抽出した指し手の確率を元に局面の実現確率という値を計算しこれを閾値に用いる実現確率探索を考案し、この手法を実装したプログラム「激指」で2002年世界コンピュータ将棋選手権を見事に制した。このように実現確率探索が優秀なアルゴリズムである事は疑いが無いが、この手法を将棋以外のゲームで使う場合に一つ重大な問題がある。激指の手法ではプロの棋譜から指し手の確率を計算しており、将棋のようにすでに強いプレイヤーがいて棋譜が十分にある場合は問題がないが、参考となる棋譜が十分に存在しない多くのゲームではこの手法は通用しない。本稿ではこの実現確率探索を将棋のように強いお手本があるゲームだけでなく、あらゆるゲームに対応させる方法を提案し、連結型の二人ゲームで将棋種とはかなり性質の違う Lines of Action (LOA) を用いてその優秀性を明らかにしていく。

2 実現確率探索とは

鶴岡 [1] の提案した実現確率探索について説明する。まず指し手を人為的に分類し、(王手, 得する駒取り, 両取り等) この各指し手をプロ棋士の棋譜 ([1] では羽生の棋譜 600 局) を元に統計を取り、指し手の確率を以下のように計算する。「指し手の確率 = 指し手を選んだ数 / 指し手が存在した数」探索ではこの指し手の確率を元に局面の実現確率を以下のように計算する。

(局面の実現確率) = (直前の局面の実現確率) × (指し手の確率)

実現確率探索では閾値として深さの代わりにこの局面の実現確率を用い、それ以外の点は普通の深さ打ち切りによる探索アルゴリズムと何ら代わりはない。だが (局面の実現確率) を閾値として探索することで、より「有りそう」な手の場合ほどどんどん深く読み、「無さそう」な手の場合はほとんど読まない、というこれまで前向き枝刈りと探索延長によって行っ

てきたことが非常に簡単なアルゴリズムでシステムティックに行うことができるのである。

3 他のゲームでの問題点

「激指」が2002年世界コンピュータ将棋選手権を制したことから、実現確率探索は将棋に関しては明らかに優れた手法であるといえるし、他の多くのゲームでも非常に有効に働くと予想される。ところが、ここで一つ重要な問題点がある。将棋のようにプロ棋士が大勢いて優れた棋譜が大量にあるゲームでは問題がないが、LOAのように多くのゲームでは手本となるべき棋譜は残念ながらほとんど存在しない。そのため、この優れた手法を他のゲームに応用するには指し手の確率を計算する新たな手法が必要になる。そこで、我々はコンピュータの自動学習によって指し手の確率を計算させあらゆるゲームに実現確率探索を応用する方法を提案する。

4 自動実現確率探索 (ARPS)

コンピュータの自動学習によって指し手の確率を計算させる新しい実現確率探索の手法、自動実現確率探索 (Automatic Realization-Probability Search, または ARPS) をここに提案する。確率を自動で計算するので、鶴岡の提案した実現確率探索と違いプロ棋士などの優れた棋譜が存在しないゲームにも応用できる。話は簡単で、まず指し手の分類を行い、これと実現確率探索のアルゴリズム ([1] または [2] 参照) を実装する。ここで、各指し手種類の確率は同じ値にしておく。

次にコンピュータに自動対戦をさせ、探索を行うたびにルート局面で「指し手が存在した数」と「指し手を選んだ数」をカウントする。一定数対局をしたら、ここで得た値を元に各指し手の確率を計算する。この値を元に実現確率探索を行いまた「指し手が存在した数」と「指し手を選んだ数」をカウントし、これを繰り返していく。ここで注意しないといけないのは、指し手種類の確率が0だとその手は絶対に選ばれないので、初期値として「指し手が存在した数」と「指し手を選んだ数」に0でない数を入れておかないといけない。各分類の確率が安定するまでこの手順を十分繰り返すことによって各指し手分類の確率を得る。値を得られれば、その値を利用して実現確

率探索を行える。

5 LOA への応用

本手法の効果を調べるため、連結型の二人ゲームで将棋種とはかなり性質の違う Lines of Action (LOA) を題材とし、我々の LOA プログラム (T-T) (うるうる) へ実現確率探索を実装した。LOA を選んだ理由は、数年前から Computer Olympiad [3] の競技種目になっており性能を試しやすいという事、オセロのように一定の手数で必ず終わることがなく将棋のように終局条件が満たされるまで続くゲームであり実現確率探索の効果が期待される、といった点である。まずは LOA のルールを紹介する。

5.1 LOA のルール

LOA は 8x8 の盤を使い黒と白の二人のプレイヤーで行う二人零和完全情報ゲームである。以下 Mark Winands のページ [4] を元に LOA のルールを示す。

1. 初期配置は図 1 のように黒は盤の上辺と下辺にそれぞれ 6 個、白は盤の右辺と左辺にそれぞれ 6 個並べる。
2. 黒番からスタートし、黒白交互に一手ずつ指す
3. プレイヤーは必ずどれか一つ駒を動かさないといけない。動かす駒は動く方向のライン上にあるすべての駒の数と同じ数だけマスを動かさないといけない。
4. 味方の駒は飛び越えることができる。
5. 敵の駒は飛び越えられないが、敵の駒のあるマスに動けるときはその駒を取ることができる。
6. ゲームの目的は盤上の自分の駒をすべてつなげることである。つなぎ方は縦横斜めのいずれかの方向でもよく、先につなげたほうが勝ちとなる。図 2 は黒の駒がすべてつながっており、黒の勝ちである。
7. どちらかのプレイヤーの駒が一つになったら勝ちである。
8. 合法手がない場合は負け
9. 同時に両プレイヤーの駒がつながった場合は引き分け。(2002 年度より Computer Olympiad で採用されたルール。それ以前は手番側の勝ちだった)
10. 同一局面 3 回で引き分け

駒の動き方の例を図 3 に示す。f4 の白駒を使って説明する。f4 の左上と右下には 2 つ駒があり 2 マス動けるので、d6 と h2 に動ける。同様に d4 にも動けるが、h6 は自分の駒があり行けない。上下方向には駒が 4 つあるが、上も下も 4 マス以内に敵の駒があるため動けない。左右方向では駒が 3 個あるので、c4 と黒駒を取って移動することができる。右へは 3 マス進むと盤外になるので動けない。

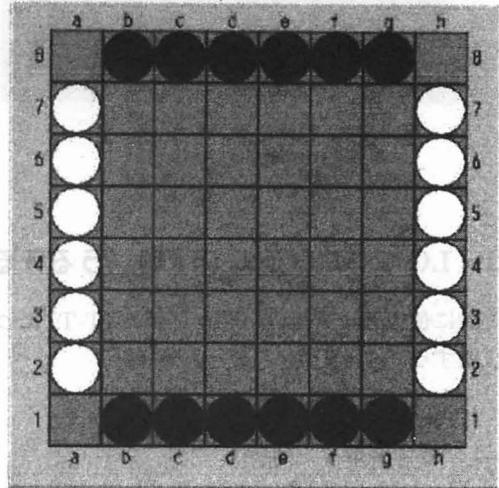


図 1: LOA 初期局面

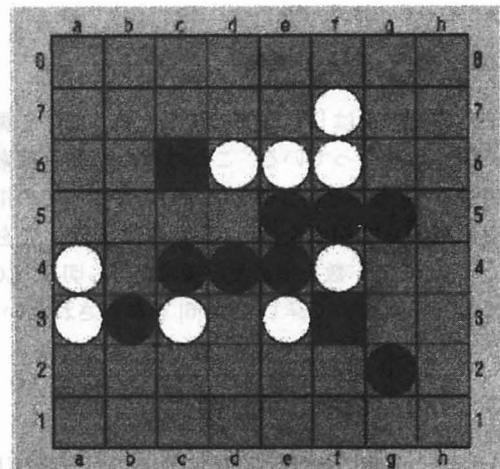


図 2: 終了局面例

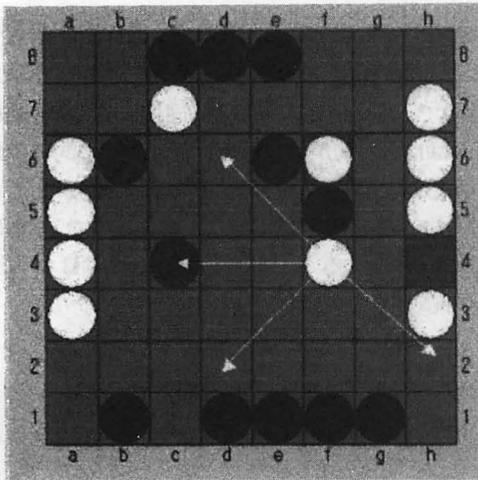


図 3: 駒の動き方例

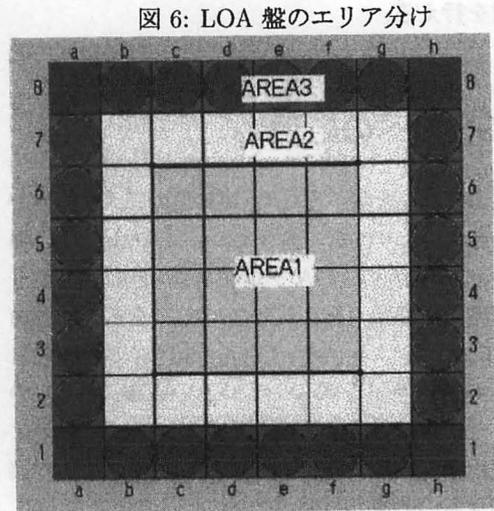


図 6: LOA 盤のエリア分け

5.2 LOA プログラム (T-T) (うるうる)

実験に使う我々の LOA プログラム (T-T) について説明する。

5.2.1 評価関数

評価関数はまず自分の駒すべての重心を計算し、重心に近い駒ほど高い点数を与えている。重心から 3 マス以上離れているとマイナスの評価値を与えており、重心から遠い駒はない方がいいと判断させている。それ以外では完全に勝ちまたは負けのときだけその評価をしている。

5.2.2 終盤 (詰み) 探索

通常の探索をはじめの前に勝ちと負けだけを調べる終盤探索を行っている。これは将棋の詰み探索に相当するが、王手という概念はないのですべての指し手を候補として探索を行う。探索には PDS [5] を使い、一定のノード数を超えたら探索を打ち切る。LOA の終盤探索について詳しくは [6] を参照されたい。

5.3 LOA の指し手分類

ARPS の実装にあたって問題になるのが分類の仕方であるが、ここでは盤を図 6 のように 3 つのエリアに分け、各エリア間での移動をそれぞれ分類し (1TO1-3TO3)、他には敵の駒を取る手 (TORI1, TORI2and3

とエリアによって分ける)、直前に動いた駒を取る手 (CHOKUTORI1, CHOKUTORI2and3) の計 13 種類に分類した。

自動対戦を 100 回行うことで各分類の「指し手が存在した数」と「指し手を選んだ数」をカウントし確率を計算した。なお、一つの局面で同じ分類の指し手が複数存在する場合は「指し手が存在した数」を 1 として計算した。図 4 に計算した確率の値を示す。ここで直前に動いた駒を取る手は C-TORI と書いている。中央 (AREA1) で駒を取る手 (C-TORI1, TORI1) が特に高い確率になっている。他に AREA1 に行く手は比較的高い。

6 実験

(T-T) に自動対戦により得た確率を用いる実現確率探索 (Realization-Probability Search, RP) と、単純な深さ打ち切りの反復深化 (Iterative Deepening search, ID) の二つを実装し毎回ハッシュに 10 万ノード登録されたら読みを打ち切るという条件で先後入れ替えて 50 回づつ対戦を行った。

結果は RP 先手で RP の 43 勝 7 敗, RP 後手で RP の 32 勝 18 敗といずれも RP が大きく勝ち越し、本手法が単純な深さ打ち切りの反復深化に比べて優秀であることが示せた。次にある局面での RP と ID それぞれの探索ノード数を図 5 に示す。[1] と同様に実現確率探索の特徴的な分布になっており、ID より効率的に探索を行っているのがわかる。

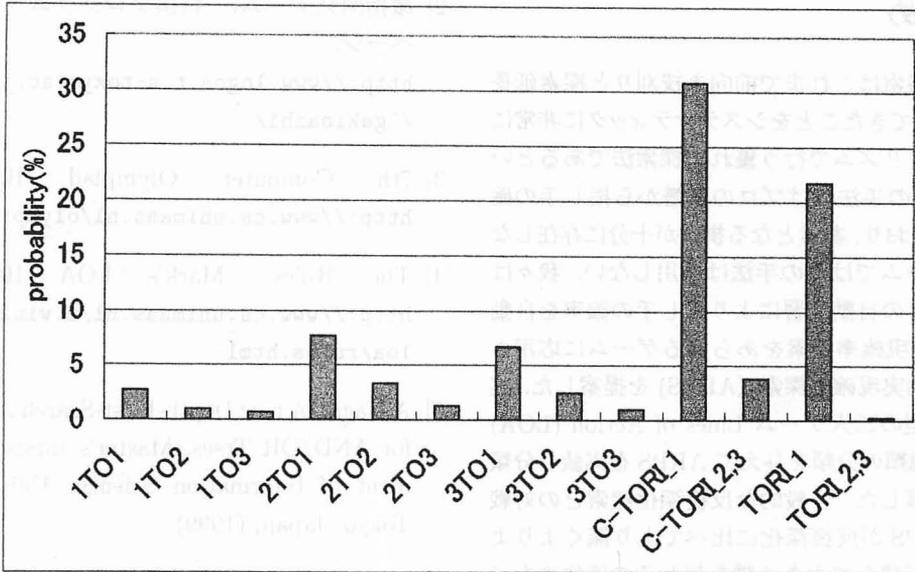


図 4: 各指し手分類の確率

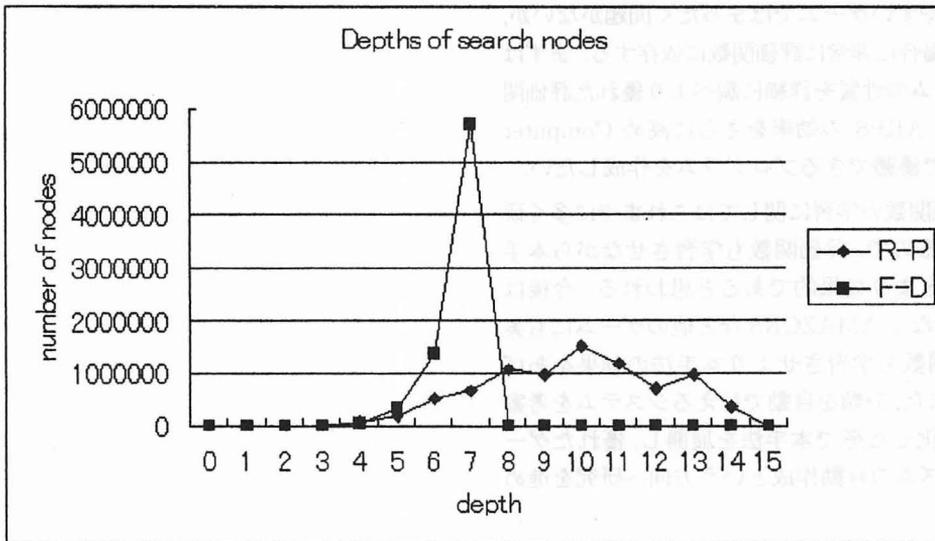


図 5: 探索ノード数の比較

7 まとめ

実現確率探索はこれまで前向き枝刈りと探索延長によって行ってきたことをシステマティックに非常に簡単なアルゴリズムで行う優れた探索法であるといえるが、激指の手法ではプロの棋譜から指し手の確率を計算しており、参考となる棋譜が十分に存在しない多くのゲームではこの手法は通用しない。我々はコンピュータの自動学習により指し手の確率を自動で計算し、実現確率探索をあらゆるゲームに応用させる手法自動実現確率探索 (ARPS) を提案した。実験には連結型の二人ゲーム Lines of Action (LOA) を選び、13 種類の分類を与えて ARPS を実装し分類の確率を計算した。一般的な反復深化探索との対戦結果は、ARPS が反復深化に比べてより深くよりより絞り込んだ読みで大きく勝ち越しその優位性を示した。

8 Future work

本手法はプログラムに探索を行わせ自身が選んだ指し手によって指し手確率を計算するため、答えがはっきり出やすいゲームではまったく問題がないが、そうでない場合は非常に評価関数に依存する。まずは LOA のゲームの性質を詳細に調べより優れた評価関数を作成し、ARPS の効率をさらに高め Computer Olympiad で優勝できるプログラムを作成したい。

また、評価関数の学習に関してはこれまでに多く研究されているので、評価関数も学習させながら本手法を用いるとより効果的であると思われる。今後は LOA だけでなく AMAZONS など他のゲームにも実装し、評価関数も学習させより本手法の効果をあげてみたい。また、分類を自動で行えるシステムを考案しより一般化した形で本手法を展開し、優れたゲームアルゴリズムの自動作成という方向へ研究を進めていきたい。

参考文献

- [1] 鶴岡慶雄, 横山大作, 丸山孝志, 近山隆. 局面の実現確率に基づくゲーム木探索アルゴリズム. *The 6th Game Programming Workshop*, pp.17-24, 2001.
- [2] 激指開発チーム. 将棋プログラム「激指」のページ.
<http://www.logos.t.u-tokyo.ac.jp/~gekisashi/>
- [3] 7th Computer Olympiad Homepage,
<http://www.cs.unimaas.nl/olympiad2002/>
- [4] The Rules, Mark's LOA Homepage,
<http://www.cs.unimaas.nl/m.winands/loa/rules.html>
- [5] A.Nagai, A new Depth-First-Search Algorithm for AND/OR Trees. Master's thesis, Department of Information Science, University of Tokyo, Japan, (1999).
- [6] 作田誠, 長嶋淳, 橋本剛, 飯田弘之. 将棋終盤で発展してきた探索手法の Lines of Action への適用. *情報処理学会論文誌*, Vol. 43, No. 10, 2002. in print.