F-009

囲碁棋譜における Suffix Array を用いた着手記号列のインデックス手法

An Efficient Indexing Method for Go Game Records based on Suffix Array

加藤 雄大† 白松 俊† 大囿 忠親† 新谷 虎松†
Yudai Kato Shun Shiramatsu Tadachika Ozono Toramatsu Shintani

1. はじめに

本研究では、囲碁棋譜における手順による検索の高速化を目標としている．囲碁では配石状態が移動，回転，鏡像関係にある配石状態を同じものとしてみなすため，この特性を考慮しつつ高速に検索することが課題である．

囲碁や将棋，チェスなどのボードゲームでは定石が存在し，定石を使えばある程度着手の選択肢が狭まる．このように囲碁を学ぶ上で定石，布石，手筋などの手順を覚えることは不可欠となる．手順を覚える手段は定石書を読む，プロの実戦の棋譜を見る，棋譜データベースで調べることがある．定石書やプロの棋譜を用いる手段ではパターンを覚えることは可能だが，囲碁学習者が知りたい手順を直ちに知ることはできない．また，「囲碁の棋譜でーたべーす[1]」のような棋譜データベースは手順によって棋譜を検索することができるが，問題点として初手からの手順しか検索できない点が挙げられる．囲碁学習者にとって棋譜の途中の手順であっても検索できることが望ましい．

本稿では，手順による棋譜検索のための Suffix Array[1](以降，SA)に基づくインデックス手法を提案する．本手法を用いて 42,784 局，約 1,145 万手のプロの棋譜から指定した手順を検索するためのシステムを実装した．評価実験より定石や布石を，0.1ms 程度で検索可能であることを示す．

2. 着手の符号化方法および Suffix Array の生成

本符号化方法では複数の棋譜に対して着手の座標列を着手記号列に符号化する．ここで，座標列とは 2 次元座標の系列とする．例えば，図 1 の移動の碁盤において座標列は番号順に(4,17), (3,17), (3,16), (2,17)となる．囲碁における棋譜は着手の座標列として表現できる．また，着手記号列とは，座標列を符号化した文字列のことである．次に，着手記号

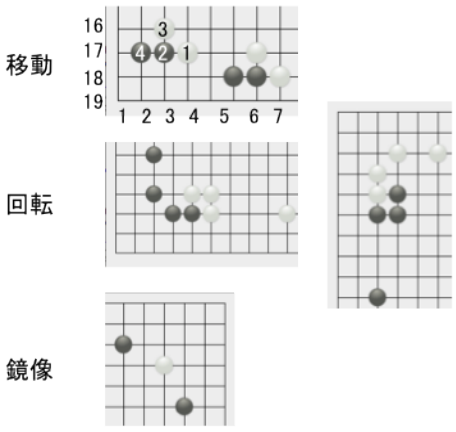

図 1:移動，回転，鏡像の例

Algorithm 1 長さ n の座標列 A から着手記号列 S への符号化

1: **for** $i = 1 \rightarrow n-1$ **do**
2: $(x_i, y_i) \Leftarrow A[i]$
3: $(x_{i+1}, y_{i+1}) \Leftarrow A[i+1]$
4: // 移動
5: $(r_{xi}, r_{yi}) \Leftarrow (x_{i+1} - x_i, y_{i+1} - y_i)$
6: // 回転
7: $t \Leftarrow (x_i, y_i)$ の領域番号 (図 3 参照)
8: $\theta \Leftarrow \pi/2 \times \lfloor (t-1)/2 \rfloor$
9: $r_{xi} \Leftarrow r_{xi}\cos\theta - r_{yi}\sin\theta$
10: $r_{yi} \Leftarrow r_{xi}\sin\theta + r_{yi}\cos\theta$
11: // 鏡像
12: **if** t は偶数 **then**
13: $(r_{xi}, r_{yi}) \Leftarrow (r_{yi}, r_{xi})$
14: **end if**
15: $S[i] \Leftarrow 37 \times r_{xi} + r_{yi}$
16: **end for**
17: **return** S

図 2:長さ n の座標列 A から着手記号列 S への符号化のアルゴリズム

列を自然言語の文字列と同様に扱い，符号化された記号列から SA を作成する．SA を用いることで高速に検索でき，頻出する手順の列挙を容易にする．

本符号化アルゴリズムは全ての着手を図 3 に示す，第 1 領域に写像することで移動，回転，鏡像関係にある配石状態を同一の符号に変換する．図 1 に移動，回転，鏡像の具体例を示す．移動の関係とは配石状態を平行移動して同一の配石状態となる関係である．回転の関係とは天元を中心に 0, 90, 180, 270 度回転して同一となる関係である．鏡像の関係とは碁盤の対角線に線対称な配石状態となる関係である．本稿の符号化方法は，文献[2]で提案されている相対符号化法に基づいている．

ある棋譜を表す長さ n の座標列 A から着手記号列 S への符号化のアルゴリズムを図 2 に示す．2，3 行目で A の i, i+1 番目の座標を取り出す．5 行目で i 番目と i+1 番目の座標の相対座標を求める．7〜10 行目は領域 1 に写像するための回転処理を行う．領域は図 3 に示すように 8 つに分けられている．これを用いて i 番目の座標(x_i, y_i)の領域番号 t を計算する．t より回転する角度 θ を求め回転行列を用いて相対座標を回転させる．12〜14 行目では鏡像の処理を行う．領域番号 t が偶数であれば相対座標の x 座標と y 座標を入れ替える．そして，求めた相対座標(r_{xi}, r_{yi})を着手記号に変換する．ここで，$-18 \leqq r_{xi}, r_{yi} \leqq 18$ なので値 $37 \times r_{xi} + r_{yi}$ を

† 名古屋工業大学工学部, Nagoya Institute of Technology

1 YODA, 囲碁の棋譜でーたべーす, http://wiki.optus.nu/igo/

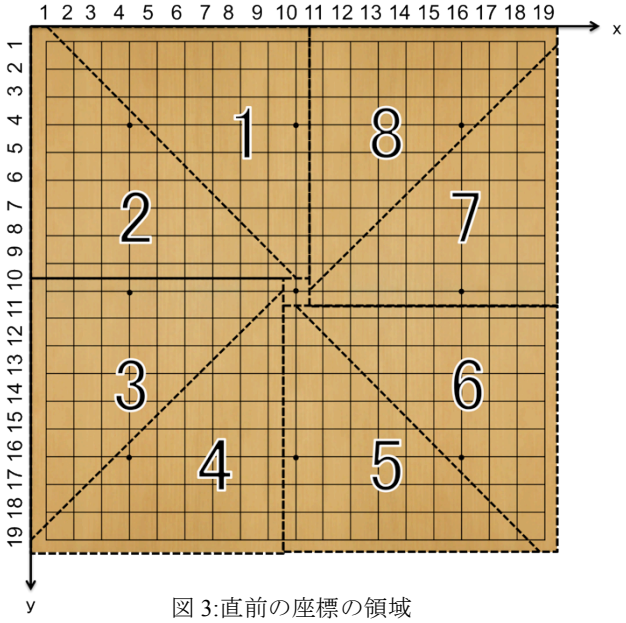

図 3:直前の座標の領域

もつ記号，つまり，2 桁の 37 進数表現の記号として座標を符号化する．

ここで，連続する 2 つの座標から 1 つの着手記号を生成する例を示す．直前座標(4,17)，現在の座標(3,15)とする．相対座標は(-1,-2)となる．直前の座標(4,17)は領域 4 に属するので(-1,-2)を$\pi/2$ 回転すると(2,-1)となる．(4,17)は偶数番目の領域に属するので，(2,-1)の x 軸 y 軸を入れ替え(-1,2)となる．よって，値-35 をもつ着手記号が生成される．

上記の方法で全ての棋譜を符号化し，着手記号列を生成する．得られた全ての着手記号列を“$”で連結する．“$”は棋譜を連結するための区切り文字である．この連結した着手記号列の SA を作成する．

3. 実験

SA による棋譜検索の検索時間を評価するために単純な文字列検索手法(以降，SS と略す)と検索時間を比較した．SS は全ての棋譜の着手記号列に対して，Java の String クラスの indexOf メソッドを使用した．

各検索手法に対して，棋譜数 n と検索時間の関係を調べた．n は 2,000 局から 42,000 局まで 2,000 ずつ増加させた．検索クエリはツケオサエ定石，ツケノビ定石，三々カタツキ定石，三連星，秀策流を符号化した着手記号列とした．これらの検索クエリの長さは 6～10 である．各検索手法において検索クエリとマッチするすべての部分着手記号列を列挙する．

実験環境は CPU：2.7GHz，メモリ：4GB，OS：Mac OS X Lion であり，実装には Java 言語を使用した．42,000 局分の SA のメモリ使用量は 258MB であった．

実験結果を図 4 に示す．SA において棋譜数が 42,000 局であっても検索時間は 0.1ms 程度である．SA による棋譜検索は棋譜数が増加するにつれ検索時間も増加するが SS と比べると検索時間の増加量が少ないことがわかる．SS による棋譜検索は棋譜数が増加するにつれ検索時間が大きく増加しており，SA による棋譜検索の優位性が示された．

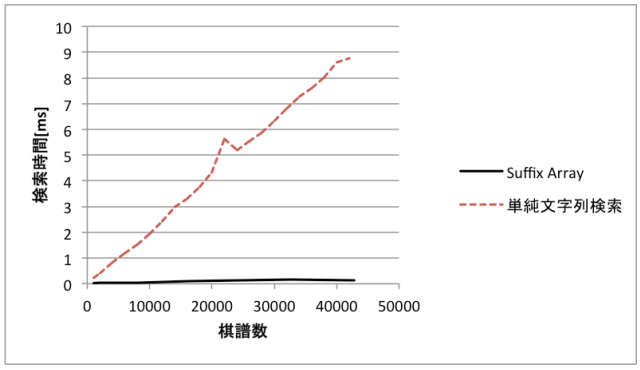

図 4:Suffix Array と単純な文字列検索の棋譜検索時間の比較

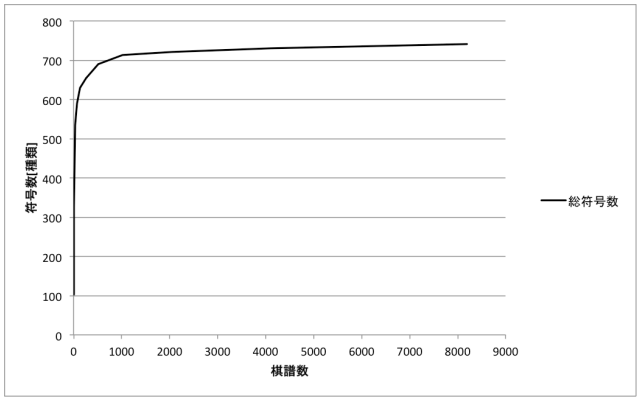

図 5: 棋譜数に対する総符号数

手順による棋譜検索の結果を人手で確認すると全く同一の配石状態だけでなく，移動，回転，鏡像の関係にあるものも検索できることを検証した．

図 5 に総符号数を示す．総符号数は符合する過程で生成された着手記号の種類である．図 5 の横軸は棋譜の数であり，棋譜の数に対して生成された着手記号の種類を示している．図 5 より，総符号数は約 1,000 局まで急激に増加しそれ以降は緩やかに増加している．最終的に 750 種類程度で頭打ちになる．2 節の符号化方法で表現可能な符号数は 37×37 = 1369 であり，750 という符号数は全体の約 54%である．つまり，囲碁において直前の着手と現在の着手の相対座標はある特定の座標に偏って分布することを示している．

4. おわりに

囲碁棋譜における着手の符号化方法と SA のインデックス手法について提案した．本インデックス手法を用いることで棋譜を高速に検索できることを検証した．棋譜検索や棋譜の頻出パターンマイニングにおいて棋譜の数は多いことが望ましい．それを処理するためのメモリ使用量は大きくなり計算時間も大きくなる．3 節の総符号数の実験結果より符号には偏りがあることが確認できたためこれらを利用した符号の圧縮やメモリ使用量の軽減の実現が期待できる．

参考文献

[1] Baeza-Yates, R. and Rieiro-Neto, B.: Modern Information Retrieval, the concepts and technology behind search 2nd Edition(2012).
[2] 中村 貞吾, “着手記号列の出現頻度に基づく囲碁棋譜からの定型手順獲得”, 情報処理学会論文誌, Vol. 43, No. 10 (2002).