Vol.2010-GI-23 No.7
2010/3/8

Nested Monte-Carlo 探索の AMAF を用いた探索数調整による改良

秋山晴彦† 小谷善行††

モンテカルロ探索を入れ子的に呼び出す Nested Monte-Carlo 探索は，Morpion Solitaire というパズルゲームにおいて成果を上げている．入れ子の深さをレベルと呼び，上位レベルの探索では実行時間が指数関数的に増加する．本研究では，AMAF 手法の手数の換算を用いて，擬似的な探索数を維持しつつ探索実行数を削減し，高速化による高レベル化を実現した．この手法により Morpion Solitaire の Touching ルールにおいて，我々はコンピュータ探索の世界記録 145 手を達成した．

Nested Monte-Carlo Search Improvement by Search Number Adjustment using AMAF

Haruhiko Akiyama† and Yoshiyuki Kotani††

Nested Monte-Carlo Search, which calls Monte Carlo search in the nested call, has succeeded in the puzzle game named Morpion Solitaire. The depth for the nest is called a level, and the execution time increases exponentially in the search for meta level. In the present study, the number of search was reduced to maintain a pseudo number of searches by using the conversion of the AMAF technique, and we achieved the high level Nested Monte-Carlo search. Our system generated a new world record 145 moves of the computer search in Morpion Solitaire Touching version by this technique.

1. はじめに

囲碁をはじめとする多くのゲームにおいて，ランダムシミュレーションにより確率的に良い手を求めるモンテカルロ法が成功を収めている．これらのゲームと同様に，膨大な探索空間を持つゲームの一つとして，最大手数を求める一人ゲームである Morpion Solitaire がある．Morpion Solitaire では，対戦ゲームと異なり，対戦相手に対する勝率の良い手ではなく，全探索空間から 1 本の最適経路を求めなくてはならない．ランダムシミュレーションや UCT などの既存手法では良い解が得られにくかった[1]が，Tristan Cazenave の提案した Nested Monte-Carlo 探索[2]は効果を上げている．これはモンテカルロ法における各手の評価をモンテカルロ探索によって行うという手法であり，このモンテカルロ探索を重ねる段階をレベルと呼ぶ．Tristan Cazenave はこの手法のレベル 4 ゲームによって，Morpion Solitaire の Disjoint ルールにおける世界記録を更新した．

Nested Monte-Carlo 探索ではレベルを上げることでより良い手筋が得られる．しかし，より上位レベルのゲームを行うには下位レベルのゲームを多数回行う必要があるため，実行時間が飛躍的に増加する．そのため，世界記録を出したレベル 4 ゲームよりさらに上位レベルのゲームは実行時間が数年，数百年となり，現実時間的に行うことができない．また，より探索空間の広いゲームにおいてはさらに時間増加率が大きくなる．

本研究では，Nested Monte-Carlo 探索の下位ゲーム探索数を削減し，1 ゲームあたりの実行時間を短くすることで，より上位レベルのゲームを実現することを試みる．

2. Morpion Solitaire 及び既存手法

2.1 Morpion Solitaire とは

本研究では，探索手法の実験において，Morpion Solitaire というゲームを題材とする．Morpion Solitaire は，フランスなどヨーロッパにおいて知られている，紙と鉛筆で行えるゲームである．決められたルールに基づき盤面に印を描いていき，最長手数の記録を目指す，一人遊びのゲームである．Morpion Solitaire は，囲碁のような格子状の線が引かれた盤面の上で行われる．ゲームの目的は，決められた初期の盤面から，後述するルールに従って，「できるだけ多くの手を指すこと」である． また，「Touching」「Disjoint」と呼ばれる 2 つのルールがあり，これらは初期局面や基本的なルールは同じであるが，手の制限がわずかに異なる．Morpion Solitaire の基本的な初期盤面を，図 1 に示す．また，基本ルールを以下に示す．

† 東京農工大学工学部情報工学科
Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology
†† 東京農工大学大学院工学府
Graduate School of Technology, Tokyo University of Agriculture and Technology

Vol.2010-GI-23 No.7
2010/3/8

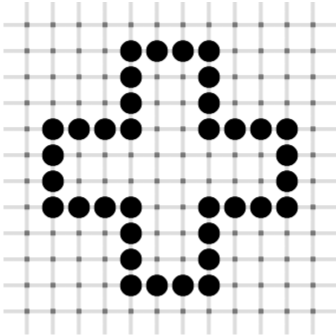

図 1. 基本初期盤面

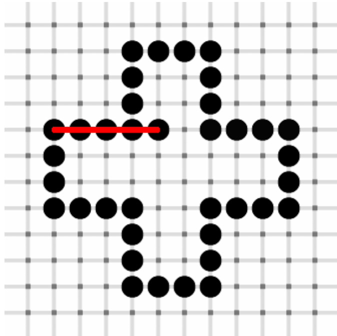

図 2． 1 手打った図

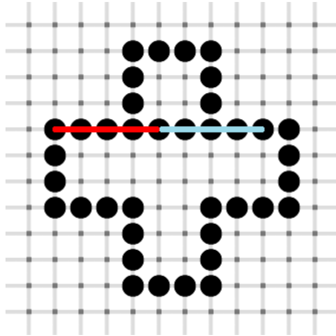

図 3. Disjoint ルールでは打てない手

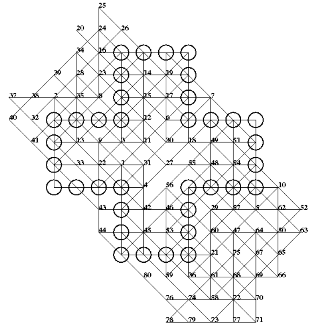

図 4 . Disjoint の世界記録 80 手[2]

[Morpion Solitaire の基本ルール]

- 「一つの駒を新たに格子の交点に置き，1 本の線を引くこと」を 1 手とする（どちらか一方のみを行うことはできない）．
- この「線」とは，連続した五つの駒の上に引かれる線分である（縦横，斜め 45 度 2 種類の 4 方向）．
- 線を引く五つの駒のうちの一つは，その手で置いた駒でなければならない．
- 同方向の線が重なってはならない．

すなわち，五目並べにおいて勝利する手を繰り返すようなゲームである．この基本ルールは，Touching ルールと呼ばれる．このルールに従って 1 手を打った図が，図 2 である．同様にして駒を置いて線を引く 1 手を繰り返してゲームを進行する．このルールを守って線を引けるような駒を置くことができなくなった時点でそのゲームは終了となり，それまでにプレイした手数がそのゲームの記録となる．

Touching ルールにおける人間の手作業による世界記録は，170 手[3]となっている．コンピュータによる探索の世界記録は 144 手[4]であるため，これに及ばない．

また，Touching ルールに以下の制限を付け足したものが，Disjoint ルールである．

- 同方向の線同士の端が接触してはならない．

このルールでは，Touching ルールより可能手が制限されるため，探索空間は狭くなる. Disjoint ルールで不可能な手の例を図 3 に示す. また世界記録 80 手を図 4 に示す.

2.2 Nested Monte-Carlo 探索

Nested Monte-Carlo 探索[2]は，Morpion Solitaire の Disjoint ルールにおいて，図 4 の 80 手の世界記録を達成した手法である．この手法の概要を説明する．

モンテカルロ法は，各局面でランダムシミュレーションを多数回行い各可能手の成果を比較し，最も良い可能手を選択するという手法である．ここで，各局面おいて，各可能手に対して 1 度ずつモンテカルロ法による探索を行い，結果を比較する「メタモンテカルロ法」というものを考える．また，このメタモンテカルロ法による探索結果を用いて可能手を比較するメタメタモンテカルロ法を考える．このように，複数のモンテカルロ探索を入れ子にして，上位のモンテカルロ探索が再帰的に各可能手に対して 1 度ずつ下位のモンテカルロ探索を呼び出して可能手を評価する探索手法が，Nested Monte-Carlo 探索である．この手法において，入れ子の重なる段数を「レベル」と呼び, シミュレーション全体の末端となるランダムプレイアウトをレベル 0 ゲーム，その上位にあるモンテカルロ探索をレベル 1 ゲーム，さらに上位にあるメタモンテカルロ探索をレベル 2 ゲーム，というように，各段階のゲームをレベルで呼び表す．このレベルを上げることで，より精密な探索を行うことができる．

モンテカルロ法は，対戦ゲームにおいては主に勝率により評価を行う手法であるが，Morpion Solitaire では手数の記録がより高いものを評価する．また，局面を進める際，シミュレーション結果の最高手数及びその手筋を常に保持しておき，次の局面でのシミュレーション結果を比較する際に以前の最高記録も比較対象とし，より良い方を残す，という手法を同時に用いることで，上位レベルの探索が確実に下位レベルの探索を上回る結果を残せるものとなる．

2.3 AMAF

囲碁における UCT 探索において，プレイアウトを擬似的に増やす方法として，AMAF（All Move As First）という手法が研究されている．ある局面 A でのシミュレーションにおいて，ある可能手 a から始まるプレイアウトを行った際，そのプレイアウト内の自分の手をどのような順番で打ったとしても同じゲームができると仮定する．それにより，可能手 a だけでなく，局面 A の可能手中の，そのプレイアウトで打ったすべての手に対して，局面 A で打ったものとして可能手 a と同様の評価を与える方法である．すなわち，手の価値はどの順で打ったかではなく，どの位置に打ったかで決まると考える評価手法であると言える．AMAF による換算を用いることにより，多数の手を少ないプレイアウト数で擬似的に評価できるため，UCT 探索のプレイアウト数が少ない間の評価方法として用いられる．ただし，対戦ゲームにおいては，実際はどの手を先に打つかで相手の対応は変わってくる可能性が高いため，AMAF によって得られた評価は，通常のプレイアウトによる評価より正確さが失われている．

3. Nested Monte-Carlo 探索の AMAF 換算による探索削減

3.1 Nested Monte-Carlo 探索の実行時間

Nested Monte-Carlo 探索は，レベルを上げることでより効率的な探索を行うことができる手法であるが，上位レベルのゲームを実行するためには下位レベルのゲームを多数行う必要が有るため，レベルを上げるごとに爆発的に実行時間が増加するという問題点がある．Morpion Solitaire の Disjoint ルールでは，レベルを 1 上げたゲームを実行するには，200 倍程度の時間が必要となる．また，Touching ルールでは，可能手が多いためにさらなる時間増加があるものと考えられる．Disjoint ルールにおけるレベル 5 以上のゲームや，Touching ルールにおけるレベル 4 以上のゲームには現状では年単位の時間がかかると考えられ，さらなる上位レベルのゲームの実行は難しい．そこで，Nested Monte-Carlo 探索の下位ゲーム探索数を削減し，1 ゲームあたりの実行時間を短くすることで，より上位レベルのゲームの実行を可能とする手法を考えた．

3.2 AMAF 換算による探索削減

Nested Monte-Carlo 探索では，各局面において，すべての可能手に対して 1 度ずつ下位ゲームを行う．これにより，すべての可能手について，それぞれその手で始まる手筋を一つずつ見つけることになるため，少なくともそのレベルにおける手筋の分岐を見逃すことはない．しかし，多くのゲームにおいて，前の局面で可能手であった手は後の局面でも可能手である．そのため，メタレベルを上げるごとに，手順だけが入れ替わり，同じ手を打っている手筋が増えることになる．対戦ゲームにおいては，どの順番で打つかによって相手の対応が変わってくるため，このように自分の手順が前後したゲームは，同じものであるとは言えない．ただし Morpion Solitaire のような一人でプレイするゲームの場合，もし手順を入れ替えても成り立つ手筋が存在すれば，それを入れ替えたゲームは手数もゲーム進行上も元のゲームと変わりないため，全く同じ評価を与えても問題がないと考えられる．すなわち，手順が入れ替わっただけのゲームを複数見つけても意味はなく，この分は無駄な重複探索となっていると言える．

ここで，AMAF による換算法を取り入れる．ある局面 A である可能手 a から始めたプレイアウトにおいて打たれた手 b が局面 A の可能手でもある場合，Morpion Solitaire においてはルール上，この手 b と手 a を入れ替えた手筋もゲームとして成立するため，手 b は入れ替えた手筋を得たものとする．これをすべての手が一つ以上の手筋を得るまで行い，Nested Monte-Carlo 探索における「すべての可能手に対して 1 度ずつ」の下位ゲーム探索を終えたとする手法を提案する．このようにすると，すべての可能手について確実に一つ以上の手筋を見つけつつ，手順が入れ替わっただけの探索は削減することができる．この手法によるレベル 2 ゲームの実行の様子は，図 5 のようになる．

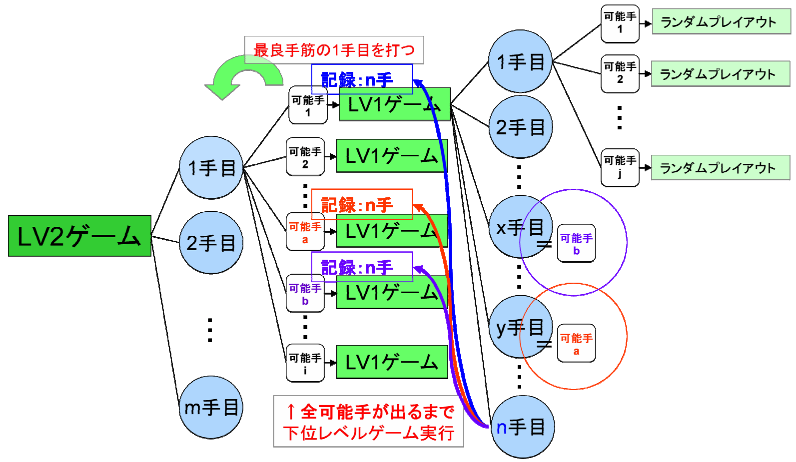

図 5. レベル 2 ゲームにおける AMAF 換算に基づく探索削減の図解

手筋の削減により，手順を入れ替えたものを削減することについては，ランダムシミュレーションの量が減るという問題以外はなく，効率的な探索削減であると言える．しかし以下のような手では問題となる．ある局面において手 a と手 b が可能手であるとする．このとき手 b を打つことで可能手となる手 c があり，さらに手 c は，手 a を打っていた場合には手 b を打っても可能手とならないとする．このような手 c が存在し，手 a から始めたシミュレーションで手 b が出てきたとすると，手 b に手 a と同じ評価を与えて探索を終了すると手 c に関する探索を行うことはできない．このような状況が多く存在する場合，見るべき手を削減してしまう場合が多くなると考えられる．

3.3 局面遷移時における着手の選択方法

AMAF に基づく換算では，各可能手につき最高一度ずつ下位ゲームを実行しても，そのシミュレーションの手筋に含まれていた可能手へ実効回数の加算を行うため，各可能手に対して複数の手筋の長さを得ることとなる．そのため，最長手筋を得た手をランダムに選択する以外にも，複数の手数記録から計算した結果により，なんらかの基準を設けて手を選択することも可能である．今回は，最長手筋であった手の中から，

- ランダムに選択する
- 手数の平均値が最大であった手を選択し，複数あった場合はランダムとする
- 打たれた回数が最多の手を選択し，複数あった場合はランダムとする

の 3 通りの方法によって，どの手を打つかを決定することとする．

3.4 実験と考察

提案手法に関して，Morpion Solitaire における実験を行う．3.3 節で示した局面遷移時における着手の選択手法を，それぞれランダム，平均，回数と名付け，各レベル 2 と 3 の実験を 10 時間行った．また，比較用に，未改造の Nested Monte-Carlo 探索の実験も行った．こちらのレベル 3 ゲームについては，10 時間では 1 度または 2 度程しか実行できないため，グラフでは省略する．また，各手法について，Disjoint ルールの結果は省略し，Touching ルールの結果のみを載せる．

まず，提案手法における各手数の分布は図 6 のようになった．

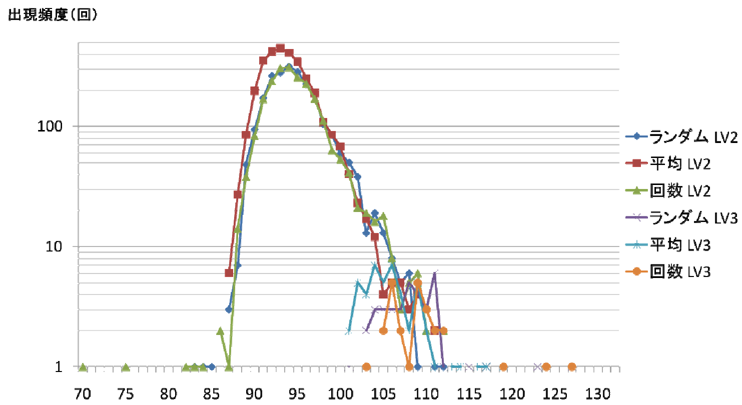

図 6. 実験結果：各手法のレベル 2 と 3 の各手数出現頻度の分布

また，各手法の最少手数と平均手数と最大手数のグラフは，図 7 のようになった．

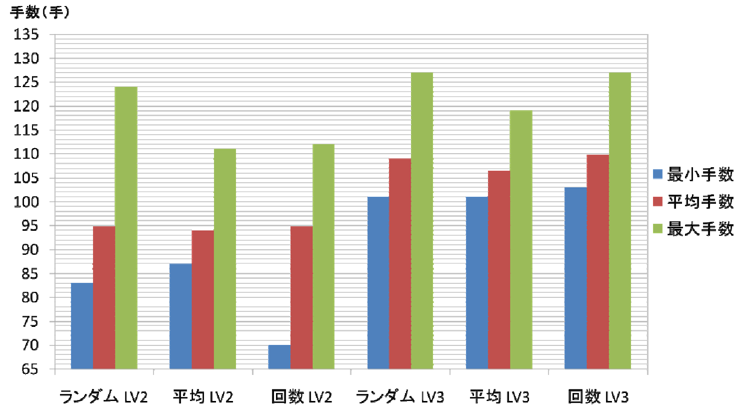

図 7. 実験結果：各手法のレベル 2 と 3 の最少・平均・最大手数

レベルを上げることで，最少，平均，最大手数がそれぞれ大きく伸びていることが分かる．各レベルとも，ランダム選択手法が最大手数を得ている．平均手数にはあまり変化はないが，回数最多の手法が最も大きくなっている．

また，各手法の 1 ゲームあたりの平均実行時間は，図 8 のようになった．

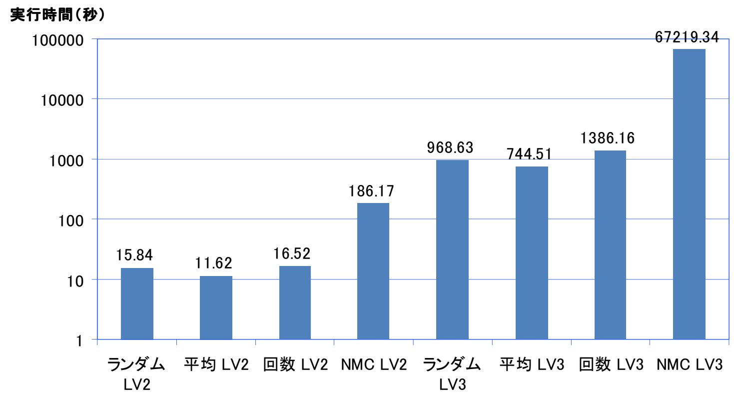

図 8. 実験結果：各手法のレベル 2 と 3 の実行時間

レベル 2 ゲームにおいて，既存の Nested Monte-Carlo 探索に対するランダム，平均，回数選択のそれぞれの手法についての平均実行時間は，0.085 倍，0.062 倍，0.089 倍となった．どれも実行速度は 10 倍速以上であり，非常に高速化できていると言える．また，レベル 3 ゲームでは，それぞれの実行時間は 0.014 倍，0.011 倍，0.021 倍と，50 倍～100 倍速になっており，レベルを上げることでさらに実行時間差が大きくなっている．このことから，さらなる上位レベルのゲームにおいても，指数関数的な高速化ができると考えられる．レベル 2 と 3 の結果から，レベル 4 は数十時間，レベル 5 は 1 箇月程度で実行することができると予測できる．これにより，高メタレベルのゲームも十分に実行が望めるものとなったと考える．

また，最大手数を得たランダム選択手法に関して，既存手法との各手数の出現頻度を比較すると，図 9 のようになる．

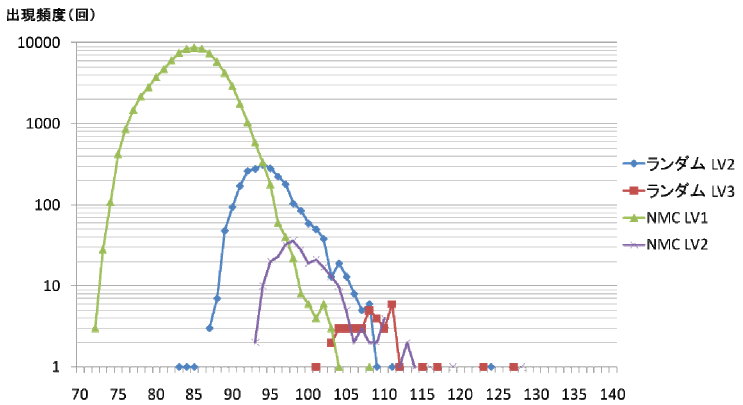

図 9．実験結果：ランダム選択と既存手法の各手数出現頻度手の分布

実際の出現頻度において，ランダム選択手法が同レベルでも既存手法を上回っていることが分かる．高レベル化によってはさらに手数が大きくなっている．

また，実際の出現頻度では，図 6 や図 9 のように，各手数の出方がばらばらで比較しづらくなっているため，最も長い手数側から順に加算し，「各手数以上」の手の出る確率を計算した．すなわち，「この手法のゲームを 1 回実行した場合に 100 手以上の手が出る確率」というグラフである．Morpion Solitaire においては，手数が長いものが良い記録であり，ある手数の短い記録が手数の長い記録より良いということはないため，このように手数以上の記録を比較すれば良いと考えられる．この結果は，図 10 のようになった．

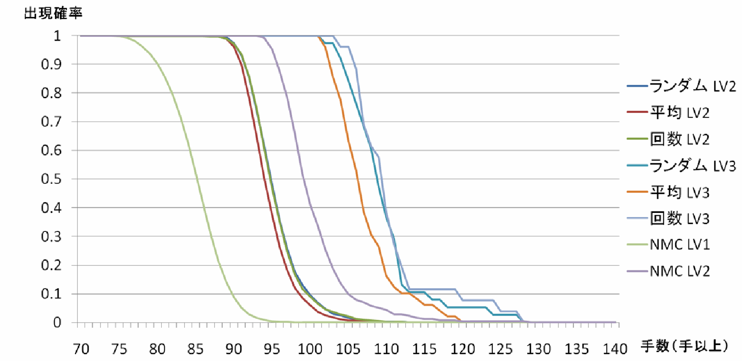

図 10．実験結果：各手法のレベル 2 と 3 のある手数以上の手の確率分布

1 ゲームごとの手の確率の分布では，ランダム手法と回数手法のレベル 2 は重なっている．平均手法がその下となっている．既存手法と比較すると，同レベルでは既存手法が勝っているが，提案手法でもレベルを上げることで記録が向上し，既存手法を大きく上回っていることが明確となっている．

また，各手法の実行時間は図 8 のようになっているため，これを用いて同時間あたりの各手数の出現確率を計算することができる．1000 秒あたりの各手法の手数の分布は図 11 のようになる．

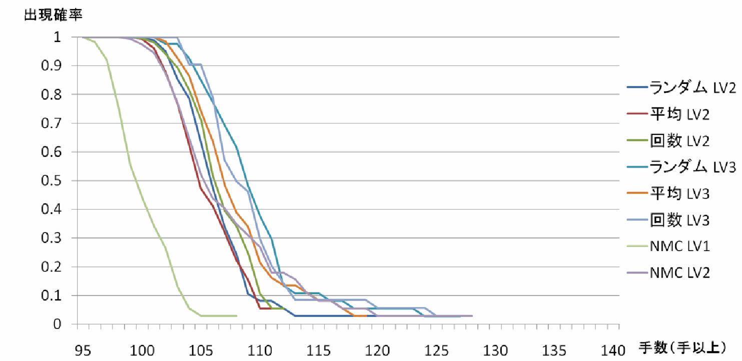

図 11．実験結果：各手法のレベル 2 と 3 の 1000 秒における手の確率分布

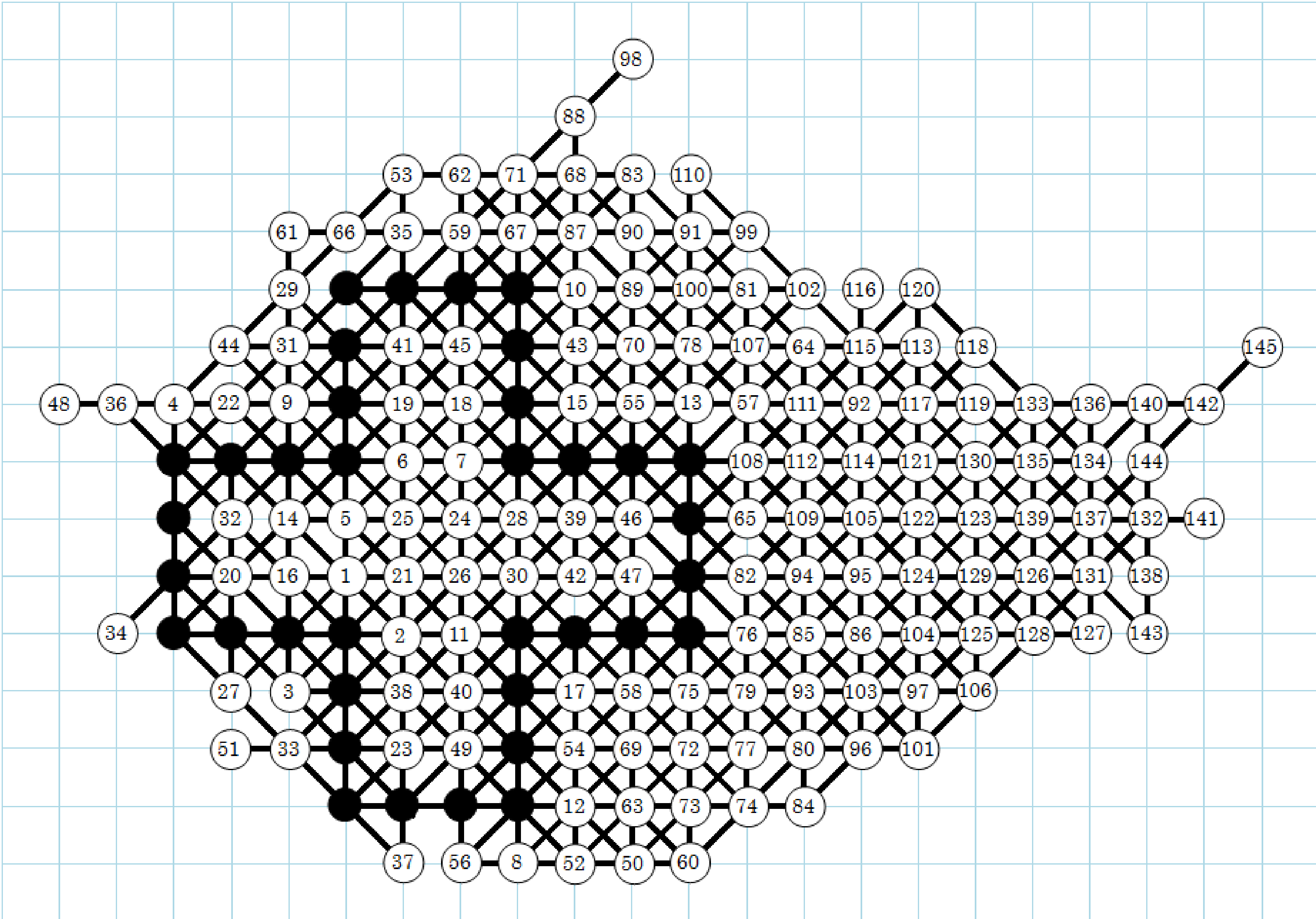

図 12. 実験結果：本手法による Touching ルールのコンピュータにおける世界新記録 145 手

それぞれ，1 回以上出ている手数までについて計算した．この計算では，実際に実行した場合と異なり図 10 にある最大手数以上の手は出ないため，それぞれ最高手数付近は正確ではないが，その前までの比較から，実行時間をそろえた場合にも高レベル化したゲームがよりよい結果を残していることは分かる．

これらの結果から，提案手法によって Morpion Solitaire のゲームを高速に探索することができ，またこれによる高レベル化で記録を改善できることが分かった．

最後に，図 8 において最も実行時間の短いことが示されている平均手法において，レベル 5 ゲームを実行したところ，およそ 21 日後，図 12 に示す 145 手の手筋が得られた．以前のコンピュータ探索による Touching ルールの Morpion Solitaire の世界記録は 144 手であったため，この結果により，世界記録を更新することができた．

4. おわりに

本研究では Morpion Solitaire を題材とし，AMAF 手法の換算法を用いて，Nested Monte-Carlo 探索の探索削減による高速化と高レベル化を行った．レベル 2 ゲームで 10 倍程度の高速化に成功し，より高レベルの探索を実現することができた．実験の結果，Morpion Solitaire の Touching ルールにおいて，本手法によって優れた結果を得ることができることが確認された．また，Touching ルールにおいてレベル 5 ゲームを実行し，コンピュータにおける探索の世界記録を更新する 145 手という記録を達成することができた．しかし，3.2 節で述べたような問題や，そもそも手の順序の入れ替えが可能なゲームであるかという問題から，Nested Monte-Carlo 探索が適用できるすべての問題に対して本手法が適用できるわけではない．また，探索数の削減手法に関しては，AMAF による換算を用いる以外の方法も考えられるため，最も効率の良い手法を得るためには，さまざまな手法による試行が必要であると考える．

参考文献

1) Tristan Cazenave: Reflexive Monte-Carlo Search, Proceedings of Computers Games Workshop 2007, pp.165-173 (2007).
2) Tristan Cazenave: Nested Monte-Carlo Search, IJCAI 2009, pp.456-461 (2009).
3) Erik D. Demaine, Martin L. Demaine, Arthur Langerman, Stefan Langerman: Morpion Solitaire, Theory of Computing Systems, Vol.39, No.3, pp.439-453 (2006).
4) Morpion Solitaire, http://www.morpionsolitaire.com/
5) MORPION SOLITAIRE, http://euler.free.fr/morpion.htm
6) Pentasol - Morpion Solitaire, http://pentasol.systemutvecklarna.se/

付録 Morpion Solitaire の「手」の記述および棋譜の表現方法

本研究におけるすべてのプログラムにおける, Morpion Solitaire のユニークに判別できる「手」の表現の仕方および，ゲームの棋譜のルールについて記載する．

Morpion Solitaire において，盤面に打つための「手」の持つ必要のある情報は，以下の 3 種類 4 データである．

- 駒の x 座標，y 座標
- 線の向き（縦，横，右上斜め，左上斜めのいずれか）
- 今打つ駒に対してどの位置に線を引くか

これらについて，以下のようにデータを持つ．

- 0～盤面サイズまでの数値の 2 変数
- 0，1，2，3 の 4 種類
- 今打つ駒が線の中で最も左上にある場合を 0，右に向かって 1～4 とする

このように，四つの整数で表される．また，人間の見ることもある棋譜データとしては，既存の記録のページ[5]や Touching ルールのゲームプログラム Pentasol[6]などで採用されている，以下の形式とする．

(x 座標,y 座標) 線の種類 線内での駒の位置

Pentasol との互換性を考える場合，各座標は，十字の内側 4 点のうち左上の駒が(28,28)にあるとした座標とする．線の種類は縦，横，右上斜め，左上斜めを「| - / ¥」で表す．「¥」記号は，本来バックスラッシュである．また駒の位置は 0～4 ではなく 2～-2 とする．これによる 1 手の記述は，以下のようになる．

(32,25) - -2

これは，x 座標が 32，y 座標が 25 の位置に駒を打ち，今打った駒を右端とするように水平な線を引く，という手を表している．