

Hexomino 解総数推定の解析と改善の提案

寺田 実^{1,a)}

受付日 2014年9月16日, 採録日 2015年4月3日

概要: 筆者は過去に大きな探索木に対する計数問題を推定するため, 乱数による選択と深さ優先探索を組み合わせた手法「複合モンテカルロ法」を考案し, 箱詰パズル hexomino の解総数を推定した. 今回, その手法に基づき長時間の計算を行った結果, 推定値の収束が安定しないという問題点が明らかになった. 分析の結果, その原因は推定に用いる分岐数の積のばらつきにあることが分かった. それを軽減するために, 分岐数の積に閾値を設けて全探索に入るという手法を新たに考案して計算を行ったところ, 従来よりも安定した推定を行うことが可能になった. またこの手法を N-Queens 問題についても適用したところ, 一定の成果が得られた.

キーワード: 箱詰パズル, 計数問題, 乱択アルゴリズム

Estimation of the Number of Solutions of the Hexomino Puzzle and Its Improvement

MINORU TERADA^{1,a)}

Received: September 16, 2014, Accepted: April 3, 2015

Abstract: We have proposed a randomized algorithm which estimates the number of certain nodes in large trees by combining conventional Monte Carlo method and depth-first search of subtrees. However the application of the algorithm to the hexomino puzzle shows an unstable behavior. We examined the result and devised a modification to the algorithm in which the product of branch factors along the path controls the start point of the depth-first search. We evaluated the revised algorithm in the hexomino and the N-Queen problem.

Keywords: polyomino tiling, counting problem, randomized algorithm

1. はじめに

筆者は文献 [1] において, 大きな木に対する計数問題のために確率的な試行と部分木の全探索を組み合わせた手法 (複合モンテカルロ法と名付けた) を適用することによって解の総数を推定する手法を提案し, 正方形 6 個を組み合わせたコマによる箱詰パズル (hexomino) の解の総数を推定した.

本論文では, まず上記の hexomino に関する計算を長時間かけて行った結果を示し, その吟味から得た知見によって推定の精度を向上させる手法について新たに提案する.

さらに hexomino と N-Queens を対象として実験を行い, その有効性を検証する.

2. 複合モンテカルロ法

本章では, 文献 [1] で提案した複合モンテカルロ法について概略を示す.

2.1 計数問題

大きな木の節点のうち, ある条件をみたすものの個数を求める問題を計数問題と総称する. 決められた形のコマを箱に詰める, いわゆる箱詰パズルの解の総数を求める問題はその一例となっている.

計数問題で厳密解を求めるには, 一般的には木の全探索を行うしかない. しかし, 木のサイズが大きい場合, 時間的

¹ 電気通信大学
University of Electro-Communications, Chofu, Tokyo 182-8585, Japan

a) terada.minoru@uec.ac.jp

に全探索は不可能である。そこで、確率的な試行を繰り返すことで解の近似値を求める手法が知られている。これはモンテカルロ法 [4] の一種であり、randomized algorithms [5] (乱択アルゴリズム) の一例となっている。

2.2 モンテカルロ法

モンテカルロ法は乱数に基づくサンプリングによって値を推定する手法である。複雑なパラメータを持つシステムにおいて多次元の積分を求めるために、パラメータ空間をランダムにサンプリングすることで近似値を求める。対象とするシステムは確率的なものも非確率的なものも含む。近年は応用分野もひろがり、物性物理、たんぱく質の立体構造、気象予測、金融工学や人口予測などにも利用されている。また、完全情報ゲーム (たとえば囲碁) のプレイヤーの戦略にも利用されている (これについては 7.1 節で本研究との関連を論じる)。

2.3 解総数の推定

箱詰パズルの解の総数の近似値をモンテカルロ法で求めるには以下のようにする。探索木の根 (箱に 1 つもコマのない状態) から出発し、その節点から移動可能な子節点を 1 つランダムに選び木を下降する (その時点で置くことのできるコマを 1 つ選んでそれを置く)。これを繰り返し、下降できなくなるところまで進む。これを 1 試行とし、停止した節点が解であれば成功、そうでなければ失敗である。失敗した試行には 0 を推定値として与え、成功した試行には、根から停止した節点までの分岐数 (branch factor ; 子節点の数) の積を推定値とする。この試行を多数回繰り返し、得られる推定値の期待値が解総数の近似値となる。

葉 i に至る分岐数の積を f_i とすると、1 回の試行による推定値 X_i は

$$X_i = \begin{cases} f_i & \text{葉 } i \text{ が解である} \\ 0 & \text{葉 } i \text{ が解でない} \end{cases}$$

であり、 X_i の期待値は、 p_i を葉 i に至る確率とすると、 $p_i = 1/f_i$ であるから、

$$\begin{aligned} E[X_i] &= \sum_{\text{葉 } i \text{ が解である}} p_i f_i \\ &= \sum_{\text{葉 } i \text{ が解である}} (1/f_i) f_i \\ &= \text{解の総数} \end{aligned}$$

となる。

しかし、この手法でも、解の密度が低い場合には試行の成功確率が小さくなり、多数の試行を行ったとしても 0 以外の推定値を求めることができない。

2.4 全探索との組合せ

これを解決するために文献 [1] で提案したのは、探索木

の一定の深さ (カットオフレベルと呼ぶ) まで確率的に下降し、その節点からは部分木の全探索を行う手法で、これを複合モンテカルロ法と名付けた。この場合、1 回の試行の結果は

- (1) カットオフレベルまで到達できなかった、
 - (2) カットオフレベルまで到達し、全探索を行ったが解は発見できなかった、
 - (3) カットオフレベルまで到達し、全探索を行って 1 つ以上の解を発見した、
- のいずれかとなる。その試行による推定値としては、(1)、(2) の場合には 0 を、(3) の場合には

$$\begin{aligned} &\text{カットオフレベルまでの分岐数の積} \\ &\times \text{全探索で求めた解の数} \end{aligned}$$

を与えることにし、その期待値として解総数の近似値を得る。

カットオフレベルにある節点 i に至る分岐数の積を f_i 、節点 i からの部分木中の解の総数を S_i とする。1 回の試行における推定値は $X_i = f_i S_i$ であるので、 X_i の期待値として下記のように解の総数がもとまる。

$$\begin{aligned} E[X_i] &= \sum_{\text{カットオフレベルの節点 } i} p_i (f_i S_i) \\ &= \sum_i (1/f_i) f_i S_i \\ &= \sum_i S_i = \text{解の総数} \end{aligned}$$

文献 [1] ではその有効性を示すために、箱詰パズルの一種である hexomino の解総数を求める問題に適用した。Hexomino は正方形 6 個からなる 35 種のコマを定められた箱に詰めるものである (図 1)。マスの総数は 210 であるが、コマのパリティの関係から長方形の箱にはおさまらず、 $19 \times 11 + 1$ と中央にでっばりのついた箱になっている。

探索木のサイズは 10^{30} 程度とかなり大きく全探索は困難である (サイズの見積りについては 7.2 節で説明する)。一方、解までランダムに下降する手法では解の密度が低いためなかなか解に到達できない (葉までのランダム下降により解の密度を実験的に測定したところ 10^{-10} 程度であっ

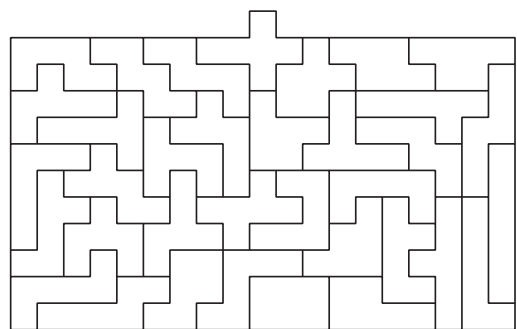


図 1 Hexomino の解の一例

Fig. 1 A solution of hexomino.

た). 複合モンテカルロ法では, 適切にカットオフレベルを選ぶことによって, 全探索開始点までの下降の成功率を高めるとともに, 部分木の全探索にもそれほどの時間を要しないようにできる. この結果有効な試行回数 (前述のタイプ (3) の試行) を増やすことができる.

実験の結果, カットオフレベルを 22 ないし 23 にした場合に最も有効な試行が得られることが分かり, hexomino の解総数を 10^{25} 程度と結論した. なお, 文献 [2] では複合モンテカルロ法の理論的考察が試みられている.

3. 複合モンテカルロ法の解析と問題点

3.1 ランダム下降と深さ優先探索の効率比較

まず, 複合モンテカルロ法の有効性について考察する.

箱詰パズルの探索木において解かどうか判明するのは, それ以上コマを置けなくなった状態である葉においてである. その深さがコマ総数に等しければ解である. したがって, モンテカルロ法で試行の効率を決定するのは葉への到達効率である. 複合モンテカルロ法の有効性は, 部分木に対して全探索を利用することでランダム下降よりも効率良く葉に到達できていることによる.

根から葉まで完全にランダムに下降する手法と根からの深さ優先探索について, 葉への到達効率を実験的に比較してみた (表 1).

これによると, 葉への到達は深さ優先探索のほうが 70 倍以上効率的である. その要因は次の 2 つである:

- 訪問すべき節点数の相違
ランダム下降は 1 試行ごとに根から葉までの節点を訪問する必要がある. これに対して深さ優先探索ではバックトラックによって効率的に葉を訪問できる. たとえば深さ n で, すべての中間節点での分岐数が k であるバランスした木を考えると, 深さ優先ですべての葉 (k^n 個) に到達する際に訪問する (葉も含めた) 節点数は $1 + k + k^2 + \dots + k^n = \frac{k^{n+1} - 1}{k - 1}$ である. k が大きい値であれば, これはほぼ葉の総数 k^n に近い. 一方, ランダム下降での訪問節点数は $n \times k^n$ となり, n 倍の節点を訪問する必要がある.
- 節点あたりの計算コストの相違
ランダム下降では, 節点ごとに枝の列挙と乱数発生が必要になる. さらに, このとき生成した枝の集合はランダム下降では再利用できないが深さ優先では一度生成した枝集合は子節点リストとして活用できる. 表 1

表 1 ランダム下降と深さ優先探索の効率比較 (100 秒あたり)
Table 1 Comparison of random descent and depth-first search (Number of nodes visited in 100 sec.).

	訪問葉数	訪問節点数
ランダム下降	4,261,336	31,825,102
深さ優先探索	308,660,538	452,042,321

の訪問節点数の比は約 14 倍となっており, この差を示している.

3.2 解総数推定値の推移

文献 [1] の発表後さらに計算を続けた結果を図 2 に示す. 横軸は実験の回数, 縦軸はそれまでの実験から推定される解総数である. 横軸は単に実験の順序を示すもので, それ以上の意味はない. 6 本の線はカットオフレベルの違いを示す.

実験の詳細は以下のとおりである.

- 1 回の実験は CPU 時間で 60 分とする (Linux 上で SIGVTALRM (Virtual alarm clock) を使用). 乱数の seed を指定して起動する.
- 1 回の試行はカットオフレベルまでの乱数を用いた下降と, 下降に成功した場合にはその節点からの全探索からなる.
- 試行中に制限時間に到達した場合, その試行はそこで打ち切り, 着手しなかったものとする. 本来の意図は計算時間を一定とすることで, カットオフレベルごとの推定値の収束の速さを比較したかったためである. しかしここに問題がある可能性がある. 試行が長時間かかるのは全探索で大きな部分木を対象としている場合なので, 多数の解を得られる可能性のある試行を打ち切りの結果失うことになり, 推定値を小さい方向にバイアスする可能性がある.

図 2 から読み取れるのは以下のとおり:

- 推定値はカットオフレベルによらずほぼ 10^{25} 程度である.
- まれに発生する大きな値によって推定値は大きな影響を受けており, 安定して収束しているとはいいいにくい.

3.3 結果の解析

図 2 のカットオフレベル 23 のデータには横軸 10,000 付

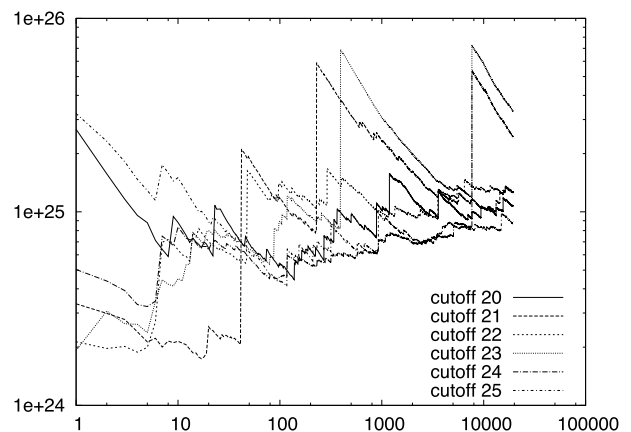


図 2 従来手法での解総数推定値の推移 (cutoff level 20-25)
Fig. 2 Estimated numbers of solutions for cutoff level 20-25 in previous method.

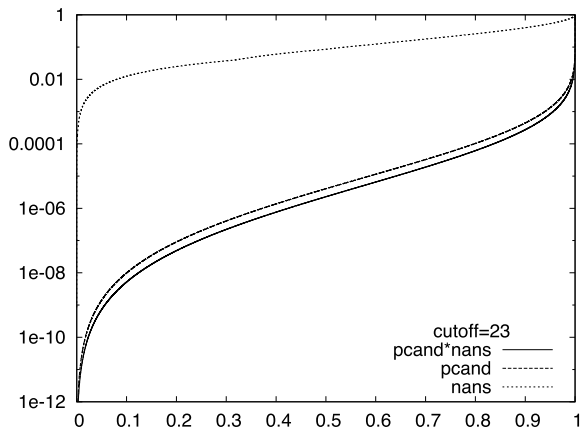


図 3 従来手法での推定値 (pcand*nans), 分岐数積 (pcand), 全探索結果 (nans) の分布

Fig. 3 Distribution of pcand*nans, pcand and nans in previous method.

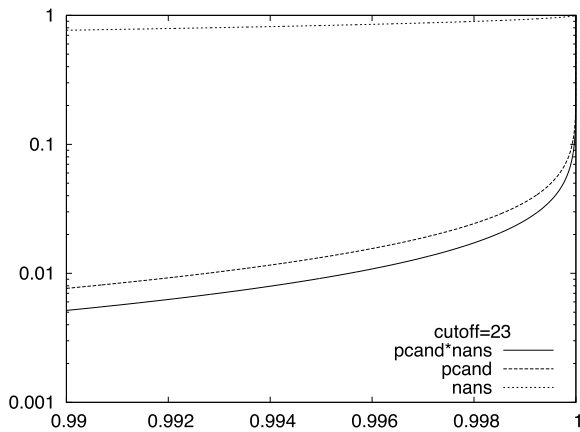


図 4 従来手法での推定値 (pcand*nans), 分岐数積 (pcand), 全探索結果 (nans) の分布 ((1.0, 1.0) 付近拡大)

Fig. 4 Distribution of pcand*nans, pcand and nans in previous method (near (1.0, 1.0)).

近に 7×10^{25} に達するピークが見られる。このデータを精査してみると、分岐数積 3.36×10^{36} 、全探索結果 6 となり、これらの積をその時点での累積の総試行数 3.3×10^{11} で割ってみると推定値に 6.1×10^{25} の影響をあたえていることになる。

推定値のばらつきの状況を把握するために、推定値とその計算に影響をあたえる 2 つの値である分岐数積 (pcand) と全探索結果 (nans) について、その分布をプロットしてみた。カットオフレベル 23 の実験データを対象として、成功した試行 (前述の (3): n 件とする) の値を小さいものから昇順にソートしたものを V_i ($i = 0, \dots, n-1$) としたとき、 $(i, \sum_{j=0}^{i-1} V_j)$ をプロットした (図 3) (縦軸は対数とし、両軸とも上限を 1.0 に正規化してある)。また図 4 には (1.0, 1.0) 付近の拡大図を示す。

拡大図を見ると推定値 (pcand*nans) のプロットは (0.998, 0.02) 付近を通っており、上位わずか 0.2% の試行が最終推定値の 98% 以上に寄与している実態が見てとれる。

さらに、推定値のばらつきをもたらしているのが全探索結果ではなく分岐数積の方であることもグラフから読み取れる。

このようなことが起きる原因は、カットオフレベルを一定にしても、盤面配置や節点における枝刈りの影響で分岐数がばらつき、その積に大きな差が生じることによる。推定値の収束を向上させるには分岐数積のばらつきを緩和する必要があることが分かった。

こうした確率変数の値が期待値から大幅にずれる確率を末端確率 (tail probability) と呼び、確率変数の分散が既知であればチェビシェフの不等式などによりその上限を見積もることが可能である [3]。しかし本研究のように分岐数の分布が未知である場合には分散そのものも実験によってしか推定できず、上限の推定は困難である。

4. 分岐数積を用いたカットオフの提案

文献 [1] では、実験を通じてカットオフレベルを固定とした。その根拠は、そこまでの乱数による下降とそこから全探索の両方がそれぞれほぼ一定のサイズとなり、試行結果である推定値のばらつきを減少させることを期待したからである。しかし、前節で見たとおり分岐数積のばらつきが推定値の収束に大きな影響を与えてしまっていた。

そこで、カットオフレベルを固定するのではなく、分岐数積に閾値を設けて、各試行において乱数を用いた下降の過程で分岐数積が閾値を超えた節点から全探索を行うことを考えた。分岐数積が大きくなり過ぎる試行において、より浅いレベルから全探索を行うことで、大きな部分木を対象とすることになる。これは局所的な異常を薄める可能性があり、安定した結果が得られることを期待する。

一方、結果として全探索に入るレベルはまちまちとなり、全探索で得られる結果の変化が大きくなることが予想される。その影響を評価するためにも実験が必要である。

5. 実験

従来手法 (カットオフレベル固定) と提案手法 (分岐数積の閾値固定) の比較を行うために、hexomino を対象として新たに実験を行った。

実験に用いたプログラムは従来と同様のもので、1 回の実験は 60 分の CPU 時間を与えて乱数の seed を変更しながら繰り返す。ただし変更点が 1 点あり、実行途中の試行は制限時間が過ぎていてもそれが完了するまで計算を続ける。これは、3.2 節で述べた全探索の打ち切りによる結果のバイアスを防止して、正しく結果に反映させるためである。

実験のパラメータとしては従来手法のほうはカットオフレベルを 20-25 とし、提案手法については全探索に移行すべき分岐数積の閾値を 10^{26} から 10 倍ずつ変化させて 10^{31} までとした。

60 分の実験を 1 セットとして、2,000 セットの実験での値である。

5.1 推定値とそのばらつき

表 2 に新旧の手法による hexomino の解総数の推定値を示す。標準偏差は推定値がゼロとなる試行（全探索に到達できないケース (1) と全探索で解が 1 つも見つからなかったケース (2)）も含んでの値である。この結果から読み取れるのは以下のとおりである：

- どちらの方式もほぼ同様の推定値（およそ 10^{25} ）を得ている。
- 提案手法は分岐数積を変えた場合の推定値のばらつきが小さい（最大値と最小値の比率が $1.2235/1.0320 = 1.186$ ；従来手法においては $1.0009/0.7629 = 1.312$ ）。
- 提案手法のほうが推定値が大きい。これについては、以前の実験（図 2）に見られるように、従来手法で今後発生する大きな値がまだ出現していないためと考えられる。

5.2 推定値の推移

推定値の推移を図 2 と同様、横軸に実験順をとり、図 5、図 6 に示す。これから読み取れることは以下のとおりである：

- 従来手法は以前の結果（図 2）と同様の不安定な傾向を示している。
- 従来手法にくらべて提案手法は推定値が安定している。

5.3 結果の分布

分岐数積、全探索結果数、推定値のばらつきの実態を確認するために、従来手法、提案手法のそれぞれについてそれらの分布を図 3、図 4 と同様の方法で示す。従来手法（カットオフレベル = 23）の分布が図 7 とその拡大図 8

で、提案手法（閾値 = 10^{29} ）の分布が図 9、図 10 である。推定値（pcand*nans）の横軸 0.998（上位 0.2%）での値を比較すると、従来手法では（0.998, 0.075）を通り最終推

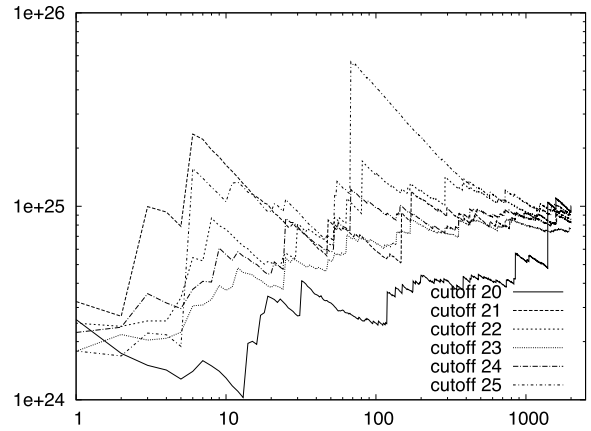


図 5 従来手法での解総数推定値の推移（新実験）(cutoff level 20-25)
Fig. 5 Estimated number of solutions for cutoff level 20-25 in previous method (new experiment).

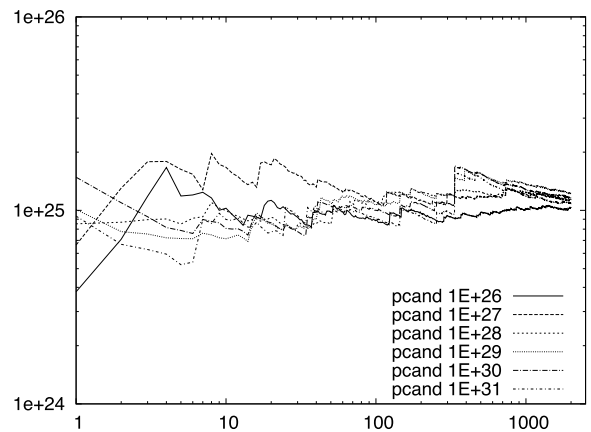


図 6 提案手法での解総数推定値の推移（分岐数積の閾値 10^{26} - 10^{31} ）
Fig. 6 Estimated number of solutions in proposed method (pcand 10^{26} - 10^{31}).

表 2 得られた解総数の推定値

Table 2 Estimated number of solutions.

手法	推定値 ($\times 10^{25}$)	標準偏差 ($\times 10^{25}$)
従来手法		
cutoff = 20	1.0009	0.4236
cutoff = 21	0.8284	0.1002
cutoff = 22	0.8542	0.1100
cutoff = 23	0.7627	0.0535
cutoff = 24	0.8997	0.0939
cutoff = 25	0.9289	0.1920
提案手法		
分岐数積 = 10^{26}	1.0320	0.0476
分岐数積 = 10^{27}	1.1310	0.0605
分岐数積 = 10^{28}	1.1562	0.0369
分岐数積 = 10^{29}	1.2235	0.0740
分岐数積 = 10^{30}	1.1652	0.1155
分岐数積 = 10^{31}	1.0956	0.1197

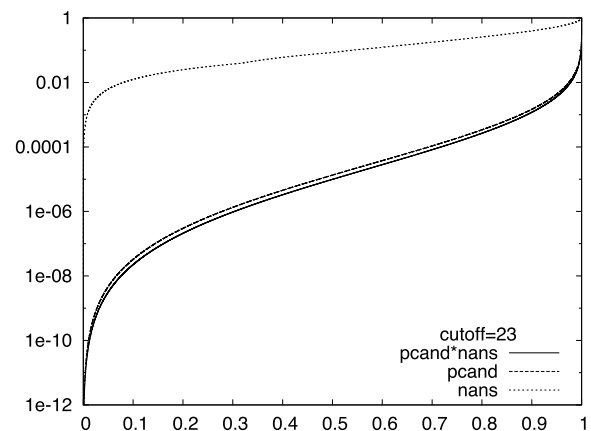


図 7 従来手法での推定値（pcand*nans）、分岐数積（pcand）、全探索結果（nans）の分布（cutoff = 23）
Fig. 7 Distribution of pcand*nans, pcand and nans in previous method (cutoff = 23).

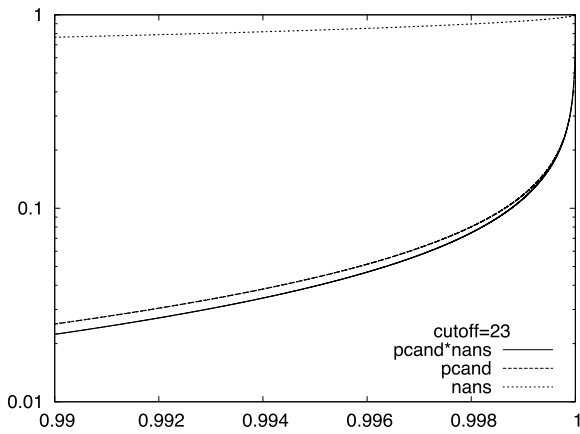


図 8 図 7 の (1.0, 1.0) 付近拡大
Fig. 8 Fig. 7, near (1.0, 1.0).

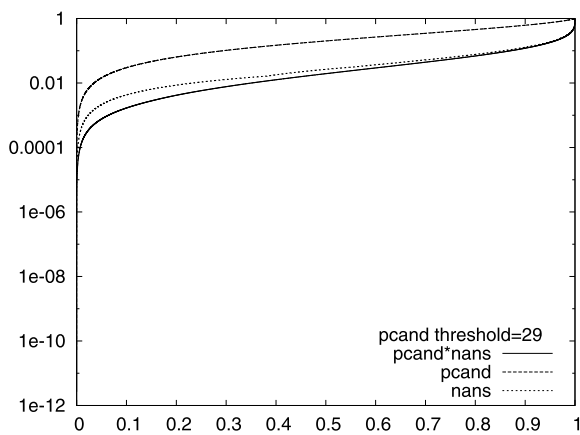


図 9 提案手法での推定値 (pcand*nans), 分岐数積 (pcand), 全探索結果 (nans) の分布 (閾値 = 10^{29})
Fig. 9 Distribution of pcand*nans, pcand and nans in proposed method (threshold = 10^{29}).

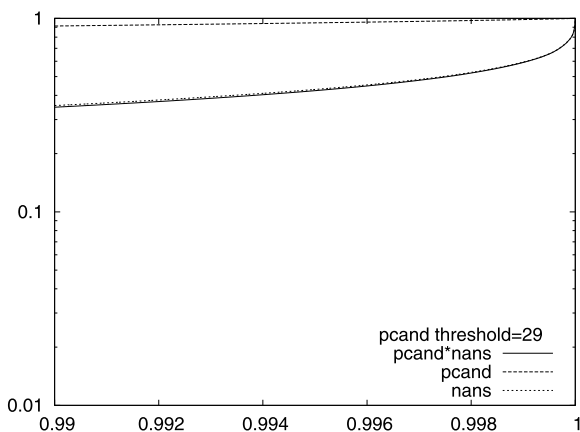
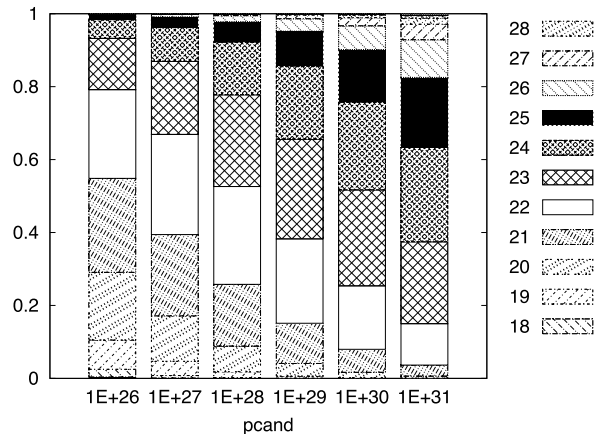


図 10 図 9 の (1.0, 1.0) 付近拡大
Fig. 10 Fig. 9, near (1.0, 1.0).

定値の約 92.5%に寄与しているのに対して、提案手法では (0.998, 0.52) を通り約 48%の寄与へと大きく減少しており、ばらつきを減少させることへの有効性を確認することができる。

提案手法の問題点とも考えられた全探索結果 (nans) の



横軸は全探索を開始する分岐数積の閾値、色分けは実際に全探索に入るレベルを示す。

図 11 提案手法のもとでの全探索開始レベルの分布 (分岐数積の閾値 10^{26} – 10^{31})

Fig. 11 Starting levels of depth-first search in proposed method (pcand 10^{26} – 10^{31}).

ばらつきについては、全探索をはじめるレベルがまちまちになったことが原因となって増加しているが、それでも従来手法の分岐数積に比較すればプロットははるかに上を通過して、上位の値の寄与がそれほど大きくないことを示しており、結果の安定性につながっているといえる。

5.4 カットオフレベルの分布

4章の末尾で述べたように、提案手法では分岐数積の閾値によって全探索に入るかを判断するため、その結果として全探索に入るレベルはまちまちとなる。それが実際にどう分布しているかを図 11 に示す。横軸は設定した分岐数積で、縦軸はそれぞれの設定での全探索に入るレベルの割合である。これを見ると、全探索開始レベルが一定の範囲で分布すること、分岐数積の閾値を大きくするほど全探索開始レベルが深くなることを確認できる。

6. N-Queens 問題への適用

Hexomino の解総数以外の問題での有効性を検討するために、N-Queens 問題 ($N = 45$) に提案手法を適用してみた。正方形の盤面に N 個の Queen をおたがいに取合わないよう配置するのが解の条件で、その解総数を hexomino と同様の手法で推定する。ランダム下降では、Queen を各行に上から順に 1 つずつ、縦とななめで取合わないという制約のもとで配置していく。従来手法ではカットオフレベルを、提案手法では分岐数積をいくつか変えて解の総数を推定した結果を表 3 に示す。

結果からは、どちらの手法も安定した推定値を得ていること、提案手法の分岐数積 = 10^{38} の場合が最も推定値のばらつきが少ないことが分かり、提案手法に一定の有効性があることがいえる。

表 3 45-Queens の解総数の推定値

Table 3 Estimation of the number of solutions of 45-Queens.

手法	推定値 ($\times 10^{38}$)	標準偏差 ($\times 10^{38}$)
従来手法		
cutoff = 34	1.082500	0.001429
cutoff = 35	1.082445	0.001329
cutoff = 36	1.082529	0.001377
cutoff = 37	1.082508	0.001674
提案手法		
分岐数積 = 10^{37}	1.082406	0.001928
分岐数積 = 10^{38}	1.082469	0.001159
分岐数積 = 10^{39}	1.082499	0.001317
分岐数積 = 10^{40}	1.082595	0.002349

7. 議論

7.1 モンテカルロ木探索との関係

近年ゲームプレーヤを主な対象としてモンテカルロ木探索 (MCTS) の研究 [6] が進んでいる。MCTS では、ゲーム木の探索に乱数に基づく着手の選択によるゲームの実行 (プレイアウト) を併用する。プレイアウトの利点は、出発点となる局面からの多数のプレイアウトによって、静的な評価の代替にできる点にある。展開済みの探索木から最も有望な節点を選んで一段展開し、その子節点からプレイアウトを実行する。プレイアウトの結果は木にフィードバックされ、節点の評価値の更新に利用される。こうして有望な局面について深い探索を行うことが可能になる。MCTS は囲碁のような静的評価の難しいゲームに効果を発揮し、現在活発に研究が行われている。

本研究と MCTS との違いは、ゲームプレーヤでは「有望な手」を探すのが目標であるが、計数問題にはそうした有望さのような尺度が存在しない点である。有望な着手を探すのであれば、そうした節点を深く探索したり min-max や枝刈りによって評価の低い節点を排除できたりするが、解総数を求めるにはバイアスのないサンプリングが必要となり、MCTS の手法は適用が困難である。

本研究と直接対応するのはプレイアウトの部分である。この部分については、乱数のみによる着手の選択ではなく $\alpha\beta$ 枝刈りによってより現実的なプレイアウトを行うという研究 [7] があるが、上で述べた相違により本研究には適用できない。一方、本研究の提案手法は MCTS のプレイアウトへも適用可能ではないかと考える。乱数を用いたプレイアウトにおいて、その末端部分を全探索することが可能であればより効率良くサンプリングできる可能性がある。これについては具体的なゲームでの実験を通して研究していく必要がある。

また、モンテカルロ法には重点サンプリングという手法があり、均等なサンプリングではなく有望そうな領域をカバーする分布を設定し、それに従ってサンプリングする。

これは本研究が対象とするような解密度の低い木には有効に思えるが、そのためにはその分布が既知である必要があり、本研究のような問題には適用ができないと考える。

7.2 探索木のサイズの見積り

本手法は探索木のサイズの見積りにも適用できる。解となっている葉だけを数えていた部分を、成否にかかわらず数えるように修正するだけである。この方法を用いた実験により hexomino の探索木のサイズを 10^{30} 程度と見積もることができた。

8. 結論

本論文では確率的な下降と全探索を組み合わせた複合モンテカルロ法に対して、hexomino の長時間の計算によって得られた知見に基づき、カットオフレベルを固定する代わりに分岐数積を手がかりに全探索に移行するという手法を提案した。Hexomino を対象とする実験により、試行による個別の解のばらつきを抑えて、より安定した推定値が得られることを示した。さらに、N-Queens 問題についても同様の手法を適用して、一定の効果を上げることが確かめられた。

今回の改善提案の一般的な意味は、カットオフレベルを固定する必要がなく問題の振舞いに応じて適応的に設定する自由度を示したことにあるといえる。それを利用して hexomino では分岐数積のばらつきを制御することができたし、異なる問題ではまた別の設定が可能であると考えられる。

謝辞 本論文の査読者各位からいただいた貴重なコメントに感謝します。

参考文献

- [1] 寺田 実：複合モンテカルロ法による探索木の解総数の見積り，情報処理学会論文誌，Vol.38, No.8, pp.1492–1497 (1997).
- [2] 中村将人，小川重義：計数問題における複合モンテカルロ法について，数理解析研究所講究録，Vol.1351, pp.134–142 (2004).
- [3] 玉木久夫：乱択アルゴリズム，アルゴリズム・サイエンスシリーズ，共立出版 (2008).
- [4] 津田孝夫：モンテカルロ法とシミュレーション (三訂版)，培風館 (1995).
- [5] Motwani, R. and Raghavan, P.: *Randomized Algorithms*, Cambridge University Press (1995).
- [6] Browne, C.B. et al.: A Survey of Monte Carlo Tree Search Methods, *IEEE Trans. Comp. Intell. AI Games*, Vol.4, No.1, pp.1–43 (2012).
- [7] Winands, M.H.M. and Björnsson, Y.: $\alpha\beta$ -based Play-outs in Monte-Carlo Tree Search, *Proc. IEEE Conf. Comput. Intell. Games*, pp.110–117 (2011).



寺田 実 (正会員)

1959年生。1981年東京大学工学部計
数工学科卒業。1983年同大学大学院
工学系研究科情報工学修士課程修了。
東京大学計数工学科助手，電気通信大
学電子情報学科助手，東京大学機械情
報工学科助教授をへて2002年電気通
信大学情報通信工学科助教授。現在，電気通信大学情報・
通信工学科准教授。工学博士。ソフトウェア科学会，ACM
各会員。