複合モンテカルロ法による探索木の解総数の見積り

寺田 実†

確率的アルゴリズムは，通常のアルゴリズムでは難しい問題を，乱数を用いて近似的に解く有効な手段の1つである．その1つであるモンテカルロ法を用いて，探索木に含まれる解の総数を見積もることができるのは広く知られている．しかし，解の密度が低い場合には，非常に多くの試行を必要とするという問題点がある．本論文では，モンテカルロ法と全探索を組み合わせた複合モンテカルロ法を提案する．この方法によって，従来よりも多くの有効な試行を行うことができ，近似の精度を高めることが可能になる．2つの例題に対して実験を行い，その有効性を示す．

Estimating Number of Solutions with Hybrid Monte Carlo Method

MINORU TERADA†

Randomized algorithm can solve problems which need long time for conventional algorithms. Counting answer nodes in a tree is one of such problems, and using Monte Carlo method one can estimate it. But if the density of the answer is low, it needs long time to estimate the value. I propose Hybrid Monte Carlo method, which is the conbination of Monte Carlo method and brute-force search. The efficiency of the method is demonstrated through two experiments.

1. 組合せ問題と木の探索

組合せ的な問題の多くは，長さ n の順列または組合せのうち，ある条件を満たすものを解とする形に定式化できる．この種の問題の解を求める最も基本的な方法は，順列（組合せ）を順次生成し，条件を満たすかどうかのテストを行うことである．実際のプログラムとしては，順列（組合せ）を先頭要素から決めていく方針で，バックトラックを用いた木の探索とすることができる．

組合せ問題は，条件を満たす解を1つだけ求める場合と解の総数を求める場合がある．後者を計数問題（counting problem）と呼び，一般には前者に比べて難しいとされている．

たとえばn Queens問題（以下では $Q(n)$ と記すことにする）を例としよう．これはチェス盤の上に8個（一般には n 個）の女王を相互に取り合わないように配置する問題である．これを解くアルゴリズムとしては，再帰呼び出しを利用して女王を順次の行に配置していくものが広く知られている．

この問題について，解を1つだけ求める問題は探索問題の例として多くの研究[1),2)]がなされており，特に，ある種の形の解はすべての n (≥ 4) について明示的に求まっている[3)]．これに対して，計数問題については，木の全探索が必要になり指数的な時間を要するため，大きな n に対しては $|Q(n)|$ は分かっていない（絶対値の記号で解総数を表すことにする）．

本論文は木に対するモンテカルロ法を拡張することで，計数問題の近似値を従来よりも効果的に求めるアルゴリズムを提案するものである．

本論文の構成は，2章でモンテカルロ法とその木の計数問題への適用を解説し，3章では提案するアルゴリズムについて述べる．4章では2種類の問題に対して実験を行い，その効果を示す．5章では提案するアルゴリズムと従来の方法との関連について考察を行い，6章をまとめとする．

2. モンテカルロ法による木の計数問題

近年研究がさかんになっている分野に確率的アルゴリズム（randomized algorithm）がある[4)]．これは，通常の解法では時間を要する問題に対して，乱数を利用することによって，

- 近似解をその近似の精度に見合った時間で求める
- 厳密解を確率的に定まる時間で求める

などを行うものである．

† 東京大学工学部
Faculty of Engineering, University of Tokyo

古典的な確率的アルゴリズムとして，モンテカルロ法（Monte Carlo method）がある．よく知られている例に π の推定がある．区間 [0..1] に一様に分布する乱数を 2 つずつ組 (x, y) で発生させると，そのうち $x^2 + y^2 < 1$ を満たす組の出現頻度は，$\pi/4$ に近付く．モンテカルロ法は，一般に多次元の定積分を求めたり，複雑な系の動作シミュレーションなどに広く利用されている．

木の計数問題に対しても，このモンテカルロ法が適用可能であることは広く知られている．アルゴリズムの概要は以下のとおり：

1 回の試行を，木の根から出発して下降する行程とする．この行程において，分岐する節点ではそのうちの 1 つをランダムに選択する．下降の結果，解に出あった場合，根からそこまでの中間節点での分岐数の積 f_i を，解総数の（その試行における）近似値 X_i とする．解に到達しなかった場合には，解総数 X_i を 0 とする．この試行を多数回繰り返せば，解総数の近似値が求まる．

葉 i に至る分岐数の積を f_i とすると，1 回の試行による近似値 X_i は

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

こうすると，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．

ここで問題となるのは解の密度である．

モンテカルロ法の近似精度は見つかる解の個数に依存する．したがって，乱数による試行によって，十分な個数の解を見つける必要がある．解の密度を，1 回の試行で解が見つかる確率とすると，一定個数の解を発見するには，その値に反比例する試行回数を必要とすることになる．

したがって，解の総数が小さかったり，探索空間が広かったりした場合にはモンテカルロ法をもってしても解総数を求めることが困難となる．

解の密度が低い場合にとられる手段として，加重サンプリング（importance sampling）がある．これは，解が密に存在する場所で重点的に試行するもので，問題の性質を利用して探索空間を狭めていることになる．

木の探索問題でも探索空間を狭めることは非常に効果的である．分岐節点において，明らかに解を持たないと分かる枝を排除して木の分岐数を減らすもので，一般には枝刈り（pruning）と呼ばれている．

$Q(8)$ において，最も原始的には，すべての女王の位置の組合せとして，${}_{64}C_8$ 通りを検討しなくてはならない．しかし，各行，各列に女王を 1 つずつ配置すればよいという考察から，[0..7] の順列，すなわち 8! 通りを検討すればよい．実際，ここまでの枝刈りを行えば，解の密度は十分高くなり，通常のモンテカルロ法でも解総数を求めることが可能になる．たとえば $Q(19)$ では，解の密度は 0.026 程度，$Q(400)$ においてさえ $3 * 10^{-4}$ 程度はある（つまり，その意味では，$Q(n)$ はそれほど難しい問題ではない）．

しかし，現実にはそれほど枝刈りが有効に機能しない問題も多い．これらは問題のサイズが大きくなるにつれて急激に解密度が下がってしまう．本論文で例としてあげた Hexomino では 10^{-10} 程度となる．

こうした難しい問題に対して，木に対するモンテカルロ法を拡張することで解総数の近似を求める，というのが本論文の主題である．

3. 複合モンテカルロ法

この章では，本論文の中心である複合モンテカルロ法（Hybrid Monte Carlo method）について，その基本的なアイデアとアルゴリズムを説明する．

3.1 基本的アイデア

本方式の基本的アイデアは，通常のモンテカルロ法における乱数を用いた降下と，部分木に対する全探索の組合せである．

1 回の試行において，木の根から一定の深さまで乱数で降下し，その中間節点を根とする部分木に対して全探索を行うのである．この「一定の深さ」をカットオフレベルと呼ぶことにする．

後述するように，解を発見する効率はランダム降下に比べて全探索の方が高いため，解密度が低い場合でも有効な試行を行いやすく，少ない計算コストで高精度の近似値が求まるのである．

3.2 アルゴリズム

複合モンテカルロ法における 1 回の試行は，2 つのフェーズに分かれる．木の根からスタートして，分岐点では乱数によって進路を選択しながらカットオフレベルに向かって降下していくフェーズ（ランダム降下）と，それによって到達した節点を根とする部分木に対する全探索である．

試行によっては，ランダム降下中に解でない節点（つまり，分岐数が0である節点）で停止してしまう場合もある．この場合は全探索フェーズはないけれども1回の試行として扱う．

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 とすると，1回の試行における近似値 X_i を以下のように定める：

$$X_i = f_i S_i$$

すると，X_i の期待値として，解の総数が求まる．

$$\begin{aligned} E[X_i] &= \sum_{\text{節点 } i \text{ 以下の部分木}} p_i f_i S_i \\ &= \sum_i (1/f_i) f_i S_i \\ &= \sum_i S_i = \text{解の総数} \end{aligned}$$

3.3 ランダム降下と全探索の比較

ランダム降下と全探索の計算量を比較してみる．簡単のために，深さ n の二進木（**図1**）を考える．

完全にバランスした木の場合（図1左）には，根からの降下を行った場合の行程長は n であり，葉は 2^n だけあるので，全体をカバーするための計算量は $n*2^n$ となる．一方，深さ優先の再帰的全探索では，中間節点と葉のすべてを1回ずつ訪問することになり，計算量は 2^{n+1} となる．これらの違いは $n/2$ 程度であり，ほとんど違いはない．

これに対して，木がバランスしていない場合には大きな違いが出る．極端な場合（図1右）を考える．ランダム降下によると，全試行の半分は最初の分岐で右の節点に吸収されてしまう．残りのさらに半分はその下のレベルで吸収され，低いレベルに到達する数は指数的に減っていく．最低レベルに到達するものの数は結局 $1/2^n$ となり，バランスした木の場合と同じになる．つまり，全体をカバーするにはやはり 2^n の回数が必要である．ただし試行の行程長については，短いものが多くなるため $1\times1/2+2\times1/4+3\times1/8+\cdots=\Sigma i/2^i$ と短縮されるが，計算量としては 2^n 以上必要である．

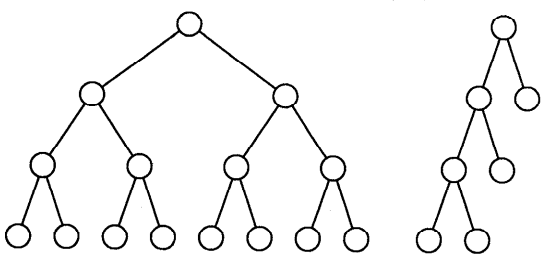

図1 バランスした二進木とバランスしていない二進木
Fig. 1 Balanced and unbalanced binary trees.

ところが全探索では，節点総数である $2n+1$ だけで済んでしまう．

したがって，純粋に解を多数求めるためには，降下によるよりも全探索の方が効率が高いことが分かる．しかも，その差は木がバランスしていないほど大きくなる．

これまでに述べてきた問題における探索木は，降下の途中で解でないことが判明する場合が多い．冒頭で述べた，順列（組合せ）の生成とテストというアルゴリズムでいえば，順列（組合せ）の一部を生成した段階でテストが可能になる場合が多いことに着目する．つまり，探索対象の木が，すべての枝が深さ n に達しているわけではない（つまり，バランスしていない）ことが多いのである．

3.4 解の密度の推定

解の総数とは別に，この方法を用いることで，解の密度を推定することもできる．従来のモンテカルロ法によるならば，解の密度は解への到達回数を試行回数で割れば求まる．しかし，このことが意味するのは，密度に反比例する試行回数が必要となる点である．

複合モンテカルロ法において解密度を求めるには，カットオフレベル以下の全探索領域での解の密度と，その開始点となる中間節点への到達率の積の期待値を計算すればよい．これらはいずれも小さな値ではあるが，その積で与えられる解密度よりは大きいので，計算に要する試行回数は少なくて済む．

なお，解の総数とは異なり，解の密度は探索アルゴリズムに依存する．探索空間の大きさが枝刈りの性能によって変化するためである．以下でいう解密度は，すべて実験に用いた特定の探索アルゴリズムでの値であることを指摘しておく．

4. 実　　験

本章では2つの例題を選び，複合モンテカルロ法の効果を実験によって確かめる．

まず正確な解の総数が分かっている問題に対して適用することで，アルゴリズムの正しさの立証や近似の精度の評価などが可能になる．次に難しい問題に適用することで，有効性を示すことにする．

4.1 19 Queens

19個の女王を用いた女王問題である．19という値の根拠は，手元の計算機で全探索を行った結果，正確な解の総数が求まった最大の n が19であったためである（ちなみに，求めた $|Q(n)|$ を**表1**に示しておく）．

計算条件は，SparcStation 5上で，カットオフレベルごとに乱数の初期値を10通り用いて，それぞれを

表1 $|Q(n)|$ の厳密値
Table 1 Exact values of $|Q(n)|$.

n	$Q(n)$	n	$Q(n)$
4	2	12	14200
5	10	13	73712
6	4	14	365596
7	40	15	2279184
8	92	16	14772512
9	352	17	95815104
10	724	18	666090624
11	2680	19	4968057848

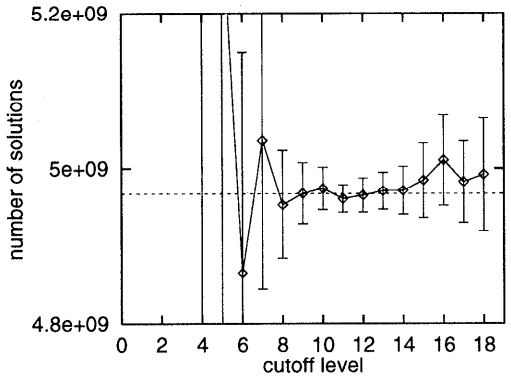

図2 $Q(19)$ の解総数の見積り
Fig. 2 Estimated number of solutions of $Q(19)$.

CPU 時間を 60 秒ずつ使用した．全探索が完了していない部分については，解を求める処理に算入していない．カットオフレベルが低い領域で，解の近似値が求まっていないのはそのためである．なお，乱数発生は Unix のライブラリ関数 **random** によっている．

まず，解総数の近似値を**図 2** に示す．乱数の初期値を変えて同じ計算を 10 回行った結果であり，実線は得られた解総数の平均値で，その標準偏差も示してある．水平の破線は全探索によって求めた正確な値である．

この結果から，かなり正確な近似が得られていることが分かる．特に標準偏差はカットオフレベルが 11 のあたりで最小となっており，最も良い近似が得られている．

計算の際に得られた各種のデータをプロットしたのが**図 3** である．グラフの横軸はカットオフレベルであり，5 種類のデータは以下の値を示している（10 通りの乱数初期値での合計である）：

(A) 試行回数
(B) 全探索の回数（完了したもの）
(C) 解を 1 つ以上得た全探索の回数
(D) 得られた解の総数（完了した全探索のみ）
(E) 得られた解の総数（未完の全探索も含む）

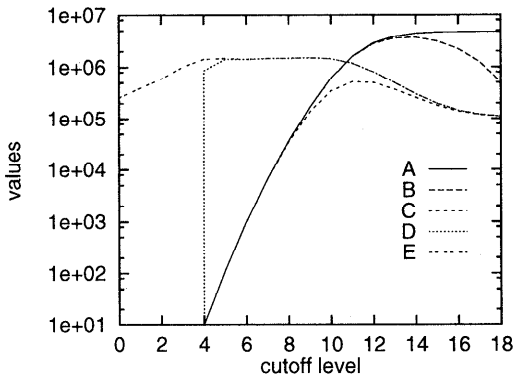

図3 $Q(19)$ の試行回数，発見解数など
Fig. 3 Various measurement for $Q(19)$.

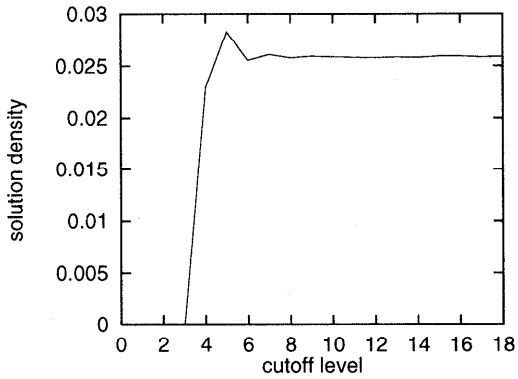

図4 $Q(19)$ の解密度の見積り
Fig. 4 Estimated solution density of $Q(19)$.

この結果から以下の傾向が分かる：

- カットオフレベルが大きい（深い）と，試行回数（A）そのものは多いが全探索に入るまで降下できるもの（B）の割合が低く，結果として求まる解数（D），全探索の成功回数（C）は少ない．
- カットオフレベルが小さく（浅く）なるにつれ，全探索に入る割合（B）は増加し，求まる解の数（D）も増えていく．しかしそれとともに，1 回の試行の所要時間も増大するため試行回数（A）が下がっていく．
- 全探索の回数（B）が減少に転じた後（カットオフ 14 以下）も，そのうちで解を得るもの（C）は増加し続ける．これはカットオフ 11 でピークに達する．
- カットオフレベルがある限界（この結果では 4）以下になると，用意した時間では全探索が一度も終了しなくなる．

図 4 は解密度の推定の結果である．推定値はカットオフには依存せず 0.026 程度である．実際，この推定値は，カットオフが 18 の場合（通常のモンテカル

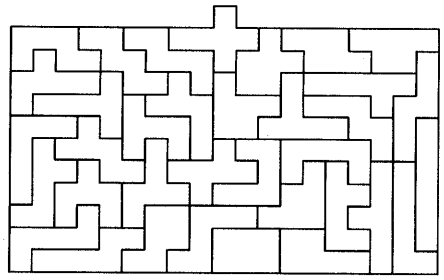

図5 Hexomino の解の一例
Fig. 5 A solution of Hexomino.

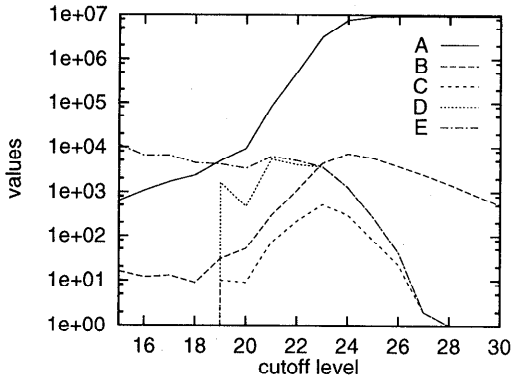

図6 Hexomino の試行回数，発見解数など
Fig. 6 Various measurement for Hexomino.

ロ法に相当）における試行回数と発見した解数の比（$110872/4763567 = 0.0233$）とかなりよく一致している．

4.2 Hexomino

Hexomino とは，一種の箱詰めパズルである．正方形6個から構成される駒（合計35個）をたとえば**図5**のように箱につめるもので，(株) テンヨーからプラパズル No.600 として発売されている．

同様のもので正方形5個の駒のものは Pentomino と呼ばれ，探索問題の例としてしばしば取り上げられている．また，解の総数（対称性を除外して2339）も全探索によって容易に求めることが可能である．

しかし，Hexomino はそれよりもはるかに難しく，これまでに解の総数については近似値さえ知られていなかった．

こちらの計算条件は，日本電算機 JP4 (PowerPC) 上で，$Q(19)$ 同様にカットオフレベル，乱数初期値を変更して，それぞれ CPU 時間 600 秒で行った．

探索回数などのプロットを**図6**に示す．図3と同様の傾向があることが分かるが，さらに条件が厳しくなっている．全探索が成功完了しているのはカットオフレベルが19から28の範囲のみで，それを外れると

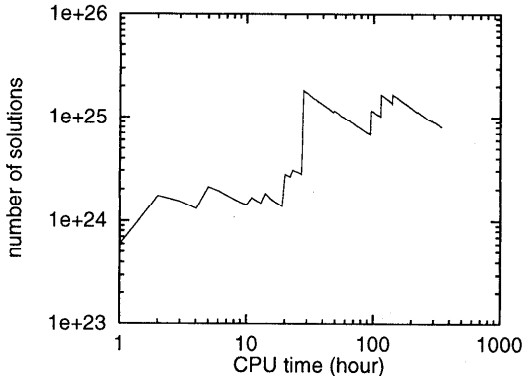

図7 Hexomino の解総数の見積り
Fig. 7 Estimated number of solutions of Hexomino.

全解数の近似値は求まらない．

全解数の近似のためにはもっと多くの計算時間を必要とするので，前の実験とは別に，カットオフレベルを適当と思われる22に固定し，長時間計算を行った．その結果を**図7**に示す．横軸は CPU 時間（単位は hour），縦軸は解総数の近似値である．両軸とも対数スケールとしてある．ばらつきが大きいが，全解数としては 10^{25} 程度と見積もることができよう．

解の密度については，図7の計算と同時に求めた結果，およそ $2.6 * 10^{-10}$ となった．

5. 考　　察

5.1 試行効率の評価

まず，有効な試行数の観点から，複合モンテカルロ法の効率を評価してみる．

Hexomino において，実験的に求めた解密度値を用いて，単純なモンテカルロ法を利用した場合に，解を見つけるために必要な時間を見積もってみる．図6においてカットオフレベルが高い領域では，6000秒の CPU 時間でおよそ 10^7 回の試行が可能であったから，解を1つ見つけるのに必要な時間は $2.3 * 10^6$ 秒（約27日）となる．

これに対して，複合モンテカルロ法では，図6の実験では，カットオフレベルが23のとき，6000秒で517回の全探索に成功（つまり1個以上の解を発見）しているので，有意な試行数の観点から，約200000倍の効率があるということができる．

5.2 複合モンテカルロ法の位置付け

複合モンテカルロ法は，従来のモンテカルロ法と全探索を連続的につなぐものとして考えることができる．

一方の極端として，カットオフレベルを0とすれば，木の全体に対し全探索を繰り返すことになる（この場合，毎回同じ値が得られるので繰返しは無意味で

あるが).

逆にカットオフレベルを木の深さに一致させると,葉に至るまでランダム降下を行うことになり,従来のモンテカルロ法となる.

ここで重要なのはそれらの中間において最良の結果を得ることができる点である.

また,ここでは木の探索問題に限ったが,もっと広い範囲に適用可能ではないかと考えている.本質的な点は,1回の乱数を用いた試行で多数のサンプルを吟味することが,多数回の試行に比べて計算コスト的に優位でありさえすれば,適用可能であることである.

5.3 カットオフレベルについて

以下でカットオフレベルについて考察する.

まずはじめに,複合モンテカルロ法の近似精度が全探索成功数によって決まるという仮説をおく.この仮説を補強するものとして,この値が通常のモンテカルロ法での有効な試行数に対応するものであること,また $Q(19)$ における実験結果(図2の標準偏差と図3の曲線Cの対応)などがあげられる.したがって,単位時間あたりの全探索成功数を最大化するものが適正なカットオフレベルということになる.

複合モンテカルロ法において,全探索1回を完了するまでの試行群をサイクルと呼ぶことにする.1サイクルは,カットオフレベルに到達するまでの多数回のランダム降下と,それに引き続く全探索からなる.前者の所要時間はカットオフレベルが深くなるにつれて指数的に増加し,逆に後者は指数的に減少する.したがって,これらの和である1サイクルの所要時間は,カットオフレベルが適当な中間の値をとるときに最小となることが期待できるといえる(実際には,全探索がすべて成功するわけではないので,それを考慮すると,全探索領域が大きい場合の方が有利になり,最適カットオフレベルを浅い方へいくぶんかシフトすることになる).

カットオフレベルの決定法については,現状では予備実験による以外のアイデアはない.乱数発生と全探索の計算コストの比率などが分からないし,そもそも対象としている探索木の構造が分かっていないことが事前の予想を困難にしている.現実的な対応策としては,試行を行いながら,カットオフレベルを適応的に自動調整していくのがよいと考えている.

6. 結 論

探索木の解の総数を見積もる問題に対して,モンテカルロ法が困難なほど解の密度が低い場合に,全探索と組み合わせることで有効な試行を増やす方法を提案した.さらに,2種類の問題に対してそれを適用し,その有効性を示した.

参 考 文 献

1) Sosic, R. and Gu, J.: 3,000,000 Queens in Less Than One Minute, *ACM SIGART Bulletin*, Vol.2, No.2, pp.22–24 (1991).
2) Kanada, Y. and Hirooka, M.: Stochastic Problem Solving by Local Computation Based on Self-organization Paradigm, *IEEE 27th Hawaii International Conference on System Sciences*, pp.82–91 (1994).
3) Bernhardsson, B.: Explicit Solutions to the N-Queens Problems for all N, *ACM SIGART Bulletin*, Vol.2, No.2, p.7 (1991).
4) Motwani, R. and Raghavan, P.: *Randomized Algorithms*, Cambridge University Press (1995).

(平成8年9月17日受付)
(平成9年6月3日採録)

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年同大学院工学系研究科情報工学修士課程修了.同大学計数工学科助手,電気通信大学電子情報学科助手を経て,1991年東京大学工学部機械情報工学科講師,1992年同助教授.工学博士.情報処理学会,ソフトウェア科学会,ACM各会員.