

## ネットワーク信頼度計算に対するモンテカルロ法と BDD を用いた厳密法の計算機実験を通じた考察

関根京子 今井浩

東京大学理学系研究科情報科学専攻

**アブストラクト:** ネットワーク信頼度計算の問題は、一般に#P 完全な問題で、大規模ネットワークの信頼度を厳密に実用的時間内で求めることは難しいと思われる。そのため、近似解法として、信頼度の上限下限を多項式時間で求めること、またモンテカルロ法で確率的に近似解を求めることが考えられてきた。後者について、計算量・アルゴリズム理論的観点から、近年、ランダム化 (randomization) を用いたランダム化 FPTAS (Randomized Fully Polynomial-Time Approximation Scheme) が開発されている。一方、関根、今井は、BDD を援用したネットワーク信頼度の厳密求解法を提案している。本稿では、ネットワーク信頼度計算に対するランダム化 FPTAS 解法 (モンテカルロ法) と BDD を援用する厳密解法の 2 つについて、後者で得たある程度大きいグラフの厳密な信頼度関数とモンテカルロ法で得た値の比較などを行ない、両者の種々の場面での性質について論じる。

### Considerations Through Computational Experiments of Monte Carlo Methods and Exact Methods by BDDs for Network Reliability Computation

Kyoko Sekine and Hiroshi Imai

Department of Information Science, University of Tokyo, Tokyo 113, Japan

**Abstract:** Computing the network reliability is a #P-complete problem, and is believed hard to solve if the problem size is large. Recently, randomized fully polynomial-time approximation schemes for computing the network reliability have been developed by Alon, Frieze, Welsh [1] and Karger [15]. There are also proposed several enhanced Monte Carlo methods which utilize lower and upper bounds for the reliability, etc. On the other hand, Sekine and Imai [24] propose an exact approach based on BDDs. By this method, the reliability of a moderate-size network can be computed rigorously. They also present a polynomial-time algorithm for the case of complete graphs. This paper performs computational experiments, and discusses features of these methods.

### 1 はじめに

グラフ  $G$  の各枝が確率  $p$  で独立に消滅するとき、残った枝からなる部分グラフが連結である確率を、このグラフの全端子信頼度  $R(G; p)$  と呼ぶ。これは無向グラフでの全端子間 1 連結度の基本問題で、関連問題としてグラフが有向である場合の到達可能性に関するもの、2 端子間の連結性に限ったもの、連結度を上げたものなど様々な場合が考えられている。元々はシステム解析などでの動機からグラフの問題として定式化され、組合せ論における一つの深い分野を構成している。たとえば、ネットワーク信頼度に関する本として [7, 11, 26] があり、またグリードイドに関する本 [5, 17] の 1 節は有向グラフでの 1 点から他の全点への到達可能性

に関する信頼度の記述がある。

計算量理論の観点からは、ネットワーク信頼度計算の問題は、一般に#P 完全な問題である。Valiant [27] は全端子信頼度・2 端子信頼度の問題の#P 完全性をその#P 完全性に関する最初の論文で示している。Provan [23] は、平面性や無有向閉路性などのもとでも 2 端子間信頼度計算が#P 完全であることを示している。以上のことから一般に、大規模ネットワークの信頼度を厳密に実用的時間内で求めることは難しいと思われる。そのため、厳密解法で組合せ爆発が起こる部分を少しでも改善しようという研究、近似解法として信頼度の上限下限を多項式時間で求めること、またモンテカルロ法で確率的に近似解を求めることが考えられ

てきた。

厳密解法は基本的に展開法であり、代数的な性質を考慮すること [26] や、よりよい展開公式を導出すること [12] などが行なわれている。

信頼度の上限・下限を多項式時間で求める方法については、古くは Kruskal-Katona 限界というのが知られていたが、それより常によりよい Ball-Provan 限界 [4] が有名である。他に、翁長上界と呼ばれる上界が厳密解を与える場合の解析や、フローから構成される下界に関する研究もなされている (Nagamochi, Ibaraki [20, 21])。

ランダム化を用いた近似解法については、計算量・アルゴリズム理論的観点から、80 年代に Karp, Luby [16] によって平面グラフで枝消滅確率が極端に小さくない場合に対するランダム化 FPTAS (Randomized Fully Polynomial-Time Approximation Scheme) が開発されていた。近年、ランダム化 (randomization) を用いたランダム化 FPTAS が Alon, Frieze, Welsh [1], Karger [15] によって開発されている。また、それ以前から様々な付加情報を利用したモンテカルロ法の効率化が、Fishman [9, 10] によって提案されており、全域木のランダム生成に基づいた信頼度計算法が Nel, Colbourn [22] によって示されている。

一方、関根、今井 [24] は、BDD を援用したネットワーク信頼度の厳密求解法を提案している。これは、展開法による厳密解法で、信頼度関数が同じマイナーが繰り返して現れることに着目し、それらのマイナーを同定して計算過程の中で 1 つにまとめるというものである。もともとは、関根ら [25] によって提案されているアプローチに基づくもので、VLSI CAD の論理設計などで注目されている 2 分決定グラフ (Binary Decision Diagram; BDD と略す) [6] をグラフなどの組合せ論に適用し、その際、単に既存の BDD アルゴリズムを用いるのではなく、トップダウン解法というこの場合に有効な手法を開発し、BDD のサイズ解析など通例理論的解析が難しかったものを新手法で解析したものである。このアルゴリズムの計算量は、そのグラフがよい消去点順をもつかどうかという性質に依存して述べられており、特にグラフのクラスとして分離定理が成り立つ場合 ([25] にも上げられているように [18, 2] 参照) には次の結果が得られている。

- $O(n^\alpha)$  分離可能な  $n$  点グラフのクラス ( $0 < \alpha < 1$ ) に対して、 $O(n^{O(n^\alpha)})$  時間アルゴリ

ズム

- $n$  点平面グラフに対して、 $O(2^{O(\sqrt{n})})$  時間アルゴリズム

これにより、単純な展開法では解けなかったサイズ (たとえば 15 点弱までの密なグラフや  $12 \times 12$  点くらいまでの格子グラフ) の問題が解けるようになってきている。

また、同じ考えに基づいているともいえるが、関根が修士論文で参照している Annan [3] の方法を信頼度に適用して、完全グラフの信頼度が多項式時間で計算できることも示している。

本稿では、ネットワーク信頼度計算に対する方法の内、ランダム化 FPTAS 解法 (モンテカルロ法) と BDD を援用する厳密解法について、後者によって 50 点の完全グラフや  $9 \times 9$  格子グラフの厳密な信頼度関数を求め、それとモンテカルロ法で得られた値を比較するなどして、両者の種々の場面での性質について論じる。それぞれのアプローチには利点・欠点があり、それらを明らかにすることを旨とする。

## 2 ネットワーク信頼度

点集合  $V$ 、枝集合  $E$  のグラフ  $G = (V, E)$  に対して、枝消滅確率が同一の値  $p$  の場合の全端子間信頼度  $R(G; p)$  を、各枝が確率  $p$  で独立に消滅するとき、残った部分グラフが連結である確率と定義される。式で表すと、

$$R(G; p) = \sum_A \left(1 - \frac{1}{p}\right)^{|A|} p^{|E-A|}$$

となる。ここで、和はすべての全域部分グラフの枝集合  $A \subset E$  についてとっている。各枝  $e$  の消滅確率  $p(e)$  が同一でない場合も考えられる。その一般の場合で、信頼度については次の削除縮約展開式が成り立つ。

$$R(G) = \begin{cases} (1 - p(e))R(G/e) & (e: \text{橋}) \\ R(G \setminus e) & (e: \text{自己閉路}) \\ p(e)R(G \setminus e) + (1 - p(e))R(G/e) & \text{他} \end{cases}$$

ここで、 $\setminus$  は枝の削除を、 $/$  は枝の縮約を意味し、橋とはその枝を除くとグラフのランクが 1 減るものである。

数式処理システム Maple では、spanpoly という関数が networks パッケージの中に用意されており、信頼度関数が計算できる。用いられている

アルゴリズムについては未確認であるが、おそらく上の展開式を用いたものと思われる。ここでは展開式に基づく計算法が爆発する例として、その spanpoly で完全グラフの信頼度関数を求めるのに要した時間のグラフを図1に示す。

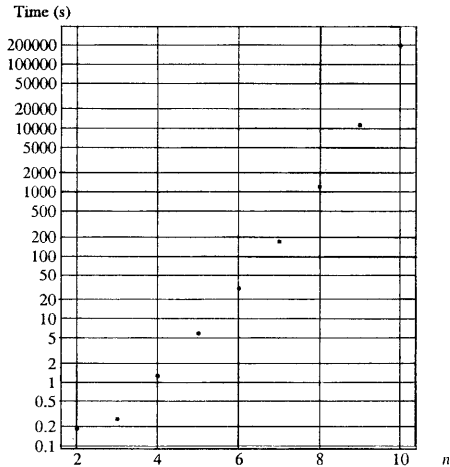


図1: 完全グラフ  $K_n$  の信頼度関数計算での spanpoly の計算時間 (SS10 ワークステーション)

一方、関根、今井 [24] は、BDD を援用したネットワーク信頼度の厳密求解法を提案している。これは、展開法による厳密解法で、信頼度関数が同じマイナーが繰り返し現れることに着目し、それらのマイナーを同一化して計算過程の中で1つにまとめるというものである。この手法は、特に小さな分離集合をもつグラフのクラスに対して効率がよい。ここでは、そのようなグラフの典型例である  $k \times k$  格子グラフの信頼度計算結果のグラフを図2示す。このように、点数で100弱、枝数で200弱のグラフの信頼度を正確に計算することができる。

### 3 完全グラフの全端子間信頼度の多項式時間計算

$n$  点の完全グラフ  $K_n$  で枝消滅確率が全て同一の場合は、Annan [3] の方法を適用して、全端子間信頼度を多項式時間で求めることができる [24]。簡単にその方法を述べる。グラフ  $U_{m,r}$  を、 $K_m$  に新しい点を1点加え、その点もとの  $m$  点の間をそれぞれ  $r$  本の並列枝で結んで得られるグラフ

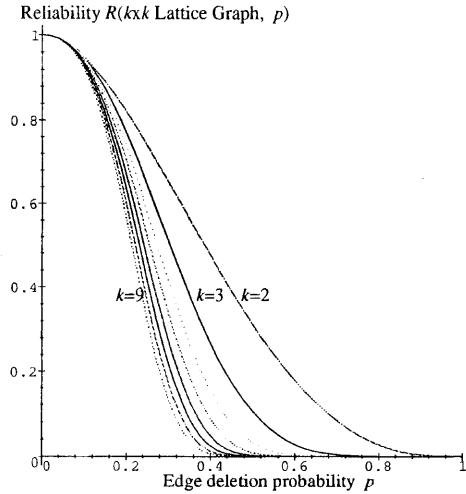


図2:  $R(G_{k,k}; p)$  ( $k = 2, \dots, 9$ )

```

rr :=
proc(n,d) option remember;
local i,s;
  if n=0 then rr(n,d):=1; else
  if n=1 then rr(n,d):=1-p^d; else
  s:=0;
  for i from 1 to n do
    s:=expand(s+binomial(n,i)*(1-p^d)^i*
              p^(d*(n-i))*rr(n-i,i));
  od;
  rr(n,d):=s; fi; fi;
end;

rel:= proc(n)
  rel(n):=sort(normal(rr(n-1,1))); end;

```

図3: 完全グラフの信頼度を計算する Maple プログラム

とする。  $K_n$  は、  $U_{n-1,1}$  である。このとき、

$$R(U_{m,r}; p) = \sum_{i=1}^m \binom{m}{i} (1-p^r)^i p^{r(m-i)} R(U_{m-i,i}; p)$$

が成り立つ。ここで、  $R(U_{0,r}; p) = 1$  である。  $R(K_n; p) = R(U_{n-1,1}; p)$  であり、これにより完全グラフの信頼度を求めることができる。この式により、完全グラフの信頼度関数を求めるプログラムは Maple で図3のように書ける。この Maple プログラムは、展開式に従ったわかりやすく書いたもので、さらなる高速化の余地はかなりある。このプログラムを SparcStation10 ワークステーション、64MB メモリで実行して、  $n = 2$  から 50 まで計算して得ら

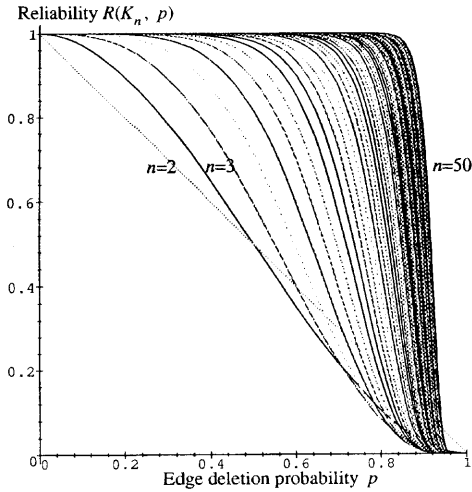


図 4:  $R(K_n; p)$  ( $n = 2, \dots, 50$ )

れた信頼度関数のグラフを図 4 に示す。また、その計算に要した計算時間を両対数グラフで  $n = 10$  から 50 の範囲で図 5 に示す。このグラフで最小 2 乗当てはめを行なうと、当てはめ直線の傾きは約 6.4 となり、この範囲では  $n$  に関して約 6.4 乗の計算時間になっていることが観察される。

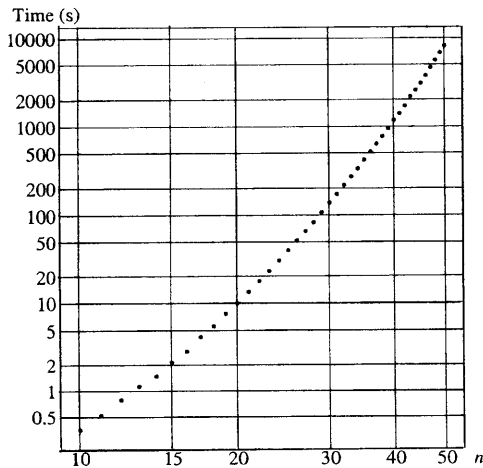


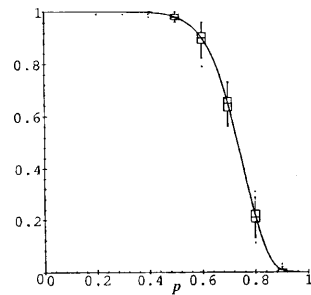
図 5: プログラム rel の計算時間 (SS10)

#### 4 素朴なモンテカルロ法の実験

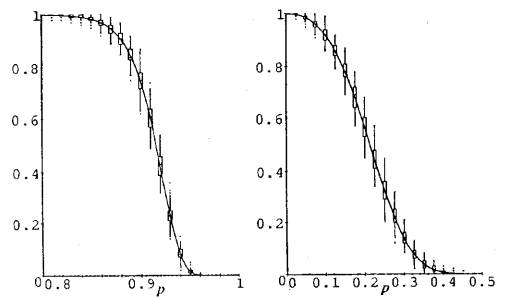
ネットワーク信頼度そのものも確率であることから、信頼度計算と確率的手法とは元々関係している。信頼度計算の最も簡単なモンテカルロ法は、

実際に各枝を独立に確率  $p$  で切断し、残りのグラフが連結な頻度を数えるというものである。実際に、Alon, Frieze, Welsh [1] らが解析しているモンテカルロ法も、この素朴な形そのもので、解析で密なグラフの場合には多項式回の試行で  $\epsilon$ -近似が得られることを示している部分が理論的に興味深いものである (この点については次節で問題点を述べる)。

この節では、完全グラフと格子グラフについて、実際にこの素朴なモンテカルロ法を実装して計算機実験を行なった結果を述べる。本来、モンテカルロ法は大規模な問題で少ない有効桁数の値を求めるのに用いることが多いが、ここでは上述のように完全グラフと格子グラフである程度の大きさのものの信頼度関数が正確に求められているので、それとの比較を行なう。実験は、SparcStation 20, 128MB メモリで行ない、疑似乱数としては M 系列に基づくものを用いた [14]。



(a)



(b)

(c)

図 6: モンテカルロ法の結果 (文中参照)

図 6 に、完全グラフで (a)  $n = 10$ , (b)  $n = 50$  で  $p = 0.8 \sim 1$ , (c)  $9 \times 9$  格子グラフで  $p = 0 \sim 0.5$  の場合を示す。各々、(a) は  $0.1, 0.2, \dots, 0.9$  で、(b) は  $0.8, 0.81, \dots, 0.99$  で、(c) は  $0.025, 0.05, \dots, 0.5$

で初期値  $1, 2, \dots, 100$  で 100 系列乱数を発生させ、各系列での頻度を箱ひげ図でプロットしている。また、正確なグラフも同時に描いている。図より、問題のサイズによらず、 $R(G; p)$  の値で縦軸信頼度の頻度分布が決定されているのが見てとれる。

## 5 ランダム化 FPTAS での近似値比の意味

近似アルゴリズムの良さの尺度としては、通常、最適解と近似解の値の比 (近似値比) が用いられる。数え上げ問題の場合、求めるべき厳密な数と近似解法の生成する数の比を考えることになる。また、ランダム化を用いる近似アルゴリズムの場合は、まずは近似値比の期待値がよいことが望まれ (期待値として不偏推定量であることが望まれる)、さらに高い確率で近似値比がある幅に入るといった評価が望まれる。

ネットワーク信頼度の場合、ある値  $p$  でのモンテカルロ法などランダム化を用いる近似解法の値を  $\tilde{\mu}(p)$  とすると、厳密値  $\mu(p) \equiv R(G; p)$  に対して

$$\Pr[\mu(p)(1 - \epsilon) \leq \tilde{\mu}(p) \leq \mu(p)(1 + \epsilon)] \geq 1 - \delta$$

が成り立つとき、この解が  $(\epsilon, \delta)$ -近似解であるという。これは相対誤差に着目している。

一方、信頼度  $R(G; p)$  そのものも確率であるから、素朴なモンテカルロ法で行なっているのは確率  $R(G; p)$  で成功 (1; 失敗は 0) するベルヌーイ試行を繰り返して頻度を計算しているものである。1 回の試行での分散は  $R(G; p)(1 - R(G; p))$  であり、 $N$  回の試行での成功回数  $X$  の和の分散はその  $N$  倍である。チェビシェフの不等式によりこのモンテカルロ法で得られる頻度  $\tilde{\mu}(p)$  は、

$$\Pr \left[ |\tilde{\mu}(p) - \mu(p)| \leq \sqrt{\frac{\mu(p)(1 - \mu(p))}{\delta' N}} \right] \geq 1 - \delta'$$

となる。すなわち、ネットワークの大きさと関係なく、単に  $\mu(p)$  に依存するだけで試行回数  $N$  を増やしていくことにより、 $O(1/\sqrt{N})$  で絶対誤差を小さくしていける。

この相対誤差と絶対誤差の 2 つの基準で、顕著な違いができるのは  $\mu(p)$  が小さい場合である。この場合、絶対誤差の観点ではよくても、相対誤差の観点では非常に悪いということが起こる。このような問題点を一般のモンテカルロ法の枠組の中で

とらえた論文に、Dagum, Karp, Luby, Ross [8] がある。そこでは、ある程度  $\mu(p)$  を推定することによって、 $\mu(p)$  が小さい場合にあたる場合でも相対誤差の観点でよい推定値を少ない標本数で得るアルゴリズムが論じられている。

信頼度近似計算のランダム化 FPTAS についても状況は同じで、Alon, Frieze, Welsh [1] の論文ではグラフが密な場合に素朴なモンテカルロ法で  $\mu(p)$  が小さい場合に対応する部分を回避できることを示しており、Kargar [15] はグラフが密でないときでその  $\mu(p)$  が小さい場合に対応する部分問題に対して、Karp, Luby [16] が平面グラフの場合でとったような DNF 論理式の充足解近似数え上げへの還元を示し、全体として信頼度関数に対するランダム化 FPTAS を構成している。

一方、信頼度計算を確率計算でなく、数え上げととらえると、相対的評価のよさが浮き出てくる場合もある。 $2^{|E|} R(G; 1/2)$  はグラフ  $G$  の全域部分グラフの数を与える (この計算も #P 完全)。従って、密なグラフでは素朴なモンテカルロ法で相対比のよい数え上げができるが、格子グラフだとより複雑な対処が必要となる (図 2 より少し大きな格子グラフでは  $R(G; 1/2) \approx 0$  である)。

これらは、近似解の評価基準として何を用いるかでアルゴリズムのよしあしの評価も変わってくることをいっている。Dagum ら [8] らでは、さまざまな論文を参照して、絶対誤差でなく相対誤差でよい近似アルゴリズムが望まれるとされているが、多くの参考文献は数え上げに関する理論的な論文である。このような点については、単に理論の分野でよいとして導入された尺度をそれだけで独立して使うだけでなく、実際の問題に立ち帰って応用に基づいた評価もまた必要になっていることを意味しているかもしれない。

## 6 考察

本稿では、#P 完全問題であるネットワーク信頼度計算について、中規模なサイズでも厳密に問題を解くことの意義、近似的に大規模な問題を解くことの意義、その他周辺のこととがらに関して、実験を行なった結果について述べてきた。

応用の観点からは、厳密解法にこだわる立場については

- 本当に厳密な値を計算する必要があるのか、単に上位数桁だけではだめなのか

ということに答える必要がある。これに対しては、

- 組合せ論での不変多項式計算プログラムとしては、多項式として厳密なものを求める意義がある。信頼度の上限・下限を求めるアプローチなどを援用して多項式を補間によって求めることも考えられているが、多項式の係数を正確に求めることは難しいことがわかっている [11].
- 関根らの厳密解法では、全体の信頼度が正確に計算できるとともに、1つの枝の重要度のような解析も厳密に全く同じ枠組で行なうことができる。
- 各枝の消滅確率が異なる場合、出力サイズそのものが指数オーダーとなる可能性がある。各枝の消滅確率が同一の場合、出力サイズは多項式サイズであるが、線形サイズというわけではない。従って、桁数も数十桁などといった係数が出てくる。その数十桁がさらに数百桁となるような場合は計算できないと思われるが、一方でそのようなサイズまで実際に必要になることはないのではないかとと思われる。
- 信頼度関数は、全域部分グラフの母関数と密接な関係がある (実際、上述したように Maple では spanpoly という名前の関数になっている)。関連して、森の母関数なども同様に厳密に計算できる。このような点でも、また、上述のように他に代替がない点で、理論的に中規模な問題でも厳密な解を求める方法としての意義は高い。

また、最後の点に関しては、関根らは結び目・絡み目の Jones 多項式を求めることもこの枠組の拡張で行なえることを示しており、そこではさらに厳密に解を求めることの意義が高いと思われる。具体的に、100 から 200 の交差点の絡み目の Jones 多項式の係数は数十桁になり、たとえば不変量の Jones 多項式の厳密な比較によって結び目が結ばれているかどうかの判定や、2つの絡み目が同型性の判定については、知見を得るためには厳密性の必要度がかかなり高い。

一方、応用の観点から見た近似解法については、

- 有効桁数が数桁で実際にいいのか。また、多項式の各点での近似値だけで実際にシステム解析をしたことになるのか。
- という点を考察する必要があるが、ランダム化 FPTAS についてはさらに
- 近似値比の観点でよくなければならないとい

#### う要請の意義

について検討が必要である。少なくとも、モンテカルロ法による近似解では (多項式時間では) 決して厳密解が得られず、

- 数え上げ問題でランダム化 FPTAS が開発されたとしても、一方で中規模の問題まで厳密に解く解法

の重要性は減るものではない。実際、他の最適化問題に対する FPTAS などについては、その理論的意義は高いものの、実際の応用性についてはあまりないものもある。

将来の研究方向性として大切と思われるのは、

- #P 完全など難しい数え上げ問題などに対して、これまでの手法では解けなかったサイズの問題で実際に解く要請がある問題を解けるようにする

ことである。数え上げの問題の例では、たとえば行列のパーマネントを求める問題 (2部グラフの完全マッチングを数え上げる問題) について、この問題に対してはまだランダム化 FPTAS は得られていない (密などの条件をつけるがあるとある)。Jerrum, Vazirani [13] は、それまでの  $O(\epsilon^{-2} \text{poly}(n) 2^{n/2})$  時間のアルゴリズムに替えて、 $\epsilon^{-2} \exp(O(\sqrt{n} \log^2 n))$  時間アルゴリズムを示している。もちろん、相変わらずこれでも指数爆発しているだけに、それだけに指数関数の中のオーダー記法で隠れている定数が問題となるが、少なくともこれは理論的解析を伴った上での上のような問題点解決の一例といえる。

関根ら [24, 25] の信頼度関数やグラフの Tutte 多項式その他の不変量 (たとえば Welsh [28] 参照) に対するアプローチもこのような問題意識での BDD とそのトップダウン構成法を土台とした 1つの統一のアプローチである。このアプローチは BDD と強く関係しているが、

- 多くの数え上げ問題が DNF 論理式の充足解数え上げ問題に帰着され、ランダム化 FPTAS の観点からはその DNF 式に対するもの (Motwani, Raghavan [19] により解説がある) を用いている、
  - 一方、DNF 式など一般の論理式の充足解全体をコンパクトに表現する BDD は、厳密解法であり、VLSI CAD の論理設計の分野などで従来解けなかったサイズの問題を解くことを可能にしている、
- ことを考えれば、計算量理論・アルゴリズム理論

の観点からはさらなる BDD の理論的解析が望まれるともいえる。

## 謝辞

本研究の一部は、文部省科学研究費の援助を受けた。

## 参考文献

- [1] N. Alon, A. Frieze and D. Welsh: Polynomial Time Randomised Schemes for the Tutte Polynomial of Dense Graphs. *Proceedings of the 35th IEEE Annual Symposium on Foundations of Computer Science*, 1994, pp.24-35.
- [2] N. Alon, P. Seymour and R. Thomas: A Separator Theorem for Graphs with an Excluded Minor and Its Applications. *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, 1990, pp.293-299.
- [3] J. D. Annan: *The Complexity of Counting Problems*. PhD Thesis, University of Oxford, 1994.
- [4] M. O. Ball and J. S. Provan: Calculating Bounds on Reachability and Connectedness in Stochastic Networks. *Networks*, Vol.13 (1983), pp.253-278.
- [5] A. Björner and G. M. Ziegler: Introduction to Greedoids. In “*Matroid Applications*” (N. White, ed.), Encyclopedia of Mathematics and Its Applications, Vol.26, Cambridge University Press, 1992, pp.284-357.
- [6] R. E. Bryant: Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Transactions on Computers*, Vol.C-35, 1986, pp.677-691.
- [7] C. J. Colbourn: *The Combinatorics of Network Reliability*. Oxford University Press, 1987.
- [8] P. Dagum, R. Karp, M. Luby and S. Ross: An Optimal Algorithm for Monte Carlo Estimation. *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, 1995, pp.142-149.
- [9] G. S. Fishman: A Monte Carlo Sampling Plan for Estimating Network Reliability. *Operations Research*, Vol.34, No.4 (1986), pp.581-594.
- [10] G. S. Fishman: A Monte Carlo Sampling Plan for Estimating Reliability Parameters and Related Functions. *Networks*, Vol.17 (1987), pp.169-186.
- [11] D. D. Harms, M. Kraetzl, C. J. Colbourn and J. S. Devitt: *Network Reliability: Experiments with a Symbolic Algebra Environment*. CRC Press, Inc., 1995.
- [12] Y. Higashiyama, H. Ariyoshi, I. Shirakawa and S. Ohba: A Factored Reliability Formula for Directed Source-to-All-Terminal Networks. *IEICE Transactions on Fundamentals*, Vol.E77-A, No.1 (1994), pp.134-143.
- [13] M. Jerrum and U. Vazirani: A Mildly Exponential Approximation Algorithm for the Permanent. *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science*, 1992, pp.320-326.
- [14] K. Kaneko: C code of a pseudorandom generator using M-sequences. Department of Mathematical Engineering, University of Tokyo.
- [15] D. R. Karger: A Randomized Fully Polynomial Time Approximation Scheme for the All Terminal Network Reliability Problem. *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, 1995, pp.11-17.
- [16] R. M. Karp and M. Luby: Monte-Carlo Algorithms for the Planar Multiterminal Network Reliability Problem. *Journal of Complexity*, Vol.1 (1985), pp.45-64.
- [17] B. Korte, L. Lovász and R. Schrader: *Greedoids*. Algorithms and Combinatorics, Vol.4, Springer-Verlag, 1991.
- [18] R. J. Lipton and R. E. Tarjan: A Separator Theorem for Planar Graphs. *SIAM Journal on Applied Mathematics*, Vol.36, No.2 (1979), pp.177-189.
- [19] R. Motwani and P. Raghavan: *Randomized Algorithms*. Cambridge University Press, 1995.
- [20] H. Nagamochi and T. Ibaraki: Maximum Flows in Probabilistic Networks. *Networks*, Vol.21 (1991), pp.645-666.
- [21] H. Nagamochi and T. Ibaraki: On Onaga's Upper Bound on the Mean Values of Probabilistic Maximum Flows. *IEEE Transactions on Reliability*, Vol.41, No.2 (1992), pp.225-229.
- [22] L. D. Nel and C. J. Colbourn: Combining Monte Carlo Estimates and Bounds for Network Reliability. *Networks*, Vol.20 (1990), pp.277-298.
- [23] J. S. Provan: The Complexity of Reliability Computations in Planar and Acyclic Graphs. *SIAM Journal on Computing*, Vol.15, No.3 (1986), pp.694-702.
- [24] K. Sekine and H. Imai: A Unified Approach via BDD to the Network Reliability and Path Numbers. *Technical Report 95-09*, Department of Information Science, University of Tokyo, 1995.
- [25] K. Sekine, H. Imai and S. Tani: Computing the Tutte Polynomial of a Graph of Moderate Size. *Proceedings of the 6th International Symposium on Algorithms and Computation (ISAAC'95)*, Lecture Notes in Computer Science, Vol.1004, 1995, pp.224-233.
- [26] D. R. Shier: *Network Reliability and Algebraic Structures*. Oxford University Press, 1991.
- [27] L. G. Valiant: The Complexity of Enumeration and Reliability Problems. *SIAM Journal on Computing*, Vol.8, No.3 (1979), pp.410-421.
- [28] D. J. A. Welsh: *Complexity: Knots, Colourings and Counting*, London Mathematical Society Lecture Note Series, Vol.186, Cambridge University Press, 1993.

表 1: 格子グラフ  $L_{k \times k}$  の信頼度関数  $R(L_{k \times k}; p)$

$k$	Reliability Polynomial $R(L_{k \times k}; p)$
2	$-3p^4 + 8p^3 - 6p^2 + 1$
3	$79p^{12} - 560p^{11} + 1668p^{10} - 2656p^9 + 2331p^8 - 960p^7 + 96p^5 + 21p^4 - 16p^3 - 4p^2 + 1$
4	$-17493p^{24} + 232144p^{23} - 1409764p^{22} + 5168576p^{21} - 12693232p^{20} + 21854512p^{19} - 26726036p^{18} + 22824576p^{17} - 12739373p^{16} + 3710880p^{15} + 139672p^{14} - 370176p^{13} - 35464p^{12} + 63968p^{11} + 5912p^{10} - 7808p^9 - 1791p^8 + 656p^7 + 204p^6 + 64p^5 - 8p^4 - 16p^3 - 4p^2 + 1$
5	$32126211p^{40} - 681809240p^{39} + 6852471548p^{38} - 43322118652p^{37} + 192968405711p^{36} - 642590690400p^{35} + 1655933457966p^{34} - 3370276114636p^{33} + 5476061558391p^{32} - 7122774813980p^{31} + 7375859530466p^{30} - 5981426876044p^{29} + 3667377815630p^{28} - 1573096624396p^{27} + 375423772810p^{26} + 9584416484p^{25} - 26112103320p^{24} - 6268146140p^{23} + 8011274210p^{22} - 1051500660p^{21} - 575028980p^{20} - 53196700p^{19} + 139031550p^{18} - 2265380p^{17} - 10705120p^{16} - 3593556p^{15} + 1357510p^{14} + 394172p^{13} + 35042p^{12} - 49636p^{11} - 10290p^{10} - 2036p^9 + 1021p^8 + 164p^7 + 250p^6 + 64p^5 - 11p^4 - 20p^3 - 4p^2 + 1$
6	$-486300871185p^{60} + 15050120960512p^{59} - 224457589381592p^{58} + 2147283930748936p^{57} - 14795904565154568p^{56} + 78164407095437208p^{55} - 329002388286585590p^{54} + 1131851787890708080p^{53} - 3239023808802078612p^{52} + 7805237324574469672p^{51} - 15970377879462783332p^{50} + 27889821900794421800p^{49} - 41669380085176682975p^{48} + 53245515820123890488p^{47} - 57994363781668793396p^{46} + 53459688480842369480p^{45} - 41189240095530447551p^{44} + 25971548039388094736p^{43} - 12912733359090221520p^{42} + 4700587401338060528p^{41} - 1030543095219489046p^{40} + 27339920836923904p^{39} + 30104608114327902p^{38} + 24389988686930648p^{37} - 21944946692702846p^{36} + 4681984115585944p^{35} + 931332145706116p^{34} - 204889051003888p^{33} - 296523309399185p^{32} + 94978228313328p^{31} + 20359609248584p^{30} - 5426001755888p^{29} - 4782579009711p^{28} + 786788558176p^{27} + 490461884648p^{26} + 42254441256p^{25} - 73196765980p^{24} - 10696424680p^{23} + 3437463142p^{22} + 2543567456p^{21} + 171986440p^{20} - 197718184p^{19} - 82482716p^{18} - 4922680p^{17} + 8093647p^{16} + 2670936p^{15} + 29356p^{14} - 175832p^{13} - 32353p^{12} - 4464p^{11} - 2624p^{10} - 2848p^9 - 550p^8 + 464p^7 + 370p^6 + 88p^5 - 22p^4 - 24p^3 - 4p^2 + 1$
7	$60466776520638151p^{84} - 2566705488866684184p^{83} + 53040005163375903596p^{82} - 710815048826712842372p^{81} + 6943483260122343346359p^{80} - 52680737958496550589136p^{79} + 323019935820016291236038p^{78} - 1644459499066017820038324p^{77} + 7086299348458292174686483p^{76} - 26220080670230614971950248p^{75} + 84212085619245470010448770p^{74} - 236724692337260223377654004p^{73} + 586132154792766933559738760p^{72} - 1284404126149925625624249800p^{71} + 2499523892569447005899179386p^{70} - 4329584748500299979015577252p^{69} + 6683012649228490250707395320p^{68} - 9193058126395018151636708536p^{67} + 11257033469333923761153205836p^{66} - 12240474845110397000080991300p^{65} + 1177092949871930055897750204p^{64} - 9948788108767798014087601380p^{63} + 7323266356513525319985839080p^{62} - 4631531857742897336377049988p^{61} + 2465012485934922602463615595p^{60} - 1067506473705984972533536596p^{59} + 354257889681938546016425400p^{58} - 79658135765962056355135324p^{57} + 9252277883955156622553503p^{56} - 1842856626562190545916460p^{55} + 2727939278480959787717768p^{54} - 1739223608155121871078156p^{53} + 493323545187094722105929p^{52} - 28098638458008672903324p^{51} - 1290570704703452129384p^{50} - 18614691083128506334284p^{49} + 10285156800612171209641p^{48} - 1048073996842678022980p^{47} - 446421497537673602588p^{46} - 125766662125684989388p^{45} + 150807831795395894245p^{44} - 11963401863024697092p^{43} - 9306352364545626068p^{42} - 2686230399955160100p^{41} + 2256435435583309099p^{40} + 84242703814280228p^{39} - 107195984543376372p^{38} - 71196218713769940p^{37} + 15287965178027703p^{36} + 6467128267421284p^{35} + 888229195483320p^{34} - 1003171094179660p^{33} - 142395942978545p^{32} + 7700222945972p^{31} + 38802911274472p^{30} + 2459402034228p^{29} - 619304181479p^{28} - 1324675381276p^{27} - 141206363784p^{26} + 29575288556p^{25} + 50196440989p^{24} + 5896476188p^{23} - 224198680p^{22} - 1912695012p^{21} - 215763548p^{20} - 34223068p^{19} + 57336572p^{18} + 10667104p^{17} + 257364p^{16} - 1838524p^{15} - 15482p^{14} + 76752p^{13} + 104260p^{12} - 13540p^{11} - 12254p^{10} - 7800p^9 - 431p^8 + 948p^7 + 522p^6 + 104p^5 - 35p^4 - 28p^3 - 4p^2 + 1$