

直並列グラフを利用した all-terminal reliability の下界導出法

大阪大学工学部 小出 武 (Takeshi Koide)
鹿児島大学理学部 新森 修一 (Shuichi Shinmori)
大阪大学工学部 石井 博昭 (Hiroaki Ishii)

1 はじめに

ネットワークシステムの信頼度を問題にする場合、ネットワークシステムを点や枝に確率の概念を付加した確率グラフと捕らえることができる。ネットワークシステムの信頼度を測る尺度の一つである総合信頼度 (all-terminal reliability) とは、確率グラフ中の全ての点が正常に機能している枝によって連結されている確率である。一般に、総合信頼度の値を求めるのに要する計算時間は枝の本数に対し指数的に増加する (NP -困難) ので、精度の良い境界値、特に下界を多項式時間で求めることが重要になる ([3])。従来より総合信頼度の下界を多項式時間で導出することを目的とした研究が数多くなされている ([2][3] など)。ところがそれらの研究で対象とするネットワークシステムは、ネットワーク中の全ての枝について、正常に機能する確率が同一であることを必要条件としている。逆に言えば、正常に機能する確率が異なる枝がネットワーク中に存在する場合には適用できない。

我々は極大木によるエッジパッキングを用いて総合信頼度の下界を多項式時間で導出する方法を提案した ([4][5])。この方法は、ネットワーク中に正常に機能する確率が異なる枝が存在する場合にも適用できるという特徴を持つ。本稿では新たに直並列グラフによるエッジパッキングを用いて下界を導出する方法を提案する。

2 準備

要素数 n の点集合 $\{v_1, \dots, v_n\}$ 、要素数 m の枝集合 $\{e_1, \dots, e_m\}$ からなる無向グラフを $G = (V, E)$ とする。グラフ中の各枝は正常に機能している状態 (正常状態) と故障している状態 (故障状態) の2つの状態があり、枝 $e \in E$ が正常である確率 (枝正常確率) を p_e とし、各枝の枝正常確率は互いに独立とする。また点は常に正常であると仮定する。このような確率グラフ G 中の全ての点が正常な枝によって連結になるとき、 G で表現されるネットワークは正常であるという。ネットワークが正常である確率を総合信頼度 (all-terminal reliability: 全点信頼度とも言う) と呼び、 $Rel(G)$ で表す ([3])。

以降使用する記号についてここで定義する。グラフ $G = (V, E)$ に枝 e を追加したグラフを $G \cup \{e\} = (V, E \cup \{e\})$ 、枝 e を削除したグラフを $G - \{e\} = (V, E - \{e\})$ と記す。またグラフ $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ に対し、グラフ G_2 中にある枝をグラフ G_1 から除去したグラフを $G_1 - G_2 = (V_1, E_1 - E_2)$ と記す。

3 エッジパッキングによる $Rel(G)$ の下界

定義 1 グラフ $G = (V, E)$ に対し、 G と同じ点集合 V を有し、互いに枝排反な G の部分

グラフの集合

$$\{G_i | G_i = (V, E_i), E_i \subset E, E_i \cap E_j = \emptyset (i \neq j)\}$$

をエッジパッキング (edge-packing) という。

定理 1 グラフ $G = (V, E)$ が要素数 k のエッジパッキング $\{G_1, \dots, G_k\}$ をもつとき、次の不等式が成立する [3]。

$$(1) \quad Rel(G) \geq 1 - \prod_{i=1}^k (1 - Rel(G_i))$$

定理 1 から、 $Rel(G_i)$ を多項式時間で計算できるようなエッジパッキングを多項式時間で構築できれば、 $Rel(G)$ の下界を多項式時間で求めることができることがわかる。また下界の精度を上げるには、

1. 各 $Rel(G_i)$ の値を大きくする。
2. エッジパッキングを構成する部分グラフの数 k を大きくする。

の 2 つの戦略があることがわかる。

4 極大木によるエッジパッキング

グラフ $G = (V, E)$ のエッジパッキング $\{G_1, \dots, G_k\}$ の各要素を極大木にして構成した場合、 $G_i = (V, E_i) (i = 1, \dots, k)$ とすると、(1) 式は次のように表せる。

$$(2) \quad Rel(G) \geq 1 - \prod_{i=1}^k (1 - \prod_{e \in E_i} p_e)$$

(2) 式より極大木からなるエッジパッキングを多項式時間で構築することができれば、総合信頼度の下界を多項式時間で導出することができる。我々は極大木からなるエッジパッキング $EPST$ を構成する 2 つのアルゴリズムを提案した ([4][5])。以下にその概要を示す。

アルゴリズム 1-1

- 1 $EPST \leftarrow \emptyset$;
- 2 グラフ G から総合信頼度最大の極大木 T を発見;
- 3 $EPST \leftarrow EPST \cup \{T\}$;
- 4 $G \leftarrow G - T$;
- 5 G が非連結なら終了。連結なら Step2 へ;

アルゴリズム 1-2

Edmonds による Matroid Partition[1] を利用して、グラフ G から最大個数の枝排反極大木を発見し、 $EPST$ とする;

アルゴリズム 1-1 は 3 節で述べた戦略 1、アルゴリズム 1-2 は戦略 2 に基づいたアルゴリズムである。アルゴリズム 1-1 の計算オーダーは $O(\max(m \log m, cm))$ 、アルゴリズム 1-2 の計算オーダーは $O(m^3)$ である。ここで c はグラフ G の枝連結度である。

5 直並列グラフによるエッジパッキング

定義 2 点数 4 の完全グラフと同相なグラフを部分グラフに持たないグラフを直並列グラフ (*series-parallel graph*) という。

ここで以下に示す 3 つの確率的に等価なグラフ変換 [4] [5] を考える。

[短絡除去] : グラフ $G = (V, E)$ 中の次数 1 の点を v 、点 v を端点とする唯一の枝を e とする。このとき、 $G' = (V - \{v\}, E - \{e\})$ とすると、

$$(3) \quad Rel(G) = p_e Rel(G')$$

である。このグラフ変換を短絡除去という。

[直列縮退] : グラフ $G = (V, E)$ 中の次数 2 の点を v 、点 v を端点とする 2 本の枝を $e_1 = (v, u_1) \in E$, $e_2 = (v, u_2) \in E$ とする。このとき、 $e' = (u_1, u_2) \notin E$, $G' = (V - \{v\}, E - \{e_1, e_2\} \cup \{e'\})$ とすると、

$$(4) \quad Rel(G) = \{1 - (1 - p_{e_1})(1 - p_{e_2})\} Rel(G')$$

$$(5) \quad p_{e'} = \frac{p_{e_1} p_{e_2}}{1 - (1 - p_{e_1})(1 - p_{e_2})}$$

である。このグラフ変換を直列縮退という。

[並列縮退] : グラフ $G = (V, E)$ 中の 2 点 $u, v \in V$ を両端点とする 2 本の並行枝を $e_1, e_2 \in E$ とする。このときを $e' = (u, v) \notin E$, $G' = (V, E - \{e_1, e_2\} \cup \{e'\})$ とすると、

$$(6) \quad Rel(G) = Rel(G')$$

$$(7) \quad p_e = 1 - (1 - p_{e_1})(1 - p_{e_2})$$

となる。このグラフ変換を並列縮退という。

以上の 3 つのグラフ変換を用いると、定義 2 は以下のように変形できることが知られている。

定義 3 短絡除去、直列縮退、並列縮退の 3 つのグラフ変換を繰り返し可能な限り施すことによって、孤立点 1 点に変換できるグラフを直並列グラフと呼ぶ。

この定義から以下のことが言える。

定理 2 直並列グラフの総合信頼度は、計算オーダー $O(m)$ で求めることが可能である。

(証明) グラフに短絡除去、直列縮退、並列縮退のいずれを適用しても枝が 1 本減少する。従って直並列グラフを孤立点 1 点まで変換するためには、グラフ変換を m 回適用すればよい。孤立点 1 点からなるグラフの総合信頼度は 1 なので、元の直並列グラフの総合信頼度を計算することが可能となる。 $\square$

従って極大木によるエッジパッキングと同様、直並列グラフによるエッジパッキングを多項式時間で構築すれば、元のグラフの総合信頼度の下界を多項式時間で計算することが可能である。

直並列グラフが元のグラフの全ての点を含まない場合、その総合信頼度は0となるので、(1)式より、エッジパッキングの要素にする意味がないことがわかる。そこで全ての点を含む直並列グラフ（以下全域直並列グラフと呼ぶ）によるエッジパッキング $EPSP$ を構成するアルゴリズムを提案する。

まず3節で述べた戦略1を重視したアルゴリズムを考える。このときグラフ G の部分グラフで、その総合信頼度を最大とする全域直並列グラフを見つけることが必要となるが、この問題はまだ解かれていない。そこで代替として、以下に示すアルゴリズムを提案する。

アルゴリズム 2-1

```

1   $EPSP \leftarrow \phi$ ;
2   $E$  中の枝を枝正常確率の大きい順にソート;
3   $T \leftarrow$  グラフ  $G$  中の総合信頼度最大の極大木;
4   $G \leftarrow G - T$ ;
5  for 各  $e \in E$  ( $p_e$ の大きい順) do
6    begin
7      if  $T \cup \{e\}$  が直並列グラフ then
8        begin
9           $T \leftarrow T \cup \{e\}$ ;
10          $G \leftarrow G - \{e\}$ ;
11       end
12    end
13   $EPSP \leftarrow EPSP \cup \{T\}$ ;
14  if  $G$  is connected goto Step 3;
```

アルゴリズム 2-1 はアルゴリズム 1-1 を発展させたものである。具体的には、以下のことを行っている。

1. G 中から総合信頼度最大の極大木 T を発見する。
2. T が直並列グラフである範囲内で、 $G - T$ の枝を枝確率の大きい順に T に追加する。
3. T を $EPSP$ に追加、 $G = G - T$ とする。
4. G が非連結なら終了。そうでないなら Step1 へ。

アルゴリズム 2-1 中に行っている、直並列グラフ T に1本の枝 e を追加してもなお直並列グラフであるか否かの判断は、定理2より $O(m)$ の時間を必要とする。アルゴリズム 2-1 自体の計算オーダーについては、次の定理が成立する。

定理 3 グラフ $G = (V, E)$ にアルゴリズム 2-1 を適用したときの計算オーダーは $O(cm^2)$ である ($c: G$ の枝連結度)。

(証明) 枝のソートには $O(m \log m)$ 必要である。Step 5 からのループの繰り返し回数は m 回以下で、1 回のループごとに直並列グラフの判断に $O(m)$ かかる。また Step 14 から Step 3 へ戻る回数は、最終的に得られるエッジパッキングの要素数に等しいので c 以下になる。従って、

$$O(\max(m \log m, cm^2)) = O(cm^2).$$

□

次に 3 節で述べた戦略 2 を重視したアルゴリズムを考える。極大木は最も枝本数の少ない全域直並列グラフだから、全域直並列グラフによるエッジパッキングの要素数の最大値は、極大木によるエッジパッキングの要素数になる。この結果を利用したアルゴリズムを以下に示す。

アルゴリズム 2-2

```

1  EPSP = {G1, ..., Gk} ← アルゴリズム 1-2 で得られる EPST;
2  G1, ..., Gk をその総合信頼度の大きい順にソート;
3  G ← G - ∪i=1k Gi;
4  G 中の枝を枝正常確率の大きい順にソート;
5  for i = 1, ..., k do
6    begin
7      for 各 e ∈ E (pe の大きい順) do
8        begin
9          if Gi ∪ {e} が直並列グラフ then
10           begin
11             Gi ← Gi ∪ {e};
12             G ← G - {e};
13           end
14         end
15     end

```

アルゴリズム 2-2 は、以下のことを行っている。

1. G 中から極大木による要素数最大のエッジパッキングを獲得する。
2. エッジパッキングで使用していない枝を枝確率の大きい順に、エッジパッキングの要素に直並列グラフである範囲内で追加する。
追加する要素の対象は総合信頼度の大きいものを優先する。

アルゴリズム 2-2 の性能については次の定理が成立する。

定理 4 アルゴリズム 2-2 で得られた全域直並列グラフによるエッジパッキング EPSP を用いて導出する総合信頼度の下界は、アルゴリズム 1-2 で得られる極大木によるエッジパッキング EPST を用いて導出する総合信頼度の下界より大きい。

(証明) アルゴリズム 2-2 はアルゴリズム 1-2 の結果得られた EPST の要素に枝を追加して EPSP を構成する。EPST = {G₁, ..., G_k} とし、G_i に枝を追加してできた全域直並列

グラフを G'_i とすると、 $EPSP = \{G'_1, \dots, G'_k\}$ となり、

$$Rel(G_i) \leq Rel(G'_i) \quad (i = 1, \dots, k)$$

となるので、(1) 式から $EPST$ による下界より $EPSP$ による下界の方が大きくなる。□

アルゴリズム 2-2 の計算オーダーについては、次の定理が成立する。

定理 5 グラフ $G = (V, E)$ にアルゴリズム 2-2 を適用したときの計算オーダーは $O(m^3)$ である。

(証明) Step 1 には $O(m^3)$ の計算時間が必要である。Step 2 でのソートや Step 4 でのソートに必要な時間は明らかに $O(m^3)$ より小さい。Step 5 での for 文は k 回の繰り返し、Step 7 での繰り返し回数は高々 m 回である。直並列グラフの判定には $O(m)$ 必要なので、Step 5 からのループ全体では、 $O(km^2)$ となる。明らかに $k < m$ なので、全体では $O(m^3)$ となる。□

6 数値結果

アルゴリズム 2-1 と 2-2 を図 1 で示した Arpanet (点数 59、枝数 71) に対し適用し、現段階で最適と言われている Ball-Provan の手法と比較した。ただし Ball-Provan の手法は全ての枝確率が等しい場合のみしか適用できないので、枝確率を全て等しいと仮定した。

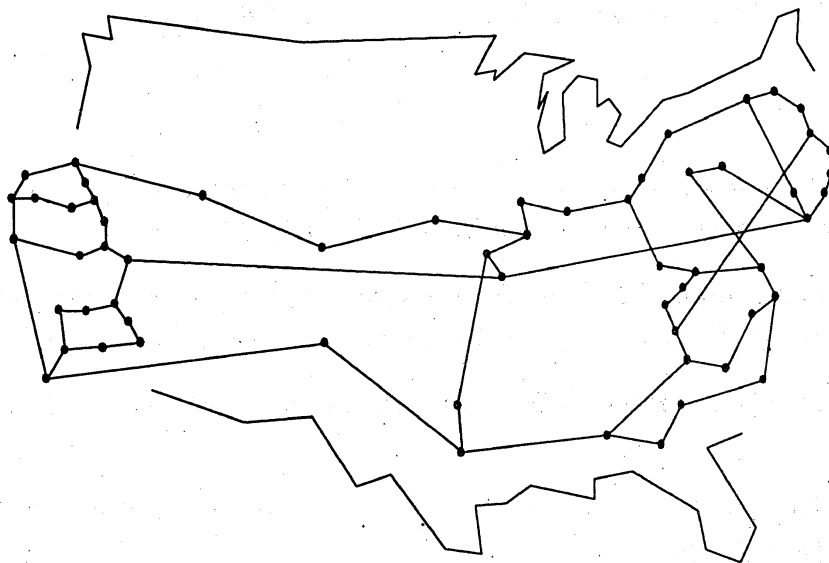


図 1: Arpanet(1979)

図 2 に結果を示した。Arpanet に対しアルゴリズム 2-1, 2-2 を適用した結果、2 つの手法による下界は等しくなった。図中では総合信頼度を Rel 、Ball-Provan による手法を B-P、本稿で提案した手法を EPSP と記した。図 2 より、Ball-Provan の手法より本稿で提案した手法による方が精度の良い下界を得ることができた。

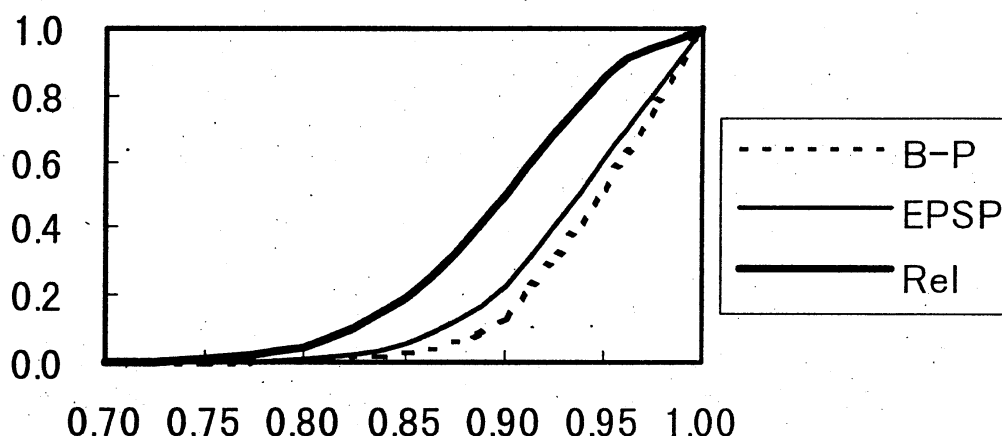


図 2: Arpanet の総合信頼度と下界

7 まとめ

本稿では直並列グラフによるエッジパッキングを用いて総合信頼度の下界を多項式時間で求めるアルゴリズムを2種類提案した。総合信頼度の下界を求めるアルゴリズムのほとんどが各枝の枝正常確率が同一でないと適用できないのに対し、このアルゴリズムは各枝の枝正常確率が同一でない場合にも適用できるという利点がある。

参考文献

- [1] Edmonds, J., Minimum partition of a matroid into independence subsets, *Journal of Research of the National Bureau of Standards*, 69B(1965), 67-72.
- [2] Ball, M.O., and Provan, J.S., Bounds on the reliability polynomial for shellable independence systems, *SIAM Journal on Algebraic and Discrete Methods*, 3(1982), 166-181.
- [3] Colbourn, C.J., *Combinatorics of Network Reliability*, Oxford University Press, New York, 1987.
- [4] 新森, 小出, 石井, エッジ・パッキングによるネットワーク信頼度の下界, 日本応用数理学会論文誌, 5(2) 139-151(1995)
- [5] Koide, T., Shinmori, S., Ishii, H., Lower bounds of all-terminal reliability by packing and graph transformations, to be submitted.