

## サンプリングと二部決定図を用いたネットワーク信頼性の近似計算 Approximate Network Reliability Computation with Sampling and BDD

佐々木 勇和<sup>†</sup> 藤原 靖宏<sup>‡†</sup> 鬼塚 真<sup>†</sup>  
Yuya Sasaki<sup>†</sup> Yasuhiro Fujiwara<sup>‡†</sup> Makoto Onizuka<sup>‡</sup>

### 1. まえがき

世の中の理解と設計のためには、オブジェクト間の関係性をモデル化し分析することが必要である。実世界のオブジェクト間の関係性は曖昧性を保持しているため、曖昧グラフによりモデル化することが有用である。曖昧グラフにおける節点と枝はそれぞれオブジェクトと関係に対応し、関係の曖昧性は枝の存在確率で表される。曖昧グラフにおける重要な問題の一つにネットワーク信頼性問題がある [4, 9, 29]。ネットワーク信頼性は曖昧グラフにおいて複数のターミナル（一部の節点）間が接続する確率である。ネットワーク信頼性は様々なアプリケーションで利用されている。例えば、タンパク質相互反応は気温や湿度等の状況により常に反応が行われるわけではなため、タンパク質相互反応ネットワークは曖昧グラフを用いてモデル化することができる [3, 15]。タンパク質の機能の解明のために、分析者は複数のタンパク質間の関係の強さをネットワーク信頼性を用いて評価している。その他にも、通信ネットワーク [4, 25] や都市計画 [11] などの広いドメインにてネットワーク信頼性は利用されている。

ネットワーク信頼性の計算は #P 完全問題として知られており、計算コストは非常に大きい。ネットワーク信頼性計算のための単純な方法は、まずすべての可能グラフを列挙する。可能グラフは、元々の曖昧グラフと同じ節点および枝のサブセットを持つグラフである。可能グラフは存在確率をもち、枝の存在確率から計算される。それぞれの可能グラフに対して、与えられたターミナルが全て接続となる場合、その可能グラフの確率がネットワーク信頼性に加算される。図 1 は曖昧グラフと 3 つの可能グラフの例を示す。図のグラフにおいて黒の点はターミナルを示す。枝の存在確率は 0.7 とする。全ての可能グラフにおいて 4 つの枝が存在枝（2 つが非存在枝）であるため、可能グラフの存在確率は  $0.0216 (0.7^4 \cdot (1 - 0.7)^2)$  である。左と中央の可能グラフのみターミナルが接続しているため、これらの存在確率がネットワーク信頼性に加算される。

ネットワーク信頼性を効率的に計算するために様々な手法が提案されている。二部決定図 [2, 5] とサンプリング [8, 16, 17] が最も主要なテクニックである。二部決定図に基づくアプローチは、厳密解を求めるために最も効率的なアプローチである。全ての可能グラフの列挙することを避けることができるが、メモリ使用量が大きく、小規模なグラフにしか適用することができない。サンプリングに基づくアプローチは、近似解を求めるために用いられる。大規模なグラフにも対しても適用できるが、サンプル数とグラフのサイズが大きくなると、計算量が大きくなる。

本研究では、メモリ使用量に制約がある状況でのネットワーク信頼性の近似計算について考える。問題定義を下記に示す。

問題定義: (メモリ制約があるネットワーク信頼性近似計算)。曖昧グラフ  $G$ , ターミナル集合  $\mathbb{T}$ , サンプル数  $s$ , およびメモリ領域が与えられたとき、ネットワーク信頼性を効率的に近似計算する。

提案アプローチは二部決定図とサンプリングを統合し、それ

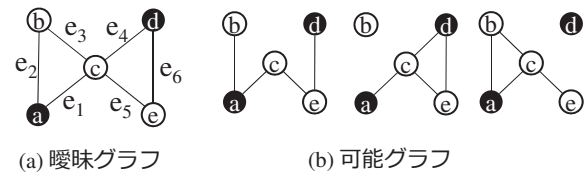


図 1: 曖昧グラフと可能グラフ

ぞれの長所とサンプリングの効率性と精度の向上を達成する。二部決定図とサンプリングは計算手順が大きく異なるため、場合には技術的な難しさがある。単純に両方の手法を統合する場合、まず二部決定図に基づくアプローチを用い、もし二部決定図に基づくアプローチがメモリ不足の場合に、サンプリングに基づくアプローチを用いる。しかし、この単純な方法では、近似解を用いるために二部決定図を構築する余分な時間がかかってしまう。技術課題は、二部決定図とサンプリングに基づく手法を同時に実行しつつ、厳密解および近似解の計算時間を小さくすることである。

新たな二部決定図である  $S^2BDD$  (scalable and sampling BDD) を提案する。もしメモリ使用量が与えられたメモリ量より大きくなった場合、提案アプローチは  $S^2BDD$  しながら可能グラフをサンプリングする。 $S^2BDD$  はターミナルの接続確率と非接続確率を保持するため、ネットワーク信頼性の上限値と下限値を計算することができる。上限値と下限値を用いることにより、精度を損なうことなくサンプル数を小さくすることができる。提案アプローチは、近似解の精度をコントロールでき、さらにメモリ量が十分な場合、厳密解を計算することができる。複数の実データセットを用いて提案手法の性能を検証し、提案手法がサンプリングに基づく手法より 51.2 倍高速、二部決定図に基づく手法より 575 倍高速であることを示す。

本稿では、2. にて関連研究を紹介し、3. 事前知識について説明する。4. にて提案アプローチをせつめいする。5. にて実験結果を示し、6. にて本稿をまとめる。

### 2. 関連研究

ネットワーク信頼性: カットに基づくアプローチ [1, 13, 20], Factoring に基づくアプローチ [27, 28, 30], 二部決定図に基づくアプローチ [12, 21, 31] など、様々なアプローチがネットワーク信頼性計算のために提案されている。これらの中で、二部決定図に基づくアプローチが最も効率的である。二部決定図に基づくアプローチは、まず二部決定図を構築し、二部決定図を辿ることでネットワーク信頼性を計算する。近年の研究において、メモリ使用量が大きいため、枝数が 200 程度のグラフにしか適用できないことが報告されている [12, 14, 21]。筆者らの知る限り、最も効率的な二部決定図構築のライブラリである TdZDD\* においても、枝数が 200 程度のグラフにしか適用できない。Herrmann と Soh は、メモリ効率が良い二部決定図を提案している [14]。この二部決定図は、ネットワーク信頼性の計算に不必要な部分を削除することでメモリ使用量を抑えている。しかし、この二部決定図でも大規模なグラフには適応できない。提案アプローチは、メモリ使用量を削減するために、この二部決定図のアイデアを一部用いる。二部決定図を用いたネットワーク信頼性の近似計算方法は、筆者らの知る限り提案

<sup>†</sup>大阪大学大学院情報科学研究科

Grad. Sch. of Information Science and Technology, Osaka University

<sup>‡</sup>NTT ソフトウェアイノベーションセンター

NTT Software Innovation Center

\*<https://github.com/kunisura/TdZdd>

されていない。

**曖昧グラフにおける他の問題:** 曖昧グラフに対する問合せおよびマイニングは数多く提案されている。多くの問題はサンプリングを用いて解かれているため、効率性と精度はサンプリング技術に依存している。Jin ら [16] は、与えた閾値よりも大きい確率で接続しているサブグラフを発見するアルゴリズムを提案した。このアルゴリズムではモンテカルロサンプリングが用いられ、サンプル数を Chernoff-Hoeffding bound を用いて決定している。Ceccarello ら [6] は曖昧グラフのクラスタリング技術を提案した。この技術では節点間の距離をネットワーク信頼性を用いて定義している。このアルゴリズムにおいてもモンテカルロサンプリングによりネットワーク信頼性を計算している。Khan らは [19] は、与えられた閾値以上の確率である節点から接続している節点集合をみつける検索を提案した。インデックス RQ-tree を提案し、効率的に節点集合を計算している。この手法においてもモンテカルロサンプリングを用いている。多くのアルゴリズムにおいてモンテカルロサンプリングを用いてネットワーク信頼性を計算しているため、サンプリングに基づくアプローチを改善することにより、多くのアルゴリズムの精度と効率性を向上させることができる。

### 3. 事前知識

提案アプローチの事前知識として、曖昧グラフおよびネットワーク信頼性について説明する。加えて、二部決定図とサンプリングを用いたネットワーク信頼性計算の方法について説明する。

#### 3.1 曖昧グラフ

連結かつ無向の曖昧グラフを  $G = (\mathbb{V}, \mathbb{E}, p)$  と定義する。  $\mathbb{V}$  は節点集合、  $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$  は、曖昧枝の集合、および  $p: \mathbb{E} \rightarrow (0, 1]$  は、枝  $e \in \mathbb{E}$  の存在確率  $p(e)$  を決定する関数である。節点  $v$  と  $v'$  間の枝  $e \in \mathbb{E}$  を  $e = (v, v')$  と定義する。枝  $e$  の状態は、確率  $p(e)$  で存在または確率  $1 - p(e)$  で非存在である。枝の存在確率は、他の枝の存在確率と独立に計算されると想定する [6, 17]。

可能グラフ  $G_p = (\mathbb{V}, \mathbb{E}_p)$  は、曖昧グラフ  $G$  の節点全てと枝のサブセットから成る。  $\mathbb{E} \setminus \mathbb{E}_p$  は非存在枝である。可能グラフは存在確率を  $Pr[G_p]$  をもち、以下の式で計算される。

$$Pr[G_p] = \prod_{e \in \mathbb{E}_p} p(e) \cdot \prod_{e \in \mathbb{E} \setminus \mathbb{E}_p} (1 - p(e)). \quad (1)$$

曖昧グラフ  $G$  の可能グラフの数は、それぞれの枝が存在と非存在の二つの状態を取りうるため、  $2^{|\mathbb{E}|}$  である。  $G$  の全ての可能グラフの集合を  $\mathbb{W}^G$  と定義する。

中間グラフ  $G_{\mathbb{E}(\mathbb{E}_\exists, \mathbb{E}_-)}$  は、存在枝集合  $\mathbb{E}_\exists$ 、非存在枝集合  $\mathbb{E}_-$ 、および曖昧枝の集合  $\mathbb{E} \setminus (\mathbb{E}_\exists \cup \mathbb{E}_-)$  から構成される曖昧グラフである。中間グラフの存在確率  $Pr[G_{\mathbb{E}(\mathbb{E}_\exists, \mathbb{E}_-)}]$  は以下の式で計算される。

$$Pr[G_{\mathbb{E}(\mathbb{E}_\exists, \mathbb{E}_-)}] = \prod_{e \in \mathbb{E}_\exists} p(e) \cdot \prod_{e \in \mathbb{E}_-} (1 - p(e)). \quad (2)$$

単純化のため、  $Pr[G_{\mathbb{E}(\mathbb{E}_\exists, \mathbb{E}_-)}]$  の代わりに  $Pr[G_{\mathbb{E}}]$  を使用する。中間グラフ  $G_{\mathbb{E}(\mathbb{E}_\exists, \mathbb{E}_-)}$  の可能グラフの数は、  $2^{|\mathbb{E} \setminus (\mathbb{E}_\exists \cup \mathbb{E}_-)|}$  である。  $G_{\mathbb{E}}$  の全ての可能グラフの集合を  $\mathbb{W}^{G_{\mathbb{E}}}$  と定義する。中間グラフにおいて、存在枝を用いて節点間に経路がある場合、節点は接続している呼び、存在枝と曖昧枝を用いて経路が無い場合、節点は非接続していると呼ぶ。ここで、曖昧枝を用いて節点に経路がある場合、節点が接続か非接続はあきからではない。

#### 3.2 ネットワーク信頼性

ネットワーク信頼性は、全てのターミナルが接続となっている可能グラフの確率を全て足し合わせることで計算できる。ネットワーク信頼性の定義は以下である。

**Definition 1 (Network reliability)** 任意の  $k$  個のターミナルを含むターミナル集合  $\mathbb{T}$  と曖昧グラフ  $G$  が与えられたと

き、ネットワーク信頼性  $R[G, \mathbb{T}]$  は下記の式で計算される。

$$R[G, \mathbb{T}] = \sum_{G_p \in \mathbb{W}^G} I(G_p, \mathbb{T}) \cdot Pr[G_p], \quad (3)$$

$G_p$  は可能グラフを示し、  $I(G_p, \mathbb{T})$  は  $G_p$  において全てのターミナルが接続になっている場合に 1 を返し、その他の場合 0 を返す関数を示す。

$\hat{R}[G, \mathbb{T}]$  をネットワーク信頼性の近似解と定義する。単純化のため、  $R$  と  $\hat{R}$  を  $R[G, \mathbb{T}]$  と  $\hat{R}[G, \mathbb{T}]$  の代わりにそれぞれ用いる。

ターミナル数が  $k$  であるネットワーク信頼性は  $k$  ターミナル信頼性と呼ばれ、最も一般的なネットワーク信頼性である [12]。  $k$  が 2 であっても、ネットワーク信頼性は #P 完全問題である [29]。平面グラフでは一般的なグラフより、より効率的に計算可能であるが、 #P 完全問題であることは変わらない [26]。

二部決定図 [2, 5] とサンプリング [8, 16, 17] がネットワーク信頼性計算のための主要な技術である。二部決定図に基づくアプローチは小規模なグラフに対して効率的に厳密解を計算可能で、サンプリングに基づくアプローチは大規模なグラフに対して近似解を計算可能である。

##### 3.2.1 二部決定図

二部決定図 (BDD) は有向非循環グラフ  $\mathcal{D} = (\mathbb{N}, \mathbb{A})$  で表現される。二部決定図のノード集合を  $\mathbb{N}$ 、二部決定図のアーキ集合を  $\mathbb{A}$  と定義する<sup>†</sup>。図 2(a) は図 1 のグラフにおける二部決定図を示す。各ノードは中間グラフに対応しており、アーキは存在枝または非存在枝と対応している。二部決定図は、入次数が 0 の枝が一つあり、ルートノードと呼ばれる (図 2(a) における  $G_1$ )。それぞれのノードは、二つの outgoing アークがあり、0-アークおよび 1-アークと呼ばれる (図 2(a) ではそれぞれ点線と実線で表されている)。0-アークおよび 1-アークはそれぞれ非存在枝と存在枝をあらわしている。それぞれのアーキは重みをもち、枝の存在確率または非存在確率をあらわしている。ルートノードからの深さを層  $l$  と定義する。層  $l$  のノードは、枝  $e_1$  から  $e_{l-1}$  が存在または非存在の中間グラフを表す ( $e_1$  から  $e_{|\mathbb{E}|}$  は曖昧枝である)。二部決定図は出次数がゼロのノードが 2 つあり、それぞれ 0-シンクと 1-シンクと呼ばれる (図 2(a) ではそれぞれラベル 0 と 1 の四角で表されている)。中間グラフのターミナルが接続した場合、対応するノードは 1-シンクに遷移する。一方、中間グラフのターミナルが非接続した場合、対応するノードは 0-シンクに遷移する。ルートノードから 1-シンクまでのアーキを辿ることにより、ターミナルが接続となる中間グラフを得ることができる。そのため、1-シンクにアーキが遷移しているノードの確率値の総和を取ることにより、ネットワーク信頼性を計算することができる。

ネットワーク信頼性計算のために二部決定図を構築する方法として、フロンティアに基づく手法がよく用いられている [18]。この手法はまず枝を順序付けし、順序に従って枝の状態を決定していく。二部決定図が既に層  $l$  まで構築済みの場合、この手法は  $e_l$  の状態を決定し、層  $l+1$  の節点集合を構築する。フロンティアに基づく手法では、存在/非存在枝と曖昧枝の両方をもつ節点はフロンティア  $f$  と呼ばれ、層  $l$  でのフロンティアの集合  $\mathbb{F}_l$  で表す。図 2(b) は、枝  $e_1$  と  $e_2$  を処理した後の中間グラフを表し、黒の実線、黒の破線、および灰色の破線はそれぞれ存在枝、非存在枝、および曖昧枝を示す。節点  $b$  と  $c$  は存在/非存在と曖昧枝の両方をもつため、フロンティアである。同じ層のノードは全て同じフロンティア集合をもつ。フロンティアに基づく手法は、フロンティアの変数 (例えば、曖昧枝の数や接続済みのターミナル数など) を保持している。フロンティアの変数が同じになった場合にノードを結合することで、ノードの数を効果的に削除することができる。

<sup>†</sup> 本論文では、グラフの節点と枝を“節点”および“枝”とそれぞれ呼び、二部決定図の節点と枝を“ノード”と“アーク”とそれぞれ呼ぶ。

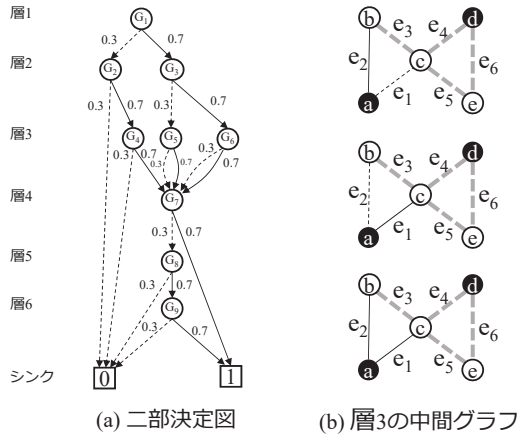


図 2: 図 1(a) のグラフにおける二部決定図。

二部決定図のサイズはノードの数で定義され、メモリ使用量は二部決定図のサイズに依存する [12]。実際のメモリ使用量は事前に推定することができないため、小規模なグラフに対してもメモリが枯渇する可能性がある。

### 3.2.2 サンプリング

サンプリングは近似解を計算するために一般的な方法である [8, 16, 17]。サンプル数  $s$  が与えられたとき、サンプリングに基づくアプローチは次に示す処理を  $s$  回繰り返す。曖昧グラフから可能グラフを取り出し、その可能グラフにおいてターミナルが接続かどうかを計算する。可能グラフを取り出すために  $O(|\mathbb{E}|)$  必要であり、接続の確認のために  $O(|V| + |\mathbb{E}|)$  必要であるため、サンプリングに基づく手法の計算量は  $O(s \cdot (|V| + |\mathbb{E}|))$  である。

サンプリングに基づく手法の精度は分散により評価される。この手法は乱択アルゴリズム [23] であるため、平均のネットワーク信頼性が最も真の値に高確率で近いためである。そのため、分散が小さいとエラー率が小さいことを示す。理論的な分散値を達成するためには、可能グラフをその存在確率に基づいて取り出す不偏サンプリング (unbiased sampling) が必要である。サンプル数が増加すると、分散は減少するが計算量が増加する。そのため、精度と時間計算量の間にはトレードオフが存在する。

## 4. 提案手法

本稿では、与えられたメモリ量の制限下においてネットワーク信頼性の近似問題を解く。4.1 は  $S^2BDD$  について説明する。4.2 は曖昧グラフの前処理について説明する。4.3 にてアルゴリズムの全体について述べる。簡単化のために、 $S^2BDD$  の最大サイズ  $w$  を与えられたメモリ使用量として用いる。

### 4.1 Scalable and Space-efficient BDD: $S^2BDD$

$S^2BDD$  はメモリ効率の向上とサンプル数の効果的な減少を目指して設計する。 $S^2BDD$  のサイズを削減するために、 $S^2BDD$  は 1 層のノード集合しか保持しない。さらに、メモリ枯渇を防ぐために、 $S^2BDD$  のサイズが  $w$  を超えないように、サンプル数の減少に効果的ではないノードを削除する。提案アプローチは削除したノードに対応する中間グラフから可能グラフをサンプルとして取り出す。まず、 $S^2BDD$  を定義し、 $S^2BDD$  の構築方法について述べる。

**Definition 2**  $S^2BDD$  は層  $l$  のノード集合  $N$  および 1-シンクと 0-シンクから構成される。 $S^2BDD$  は下記に示す変数をノード毎  $n \in N$  に保持する:

- $p_n$  :  $n$  が対応する中間グラフの確率。

- 全ての  $f \in \mathbb{F}_l$  における  $\{c_{n,f}\}$  : 接続している節点の識別子。フロンティア  $f$  と  $f'$  が存在枝にて接続している場合、 $c_{n,f}$  と  $c_{n,f'}$  は同じ値である。
- 全ての  $f \in \mathbb{F}_l$  における  $\{d_{n,f}\}$  :  $\{f' \in \mathbb{F}_l | c_{n,f} = c_{n,f'}\}$  となるフロンティアに接続している曖昧枝の総数  $\{f' \in \mathbb{F}_l | c_{n,f} = c_{n,f'}\}$ 。
- 全ての  $f \in \mathbb{F}_l$  における  $\{t_{n,f}\}$  : 存在枝によって  $f$  に接続しているターミナルの数。

1-シンクと 0-シンクはターミナルが接続または非接続となった確率  $p_c$  と  $p_d$  をそれぞれ保持している。

$S^2BDD$  は一層のノード節点しか保持しないため、一般的な二部決定図よりサイズが小さい。加えて、中間グラフではなく、次の層のノードを構築するための最小の情報を保持する。

$S^2BDD$  を構築するために、枝  $e_l$  を処理し、層  $l+1$  のノード集合  $N_{next}$  を生成する。構築方法は 4 つの手順からなる。

生成 :  $N_{next}$  を生成する。

結合 : ノードを結合する。

削除 :  $S^2BDD$  のサイズが  $w$  を超えたとき、ノードを削除する。

サンプリング : 可能グラフをサンプルとして取り出す。

生成と結合の手順は、二部決定図を構築するための一般的な手順である。生成と結合手順を  $S^2BDD$  をよりメモリ効率が良いように拡張する。削除とサンプリングの手順は、 $S^2BDD$  は新たな手順であり、近似解の計算のために実行する。

### 4.2 生成手順と結合手順

二部決定図に基づくアプローチは、二部決定図を構築するために生成と結合手順を実行する。この二つの手順を拡張し、ネットワーク信頼性の精度を損なわずに、 $S^2BDD$  のサイズを効果的に小さくする。

まず、生成手順について説明する。生成手順では、枝  $e_l$  を処理し、層  $l+1$  のノード集合  $N_{next}$  を生成する。基本的な手順と同様に、枝  $e_l$  の状態を決定し、層  $l$  のノードから層  $l+1$  のノードを二つ生成する。新しいノードを生成するために、ノードが保持する変数 (つまり、 $p_n$ ,  $\{c_{n,f}\}$ ,  $\{d_{n,f}\}$ , および  $\{t_{n,f}\}$ ) を更新する。 $\{c_{n,f}\}$ ,  $\{d_{n,f}\}$ , および  $\{t_{n,f}\}$  は新しいノードに対応する中間グラフに応じて更新する。 $p_n$  においては、 $e_l$  が存在枝とする場合、 $p_n \cdot p(e_l)$  とし、非存在枝とする場合、 $p_n \cdot (1 - p(e_l))$  と更新する。中間グラフのターミナルが接続となった場合、 $p_c$  に確率値を足し合わせ、非接続となった場合、 $p_d$  に確率値を足し合わせる。

$S^2BDD$  のサイズを減少させるためには、ターミナルが接続か非接続かを早い段階で判断することが必要である。しかし、 $S^2BDD$  におけるノードはフロンティアの変数しか保持しておらず、中間グラフ全体を保持しているわけではない。そのため、中間グラフにおいてターミナルが接続か非接続かをフロンティアの変数を用いて判断する。下記の定理に基づいて接続および非接続の判断を行う。定理において、 $n$  と  $n'$  は枝  $e_l = (v, v')$  の処理後と処理前のノードをそれぞれ表す。また、同様に  $\mathbb{F}$  と  $\mathbb{F}'$  は  $e_l$  の処理後と処理前のフロンティア集合をそれぞれ表す。

**Lemma 1** 下記の条件を一つでも満たす場合、すべてのターミナル  $t \in \mathbb{T}$  は接続している:

条件 1: 枝  $e_l = (v, v')$  が存在枝の場合、 $\exists f \in \mathbb{F}$  において  $t_{n,f} = k$  となる。

条件 2: 枝  $e_l = (v, v')$  が存在枝の場合、(1)  $v \in \mathbb{F}'$ , (2)  $v' \notin \mathbb{F}' \cup \mathbb{F}$ , (3)  $t_{n',v_1} = k - 1$ , および (4)  $v \in \mathbb{T}$  となる ( $v$  と  $v'$  を入れ替えた場合においても同様)。

条件 3: 枝  $e_l = (v, v')$  が存在枝の場合、(1)  $v, v' \in \mathbb{F}$ , (2)  $c_{n',v} \neq c_{n',v'}$ , および (3)  $t_{n',v} + t_{n',v'} = k$  となる。

*Proof:* 定義より、全てのターミナルの接続はあきらかである。  
□

**Lemma 2** 下記の条件の一つでも満たす場合、ターミナル  $t \in \mathbb{T}$  は非接続している:

条件 1: 枝  $e_l = (v, v')$  が非存在枝の場合、(1)  $v \notin \mathbb{F}' \cup \mathbb{F}$ , および (2)  $v \in \mathbb{T}$  ( $v'$  においても同様) となる。

条件 2: 枝  $e_l = (v, v')$  が非存在枝の場合、(1)  $v \in \mathbb{F}'$ , (2)  $t_{n',v} > 0$ , および (3)  $d_{n',v} = 1$  ( $v'$  においても同様) となる。

条件 3: 枝  $e_l = (v, v')$  が存在または非存在枝の場合、(1)  $v, v' \in \mathbb{F} \setminus \mathbb{F}'$  および (2) ( $t_{n',v} > 0$  or  $t_{n',v'} > 0$ ) となる。

*Proof:* 定義より、ターミナルの非接続はあきらかである。□  
既存の最新の構築方法においても、定理 1 と 2 の条件 1 のみしか用いていない。結果として、 $S^2BDD$  はより効率的にサイズが減少する。

次に、結合手順を説明する。 $S^2BDD$  における中間グラフはそれぞれ異なる存在枝と非存在枝から構成されるため、フロンティアの変数も通常異なる。結合手順では、同じシンクに遷移するノードを結合し、その確率値を足し合わせる。

**Lemma 3** 層  $l$  のノード  $n_1$  と  $n_2$  が与えられたとき、もし  $\forall f \in \mathbb{F}_l$  において (1)  $c_{n_1,f} = c_{n_2,f}$ , および (2) ( $t_{n_1,f} = 0$  and  $t_{n_2,f} = 0$ ) or ( $t_{n_1,f} > 0$  and  $t_{n_2,f} > 0$ ) の場合、 $n_1$  と  $n_2$  は同じシンクに遷移する。

*Proof:* もし  $n_1$  と  $n_2$  が (1) 全ての  $f \in \mathbb{F}_l$  に対して  $\{c_{n_1,f}\} = \{c_{n_2,f}\}$  ならば、 $n_1$  と  $n_2$  のフロンティアは同じフロンティアと接続している。それぞれ  $n_1$  と  $n_2$  から生成されるノード  $n'_1$  は常に  $\{c_{n'_1,f}\} = \{c_{n'_2,f}\}$  となる。従って、シンクに遷移するまで  $\{c_{n_1,f}\}$  と  $\{c_{n_2,f}\}$  は同じ値となる。 $\{c_{n_1,f}\}$  と  $\{c_{n_2,f}\}$  は同じ接続コンポーネントとなるため、 $\{d_{n_1,f}\}$  と  $\{d_{n_2,f}\}$  も同じ値となる。もしフロンティア  $f$  と  $f'$  が少なくとも一つターミナルと接続しているなら (つまり、 $t_{n_1,f} > 0$  および  $t_{n_2,f} > 0$ ) なら、フロンティア  $f$  と  $f'$  は接続する必要がある。(1)  $\{c_{n_1,f}\} = \{c_{n_2,f}\}$  および (2) ( $t_{n_1,f} = 0$  and  $t_{n_2,f} = 0$ ) or ( $t_{n_1,f} > 0$  and  $t_{n_2,f} > 0$ ) であるなら、新しく生成される  $n_1$  と  $n_2$  は同じ変数を保持する。そのため、 $n_1$  と  $n_2$  は同じシンクに遷移する。 □

## 4.2 削除

二部決定図のサイズはグラフサイズの増加に伴い指数関数的に増加する。そのため、グラフが大規模な場合、メモリ制約の問題上全てのノードを保持することは難しい。削除手順では、 $S^2BDD$  のサイズが  $w$  を超えないようにノードを削除する。削除手順における課題は、高い効率性と精度を保つために、どのノードを  $S^2BDD$  から削除するかである。後述するサンプリングの定理より、 $p_c$  と  $p_d$  が少ない枝の処理により大きく増加する場合、効果的にサンプル数を減少させることができる。ターミナルの接続性において下記の観測を用いる。

*Observation 1:*  $n$  に対応する中間グラフのターミナルは  $t_{n,f}$  が大きい  $f$  が存在するほど接続となりやすい。

*Observation 2:*  $n$  に対応する中間グラフのターミナルは  $d_{n,f}$  が小さく、 $t_{n,f} > 0$  となる  $f$  が存在するほど非接続となりやすい。

さらに、ノードの確率値  $p_n$  が大きい場合、 $n$  がシンクに遷移したときに  $p_c$  と  $p_d$  が大きく増加する。これらの観測に基づく、 $A^*$ フロンティア法を提案する。 $A^*$ フロンティア法は、ノードの優先度を決定し、優先度が高いノードを優先的に保持する。優先度の決定方法は、関数  $h$  で求める。

$$h(n) = p_n \cdot \max_{f \in \mathbb{F}} \left( \frac{t_{n,f}}{k}, \frac{1}{d_{n,f}} \right) \text{ if } t_{n,f} > 0. \quad (4)$$

ターミナルが少なくとも一つがつながっているフロンティアに対して、フロンティアが多くのターミナルに接続している、もしくはフロンティアが少ない曖昧枝をもっている場合に、この関数値は大きくなる。優先度が低いノードは削除され、削除されたノードの確率値はネットワーク信頼性に加算されない。そのため、削除したノードを近似解を求めるためにサンプリング手順で用いる。

## 4.2 サンプリング手順

提案アプローチは可能グラフをサンプリングしてネットワーク信頼性の近似解を計算する。本章では、 $S^2BDD$  構築中にどのように効率的にサンプリングを実施するか、およびどのように精度を保ちながらサンプル数を削減するかを説明する。効率的に可能グラフをサンプリングするために、層化サンプリングを拡張する。サンプリングにおけるネットワーク信頼性の近似では、モンテカルロサンプリングが主に用いられている。そのため、モンテカルロサンプリングにおいて、精度を保ちながらサンプル数を削減できることを証明する。

提案アプローチでは、 $S^2BDD$  の構築中にサンプリングを行うため、サンプリングの対象となる中間グラフの集合が未知である。そのため、ノードが削除される度に、サンプリングするかを判定しなければならない。単純に散布寝具を実施する場合、乱数を  $s \cdot |G_D|$  回生成する必要がある ( $s$  個の乱数を  $|G_D|$  の削除される中間グラフに対して実行する)。効率性を改善するために、層化サンプリングを用いる [24]。層化サンプリングは、サンプリングの対象となる全体の集合をいくつかのサブグループに分割、およびサブグループ内の要素に応じてサブグループに対するサンプル数を決定する。提案アプローチでは、削除される中間グラフの集合が全体集合となり、サブグループ内の中間グラフの確率値に応じてサンプル数を決定する。これにより、あるノードが削除された場合に、ノードに対応するサブグループのサンプル数の乱数の生成で済むため、効率的にサンプリングが実行可能である。

しかし、さきほど述べたように削除される中間グラフの全体集合が未知のため単純にサブグループに分割できない。本研究では、それぞれの層において削除される中間グラフをサブグループとする。ある層において一つ目のノードが削除される際に、その層のサンプル数を決める必要がある。そのためには、削除される中間グラフの確率値の総和を求める必要がある。層  $l$  において削除されるノードの確率値の総和  $p_s$  を下記の式で決定する。

$$p_s = 1 - \sum_{i=1}^{l-1} p_{s_i} - p_{N_{next}} - p_c - p_d, \quad (5)$$

この式において、 $p_{N_{next}}$  は  $n \in N_{next}$  の確率値を合計である。 $p_s$  は、 $l+1$  層のノード集合のサイズが最大値に達したときの層  $l$  における  $N$  内のノードの確率値の合計値を示す。各層において  $s \cdot p_s$  個の可能グラフをサンプリングする。  $s$  より  $s \cdot p_s$  は小さいため、層化サンプリングは効率的である。

**サンプル数の減少** 層化サンプリングにより、計算量を減少させることができる。さらに計算量を削減させるため、サンプル数を減少させる。しかし、サンプル数を減少させると、計算量の減少と同時に近似の精度も低下してしまう。そこで、近似の精度を損なわずにサンプル数を減少させることが重要である。さきほど述べたように、サンプリングの精度は分散で評価される。ネットワーク信頼性の分散は、下限値と上限値を用いることにより小さくなる [10, 22]。この性質を利用し、近似の精度を低下させずにサンプル数を減少させる。本稿では、モンテカルロサンプリングを用いてネットワーク信頼性を計算する場合について、サンプル数の削減について説明するが、本研究の成果は他のサンプリング技術についても適応可能である。

モンテカルロサンプリングのエスティメータは下記の式となる。

$$\hat{R} = \frac{\sum_{i=1}^s I(G_{p_i}, \mathbb{T})}{s}. \quad (6)$$

分散は下記の式で計算される [10].

$$\text{Var}[\hat{R}] = \frac{R(1-R)}{s}. \quad (7)$$

層化サンプリングは不偏サンプリングであるため (つまり  $E(\hat{R}) = R$ ), 分散は下記の式で計算できる [22].

$$\text{Var}[\hat{R}] = \frac{R(1-R)}{s} \approx \frac{\hat{R}(1-\hat{R})}{s}. \quad (8)$$

上限値と下限値を用いた場合, 分散は下記の式で計算できる [10, 22].

$$\text{Var}[\hat{R}]' = \frac{(\hat{R}-p_c)(1-p_d-\hat{R})}{s}. \quad (9)$$

下記の式を得ることでできる.

$$\frac{\hat{R}(1-\hat{R})}{s} \geq \frac{(\hat{R}-p_c)(1-p_d-\hat{R})}{s}. \quad (10)$$

式 (10) より, 下記の定理を得られる.

**Theorem 1** サンプル数  $s$ , 下限値  $p_c$ , および上限値  $1-p_d$  が与えられたとき, サンプル数  $s' (\leq s)$  が下記となる場合, サンプル数が  $s$  のモンテカルロサンプリングによるネットワーク信頼性の分散は, サンプル数  $s'$  の場合と分散が等しくなる.

$$s' = \begin{cases} s(1-p_d). & (p_c = 0) \\ s(1-p_c). & (p_d = 0) \\ s(1-4 \cdot p_c(1-p_c)). & (p_c = p_d) \\ s(1-4 \cdot p_c(1-p_d)). & (p_c < p_d) \\ s(1-\min(4p_c(1-p_c), \\ 4(p_c(1-p_d) + (p_d-p_c))). & (p_c > p_d) \end{cases}$$

*Proof:* 式 (10) より, 下記の不等式を得る:

$$\frac{(p_c-\hat{R})(1-p_d-\hat{R})}{s'} \leq \frac{\hat{R}(1-\hat{R})}{s}$$

$s'$  は下記で計算される:

$$\begin{aligned} s' &= s \cdot \frac{(\hat{R}-p_c)(1-p_d-\hat{R})}{\hat{R}(1-\hat{R})} \\ &= s \cdot \left(1 - \frac{p_c(1-\hat{R})+p_d(\hat{R}-p_c)}{\hat{R}(1-\hat{R})}\right) \end{aligned} \quad (11)$$

$p_c \leq \hat{R} \leq 1-p_d$  となるため,  $s' \leq s$  である.  $\hat{R}$  を式 (11) から取り除くために  $p_c$  と  $p_d$  のパターンを分ける.

- (1)  $p_c=0$ :  $s' = s \left(1 - \frac{p_d \hat{R}}{\hat{R}(1-\hat{R})}\right) \geq s(1-p_d)$ .
- (2)  $p_d=0$ :  $s' = s \left(1 - \frac{p_c(1-\hat{R})}{\hat{R}(1-\hat{R})}\right) \geq s(1-p_c)$ .
- (3)  $p_c=p_d$ :  $s' = s \left(1 - \frac{p_c(1-\hat{R})+p_c(\hat{R}-p_c)}{\hat{R}(1-\hat{R})}\right) \geq s(1-4p_c(1-p_c))$ .
- (4)  $p_c < p_d$ :  $s' = s \left(1 - \frac{p_c(1-\hat{R})+p_d(\hat{R}-p_c)}{\hat{R}(1-\hat{R})}\right) \geq s(1-4p_c(1-p_d))$ .
- (5)  $p_c > p_d$ : (i)  $s' \geq s(1-4p_c(1-p_c))$ . (ii)  $s' \geq s(1-4(p_c(1-p_c) + (p_d-p_c)))$ . Thus,  $s' = s(1-\min(4p_c(1-p_c), 4(p_c(1-p_c) + (p_d-p_c))))$ .  $\square$

提案アプローチはサンプル数を定理 1 に基づき減少させるため, サンプリングに基づくアプローチより効率的である.

### 4.3 曖昧グラフの前処理

ネットワーク信頼性を計算する際, グラフのサイズが小さくなると, 計算コストおよびメモリ使用量が小さくなる. そのため,  $S^2BDD$  構築の前に曖昧グラフを前処理し, ネットワーク信頼性を保ちながらサイズを小さくする. 前処理は, 2 枝接続コンポーネントを用いる [7].

**Definition 3 (2 枝接続コンポーネント)** グラフ  $G = (V, E)$  が与えられたとき, サブグラフ  $C = (V_C, E_C)$  は,  $E_C$  からいかなる枝の一つ取り除いてもグラフが分割されない場合, 2 枝接続コンポーネントである.  $E$  から枝を取り除いた時, グラフが分割される場合, その枝は橋と呼ばれる. 橋に接続する節点は, 関節点と呼ばれる. 2 枝接続コンポーネントの集合, 橋の集合, および関節点の集合をそれぞれ  $\mathbb{C}$ ,  $\mathbb{B}$ , および  $\mathbb{A}$  とする.

ある枝および節点を削除した際にグラフが分割されるか, されないかを 2 枝接続コンポーネント, 橋, および間接点から把握することができる. 2 枝接続コンポーネントはネットワークポロジからのみ計算できるので, インデックスとして事前に計算しておく.

前処理は, 枝刈り, 分解, 変換の 3 つのステップからなる. 枝刈りは不要な枝と節点を削除し, 分解は曖昧グラフを複数のグラフに分解, 変換はグラフの変換を行う

**枝刈り:** 枝刈りはネットワーク信頼性計算に影響しない枝と節点を削除する. ある枝および節点を削除した際にグラフが分割し, 分割されたグラフのどちらかがターミナルを含まない場合, その枝および節点は不要である.

**分解:** グラフを複数の小さいグラフに分割することにより, 計算量を削減することができる. 下記の補題に基づきグラフを分解する.

**Lemma 4** 曖昧グラフおよび橋が与えられたとき,  $R[G, \mathbb{T}] = p_b \cdot \prod_i^m R[G_i, \mathbb{T}_i]$  となる.  $p_b = \prod_{e_b \in \mathbb{B}} p(e_b)$  および  $\mathbb{T}_i$  は  $G_i$  におけるターミナル集合である.

*Proof:* 曖昧グラフ  $\mathcal{G}_{\mathbb{E}(\mathbb{E}_\exists, \mathbb{E}_-)}$  および枝  $e \in \mathbb{E} \setminus (\mathbb{E}_\exists \cup \mathbb{E}_-)$ , ネットワーク信頼性は Factoring Theorem [9] を用いて計算できる:

$$\begin{aligned} R[\mathcal{G}_{\mathbb{E}(\mathbb{E}_\exists, \mathbb{E}_-)}] &= p(e) \cdot R[\mathcal{G}_{\mathbb{E}(\mathbb{E}_\exists \cup e, \mathbb{E}_-)}] \\ &\quad + (1-p(e)) \cdot R[\mathcal{G}_{\mathbb{E}(\mathbb{E}_\exists, \mathbb{E}_- \cup e)}]. \end{aligned} \quad (12)$$

橋  $e_b = (v, v') \in \mathbb{B}$  を式内の  $e$  として選んだ場合,  $R[\mathcal{G}_{\mathbb{E}(\mathbb{E}_\exists, \mathbb{E}_- \cup e)}]$  はゼロとなる. 従って, 下記の式を得られる:

$$R[\mathcal{G}_{\mathbb{E}(\mathbb{E}_\exists, \mathbb{E}_-)}] = p(e_b) \cdot R[\mathcal{G}_{\mathbb{E}(\mathbb{E}_\exists \cup e_b, \mathbb{E}_-)}]. \quad (13)$$

全てのターミナルが接続するためには,  $e_b$  は存在する必要がある. そのため, 中間グラフ  $\mathcal{G}_{\mathbb{E}}$  を二つのグラフ  $\mathcal{G}_{\mathbb{E}_1}$  と  $\mathcal{G}_{\mathbb{E}_2}$  に分解可能である. 同様に, ターミナル集合  $\mathbb{T}$  を  $\mathbb{T}_1$  と  $\mathbb{T}_2$  に分割する. この際,  $\mathbb{T}_1$  は  $\{t \in \mathbb{T}, v, v' | t, v, v' \in \mathbb{V}_1\}$  となる (同様に,  $\mathbb{T}_2$  も計算可能である). 従って,  $R[\mathcal{G}_{\mathbb{E}}] = p(e_b) \cdot R[\mathcal{G}_{\mathbb{E}_1}] R[\mathcal{G}_{\mathbb{E}_2}]$  である.  $\mathcal{G}_{\mathbb{E}_1}$  と  $\mathcal{G}_{\mathbb{E}_2}$  は同じ方法により分解可能である. 最終的に  $R[G] = p_b \cdot \prod_i^m R[G_i]$  が得られる.  $\square$

**変換:** グラフを変換することにより, さらにグラフを小さくする. グラフの変換では, 下記のルールに基づき変換する.

- 連続枝 ( $e = (v, v'), e' = (v, v'')$ ):  $v$  がターミナルではなく次数が 2 の場合において,  $v, e$  および  $e'$  を削除し, 確率値  $p(e) \cdot p(e')$  がである枝を  $v'$  と  $v''$  間に追加する.
- 並列枝 ( $e = (v, v'), e' = (v, v')$ ):  $e$  および  $e'$  を削除し, 確率値が  $(1-(1-p(e)) \cdot (1-p(e')))$  である枝を  $v$  と  $v'$  に追加する.
- ループ: ループはネットワーク信頼性に影響がないため削除する. ループは連続枝と並列枝の変換において生成される可能性がある.

### 4.4 提案アプローチのアルゴリズム

本章では, 提案アプローチのアルゴリズムについて述べる. アルゴリズム 1 は, 提案アプローチの概要の疑似コードである. 提案アプローチはまずグラフの前処理を行う (line 3). そ

**Algorithm 1:** Computing the approximate network reliability

---

```

input : Uncertain graph  $\mathcal{G}$ , terminals  $\mathbb{T}$ , maximum BDD
         size  $w$ , size of samples  $s$ , 2-edge connected
         components  $\mathbb{C}$ , bridges  $\mathbb{B}$ , articulation points  $\mathbb{A}$ 
output: Approximate network reliability  $\hat{R}$ 
1 procedure our approach
2   set  $\mathbb{T}$  to  $\mathcal{G}$ ;
3    $\hat{R}, S_{\mathcal{G}} \leftarrow \text{Preprocess}(\mathcal{G}, \mathbb{C}, \mathbb{B}, \mathbb{A})$ ;
4   for  $\mathcal{G}_i \in S_{\mathcal{G}}$  do
5      $r \leftarrow \text{Construction}(\mathcal{G}_i, w, s)$ ;
6      $\hat{R} \leftarrow \hat{R} \cdot r$ ;
7   return  $\hat{R}$ ;
8 end procedure

```

---

**Algorithm 2:** Constructing  $S^2\text{BDD}$ 


---

```

input : Uncertain graph  $\mathcal{G}$ , maximum size  $w$ , number of
         samples  $s$ 
output: Approximate network reliability  $\hat{R}$ 
1 procedure Construction( $\mathcal{G}, w, s$ )
2   Ordering( $\mathbb{E}$ );
3    $p_c, p_d, p_s, c \leftarrow 0$ ; /* initialize probabilities and sampling
         count */
4    $s' \leftarrow s$ ;
5    $\mathbb{N} \leftarrow \text{CreateRoot}$ ;  $\mathbb{F} \leftarrow \text{null}$ ;
6   for  $l$  for  $1, \dots, |\mathbb{E}|$  do
7      $p_{\mathbb{N}}, p_{s_i} \leftarrow 0$ ;
8      $\mathbb{F}' \leftarrow \mathbb{F}$ ; compute  $\mathbb{F}$  based on  $e_l$ ;
9     while  $\mathbb{N}$  is empty do
10       $n \leftarrow \mathbb{N}.pop$ ;
11      for  $state \in \{\text{non-existent}, \text{existent}\}$  do
12        update( $n, \mathbb{F}', \mathbb{F}, state, \mathcal{G}, e_l$ );
13        if  $n$  is 0-sink then  $p_d \leftarrow p_d + p_n$ ;
14        else if  $n$  is 1-sink then  $p_c \leftarrow p_c + p_n$ ;
15        else
16          if  $\text{hashmap}[n]$  is not null then
17             $\text{Phashmap}[n] \leftarrow \text{Phashmap}[n] + p_n$ ;
18          else
19            if  $|\mathbb{N}_{next}| \leq w$  then
20               $h_n \leftarrow h(n)$ ;
21               $\mathbb{N}_{next}.add(n)$ ;  $\text{hashmap}[n] \leftarrow n$ ;
22               $p_{\mathbb{N}_{next}} \leftarrow p_{\mathbb{N}_{next}} + p_n$ ;
23            else
24               $p_{s_i} \leftarrow p_{s_i} + p_n$ ;
25              for  $i$  for  $1, \dots, \lfloor s' \cdot (1 - p_s -$ 
                 $\left. p_{\mathbb{N}_{next}} - p_c - p_d) \right\rfloor$  do
26                if Sampling( $\mathcal{G}, n$ ) then
27                   $c \leftarrow c + 1$ ;
28            end if
29          end if
30        end for
31      if  $c + \lfloor s' \cdot p_{\mathbb{N}_{next}} \rfloor \geq s'$  then
32        for  $n \in \mathbb{N}$  do
33          for  $i$  for  $1, \dots, \lfloor s' \cdot p_{\mathbb{N}_{next}} \rfloor$  do
34            if Sampling( $\mathcal{G}, n$ ) then  $c \leftarrow c + 1$ ;
35          end for
36        end for
37      break;
38    end while
39    if  $\mathbb{N}_n$  is empty then
40      break;
41     $\mathbb{N} \leftarrow \mathbb{N}_{next}$ ;
42    sort  $\mathbb{N}$  in descending order of  $h(n)$ ;
43     $p_s \leftarrow p_s + p_{s_i}$ ; compute  $s'$ ; clear  $\mathbb{N}_{next}$ ; clear
          $\text{hashmap}$ ;
44  end for
45  compute  $\hat{R}$  based on the sampling;
46  return  $\hat{R}$ ;
47 end procedure

```

---

それぞれの分解されたグラフに対して、 $S^2\text{BDD}$  を構築し、ネットワーク信頼性の近似解を得る (lines 4-5). 全ての近似解を掛け合わせたものが、全体のグラフのネットワーク信頼性となる (line 6).

あるゴリズム 2 は  $S^2\text{BDD}$  構築の疑似コードである。予め決められた枝の順序に従い、枝を選択しフロンティア集合を計算する (lines 6-8). **update** 関数 (line 12) はノードの変数を更新し、ターミナルが接続しているかどうかを計算する。更新されたノードが 0 シンクの場合、 $p_n$  を  $p_d$  に足し (line 13), 1 シンクの場合、 $p_n$  を  $p_c$  に足す (line 14). その他の場合は、ハッシュ値を計算し、 $n$  のハッシュが空ではない場合、 $n$  の確率値をハッシュ値に対応するノードの確率値に足す (lines 16-17). ハッシュが空の場合、 $n$  の優先度を更新した後、 $n$  を  $\mathbb{N}_{next}$  およびハッシュに挿入する (lines 19-21).  $\mathbb{N}_n$  の中のノード数が  $w$  を超えたとき、 $n$  を削除し  $n$  に対応する中間グラフから可能グラフをサンプリングする (lines 22-25).

**4. 実験****4.4.1 実験設定**

実験にて使用したデータセットを表 1 にまとめる。最初の 4 つのデータセット; Zachary-karate-club, American-revolution, Corporate-leadership, および Club-membership は, KONECT<sup>†</sup> からダウンロードした小規模なグラフデータである。小規模なグラフでは、枝の存在確率はガウス分布に基づいてランダムに与えた [8]. 他の 5 つのデータセット; DBLP before 2000, DBLP after 2000, Tokyo, New York City, および Hit-direct は、大規模なグラフデータである。枝の存在確率は、それぞれのデータの枝の属性値に基づいて計算した。DBLP before 2000 および DBLP after 2000 は DBLP<sup>‡</sup> からダウンロードし、節点と枝はそれぞれ著者と共著関係を表す。DBLP のデータは 2000 年を以前と以降に分けた。枝の存在確率は  $\frac{\log(\alpha+1)}{\log(\alpha_M+2)}$ ,  $\alpha$  および  $\alpha_M$  は共著数およびデータセット内の最大共著数をそれぞれ表す [6]. Tokyo と New York City は OpenStreetMap<sup>¶</sup> からダウンロードした道路ネットワークである。枝の存在確率は道路の長さを共著数に置き換えて DBLP データと同様に計算した。Tokyo と New York City どちらのデータセットも平面グラフではない。Hit-direct は Human Genome Center<sup>||</sup> からダウンロードしたタンパク質相互反応ネットワークである。それぞれのタンパク質の反応につけられたインタラクションスコア  $\in (0, 1]$  を枝の存在確率とした。

**4.4.2 実験結果**

提案アプローチ, サンプリングに基づくアプローチ, および二部決定図に基づくアプローチの効率性を比較する。図 3 は  $k$  が 5, 10, 20 の場合における応答時間を示す。DNF はメモリ制限により計算できなかった場合を表す。提案アプローチにおいては、 $w$  を 1,000, 10,000, および 100,000 を用いて、それぞれ Pro(1K), Pro(10K), および Pro(100K) と示すサンプル数は提案アプローチとサンプリングに基づくアプローチにおいて、10,000 を用いる。図 3 より、提案アプローチはサンプリングに基づくアプローチより高性能であることを示している。二部決定図に基づくアプローチは大規模なグラフではメモリ枯渇より計算ができない。提案アプローチは Tokyo と NYC データセットにおいての効率性が高い。これは、道路ネットワークは平面グラフに近い、 $S^2\text{BDD}$  の構築が効率的および前処理によるグラフサイズの削減が大きいためである。一方で、Hit-direct データセットでは、節点の平均次数が大きく上限値と下限値の幅が狭まらないため、効率性が低いが、それでもサンプリングに基づくアプローチより効率的である。

図 4 は、 $k$  を 5 とした場合の小規模グラフの応答時間を示す。この結果より、Am-Rv データセットを除いて、サンプリングに基づくアプローチが提案アプローチより効率的である。これは、提案アプローチが厳密解の計算を行おうとするためであ

<sup>†</sup><http://konect.uni-koblenz.de/>

<sup>‡</sup><http://dblp.uni-trier.de/>

<sup>¶</sup><https://www.openstreetmap.org>

<sup>||</sup><http://hintdb.hgc.jp/http/download.html>

表 1: データセットデータセット

名称	略称	タイプ	節点数	枝数	平均次数	平均確率
Zachary-karate-club	Karate	Social	34	78	4.59	0.527
Corporate-leadership	Co-Le	Affiliation	44	95	4.5	0.538
Club-membership	Club	Affiliation	40	99	4.75	0.532
American-Revolution	Am-Rv	Affiliation	141	160	2.27	0.528
DBLP before 2000	DBLP1	Coauthorship	25,871	32,297	8.38	0.222
DBLP after 2000	DBLP2	Coauthorship	48,938	136,034	5.56	0.203
Tokyo	Tokyo	Road network	26,370	32,298	2.45	0.391
New York City	NYC	Road network	180,188	208,441	2.31	0.294
Hit-direct	Hit-d	Protein	18,256	248,770	27.25	0.470

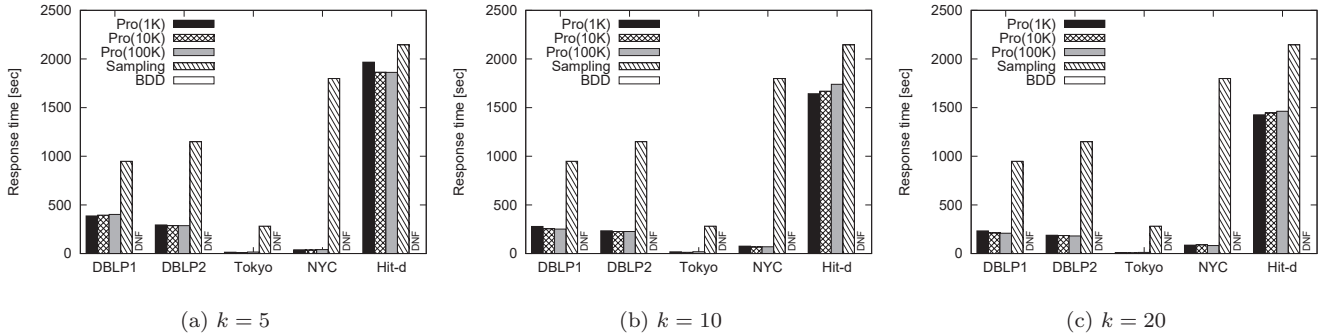


図 3: 大規模なグラフにおける効率性

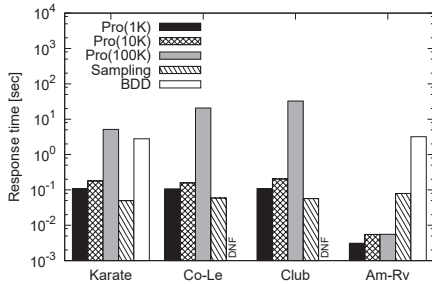


図 4: 小規模なグラフにおける効率性

表 2: Karate データセットにおける精度

k	Method	Variance	Error rate
5	Pro(10K)	0.025	0.036
	Sampling	0.025	0.037
10	Pro(10K)	0.013	0.058
	Sampling	0.013	0.058
20	Pro(10K)	$0.76 \cdot 10^{-3}$	0.054
	Sampling	$0.78 \cdot 10^{-3}$	0.056

表 3: Am-Rv データセットにおける精度

k	Method	Variance	Error rate
5	Pro(10K)	0	0
	Sampling	$0.43 \cdot 10^{-4}$	0.061
10	Pro(10K)	0	0
	Sampling	$0.099 \cdot 10^{-5}$	0.38
20	Pro(10K)	0	0
	Sampling	$0.10 \cdot 10^{-3}$	1.00

表 2 および 3 は Karate と Am-Rv データセットにおける精度を示す。表 2 より、提案アプローチは精度においてサンプリングに基づくアプローチを上回っていることがわかる。表 3 より、提案アプローチのエラー率が常にゼロであるため、常に厳密解を計算できていることがわかる。サンプリングに基づくアプローチは  $k$  が 20 の場合に、分散は小さいがエラー率が大きくなっている。これは、ネットワーク信頼性が非常に小さいため、サンプリングに基づくアプローチでは、ターミナルが接続している可能グラフをサンプリングすることができないためである。これにより、近似解は頻繁に 0 となり、エラー率は 1 となりやすい。これらの結果より、提案アプローチは少ないサンプル数で高い精度を達成していることがわかる。

### 5. むすび

本論文では、ネットワーク信頼性において高い効率性と精度を達成するアプローチを提案した。提案アプローチは二部決定図とサンプリングを効果的に統合させることにより、サンプル数の減少を行った。実験結果より、 $S^2BDD$  は既存の手法より応答時間および精度の面で優れていることを示した。

今後の課題としては、提案アプローチをより一般化させ、その他の問題に適応させることである。また、ネットワーク信頼性問題においては、性能を改善するために、枝の順序の最適化や並列分散処理への適応が考えられる。

### [謝辞]

本研究は科学研究費 (15K21069) の支援によって行われた。ここに記して謝意を表す。

る。二部決定図に基づくアプローチは Karate と Am-Rv データセットにおいて厳密解を計算している。提案アプローチも  $w$  が大きい場合に厳密解を同様に計算できている。特に、Am-Rv データセットにおいては、提案アプローチはサンプリングより小さい応答時間で厳密解を計算できている。

次に、提案アプローチとサンプリングに基づくアプローチの精度を検証する。実験において厳密解を計算できた Karate と Am-Rv データセットを用いる。分散とエラー率を近似解の精度の指標とし、それぞれ次の式で計算する。 $variance = \frac{\sum_{i=1}^{q_1} \sum_{j=1}^{q_2} (R_i - \hat{R}_{i,j})^2}{q_1 \cdot q_2}$  および  $error\ rate = \frac{\sum_{i=1}^{q_1} \sum_{j=1}^{q_2} |R_i - \hat{R}_{i,j}|}{q_1 \cdot q_2 \cdot R_i}$  とし、 $R_i$  と  $\hat{R}_{i,j}$  は  $i$  番目のネットワーク信頼性の厳密解と、 $i$  番目の計算における  $j$  番目の近似解をそれぞれ示す。100 回の計算を行い、それぞれに対して 100 回近似解を計算する。

## 参考文献

- [1] S. H. Ahmad. Simple enumeration of minimal cutsets of acyclic directed graph. *IEEE transactions on reliability*, 37(5):484–487, 1988.
- [2] S. B. Akers. Binary decision diagrams. *IEEE Transactions on computers*, (6):509–516, 1978.
- [3] S. Asthana, O. D. King, F. D. Gibbons, and F. P. Roth. Predicting protein complex membership using probabilistic network reliability. *Genome research*, 14(6):1170–1175, 2004.
- [4] M. O. Ball, C. J. Colbourn, and J. S. Provan. Network reliability. *Handbooks in operations research and management science*, 7:673–762, 1995.
- [5] D. Bergman, A. A. Cire, W.-J. van Hoeve, and J. Hooker. *Decision diagrams for optimization*. Springer, 2016.
- [6] M. Ceccarello, C. Fantozzi, A. Pietracaprina, G. Pucci, and F. Vandin. Clustering uncertain graphs. *PVLDB*, 11(4):472–484, 2017.
- [7] L. Chang, J. X. Yu, L. Qin, X. Lin, C. Liu, and W. Liang. Efficiently computing k-edge connected components via graph decomposition. In *SIGMOD*, pages 205–216, 2013.
- [8] Y. Cheng, Y. Yuan, L. Chen, G. Wang, C. Giraud-Carrier, and Y. Sun. DISTR: a distributed method for the reachability query over large uncertain graphs. *IEEE Transactions on Parallel and Distributed Systems*, 27(11):3172–3185, 2016.
- [9] C. J. Colbourn. *The combinatorics of network reliability*. Oxford University Press New York, 1987.
- [10] G. S. Fishman. A comparison of four monte carlo methods for estimating the probability of st connectedness. *IEEE Transactions on reliability*, 35(2):145–155, 1986.
- [11] R. Hamer, G. De Jong, E. Kroes, and P. Warffemius. The value of reliability in transport—provisional values for the netherlands based on expert opinion. *Transport Research Centre of the Dutch Ministry of Transport*, 2005.
- [12] G. Hardy, C. Lucet, and N. Limmios. K-terminal network reliability measures with binary decision diagrams. *IEEE Transactions on Reliability*, 56(3):506–515, 2007.
- [13] S. Hariri and C. S. Raghavendra. Syrel: A symbolic reliability algorithm based on path and cutset methods. *IEEE transactions on Computers*, 36(10):1224–1232, 1987.
- [14] J. U. Herrmann and S. Soh. A memory efficient algorithm for network reliability. In *Asia-Pacific Conference*, pages 703–707, 2009.
- [15] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein. A bayesian networks approach for predicting protein-protein interactions from genomic data. *science*, 302(5644):449–453, 2003.
- [16] R. Jin, L. Liu, and C. C. Aggarwal. Discovering highly reliable subgraphs in uncertain graphs. In *SIGKDD*, pages 992–1000, 2011.
- [17] R. Jin, L. Liu, B. Ding, and H. Wang. Distance-constraint reachability computation in uncertain graphs. *PVLDB*, 4(9):551–562, 2011.
- [18] J. Kawahara, T. Inoue, H. Iwashita, and S. Minato. Frontier-based search for enumerating all constrained subgraphs with compressed representation. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 100(9):1773–1784, 2017.
- [19] A. Khan, F. Bonchi, A. Gionis, and F. Gullo. Fast reliability search in uncertain graphs. In *EDBT*, pages 535–546, 2014.
- [20] M. O. Locks. A minimizing algorithm for sum of disjoint products. *IEEE Transactions on Reliability*, 36(4):445–453, 1987.
- [21] T. Maehara, H. Suzuki, and M. Ishihata. Exact computation of influence spread by binary decision diagrams. In *WWW*, pages 947–956, 2017.
- [22] E. Manzi, M. Labbé, G. Latouche, and F. Maffioli. Fishman’s sampling plan for computing network reliability. *IEEE Transactions on Reliability*, 50(1):41–46, 2001.
- [23] R. Motwani and P. Raghavan. *Randomized algorithms*. Chapman & Hall/CRC, 2010.
- [24] J. Neyman. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, 97(4):558–625, 1934.
- [25] W. G. Ortel. Broad band optical fiber telecommunications network, Jan. 19 1999. US Patent 5,861,966.
- [26] J. S. Provan. The complexity of reliability computations in planar and acyclic graphs. *SIAM Journal on Computing*, 15(3):694–702, 1986.
- [27] A. Satyanarayana and M. K. Chang. Network reliability and the factoring theorem. *Networks*, 13(1):107–120, 1983.
- [28] O. R. Theologou and J. G. Carlier. Factoring and reductions for networks with imperfect vertices. *IEEE Transactions on Reliability*, 40(2):210–217, 1991.
- [29] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.
- [30] R. K. Wood. A factoring algorithm using polygon-to-chain reductions for computing k-terminal network reliability. *Networks*, 15(2):173–190, 1985.
- [31] F.-M. Yeh, S.-K. Lu, and S.-Y. Kuo. OBDD-based evaluation of k-terminal network reliability. *IEEE Transactions on Reliability*, 51(4):443–451, 2002.