日本データベース学会和文論文誌
Vol. 18-J, Article No. 7, 2020 年 3 月

曖昧グラフにおける効率的なネットワーク信頼性の近似計算

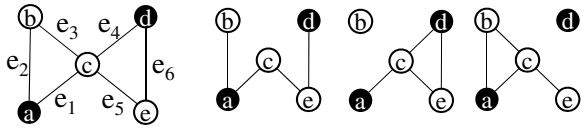
佐々木 勇和 [1]　藤原 靖宏 [2]　鬼塚 真 [3]

ネットワーク信頼性は曖昧グラフにおいて与えられた節点間の接続確率を評価する重要な指標のひとつである．ネットワーク信頼性の計算は #P 完全問題であるため，近似解を計算する手法が提案されている．本稿では，サンプリングに基づく新たな近似解計算手法を提案する．提案アプローチは層化サンプリングを拡張し，下限値と上限値を用いてサンプル数を削減する．これにより近似計算の効率化と近似解の精度の向上を可能にする．下限値と上限値を効率的に計算するために，二分決定図を拡張した **S^{2}BDD** を開発する．提案アプローチは S^{2}BDD を構築しながら，動的計画法を用いて効率的なサンプリングも実現する．実データを用いた実験により，提案アプローチが高精度を保ちながら，既存のサンプリングに基づくアプローチより最大で 51.2 倍高速であることを示す．

1　はじめに

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解析することが必要であり，実世界のオブジェクト間の関係性はグラフによりモデル化することができる．グラフの解析における基礎的なトピックとして，ネットワーク信頼性の計算がある [2, 15]．ネットワーク信頼性は枝に存在確率が付与されている曖昧グラフにおいて，与えられた複数の節点（ターミナルと呼ぶ）が相互接続する確率である．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反応が常に行われるわけではないため，曖昧グラフとしてモデル化することができる [1]．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 [2] や都市計画 [5] などの応用にてネットワーク信頼性は広く利用されている．

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 [15]．これは，可能グラフを列挙することが必要であることに起因する．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．ターミナルが全て接続となる可能グラフの確率を足し合わせることで，ネットワーク信頼性を計算できる．図 1 は曖昧グラフと 3 つの可能グラフの例を示し，黒の点はターミナルを表す．枝の存在確率を 0.7 とする場合，それぞれの可能グラフは 4 つの枝が存在（2 つが非存在）するため，可能グラフの存在確率は 0.0216 $(0.7^4 \cdot (1-0.7)^2)$ となる．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，これらの存在確率はネットワーク信頼性に加算される．



(a) 曖昧グラフ　　(b) 可能グラフ

図 1: 曖昧グラフ

本稿では，ネットワーク信頼性をサンプリング技術を用いて近似計算する．問題定義は以下となる．

問題定義 **(ネットワーク信頼性の近似計算)** : 曖昧グラフ $\mathcal{G}$，ターミナル集合 $\mathbb{T}$，およびサンプル数 s が与えられたとき，ネットワーク信頼性の近似値 $\hat{R}[\mathcal{G},\mathbb{T}]$ を効率的に計算する．

サンプリングの計算コストはサンプル数の増加に従い大きくなる．本研究では，近似精度を損なわずにサンプル数を削減させ，効率的な計算を行う．研究課題は，(1) どのように精度を理論的に保証しながらサンプル数の削減させるか，および (2) どのように理論的な結果を効率的に実現するかである．まず，最初の課題においては，層化サンプリング [14] を拡張する．層化サンプリングは予測値の下限値と上限値を用いることにより予測値の精度を向上させることが可能であり，近似精度を損なわずにサンプル数を削減可能なことを証明する．

精度を保ちつつサンプル数を削減できることを理論的な結果により保証することができるが，ネットワーク信頼性の効率的な計算に用いるには二つの課題ある．一つ目の課題は，効率的なネットワーク信頼性近似計算のために，ネットワーク信頼性の下限値と上限値を効率的に求めることが必要がある．二つ目は，精度の保証のために，下限値と上限値の計算に用いた可能グラフをサンプリング対象としてはいけないことである．これらを達成する効果的な方法は自明ではない．そこで，二分決定図を拡張した **scalable and sampling BDD (S^{2}BDD)** を開発する．S^{2}BDD はターミナルが接続または非接続となる可能グラフを優先的に探索することにより，下限値と上限値を効果的に求めることができる．また提案アプローチでは，S^{2}BDD を構築しながら，動的計画法を実施し，効率的に可能グラフをサンプリングする．このサンプリングは，下限値および上限値の計算に用いられた可能グラフをサンプリングすることを避けることができる．さらに，提案アプローチは，曖昧グラフのサイズが減少することにより，さらに効率化可能である．そこで 2 枝連結グラフを用いた拡張技術を提案する．拡張技術は，ネットワーク信頼性を損なうことなく，不必要な部分グラフの枝刈り，グラフの分割，およびより小さいグラフへの変換を行う．実データを用いた評価実験を行い，提案アプローチは既存のサンプリングに基づくアプローチより最大 51.2 倍高速かつ高精度であること示す．

本稿の構成は次の通りである．まず，2 にて事前知識について

[1] 正会員　大阪大学大学院情報科学研究科
sasaki@ist.osaka-u.ac.jp
[2] 正会員　NTT コミュニケーション科学基礎研究所
yasuhiro.fujiwara.kh@hco.ntt.co.jp
[3] 正会員　大阪大学大学院情報科学研究科
onizuka@ist.osaka-u.ac.jp

説明する．3 にて提案アプローチ，4 にて拡張手法を説明する．5 にて提案アプローチのアルゴリズムを述べる． 6 にて実験結果を示し， 7 にて本稿をまとめる．

2 事前知識

本章では，事前知識として，曖昧グラフ，ネットワーク信頼性，二分決定図，およびサンプリングについて説明する．

2.1 曖昧グラフ

連結かつ無向な曖昧グラフを $\mathcal{G} = (\mathbb{V}, \mathbb{E}, p)$ 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，曖昧枝 $e \in \mathbb{E}$ 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．曖昧枝 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．曖昧枝の存在確率は，他の枝の存在確率と独立に計算されると想定する．

可能グラフ $G_p = (\mathbb{V}, \mathbb{E}_p)$ は，曖昧グラフ $\mathcal{G}$ の全節点と枝のサブセットから成る．枝には存在確率は無く，$\mathbb{E} \backslash \mathbb{E}_p$ 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．

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

曖昧グラフ $\mathcal{G}$ 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．$\mathcal{G}$ の全ての可能グラフの集合を $\mathbb{W}^{\mathcal{G}}$ と定義する．

中間グラフ $\mathcal{G}_{\mathbb{E}}(\mathbb{E}_{\exists}, \mathbb{E}_{\neg})$ は，存在枝集合 $\mathbb{E}_{\exists}$, 非存在枝集合 $\mathbb{E}_{\neg}$, および曖昧枝の集合 $\mathbb{E} \backslash (\mathbb{E}_{\exists} \cup \mathbb{E}_{\neg})$ 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 $Pr[\mathcal{G}_{\mathbb{E}}(\mathbb{E}_{\exists}, \mathbb{E}_{\neg})]$ は以下の式で計算される．

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

簡単化のため，$Pr[\mathcal{G}_{\mathbb{E}}(\mathbb{E}_{\exists}, \mathbb{E}_{\neg})]$ の代わりに $Pr[\mathcal{G}_{\mathbb{E}}]$ を使用する．中間グラフ $\mathcal{G}_{\mathbb{E}}(\mathbb{E}_{\exists}, \mathbb{E}_{\neg})$ の可能グラフの数は，$2^{|\mathbb{E} \backslash (\mathbb{E}_{\exists} \cup \mathbb{E}_{\neg})|}$ である．$\mathcal{G}_{\mathbb{E}}$ の全ての可能グラフの集合を $\mathbb{W}^{\mathcal{G}_{\mathbb{E}}}$ 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と呼び，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らかではない．

2.2 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（ネットワーク信頼性） 任意の k 個のターミナル集合 $\mathbb{T}$ と曖昧グラフ $\mathcal{G}$ が与えられたとき, ネットワーク信頼性 $R[\mathcal{G}, \mathbb{T}]$ は下記の式で計算される．

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

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．

$\hat{R}[\mathcal{G}, \mathbb{T}]$ をネットワーク信頼性の近似解と定義する．簡単化のため，R と $\hat{R}$ を $R[\mathcal{G}, \mathbb{T}]$ と $\hat{R}[\mathcal{G}, \mathbb{T}]$ 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 [6]．k が 2 の場合でも，ネットワーク信頼性は #P 完全問題である [15].

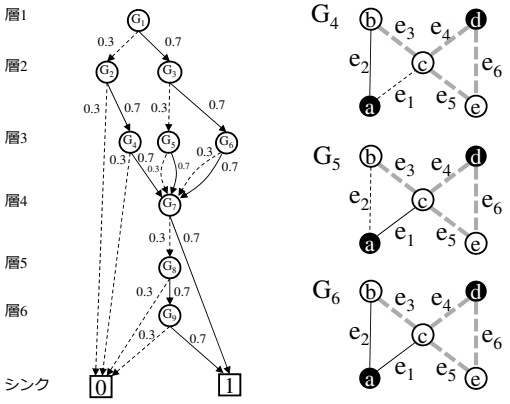
二分決定図 [6] とサンプリング [7] がネットワーク信頼性計算のための主要な技術である．二分決定図に基づくアプローチは小規模なグラフに対して効率的に厳密解を計算可能で，サンプリングに基づくアプローチは大規模なグラフに対して近似解を計算可能である．

2.3 二分決定図

二分決定図（BDD）は有向非循環グラフ $\mathcal{D} = (\mathbb{N}, \mathbb{A})$ で表現される．二分決定図のノード集合を $\mathbb{N}$，二分決定図のアーク集合を $\mathbb{A}$ と定義する[*1]．図 2(a) は図 1 のグラフにおける二分決定図を示す. 各ノードは中間グラフに対応しており，アークは存在枝または非存在枝と対応している．二分決定図は，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，二つの出次 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表しており，それぞれのアークの重みは枝の存在確率または非存在確率を表している．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し，e_l から $e_{|\mathbb{E}|}$ は曖昧枝である．二分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-シンクまでのアークを辿ることにより，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-シンクにアーク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の二分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 [8]．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．二分決定図が既に層 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を $\mathbb{F}_l$ とする．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）を保持しており，フロンティアの変数が同じになった場

[*1] 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，二分決定図の節点と枝を “ノード” と “アーク” とそれぞれ呼ぶ.



(a) 二分決定図　(b) 層3の中間グラフ

図 2: 図 1(a) のグラフにおける二分決定図.

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．

二分決定図のサイズはノードの数で定義され，メモリ使用量は二分決定図のサイズに依存する [6]．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．

2.4 サンプリング

サンプリングは近似解を計算するために用いられる一般的な方法である．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，(1) 曖昧グラフから可能グラフを取り出す，(2) 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，という処理を s 回繰り返す．可能グラフを取り出しと接続確認のために，それぞれ時間計算量 $O(|\mathbb{E}|)$ および $O(|\mathbb{V}|+|\mathbb{E}|)$ が必要であるため，サンプリングに基づく手法の計算量は $O(s\cdot(|\mathbb{V}|+|\mathbb{E}|))$ となる．

サンプリングに基づく手法は乱択アルゴリズム [12] であるため，その精度は分散により評価される．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編サンプリング (unbiased sampling) が必要である．また，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ため，精度と時間計算量にはトレードオフが存在する．

サンプリングに基づく s-t 到達可能性問合せ [9, 10] に関する研究は活発に行われている．しかし，これは 2 点のターミナル間の接続性にのみ着目しており，k 点のターミナルには単純には拡張できない．筆者らの知る限り，ネットワーク信頼性を効率的に近似計算可能な乱択アルゴリズムは提案されていない．

3 提案アプローチ

本章では，提案アプローチを説明する．3.1 にて，サンプル数の削減方法について述べ，3.2 にて S^{2}BDD を説明する．

3.1 サンプル数の削減

本節では，層化サンプリング [4, 11] に基づいて高精度を維持しつつサンプル数を削減できることを証明する．層化サンプリングはネットワーク信頼性の近似計算の分散を小さくするこができるため，近似精度を損なわずにサンプル数を削減することができる．

層化サンプリングを適用するために，可能グラフの集合 $\mathbb{W}^{\mathcal{G}}$ を 3 つのサブ集合グループ $\mathbb{W}_c^{\mathcal{G}}, \mathbb{W}_d^{\mathcal{G}}$, および $\mathbb{W}_u^{\mathcal{G}}$ に分ける．$\mathbb{W}_c^{\mathcal{G}}$ および $\mathbb{W}_d^{\mathcal{G}}$ は，それぞれターミナルが接続および非接続となる可能グラフのみを含んでおり，$\mathbb{W}_u^{\mathcal{G}}$ は二つの集合に含まれていない可能グラフが含まれている．$\mathbb{W}_c^{\mathcal{G}}$ と $\mathbb{W}_d^{\mathcal{G}}$ に含まれる可能グラフの確率の総和をそれぞれ p_c と p_d とする．ここで，定義 1 より，上限値と下限値は下記の式により計算できる．

$$
\begin{aligned}
R &= \textstyle\sum_{G_p\in\mathbb{W}_c^{\mathcal{G}}} Pr[G_p] + \sum_{G_p\in\mathbb{W}_u^{\mathcal{G}}} I(G_p,\mathbb{T})Pr[G_p] \\
&= p_c + \textstyle\sum_{G_p\in\mathbb{W}_u^{\mathcal{G}}} I(G_p,\mathbb{T})Pr[G_p] \\
&\geq p_c.
\end{aligned}
$$

$$
\begin{aligned}
R &= 1 - \textstyle\sum_{G_p\in\mathbb{W}_d^{\mathcal{G}}} Pr[G_p] - \sum_{G_p\in\mathbb{W}_u^{\mathcal{G}}} (Pr[G_p] - I(G_p,\mathbb{T})Pr[G_p]) \\
&= 1 - p_d - \textstyle\sum_{G_p\in\mathbb{W}_u^{\mathcal{G}}} (Pr[G_p] - I(G_p,\mathbb{T})Pr[G_p]) \\
&\leq 1 - p_d.
\end{aligned}
$$

これらの式より，$p_c \leq R \leq 1-p_d$ を得ることができる．

分散は予測手法にも依存する．提案アプローチでは，代表的な二つの予測手法であるモンテカルロ法と Horvitz-Thompson 法を用いる．それぞれの手法について，下限値と上限値を用いてサンプル数を削減できることを下記で述べる．

モンテカルロ法: モンテカルロ法による $\hat{R}$ の計算式は下記となる．

$$\hat{R} = \frac{\sum_{i=1}^{s} I(G_{p_i},\mathbb{T})}{s}. \tag{4}$$

分散は下記の式で計算される [4].

$$Var[\hat{R}] = \frac{R(1-R)}{s}. \tag{5}$$

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 [11].

$$Var[\hat{R}] = \frac{R(1-R)}{s} \approx \frac{\hat{R}(1-\hat{R})}{s}. \tag{6}$$

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 [4, 11].

$$Var[\hat{R}]' = \frac{(\hat{R}-p_c)(1-p_d-\hat{R})}{s}. \tag{7}$$

式 6 および式 7 より，下記の式を得ることできる．

$$\frac{\hat{R}(1-\hat{R})}{s} \geq \frac{(\hat{R}-p_c)(1-p_d-\hat{R})}{s}. \tag{8}$$

式 (8) 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' $(\leq s)$ が下記の式となる場合，サンプル数が s' の上限値と下限値を用いたモンテカルロ法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い，

もしくは小さくなる．

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

Proof: 式 (8) 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:

$$\begin{aligned} s' &= s \cdot \frac{(\hat{R}-p_c)(1-p_d-\hat{R})}{\hat{R}(1-\hat{R})} \\ &= s \cdot \left(1 - \frac{p_c(1-\hat{R})+p_d(\hat{R}-p_c)}{\hat{R}(1-\hat{R})}\right) \end{aligned} \quad (9)$$

$p_c \leq \hat{R} \leq 1-p_d$ となるため, $s' \leq s$ である．$\hat{R}$ を式 (9) 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) $s' \geq s(1-4(p_c(1-p_c)+(p_d-p_c))$. よって，$s' = s(1-\min(4p_c(1-p_c), 4(p_c(1-p_d)+(p_d-p_c))))$.
従って，定理 1 が成り立つ． □

Horvitz-Thompson 法: Horvitz-Thompson 法による $\hat{R}$ の計算式は下記となる．

$$\hat{R} = \frac{\sum_{i=1}^{s} Pr[G_{p_i}] \cdot I(G_{p_i}, \mathbb{T})}{\pi_i},$$

$\pi_i = 1-(1-Pr[G_{p_i}])^s$ である．分散は下記で計算される．

$$\begin{aligned} Var[\hat{R}] &= \sum_{i=1}^{s} \left(\frac{1-\pi_i}{\pi_i}\right) I(G_{p_i}, \mathbb{T}) Pr[G_{p_i}]^2 \\ &+ \sum_i^s \sum_{j,i\neq j}^{s} \left(\frac{\pi_{ij}-\pi_i\pi_j}{\pi_i\pi_j}\right) I(G_{p_i}, \mathbb{T}) I(G_{p_j}, \mathbb{T}) Pr[G_{p_i}] Pr[G_{p_j}], \end{aligned}$$

$\pi_{ij} = 1-(1-Pr[G_{p_i}])^s - (1-Pr[G_{p_j}])^2 + (1-Pr[G_{p_i}]-Pr[G_{p_j}])^s$ である．分散は下記の式で計算できる [7].

$$Var[\hat{R}] = \frac{R(1-R)}{s} - \frac{\sum_{i=1}^{s}(s-1) I(G_{p_i}, \mathbb{T}) Pr[G_{p_i}]^2}{2s}. \quad (10)$$

下限値と上限値を用いた分散は下記により計算できる．

$$Var[\hat{R}]' = \frac{(\hat{R}-p_c)(1-p_d-\hat{R})}{s} - \frac{\sum_{i=1}^{s}(s-1) I(G_{p_i}, \mathbb{T}) Pr[G_{p_i}]^2}{2s}. \quad (11)$$

Theorem 2 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，サンプル数が s' の上限値と下限値を用いた Horvitz-Thompson 法によるネットワーク信頼性の分散は，サンプル数 s（1 と同式）の場合と分散が等しく，もしくは小さくなる．

Proof: 式 (10)，(11) より，以下の式を得る．

$$\begin{aligned} &\frac{(\hat{R}-p_c)(1-p_d-\hat{R})}{s'} - \frac{\sum_{i=1}^{s}(s'-1) I(G_{p_i}, \mathbb{T}) Pr[G_{p_i}]^2}{2s'} \\ &= \frac{\hat{R}(1-\hat{R})}{s} - \frac{\sum_{i=1}^{s}(s-1) I(G_{p_i}, \mathbb{T}) Pr[G_{p_i}]^2}{2s}. \end{aligned}$$

Horvitz-Thompson 法は不偏サンプリングであるため，右項は同じ値となる．証明は定理 1 と同様である． □

提案アプローチは定理 1 と 2 に従い，サンプル数を削減することができるため，既存のサンプリングに基づく手法より効率的に近似値を計算可能である．

3.2 Scalable and Sampling BDD: S^{2}BDD

ネットワーク信頼性の下限値と上限値を用いることにより，サンプル数を削減することができる．効率的な下限値と上限値の計算方法として，S^{2}BDD を開発する．S^{2}BDD を用いることにより，ターミナルが接続もしくは非接続となる可能グラフのうち存在確率が高いものを効率的に探索することが出来る．さらに，S^{2}BDD を構築しながら，上限値と下限値の計算に用いていない可能グラフのサンプリング行う．S^{2}BDD はメモリ使用量を抑えるため，一層のみから構成される．これは，l 層の BDD を構築した後は，$l+1$ 層の構築および下限値と上限値の計算に $l-1$ 層が必要ないという観測に基づく．

S^{2}BDD を下記に定義し，構築方法について述べる．

Definition 2 S^{2}BDD は層 l のノード集合 $\mathbb{N}$ および 1-シンクと 0-シンクから構成される．S^{2}BDD は下記に示す変数を各ノード $n \in \mathbb{N}$ が保持する:

- 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．

提案アプローチは，S^{2}BDD を用いて下限値と上限値の計算およびサンプリングの両方を行う．通常の BDD と S^{2}BDD の違いについて述べると, BDD は全ての層を保持するのに対し, S^{2}BDD は一層とシンクのみ保持する．例えば，図 2 では，S^{2}BDD は三層目とシンクを保持するが，一層目と二層目は保持しない．

S^{2}BDD の構築のためには，枝 e_l を処理し，$l+1$ におけるノード集合 $\mathbb{N}_{next}$ を計算する．構築法は，生成，結合，削除，サンプリングの 4 つの手順からなる．下記にそれぞれの手順を説明する．

3.2.1 生成手順と結合手順

二分決定図に基づくアプローチは，二分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^{2}BDD 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 $\mathbb{N}_{next}$ 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．

S^{2}BDD 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^{2}BDD 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．

次に，結合手順を説明する．S^{2}BDD 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 1　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, $e_{l+1}, \ldots, e_{|\mathbb{E}|}$ の状態が同じである n_1 と n_2 の子孫ノードは同じシンクに遷移する．

Proof: 割愛する．[13] を参照されたい．　□

3.2.2　削除手順

二分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ため，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く，S ^{2}BDD 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^{2}BDD から削除するかである．定理 1 と 2 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 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く，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．これらの考えに基づいて，ノードの優先度を決定するヒューリスティック関数 h を定義する．

$$h(n) = p_n \cdot \max_{f \in \mathbb{F}} \left(\frac{t_{n,f}}{k}, \frac{1}{d_{n,f}} \right) \text{ if } t_{n,f} > 0. \qquad (12)$$

ターミナルが少なくとも一つがつながって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．

3.2.3　サンプリング手順

提案アプローチは，層化サンプリングを用いた理論を達成するために，下限値と上限値の計算に用いた可能グラフのサンプルとせずに，可能グラフをサンプリングする必要がある．そのため，まだターミナルが接続または非接続となっていない可能グラフをサンプリングする．このような可能グラフのセットを $\mathbb{W}_u^{\mathcal{G}}$ とする．$\mathbb{W}_u^{\mathcal{G}}$ は，削除されたノードと S^{2}BDD に存在するノードに対応する中間グラフから得ることができる．効率的なサンプリングのために，$\mathbb{W}_u^{\mathcal{G}}$ から動的計画法を用いてサンプリングを実施する．加えて，層化ランダムサンプリングの考えを用いて，サブグループ毎のサンプル数を決定する．

まず，$\mathbb{W}_u^{\mathcal{G}}$ をサブグループにわけ，その後それぞれのサブグループから可能グラフをサンプリングする．サブグループ毎のサンプル数は，サブグループ内の中間グラフの確率の総和に比例して決定する．ここでは，削除されたノードと S^{2}BDD 内のノードに分けて説明する．削除されたノードにおけるサブグループは S^{2}BDD の同じ層で削除されたものをサブグループとする．これは削除された節点の確率がサンプル数を決定するには小さすぎるためである．層 l におけるサンプル数 s_l は，s と層 l で削除されたノードの確率の総和 $\hat{p_{s_l}}$ の掛け合わせで求めることができる．S^{2}BDD に存在するノードにおいては，それぞれのノードに対応する中間グラフそのものがサブグループに対応し，ノードがもつ確率に従ってサンプル数を決定できる．

3.3　計算量

提案アプローチの時間と空間計算量について説明する．

Theorem 3　曖昧グラフ $\mathcal{G}$，更新後のサンプル数 s'，および S^{2}BDD の最大幅 w が与えられたとき，提案アプローチの時間と空間計算量はそれぞれ $O(w^2 \log w + s'(|\mathbb{V}| + |\mathbb{E}|))$ と $O(w \log w + |\mathbb{V}| + |\mathbb{E}|)$ なる．

Proof:　まず時間計算量について述べる．提案アプローチは，S^{2}BDD の構築とサンプリングの二つに分けることができる．S^{2}BDD の構築では，生成および結合の処理にて，それぞれのノードの属性を比較する．ノードの属性数はフロンティアの数に比例し，フロンティア数は $O(\log w)$ となる．これは，存在または非存在となる枝数が最大で $\log w$ となるためである．そのため，S^{2}BDD 構築の時間計算量は $O(w^2 \log w)$ となる．サンプリングの時間計算量は $O(s'(|\mathbb{V}| + |\mathbb{E}|))$ である．従って，時間計算量は $O(w^2 \log w + s'(|\mathbb{V}| + |\mathbb{E}|))$ となる．

空間計算量について述べる．空間計算量は S^{2}BDD のサイズと曖昧グラフのサイズに依存する．S^{2}BDD のサイズは，ノード数と属性数を掛けたものとなる．従って，空間計算量は $O(w \log w + |\mathbb{V}| + |\mathbb{E}|)$ となる．　□

4　拡張手法

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^{2}BDD 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．拡張手法は 2 枝接続コンポーネントを用いる [3].

Definition 3（2 枝接続コンポーネント）　グラフ $G = (\mathbb{V}, \mathbb{E})$ が与えられたとき, サブグラフ $C = (\mathbb{V}_C, \mathbb{E}_C)$ は，$\mathbb{E}_C$ からいかなる枝を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．$\mathbb{E}$ 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

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 𝕋 to 𝒢;
3 R̂, 𝒮_𝒢 ← Preprocess(𝒢, 𝕋, ℂ, 𝔹, 𝔸);
4 for 𝒢_i ∈ 𝒮_𝒢 do
5     r ← Construction(𝒢_i, w, s);
6     R̂ ← R̂ · r;
7 return R̂;
8 end procedure
```

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トポロジからのみ計算できるため事前に計算する．

拡張手法は，枝刈り，分解，変換の 3 つのステップからなる．枝刈りは不必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．枝刈りでは，$R[\mathcal{G}] = R[\mathcal{G}']$ となる $\mathcal{G}'$ を求める．$\mathcal{G}'$ の枝数は，ネットワーク信頼性に影響しない枝を削除するため，$\mathcal{G}$ より小さい．分解では，$R[\mathcal{G}'] = \Pi_{i=1}^{m} R[\mathcal{G}_i]$ となるサブグラフ $\mathcal{G}_1, \ldots, \mathcal{G}_m$ を求める．変換では $R[\mathcal{G}_i] = R[\mathcal{G}_i']$ となる $\mathcal{G}_i'$ を求める．より小さいグラフになるよう変換するため，$\mathcal{G}_i'$ の枝数は $\mathcal{G}_i$ より小さい．より具体的な処理においては [13] を参照されたい．

拡張技術はネットワーク信頼性の計算にかかる計算コストを効果的に小さくすることができる．さらに，サンプリングの精度も向上することができる．

Theorem 4　$R[\mathcal{G}] = \quad p_b \Pi_{i=1}^{m} R[\mathcal{G}_i]$ となる $\mathcal{G}_1, \ldots, \mathcal{G}_m$ が与えられたとき，$0 < \hat{R} < 1$ and $0 < p_b < 1$ の場合において，ネットワーク信頼性の分散は小さくなる．

Proof:　ネットワーク信頼性は $\hat{R} = p_b \cdot \Pi_i \hat{R}[\mathcal{G}_i]$ となり，分散は下記の式により計算できる．

$$
\begin{aligned}
Var[\hat{R}] &= Var[p_b \cdot \Pi_i \hat{R}[\mathcal{G}_i]] \\
&= (Var[p_b] + {p_b}^2)(Var[\hat{R}[\mathcal{G}_1]] + \hat{R}[\mathcal{G}_1]^2) \cdots \\
&\quad (Var[\hat{R}[\mathcal{G}_m]] + \hat{R}[\mathcal{G}_m]^2) - {p_b}^2 \cdot \Pi_{i=1}^{m} \hat{R}[\mathcal{G}_i]^2 \\
&= {p_b}^2 \Pi_i (Var[\hat{R}[\mathcal{G}_i]] + \hat{R}[\mathcal{G}_i]^2) - {p_b}^2 \Pi_i \hat{R}[\mathcal{G}_i]^2 \\
&= {p_b}^2 \Pi_i \left(\frac{\hat{R}[\mathcal{G}_i](1-\hat{R}[\mathcal{G}_i])}{s} + \hat{R}[\mathcal{G}_i]^2\right) - {p_b}^2 \Pi_i \hat{R}[\mathcal{G}_i]^2 \\
&= {p_b}^2 \Pi_i \hat{R}[\mathcal{G}_i] \left(\frac{(1+(s-1)\hat{R}[\mathcal{G}_i])}{s}\right) - {p_b}^2 \Pi_i \hat{R}[\mathcal{G}_i]^2 \\
&< \frac{{p_b}^2 \Pi_i \hat{R}[\mathcal{G}_i]}{s} - \frac{{p_b}^2 \Pi_i \hat{R}[\mathcal{G}_i]^2}{s} \\
&= p_b \frac{\hat{R}[\mathcal{G}_i](1-\hat{R}[\mathcal{G}_i])}{s} < \frac{\hat{R}(1-\hat{R})}{s}
\end{aligned}
\tag{13}
$$

ここで，$Var[p_b] = 0$ である．$Var[\hat{R}]$ は分解せずにネットワーク信頼性を計算した場合に比べて小さい．　□

Algorithm 2: Constructing S^{2}BDD

input : Uncertain graph $\mathcal{G}$, maximum size w, number of samples s
output Approximate network reliability $\hat{R}$
:

```
1  procedure Construction(𝒢, w, s)
2  Ordering(𝔼);
3  p_c, p_d, p̂_{s_l}, c ← 0;  /* initialize probabilities and sampling
     count */
4  s' ← s;
5  ℕ ← CreateRoot; 𝔽 ← null;
6  for l for 1, ··· , |𝔼| do
7      p_ℕ, p_{s_i} ← 0;
8      𝔽' ← 𝔽; compute 𝔽 based on e_l;
9      while ℕ is empty do
10         n ← ℕ.pop;
11         for state ∈ { non-existent, existent } do
12             set(n, 𝔽', 𝔽, state, 𝒢, e_l);
13             if n is 0-sink then p_d ← p_d + p_n;
14             else if n is 1-sink then p_c ← p_c + p_n;
15             else
16                 if hashmap[n] is not null then
17                     p_{hashmap[n]} ← p_{hashmap[n]} + p_n;
18                 else
19                     if |ℕ_next| ≤ w then
20                         h_n ← h(n);
21                         ℕ_next.add(n); hashmap[n] ← n;
                           p_{ℕ_next} ← p_{ℕ_next} + p_n;
22                     else
23                         p_{s_i} ← p_{s_i} + p_n;
24                         for i for
                             1, ··· , ⌊s' · (1 − p̂_{s_l} − p_{ℕ_next} − p_c − p_d)⌋ do
25                             if Sampling(𝒢, n) then c ← c + 1;
26         if c + ⌊s' · p_{ℕ_next}⌋ ≥ s' then
27             for n ∈ ℕ do
28                 for i for 1, ··· , ⌊s' · p_{ℕ_next}⌋ do
29                     if Sampling(𝒢, n) then c ← c + 1;
30             break;
31         if ℕ_n is empty then
32             break;
33         ℕ ← ℕ_next;
34         sort ℕ in descending order of h(n);
35         p̂_{s_l} ← p̂_{s_l} + p_{s_i}; compute s'; clear ℕ_next; clear hashmap;
36 compute R̂ based on the sampling;
37 return R̂;
38 end procedure
```

5　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拡張手法を用いてグラフを分割する (line 3)．拡張手法のアルゴリズムにおいては [13] を参照されたい．それぞれの分解されたグラフに対して，S^{2}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)．

表 1: データセット

名称	略称	タイプ	節点数	枝数
Zachary-karate-club	Karate	Social	34	78
American-Revolution	Am-Rv	Affiliation	141	160
DBLP before 2000	DBLP1	Coauthorship	25,871	108,459
DBLP after 2000	DBLP2	Coauthorship	48,938	136,034
Tokyo	Tokyo	Road network	26,370	32,298
New York City	NYC	Road network	180,188	208,441
Hit-direct	Hit-d	Protein	18,256	248,770

アルゴリズム 2 は S^{2}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).

6 評価実験

提案アプローチを計算効率，精度，およびメモリ使用量の観点から評価を行う．

6.1 データセット

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 2 つのデータセット; Zachary-karate-club と American-revolution は，KONECT*2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，枝の存在確率は一様分布に基づいてランダムに与えた．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．

6.2 設定と実装

それぞれのデータセットに対して，ターミナルをランダムに選び 20 回の検索を行う．ターミナル数 k，および S^{2}BDD の最大幅を w 変化させる. 比較手法として，サンプリングに基づく手法と BDD に基づく手法を用いる．BDD に基づく手法は最新ライブラリである TdZDD*3を用いる．全てのアルゴリズムは C++ で実装され，Intel Xenon E7-8860v4，256GB RAM が搭載された計算機にて実験を行う．

6.3 効率性

提案アプローチ，拡張手法を用いない提案アプローチ，サンプリングに基づくアプローチ，および二分決定図に基づくアプローチの効率性を比較する．提案アプローチとサンプリングに基づくアプローチは，モンテカルロ法を用い，s を 10,000 とする．提案アプローチにおいては，w を 10,000 とする．提案アプローチ，拡張手法を用いない提案アプローチ，サンプリングに基づくアプローチ，および二分決定図に基づくアプローチの凡例をそれぞれ Pro(MC)，Pro(MC)w/o ext， Sampling(MC)，および BDD とする．Horvits-Thompson 法は同傾向の結果であるため割愛する．

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．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．二分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に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ため，S^{2}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．

6.4 最大幅の影響

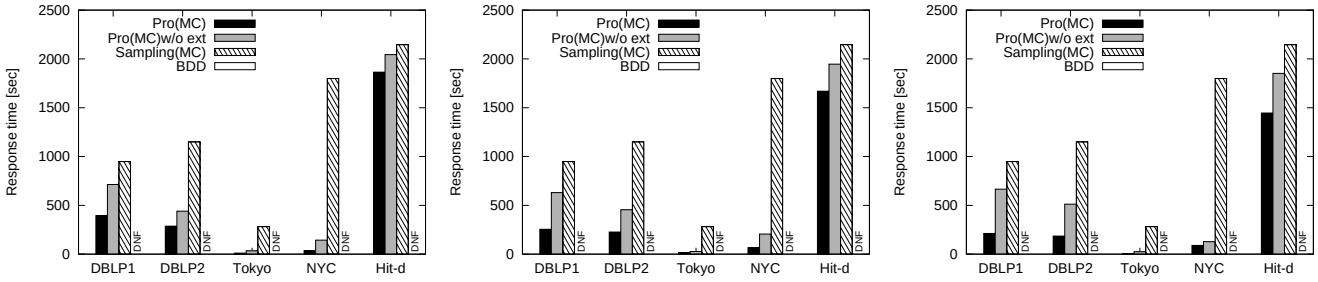
S^{2}BDD の最大幅を変化させた場合のメモリ使用量と応答時間を評価する．図 4 は，(a) メモリ使用量と (b) 応答時間を示す．図 4(a) より，最大幅を大きくするとメモリ使用量が大きくなることがわかる．一方で，グラフサイズには影響されていないことがわかる．提案アプローチはメモリ使用量の観点では，大規模なグラフに対応することができることを示している．図 4(b) より，最大幅は応答時間にそれほど影響していないことがわかる．最大幅が大きいとき，サンプル数はより削減可能だが，S^{2}BDD の構築に時間がかかる．提案アプローチの応答時間は最大幅にロバストであることがわかる．これらの結果は，提案アプローチは大規模なグラフに対しても効果的に応答時間を削減できることを示している．

6.5 精度

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．サンプリング手法として，モンテカルロ法と Horvitz-Thompson 法（Pro(HT) と Sampling(HT)）の両方を用いる．大規模なグラフでは厳密解を計算できないため，厳密解が計算可能な小規模なグラフである Karate と Am-Rv データセットを評価に用いる．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．$variance = \frac{\Sigma_{i=1}^{q_1}\Sigma_{j=1}^{q_2}(R_i - \hat{R}_{i,j})^2}{q_1 \cdot q_2}$ および $error\ rate = \frac{\Sigma_{i=1}^{q_1}\Sigma_{j=1}^{q_2}|R_i - \hat{R}_{i,j}|}{q_1 \cdot q_2 \cdot R_i}$ とし，R_i と $\hat{R}_{i,j}$ は i 番目の検索のネットワーク信頼性の厳密解と，i 番の検索における j 番目の近似解をそれぞれ示す．検索回数を 100 回とし，それぞれの検索に対して 100 回近似解を計算する．

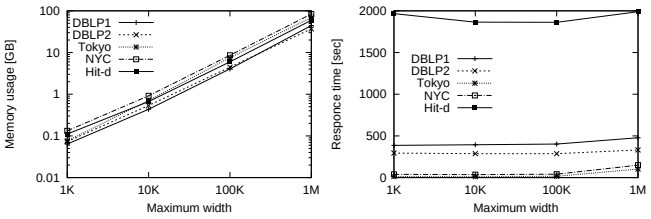
表 2 は Karate と Am-Rv データセットにおける精度を示す．表 2 より，提案アプローチは精度においてサンプリングに基づくアプローチを上回っていることがわかる．また，Am-Rv データセットでは，提案アプローチのエラー率が常にゼロであるため，常に厳密解を計算できていることがわかる．サンプリングに基づくアプローチは k が 20 の場合に，分散は小さいがエラー率が大きくなっている．これは，ネットワーク信頼性が非常に小さいため，サンプリングに基づくアプローチでは，ターミナルが接続している可能グラフをサンプリングすることができないためであ

*2 http://konect.uni-koblenz.de/
*3 https://github.com/kunisura/TdZdd



(a) $k = 5$ (b) $k = 10$ (c) $k = 20$

図 3: 効率性



(a) メモリ使用量 (b) 応答時間

図 4: 最大幅の影響

表 2: 精度

k	Method	Karate		Am-Rv	
		分散	エラー率	分散	エラー率
5	Pro(MC)	0.025	0.036	0	0
	Pro(HT)	0.025	0.036	0	0
	Sampling(MC)	0.025	0.037	$0.43{\cdot}10^{-4}$	0.061
	Sampling(HT)	0.029	0.042	$0.31{\cdot}10^{-4}$	0.059
10	Pro(MC)	0.013	0.058	0	0
	Pro(HT)	0.014	0.059	0	0
	Sampling(MC)	0.013	0.058	$0.099{\cdot}10^{-5}$	0.38
	Sampling(HT)	0.015	0.062	$0.12{\cdot}10^{-5}$	0.37
20	Pro(MC)	$0.76 \cdot 10^{-3}$	0.054	0	0
	Pro(HT)	$0.85{\cdot}10^{-3}$	0.057	0	0
	Sampling(MC)	$0.78{\cdot}10^{-3}$	0.056	$0.10{\cdot}10^{-3}$	1.00
	Sampling(HT)	$0.86{\cdot}10^{-3}$	0.057	$0.10{\cdot}10^{-3}$	1.00

る．これにより，近似解は頻繁にゼロとなり，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，小規模グラフにおいては厳密解を計算できることがわかる．

7 まとめ

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．提案アプローチは精度を損なわずにサンプリング数を削減させた．S^{2}BDD より効果的に下限値と上限値を求め，ネットワーク信頼性の効率的な近似計算を実現した．実験結果より，提案アプローチは既存の手法より 51.2 倍高速かつ高精度であることを示した．

謝辞

本研究は JST ACT-I（JPMJPR18UD）および科学研究費（JP15K21069）の支援によって行われた．謝意を表す．

参考文献

[1] Saurabh Asthana, Oliver D King, Francis D Gibbons, and Frederick P Roth. Predicting protein complex membership using probabilistic network reliability. *Genome research*, 14(6):1170–1175, 2004.
[2] Michael O Ball, Charles J Colbourn, and J Scott Provan. Network reliability. *Handbooks in operations research and management science*, 7:673–762, 1995.
[3] Lijun Chang, Jeffrey Xu Yu, Lu Qin, Xuemin Lin, Chengfei Liu, and Weifa Liang. Efficiently computing k-edge connected components via graph decomposition. In *SIGMOD*, pages 205–216, 2013.
[4] George 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.
[5] 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.
[6] Gary Hardy, Corinne Lucet, and Nikolaos Limnios. K-terminal network reliability measures with binary decision diagrams. *IEEE Transactions on Reliability*, 56(3):506–515, 2007.
[7] Ruoming Jin, Lin Liu, Bolin Ding, and Haixun Wang. Distance-constraint reachability computation in uncertain graphs. *PVLDB*, 4(9):551–562, 2011.
[8] Jun Kawahara, Takeru Inoue, Hiroaki Iwashita, and Shinichi 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.
[9] Xiangyu Ke, Arijit Khan, and Leroy Lim Hong Quan. An in-depth comparison of st reliability algorithms over uncertain graphs. *PVLDB*, 12(8):864–876, 2019.
[10] Rong-Hua Li, Jeffrey Xu Yu, Rui Mao, and Tan Jin. Recursive stratified sampling: A new framework for query evaluation on uncertain graphs. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):468–482, 2015.
[11] Eugène Manzi, Martine Labbé, Guy Latouche, and Francesco Maffioli. Fishman's sampling plan for computing network reliability. *IEEE Transactions on Reliability*, 50(1):41–46, 2001.
[12] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Chapman & Hall/CRC, 2010.
[13] Yuya Sasaki, Yasuhiro Fujiwara, and Makoto Onizuka. Efficient network reliability computation
in uncertain graphs. In *EDBT*, 2019.
[14] S Thompson. *Sampling*. Wiley, 2002.
[15] Leslie G Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.