Vol.2017-DBS-166 No.16
2017/12/23

BDDによるネットワーク信頼性の近似計算

佐々木 勇和[1,a)]　藤原 靖宏[2,1,b)]　鬼塚 真[1,c)]

概要：ネットワーク信頼性は任意の節点郡の接続性を評価するための重要な指標である．通信ネットワークの設計やバイラルマーケティングなど様々な応用があるが，#P 完全問題であるため計算コストが非常に大きい．ネットワーク信頼性を求める手法としての主流は，Binary decision diagram (BDD) を用いた厳密解を求めるアルゴリズムとサンプリングによる近似解を求めるアルゴリズムである．本研究では，グラフ規模に応じて適応的に近似解もしくは厳密解を計算する BDD を用いたアルゴリズムを提案する．BDD の最も大きな欠点はメモリ消費量が大きいことである．そこで，メモリ使用量を削減する Relaxed and Restricted BDD (R^2BDD) を提案する．R^2BDD はメモリ使用量を抑えつつ，下限値と上限値を算出可能である．また，上限値，下限値，および近似解を一度の R^2BDD の構築で算出するアルゴリズムを提案する．実データを用いた実験を行い，提案手法の性能を確認する．

キーワード：Binary decision diagram, ネットワーク信頼性，曖昧グラフ

Yuya Sasaki[1,a)]　Yasuhiro Fujiwara[2,1,b)]　Makoto Onizuka[1,c)]

1. はじめに

グラフはデータの関係性を表すための基礎的なデータ構造であり，多くのアプリケーションで利用されている．例えば，ソーシャルグラフやタンパク質ネットワーク，道路ネットワーク等のモデル化に用いられている．グラフは曖昧性を含んでることが多々ある [1]．例えば，タンパク質ネットワークにおいてはタンパク質の反応は確率的であり，道路ネットワークにおいては突然の渋滞や事故により道路が使用不可能になることがある．このようなグラフは，枝に存在確率がある曖昧グラフとしてモデル化することができる．

曖昧グラフにおいて，ネットワーク信頼性は接続性の指標としてよく使われており盛んに研究されている [7]．ネットワーク信頼性問題は，与えられた節点集合（ターミナルと呼ばれる）が相互接続する確率を求める問題である．ネットワーク信頼性問題は，与えられるターミナル数により，全ターミナル問題，2 ターミナル問題，k ターミナル問題と分けることができる．そのなかで k ターミナル問題が最も一般的で，任意の数のターミナルが与えられた問題である．ネットワーク信頼性問題は重要であるが，#P 問題として知られている．そのため，計算コストの大きさが問題である．最も単純なネットワーク信頼性を求める方法は，グラフの枝の全ての有無のパターンを列挙し，それぞれのパターンのグラフに対してターミナルが接続するかを求める．その後，接続となったグラフの存在確率の総和を求めることで求めることができる．節点数を $|V|$，枝数を $|E|$ とおいたとき，パターン数は $2^{|E|}$ となり，接続確認の計算量が $O(|V|+|E|)$ であるため，ネットワーク信頼性問題の計算量は，$O(2^{|E|}(|V|+|E|))$ となる．ネットワーク信頼

1 大阪大学大学院情報科学研究科
2 NTT ソフトウェアイノベーションセンタ
a) sasaki@osaka-u.ac.jp
b) fujiwara.yasuhiro@ntt.co.jp
c) onizuka@osaka-u.ac.jp

性の厳密解を求めるアルゴリズムとして，binary decision diagram (BDD) を用いた方法が最も効率的と知られている [8], [15]．BDD を用いたアルゴリズムは効率的であるが，メモリ使用量が $O(2^{|E|})$ となる問題点がある．そのため，非常に小さいグラフが限界である．また，実際のメモリ使用量を事前に算出できないという問題点もある．グラフの増大化に伴って，サンプリングにより近似解を求めるアプローチが近年では多く提案されている [6], [11], [12]．サンプリングを用いたアルゴリズムは，グラフの規模に関わらず近似解を求めることが可能であるが，2 つの大きな欠点がある．まず，小さいグラフに対しても厳密解を求めることができないことである．2 つ目は，中心極限定理やモンテカルロシミュレーションのような統計手法を用いるため，下限値および上限値を求めることができない [16]．これら 2 つの手法の欠点を避けつつ，利点をもつようなアルゴリズムは筆者らの知る限り存在していない．

そこで，本研究では，新たな BDD である R^2BDD を提案し，下限値，上限値，および近似解を一度の BDD 構築で求めるアルゴリズムを提案する．R^2BDD は，許容解を上限近似と下限近似の両方を同時に行うことで可能である [3]．一般的な BDD はメモリ使用量が大きいため，R^2BDD は Augumented ordered binary decision diagram (OBDD-A) をベースとして用いる [9]．OBDD-A は，それぞれの BDD 節点が確率値を追加情報を持つことが可能であり，BDD の子節点を計算した後，親節点は削除してもよい．そのため，メモリ使用量が BDD の高さに依存しなくなる．しかし，一層の BDD 節点の数が多くなると，BDD のメモリ使用量は非常に大きくなるため，一層毎の BDD 節点の数を抑制する最大幅を設けることで，メモリ使用量を小さくする．BDD 節点数が最大幅を超えた場合，その節点を削除し確率値のみを記憶する．また，削除した点に関してランダムサンプリングすることにより近似値も推定する．これにより，メモリ使用量を抑えつつ，上限値，下限値，近似値を求めることができる．メモリ使用量が大きくならない場合では，厳密解が計算可能である．

BDD の構築では，事前にグラフを小規模化することにより計算量を削減することができる．そこで，計算値の結果を変えずに小規模化するために，トポロジ情報を用いて行う．まず，ネットワーク信頼性計算に関係がない部分を枝刈りする．次に，グラフの等価性を保ちつつ枝および節点の削減を行う．最後に，グラフを分解することにより，複数のグラフに対してネットワーク信頼性を個別に計算可能にする．

本稿の構成は以下の通りである．2 章で関連研究，3 章で事前準備ついて述べる．4 章で提案手法の詳細を説明する．5 章で提案手法の評価を行い，6 章で本稿のまとめと今後の課題について論ずる．

2. 関連研究

曖昧グラフにおけるマイニングや問合せに関する研究はデータマイニングやデータベース分野で広く行われている．

ネットワーク信頼性問題: ネットワーク信頼性の計算方法は様々な提案手法が提案されており，BDD を用いる手法の他に最小カットや Factoring 定理に基づく手法等がある [2], [8], [9], [15], [18]．近年の研究成果により，BDD を用いた手法が最も効率的ということが一般的である．しかし，BDD はメモリ使用量が膨大であり，100 から 200 程度の枝数の処理が限界と知られている [8], [15]．そのため，大規模なグラフに適応することができない．

曖昧グラフにおける到達可能性問合せ: 到達可能性問合せは，s-t ネットワーク信頼性問題（節点 s から節点 t への接続確率）という特殊なネットワーク信頼性問題である．Jin らは，距離制約を加えた到達可能性問合せを対象として，サンプリングによる近似解手法を提案している [11]．Cheng らは，DAG を対象として到達可能性問合せを分散環境で計算する手法を提案している [5]．

曖昧グラフにおける他の問題: データマイニング分野に置いて曖昧グラフを対象としたマイニングが盛んに行われている．ネットワーク信頼性問題に関係する問題をいくつか述べる．まず，節点間の接続性を基にしたクラスタリングやサブグラフマッチがある [10], [19], [20]．Jin ら [10] は接続確率が閾値より高い節点集合の検出を行っており，モンテカルロシミュレーションを基にしたアルゴリズムを提案している．Zou ら [19] は，曖昧グラフのセットから頻出サブグラフの検出をモンテカルロシミュレーションにより行っている．さらに，Zou ら [20] は極大クリークの検出も行っている．Liu ら [14] は k-means を基にしたアルゴリズム Kollios ら [13] は編集距離を最大化するクラスタリングアルゴリズムを提案している．Bonch ら [4] は曖昧グラフにおけるコア分解アルゴリズムを提案している．

3. 事前知識

3.1 曖昧グラフ

曖昧グラフは $\mathcal{G} = (\mathbb{V}, \mathbb{E}, \mathbb{P})$ と表し，$\mathbb{V}$， $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$，$\mathbb{P} : \mathbb{E} = (0, 1]$ は，それぞれ節点集合，枝集合，および枝の存在確率値 $p(e)$ を計算する関数を表す．それぞれの枝の存在確率は他の枝と独立と想定する．

曖昧グラフの枝のサブセットをもつグラフを可能グラフとよぶ．可能グラフは $G_p = (\mathbb{V}, \mathbb{E}_G)$ と表す．可能グラフは，曖昧グラフとは異なり，枝は確率値を持たないかわりに，グラフそのものに存在確率をもつ．可能グラフの存在確率 $Pr[G_p]$ を以下の式で表される．

$$Pr[G_p] = \Sigma_{e \in \mathbb{E}_G} p(e) \cdot \Sigma_{e \in \mathbb{E} \setminus \mathbb{E}_G} (1 - p(e)). \qquad (1)$$

可能グラフの数は，それぞれの枝に対して有無を判定するため，$2^{|\mathbb{E}|}$ となる．

さらに，中間グラフ $G_i = (\mathbb{V}, \mathbb{E}_f, \mathbb{E}_d)$ を定義する．中間グラフ G_i は，削除された枝と固定された枝のセットを持ち，それ以外の枝は曖昧な枝として存在する．中間グラフの確率値 $Pr[G_i]$ は以下の式で表される．

$$Pr[G_i] = \Sigma_{e \in \mathbb{E}_f} p(e) \cdot \Sigma_{e \in \mathbb{E}} (1 - p(e)). \tag{2}$$

3.2 ネットワーク信頼性

ネットワーク信頼性はターミナルが相互接続している可能グラフの確率の総和で求めることができる．

定義：(k-terminal reliability). 任意の k 個の節点と曖昧グラフ $\mathcal{G}$ が与えられたとき, k ターミナル信頼性問題は接続確率 $R[\mathcal{G}]$ を求める．

$$R[\mathcal{G}] = \Sigma_{G \in \mathcal{G}} I(G, \mathbb{V}_k) \cdot Pr[G], \tag{3}$$

G は可能グラフ，$I(G, \mathbb{V}_k)$ は G が相互接続している場合，1 を返す関数である（接続していない場合は 0 を返す）．

ネットワーク信頼性問題は，#P 完全問題である [17].

3.3 Binary decision diagram

Binary decision diagram (BDD) はデータ構造である．BDD は DAG で表すことができ, 内部節点, 0-枝, 1-枝, 0-節点，および 1 節点から成る．それぞれの内部節点は，0-枝と 1-枝を持つ．ルートから 0-節点および 1-節点までの経路は，ブール値が真および偽となる変数状態を表す．BDD の特別な形として，augmented ordered binary decision diagram (OBDD-A) がある．OBDD-A はルートからの全ての経路が同じ変数順で表され，さらにそれぞれの節点が追加の情報をもつ．ネットワーク信頼性においては，それぞれの節点は，中間グラフを保持するとみなすことができ，その確率値を追加の情報として保持する．

4. 提案アルゴリズム

提案するアルゴリズムでは，新たな BDD である R^{2}BDD を構築しながら，ネットワーク信頼性を計算する．まず，R^{2}BDD について述べ，上限値，下限値，近似値の計算方法ついて説明する．その後，効率的にネットワーク信頼性を計算するための最適化について述べる．

4.1 R^{2}BDD

R^{2}BDD の構築は，グラフ問題においてサブグラフ列挙のフレームワークであるフロンティア法を用いる．フロンティア法は，グラフ探索において境界となる節点（フロンティア）の情報を保持し，フロンティアの情報が同じサブグラフをマージする方法である．これにより，探索空間が大幅に削減される．提案手法における R^{2}BDD では，ネットワーク信頼性の計算のために，フロンティアはコンポーネント識別子，次数，およびターミナル数を保持する．既に接続された節点は同じコンポーネント識別子を持ち，次数は接続されたそのコンポーネントの枝のうち未探索の数，およびターミナル数はコンポーネント内のターミナル数を表す．

R^{2}BDD の具体的な構築手順は，あらかじめ順序付けされた枝を固定もしくは削除をしながらフロンティアを広げながら行う．固定および削除は，BDD の枝における 1-枝と 0-枝を示す．また，それぞれの節点は，中間グラフを表しており，その確率値を追加情報として保持する．フロンティアのターミナル数が k となった場合，その内部節点の中間グラフは既に接続済みであるため，1-節点に遷移する．一方，ターミナル数が 1 以上の節点の次数が 0 となった場合，その内部節点の中間グラフはターミナルが接続することはないため，0-節点に遷移する．また，ある内部節点が 0-枝と 1-枝の遷移先となる内部節点を追加した後は，その遷移元となった内部節点を削除する．これにより，R^{2}BDD では，メモリ使用量が高さに依存しなくなり，レベル毎の内部節点数がメモリに依存する．ここで，レベル毎 l の最大内部節点数は以下の第 2 種スターリング数とベル数により表すことができる．

$$A_{i,j} j = \cdot A_{i-1,j} + A_{i-1,j-1} \tag{4}$$

ここで，$A_{i,1} = 1$，$0 < j \leq i$ の場合，$A_{i-1,j-1} = 1$ となる．

$$B_{|F_l|} = \Sigma_{j=1}^{|F_k|} A_{|F_k|,j} \tag{5}$$

内部節点数を抑制することがメモリ使用量の減少に重要である．そこで，R^{2}BDD では，内部節点数の最大数を決定する閾値を与える．これにより，メモリ使用量を計算することができる．一方で，このときどの内部節点を削除するかが重要である．ネットワーク信頼性の計算では，内部節点が保持する確率が大きいものが計算に与える影響が大きい．そこで，内部節点を確率値の降順に並べ替え，確率値が大きい内部節点を優先的に残す．また，削除した内部節点の確率を保持することにより，上限値・下限値の算出に用いる．

4.2 上限値，下限値，近似値の算出

上限値と下限値は，0-節点および 1-節点が保持する確率および削除した節点の確率を用いて計算することができる．一方，近似値は通常の BDD では計算することができない．そこで，削除した内部節点の中間グラフから可能グラフをサンプリングすることにより近似値を計算する．

4.3 最適化

ネットワーク信頼性の計算では，グラフのサイズを事前

に小さくすることにより，全体の計算量を小さくすることができる．また，BDD の構築では枝の順序付けが内部節点の数に大きな影響があることが知られている．

トポロジベースのグラフ最適化： グラフのサイズが小さくなれば，計算する変数が減るだけでなく，フロンティア数を減らすことができる．また，トポロジベースでグラフを最適化することにより，ネットワーク信頼性の精度を下げることなくサイズを小さくすることが可能である．グラフの最適化として，3 つの方法: 枝刈り，等価変換，分解を実施する．

まず，枝刈りでは，枝接続によりグラフを再構築し，シュタイナー木を計算し，シュタイナー木に含まれる枝と節点がネットワーク信頼性の計算に必要なグラフとなる．グラフの再構築の方法は，まず 1-辺連結グラフを構築し，連結点と橋を求める．連結点と連結グラフそのものを一つの節点（連結グラフ点と呼ぶ）とし，連結点と連結グラフ点を接続する．その後，それぞれの連結点を対応する橋で接続することにより，一つの木が生成できる．ターミナルとなっている連結点と連結点以外のターミナルを含む連結グラフ点が接続するよにシュタイナー木を構築する．構築されたシュタイナー木に含まれる節点と枝（および連結グラフ）が必要なサブグラフとなる．ここで，一般グラフにおけるシュタイナー木を求めるのは NP 困難であるが，木構造におけるシュタイナー木は $0(|V|+|E|)$ となる．

次に，グラフをより小さいものへ等価変換を実施する．まず，次数 2 が 2 である節点と隣接する二つの枝を削除し，削除した二つの枝の存在確率を掛けた存在確率をもつ枝を追加する．次に，並列（つまり，同じ端点をもつ枝）となっている枝を削除し，$(1-(1-p(e_1)\cdot(1-p(e_2))$ の存在確率をもつ枝を追加する．また，上記二つの処理により，ループとなる枝が生成される可能性があるため，ループとなっている枝は削除する．これらの処理をグラフに変化がなくなるまで繰り返す．等価変換では，ネットワーク信頼性の正確性を損なうことなくグラフサイズを小さくすることができる．

最後に，グラフを分解することにより，ネットワーク信頼性をそれぞれのグラフで計算することを可能にする．グラフの分解では，橋で接続されているコンポーネントに分解する．橋に接続する節点は，ターミナルがすべて接続するためには，必ず接続されていなければならない．そのため，グラフ分解後にネットワーク信頼性を計算する場合，それぞれのコンポーネントの橋との連結点をターミナルとして追加する．グラフを分解することにより，ネットワーク信頼性の計算量およびメモリ使用量を削減することができる．

枝の順序付け： BDD における重要な課題は，フロンティア数を削減することである．フロンティア数は枝の順序に大きく依存する．本研究では，フロンティア数が比較的小さくなることが知られている幅優先探索を使用する．よりよい枝の順序付けについては今後の課題である．

表 1: データセット

記号	節点数	枝数
CA-GrQc	4,158	13,428
DBLP	26,178	108,662

5. 実験

全てのアルゴリズムは C++で実装され，CPU が Intel Xenon E7-8860v4 @2.20GHz で，メモリが 256GB の計算機を用いた．アルゴリズムの評価としては，応答時間と上限値・下限値の幅を評価する．提案手法としては，グラフの最適化の有り，無しの場合を用いる．

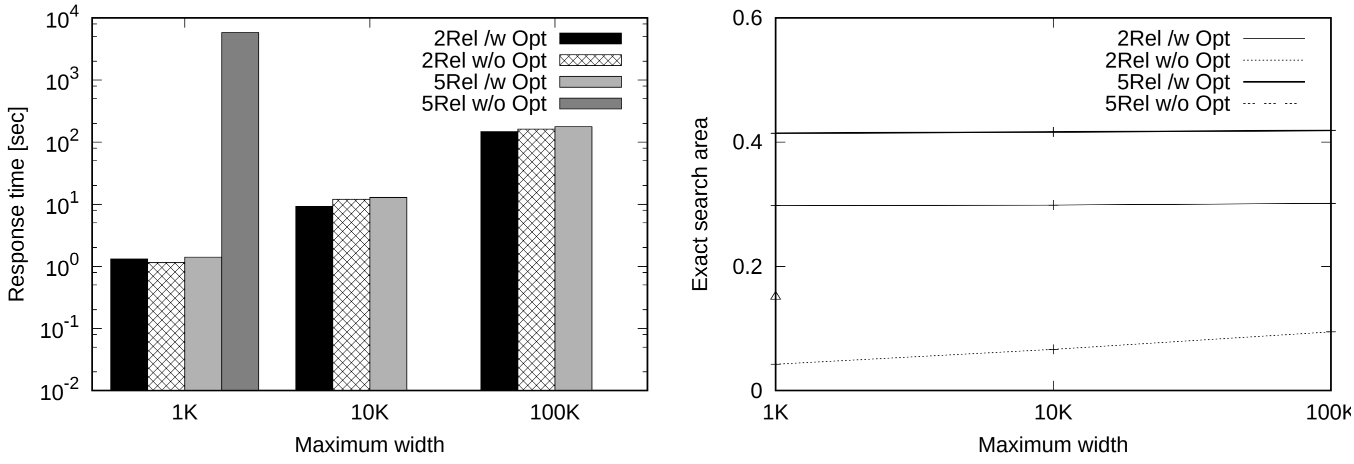
5.0.1 データセット

データセットの概要を表 1 に示す．データセットとして 2 つの共著関係を表すグラフを用いた．節点と枝は，それぞれ著者と共著関係を表す．CA-GrQc データセットは，Arkive の共著関係のグラフである [*1]．枝の存在確率は，正規分布に基づきランダムに決定する．DBLP データセットは，DBLP から取得したデータを用い，枝の確率は共著数に基づいて決定する．具体的には，α と α_M をそれぞれ共著数と最大共著数として，枝の存在確率を $\frac{\log(\alpha+1)}{\log(\alpha_M+2)}$ とする．

5.1 結果

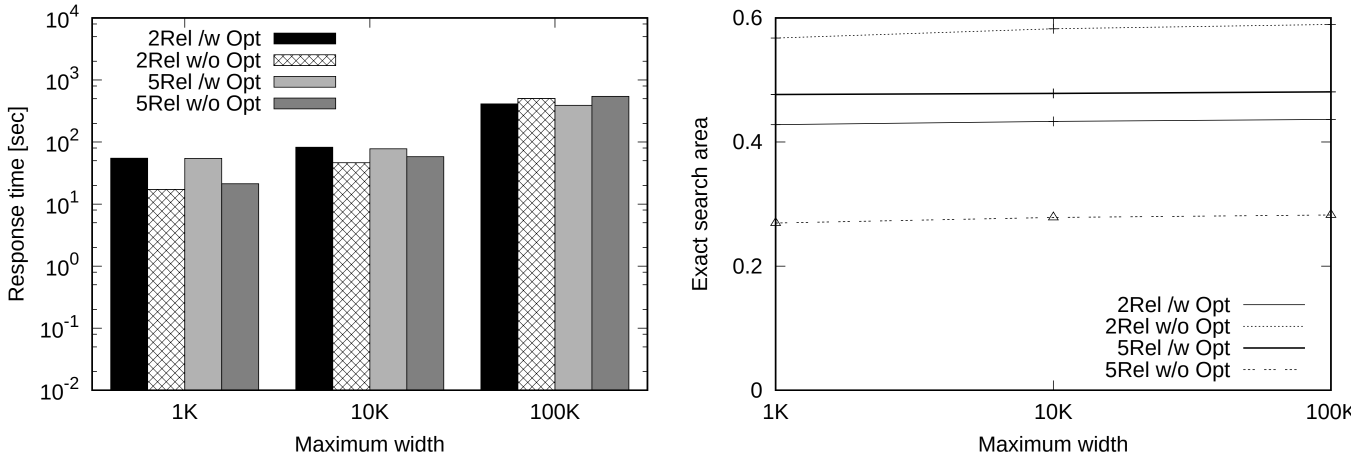
図 1 に CA-GrQc データセットにおける最大の幅を変えた場合の応答時間および探索済み領域の結果を示す．探索済み領域は，提案手法において接続もしくは非接続を厳密に判定した割合を表す．そのため，探索済み領域が大きいほど，上限値下限値の幅は狭くなる．提案手法の評価として，ターミナル数を 2 および 5 と設定し，グラフの最適化の有無を評価する．まず，図 1(a) よりターミナル数が 2 の場合，探索時間はほぼ変わらないことがわかる．これは，グラフの最適化に時間を要しているため，ネットワーク信頼性の計算時間を削減しても同程度となっている．一方で，ターミナル数が 5 の場合，最適化有りではターミナル数が 2 と同程度に対して，最適化無しの場合では非常に大きな時間が掛かっている．これは，グラフを小規模化することにより，複数のターミナルの接続性を判定しやすいことに起因する．最大幅を 10,000 以上とした場合，1 日では終了しなかったため処理を終了している．図 1(b) より，最適化の有無により，探索済み領域に大きな差があることがわかる．ターミナル数が 2 の場合，応答時間は同程度であるが，探索済み領域は 7 倍程度の差が開いている．これは，不要な枝および節点を削除することにより，そもそもの探索領域を削減しているためである．

[*1] http://snap.stanford.edu/data/



(a) 応答時間　　(b) 探索済み領域

図 1: CA-GrQc データセットにおける提案手法の比較



(a) 応答時間　　(b) 探索済み領域

図 2: DBLP データセットにおける提案手法の比較

次に図 2 に DBLP データセットにおける最大の幅を変えた場合の応答時間および探索済み領域の結果を示す．CA-GrQc データセットと異なり，ターミナル数が 2 の場合は，最適化無しの方が応答時間も短く，探索済み検索領域も大きい．これは，枝の順序付けにより最適化しない方がよいというケースになったためと考えられる．

6. おわりに

本研究は，ネットワーク信頼性の上限値，下限値，近似値を同時に出力可能なアルゴリズムを提案した．また，そのためのデータ構造として，R^2BDD を提案した．

今後の課題は，精度の向上と効率的な計算である．まず，精度の向上では，サンプリングによる定理を用いた精度の証明が第一の課題である．次に，より最適な BDD の節点の削除方法，および枝の順序を考案する予定である．さらに，上限値と下限値をそれぞれの別の BDD の構築することにより，改善することが期待できる．計算の効率化では，並列計算をすることが求められる．メモリ使用量を大きくしない並列計算方法を考案する．

7. 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．

参考文献

[1] Aggarwal, C. C.: *Managing and Mining Uncertain Data*, Vol. 35, Kluwer (2009).

[2] Agrawal, A. and Satyanarayana, A.: An O (— E—) time algorithm for computing the reliability of a class of directed networks, *Operations research*, Vol. 32, No. 3, pp. 493–515 (1984).

[3] Bergman, D., Cire, A. A., van Hoeve, W.-J. and Hooker, J.: *Decision diagrams for optimization*, Springer (2016).

[4] Bonchi, F., Gullo, F., Kaltenbrunner, A. and Volkovich,

Y.: Core decomposition of uncertain graphs, *SIGKDD*, pp. 1316–1325 (2014).

[5] Cheng, Y., Yuan, Y., Chen, L. and Wang, G.: The reachability query over distributed uncertain graphs, *ICDCS*, pp. 786–787 (2015).

[6] Cheng, Y., Yuan, Y., Chen, L., Wang, G., Giraud-Carrier, C. and Sun, Y.: Distr: a distributed method for the reachability query over large uncertain graphs, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 27, No. 11, pp. 3172–3185 (2016).

[7] Colbourn, C. J.: *The combinatorics of network reliability*, Vol. 200, Oxford University Press New York (1987).

[8] Hardy, G., Lucet, C. and Limnios, N.: K-terminal network reliability measures with binary decision diagrams, *IEEE Transactions on Reliability*, Vol. 56, No. 3, pp. 506–515 (2007).

[9] Herrmann, J. U. and Soh, S.: A memory efficient algorithm for network reliability, *Communications, 2009. APCC 2009. 15th Asia-Pacific Conference on*, IEEE, pp. 703–707 (2009).

[10] Jin, R., Liu, L. and Aggarwal, C. C.: Discovering highly reliable subgraphs in uncertain graphs, *SIGKDD*, pp. 992–1000 (2011).

[11] Jin, R., Liu, L., Ding, B. and Wang, H.: Distance-constraint reachability computation in uncertain graphs, *PVLDB*, Vol. 4, No. 9, pp. 551–562 (2011).

[12] Khan, A., Bonchi, F., Gionis, A. and Gullo, F.: Fast Reliability Search in Uncertain Graphs., *EDBT*, pp. 535–546 (2014).

[13] Kollios, G., Potamias, M. and Terzi, E.: Clustering large probabilistic graphs, *TKDE*, Vol. 25, No. 2, pp. 325–336 (2013).

[14] Liu, L., Jin, R., Aggarwal, C. and Shen, Y.: Reliable clustering on uncertain graphs, *ICDM*, pp. 459–468 (2012).

[15] Maehara, T., Suzuki, H. and Ishihata, M.: Exact Computation of Influence Spread by Binary Decision Diagrams, *Proceedings of the 26th International Conference on World Wide Web*, International World Wide Web Conferences Steering Committee, pp. 947–956 (2017).

[16] Rosenblatt, M.: A central limit theorem and a strong mixing condition, *Proceedings of the National Academy of Sciences*, Vol. 42, No. 1, pp. 43–47 (1956).

[17] Valiant, L. G.: The complexity of enumeration and reliability problems, *SIAM Journal on Computing*, Vol. 8, No. 3, pp. 410–421 (1979).

[18] Yeh, F.-M., Lu, S.-K. and Kuo, S.-Y.: OBDD-based evaluation of k-terminal network reliability, *IEEE Transactions on Reliability*, Vol. 51, No. 4, pp. 443–451 (2002).

[19] Zou, Z., Gao, H. and Li, J.: Discovering frequent subgraphs over uncertain graph databases under probabilistic semantics, *SIGKDD*, pp. 633–642 (2010).

[20] Zou, Z., Li, J., Gao, H. and Zhang, S.: Finding top-k maximal cliques in an uncertain graph, *ICDE*, pp. 649–652 (2010).