4K2-OS-16b-03

The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

ZDD を用いた種々のネットワーク設計問題の解法

Solving Various Network Design Problems Using ZDDs

鈴木　浩史　　石畠　正和　　湊　真一
Hirofumi Suzuki　　Masakazu Ishihata　　Shin-ichi Minato

北海道大学 大学院情報科学研究科
Graduate School of Information Science and Technology, Hokkaido University

Network design is an important issue for several services and systems such as transportation and telecommunication, and is formulated as *network design problem* using graph structures. Given a graph and constraints, the goal of the problem is to find a constrained subgraph that satisfies given constraints. For some representative constraints, many algorithms have been proposed. However, they are not enough in some applications, because more general and non-mathematical constraints can be required. Therefore, we aim to obtain a set of all constrained subgraphs for supporting to select a suitable one, and propose a unified approach for network design problems. Our approach describes a set of all constrained subgraphs by an equation using some set family operations. To process described equation, we utilize *zero-suppressed binary decision diagrams* (ZDDs) that manipulate set families in compact form. We applied our approach to some benchmark instances with representative constraints, and obtained sets of all constrained subgraphs in reasonable time.

1. はじめに

ネットワークは通信，配電，運送などの様々なサービスやシステムの基盤となっている．質の高いサービスやシステムを実現するためには，頑健性が高いネットワークや通信の遅延が短いネットワークなどが要求される．例えば，通信ネットワークでは，通信の妨害や機器の故障が起きても，通信がなるべく滞らずかつ高速に行われるべきである．このように，サービスやシステムの特性を考慮することはネットワーク設計において重要である．ネットワークはグラフ構造で表現され，頑健性や遅延の度合いはグラフの特性として記述できる．そのため，グラフ構造を用いてモデル化されたネットワーク設計問題が広く研究され，代替コミュニケーション [Balakrishnan 92]，野性生物保護 [Le Bras 13] などのネットワークを対象に応用されている．

ネットワーク設計問題は，候補グラフ集合および制約が与えられたとき，候補グラフ集合の中で制約を満たす目的グラフ集合から，コスト最小の目的グラフを探す問題である．例えば，通信機器を繋ぐ限られた配線の中で，通信の遅延が閾値以下となる安価なネットワークを探す問題はネットワーク設計問題に定式化できる．ネットワーク設計問題における代表的な制約として以下の 3 つが挙げられる [Kerivin 05].

- Hop-Constraints: 指定された頂点間の距離が指定長以下であることを要求する．例えば，通信ネットワークにおける機器間の通信において，経由するルータの数を制限し遅延を少なくすることに対応する．
- k-Edge- (Vertex-) Connected: $k-1$ 個の任意の辺（頂点）が削除されても，指定された頂点間が連結であることを要求する．例えば，通信ネットワークでいくつかの配線やルータが同時に故障しても，拠点間の通信が途切れないようにすることに対応する．
- Bounded-Rings: 各辺が指定長以下のサイクルに属することを要求する．Bounded-Rings を満たすならば，全頂点間が 2-Edge-Connected を満たす．そのため，Bounded-Rings は 2-Edge-Connected の特殊な場合として用いられている．

Contact: 鈴木 浩史, 北海道大学 大学院情報科学研究科, 北海道札幌市北区北 14 条西 9 丁目, 011-706-6514, h-suzuki@ist.hokudai.ac.jp

これらの制約に対して，これまでに様々なアルゴリズムが提案されている．例えば，[Diarrassouba 18] では候補グラフを任意の部分グラフ，Hop-Constraints および k-Vertex-Connected の両方を制約とし，辺重み総和を最小化する場合のアルゴリズムが提案されている．全域かつ連結な部分グラフを候補とし，Bounded-Rings の下で辺重みを最小化する場合のアルゴリズムが [Fortz 02] で提案されている．また，Hop-Constraints を満たし辺重み総和や辺数が最小な木型のネットワークを求める多様なアルゴリズムが知られている [Monteiro 14].

上記の手法は代表的な制約を満たすひとつのネットワークを発見する手法である．一方で，以下のように，ひとつのネットワークを発見するだけでは不十分な場合がある．

1. 状況に応じて制約が変化する場合がある．例えば，通信ネットワークの頑健性を求めつつも，一部の配線を同時に使用できない場合が考えられる．すなわち，より一般的な制約を伴うネットワーク設計問題を解く必要がある．
2. 数理的に記述できない制約が存在する場合がある．例えば，実機を用いた負荷テストにおいて基準を満たすか確認する場合が考えられる．このような場合，ネットワークの候補を幾つか提示し，その中から適切なものを人手で選択する方法が考えられる．

このため，既存のアルゴリズムを用いて実運用に適したネットワークが得られるとは限らない．そこで，本研究ではこれらの問題を解決するために，以下のような手法を考える．

1. 統一的なアプローチにより様々な候補グラフ集合や制約に対してネットワーク設計問題を解くことを可能にする．具体的には，目的グラフ集合を集合族およびそれらの演算のみで記述する．
2. 目的グラフ集合全体を求め，後から絞り込みや提示および最適化ができる形で保持することで，数理的に記述できない制約にも適応しやすくする．

ただし，管理すべき集合族の大きさがネットワークの大きさに対して指数的になりえるため，集合族を明示的に保持し演算することは非現実的である．そこで，集合族を圧縮処理するデータ構造 Zero-suppressed Binary Decision Diagram (ZDD) [Minato 93] を用いて集合族を管理する．

本稿では，提案手法が様々な制約を扱えることを示すために Hop-Constraints，k-Edge- (Vertex-) Connected，および Bounded-Rings の各制約に対するネットワーク設計問題の目的グラフ集合を，集合族およびそれらの演算のみで記述できることを示す．また，ZDD を用いて実際に目的グラフ集合を求める手順を示す．さらに，ネットワーク設計問題におけるベンチマークのグラフデータを用いて実験を行い，実用的な計算時間で目的グラフ集合を求められることを示す．

2. 準備

2.1 グラフの諸記法

頂点集合 V と辺集合 E からなるグラフを $G = (V, E)$ と書く．提案手法は，有向・無向のいずれのグラフも扱えるが，本稿では無向，すなわち $E \subseteq \{\{u, v\} \mid u, v \in V\}$ の場合について説明する．任意の辺部分集合 $X \subseteq E$ について，$V[X] := \bigcup_{\{u,v\}\in X}\{u, v\}$ を X による誘導頂点という．同様に，$G[X] := (V[X], X)$ を X による辺誘導部分グラフという．以下，辺や頂点の集合を大文字（例えば P），集合族を花文字（例えば $\mathcal{P}$），集合族の集合を筆記体（例えば $\mathscr{P}$）で記述する．

任意の頂点対 $(u, v) \in V^2$ について，$\mathcal{P}(u, v; G) \subseteq 2^E$ を G 中のすべての u-v パスを表す辺部分集合の族とする．$dst(u, v; G) := \min_{P\in\mathcal{P}(u,v;G)} |P|$ を u, v 間の距離という．頂点対 (u, v) と整数 k に対して集合 $\mathscr{P}_{\mathrm{EC}}(u, v; G, k) := \{\mathcal{P} \subseteq \mathcal{P}(u, v; G) \mid |\mathcal{P}| = k, \forall P, P' \in \mathcal{P}, P \cap P' = \emptyset\}$ を u-v 間の k-辺素パスの集合とする．すなわち，k-辺素パスとは互いに辺を共有しない k 個のパスからなる集合である．同様に，集合 $\mathscr{P}_{\mathrm{VC}}(u, v; G, k) := \{\mathcal{P} \subseteq \mathcal{P}(u, v; G) \mid |\mathcal{P}| = k, \forall P, P' \in \mathcal{P}, V[P] \cap V[P'] = \{u, v\}\}$ を u-v 間の k-点素パスの集合とする．すなわち，k-点素パスとは始点と終点以外で互いに頂点を共有しない k 個のパスからなる集合である．$\mathcal{R}(G) \subseteq 2^E$ を G 中のすべてのサイクルを表す辺部分集合の族とする．

2.2 ネットワーク設計問題

グラフ G に対して集合族 $\mathcal{X} \subseteq 2^E$ を候補グラフ集合，関数 $C\colon 2^E \to \{0, 1\}$ を制約と呼ぶ．このとき，

$$\mathcal{X}_C := \{X \in \mathcal{X} \mid C(X) = 1\} \tag{1}$$

である集合族 $\mathcal{X}_C \subseteq \mathcal{X}$ を目的グラフ集合と呼ぶ．ネットワーク設計問題とは，グラフ G と候補グラフ集合 $\mathcal{X}$ および制約 C が与えられたとき，目的グラフ集合 $\mathcal{X}_C$ を求める問題である．

候補グラフ集合 $\mathcal{X}$ の例として任意の部分グラフの集合，シュタイナー木の集合，全域連結部分グラフの集合などが考えられる．各候補グラフ $X \in \mathcal{X}$ について $C(X) = 1$ であるとき，X は制約 C を満たすという．現実のネットワーク設計問題において，以下のような代表的な制約が知られている [Kerivin 05].

- Hop-Constraints: 頂点対の集合 $M \subseteq V^2$ と整数 h が与えられる．辺部分集合 $X \in \mathcal{X}$ に対して，$G[X]$ が (M, h) による Hop-Constraints を満たすことを $C^h_{\mathrm{HC}}(X; M) = 1$ と表す．C^h_{HC} は以下のように定義される．

$$C^h_{\mathrm{HC}}(X; M) := \begin{cases} 1 & \forall(u, v) \in M, dst(u, v; G[X]) \le h, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

- k-Edge-Connected: 頂点対の集合 $T \subseteq V^2$ と整数 k が与えられる．辺部分集合 $X \in \mathcal{X}$ に対して，$G[X]$ が (T, k) による k-Edge-Connected を満たすことを $C^k_{\mathrm{EC}}(X; T) = 1$ と表す．C^k_{EC} は以下のように定義される．

$$C^k_{\mathrm{EC}}(X; T) := \begin{cases} 1 & \forall(u, v) \in T, \mathscr{P}_{\mathrm{EC}}(u, v; G[X], k) \ne \emptyset \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

- k-Vertex-Connected: 頂点対の集合 $T \subseteq V^2$ と整数 k が与えられる．辺部分集合 $X \in \mathcal{X}$ に対して，$G[X]$ が (T, k) による k-Vertex-Connected を満たすことを $C^k_{\mathrm{VC}}(X; T) = 1$ と表す．C^k_{VC} は以下のように定義される．

$$C^k_{\mathrm{VC}}(X; T) := \begin{cases} 1 & \forall(u, v) \in T, \mathscr{P}_{\mathrm{VC}}(u, v; G[X], k) \ne \emptyset \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

- Bounded-Rings: 整数 l が与えられる．辺部分集合 $X \in \mathcal{X}$ に対して，$G[X]$ が l による Bounded-Rings を満たすことを $C^l_{\mathrm{BR}}(X) = 1$ と書く．C^l_{BR} は以下のように定義される．

$$C^l_{\mathrm{BR}}(X) := \begin{cases} 1 & \forall e \in X, \exists R \in \mathcal{R}(G[X]), e \in R, |R| \le l, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

3. 提案手法

提案手法は，$\mathcal{X}_C$ を集合族上の演算を用いて記述し，その記述に従って演算を処理することで目的グラフ集合を求める．しかし，扱う集合族の大きさが $|E|$ に対して指数的になる場合があるため，集合族を明示的に保持し演算することは難しい．そこで，集合族の保持と演算のために，集合族を圧縮処理するデータ構造 ZDD を用いる．すなわち，ZDD がサポートする演算で $\mathcal{X}_C$ を記述し，必要な集合族の ZDD を構築できれば，提案手法により C を制約とするネットワーク設計問題を解くことができる．

本章では，まず 2.1 節で述べた各制約に対して，集合族上の演算を用いた $\mathcal{X}_C$ の記述を与える．その後，$\mathcal{X}_C$ の記述に必要な各集合族に対して，ZDD を構築できることを示す．

3.1 集合族演算による制約の記述

ZDD は基本的な演算 $\cup$（Union），$\cap$（Intersection），$\setminus$（Difference）に加えて以下のような演算をサポートしている [Minato 93, Knuth 09, Kawahara 17b]．ここで，$\mathcal{A}$ および $\mathcal{B}$ は集合族である．

- Restriction: $\mathcal{A} \lhd \mathcal{B} = \{A \in \mathcal{A} \mid \exists B \in \mathcal{B}, B \subseteq A\}$
- Disjoint Join: $\mathcal{A} \boxtimes \mathcal{B} = \{A \cup B \mid A \in \mathcal{A}, B \in \mathcal{B}, A \cap B = \emptyset\}$
- Division: $\mathcal{A}/\mathcal{B} = \{A \setminus B \mid A \in \mathcal{A}, B \in \mathcal{B}\}$

以下では，これらの演算を用いて各目的グラフ集合を記述する．

- Hop-Constraints: 集合 E を台集合とし大きさが h 以下の集合からなる集合族を $\mathcal{K}(h; E) := \{X \subseteq E \mid |X| \le h\}$ と定義する．このとき，各 $(u, v) \in M$ について，集合族

$\mathcal{P}^{\leq h}(u,v;G) \subseteq 2^E$ を

$$\begin{aligned}\mathcal{P}^{\leq h}(u,v;G) &:= \mathcal{P}(u,v;G) \cap \mathcal{K}(h;E) \\ &= \{P \in \mathcal{P}(u,v;G) \mid |P| \leq h\}\end{aligned} \tag{6}$$

と定義する．ここで，以下の補題が成り立つ．

補題 1. $\mathcal{X}_{C^h_{HC}} = \bigcap_{(u,v)\in M} \left(\mathcal{X} \lhd \mathcal{P}^{\leq h}(u,v;G)\right)$ である．

証明. 各 $(u,v) \in M$ について，演算 $\lhd$ の定義より，

$$\begin{aligned}&\mathcal{X} \lhd \mathcal{P}^{\leq h}(u,v;G) \\ &= \{X \in \mathcal{X} | \exists P \in \mathcal{P}^{\leq h}(u,v;G), P \subseteq X\} \\ &= \{X \in \mathcal{X} \mid dst(u,v;G[X]) \leq h\}\end{aligned} \tag{7}$$

である．よって，

$$\begin{aligned}&\bigcap_{(u,v)\in M} \left(\mathcal{X} \lhd \mathcal{P}^{\leq h}(u,v;G)\right) \\ &= \bigcap_{(u,v)\in M} \{X \in \mathcal{X} \mid dst(u,v;G[X]) \leq h\} \\ &= \{X \in \mathcal{X} \mid \forall (u,v) \in M, dst(u,v;G[X]) \leq h\} \\ &= \mathcal{X}_{C^h_{\mathrm{HC}}}\end{aligned} \tag{8}$$

である． □

- k-Edge-Connected: 各 $(u,v) \in T$ について，集合族 $\mathcal{D}_{\mathrm{EC}}(u,v;G,k) \subseteq 2^E$ を

$$\mathcal{D}_{\mathrm{EC}}(u,v;G,k) := \bowtie_{i=1,\ldots,k} \mathcal{P}(u,v;G) \tag{9}$$

と定義する．ここで，以下の補題が成り立つ．

補題 2. $\mathcal{X}_{C^k_{EC}} = \bigcap_{(u,v)\in T} (\mathcal{X} \lhd \mathcal{D}_{EC}(u,v;G,k))$ である．

証明は項の都合上割愛する．

- k-Vertex-Connected: 集合族 $\mathcal{A} \subseteq 2^E$ に対して，$\mathcal{V}[\mathcal{A}] := \{X \cup V[X] \mid X \in \mathcal{A}\}$ と定義する．このとき，各 $(u,v) \in T$ について，$\widehat{\mathcal{V}}[\mathcal{P}(u,v;G)] := \mathcal{V}[\mathcal{P}(u,v;G)]/\{\{u,v\}\}$ と定義する．さらに，集合族 $\mathcal{D}_{\mathrm{VC}}(u,v;G,k) \subseteq 2^E$ を

$$\mathcal{D}_{\mathrm{VC}}(u,v;G,k) := (\bowtie_{i=1,\ldots,k} \widehat{\mathcal{V}}[\mathcal{P}(u,v;G)])/\{V\} \tag{10}$$

と定義する．ここで，以下の補題が成り立つ．

補題 3. $\mathcal{X}_{C^k_{EC}} = \bigcap_{(u,v)\in T} (\mathcal{X} \lhd \mathcal{D}_{VC}(u,v;G,k))$ である．

証明は項の都合上割愛する．

- Bounded-Rings: 各 $e \in E$ について，集合族 $\mathcal{R}^{\leq l}(e;G) \subseteq 2^E$ を

$$\begin{aligned}\mathcal{R}^{\leq l}(e;G) &:= (\mathcal{R}(G) \lhd \{\{e\}\}) \cap \mathcal{K}(l;E) \\ &= \{R \in \mathcal{R}(G) \mid e \in R, |R| \leq l\}\end{aligned} \tag{11}$$

と定義する．同様に，集合族 $\mathcal{X}(e) \subseteq \mathcal{X}$ を

$$\begin{aligned}\mathcal{X}(e) &:= \mathcal{X} \lhd \{\{e\}\} \\ &= \{X \in \mathcal{X} \mid e \in X\}\end{aligned} \tag{12}$$

と定義する．ここで，以下の補題が成り立つ．

補題 4. $\mathcal{X}_{C^l_{BR}} = \bigcap_{e\in E} \left((\mathcal{X} \setminus \mathcal{X}(e)) \cup \left(\mathcal{X}(e) \lhd \mathcal{R}^{\leq l}(e;G)\right)\right)$ である．

証明. 各 $e \in E$ について，演算 $\lhd$ の定義より，

$$\begin{aligned}&\mathcal{X}(e) \lhd \mathcal{R}^{\leq l}(e;G) \\ &= \{X \in \mathcal{X}(e) \mid \exists R \in \mathcal{R}(G[X]), e \in R, |R| \leq l\}\end{aligned} \tag{13}$$

である．従って，

$$\begin{aligned}&(\mathcal{X} \setminus \mathcal{X}(e)) \cup \left(\mathcal{X}(e) \lhd \mathcal{R}^{\leq l}(e;G)\right) \\ &= \{X \in \mathcal{X} \mid e \in X \Rightarrow \exists R \in \mathcal{R}(G[X]), e \in R, |R| \leq l\}\end{aligned} \tag{14}$$

である．よって，

$$\begin{aligned}&\bigcap_{e\in E} \left((\mathcal{X} \setminus \mathcal{X}(e)) \cup \left(\mathcal{X}(e) \lhd \mathcal{R}^{\leq l}(e;G)\right)\right) \\ &= \bigcap_{e\in E} \{X \in \mathcal{X} \mid e \in X \Rightarrow \exists R \in \mathcal{R}(G[X]), e \in R, |R| \leq l\} \\ &= \{X \in \mathcal{X} \mid \forall e \in X, \exists R \in \mathcal{R}(G[X]), e \in R, |R| \leq l\} \\ &= \mathcal{X}_{C^l_{\mathrm{BR}}}\end{aligned} \tag{15}$$

である． □

3.2 Zero-suppressed Binary Decision Diagrams の構築

ZDD を構築する方法として集合演算を用いた bottom-up な手法 [Minato 93] と，フロンティア法による top-down な手法 [Kawahara 17a] がある．提案手法では候補グラフ集合 $\mathcal{X}$ を表現する ZDD がどのように与えられるかは気にしないが，例えばフロンティア法によりシュタイナー木の集合，全域連結部グラフの集合などの候補グラフ集合の ZDD が構築できる．また，目的グラフ集合 $\mathcal{X}_C$ の ZDD は $\mathcal{X}$ の ZDD に様々な演算を施すことで構築する．以下では 3.1 節の各補題で与えた記述に関わる各集合族の ZDD を構築できることを述べる．

ひとつの集合のみを持つ集合族 $\{X\}$ $(X \subseteq E)$，大きさが b 以下の集合からなる集合族 $\mathcal{K}(b;E)$ は，演算を用いずに ZDD を構築できる集合族の代表例として知られている．また，フロンティア法により $\mathcal{P}(u,v;G)$ および $\mathcal{R}(G)$ の ZDD を構築できる．さらに，集合族 $\mathcal{A} \subseteq 2^E$ の ZDD から，$\mathcal{V}[\mathcal{A}]$ の ZDD を構築する手法も知られている [Kawahara 17b].

4. 実験

提案手法の性能を評価するために計算機実験を行った．実験に用いたプログラムは C++（g++4.8.4 と -O3 オプション）で記述した．実験環境は OS が 64-bit Ubuntu 16.04 LTS，CPU が Intel Core i7-3930K 3.2 GHz，RAM が 64 GB である．

ネットワーク設計問題のベンチマークとして用いられているグラフのうち 4 つを SNDlib（http://sndlib.zib.de/home.action）から選択した．これら 4 つの各グラフに対して，候補グラフ集合 $\mathcal{X}$ を全域連結部分グラフの集合とし，制約 $C^7_{\mathrm{HC}}, C^8_{\mathrm{HC}}, C^2_{\mathrm{EC}}, C^3_{\mathrm{EC}}, C^7_{\mathrm{BR}}, C^8_{\mathrm{BR}}, C^7_{\mathrm{HC}} \wedge C^2_{\mathrm{EC}}, C^7_{\mathrm{HC}} \wedge C^7_{\mathrm{BR}}$ のそれぞれでネットワーク設計問題を解いた．ここで，Hop-Constraints と k-Edge-Connected における頂点対の集合は次のように定めた．グラフ G 中で接続する辺が多い順に 10 個の頂点を取り出し，拠点となる頂点集合 W とする．最も接続する辺が多い頂点 r に対して $M = T = r \times (W \setminus \{r\})$ とする．本実験では，計算

表 1: 計算時間（sec）と目的グラフ集合の大きさ $|\mathcal{X}|$（M.O. はメモリ制限により計算できなかったことを表す）

	計算時間（sec）				$|\mathcal{X}_C|$			
C	norway	india35	germany50	pioro40	norway	india35	germany50	pioro40
C^7_{HC}	0.046	5.676	0.222	85.074	4.1×10^{13}	4.3×10^{22}	2.9×10^{22}	3.1×10^{24}
C^8_{HC}	0.086	52.043	0.572	M.O.	4.4×10^{13}	4.5×10^{22}	3.9×10^{22}	M.O.
C^2_{EC}	0.032	42.005	5.221	43.658	5.0×10^{12}	1.1×10^{22}	6.9×10^{21}	2.1×10^{24}
C^3_{EC}	0.038	56.468	5.299	8.819	3.4×10^{10}	4.2×10^{20}	2.7×10^{19}	2.8×10^{21}
C^7_{BR}	0.048	76.534	0.779	6.929	1.7×10^{9}	1.6×10^{18}	9.2×10^{12}	1.4×10^{21}
C^8_{BR}	0.084	254.981	2.561	29.726	6.0×10^{9}	2.9×10^{18}	1.3×10^{14}	5.9×10^{21}
$C^7_{\mathrm{HC}} \wedge C^2_{\mathrm{EC}}$	0.138	125.531	7.539	243.711	5.0×10^{12}	1.1×10^{22}	4.8×10^{21}	5.9×10^{23}
$C^7_{\mathrm{HC}} \wedge C^7_{\mathrm{BR}}$	0.126	532.941	2.183	311.270	1.7×10^{9}	1.6×10^{18}	7.8×10^{12}	5.4×10^{20}

時間と目的グラフ集合の大きさ $|\mathcal{X}_C|$ を計測した．各グラフの名称，頂点数，辺数，候補グラフ集合の大きさ $|\mathcal{X}|$，また $\mathcal{X}$ の ZDD を構築するのにかかった時間を表 2 に示す．$\mathcal{X}$ の ZDD を構築する時間は十分に短い．

表 2: データセット

名称	$|V|$	$|E|$	$|\mathcal{X}|$	$\mathcal{X}$ の構築時間 (sec)
norway	27	51	4.8×10^{14}	0.001
india35	35	80	4.8×10^{23}	0.007
germany50	50	88	8.1×10^{23}	0.010
pioro40	40	89	3.8×10^{26}	0.009

各グラフおよび制約に対する計算時間と目的グラフ集合の大きさを表 1 に示す．pioro40 における制約 C^8_{HC} の場合を除いて各目的グラフ集合を求めることに成功した．norway と germany50 においては，いずれの制約に対しても高速に動作した．india35 と pioro40 においては，劇的に速いわけではないが十分に現実的な時間で動作したと言える．目的グラフ集合の大きさはいずれの場合も 10^9 を越え，最も多い場合では 10^{24} に及んでいる．この大きさの目的グラフ集合を求めることは，既存のアルゴリズムでは達成できなかったことであり，本実験の結果が提案手法の実用性を示すものであると言える．

5. おわりに

本稿では，ネットワーク設計問題に対する統一的なアプローチとして，目的グラフ集合を集合族およびそれらの演算で記述し，ZDD により実際に目的グラフ集合を求める手法を提案した．また，様々な設定での実験を行い，提案手法の有効性を示した．既存のアルゴリズムはひとつの目的グラフを求めるのに対して，提案手法は目的グラフすべてを求めている点が新しい．

目的グラフ集合の ZDD を構築できれば，線形関数の最適化問題や曲率がわかっている場合の劣モジュラ最大化問題などを解くことができる．すなわち，制約を満たしつつ安価なネットワークや，なるべく多くの地域を被覆できるネットワークなどを設計できる．

今後の課題として，より複雑な制約に対して提案手法による記述を試みること，いくつかの指標で最適解の求解を高速に行えるかを確認することなどが挙げられる．

謝辞

本研究の一部は科研・基盤 (S) 15H05711 の助成による．

参考文献

[Balakrishnan 92] Balakrishnan, A. and Altinkemer, K.: Using a Hop-Constrained Model to Generate Alternative Communication Network Design, *ORSA Journal on Computing*, Vol. 4, No. 2, pp. 192–205 (1992)

[Diarrassouba 18] Diarrassouba, I., Mahjoub, M., Mahjoub, A. R., and Yaman, H.: k-node-disjoint hop-constrained survivable networks: polyhedral analysis and branch and cut, *Annals of Telecommunications*, Vol. 73, No. 1, pp. 5–28 (2018)

[Fortz 02] Fortz, B. and Labbé, M.: Polyhedral results for two-connected networks with bounded rings, *Mathematical Programming*, Vol. 93, No. 1, pp. 27–54 (2002)

[Kawahara 17a] Kawahara, J., Inoue, T., Iwashita, H., and Minato, S.: Frontier-based Search for Enumerating All Constrained Subgraphs with Compressed Representation, *IEICE Trans. Fundamentals*, Vol. E100-A, No. 9 (2017)

[Kawahara 17b] Kawahara, J., Saitoh, T., Suzuki, H., and Yoshinaka, R.: Solving the Longest Oneway-Ticket Problem and Enumerating Letter Graphs by Augmenting the Two Representative Approaches with ZDDs, in Phon-Amnuaisuk, S., Au, T.-W., and Omar, S. eds., *Computational Intelligence in Information Systems*, pp. 294–305, Cham (2017), Springer International Publishing

[Kerivin 05] Kerivin, H. and Mahjoub, A. R.: Design of Survivable Networks: A survey, in *In Networks*, pp. 1–21 (2005)

[Knuth 09] Knuth, D. E.: *The art of computer programming: Bitwise tricks & techniques; binary decision diagrams, volume 4, fascicle 1* (2009)

[Le Bras 13] Le Bras, R., Dilkina, B., Xue, Y., Gomes, C. P., McKelvey, K. S., Schwartz, M. K., and Montgomery, C. A.: Robust Network Design for Multispecies Conservation, in *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI'13, pp. 1305–1312, AAAI Press (2013)

[Minato 93] Minato, S.: Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems, in *DAC*, pp. 272–277 (1993)

[Monteiro 14] Monteiro, M. S. R., Fontes, D. B. M. M., and Fontes, F. A. C. C.: Hop-Constrained Tree-Shaped Networks, in Butenko, S., Pasiliao, E. L., and Shylo, V. eds., *Examining Robustness and Vulnerability of Networked Systems*, No. 37 in NATO Science for Peace and Security Series - D: Information and Communication Security, pp. 192 – 208, IOS (2014)