Kobe University Repository : Kernel

PDF issue: 2025-12-05

双方向グラフの最大重み最小帰還辺集合問題について

荒木，徹也
増田，澄男
斎藤，寿樹
山口，一章

(Citation)
神戸大学大学院工学研究科・システム情報学研究科紀要,5:59-64

(Issue Date)
2013

(Resource Type)
departmental bulletin paper

(Version)
Version of Record

(URL)
https://hdl.handle.net/20.500.14094/81005528

神戸大学大学院工学研究科・システム情報学研究科紀要　第 5 号　59〜64 頁　2013 年
doi:10.5047/gseku.j.2013.002

双方向グラフの最大重み最小帰還辺集合問題について

荒木　徹也 [1]・増田　澄男 [1*]・斎藤　寿樹 [1]・山口　一章 [1]

[1] 工学研究科電気電子工学専攻

（受付:June 26, 2013　受理:August 9, 2013　公開:August 27, 2013）

キーワード：　有向グラフ，アルゴリズム，帰還辺集合，NP 完全，階層グラフ描画

任意の有向グラフを $G = (V, E)$ とする．任意の 2 頂点 $v, w \in V$ に対し，$(v \to w) \in E$ であるとき且つそのときに限り $(w \to v) \in E$ であるならば，G を双方向グラフと呼ぶ．本研究では，各辺が非負の重みをもつ 双方向グラフ G が与えられたときに，その最小帰還辺集合のうち，辺の重みの総和が最大のものを求める問題を考える．この問題は，階層グラフに対して，辺の交差が少ない直交描画を求める問題に応用可能である．本稿では，まず，双方向グラフに対して辺の重みの総和が最大の最小帰還辺集合を求める問題が NP 困難であることを証明する．次に，この問題に対するある発見的手法の有効性を，厳密解法との比較実験により示す．

1．緒　言

任意の有向グラフを $G = (V, E)$ とする．G において，頂点 v から w へ向かう辺が存在するとき，その辺を $(v \to w)$ と表す．E の任意の部分集合 F に対し，G から F の全ての辺を削除して得られるグラフを $G - F$ と表す．集合 F が，G 中の任意の有向閉路 C に対してその上の辺を 1 本以上含むとき，F は G の帰還辺集合であるという．次の問題 FAS は有名な NP 完全問題の一つである [1)]．

［FAS］任意の有向グラフ $G_{fas} = (V_{fas}, E_{fas})$ とある整数 K $(\leq |E_{fas}|)$ が与えられる．G_{fas} の帰還辺集合で，要素数 K 以下のものが存在するか？　□

有向グラフ $G = (V, E)$ において，任意の 2 頂点 $v, w \in V$ に対し，$(v \to w) \in E$ であるとき且つそのときに限り $(w \to v) \in E$ であるならば，G を双方向グラフと呼ぶ．双方向グラフには，$(v \to w), (w \to v)$ のように，同じ 2 頂点の間を結ぶ逆方向の有向辺の対が含まれる．このような各対のことを逆並行辺対と呼ぶ．

以下の 2 条件を満たす有向グラフ $H = (V_H, E_H)$ を複合双方向グラフと呼ぶことにする．

(a) $V_H = V_1 \cup V_2$ $(V_1 \cap V_2 = \emptyset)$, $E_H = E_1 \cup E_2 \cup E_3$ であり，部分グラフ $(V_1, E_1), (V_2, E_2)$ のそれぞれが双方向グラフである．

(b) $E_3 \subseteq \{(v \to w) \mid v \in V_1, w \in V_2\}$ である．

各辺に重みが与えられた有向グラフに対し，その最小帰還辺集合のうち，辺の重みの総和が最大のものを最大重み最小帰還辺集合と呼ぶ．また，各辺の長さが指定されているとき，有向グラフ中の任意の有向道の長さを，その上の辺の長さの総和と定義し，長さ最大の有向道を最長道と呼ぶことにする．

筆者らは，文献 2) において，階層グラフが与えられたときに，辺交差数が少ない直交描画のうち，水平線分の y 座標の個数が少ないものを求めるアルゴリズムを提案している *．ここで，辺交差数はグラフ描画の質を評価するための非常に重要な尺度であり，階層グラフの直交描画アルゴリズムを示した過去の文献 3), 4) でも，辺交差数を少なく抑えることを目的としている．それに対し，水平線分の y 座標の個数を少なくすることは，直交描画中の水平線分の間隔を拡げ，辺をたどることを容易にするという意味をもつが，辺交差数の削減に比べると重要ではないと考えられ，文献 3), 4) では特に考慮されていない．

文献 2) の描画アルゴリズムでは，各辺が非負の重みと 1 あるいは 0 の長さをもつ複合双方向グラフ H に対して，H の最大重み最小帰還辺集合 X のうち，$H - X$ における最長道の長さが最小となるものを求めるという問題を考え，それを MWFAS と呼ばれる発見的手法により解いている．この問題において，$H - X$ における最長道の長さを考えているのは，直交描画における水平線分の y 座標の個数を少なくするためである [2)]．より重要度の高い辺交差数削減だけを目的とする場合には，次の問題 A を解けばよい．

［問題 A］各辺が非負の重みをもつ複合双方向グラフ H が与えられたとき，H の最大重み最小帰還辺集合を一つ求めよ．□

＊　階層グラフの直交描画では，垂直・水平線分を用いて辺を描く．直交描画における辺の交差数とは，異なる辺の垂直・水平線分が交差する回数である．描画中の全水平線分の集合を，同じ y 座標をもつ水平線分の極大な部分集合に分割したものとする．このとき，水平線分の y 座標の個数とは，そのような部分集合の総数を意味する．直交描画の例や詳しい説明については，文献 2) を参照されたい．

複合双方向グラフ H は二つの双方向グラフとそれらの間を結ぶ辺の集合 E_3 よりなるが，H の任意の最小帰還辺集合は E_3 の辺を含まない．よって，問題 A を解くためには，二つの双方向グラフのそれぞれに対して，次の問題 B を解けばよい．

［問題 B］各辺が非負の重みをもつ双方向グラフ G が与えられたとき，G の最大重み最小帰還辺集合を一つ求めよ．　□

本研究は，まず問題 B の NP 困難性を証明する．次に，文献 2) のアルゴリズム MWFAS に対して，$H - X$ の最長道の長さを考慮するための部分を除くなどの変更を行うことにより，問題 B に対する発見的手法（find_FAS1 と呼ぶ）を示す．一般に，発見的手法が求める解の質の評価を行う一つの方法は最適解との比較を行うことであるが，問題 B に対する厳密解法は知られていない．そこで本研究ではそのような厳密解法を提案する．そして，階層グラフの直交描画アルゴリズム [2] で生成されるグラフデータを用いた計算機実験により，発見的手法 find_FAS1 の有効性の評価を行う．

以下，2 章において，問題 B が NP 困難であることを証明する．3 章では，問題 B に対する発見的手法 find_FAS1 を示す．4 章では，問題 B に対する厳密解法を提案する．5 章では，その厳密解法との比較実験により，find_FAS1 の有効性を示す．最後に 6 章において，本稿の結果をまとめ，今後の課題について述べる．

2．問題 B の NP 困難性の証明

1 章で述べたように，有向グラフの最小帰還辺集合を求める問題は一般に NP 困難である [1]．しかし，双方向グラフ $G = (V, E)$ の最小帰還辺集合を求めることは容易である．例えば，V の頂点を任意の順序で水平線上に左から右へと並べ，より右にある頂点から左の頂点に向かう辺全てからなる集合を X とすれば，X は G の帰還辺集合となる．X は，G 中の $|E|/2$ 個の逆並行辺対のそれぞれからちょうど 1 本の辺を含むから，$|X| = |E|/2$ である．G の任意の帰還辺集合が $|E|/2$ 以上本の辺を含むことは明らかであるから，X は G の最小帰還辺集合である．複合双方グラフ H については，二つの双方向グラフの最小帰還辺集合を求めれば，それらの和集合が H の最小帰還辺集合となる．

本章では，次の判定問題 B′ が NP 完全であることを示す．

［問題 B′］各辺が非負の重みをもつ双方向グラフ G と自然数 B が与えられる．G の最小帰還辺集合のうちに辺の重みの総和が B 以上となるものは存在するか．　□

問題 B′ がクラス NP に属することは自明である．問題 B′ の NP 困難性を証明するため，1 章で示した問題 FAS からの変換を示す．

FAS の入力として，有向グラフ $G_{fas} = (V_{fas}, E_{fas})$ と非負整数 K $(\leq |E_{fas}|)$ が与えられたとき，問題 B′ の入力として，双方向グラフ $G = (V, E)$ と自然数 B を作る．まず，V と E は以下のように定める．

$$V = V_{fas}, \tag{1}$$

$$E = E_{fas} \cup E_{new}, \tag{2}$$

$$E_{new} = \{(v \to w) \mid (w \to v) \in E_{fas} \text{かつ} (v \to w) \notin E_{fas}\}. \tag{3}$$

E の辺のうち，E_{fas} に属していたものは全て重みを 0 とし，E_{new} に属するものは全て重みを 1 とする．最後に，B を次式で定める．

$$B = \frac{|E|}{2} - K. \tag{4}$$

以上の変換が，$|V_{fas}| + |E_{fas}|$ に関する多項式時間で実行できることは明らかである．Fig. 1 に簡単な例を示す．問題 FAS の入力が Fig. 1(a) のグラフと $K = 2$ であったとすると，変換により作られるグラフ G は Fig. 1(b) のようになり，B は $12/2 - K = 4$ となる．なお，Fig. 1(b) において，各辺の近くに示している整数は辺の重みである．

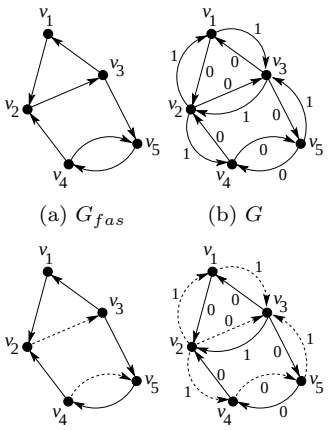

Fig. 1　FAS から問題 B′ への変換の例

グラフ G が，辺の重みの総和が B 以上の最小帰還辺集合 X をもつものとする．X は G の逆並行辺対のそれぞれから 1 本の辺を含み，$|X| = |E|/2$ である．集合 $X \cap E_{fas}$ を X' とおく．G では，E_{new} の各辺が重み 1 をもち，E_{fas} の各辺の重みは 0 であるから，$|X \cap E_{new}| \geq B$．よって

$$\begin{aligned} |X'| &= |X| - |X \cap E_{new}| \\ &\leq \frac{|E|}{2} - B \\ &= K \end{aligned} \tag{5}$$

である．グラフ $G - X$ はアサイクリックであり，$E_{fas} - X' \subseteq E - X$ であるから，グラフ $G_{fas} - X'$ もアサイクリックである．よって，X' は G_{fas} の帰還辺集合である．

次に，G_{fas} の任意の最小帰還辺集合を X' とし，$|X'| \leq K$ であるものとする．X' の任意の要素を $(v \to w)$ としたとき，X' の最小性より，$G_{fas} - X'$ には w から v への有向道が存在する．$G_{fas} - X'$ はアサイクリックであるから，次の (a) が成立する．

(a) 任意の $(v \to w) \in X'$ に対し，グラフ $G_{fas} - X'$ には v から w への有向道は存在しない．

さらに，(a) と X' の最小性より，次の (b) が成立する．

(b) グラフ G_{fas} が逆並行辺対 $\{(v \to w), (w \to v)\}$ をもつ場合，2 辺 $(v \to w), (w \to v)$ の両方が X' に含まれることはない．

G において，集合 X を以下のように定めるものとする．

$$X = X' \cup \{(v \to w) \in E_{new} \mid (w \to v) \notin X'\}. \tag{6}$$

例えば，Fig. 1(a) のグラフ G_{fas} に対し，Fig. 1(c) に破線で示した辺の集合を X' とすると，G における集合 X は Fig. 1(d) に破線で示した辺の集合となる．

グラフ $G-X$ は，$G_{fas}-X'$ に，$\{(w \to v) \in E_{new} \mid (v \to w) \in X'\}$ なる集合の辺を加えたものである．任意の $(v \to w) \in X'$ に対し，前述の (a) より，$G_{fas}-X'$ には v から w への有向道は存在しない．よって，$G-X$ に有向閉路は存在しないから，X は G の帰還辺集合である．

前述の (b) と Eq. (6) より，集合 X は G の逆並行辺対のそれぞれからちょうど 1 本の辺を含む．よって $|X|=|E|/2$ であり，X は G の最小帰還辺集合である．$|X'| \leq K$ であるから，X の辺の重みの総和 $|X|-|X'|$ は $|E|/2-K=B$ 以上である．

以上の議論より，NP 完全問題 FAS は問題 B′ に多項式変換可能である．前述のように問題 B′ はクラス NP に属するから，次の定理が成立する．

［定理 1］双方向グラフ G の各辺の重みを 0 あるいは 1 に限定した場合でさえ，問題 B′ は NP 完全である．□

この定理より，問題 A が NP 困難であることも容易に分かる．したがって，問題 A, B ともに，最適解を求める多項式時間アルゴリズムの存在は絶望的である．

3．問題 B に対する発見的手法

本章では，問題 B に対する発見的手法 find_FAS1 を示す．

文献 2) の階層グラフ描画アルゴリズムでは，各辺が非負の重みと 1 あるいは 0 の長さをもつ複合双方向グラフ H に対して，H の最大重み最小帰還辺集合 X のうち，$H-X$ における最長道の長さが最小となるものを求めるという問題を考え，それを発見的手法 MWFAS により解いている．find_FAS1 は，MWFAS に対し，$H-X$ の最長道の長さを考慮するための部分を除くなどの変更を行ったものである．

find_FAS1 の記述を以下に示す．問題 B の入力として，各辺が非負の重みをもつ双方向グラフ $G=(V,E)$ が与えられたとき，この方法は，G の最小帰還辺集合のうち，辺の重みの総和が大きなもの X を求める．

(1) $X=\emptyset$，$G'=G$ とする．
(2) G' から全ての孤立点を削除する．
(3) $G'=(\emptyset,\emptyset)$ となるまで，次の (3a)〜(3c) を実行する．
　(3a) G' の頂点のうち，(射入辺の重みの和) − (射出辺の重みの和) の値が最大のものを任意に選び，v^* とする．
　(3b) G' における v^* の射入辺を全て X に加える．
　(3c) G' から頂点 v^* を削除し，さらに G' から全ての孤立点を削除する．
(4) 集合 X を出力する．

有向グラフの最小帰還辺集合を求める問題に対し，文献 5) は発見的手法 GR を提案している．有向グラフにおいて，射出辺をもたない頂点をシンク，射入辺をもたない頂点をソースと呼ぶ．任意の有向グラフ G_{fas} が与えられたとき，GR は以下の処理を順に行う．

(i) シンクあるいはソースを見つけてグラフ G_{fas} から削除する処理を，G_{fas} 中にシンク及びソースがなくなるまで繰返す．
(ii) (射出辺の本数) − (射入辺の本数) の値が最大の頂点 v を見つけ，v の射入辺を帰還辺集合に加えた後，v を G_{fas} から削除する．この処理を，G_{fas} が $(\emptyset,\emptyset)$ になるまで繰返す．

本章で示した find_FAS1 は，GR に

- 入力を双方向グラフに限定する．
- 処理 (i) を，孤立点の削除を行う処理にする．
- 処理 (ii) を，辺の重みを考慮した処理にする．

という変更を行ったものだとみなすことができる．

詳細は省略するが，適切なデータ構造を用いることにより，アルゴリズム find_FAS1 を $O((|V|+|E|)\log|V|)$ 時間で実行することができる．

問題 A に対しては，与えられた複合双方向グラフ $H=(V_H,E_H)$ のそれぞれの双方向グラフに対して find_FAS1 を実行し，得られた集合の和集合を出力するという方法を考える．この方法を find_FAS2 と呼ぶことにする．find_FAS2 の時間計算量は $O((|V_H|+|E_H|)\log|V_H|)$ である．

4．問題 B に対する厳密解法

前章で，問題 B に対する発見的手法 find_FAS1 を示した．発見的手法が求める解の質を評価する一つの方法は，最適解との比較を行うことである．そこで本章では，問題 B に対する厳密解法を提案する．このアルゴリズムを Opt と呼ぶ．

各辺が非負の重みをもつ双方向グラフ $G=(V,E)$ が与えられるものとする．$|E|=2m$ とし，$E=\{e_1,e_2,\ldots,e_{2m}\}$ であるものとする．$j=1,2,\ldots,2m$ に対し，辺 e_j の重みを wt_j と表す．$j=1,2,\ldots,m$ について，e_{2j-1} と e_{2j} が逆並行辺対をなしており，$wt_{2j-1} \leq wt_{2j}$ であるものとする．

アルゴリズム Opt は，G に対し，E の部分集合で，各逆並行辺対からちょうど 1 本の辺を含むようなものをしらみつぶしで探索していく．そして，そのような部分集合で有向閉路を含まないもののうち，辺の重みの総和が最小のもの F_{min} を求め，G の最大重み最小帰還辺集合として $E-F_{min}$ を出力する．

Opt は，簡単な分枝限定法によるものである．Fig. 2 に，Opt が用いる探索木を示す．この木の高さは m であり，各葉の深さも m である．Fig. 2 中に示したように，$j=0,1,\ldots,m-1$ について，深さ j の各節点から左の子への辺には e_{2j+1} を，右の子への辺には e_{2j+2} を，それぞれ割り当てている．Opt は，この木の節点を，先行順 (preorder)[6] で構成していく．

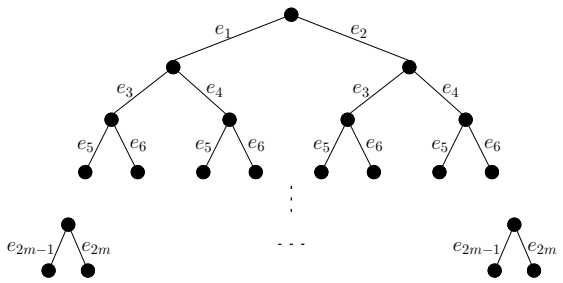

Fig. 2　探索木

Opt は，まず前処理として，次の (a), (b) を実行する．

(a) 各整数 $j=1,2,\ldots,m$ について，

$$wt_{2j-1}=0,\ wt_{2j}=wt_{2j}-wt_{2j-1} \tag{7}$$

とする．辺の重みをこのように変更したグラフの最大重み最小帰還辺集合は，変更前のグラフの最大重み最小帰還辺集合でもある．

(b) 重み変更後の G に発見的手法 find_FAS1 を適用し，得られる集合を X とする．$E-X$ を初期暫定解とし，$E-X$ の辺の重みの総和を W_{min} とする．

前処理を行った結果，もし $W_{min}=0$ であれば，find_FAS1 の求めた集合 X が最適解（G の最大重み最小帰還辺集合）である．この場合には，探索木を全く作ることなく Opt を終了する．一方 $W_{min}>0$ であれば，探索木の節点を前述のように先行順で構成していきながら，F_{min} の探索を行う．

探索木の各節点 x に対し，根から x への道上の辺に割り当てられている（E の）辺を全て集めて得られる集合を E_x とし，E_x に属する辺の重みの総和を W_x とする．探索木の根 r に対しては，$E_r=\emptyset$, $W_r=0$ とする．探索木の各葉 l に対する E_l は，E の部分集合で，各逆並行辺対からちょうど 1 本の辺を含むものになっている．よって，E_l が有向閉路を含まないような葉 l のうち，W_l が最小のものを見つけ，$F_{min}=E_l$ とすればよいことになる．

探索木のある節点 p から子 x をたどったものとする．p と x の間の辺に割り当てられた（E の）辺が $e_k=(v\rightarrow w)$ であるものとする．このとき，以下の (i)〜(iii) の処理を行う．

(i) $W_x=W_p+wt_k$ とする．この値が，暫定解の辺の重みの和 W_{min} 以上であれば，x の子孫の任意の葉 l に対して $W_l\geq W_{min}$ となるから，x での処理を打ち切る（探索木において x の真の子孫となる節点は構成しない）．

(ii) E_p の辺のみで w から v への有向道を構成できる場合も，$E_p\cup\{e_k\}$ が有向閉路を含むことになるから，x での処理を打ち切る．

(iii) (i), (ii) で処理が打ち切りにならなかった場合，$E_x=E_p\cup\{e_k\}$ とする．もし x の深さが m である（即ち，x が探索木の葉である）ならば，E_x を新たに暫定解とし，$W_{min}=W_x$ とする．

以上，分枝限定法により問題 B の最適解を求めるアルゴリズム Opt について述べた．この方法は指数時間アルゴリズムである．問題 A に対して最適解を求めるためには，与えられた複合双方向グラフ H のそれぞれの双方向グラフに対して Opt を実行し，得られた集合の和集合を出力すればよい．

5．計算機実験

問題 B に対する発見的手法 find_FAS1 と厳密解法 Opt との比較実験を行った．本章では，その実験の方法と結果について述べる．

まず，単純な階層グラフをランダムに作成した．グラフの階層数，頂点数，辺数の組合せは $(2,20,20)$, $(2,20,40)$, $(2,20,60)$, $(3,20,20)$, $(3,20,40)$, $(3,20,60)$, $(4,20,20)$, $(4,20,40)$, $(4,20,60)$, $(8,40,40)$, $(8,40,60)$, $(8,40,80)$, $(8,40,120)$ の 13 通りとし，それぞれの組合せについて 200 個のグラフを用意した（階層グラフを描画する際には，連続していない階層の頂点を結ぶ辺上にダミー頂点と呼ぶ仮の頂点を導入する．ここで示した頂点数及び辺数は，ダミー頂点を導入する前のものである）．次に，合計 2600 個の階層グラフのそれぞれに対して文献 2) の直交描画アルゴリズムを実行し，その第 5 段階のステップ 3 で作成される各複合双方向グラフから二つの双方向グラフを取りだした．各辺の重みの値は，直交描画アルゴリズムが定めたものをそのまま用いた．階層数 h の各階層グラフから，$h-1$ 個の複合双方向グラフが作られるので，双方向グラフは $2h-2$ 個作られることになる．以上のようにして作成した双方向グラフのうち，辺数が 6 以上のものを実験データとし，発見的手法である find_FAS1 と厳密解法である Opt を実行した．実験に使用した計算機の CPU は Intel Core i7 2600，OS は Linux 2.6，プログラミング言語は Java 6.0 である．

実験結果を Table 1 に示す．この表の左から 2〜4 列は実験データの作成に用いた階層グラフの階層数，頂点数，辺数を示している．5〜7 列目は，作成した双方向グラフに関する情報で，$|E|$ は辺数の平均値を，W_{all} は全ての辺の重みの総和の平均値を，それぞれ示している．8〜9 列目の W_{sol} は，それぞれの手法で求めた最小帰還辺集合の辺の重みの総和の平均値である（Opt についても，前処理 (a) を実行する前の値を使って，重みの総和を計算している）．また，10 列目の最適化率とは，発見的手法である find_FAS1 により最適解が得られたデータの割合である．11〜12 列目には，それぞれの手法の実行時間を示している．

Table 1 を見ると，(a) ∼ (m) のいずれの場合についても，find_FAS1 の W_{sol} の値は，厳密解法である Opt の W_{sol} の値

Table 1. 実験結果

	階層グラフ			双方向グラフ			W_{sol}		最適解	実行時間 [msec]	
	階層数	頂点数	辺数	個数	$\|E\|$	W_{all}	Opt	find_FAS1	率 [%]	Opt	find_FAS1
(a)	2	20	20	102	9.27	15.61	9.74	9.58	87.26	0.056	0.048
(b)	2	20	40	315	30.41	55.09	34.51	33.60	49.21	0.989	0.094
(c)	2	20	60	371	42.50	84.94	54.30	52.89	43.40	12.845	0.145
(d)	3	20	20	36	8.06	11.31	6.92	6.86	94.44	0.020	0.016
(e)	3	20	40	392	17.57	30.30	18.97	18.58	70.92	0.070	0.047
(f)	3	20	60	471	28.16	53.59	33.79	32.95	52.65	0.638	0.089
(g)	4	20	20	10	6.80	8.00	5.00	5.00	100.00	0.027	0.023
(h)	4	20	40	389	14.54	23.10	14.38	14.16	81.49	0.066	0.046
(i)	4	20	60	579	20.84	35.91	22.59	22.04	64.77	0.278	0.070
(j)	8	40	40	39	6.97	9.13	5.64	5.64	100.00	0.030	0.026
(k)	8	40	60	543	16.46	22.04	13.51	13.32	84.35	0.198	0.067
(l)	8	40	80	868	24.10	34.11	21.16	20.73	73.62	1.339	0.107
(m)	8	40	120	1308	35.90	55.66	35.04	34.04	60.70	5739.313	0.133

に近くなっている．

今回の実験で使用した双方向グラフでは，辺数の最大値は 218 であった．この辺数最大のデータに対し，指数時間アルゴリズムである Opt の実行時間は約 38 分であったが，find_FAS1 の実行時間はわずか 1.54msec であった．

以上より，今回実験した規模の階層グラフに直交描画アルゴリズムを適用して得られる双方向グラフに対しては，find_FAS1 は有効であることが分かる．

6．結　言

本稿では，まず，各辺が非負の重みをもつ双方向グラフに対し，その最小帰還辺集合のうち，辺の重みの総和が最大のものを求める問題が NP 困難であることを証明した．次に，この問題に対する発見的手法 find_FAS1 と厳密解法 Opt を示した．階層グラフの直交描画アルゴリズムで生成される双方向グラフを用いて計算機実験を行ったところ，find_FAS1 は最適解に比較的近い解を高速に求めることができていた．

より規模の大きな双方向グラフに対しては，find_FAS1 が求める解と最適解との差が広がっていくものと予想される．今後，Opt の高速化を行うことにより，そのような場合の検討も行っていきたいと考えている．また，さらに 最適解に近い 解を求め得るよう，find_FAS1 の改良も行っていく予定である．

［謝辞］貴重なご意見をいただいた査読者の方に深く感謝する．本研究の一部は，科学研究費補助金（課題番号：24500040）の援助を受けて行ったものである．

References

1) M.R. Garey and D.S. Johnson, Computers and Intractability - A Guide to the Theory of NP-Completeness, Freeman, New York, NY (1979).

2) T. Araki, S. Masuda, I. Matoba, K. Yamaguchi, and T. Saitoh, “An algorithm for finding an orthogonal drawing of a hierarchical graph” (in Japanese), Technical Report of IEICE, COMP2013-7 (2013); also submitted to Transactions of IEICE.

3) G. Sander, “Layout of directed hypergraphs with orthogonal hyperedges,” Proc. 11th Int’l Symp. on Graph Drawing (GD2003), Lecture Notes in Computer Science, vol.2912, pp.381-386, Springer, Berlin (2004).

4) T. Eschbach, W. Günther, and B. Becker, “Orthogonal hypergraph drawing for improved visibility,” Journal of Graph Algorithms and Applications, vol.10, pp.141-157 (2006).

5) P. Eades, X. Lin and W.F. Smyth, “A fast and effective heuristic for the feedback arc set problem,” Information Processing Letters, vol.47, pp.319-323 (1993).

6) A.V. Aho, J.E. Hopcroft, and J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA (1974).

On the Maximum-Weight Minimum Feedbak Arc Set Problem for Bidirected Graphs

Tetsuya ARAKI[1], Sumio MASUDA[1], Toshiki SAITOH[1], and Kazuaki YAMAGUCHI[1]

[1] *Graduate School of Engineering, Department of Electrical and Electronic Engineering*

Key words: Directed Graph, Algorithm, Feedbak Arc Set, NP-Complete, Hierarchical Graph Drawing

A bidirected graph is a directed graph (V, E) such that, for any two vertices v and w in V, $(v \to w) \in E$ if and only if $(w \to v) \in E$. Suppose that we are given a bidirected graph $G = (V, E)$ in which every arc is assigned a nonnegative weight. We consider the problem of finding a minimum feedback arc set of G such that the total sum of arc weights is the maximum. We call such a problem the maximum-weight minimum feedbak arc set problem. It is applicable to finding an orthogonal drawing of a hierarchical graph. In this article, we first prove that the maximum-weight minimum feedbak arc set problem is NP-hard for bidirected graphs. We then show that a heuristic algorithm for this problem is effective by comparing it experimentally with an exponential-time exact algorithm.