

平面的2辺連結化問題に対する解法の実験的評価

今井 英敏 高藤 大介 田岡 智志 渡邊 敏正

広島大学大学院 工学研究科 情報工学専攻

〒739-8527 東広島市鏡山一丁目 4-1

(電話) 0824-24-7662 (渡邊), -7661(高藤), -7666 (田岡)

(ファクシミリ) 0824-22-7028

(電子メール) {imai, takafuji, taoka, watanabe}@infonets.hiroshima-u.ac.jp

あらまし 平面的2辺連結化問題を次のように定義する:「連結な平面的グラフ $G = (V, E)$ が与えられたとき, $G' = (V, E \cup E')$ が2辺連結平面的グラフとなる最小本数の付加辺集合 E' を見つけよ。」ここで, グラフが2辺連結であるとは, どの1本の辺を取り除いても非連結にならないことである. また, 平面的グラフとは辺を交差せずに平面に描くことができるグラフである. 本稿では, 既存の2-近似解法 $2OPT$ の改良版である2つの2-近似解法 $B2OPT$ と $SETP$, および発見的解法 ETP を提案し, これら4解法の実験的比較により, 提案解法の有用性を示す.

キーワード 平面的グラフ, 2辺連結性, 2-近似解法, 発見的解法

Experimental Evaluation of Algorithms for the Planar 2-Edge-Connectivity Augmentation Problem

Hidetoshi Imai Daisuke Takafuji Satoshi Taoka and Toshimasa Watanabe

Graduate School of Engineering, Hiroshima University

1-4-1, Kagamiyama, Higashi-Hiroshima, 739-8527 Japan

Phone : +81-824-24-7662 (Watanabe), -7666 (Taoka) Facsimile : +81-824-22-7028

E-mail : {imai, takafuji, taoka, watanabe}@infonets.hiroshima-u.ac.jp

Abstract The planar 2-connectivity augmentation problem is defined as follows: given a connected planar graph $G = (V, E)$, find an edge set E' of minimum cardinality such that $G' = (V, E \cup E')$ is 2-connected and still planar. In this paper, we propose three algorithms: two 2-approximation algorithms $B2OPT$ and $SETP$, which are modified versions of the 2-approximation algorithm $2OPT$, and a heuristic algorithm ETP . Experimental comparison shows superiority of our proposed algorithms over $2OPT$.

key words planar graphs, 2-connectedness, 2-approximation algorithms, heuristic algorithms

1 はじめに

正の整数 k に対して, k 辺連結化問題 (k ECA) (それぞれ, k 点連結化問題 (k VCA)) を次のように定義する: 「グラフ $G = (V, E)$ が与えられたとき, $G' = (V, E \cup E')$ が k 辺 (k 点) 連結となる最小本数の付加辺集合 E' を求めよ。」但し, グラフが k 辺連結 (k 点連結) であるとは, 任意に $k-1$ 本の辺 ($k-1$ 個の点) を取り除いても非連結にならないことである。なお, k 点連結グラフは $k+1$ 個以上の頂点を含むものとする。さらに, k ECA あるいは k VCA において, G と G' のいずれも平面的グラフとする問題を, それぞれ平面的 k 辺連結化問題 (Pk ECA) あるいは平面的 k 点連結化問題 (Pk VCA) と呼ぶ。なお, 平面的グラフとは辺を交差せずに平面に描くことができるグラフである。実際にそのように描画したものを平面描画という。

P2VCA については [1] で NP 困難性の証明が示されていたが, それが, 必ずしも多項式時間還元性を示したことにはなっていないことを簡単に指摘しておく。その修正版の存在は今のところ著者らはつかんでいない。一方, P2ECA については NP 困難かどうか, 現在までのところ証明されていないようである。

与えられた $E_0 \subseteq V \times V$ から $E' \subseteq E_0$ なる E' を選ぶ, という制限を 2ECA (それぞれ 2VCA) に加えた形の決定問題 D2ECA (D2VCA) を定義する: 「グラフ $G_0 = (V, E_0)$, および G_0 の部分グラフで, $2B$ 個の次数 1 の点 (葉と呼ぶ) を持つ木 $T = (V, F)$ が与えられたとき, $(V, F + E')$ が 2 辺連結グラフ (2 点連結グラフ) で $|E'| \leq B$ なる $E' \subseteq E_0 - F$ が存在するか否か。」[5] で示されている NP 完全性の証明により, これらの問題も NP 完全性であることになる。 G_0 を極大平面的グラフと限定した D2ECA (それぞれ D2VCA) が NP 完全かどうかは示されていないが, 問題の難しさの本質は非平面的グラフも許す場合と変わっていないものと思われる。これは, P2ECA, P2VCA の難しさは G_0 が非平面的か, 平面的かで差が出る可能性が少ないことを予想させる。厳密解を求める多項式時間解法の存在を否定することはできないが, 種々の考察や実験に基づけば, P2ECA あるいは P2VCA に関する判定問題が NP 完全であるという可能性も十分に考えられる。しかしながら, いずれの研究方向でも, グラフの構造と解の性質についての考察が必要であり, その基礎となるという意味で, P2ECA および P2VCA に対する近似解法を提案する意義は十分あるものと思われる。

次に, P2ECA および P2VCA に対する近似解法

についての既存結果を述べておく。[1] では G を連結グラフとした場合の P2VCA に対して近似比が 2 の近似解法が提案されている。なお, 最小化問題に対する r -近似解法とは, 近似解法で得られた解の大きさが最適解の r 倍以下となることを保証した解法のことである。(そのような保証がない場合を発見的解法と呼んでいる。) [2] では G を連結グラフとした場合の P2VCA に対する $\frac{7}{4}$ -近似解法および $\frac{5}{3}$ -近似解法が提案され, 前者の実験的な性能評価を行っている。また, P2ECA に対しては, G が連結グラフの場合について, [1] で 2-近似解法 $2OPT$ が提案されている。

本稿では, G を連結グラフとした場合の P2ECA に対して, 既存解法 $2OPT$ の改良版である 2 つの 2-近似解法 $B2OPT$ と $SETP$, および発見的解法 ETP を提案し, $2OPT$ とこれらの比較実験に基づく性能評価を行い, 提案解法が優れていることを示す。以降では, 断りがない限り G を連結グラフと仮定する。

2 諸定義

グラフ $G = (V, E)$ は, 有限な空でない点集合 V と辺集合 E とから成る。 V, E をそれぞれ $V(G), E(G)$ と表すこともある。辺に向きが付いていないもの (無向辺) と, 付いているもの (有向辺) があり, 2 つの点 u, v ($u, v \in V$, u と v は同一であってもよい) を結ぶ場合 (u, v) と表す。 u, v をその辺の端点と呼ぶ。無向辺の場合には, (u, v) と (v, u) は同じ辺を表す。有向辺 (u, v) は, u から v への向きであるとする。すべての辺が無向辺 (それぞれ有向辺) のグラフを, 無向グラフ (有向グラフ) という。本稿では, 特に断りがない限り無向辺および無向グラフを, それぞれ辺とグラフと呼ぶ。

無向グラフ G において, 辺 $e = (u, v)$ が G の辺であれば, 点 u と点 v は隣接しているといい, 点 u と辺 e , または点 v と辺 e は接続しているという。また, 2 つの異なる辺 e_1, e_2 が 1 つの端点を共有しているとき, 辺 e_1, e_2 は隣接しているという。2 つの異なる辺 e_1, e_2 がともに点 u, v と接続しているとき, すなわち, $e_1 = (u, v), e_2 = (u, v)$ であるとき, これらの辺を多重辺という。辺 e がただ 1 つの点 u を両端点としているとき, すなわち, $e = (u, u)$ のとき, 辺 e を自己閉路という。自己閉路と多重辺を含まないグラフを単純グラフという。完全グラフとは, その任意の 2 点間に辺が存在する単純グラフである。

グラフ $G' = (V', E')$ が, $V' \subseteq V$ かつ $E' \subseteq E$

であるならば、 G' はグラフ G の部分グラフという。特に $V' = V$ ならば、 G' はグラフ G の生成部分グラフという。

グラフ G において、点と辺の交替列 $P = v_0, e_1, v_1, \dots, v_{n-1}, e_n, v_n$ が存在するとき、 P を点 v_0 から点 v_n への長さ n のウォークという。ここで辺 $e_i (1 \leq i \leq n)$ は、 v_{i-1} と v_i を端点とする辺である。このとき、 P は点 $v_i (0 \leq i \leq n)$, 辺 $e_i (1 \leq i \leq n)$ を通る、あるいはこれらの点 v_i , 辺 e_i は P 上にあるという。点 v_0, v_n を端点といい、 $v_i (1 \leq i \leq n-1)$ を内点と呼ぶ。同じ辺を2度以上通らないウォークはトレイル、同じ点を2度以上通らないウォークはパスと呼ばれる。点 v_0 から v_n へのパスに v_0, v_n を両端点とする新しい辺を追加したものをサイクルという。

グラフ G において、どの2点 u, v に対しても、 u と v を端点とするパスが存在するならば、 G は連結であるという。連結成分とは、極大な連結部分グラフ (または、その点集合) のことである。

グラフ $G = (V, E)$ から点集合 $V' \subseteq V$ (点 $v \in V$) とそれに接続するすべての辺を取り去ることを、点集合 V' (点 v) の除去といい、取り去った後のグラフを $G - V'$ と表す。 G から辺集合 $E' \subseteq E$ (辺 $e \in E$) を両端点を残したまま取り去ることを、辺集合 E' (辺 e) の開放除去といい、取り去った後のグラフを $G - E'$ ($G - v$) と表す。 G から辺 $e \in E$ を開放除去し、 e の両端点を1つの点にまとめることを、辺 e の短絡除去という。 E' が V の2点を結ぶ辺集合で、 $E' \cap E = \emptyset$ であるとき、 G に E' を付加してできるグラフを $G + E'$ と表す。特に、 $E' = \{e\}$ であるときには $G + e$ と表す。点集合 $S \subseteq V$ を1点 s に縮約するとは、 G から点集合 S を除去し新たに点 s を加え、元の G において点 $u \in S$ と点 $v \in V - S$ を結んでいた全ての辺 (u, v) に対して、辺 (s, v) を付加することである。

グラフ G と $G - P$ (それぞれ $G - Q$) が非連結となる $P \subset V$ (それぞれ、 $Q \subseteq E$) の中で、 $|P|$ ($|Q|$) の最小値を G の点連結度 (辺連結度) と呼び、 $vc(G)$ ($ec(G)$) と表す。 k 点連結グラフ (k 辺連結グラフ) とは、 $vc(G) \geq k$ ($ec(G) \geq k$) なるグラフであり、これは任意の2点間に点 (辺) を共有しないパスが k 本以上存在するようなグラフであるともいえる。 k 点連結成分 (k 辺連結成分) とは、点の部分集合で、それに含まれる任意の2点間に内点 (辺) を共有しないパスが k 本以上存在するような極大なものである。

G から $e \in E$ を取り除くとグラフが非連結になる辺 e を bridge と呼び、bridge に接続する頂点を

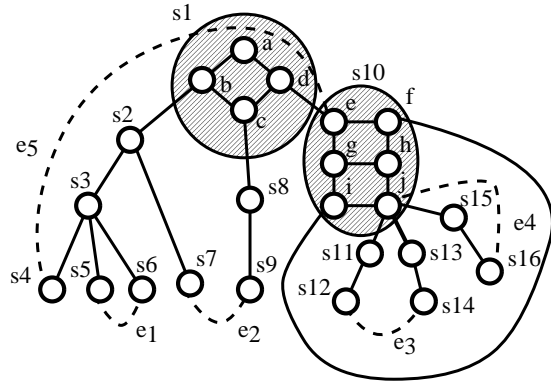


図 1: 入力グラフ G (実線) と P2ECA に対する最適解 (破線)

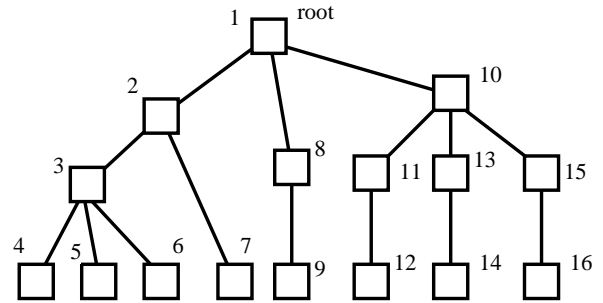


図 2: $S(G)$ と root を開始点とした場合の DFS 番号 (各点の横の数字)

丁度1つ持つような2辺連結成分を leaf-2 成分と呼ぶ。また G から $v \in V$ を取り除くとグラフが非連結になる点 v をカット点と呼ぶ。また、 $G - v$ において、新たにできる各連結成分を B' とすると G の v ブロックは $B' \cup \{v\}$ なる点集合とする。

サイクルが存在しない連結で無向グラフを木という。木において次数が1の点を葉またはリーフと呼ぶ。

グラフ G の各2辺連結成分をそれぞれ1頂点に縮約したグラフを $S(G)$ と表す。 $S(G)$ は木であり、その各辺が G の bridge に対応している。 $S(G)$ の各点 v と G の2辺連結成分 S には1対1の対応があり、 $\epsilon_G(S) = s$, $\epsilon_G^{-1}(s) = S$ とする。さらに、 $v \in S$ において $\epsilon_G(v) = s$ とする。

3 既存解法

まず、[1] での P2VCA の NP-困難性の証明について簡単に言及し、次に $G + E'$ が2辺連結となる最小辺集合 E' を求めるための基本的な考え方を示すために、[4] で提案されている 2ECA に対する手法 BRC

を示す。その後で [1] で提案されている P2ECA に対する 2-近似解法 $2OPT$ を示す。

3.1 P2VCA について

[1] では、NP-完全問題 3-partition から DP2VCA (P2VCA に関する決定問題で以下に定義する) への多項式時間還元性を示すことにより、P2VCA の NP-困難性の証明を与えている。これらの問題の定義は以下の通りである。

3-PARTITION

入力例: $3m$ 個の要素を持つ集合 $A = \{a_1, \dots, a_{3m}\}$.
但し、各 $a_i \in A$ は非負整数の size $s(a_i)$ を持ち、 $B/4 < s(a_i) < B/2$ (B は非負整数) であり、 $\sum_{a_i \in A} s(a_i) = mB$ とする。

質問: A の m 個の部分集合 A_1, A_2, \dots, A_m への分割で、 $\sum_{a \in A_i} s(a) = B$ ($i = 1, \dots, m$) をみたすものが存在するか否か。

DP2VCA

入力例: 平面的グラフ $G = (V, E)$, 非負整数 h .

質問: V の 2 点を結ぶ辺の集合 E' で、 $E' \cap E = \emptyset$ かつ $|E'| \leq h$ であり、しかも $G' = (V, E \cup E')$ が平面的 2 辺連結グラフとなるものが存在するか否か。

[1] での証明においては、3-partition の入力例における各 $a_i \in A$ とそのサイズ $s(a_i)$ に対して、DP2VCA の入力例のグラフ G を構成する際に、点 a とそれに接続する $s(a_i)$ 本の辺を付加する。ここに [1] の操作が、多項式時間還元性を持たない原因がある。一般に、アルゴリズムの計算時間複雑度は入力データサイズをパラメータにした関数で表され、それが多項式となる場合を多項式時間アルゴリズムと呼んでいる。簡単のために、 a_i と $s(a_i)$ をそれぞれ a, s と表すと、 s の入力サイズは $\lceil c \cdot \log_2 s \rceil$ ビットである。ここで c は定数である。しかし、 s 本の辺の付加には少なくとも $c' \cdot s$ 時間かかる。ここで、 c' は定数である。いま $c' \cdot s / \lceil c \cdot \log_2 s \rceil$ なる比を考えると、 $q = \log_2 s$ とおけば、 $c' \cdot s / \lceil c \cdot \log_2 s \rceil \leq (c'/c)(s / \log_2 s) = (c'/c)(2^q / q)$ となる。つまり、入力データサイズの多項式ではない。したがって、[1] で示された 3-partition の入力例から DP2VCA の入力例を構成するアルゴリズムは多項式時間アルゴリズムとはなり得ない。すなわち、2つの問題の解の

等価性は与えているが、多項式時間還元性を示している訳ではない。

3.2 アルゴリズム BRC [4]

2ECA に対する解法 BRC について考える。図 1 に P2ECA の例を示す。 G の leaf 成分の集合を $\mathcal{L}(G) = \{L_i \mid 1 \leq i \leq p\}$ とおく。ただし、 p は leaf 成分数である。また、各 leaf 成分 L_i に対応する $S(G)$ の葉を l_i で表す。図 2 に図 1 の G に対する $S(G)$ を示す。各 leaf 成分 L_i には 1 本のみ辺が接続し、 $G + E'$ が 2 辺連結であることから、 L_i に含まれる点を端点とする辺が E' に含まれている必要がある。このことから、

$$|E'| \geq \lceil \frac{p}{2} \rceil \quad (1)$$

が成り立つ。また、次で示す BRC は、深さ優先探索 (DFS) を利用して、 $G + E'$ が 2 辺連結で、 E' の各辺の両端点のそれぞれが leaf 成分に含まれ、 $|E'| = \lceil \frac{p}{2} \rceil$ である辺集合 E' を求める (図 3 参照)。この際、処理の簡単化のため $S(G)$ を用いる。

アルゴリズム BRC

入力: 連結グラフ $G = (V, E)$

出力: 付加辺集合 E'

1. $S(G) = (V_s, E_s)$ を求める。
2. $S(G)$ の葉でない任意の 1 点を選び (以下 root と呼ぶ)、root を開始点として DFS を行い、各点 $v \in V_s$ に DFS 番号 $\delta(v)$ を付ける。 $S(G)$ の各葉 l_j, l_k ($j < k$) において、 $\delta(l_j) < \delta(l_k)$ となるように、葉集合および $\mathcal{L}(G)$ を並び替える。
3. 次の辺集合 E' を求める。ただし、 n_k は L_k に含まれる適当な 1 つの点を表す。

$$E' = \{(n_i, n_j) \mid 1 \leq i \leq \lceil \frac{p}{2} \rceil, j = i + \lfloor \frac{p}{2} \rfloor\} \quad (2)$$

以上より、次の定理を得る。

定理 3.1 ([4]) BRC は、2ECA に対する $|E'| = \lceil \frac{p}{2} \rceil$ なる最適解 E' を $O(|V| + |E|)$ 時間で求める。

3.3 アルゴリズム 2OPT [1]

まず、P2ECA に対する解法の基本的な考え方を示す。この問題においても、各 leaf 成分に対して少なくとも 1 本の辺を付加しなければならない。しかし、P2ECA の最適解 E' に関しては $G + E'$ が平面的グラフになる必要があるため、 E' の各辺の両端点が leaf 成分に含まれるとは限らない。P2ECA の近

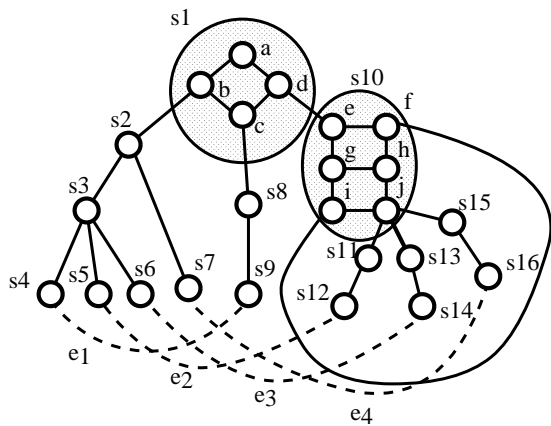


図 3: 図 2 の DFS 番号を用いて BRC で求めた解 (破線)

似解 E' を求める際には、解の大きさを抑えるために、できるだけ 2 つの leaf 成分を繋ぐ辺を求める方が良いが、それが無理な場合には、付加辺 (v, w) の両端点を端点とする G 上のパスが、できるだけ多くの bridge が含むようなものを求める方が良い。

次に示す $2OPT$ は、後者の方針を重視した手法である。なお、より小さな大きさの解を求めるには、もっと前者の方針を重視する必要がある。この点を改良したのが次節以降で示す提案手法である。

アルゴリズム $2OPT$

入力: 連結平面的無向グラフ $G = (V, E)$

出力: 付加辺集合 E'

1. $E' \leftarrow \emptyset, G' \leftarrow G$.
2. $S(G')$ を求め、任意の一点を開始点 (root) として DFS を行い、各点 v に DFS 番号 $\delta(v)$ を付ける。ただし、 $|V_s| > 3$ の場合、root は leaf 成分以外を選ぶ。
3. $\mathcal{P} = \{P_{G'}(s) \in \mathcal{L}(G') \mid s \in V, P_{G'}(s) \neq \emptyset\}$ とおく。ただし、 $P_{G'}(s)$ は G' において s に隣接する leaf 成分 S の族を表す (s が点集合 S に隣接しているとは、 $s \notin S$ であり、 s に隣接する $v \in S$ が存在する場合に言う)。そこで、各 $P_{G'}(s) \in \mathcal{P}$ に対し、 $|P_{G'}(s)| \leq 2$ となるまで次の操作を繰り返す:
 - 3.1. 任意の $L_i, L_j \in P_{G'}(s)$ ($i \neq j$) を選ぶ。 L_i, L_j に含まれる唯一のカット点を、それぞれ v_i, v_j とするとき、 $E' \leftarrow E' \cup \{(v_i, v_j)\}$, $G' \leftarrow G' + \{(v_i, v_j)\}$ とする。
4. 各 $L_i \in \mathcal{L}(G')$ において、次の操作を行う:

- 4.1. $\epsilon_{G'}(L_i) = u_i$ とし、 v_i は L_i に含まれる唯一のカット点とする。 $S(G')$ において u_i の祖先で、 $\delta(a)$ が最小であり、 $G' + \{(v_i, v_a)\}$ が平面的グラフとなる a を求める。但し、 v_a は、 $a = \epsilon_{G'}(v_a)$, G のカット点であり、 v_a ブロック B が $(B - \{v_a\}) \cap \epsilon^{-1}(a) = \emptyset$ かつ $L_i \subseteq B$ を満たす点とする。また、現在の $S(G')$ は、Step 2 の $S(G')$ から leaf のいくつかを削除したものと一致することから、Step 2 で求めた DFS 番号 $\delta(a)$ を用いることができる。 $E' \leftarrow E' \cup \{(v_i, v_a)\}$, $G' \leftarrow G' + \{(v_i, v_a)\}$ とする。

5. グラフ G' が 2 辺連結ならば終了。そうでなければ Step 2 へ戻る。

$2OPT$ の動作と計算時間について補足しておく。 G' は平面的グラフなので $|E \cup E'| \leq 3|V| - 6$ が成り立つ。Step 3.1 において、 $P_{G'}(S)$ の性質から、 $G' + \{(v_i, v_j)\}$ は平面的グラフとなる。Step 4.1 において、 $G' + \{(v_i, v_a)\}$ が平面的グラフでなければ、任意の $v'_a \in S_a$ において、 $G' + \{(v_i, v'_a)\}$ は平面的グラフではないという性質が成り立つ。したがって、 $S(G')$ で、 u_i から root に向かって点を訪問し、それぞれで $G' + \{(v_i, v'_a)\}$ が平面的グラフかどうかをチェックすることで a を求めることができる。また、Step 4.1 の実行直前と比べ直後には、必ず bridge の数が少なくなる。また、 $G' + \{(v_i, v'_a)\}$ が平面的グラフかどうかのチェックは、[3] の手法を用いることで $O(\alpha(|V|, |V|))$ で行うことができ、よって、 $2OPT$ 全体で、Step 4 にかかる計算時間は $O(|V|\alpha(|V|, |V|))$ となる。これら以外の操作は、1 回の DFS の実行と $S(G')$ の更新である。これらは $2OPT$ 全体で、 $O(|V|)$ 時間となる。以上のことから、次の定理を得る。

定理 3.2 ([1]) $2OPT$ は、計算時間が $O(|V|\alpha(|V|, |V|))$ 、メモリ使用量が $O(|V|)$ なる 2-近似解法である。

4 提案解法

4.1 近似アルゴリズム $B2OPT$

まず、 $2OPT$ の改良点について図 1, 図 4 を用いて考える。これらから $2OPT$ では、最適解に比べ両端点が leaf 成分に含まれる付加辺が少ないことが分かる。図 4 において、 e_5, e_6 の leaf 成分に含まれる端点を結ぶ 1 本の辺で置き換えても平面的 2 辺連結グラフとなる。 $2OPT$ の解精度 (近似比) を維持

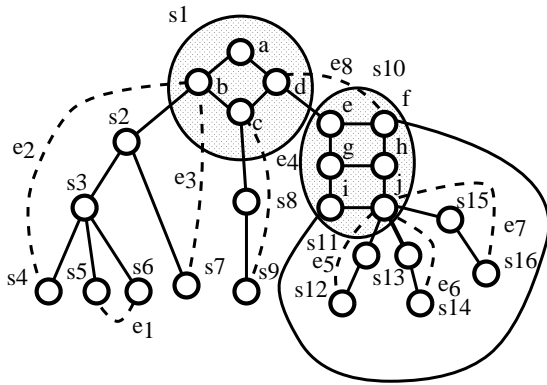


図 4: $2OPT$ で求めた近似解 (破線). ただし, e_1 は Step 3.1 で求めた辺, e_i ($2 \leq i \leq 7$) は Step 4 で求めた辺, e_8 は 2 回目の Step 4 で求めた辺.

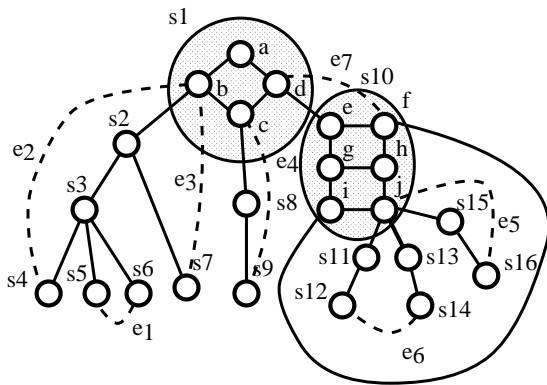


図 5: $B2OPT$ で求めた近似解 (破線). ただし, 図 4 の e_5, e_6 の代わりに Step 4 で e_6 が加えられている.

し, このような置き換え操作を行う手法が $B2OPT$ である. この置き換え操作による計算時間の増加はなく, 実験結果より実際の計算時間は $2OPT$ とほとんど変わらなかった.

アルゴリズム $B2OPT$ は, $2OPT$ が終了した後に, 後処理として次の操作を加えた手法である.

6. 各 $(v_u, v_a), (v'_u, v_a) \in E'$ において, v_u (それぞれ v'_u) を含む G の v_a ブロックを B (B') とするとき $B \cap B' = \{v_a\}$, かつ, $G' + \{(v_u, v'_u)\}$ が平面的グラフとなるならば, $E' \leftarrow E' \cup \{(v_u, v'_u)\} - \{(v_u, v_a), (v'_u, v_a)\}$ とする.

$B2OPT$ の計算時間について補足しておく. Step 6 では, 全体で $O(|V(S(G))|^2)$ 回の平面的グラフのチェックを行う. 以上のことから, 次の定理をえる.

定理 4.1 $B2OPT$ は, 計算時間が $O(|V|^2 \alpha(|V|, |V|))$, メモリ使用量が $O(|V|)$ なる 2-近似解法である.

4.2 発見的アルゴリズム ETP

ETP は, $B2OPT$ の Step 3, 6 (ただし, 前者は $2OPT$ の Step 3 と同じ) を行う前に, leaf 成分間を結ぶ辺で, 平面性を維持できる付加辺 e をできるだけ多く求める処理を加えた手法である. ただし, e を求める際に, すべての 2 つの leaf 成分に辺を加え, 平面的グラフになるチェックを行うため $2OPT$, $B2OPT$ に比べ計算時間がかかってしまう.

アルゴリズム ETP

入力: 連結な平面的無向グラフ $G = (V, E)$

出力: 付加辺集合 E'

1. $E' \leftarrow \emptyset, G' \leftarrow G$.
2. $\mathcal{L}(G) = \{L_i \mid 1 \leq i \leq p\}$ を求める, 任意の一点を root とする. ただし, $|V_s| > 3$ の場合, root は leaf 成分以外を選ぶ. root を開始点として DFS を行い, 各点 $v \in V_s$ に DFS 番号 $\delta(v)$ を付ける. $S(G)$ の各葉 l_j, l_k ($j < k$) において, $\delta(l_j) < \delta(l_k)$ となるように, $\mathcal{L}(G')$ を並び替える.
3. k の値を $0, 1, -1, 2, \dots, i, -i, \dots, \lfloor p/2 \rfloor, -\lfloor p/2 \rfloor$ の順にセットしながら, 各 k に対して次の操作を行う:
 - 3.1. $i = 1, 2, \dots, p$ なる順に以下を行う. $j = i + \lfloor p/2 \rfloor + k \pmod{p}$ なる j に対して, $L_i, L_j \in \mathcal{L}(G')$ ($i \neq j$) を選び, もし, $G' + \{(v_i, v_j)\}$ が平面グラフとなるならば, $E' \leftarrow E' \cup \{(v_i, v_j)\}, G' \leftarrow G' \cup \{(v_i, v_j)\}$ とする. ただし, v_i は L_i に含まれる唯一のカット点を表す.
4. G' に対して, $2OPT$ の Step 4 を行い, E' と G' を更新する.
5. グラフ G' が 2 辺連結ならば終了. そうでなければ Step 2 へ戻る.

Step 3 に $O(|V|^2 \alpha(|V|, |V|))$ 時間かかることと, $2OPT$, $B2OPT$ の結果から, 次の定理を得る.

定理 4.2 ETP は, 計算時間が $O(|V|^2 \alpha(|V|, |V|))$, メモリ使用量が $O(|V|)$ なる発見的解法である.

4.3 近似アルゴリズム $SETP$

$B2OPT$ の解を示した図 5 の中で e_3, e_4 は, 図 7 の e_3 に置き換えることが可能であるが, 前者の 2 辺は G の同じカット点に接続していないので, $B2OPT$ では後者の辺に置き換ええない. また, $B2OPT$ の Step

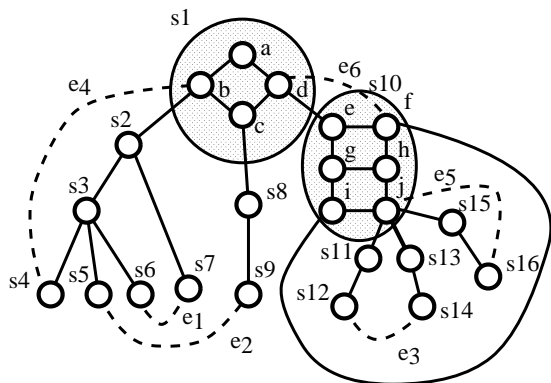


図 6: ETP で求めた近似解 (破線). ただし, e_i ($1 \leq i \leq 3$) は Step 3 で求めた辺.

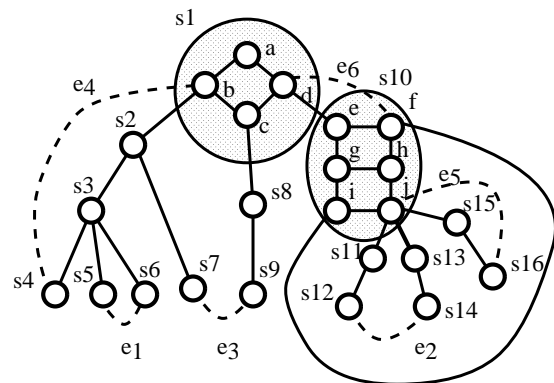


図 7: $SETP$ で求めた近似解 (破線). ただし, e_1, e_2 は Step 3 で求めた辺, e_3 は Step 6 で求めた辺.

6で, $2OPT$ の解である図 4 の e_5, e_6 を, 図 5 の e_6 に置き換えているが, 図 5 の e_6 (図 7 の e_2) は G で求めることが可能である. これにより若干の高速化が可能となる. これらの改善を施した手法が次で示す $SETP$ である.

アルゴリズム $SETP$ は $B2OPT$ ($2OPT$) の Step 3 を次の Step 3 に置き換え, さらに, $B2OPT$ で導入した後処理 Step 6 を, 次の Step 6 に置き換えた手法である.

3. $\mathcal{A} = \{A_{G'}(v) \subseteq \mathcal{L}(G') \mid v \in V, A_{G'}(v) \neq \emptyset\}$ とおく. ただし, $A_{G'}(v)$ を, leaf 成分 T が G の v ブロックに含まれ, それ以外の leaf 成分を含まないような T からなる族とする. なお, T の唯一のカット点を v_t としたとき, $G' + \{(v, v_t)\}$ は平面的グラフである. そして $2OPT$ の Step 3 を \mathcal{P} の代りに \mathcal{A} を用いて操作することで, E' と G' を更新する.
6. G の各 2 辺連結成分 S において, 次で定義する leaf 成分の族 $C_G(S)$ に対し Step 6.1 を行う. ただし, $v_a \in S, (v_u, v_a) \in E'$ が存在し, $\delta(\epsilon(v_u)) > \delta(\epsilon(v_a))$, v_a ブロック B_a に v_u が含まれるような v_u を含む 2 辺連結成分からなる族を $C_G(S) \in \mathcal{C}$ とする.
 - 6.1. 任意の異なる $S_1, S_2 \in C_G(S)$ を選ぶ. S_1 (それぞれ S_2) を定義した上記の v_a, v_u と B_a を v_{a_1}, v_{u_1} と B_{a_1} (v_{a_2}, v_{u_2} と B_{a_2}) とする. $B_{a_1} \cap B_{a_2} \subset \{v_{a_1}, v_{a_2}\}$ であり, $G + \{(v_{u_1}, v_{u_2})\}$ が平面的グラフとなるならば, $E' \leftarrow E' \cup \{(v_{u_1}, v_{u_2})\}$ $\{(v_{u_1}, v_{a_1}), (v_{u_2}, v_{a_2})\}$ とする.

Step 3 において, $A_{G'}(v)$ の性質から, $G' + \{(v_i, v_j)\}$ は平面グラフとなる. 但し, L_i からその先祖に

向かって点を訪問し, それぞれで, $G' + \{(v_i, v_j)\}$ が平面グラフかどうかチェックすることで v_i を求めることができる. このチェックは $SETP$ 全体で辺数以上行うことはないので, $O(|V|\alpha(|V|, |V|))$ となる. Step 6 では, 最悪 ETP の Step 3 と同様に全体ですべての頂点間で, $G' + \{(v_i, v_j)\}$ が平面かどうかチェックしなければならない. $2OPT$, $B2OPT$ と以上より次の定理を得る.

定理 4.3 $SEPT$ は, 計算時間が $O(|V|^2\alpha(|V|, |V|))$, メモリ使用量が $O(|V|)$ なる 2 -近似解法である.

5 計算機実験

5.1 実験環境

CPU: Pentium III 800Mhz, OS: FreeBSD 4.3-RELEASE の計算機上で C 言語により $2OPT$, $B2OPT$, ETP , $SETP$ を実装し実験を行った. ただし, 平面判定のアルゴリズムは $O(|V| + |E|)$ の [6] を用いた.

5.2 入力データ

bridge が少ないグラフでは解法ごとの差がでにくいと考え,

$$|V| = 100, 200, \dots, 1000,$$

$$|E| = a(3|V| - 6) \quad (0.7 \leq a \leq 1),$$

bridge の数が $0.5(3|V| - 6)$ 以上

となる入力グラフ $G = (V, E)$ を, それぞれ 1000 個作成し, $2OPT$, $B2OPT$, ETP , $SETP$ を適用して実験を行った.

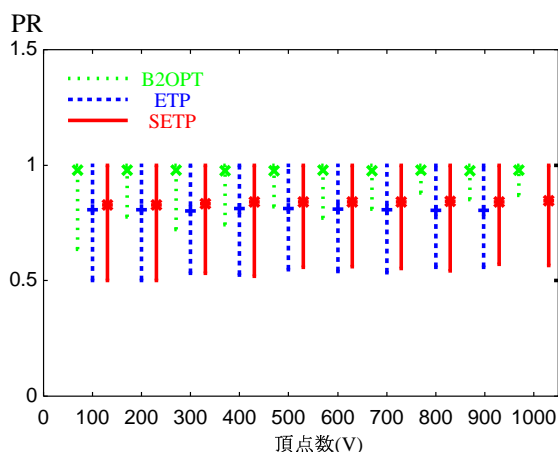


図 8: 各解法と $2OPT$ との PR による比較

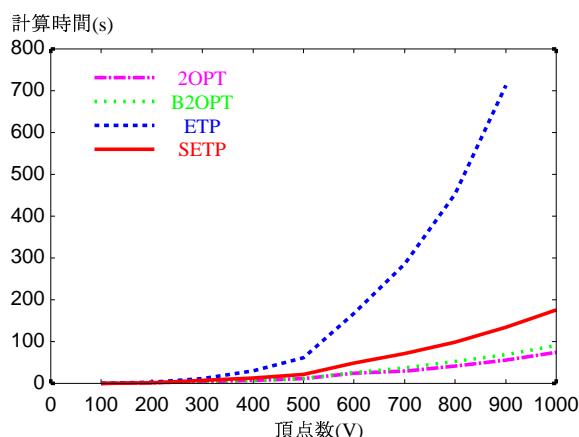


図 9: 各解法の計算時間

5.3 実験結果

各解法を比較するために、既存解法である $2OPT$ を基準とする以下の比 PR を用いる:

$$PR = \frac{B2OPT, ETP \text{ または } SETP \text{ による付加辺数}}{2OPT \text{ での付加辺数}}$$

図 8 において、 PR の範囲を縦の線で、その縦の線上の点は平均値を表している。図 9 では、各解法の平均計算時間を表している。

図 8, 9 より、 $B2OPT$ は、 $2OPT$ とほぼ同じ計算時間で、より小さな大きさの解を求めることが分かる。 ETP と、 $SETP$ は、 $2OPT$ と $B2OPT$ に比べ格段に小さな解を求めることができた。図 9 から ETP は他の解法に比べかなりの計算時間が必要となり、予備実験として行った点数 2000, 3000 の場合では、解を求めることができなかった。一方、他の解法では解を求めることができたが、 $SETP$ は $2OPT$ や $B2OPT$ 対し、2000 点で最大 2 倍、3000 点で最大 10 倍の計算時間を必要とした。以上から、総合的には $SETP$ が最も良く、グラフのサイズが巨大な場合には $B2OPT$ を使用すると良いことになる。

6 まとめと今後の課題

平面的 2 辺連結化問題に対する 2 つの近似解法 $B2OPT$ と $SETP$ 、および発見的解法 ETP を提案し、[1] の近似解法 $2OPT$ と実験的に性能比較をした。これにより、 $SETP$ が最も良い解法であることを示した。今後の課題としては、(1) 入力グラフの辺密度が小さい場合の実験。(2) さらに頂点数が大きい場合の実験。(3) 平面的 2 点連結化問題および平面的 2 辺連結化問題の NP 困難性の証明、などが

あげられる。

参考文献

- [1] G. Kant, "Algorithms for Drawing Planar Graphs", Ph. D thesis, Dept. of Computer Science, Utrecht University, 1993. (<http://www.library.uu.nl/digiarchief/dip/diss/01855269/inhoud.htm>).
- [2] S. Fialko and P. Mutzel, "A New Approximation Algorithm for the Planar Augmentation Problem", Proc. of 9th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, pp. 260-269, 1998.
- [3] J. A. La Poutré, "Alpha-Algorithms for Incremental Planarity Testing(Extended Abstract)", Proceedings of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing, pp. 706-715, 1994.
- [4] P. Eswaran and R. E. Tarjan, "Augmentation Problems", SIAM J. Comput., 5(4), pp. 635-665, 1976.
- [5] G. N. Frederickson and J. Ja'Ja', "Approximation algorithms for several graph augmentation problems", SIAM Journal on Computing, Vol. 10, No. 2, pp. 270-283, 1981.
- [6] K. Mehlhorn and S. Näher, "LEDA: a Library of Efficient Data Types and Algorithms", Proc. 17th Intl. Conf. on Automata, Languages and Programming, pp. 1-5, Springer, 1990.
- [7] K. Onoguchi, N. Chiba and T. Nishizeki, "An Algorithm for Biconnectivity Augmentation of Plane Graphs", Proc. of Workshop 534 held at Inst. Math. Analysis, Kyoto University, pp. 221-233, 1984.
- [8] G. Kant. "Augmenting outerplanar graphs", Journal of Algorithms, 21(1):1-25, July 1996.