描画固定部分グラフを有するグラフにおける全域平面グラフの階層的抽出法

姉ヶ山 伸一郎　高藤 大介　田岡 智志　渡邉 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) 0824-24-7661（直通）, -7662 (渡邉) -7666 (田岡)
(ファクシミリ) 0824-22-7028
(電子メール) {anegayama, takafuji, taoka, watanabe}@infonets.hiroshima-u.ac.jp

あらまし　本稿では描画固定の部分グラフを含むグラフに対する全域平面最大部分グラフ抽出問題の発見的解法として，$O(|V|^2)$ アルゴリズム *plan_divide* を提案する．描画が固定された部分グラフを含むグラフからの平面的グラフの抽出は応用上極めて有用である．*plan_divide* はこのような機能を持ち，かつ既存手法では処理不可能な大規模グラフ $G=(V,E)$ からの抽出を高速に行う．まず，$|E| > max\_edge$ なる $G=(V,E)$ を，辺数が max_edge 以下のいくつかのグラフ G_i に分割した後，各 G_i において描画固定の部分グラフを含む全域平面的部分グラフを抽出し，このことと分割された部分グラフ間を接続する辺集合からの平面辺抽出に基づいて G の全域平面部分グラフを求める．但し，max_edg は既存手法で処理可能な辺数の上限を表わす．さらに，計算機による既存手法との比較実験を行い，*plan_divide* の性能を実験的に比較評価する．

キーワード　平面的グラフ,　全域部分グラフ,　グラフ分割,　PQR 木

Hierarchical Extraction of a Spanning Planar Subgraph under Fixed Embedding of Specified Subgraphs

Shinichiro Anegayama, Daisuke Takafuji, Satoshi Taoka and Toshimasa Watanabe

Graduate School of Engineering, Hiroshima University
1-4-1, Kagamiyama, Higashi-Hiroshima, 739-8527 Japan
Phone : +81-824-24-7661, -7662 (Watanabe), -7666 (Taoka)
Facsimile : +81-824-22-7028
E-mail : {anegayama, takafuji, taoka, watanabe}@infonets.hiroshima-u.ac.jp

Abstract　In this paper, we present an $O(|V|^2)$ heuristic algorithm *plan_divide* for hierarchically extracting a spanning subgraph under fixed embedding of some specified subgraphs. The purpose of *plan_divide* is to find a spanning planar subgraph of a huge given graph $G=(V,E)$ having some subgraphs that requires fixed embedding. Let max_edge be the maximum cardinality of an edge set that existing planarization algorithm can deal with. First, *plan_divide* divides G with $|E| > max\_edge$ into some small graphs $G_i=(V_i,E_i)$ with $|E_i| \leq max\_edge$ for some $i \geq 1$. Then *plan_divide* extracts a spanning planar subgraph of each G_i, and then finds planar edges from those connecting pairs of subgraphs G_i and G_j $(i \neq j)$. Furthermore, experimental results are given to compare *plan_divide* with other planarization algorithms, and to evaluate performance of *plan_divide* experimentally.

key words　planar graphs,　spanning subgraphs ,　partitioning graphs,　PQR-trees

1 はじめに

本稿で扱う最大全域平面部分グラフ抽出問題は次のように定義される:「グラフ $G=(V,E)$ が与えられたとき，$E' \subseteq E$ なる G の全域平面的部分グラフ $G'=(V,E')$ のうち $|E'|$ が最大のものを求めよ.」

$H_i=(V_i,E_i)(i=1,\ldots,k;k\geq 1)$ を G の連結な部分グラフとする．但し，$V_i \cap V_j=\emptyset(i\neq j)$ とする．G から $\bigcup_{i=1}^{k} E_i$ を開放除去したグラフの全域平面的グラフを G_S と表し，G_S に $\bigcup_{i=1}^{k} E_i$ を付加したグラフを G_f と表す.（G_f は非平面的かもしれない.）$H_i(i=1,\ldots,k)$ が与えられたとき，G から G_f を抽出することを，部分グラフ $H_i(i=1,\ldots,k)$ を固定した（G の）全域平面的グラフ抽出ということにする．各 H_i を描画固定部分グラフとよぶ．さらに，G を（たとえ非平面的であっても）平面上に描いたとする．非平面的ならば何本かの辺は交差して描かれる．このとき，各 H_i もこの平面上に描画されているが，ここで，このときの $\bigcup_{i=1}^{k} V_i$ 中の点配置を固定してみる.（したがって E_H の描き方も固定される.）G_S は平面的であるから平面描画が存在するが，G_S の平面描画では常に上記の $\bigcup_{i=1}^{k} V_i$ の点配置を固定するものとする．このような平面描画に $\bigcup_{i=1}^{k} E_i$ を書き加えた描画を，部分グラフ $H_i(i=1,\ldots,k)$ を固定した（G_f の）平面描画とよぶことにする．

ところで，プリント基板や VLSI のレイアウト設計などでは，部品やモジュールの各々をグラフとして表現し，また，接続要求のある端子集合（ネット）の各々をスター形の木やスパニング木として表現することにより，設計対象とする回路をグラフモデル化することが多い．一般に，部品のほとんどが反転配置が禁止されている．また，モジュールの中にはこのようなものもある．一層設計，多層設計いずれでも，1 つの層でのレイアウト設計はグラフモデルの平面的グラフ抽出に帰着されるが，この際には「反転禁止」なる物理的制約を扱うことが必要となる．たとえばプリント基板レイアウト設計などでは，反転禁止部品を右向きの有向サイクルとして表現し，この「右向き」を常に維持する，という方法などがある．この場合，平面的グラフ抽出においても右向きサイクルを反転することなく（つまり，固定した）平面的グラフ抽出が必要とされる．上述の,「いくつかの連結部分グラフを固定した全域平面的グラフ抽出」はこれを一般化して定義したものであり，実用上は極めて大きな意味を持つものと考えられる．

ところで，最大全域平面部分グラフ抽出問題は NP 完全であり [7]，これまで [2, 3, 4, 8, 9, 10, 14, 16] などを始めとする様々な発見的解法が提案されてきた．しかしながら，このうち [10, 14, 15, 17] 以外はグラフ中の指定部分グラフを固定した平面グラフ抽出を扱うことはできず，[10, 14, 15, 17] のみがそのような抽出が可能である．以下では，指定部分グラフを固定した平面グラフ抽出を行える手法を「描画固定可能な手法」と表現する．上述の通り，描画固定可能な平面グラフ抽出法は接地面が指定された部品を含むプリント基板レイアウト設計などを始めとする幅広い応用分野を持っている．さらに，プリント基板レイアウト設計などの分野では，対象回路の大規模化に伴い，大規模回路において高速に平面グラフを抽出する手法も望まれている．

本稿では，[10] の描画固定可能な全域平面部分グラフ抽出法 *plan_pwb* を応用し，描画固定部分グラフを持つ大規模グラフに対する最大全域平面部分グラフ抽出問題を，以下のように階層的に解く手法 *plan_divide* を提案する．なお，対象とするグラフは非連結でも良いが，議論を簡単にするため，一般性を失うことなく G は連結グラフとする．

<u>階層的全域平面部分グラフ抽出法 *plan_divide*</u>

/* 入力 : グラフ $G=(V,E)$ と描画固定部分グラフ族 $\mathcal{K}=\{H_1,\cdots,H_k\}(k\geq 1)$，出力 : $H_1,\cdots,H_k$ を固定した（G の）全域平面的部分グラフ $G'=(V,E')$ */

step 1. $G=(V,E)$ を d 個の部分グラフ $G_i=(V_i,E_i)$ に分割する．但し，各 $i=1,\cdots,d$ と任意の $H_j\in\mathcal{K}$ に対して，$V(H_i)\cap V_i=\emptyset$ または $V(H_j)\subseteq V_i$ とする．さらに，
$E_C \leftarrow \{(u,v)\in E \mid u\in V(H_i), v\in V(H_j)$ なる $H_i,H_j(i\neq j)$ が存在する $\}$,
$E' \leftarrow E-E_C$ とし，$\mathcal{K}$ を
$\mathcal{K}=\mathcal{K}_1\cup\cdots\cup\mathcal{K}_d$ と分割する．但し，
$\mathcal{K}_x=\{H_y\in\mathcal{K}\mid V(H_y)\subseteq V_x\}(x=1,\cdots d)$
である．

step 2. 各 $i=1,\cdots,d$ について，$\mathcal{K}_i$ 中のすべての部分グラフを固定した（G_i の）全域平面的部分グラフ $G_{f(i)}$ を抽出し，G_i の非平面辺集合 $E_{np(i)}$ を求める．$E' \leftarrow E' - \bigcup_{i=1}^{d} E_{np(i)}$.

step 3. $G'=(V,E')$ に付加しても平面性を維持するような辺集合 $E'_C\subseteq E_C$ を求め，
$E' \leftarrow E'\cup E'_C$,
$E_{np} \leftarrow (\bigcup_{i=1}^{d} E_{np(i)})\cup(E_C-E'_C)$
とする．

step 4. $G'=(V,E')$ を出力する．

また，計算機による既存手法との比較実験を行い提

案手法 $plan\_divide$ の性能を実験的に評価する．‘

2 諸定義

グラフ中のある頂点集合において，任意の 2 頂点間に頂点を共有しないパスが 2 本以上存在する極大な頂点集合を 2 点連結成分と呼ぶ．また，グラフ全体が 1 つの 2 点連結成分で構成されたグラフのことを 2 点連結グラフと呼ぶ．

グラフのどの 2 辺も交差せずに平面に描画できるグラフを平面的グラフと呼び，そうでないグラフを非平面的グラフと呼ぶ．平面的グラフ G に対し，どの 2 辺も交差せずに平面に描画されたグラフを G の平面グラフ，または G の平面描画という．グラフ $G=(V,E)$ の部分グラフ $G'=(V',E')$ が $V'=V$ かつ平面的であるとき，G' を G の全域平面的部分グラフという．さらに，任意の $e\in E-E'$ を付加したグラフ $G'=(V,E'\cup\{e\})$ が非平面的ならば，G' を G の極大平面的部分グラフという．

一般に，グラフの平面描画は平面を 1 つ以上の領域に分割し，この領域を面分 (face) と呼ぶ．面分のうち，グラフの外側で無限に広がっているものを無限面分と呼ぶ．また，領域に接するような，頂点と辺の集合からなる部分グラフを，その面分の境界 (contour) という．平面的グラフ G_p の平面描画を G'_p とする．G'_p における面分の集合を $F(G'_p)$ と表す．G'_p の任意の面分 $f\in F(G'_p)$ に対し，面分の境界上に含まれる頂点と辺の重み総和を面分重みと呼び，$w(f)$ と記す．$w(f')=max\{w(f)\mid f\in F(G'_p)\}$ なる G'_p の面分 f' を G'_p の最大重み面分といい，$f_{max}(G'_p)$ と記す．$w(f_{max}(G''_p))=max\{w(f_{max}(G'_p))\mid G'_p$ は G_p の平面描画 $\}$ なる G''_p を G_p の最大重み面分平面描画と呼び，$f_{max}(G''_p)$ を G_p の最大重み面分と呼ぶ．

グラフを $G=(V,E)$ とし，G における $S\subseteq V$ の誘導部分グラフを $G[S]$ と記す．$S\subset V$ に対し，$G[S]$ を G の子グラフ，G を $G[S]$ の親グラフと呼ぶ．また，$S_1\cap S_2=\emptyset$ なる任意の S_1,S_2 に対し，$G[S_i]$ は $G[S_j]$ の兄弟グラフと呼ぶ．但し，$\{i,j\}=\{1,2\}$．V，E をそれぞれ $V(G)$，$E(G)$ と表わすこともある．$e_i=(v_{i-1},v_i)\in E$，$v_i\in V(1\le i\le n)$ なる頂点と辺の順序列 $v_0,e_1,v_1,\cdots,e_n,v_n$ で，点が重複して出現しないものを v_0,v_n 間のパスと呼ぶ．但し，$e_i=(v_{i-1},v_i)\in E$ は v_{i-1} から v_i への有向辺かまたは，v_{i-1} と v_i を結ぶ無向辺である．このとき，パス長を $n-1$ とする．v_0,v_n 間のパスにこれに含まれない新しい辺 (v_n,v_0) を加えたグラフをサイクルと呼ぶ．任意の $S\subseteq V$，$T\subseteq V$ に対し，$K(S,T;G)=\{(u_1,u_2)\in E\mid u_i\in S\,and\,u_j\in T,\{i,j\}=\{1,2\}\}$ とおく．

3 既存結果のまとめ

3.1 全域平面最大部分グラフ抽出法

これまでに多くの発見的解法が提案されている．このうち，代表的なものを表 1，2 に示す．表中の PR は近似比と呼ばれ，$PR=\min\{\frac{C'}{C}\mid C$ は最適解の, C'は各手法の解の辺数 $\}$ である．また，表中の PR 欄の “—” は PR が現在未知であることを示す.（但し，解は全域平面部分グラフとして求まるため 1/3 以上であることは保証されている.）

表 1 は描画固定部分グラフが存在しない場合の手法を示す．グラフ中の全てのサイクルの長さが 3 であるようなグラフを構成することにより全域平面部分グラフを求める $triangulation$[3] が最も解精度が良いことが [6] で実験的に示されている．

表 2 は描画固定可能な手法を示す．表 2 に示す 2 つの手法があり，PQR 木 [12] を用いた $planpwb$[10] が最も精度が高いことが [14] で実験的に示されている．さらに [15, 17] でその改良手法が提案されている．

表 1: 描画固定不可能な全域平面的部分グラフ抽出法

抽出法	計算時間	PR
edge_embedding [2]	$O(\lvert E\rvert\log\lvert V\rvert)$	1 / 3
triangulation (TR)[3]	$O(\lvert V\rvert^3)$	7 / 18
triangulation [3]	$O(\lvert E\rvert^{\frac{3}{2}}\lvert V\rvert\log^6\lvert V\rvert)$	2 / 5
path_embedding (PE) [4]	$O(\lvert V\rvert\lvert E\rvert)$	1 / 3
cycle_packing (CP) [8]	$O(\lvert V\rvert\lvert E\rvert^2)$	—
incremental [9]	$O(\lvert V\rvert+\lvert E\rvert)$	1 / 3
vertex_addition [16]	$O(\lvert V\rvert^2)$	—

表 2: 描画固定可能な全域平面的部分グラフ抽出法

抽出法	計算時間	PR
plan_pwb [10]	$O(\lvert V\rvert^2)$	—
path_addition_algorithm [14]	$O(\lvert V\rvert\lvert E\rvert)$	1 / 3
plan_MNC [15]	—	—
plan_MIS [17]	—	—

3.2 最大重み面分発見手法

1 つの最大重み面分を線形時間で求める手法が [11] で提案され，求めた最大重み面分を無限面分とする平面描画を線形時間で与える手法も示されている．

4 関連手法

4.1 *PQ* 木[1]

$G = (V, E)$ を 2 点連結グラフとするとき，st-numbering（たとえば [5] 参照）は以下の (1)，(2) をみたす全単射写像 $r : V \to \{1, \cdots, |V|\}$ である．($r(v) (v \in V)$ を st-number と呼ぶ.)

(1) $(s、t) \in E, r(s) = 1$，かつ $r(t) = |V|$.

(2) 任意の $v \in V - \{s, t\}$ に対し，$r(v) > r(v')$ なる隣接頂点 $v' \in V$ と，$r(v) < r(v'')$ なる隣接頂点 $v'' \in V$ が存在する.

PQ 木とは P 点，Q 点，葉からなる根付き木であり，PQ 木の各点には以下のような変形操作が可能である.

- P 点の子の順序を任意に置換する.
- Q 点の子の順序を反転する.

PQ 木 T_i は与えられた 2 点連結グラフ $G = (V, E)$ の部分グラフ $G_i = (V_i, E_i)$ に対応している．但し，$V_i = \{v \in V \mid 1 \leq r(v) \leq i\}$，$E_i = \{(u, v) \in E \mid 1 \leq r(u)$，$r(v) \leq i\}$ である．予め用意されている可能な変形操作を適用して，PQ 木 $T_i (1 \leq i \leq n-1)$ を，頂点 $(i+1)$ に対応する葉が連続した PQ 木 T_i^R に変形することを縮約 (reduction) という．この縮約は，対応する部分グラフ $G'_i = (V'_i, E'_i)$ に，辺の交差を生じることなく，頂点 $(i+1)$ とその接続辺が付加 (vertex addition) 可能なことを保証する．

PQ 木を用いた平面判定法とは，st-number をつけた 2 点連結グラフに対して，st-number の小さい頂点から順に縮約を可能な限り繰り返す方法である．つまり，必要ならばすでに平面に埋め込んだグラフの一部を可能な限り書き直しながら順次グラフ全体まで平面描画を拡大していく手法である．もし，ある頂点に対し，縮約ができないならば，そのグラフは非平面と判定される.

4.2 *PQR* 木[12]

グラフ中に描画固定部分グラフが存在する場合にも，平面判定が行えるよう PQ 木を拡張したものが PQR 木である.

PQR 木 [12] には，PQ 木 [1] での葉，P 点，Q 点に加えて，子の順序を変更できない R 点という概念が新たに追加されている．PQR 木では描画固定部分グラフを R 点で表現し，R 点の子の順序を一切変更しないことで対応するグラフの描画固定を扱うことができる．このため，PQR 木は一部に反転不可能な部分グラフ（したがって，描画固定部分グラフ）を含むグラフについても，正確に平面判定を実行できる.

4.3 全域平面部分グラフ抽出法 $plan\_pwb$

PQR 木を用いた全域平面部分グラフ抽出法 $plan\_pwb$ [10] のアルゴリズムについて説明する．$plan\_pwb$ は反転不可部分を表わす右向き有向サイクル含んだグラフ $G = (V, E)$ が与えられたとき，すべての右向き有向サイクルを維持したまま全域平面部分グラフ $G' = (V, E')$ を求めるものである．$plan\_pwb$ では，まず G の各 2 点連結成分 $S \subseteq V$ に対し以下の手順を行う．これより $G[S]$ の全域平面部分グラフ $G'[S] = (S, E'_S)$ を求めることができる．よって，G のすべての 2 点連結成分 S についての E'_S の和集合 E' を求めて，G の全域平面部分グラフ G' を得る．なお，$plan\_pwb$ では平面判定法に PQR 木を用いる.

(1.1) $G[S] = (S, E_S)$，$n \leftarrow |S|$ とする.

(1.2) 各頂点 $v \in S$ に対し，st-number $r(v)$ を計算.

(1.3) $V_0 \leftarrow \emptyset$，$E_0 \leftarrow \emptyset$，$H_0 \leftarrow (V_0, E_0)$ とする.

(1.4) 任意の $v \in V$ に対し，$r(v)$ の小さい順に以下を行う.
1. $V_i \leftarrow V_{i-1} \cup \{v\}$,
   $E_i \leftarrow E_{i-1} \cup \{e = (v, v') \in E_S \mid v' \in V_{i-1}\}$.
2. $H_i = (V_i, E_i)$ が非平面グラフならば，$H'_i = (V_i, E_i - E_{del})$ が平面的グラフとなる辺数最小の辺集合 $E_{del} \subseteq E_i$ を求め，$E_i \leftarrow E_i - E_{del}$ とする.
3. $i \leftarrow i + 1$.

5 提案手法 $plan\_divide$

5.1 有向サイクルへの置き換えと $plan\_divide$ の流れ

1 節の「はじめに」で示した $plan\_divide$ の流れに従って説明する．そこで述べた通り，$G = (V, E)$ を d 個の部分グラフ $G_i = (V_i, E_i) (i = 1, \ldots, d)$ に分割する．但し，$|E| \leq max\_edge$ ならば $d = 1$ であり，$|E| > max\_edge$ ならば $|E_i| \leq max\_edge$ である．また各 $V(H_j)$ は分割されるこのなくいずれかの G_i に含まれるものとする．G_i を G の子グラフ，G を G_i の親グラフとよぶこともある．この結果，$\mathcal{K}$ は $\mathcal{K}_1 \cup \cdots \mathcal{K}_d$ と分割される．$\mathcal{K}_i$ は G_i に含まれる描画固定部分グラフ族である.

ここで，$plan\_pwb$ を適用するために，以下に (i), (ii) により各 $H_j(j=1,\ldots,k)$ を右向き有向サイクル C_j に置き換える．

(i) $V-V(H_j)$ の点に隣接する $V(H_j)$ の点の集合を N_j とおく：

$N_j=\{v\in V(H_j)|(u,v)\in E$ なる $u\in V-V(H_j)$ が存在する $\}$

(ii) G から $E(H_j)$ のすべての辺を開放除去し，$V(H_j)-N_j$ のすべての点を除去し，かつ N_j の点を適当な順に有向辺で結んで長さ $|N_j|$ の右向き有向サイクル C_j を構成する．

以上の結果，各 H_j は右向き有向サイクル C_j に変形され，H_j の描画固定は C_j を右向きサイクルとして維持することの置き換えられる．これにより，$plan\_pwb$ が適用できる．以下では，任意の $H_j\in\mathcal{K}$ は右向き有向サイクルとし，C_j と表すことにする．C_j を反転不可部分とよぶこともある．各 $i=1,\ldots,d$ について，$\mathcal{K}_i$ 中のすべての有向サイクルを固定した（G_i の）全域平面的部分グラフ $G_{f(i)}$ の抽出は $plan\_pwb$ により行うが，異なる G_i,G_j を結ぶ辺の集合 E_C からの平面辺抽出には，本稿では以下の手法を採用する．各 G_i から抽出された平面的グラフ $G_{f(i)}$ の平面描画を求め，その無限面分の境界を右向き有向グラフ C_i と表す．このような有向グラフの族に E_C を付加して構成されるグラフを $G_{red}=(V_{red},E_{red})$ とおく．もし $|E_{red}|\le max\_edge$ ならば G_{red} に対して $plan\_pwb$ を適用して E_C から平面辺を抽出する．一方，もし $|E_{red}|>max\_edge$ ならば，$G\leftarrow G_{red}$ とおいて再帰的にここまでの操作を反復する．これを有向サイクルへの再帰的置き換えとよぶ.（この反復回数の上限を loop と表しておく.）適当な反復後には，必ず $|E_{red}|\le max\_edge$ なる G_{red} に到達し，E_C からの平面辺抽出が完了する．任意の $v\in V$ に対し，$v\in V(C)$ なる有向サイクル $C\in\mathcal{K}$ を $C(v)$ と記す．任意の $v\in V$ に対し，$v\in V(C_i)$ なる $C_i\in\mathcal{K}$ が存在するならば，$C_{size}(v)\leftarrow|V(C_i)|-1$，そうでないならば $C_{size}(v)\leftarrow 0$ とする．

step 1. $\mathcal{C}_{\prime}\leftarrow\emptyset$，$E'\leftarrow\emptyset$ とする．

$|E|>max\_edge$ である限り以下を反復する．

関数 *Find_vertex_set*(G) を行うことにより互いに素な頂点集合の族 $N(G)=\{S\subseteq V\mid |E_S|\le max\_edge,\ G[S]=(S,E_S)\}$ を求める．但し，任意の $C'\in\mathcal{C}$ に対し，($V(C')\subseteq S$ なる $S\in N(G)$ が存在する）かまたは（$V(C')\cap S=\emptyset$）である．もし，$N(G)=\emptyset$ ならば step 2. へ行く．

$S\neq\emptyset$ なる各 $S\in N(G)$ に対し，以下 (1)-(5) を行う．

(1) $plan\_pwb(G[S])$ を行うことにより，$G[S]=(S,E_S)$ の全域平面部分グラフ $G[S]'=(S,E'_S)$ を求める．

(2) 各 $v\in S$ に対し，$W(v)=|K(\{v\},V-S;G)|$ とする．

(3) $G[S]'$ の最大重み面分 f_{max} を求め，これを無限面分として持つ平面描画 $G[S]''$ を求める．

(4) $G[S]''$ における無限面分 f_{max} の境界に含まれる頂点集合を $V(f_{max})$ とする．

$i\leftarrow 1$ とする．

任意に $v\in V(f_{max})$ を選び，これを v_0 とする．各 $v\in V(f_{max})$ に対し，v_0 から時計回りの出現順に $W(v)>0$ ならば $vst(v)\leftarrow i$ かつ $i\leftarrow i+1$ とする．

$V'_f=\{v\in V(f_{max})\mid W(v)>0\}$，$n=|V'_f|$ とすると，$C'_S=\{\langle v_1,v_2\rangle\mid vst(v_1)+1=vst(v_2)=j,\ 2\le j\le n\}\cup\{\langle u_1,u_2\rangle\mid vst(u_1)=n,vst(u2)=1\}$．但し $\langle v_i,v_j\rangle$ は頂点 v_i から v_j への有向辺を示す.

(5) $V\leftarrow V\cup V'_f-S$，$E\leftarrow(E-E_S)\cup K(V'_f,V-S;G)\cup C'_S$，$\mathcal{K}\leftarrow\mathcal{K}\cup\{C'_S\}$，$\mathcal{K}_{\mathcal{V}}\leftarrow\mathcal{K}_{\mathcal{V}}\cup\{C'_S\}$.

step 2. $|E|\le max\_edge$ ならば $plan\_pwb(G)$ を行うことにより，G の全域平面部分グラフ $G'=(V,E'')$ を求める．

step 3. $E'\leftarrow E'\cup E''-\bigcup_{C'\in\mathcal{K}_{\mathcal{V}}}C'$.

5.2 *Find_vertex_set*

関数 *Find_vertex_set*(G) はグラフ $G=(V,E)$ と有向サイクル集合 $\mathcal{C}$ が与えられたとき，以下の条件を満たす互いに素な頂点集合 $N(G)$ を求めるものである．

各 $S'\in N(G)$ に対して $G[S]=(S,E_S)$ とするとき，

(i) $|E_S|\le max\_edge$

(ii) 任意の $C'\in\mathcal{K}$ と $S\in N(G)$ に対し,（$V(C')\subseteq S$）かまたは,（$V(C')\cap S=\emptyset$）である．

step 1. $N(G)\leftarrow\emptyset$. $i\leftarrow 1$，各 $v\in V$ に対し，$check(v)\leftarrow NEW$ とする．

step 2. $check(v)=NEW$ なる点 $v\in V$ が存在する限り，以下を行う．

(2.1) $V_i\leftarrow\emptyset$，$E_i\leftarrow\emptyset$，キュー $V_Q\leftarrow\emptyset$.

(2.2) $check(v) = NEW$ なる任意の点 $v \in V$ を選び now_v とする．now_v を開始点として幅優先探索を開始．

$check(now\_v) \leftarrow SEARCHING$.

$V_i \leftarrow \{now\_v\}$.

$cv \leftarrow cv + C_{size}(now\_v)$.

$C_{size}(now\_v) \neq 0$ ならば，各 $v' \in C(now\_v)$ に対し，$C_{size}(v') \leftarrow 0$.

(2.3) $now\_v \in V$ に対し，以下を行う．

任意の $e = (now\_v, w) \in E$ に対し以下 (a) または (b) の条件と一致すればそれぞれの処理を実行．但し，$Worst(E_i)$ は $V_i \leftarrow V_i \cup \{w\}$ とした場合に $|E_i|$ が取り得る最大の値を示し，以下の計算式によって算出される．

$Worst(E_i) = |E_i| + (qv + cv + 2) * (qv + cv + 1)/2$.

(a) $check(w) = NEW$ かつ $Worst(E_i) \leq max\_edge$ ならば，
$check(w) \leftarrow SEARCHING$.
キュー V_Q に頂点 w を挿入.
$qv \leftarrow qv + 1$.
$V_i \leftarrow V_i \cup \{w\}$.
$E_i \leftarrow E_i \cup \{e\}$.
$C_{size}(w) \neq 0$ ならば，
$cv \leftarrow cv + C_{size}(w)$.
各 $v' \in C(w)$ に対し，$C_{size}(v') \leftarrow 0$.
$w \in V(C)$ なる有向サイクル $C \in \{\mathcal{K}\}$ が存在するならば，$cv \leftarrow cv - 1$.

(b) $check(w) = SEARCHING$ ならば $E_i \leftarrow E_i \cup \{e\}$.

(2.4) キューが空ならば (2.5) へ．空でなければ，キューから頂点を 1 つ取り出し，この頂点を now_v，$qv \leftarrow qv - 1$ として (2.3) へ.

(2.5) $|V_i| = 1$ ならば以下 (a)，(b) をともに行う．

$|V_i| \leq 2$ ならば以下 (a)，(b)，(c) をともに行う．

(a) 各 $v' \in V_i$ に対し $v' \in C'$ なる $C' \in \mathcal{C}$ が存在するならば，$V_i \leftarrow V_i \cup V(C')$ とする．

(b) 各 $v' \in V_i$ に対し，$check(v') \leftarrow OLD$ とする．

(c) $N(G) \leftarrow N(G) \cup V_i$．$i \leftarrow i + 1$.

6 計算機実験

まず予備実験として，既存手法の比較実験を行った．乱数を利用して作成した 2 点連結なグラフ $G = (V, E)$ に対し，各全域平面部分グラフ抽出法を適用し，除去辺数 $|E_{np}| = |E - E'|$，計算時間 CPU(単位：秒) について比較し，表 4 の結果を得た．また手法 GR は $E' \leftarrow \emptyset$ とし，$G = (V, E)$ 各 $e \in E$ について，$E' \leftarrow E' + \{e\}$ とした $G = (V, E')$ が非平面グラフであれば $E' \leftarrow E' - \{e\}$ とすることを繰り返した手法である．なお，この手法の平面判定には [13] のプログラムを利用している．

次に，描画固定可能な手法であり，予備実験においても他の手法に比べて高速な抽出を行っていた $plan\_pwb$[10] を比較対象とし，本提案解法 $plan\_divide$ との比較実験を行った．乱数を利用して作成した 2 点連結なグラフ $G = (V, E)$ に対して，両全域平面部分グラフ抽出法を適用し，除去辺数 $|E_{np}| = |E - E'|$，計算時間 $CPU(s)$ について比較した．

手法 div_k は，max_edge を $max\_edge = |E|/k$ と定めた場合の $plan\_divide$ を表す.

$plan\_divide$ の step1. 終了時に $G = (V.E)$ について $|E| > max\_edge$ である場合，G 中の仮想有向サイクルを通常のサイクルに置き換え，各変数を初期化すると，G に step1. の分割処理を繰り返し適用することが可能である．この繰り返し回数の上限値を $loop$ であらわす.

表 4，表 5，表 6，にそれぞれ有向サイクルへの再帰的置き換えとの上限値 $loop = 0, 1, 2$ とした場合の実験結果を示す．各値は，それぞれ 10 データ毎の平均値である．

今回の実験データでは，$loop$ の値，入力非平面グラフのサイズに関わらずほぼ全ての場合において $|E_{np}|$，$CPU(s)$ いずれについても div_3 が $plan\_divide$ の中で最も優れていることが示された．除去辺数については div_3 は最悪の場合でも $plan\_pwb$ の除去辺数の 102% 未満であり，グラフの辺数が増加しても $plan\_pwb$ と除去辺数の差が開いていく傾向は現在のところ見られない．計算時間については $plan\_pwb$ よりも高速で，またこの傾向はグラフの頂点数が増大するにつれ顕著になっており，頂点数 1000 のグラフでは $plan\_pwb$ の半分以下となっている．

除去辺数が $plan\_pwb$ と比較して著しく増大しないのは，step(2.6) で子グラフを最大重み面分を無限面分として再描画し，グラフ分割により除去辺となる辺の数をなるべく少なくしているためだと思われる．また，分割処理や再描画などの処理を追加したにも関わらず計算時間が減少したのは，分割により同時に扱うグラフサイズが減少し，$O(|V|^2)$ である $plan\_pwb$ の計算時間が短縮されたためと考えられる．

今回の実験データにおいては div_3 は *plan_pwb* とほぼ同精度の除去辺数を維持したまま，メモリオーバーを起こすことなく，より高速に平面グラフ抽出を実行できており，大規模グラフでの平面グラフ抽出に有効なのではないかと考える．

今後は大規模グラフに対しての実験データを充実させ，本稿での考察の正当性を確認していく予定である．

7 まとめ

本稿では $O(|V|^2)$ で大規模グラフの平面グラフ抽出を階層的に行う手法 *plan_divide* を提案し，実験的にその性能を比較評価した．

今後の課題としては

- より精度が向上するようなグラフ分割手法の提案．
- より大規模なグラフでの実験など各種実験の充実．

等があげられる．

参考文献

[1] K. S. Booth and G. S. Lueker "Testing for the Consecutive Ones Property, Interval Graphs, and Graph Planarity Using PQ-Tree Algorithms", Journal of Computer and System Sciences 13, pp. 335-379, 1976.

[2] J. Cai, X. Han and R. E. Tarjan, "An $O(m \log n)$ - Time Algorithm for Maximal Planar Subgraph Problem", SIAM Journal on Computing, Vol. 22, pp. 1142–1162, 1993.

[3] C. Călinescu, C. G. Fernandes, U. Finkler and H. Karloff, "A Better Algorithm for Finding Planar Subgraphs", Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 16–25, 1996.

[4] T. Chiba, I. Nishioka and I. Shirakawa, "An Algorythm of Maximal Planarization of Graphs", In Proc. IEEE Symp. on Circuits & Sys. , pp. 649–652, 1979.

[5] S. Even and R. E. Tarjan, "Computing an st-Numbering", Theoretical Computer Science 2, pp. 339–344, 1976.

[6] 藤原 裕久, "辺付加順序に基づく全域平面部分グラフ抽出法の高精度化と高速化", 平成１２年度 広島大学 工学部 第二類 (電気系) 卒業論文, Mar. 2001.

[7] M. R. Garey and D. S. Johnson, "Computers and Intractibility", Freeman & Co. , San Francisco, 1979.

[8] O. Goldschmidt and A. Takvorian, "An Efficient Graph Planarization Two-Phase Huristic", Network, vol. 24, pp. 69–73, 1994.

[9] H. Djidjev, "A Linear Algorithm for the Maximal Planar Subgraph Problem", in "Algorithms and Data Structures" (Proc. 4th International Workshop), Lecture Notes in Computer Science, Vol. 955, pp. 369–380, 1995.

[10] 岩元 圭一郎, "平面部分グラフ抽出，矩形双対化及び端子表現グラフモデル化に基づいたアナログ回路用プリント基板設計支援システム PRIDE", 広島大学大学院工学研究科修士論文, February 1992.

[11] 小谷 健, 増田 澄男, 柏原 敏伸, "外窓上の頂点及び辺の重みの和を最大にするような，２連結平面グラフの描写アルゴリズム", 電子情報通信学会論文誌 Vol. J 74-A, No. 7, pp. 1041–1052, July 1991.

[12] 増田 澄男, 柏原 敏伸, 藤澤 俊男, "部品の反転を許さない一層平面判定問題について", 電子通信学会論文誌 Vol. J 66–A, No. 3, pp. 235–242, March 1983.

[13] K. Mehlhorn and S. Näher, LEDA: a Library of Efficient Date Types and Algorithms, Proc. 17th Intl. Conf. on Automata, Languages and Programming, pp. 1–5, Springer, 1990.

[14] 水口 幸則, "部品を指定位置に配置するプリント基板設計の一手法", 平成８年度 広島大学 工学部 第二類 (電気系) 卒業論文, March 1993.

[15] K. Mizuno, T. Kobayashi and T. Watanabe, "Extracting Nonplanar Connections in a Terminal–Vertex Graph", *Proceedings of 1999 IEEE International Symposium on Circuits and Systems*, pp. VI-121–VI124, 1999.

[16] T. Ozawa and H. Takahashi, "A Graph Planarization Algorithm and its Application to Random Graphs", In Graph Theory and Algorithms, Lecture Notes in Computer Science, Vol. 108, pp. 95–107, 1981.

[17] T. Yamaoki, S. Taoka and T. Watanabe, "Extracting a Planar Spanning Subgraph of a Terminal–Vertex Graph by Solving the Independent Set Problem", *Proceedings of 2001 IEEE International Symposium on Circuits and Systems*, pp. V-153–V-156, 2001.

表 3: 既存手法の比較実験

$\|V\|$	$\|E\|$	*plan_pwb*		PE[4]		TR[3]		CP[8]		GR	
		E_{np}	CPU	E_{np}	CPU	E_{np}	CPU	E_{np}	CPU	E_{np}	CPU
100	294	142	0.210	153	0.242	143	0.258	127	0.508	143	0.398
100	588	418	0.456	456	0.898	404	0.719	399	2.023	404	0.852
100	882	697	1.159	742	1.648	693	1.148	691	6.648	693	1.398
500	1494	866	7.895	950	353.1	845	14.58	812	27.71	845	23.60
500	2988	2279	21.41	2441	896.5	2233	38.90	2175	170.8	2233	48.67
500	4482	3728	36.05	3919	1466	3681	799.7	3595	564.4	3681	77.95
1000	2994	1742	16.48	1900	4662	1670	104.0	1635	408.6	1930	171.4
1000	5988	4572	42.78	4897	15504	4479	281.2	4351	291.9	4482	280.6
1000	8982	7470	72.65	7888	20163	7365	1627	–	–	7355	565.5

表 4: $loop = 0$ とした実験結果

$\|V\|$	$\|E\|$	*plan_pwb*		div_2		div_3		div_4		div_5	
		E_{np}	CPU	E_{np}	CPU	E_{np}	CPU	E_{np}	CPU	E_{np}	CPU
100	300	153	0.134	159	0.690	159	0.876	162	1.009	163	1.077
100	400	241	0.284	240	0.558	247	0.598	248	0.841	249	0.864
100	500	331	0.442	332	0.570	330	0.577	335	0.798	336	0.727
100	600	421	0.577	419	0.520	420	0.427	425	0.827	427	0.736
500	1500	912	7.156	936	6.789	920	5.381	958	6.864	946	5.897
500	2000	1376	10.735	1403	7.732	1393	5.618	1418	7.105	1418	5.855
500	2500	1855	14.221	1871	8.830	1871	6.177	1890	7.541	1886	6.126
500	3000	2327	17.808	2346	9.826	2352	6.677	2360	8.831	2358	6.604
1000	3000	1886	27.788	1929	19.469	1901	13.598	1965	16.183	1937	12.467
1000	4000	2843	41.613	2881	23.819	2866	15.956	2921	18.761	2904	14.516
1000	5000	3802	55.502	3817	29.185	3801	19.824	3877	21.638	3857	16.360
1000	6000	4758	70.635	4801	38.566	4781	23.188	4824	25.044	4819	18.278

表 5: $loop = 1$ とした実験結果

$\|V\|$	$\|E\|$	*plan_pwb*		div_2		div_3		div_4		div_5	
		E_{np}	CPU	E_{np}	CPU	E_{np}	CPU	E_{np}	CPU	E_{np}	CPU
100	300	156	0.141	163	0.739	158	0.928	162	1.070	164	1.104
100	400	243	0.300	246	0.688	245	0.526	249	0.935	247	0.938
100	500	329	0.458	328	0.552	326	0.507	335	0.813	336	0.828
100	600	420	0.597	415	0.575	415	0.485	426	0.823	423	0.743
500	1500	908	6.602	933	5.604	926	4.443	950	5.643	936	4.807
500	2000	1381	9.814	1402	6.506	1394	4.474	1409	5.918	1416	4.702
500	2500	1852	13.136	1871	7.502	1869	5.033	1886	6.223	1885	5.006
500	3000	2330	16.404	2350	8.780	2343	5.863	2360	7.066	2358	5.199
1000	3000	1880	28.033	1921	19.723	1891	13.027	1960	16.009	1937	15.366
1000	4000	2842	40.538	2869	23.264	2859	15.677	2914	22.777	2905	18.052
1000	5000	3801	53.905	3830	28.467	3821	19.357	3871	26.152	3865	19.524
1000	6000	4762	66.237	4781	33.710	4797	23.148	4819	28.855	4814	21.834

表 6: $loop = 2$ とした実験結果

$\|V\|$	$\|E\|$	*plan_pwb*		div_2		div_3		div_4		div_5	
		E_{np}	CPU	E_{np}	CPU	E_{np}	CPU	E_{np}	CPU	E_{np}	CPU
100	300	156	0.120	160	0.523	159	0.663	165	0.841	166	0.810
100	400	240	0.252	242	0.427	246	0.568	250	0.774	252	0.721
100	500	326	0.397	330	0.419	333	0.491	332	0.625	334	0.640
100	600	419	0.533	418	0.473	410	0.343	421	0.603	422	0.563
500	1500	912	6.587	931	5.949	927	4.118	952	5.555	943	4.398
500	2000	1379	9.398	1397	6.571	1394	4.483	1416	5.883	4.997	1411
500	2500	1849	12.867	1874	7.580	1863	5.500	1882	6.018	1884	4.915
500	3000	2323	15.855	2344	8.645	2327	6.061	2358	6.797	2358	5.317
1000	3000	1881	29.910	1929	22.856	1908	15.511	1961	16.001	1930	12.505
1000	4000	2845	43.936	2872	27.564	2845	18.562	2918	19.038	2905	14.874
1000	5000	3805	62.218	3834	34.871	3828	22.001	3874	22.102	3857	16.873
1000	6000	4765	75.102	4794	39.814	4799	26.149	4822	24.854	4822	18.902