階層グラフの描画アルゴリズムに関する研究

荒木，徹也

(Degree)
博士（工学）
(Date of Degree)
2013-09-25
(Date of Publication)
2015-09-25
(Resource Type)
doctoral thesis
(Report Number)
甲第5931号
(URL)
https://hdl.handle.net/20.500.14094/D1005931

博　士　論　文

階層グラフの描画アルゴリズムに関する研究

平成 25 年 7 月

神戸大学大学院工学研究科

荒 木 徹 也

内容梗概

階層グラフは，頂点集合が階層と呼ばれる部分集合に分割されており，各辺が異なる階層に属す頂点を結ぶというグラフである．階層グラフの描画のことを階層描画と呼ぶ．階層描画は，作業工程図，科目間関係図，有向グラフなどの表現に広く用いられている．階層描画を求める方法について，今までに多くの研究が行われてきている．それらの研究の多くはグラフがより見やすくなるように，描画中の辺の交差数を減少させる方法や，頂点の配置位置を決定する方法についてのものである．

階層グラフにおいて，連続しない階層上の頂点を結ぶ辺を長辺と呼ぶ．階層グラフの代表的な描画アルゴリズムとして Sugiyama らの方法が知られている．このアルゴリズムはまず，各長辺上にダミー頂点と呼ぶ頂点を導入する．次に，各階層における頂点の配置順序決定を行い，各階層における頂点の配置位置の決定を行う．頂点順序決定では隣接頂点の順番の平均値から各階層の頂点の順序を決定し，辺交差数をできるだけ少なくする重心法，頂点座標決定では頂点に優先度を与えて優先度の大きい順に配置位置を決定する優先度法などが用いられている．頂点の配置位置を決定した後，Sugiyama らのアルゴリズムは各辺を直線で描く．このような描画を階層グラフの直線描画と呼ぶ．

直線描画では，グラフが密になると辺交差数が非常に多くなり，グラフの構造が把握しづらくなる問題点がある．また，与えられた階層グラフに長辺が多くある場合，ダミー頂点の個数が多くなり，それに伴って描画幅も大きくなってしまう問題点もある．階層描画では，一般に，描画幅は小さいことが望ましいとされている．

本論文ではまず，動的計画法を用いた新しい頂点配置決定法を提案する．提案法は，描画中の重み付き辺長の 2 乗の総和を小さくすることを目的としており，グラフの各階層を順に見ていき，それぞれの頂点の配置を動的計画法を用いて求めるものである．従来法の一つである優先度法と比べて，提案法は，近接性，バランス性，直線性などの評価基準に関して，より優れた結果を得られることを計算機実験により示す．

本論文では次に，頂点を上の階層から下の階層へたどっていくときの到達可能性を変えない範囲で，複数の長辺がダミー頂点を共有することを許すことにより，ダミー頂点や辺の総数を減らす方法を提案する．そして，計算機実験により，その提案法がグラフ描画を

簡潔なものにし，描画幅を小さくするために有効であることを示す．ダミー頂点の共有を許す場合，処理後のグラフの頂点数や辺の本数をできるだけ少なくすることが望まれる．本論文では，問題の入力を連結 3 階層グラフに限定した場合でも，頂点数を最小化する問題及び辺数を最小化する問題が NP 困難であることを示す．

本論文では最後に，辺を垂直・水平線分からなる経路として描くことにより辺交差数を減らす方法を提案する．各辺を直線で描く直線描画に対し，各辺を垂直・水平線分からなる経路として描く描画を直交描画と呼ぶ．直交描画では，連続する階層間の辺集合を，高階辺と呼ぶ辺集合に分割して，各高階辺を垂直・水平線分を用いて交差なしに描く．階層グラフの直交描画を求める方法は，これまでにもいくつか提案されているが，高階辺がもつ頂点数や描画の方法に制限を設けている．本論文では，階層グラフから，高階辺の集合を決定する方法，各頂点の座標の決定方法，及び各高階辺を垂直・水平線分を用いて描く方法を提案する．高階辺を描く方法は，異なる辺が水平線分の一部を共有することを許し，各高階辺の描画に 2 本の水平線分を用い得るものとして，辺交差数と水平線分が占める y 座標の個数を共に少なく抑えながら，直交描画を求めるものである．提案法の有効性は計算機実験により示す．

目次

図目次

表目次

第 1 章

緒論

グラフはネットワーク，路線図，分子構造などのさまざまな構造や関係を表すために用いられている．グラフを分りやすく，また美しく描画する方法の研究は古くから盛んに行われている [1, 2]．グラフの代表的な描画形態の一つである階層描画は，作業工程図，科目間関係図，有向グラフなどの描画に広く用いられている．階層描画を求める方法について，今までに多くの研究が行われてきている [1]〜[3]．それらの研究の多くはグラフがより見やすくなるように，描画中の辺の交差数を減少させる方法や，頂点の配置位置を決定する方法についてのものである．

有向グラフの代表的な階層描画アルゴリズムとして Sugiyama ら [2, 3] の方法が知られている．このアルゴリズムは四つの段階からなっており，第 1 段階は有向グラフの非閉路化，第 2 段階は頂点の階層割当て，第 3 段階は各階層における頂点の配置順序決定，第 4 段階は各階層における頂点の配置位置の決定を行う．これらのうち，第 3 段階と第 4 段階を用いることにより，階層グラフの描画を行うことができる．第 3 段階に関してはこれまでに多くのアルゴリズムが提案されており [4]，代表的な発見的手法として重心法 [2, 3] が知られている．また，頂点の座標決定に関しても，いくつかの方法が提案されている [2],[3],[5]〜[7].

一般に階層描画アルゴリズムでは，連続していない階層の 2 頂点を結ぶ辺（長辺と呼ぶ）のそれぞれに対して，それがまたぐ階層ごとにダミー頂点と呼ぶ仮の頂点を追加する．ダミー頂点の導入により，グラフ中の各辺が，連続した階層の頂点間を結ぶことになる．Sugiyama らの方法は，頂点座標の決定後，各辺を直線で描くが，このような描画を直線描画と呼ぶ．図 1.1(b) は，同図 (a) のグラフの直線描画の例であり，黒丸で示した七つの頂点がダミー頂点である．

直線描画では，通常，

- 最小分離：各階層において頂点をある最小距離以上離して配置すること，

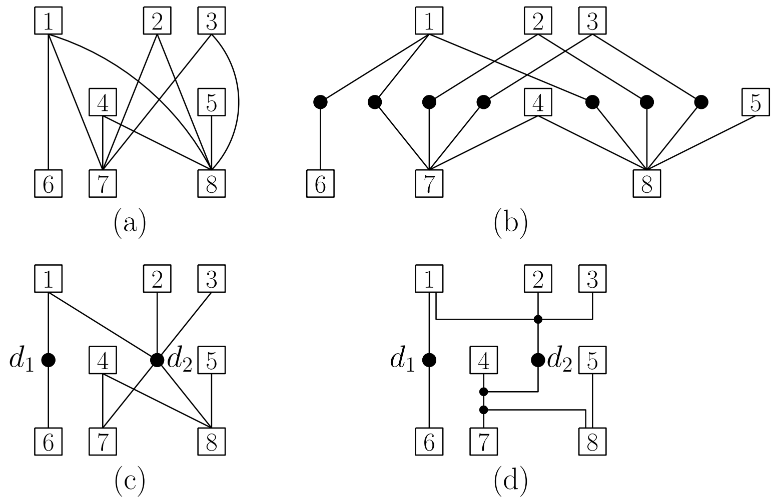


図 1.1: 階層描画の例 1

- 辺交差数小：辺の交差を少なくすること，
- 近接性：連続する階層の隣接頂点はできるだけ近くに配置すること，
- 直線性：非連続階層上の頂点間を結ぶ辺に対し，折れ点を少なくし，できれば垂直線を用いて描くこと，
- バランス性：各頂点は隣接頂点の重心あるいはそれに近い位置に配置すること，
- 描画幅小：描画幅は小さくすること

などが望まれる [2, 4].

頂点の座標決定手法のうち，優先度法 [2, 3] は，最小分離，直線性及びバランス性を主に考慮した発見的手法である．この方法は，各階層を上から下，下から上と順に見ていき，それぞれの階層上の頂点座標を，頂点に与えた優先度の順に改善していくものである．QP 法 [2, 3] は，最小分離と直線性を考慮した制約条件の下で，近接性とバランス性を反映した目的関数の最小化を行う二次計画問題を解くものである．文献 [5] は，近接性と直線性を重視した方法を提案している．この方法は，重み付き辺長（辺の端点の x 座標の差に辺の重みを乗じたもの）の総和を最小とする問題を線形計画問題に帰着して解くものである．また文献 [6, 7] は，直線性を重視した効率的な頂点配置決定法を提案している．ただし，文献 [6, 7] のアルゴリズムは，基本的に，ダミー頂点どうしをつなぐ任意の 2 辺が互いに交差しないように，頂点の配置順序が決められていることを前提としている．

本論文では，優先度法と同様，各階層の頂点座標の改善を階層ごとに順に行うというアプローチを採ることにし，これに従った新しい頂点座標決定法を提案する．この方法を提案法 1 と呼ぶことにする．提案法 1 は，描画中の重み付き辺長の 2 乗の総和を小さくすることを目的としており，グラフの各階層を順に見ていき，それぞれの頂点の配置を動的計

画法を用いて求めるものである．そして，優先度法との比較実験により，提案法 1 が，近接性，直線性及びバランス性に関して良好な描画を得ることを示す．

与えられた階層グラフに長辺が多くある場合，ダミー頂点の個数が多くなり，それに伴って描画幅も大きくなる．前述のように，一般に，階層描画の描画幅は小さいことが望ましいとされている．本研究では，頂点を上の階層から下の階層へたどっていくときの到達可能性を変えない範囲で，ダミー頂点を複数の辺に共有させることを考える．このような処理は，文献 [8] にもみられるが，そこでの方法は，長辺を順に見ていき，先に作成したダミー頂点を利用できる場合に共有させるという極めて簡単なものであった．

図 1.1(c) は，同図 (a) のグラフに対してダミー頂点の共有化処理を行った後，直線描画を求めたものである．ダミー頂点を複数の辺に共有させる処理は，描画を簡潔にし，描画幅をより小さくするために有効であると考えられる．本論文では，まずその有効性を示すため，ダミー頂点の共有処理を行う単純なアルゴリズムを提案し，計算機実験の結果を示す．この方法を提案法 2 と呼ぶことにする．描画をより簡潔にするためには，一般に，共有処理後のグラフの頂点数や辺数をできるだけ少なくすることが望ましいが，本論文では，頂点数を最小化する問題，及び辺数を最小化する問題が NP 困難であることも示す．

直線描画に対し，各辺を垂直・水平線分からなる経路として描いた描画を直交描画と呼ぶ．階層描画アルゴリズムに関するこれまでの多くの研究は直線描画についてのものであるが，直交描画を求める方法もいくつか提案されている [9]〜[11]．これらの方法では，同一頂点に接続する辺が垂直・水平線分の一部を共有することを許している．

直線描画は，グラフが密になると辺交差数が非常に多くなり，グラフの構造が把握しづらくなることがある．直線描画では辺交差数が多くなるグラフでも，直交描画では辺交差数が大幅に少なくなることがある．これは，異なる辺に垂直・水平線分を共有させることにより，描画をより簡潔にできるためである．図 1.1(d) は，ダミー頂点の共有化処理を行った後，直交描画を求めたものである（図中，ダミー頂点より小さい黒丸は辺の分岐点を示している）．

グラフのある階層の頂点のある集合 VS とすぐ下の階層の頂点のある集合 VS' について，$VS \cup VS'$ から誘導される部分グラフ G' が完全 2 部グラフをなすとき，G' の辺集合を完全 2 部辺集合と呼ぶ．階層グラフの直交描画アルゴリズムでは，連続する 2 階層ごとにそれらの間の辺の集合を完全 2 部辺集合に分割する．分割された完全 2 部辺集合のそれぞれを高階辺と呼ぶことにする．本研究では，高階辺を辺交差なしで描くことができるものとする（図 1.2 に例を示す）．そして，この前提の下で，階層グラフの直交描画を求める新しいアルゴリズムを提案する．この方法を提案法 3 と呼ぶことにする．

階層グラフの直交描画を求めるためのこれまでの方法 [9]〜[11] では，高階辺がもつ頂点数や描画方法に制限を加えている．例えば，文献 [9] は，異なる辺の水平線分が一部を共有することを禁止しており，文献 [10] は，高階辺がもつ上階層の頂点数 $|VS|$ が 1 に限

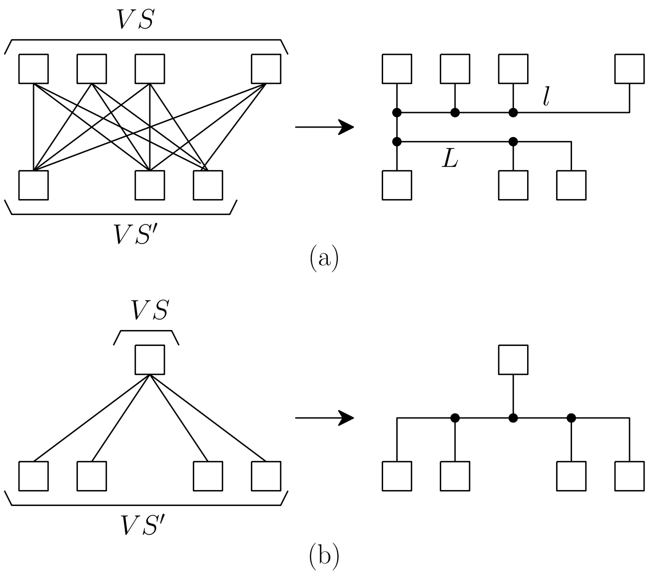


図 1.2: 高階辺の描画

るという制限を設けている．また，文献 [11] は，$|VS| \geq 2$ となることを許しているが，各高階辺の描画に用い得る水平線分の本数を 1 以下に制限している．図 1.2(a) 右は，2 本の水平線分 l, L を用いることにより，直線描画をする場合に比べ辺交差数を大きく削減し得る例でもあるが，文献 [9]〜[11] の方法は，このような描画を許していない．提案法 3 は，高階辺がもつ頂点数 $|VS|$, $|VS'|$ の値に制限を設けておらず，また，異なる辺が水平線分の一部を共有することを許している．さらに，各高階辺の描画に 2 本の水平線分を用い得るものとしている．

本論文の構成は以下のとおりである．まず第 2 章においていくつかの定義を行い，代表的な階層描画アルゴリズムである Sugiyama らのアルゴリズムを説明する．第 3 章では動的計画法を用いた頂点配置アルゴリズムを提案する．第 4 章ではダミー頂点の共有化アルゴリズムの提案を行った後，頂点数を最小化する問題，及び辺数を最小化する問題が NP 困難であることを示す．第 5 章では階層グラフの直交描画を求めるアルゴリズムを提案する．最後に第 6 章において，本論文の結論をまとめ，今後の課題について述べる．

第 2 章

従来の直線描画アルゴリズム

2.1 緒言

階層グラフの描画アルゴリズムに関して，これまで広く研究が行われているが，第 1 章で述べたように，それらの多くは直線描画を求めるためのものであった．本章では，階層グラフと階層描画に関するいくつかの定義を行った後に，代表的な直線描画アルゴリズムである Sugiyama ら [2, 3] の方法について説明する．さらに，このアルゴリズムと直線描画の問題点について考える．

2.2 諸定義

階層グラフ及び階層描画に関する定義を示す．$G = (V, E)$ を単純なグラフとする．頂点集合 V が空でない部分集合 $V_1, V_2, \ldots, V_h$ に分割されており，各辺 $(v, w) \in E$ に対して，v, w が異なる部分集合に属するとき，G を階層グラフと呼び，各部分集合 $V_1, V_2, \ldots, V_h$ を G の階層と呼ぶ．h を G の階層数と呼び，各 V_i $(1 \leq i \leq h)$ を第 i 階層と呼ぶ．また，$i = 1, 2, \ldots, h-1$ について，V_i と V_{i+1} は連続しているという．階層数が 3 以上の階層グラフを多階層グラフと呼ぶ．本論文では，階層描画を行う際，平面上に距離 $Dist$（> 1）ごとに引いた h 本の水平線を考え，各 V_i の頂点を上から i 番目の水平線上に配置するものとする．頂点の x 座標の最大値と最小値の差を描画幅と呼ぶ．直線描画とは，連続した 2 階層にはさまれる部分ごとに各辺を直線で描いた描画である．

階層グラフの直線描画では,

- 最小分離：各階層において頂点をある最小距離以上離して配置すること,
- 辺交差数小：辺の交差を少なくすること,
- 近接性：連続する階層の隣接頂点はできるだけ近くに配置すること,
- 直線性：非連続階層上の頂点間を結ぶ辺に対し，折れ点を少なくし，できれば垂直線

を用いて描くこと，

- バランス性：各頂点は隣接頂点の座標の平均あるいはそれに近い位置に配置すること，
- 階層幅小：グラフを描画するのに必要な幅は小さくすること

などが望まれる [2, 4].

辺 $e = (v, w)$ に対して，$v \in V_i, w \in V_j$ $(i < j)$ としたとき，$j - i$ を e のスパンと呼ぶ．スパンが 2 以上の辺を長辺と呼ぶ．通常，階層描画では，各長辺 $e = (v, w)$（$v \in V_i$, $w \in V_j$, $i + 1 < j$）に対し，$j - i - 1$ 個のダミー頂点 $d_1, d_2, \ldots, d_{j-i-1}$ を作り，(v, w) を道 $[v, d_1, d_2, \ldots, d_{j-i-1}, w]$ に置き換える．各 d_t（$t = 1, 2, \ldots, j - i - 1$）は第 $i + t$ 階層に加える．ダミー頂点に対し，G にもともと存在していた頂点を実頂点と呼ぶ．これ以降，単に頂点といえば，実頂点とダミー頂点の両方を意味するものとする．G のどの階層 V_i も空ではないので，$h \leq |V|$ である．$n = |V|$ とし，各階層 V_i の頂点数を n_i とする．

描画の際には，各実頂点は一辺の長さが W_R の正方形で描き，各ダミー頂点は直径が W_D の円で描くものとする．ここで，$W_D \leq W_R < 1$ とする．各頂点 v の x 座標は整数値に限るものとし (これにより，前述の最小分離を実現する)，その値を $x(v)$ と表す．v が実頂点，ダミー頂点のいずれである場合についても，正方形あるいは円の中心の座標が $x(v)$ に等しくなるようにするものとする．また，各頂点 v に対して，上隣接頂点の x 座標の平均値を上重心，下隣接頂点の x 座標の平均値を下重心と呼ぶ．さらに，v のすべての隣接頂点の x 座標の平均値を重心と呼ぶ．前述のバランス性より，重心と $x(v)$ との差は小さいことが望ましい．

Sugiyama らの方法は，頂点座標の決定後，各辺を直線で描くが，このような描画を直線描画と呼ぶ．図 2.1(b) は，同図 (a) のグラフの直線描画の例であり，黒丸で示した四つの頂点がダミー頂点である．直線描画に対して，各辺を垂直・水平線分からなる経路として描く描画法を直交描画と呼ぶ．図 2.1(c) は，同図 (a) のグラフの直交描画の例である．

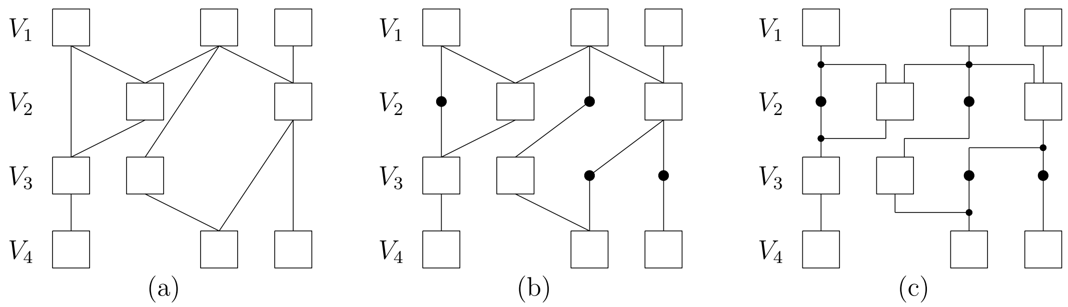


図 2.1: 階層描画の例 2

2.3 Sugiyama らのアルゴリズム

Sugiyama らのアルゴリズムは有向グラフの代表的な階層描画アルゴリズムである．このアルゴリズムは四つの段階からなっており，第 1 段階は有向グラフの非閉路化，第 2 段階は頂点の階層割当て，第 3 段階は各階層における頂点の配置順序決定，第 4 段階は各階層における頂点の配置位置の決定を行う．各長辺上のダミー頂点の作成は第 2 段階と第 3 段階の間に行われる．第 3,4 段階が階層グラフの描画に用いることができる．それぞれの段階について多くの研究が行われている [4]．第 3 段階では隣接頂点の順番の平均値から各階層の頂点の順序を決定し，辺交差数をできるだけ少なくする重心法，第 4 段階では頂点に優先度を与えて優先度の大きい順に配置位置を決定する優先度法などが提案されている．**2.3.1** では重心法について，**2.3.2** では優先度法について説明する．

2.3.1 重心法

重心法は多階層グラフの各段における頂点順序の改善を，全段に渡って上から下，下から上と順次繰返す．上から下への改善試行を Down 過程，下から上への改善試行を Up 過程と呼ぶ．

Down 過程では，まず，V_1 の頂点順序を固定し，V_2 の頂点順序を入れ替える．各頂点 $v \in V_2$ について，上隣接頂点が V_1 上で左から何番目に置かれているかという順番を求め，それらの平均値を計算する．この平均値のことを v の列重心と呼ぶ．そして，V_2 の頂点を列重心の非減少順に左から順に並べ替える．次に，$i = 2, 3, \cdots, h-1$ について，上と同様に，V_i 上の頂点順序を固定し，V_{i+1} の頂点を列重心の順に並び替えるという処理を行う．

Up 過程では，$i = h, h-1, \cdots, 2$ について，Down 過程とは逆に，V_i 上の頂点順序を固定して，V_{i-1} の各頂点を，下隣接頂点の順番の平均値（行重心）の順に並び替えていく処理を行う．

重心法は，Down 過程と Up 過程を交互に，辺交差数が減少しなくなるまで繰返すことにより，各階層上の頂点順序を決定するものである．

2.3.2 優先度法

優先度法は重心法と同様に多階層グラフの各段における頂点配置座標の改善手順を，全段に渡って上から下，下から上と順次繰返す．上から下への改善試行を Down 過程，下から上への改善試行を Up 過程と呼ぶ．各段における頂点配置座標を改善する際に，各頂点に Down 過程なら上隣接頂点数を，Up 過程なら下隣接頂点数を優先度として与える．た

だし，ダミー頂点は Up 過程，Down 過程両方において最大の優先度を与える．Down 過程では，まず，V_1 の頂点順序を固定し，V_2 の各頂点を優先度が高い頂点から順に，頂点順序を変えることなく，なるべく上重心に近い座標に頂点の座標を配置する．その際，対象としている頂点より優先度が低い頂点は動かすことができるが，より優先度が高い頂点は動かすことができない．次に，$i = 2, 3, \cdots, h-1$ について，上と同様に，V_i 上の頂点順序を固定し，V_{i+1} の頂点の配置を行う．

Up 過程では，$i = h, h-1, \cdots, 2$ について，Down 過程とは逆に，V_i 上の頂点順序を固定して，V_{i-1} の各頂点を，優先度が高い順に，頂点順序を変えることなく，なるべく下重心に近い座標に頂点の座標を配置する．Down 過程と同様に，対象としている頂点より優先度が低い頂点は動かすことができるが，より優先度が高い頂点は動かすことができない．

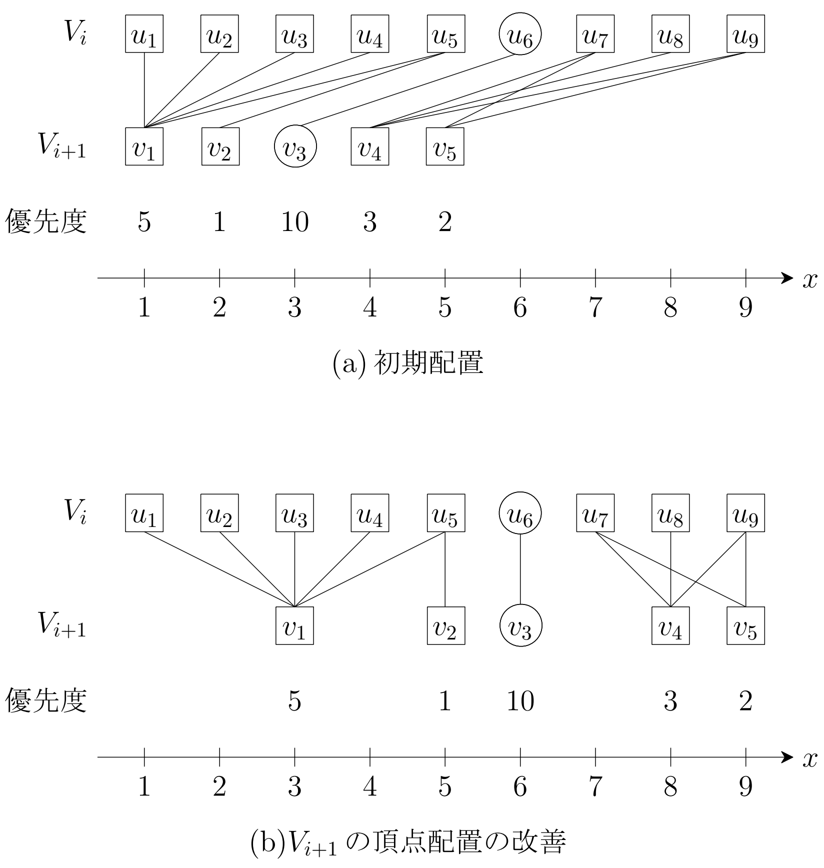


図 2.2: 優先度法の操作の説明

優先度法における頂点の配置座標の改善の手順の例を図 2.2 に示す．(a）はある多階層グラフの第 $i-1$ 段および第 i 段の部分 2 段グラフを示している．いま Down のプロセスの最中であるとして第 i 段を改善する．なお u_6 と v_3 はダミー頂点であり，この例では最大の優先度として，優先度を 10 と与えている．$v_1, \cdots, v_5$ のうちで最も優先度の高いの

は v_3 であるので，まず v_3 を動かす．上重心が 6 なので，$v_1, \cdots, v_5$ の座標は $1,2,3,4,5$ から $1,2,6,7,8$ に変わり，(u_6, v_3) は垂直な辺となる．次に優先度の高い v_1 を改善する．上重心が 3 なので，座標は $3,4,6,7,8$ となる．次に v_4 の改善をする．上重心が 8 なので，$3,4,6,8,9$ となる．次に優先度が高い v_5 の改善をするが，上重心が 8 であり，座標 8 にはより優先度が高い v_4 が存在しているので，動かすことができない．最後に v_2 を改善する．上重心が 5 なので，$3,5,6,8,9$ となり，図 2.2（b）が得られる．

2.4 Sugiyama らのアルゴリズムと直線描画の問題点

優先度法は直線性とバランス性を主に考慮した手法であり，優先度の高い頂点から順に頂点座標を決めていくので，優先度の低い頂点は近接性が悪い位置に配置されることがある．このことから優先度の改良点として，頂点の近接性をより考慮することが考えられる．

ダミー頂点は従来，各長辺に対して，長辺が跨ぐ各階層に一つずつ導入されてる．よって，図 2.3 のように長辺が多いグラフだと，ダミー頂点が多くなり，階層幅が大きくなってしまう問題がある．

また，直線描画では，グラフが密になると図 2.3 に示すように，辺同士の交差が非常に多くなり，頂点の隣接関係を把握することが困難になる．頂点の順序を入れ替えることで，交差を少なからず削減することができるが，それにも限度があり，辺数が多い完全 2 部グラフを部分グラフとしてもつとき，辺交差数が多くなることを避けられない．そこで，連続する階層間の辺集合を，高階辺と呼ぶ辺集合に分割して，各高階辺を垂直・水平線分を用いて交差なしに描くことにより，交差を削減し，見やすさの向上を目指した方法の研究が行われている [9]〜[11]．しかし，これらの方法は，高階辺がもつ頂点数や描画の仕方に制限を設けており，描画アルゴリズムとして不十分な部分がある．

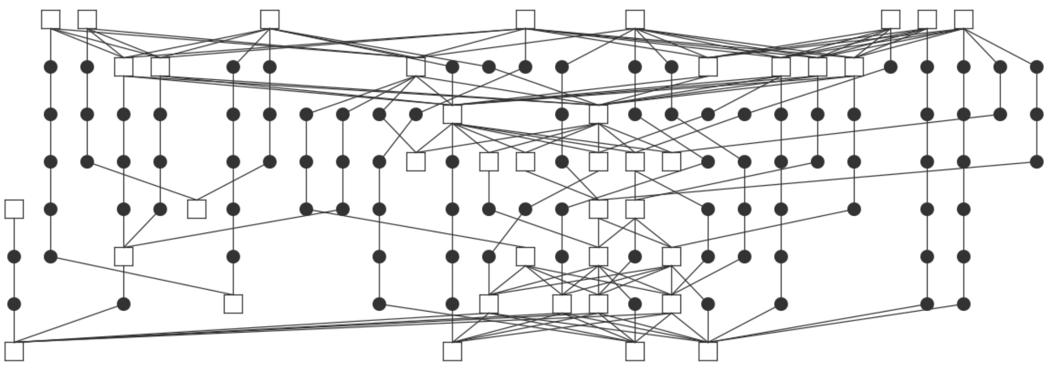


図 2.3: 長辺が多い階層グラフの直線描画の描画例

2.5 結言

本章では，階層グラフと階層描画に関するいくつかの定義を行った後，階層描画を求める代用的なアルゴリズムである Sugiyama らのアルゴリズムの説明を行い，このアルゴリズムと直線描画の問題点の考察を行った．第 3 章では，優先度法では考慮されていなかった近接性を考慮した新しい頂点座標決定アルゴリズムを提案する．長辺が多いとダミー頂点が増加し，それに伴い描画幅が増加するという問題に対しては，第 4 章で，複数の長辺にダミー頂点を共有させることにより，ダミー頂点の数を減らす方法を提案する．密なグラフでは交差数が多くなり，頂点の隣接関係を把握することが困難になる問題に対しては，第 5 章において，グラフの各辺を垂直・水平線分からなる経路とみなし，異なる辺の水平線分が一部を共有させることにより，従来の描画に比べ，辺の交差を大幅に減らすことができる直交描画アルゴリズムを提案する．

第 3 章

階層グラフの直線描画における頂点座標決定アルゴリズム

3.1 緒言

直線描画における頂点の座標決定法については，これまでにいくつかの方法が提案されている [2],[3],[5]〜[7]．それらのうち，優先度法 [2, 3] は，最小分離，直線性及びバランス性を主に考慮した発見的手法である．この方法は，各階層を上から下，下から上と順に見ていき，それぞれの階層上の頂点座標を，頂点に与えた優先度の順に改善していくものである．QP 法 [2, 3] は，最小分離と直線性を考慮した制約条件の下で，近接性とバランス性を反映した目的関数の最小化を行う二次計画問題を解くものである．文献 [5] は，近接性と直線性を重視した方法を提案している．この方法は，重み付き辺長（辺の端点の x 座標の差に辺の重みを乗じたもの）の総和を最小とする問題を線形計画問題に帰着して解くものである．また文献 [6, 7] は，直線性を重視した効率的な頂点配置決定法を提案している．ただし，文献 [6, 7] のアルゴリズムは，基本的に，ダミー頂点どうしをつなぐ任意の 2 辺が互いに交差しないように，頂点の配置順序が決められていることを前提としている．

本章では，優先度法と同様，各階層の頂点座標の改善を順に行うというアプローチを採ることにし，これに従った新しい頂点座標決定法を提案する．提案法は，描画中の重み付き辺長の 2 乗の総和を小さくすることを目的としており，グラフの各階層を順に見ていき，それぞれの頂点の配置を動的計画法を用いて求めるものである．そして，優先度法との比較実験により，この提案法が，近接性，直線性及びバランス性に関して良好な描画を得ることを示す．

重み付き辺長の 2 乗の総和が最小となる頂点配置を求める問題は，QP 法と同様，二次計画問題に帰着できる．二次計画問題に帰着し，その厳密解を求めることにより最適解を得ることができる．一方，提案法は発見的手法であり，一般に最適解を求めることはでき

ないが，二次計画問題の厳密解を計算しないため，二次計画問題による解法に比べ，計算時間や扱い得る問題のサイズに関して優位性をもっている．また，動的計画法を用いて各階層の頂点配置を順に求めるという方法は，描画中の重み付き辺長の 2 乗の総和を目的関数とする場合だけでなく，後述するように，辺長（辺の端点の x 座標の差）の総和を目的関数とする場合や，近接性，直線性，バランス性の三つを反映したある値を目的関数とする場合でも用いることができる [12, 13].

本章では，まず，**3.2** で頂点座標決定アルゴリズムに関する諸定義を行う．次に，**3.3** で提案法 1 について述べ，**3.4** において，**2.3.2** で説明した優先度法との比較実験の結果を示す．

3.2 頂点座標決定アルゴリズムに関する諸定義

ダミー頂点導入後の階層グラフを $G = (V, E)$ とし，その階層を上から順に $V_1, V_2, \ldots, V_h$ とする．$n = |V|$ とし，$i = 1, 2, \ldots, h$ について，$n_i = |V_i|$ とする．G は連結であるものとする．G のすべてのダミー頂点からなる集合を DS と表す．各頂点 $v \in V_i$ の隣接頂点のうち，V_{i-1} に属するものを v の上隣接頂点と呼び，その集合を $A^U(v)$ と表す．同様に，V_{i+1} に属するものを下隣接頂点と呼び，その集合を $A^D(v)$ と表す．v のすべての隣接頂点からなる集合 $A^U(v) \cup A^D(v)$ を $A(v)$ と表す．G は連結であるので，任意の $v \in V$ に対して $A(v) \neq \emptyset$ である．各 i に対し，V_i の頂点と V_{i-1} の頂点との間の辺すべてからなる集合を $E^U(i)$，V_i の頂点と V_{i+1} の頂点との間の辺すべてからなる集合を $E^D(i)$ と表す．

ダミー頂点の導入により，G 中のすべての辺は連続する階層上の頂点間を結ぶことになる．連続階層間の距離は一定であるので，各辺 $e = (v, w)$ の長さは，端点 v, w の x 座標の差で決まる．これ以降，特に断らない限り，v, w の x 座標の差 $|x(v) - x(w)|$ を e の辺長と呼ぶ．この値は，v, w 間の L_1 距離（マンハッタン距離）から階層間の距離を引いたものである．

頂点の座標決定における階層描画の質の評価値として，近接性，直線性，バランス性のそれぞれに対応した以下の三つの値を用いる．まず近接性に関しては，次の $EL2$ を用いる．連続階層間の距離は一定であるから，$EL2$ の値を最小にする頂点配置は，辺長を端点間のユークリッド距離としたときの辺長の 2 乗の総和を最小にするものである．

$$EL2 = \sum_{(v,w) \in E} |x(v) - x(w)|^2. \tag{3.1}$$

直線性に関しては，次の $DL2$ を用いる．

$$DL2 = \sum_{v \in DS} \Bigl(\sum_{w \in A(v)} |x(v) - x(w)|^2 \Bigr). \tag{3.2}$$

バランス性に関しては，次の $BAL2$ を用いる．

$$BAL2 = \sum_{v \in V} \left| x(v) - \frac{\sum_{w \in A(v)} x(w)}{|A(v)|} \right|^2. \tag{3.3}$$

三つの評価値のいずれも値が小さいほど望ましい．

3.3 提案法 1

ダミー頂点導入後の連結階層グラフ $G = (V, E)$ に対し，各階層における頂点の配置順序が決められているものとする．本節では，G の階層描画における頂点座標を決定する新しいアルゴリズムを提案する．

3.3.1 概略

提案法 1 は，次に示す値 $WEL2$ が小さくなるように頂点座標を定めようとするものである．

$$WEL2 = \sum_{(v,w) \in E} \Big(\omega(v,w)^2 \cdot |x(v) - x(w)|^2 \Big). \tag{3.4}$$

ここで $\omega(v,w)$ は辺 (v,w) の重みであり，端点が実頂点かダミー頂点かの区別により値を決めるものとする．$\omega(v,w) \cdot |x(v) - x(w)|$ なる値を，辺 (v,w) の重み付き辺長と呼ぶ．このような重み付けは,

$$WEL = \sum_{(v,w) \in E} \Big(\omega(v,w) \cdot |x(v) - x(w)| \Big) \tag{3.5}$$

の最小化を目的とした文献 [5] の頂点座標決定法でも行われている．そこでは，直線性を考慮して，v, w の両方が実頂点の場合は $\omega(v,w) = 1$，一方が実頂点で他方がダミー頂点の場合は $\omega(v,w) = 2$，両方ともダミー頂点の場合は $\omega(v,w) = 8$ としている．本研究でも，各辺の重みを同様に定めるものとする．

各頂点 $v \in V$ について,

$$wbar(v) = \frac{\sum_{w \in A(v)} (\omega(v,w)^2 \cdot x(w))}{\sum_{w \in A(v)} \omega(v,w)^2} \tag{3.6}$$

なる値を v の重み付き重心と呼ぶ．すべての $w \in A(v)$ に対して $\omega(v,w)$ が同一の値であるとき，$wbar(v)$ は v の重心に一致する．式 (3.6) における $A(v)$ を $A^U(v)$ で置き換えたときの値を v の重み付き上重心，$A^D(v)$ で置き換えたときの値を重み付き下重心と呼ぶ．

$WEL2$ を小さく抑えることは，WEL に比べ，非常に長い辺の存在を抑制する効果がある．また，$WEL2$ を目的関数とすることは，$EL2, DL2$ の他に，$BAL2$ の値を小さく

する効果もあると考えられる．これは，任意の頂点 v に対し，その隣接頂点の座標を一旦固定したとき，$\sum_{w \in A(v)} |x(v) - x(w)|^2$ が最小になるのは $x(v)$ が v の重心に一致するときであり，この値は，$x(v)$ と v の重心の差が大きくなるにつれて単調増加するためである．

提案法 1 は，頂点の初期配置を定めた後，G の各階層に順に注目し，その階層の頂点配置を改善する．上の階層から順に，V_{i-1}（$i = 2, 3, \ldots, h$）の頂点座標を固定し，V_i の頂点座標を改善していく一連の処理を Down 過程と呼ぶ．また，下の階層から順に，V_{i+1}（$i = h-1, h-2, \ldots, 1$）の頂点座標を固定し，V_i の頂点座標を改善していく一連の処理を Up 過程と呼ぶ.

Down 過程で V_i の頂点配置を更新する際，固定した階層 V_{i-1} との間の評価値 $Lwel2^D(i)$ が最小となる頂点配置を動的計画法により求める．同様に，Up 過程で V_i の頂点配置を更新する際，V_{i+1} との間の評価値 $Lwel2^U(i)$ が最小となる頂点配置を求める．ここで，$Lwel2^D(i), Lwel2^U(i)$ は以下のようなものである．

$$Lwel2^D(i) = \sum_{(v,w) \in E^U(i)} \Big(\omega(v,w)^2 \cdot |x(v) - x(w)|^2 \Big), \tag{3.7}$$

$$Lwel2^U(i) = \sum_{(v,w) \in E^D(i)} \Big(\omega(v,w)^2 \cdot |x(v) - x(w)|^2 \Big). \tag{3.8}$$

多階層グラフでは，連続する片方の階層との間の評価値だけを最小にするより，連続する上下両方の階層との間の評価値を最小にする方が描画全体の質の改善につながると期待できる．Down 過程（Up 過程）において，$i = 2, 3, \ldots, h-1$ に対して，V_{i-1} と V_{i+1} の頂点の座標を固定し，$Lwel2^D(i) + Lwel2^U(i)$ が最小になるように V_i の頂点座標を改善する処理を考え，BDown 過程（BUp 過程）と呼ぶことにする．

以下に，提案法 1 の処理手順を示す.

(A) G の各頂点 v に対し，以下のようにして初期 x 座標を定める．
　　v を含む階層を V_i とする．v が V_i の頂点のうち
　　左から j 番目のものであれば，$x(v) = j$ とする．

(B) Down 過程と Up 過程を一回ずつ行う．

(C) BDown 過程と BUp 過程を交互に繰返し，頂点座標を更新する．

(D) (C) で求めた頂点配置のうち，最も評価値のよいものを選び，それに対して後処理を行う．

上記の (C) の終了条件は，各過程の繰返し回数があらかじめ定めた定数 N に到達するか，2 回の連続した過程で $WEL2$ の値が改善されないこととする．

次節以降で，各過程と後処理について述べる．

3.3.2 Down 過程と Up 過程

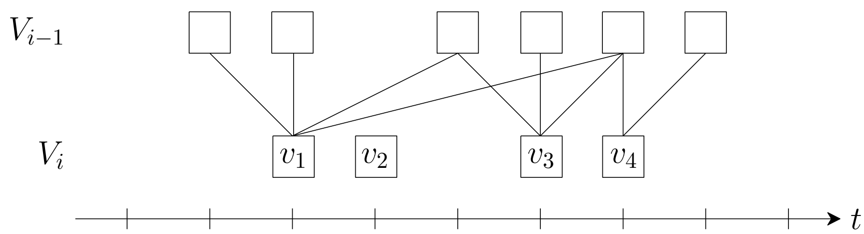
以下，Down 過程について説明する．Up 過程については，同様であるので，説明を省略する．

3.3.2.1 動的計画法を用いた頂点配置の改善

階層 V_i（$i = 2, 3, \ldots, h$）の頂点を左から順に v_1, $v_2, \ldots, v_{n_i}$ とする．V_{i-1} の頂点の x 座標を一旦固定し，頂点 $v_j \in V_i$ をある x 座標 $t \in Z$（Z：整数全体の集合）に置いたとき，v_j と V_{i-1} の頂点を結ぶ辺の重み付き辺長の 2 乗の和を $el^D(v_j, t)$ とする．すなわち，

$$el^D(v_j, t) = \sum_{w \in A^U(v_j)} (\omega(v_j, w)^2 \cdot |t - x(w)|^2) \tag{3.9}$$

である．図 3.1 に $el^D(v_j, t)$ の例を示す．ここでは，図中の各辺の重みを 1 としている．



t	1	2	3	4	5	6	7	8	9
$el^D(v_1, t)$	57	35	21	15	17	27	45	71	105
$el^D(v_4, t)$	85	61	41	25	13	5	1	1	5

図 3.1: $el^D(v_j, t)$ の例

v_j を x 座標 t に置いたとき，V_{i-1} の頂点と v_1, $v_2, \ldots, v_j$ を結ぶ辺の重み付き辺長の 2 乗の和の最小値を $mel^D(v_j, t)$ と定義する．$j = 1$ のとき，

$$mel^D(v_1, t) = el^D(v_1, t) \tag{3.10}$$

であり，$j > 1$ の場合には次式が成立する．

$$mel^D(v_j, t) = \min_{\substack{s \in Z \\ s < t}} mel^D(v_{j-1}, s) + el^D(v_j, t). \tag{3.11}$$

各頂点 v_j に対して，

$$optmel^D(v_j) = \min_{t \in Z} mel^D(v_j, t) \tag{3.12}$$

とする．V_i の頂点配置で，$Lwel2^D(i) = optmel^D(v_{n_i})$ となるものを，（Down 過程における）V_i の最適解と呼ぶことにする．

Down 過程における V_i の頂点座標の更新の際には，次の手順を実行する．

(a) 各頂点 $v_j \in V_i$ と各 t に対して，$el^D(v_j, t)$ を求める．
(b) $j = 1, 2, \ldots, n_i$ の順に，各 t に対して，式 (3.10), (3.11) に従って $mel^D(v_j, t)$ を計算する．

このような動的計画法により $optmel^D(v_{n_i})$ まで求めた後，逆追跡を行うことにより，最適解となる頂点配置を定める．

以下では，各頂点 v_j の x 座標 t を，ある集合 $R(v_j)$ の要素に限定してよいことを示す．

$i = 2, 3, \ldots, h$ に対し，階層 V_i の頂点配置の更新が終わった時点における V_i の頂点の x 座標の最小値を $xmin_i$，最大値を $xmax_i$ とする．**3.3.1** で述べたように，初期描画において，V_1 の頂点の x 座標の最小値は 1，最大値は n_1 である．そこで，$xmin_1 = 1$，$xmax_1 = n_1$ としておく．次の補題が成立する．

補題 3.1 各整数 $i = 2, 3, \ldots, h$ に対し, V_i の最適解のうちに, 各頂点 v_j $(j = 1, 2, \ldots, n_i)$ の x 座標が

$$xmin_{i-1} - n_i + j \leq x(v_j) \leq xmax_{i-1} + j - 1 \tag{3.13}$$

を満たすようなものが存在する．

（証明）まず，V_i の最適解のうちに，以下の 3 条件を満たすものが存在することを示す．

（条件 1） V_i の頂点のうち少なくとも一つの x 座標が $xmin_{i-1}$ 以上 $xmax_{i-1}$ 以下である．

（条件 2） $xmin_{i-1}$ 以下の x 座標をもつ頂点が存在するとき，それらのうちで連続する任意の 2 頂点の x 座標の差は 1 である．

（条件 3） $xmax_{i-1}$ 以上の x 座標をもつ頂点が存在するとき，それらのうちで連続する任意の 2 頂点の x 座標の差は 1 である．

V_i のある頂点配置において，x 座標が $xmin_{i-1}$ 未満の頂点が存在する場合，次の操作を実行する．

（操作 1） V_i の頂点のうち，$xmin_{i-1}$ 未満の x 座標をもつものを $v_1, v_2, \ldots, v_k$（$1 \leq k \leq n_i$）とする．V_i の頂点で，x 座標が $xmin_{i-1}$ のものが存在するならば，$l = 1, 2, \ldots, k$ について $x(v_l) = xmin_{i-1} - k - 1 + l$ とする．一方，x 座標が $xmin_{i-1}$ の頂点が存在しないならば，$l = 1, 2, \ldots, k$ について $x(v_l) = xmin_{i-1} - k + l$ とする．

操作 1 は 0 個以上の頂点を右に移動するものである．移動する頂点と V_{i-1} の頂点を結ぶ辺がある場合には，それらの各辺について，辺長がより短くなる．したがって，操作 1

の実行により，以下の 2 条件 (i), (ii) に違反することはない．

(i) $E^U(i)$ に属する辺の重み付き辺長の 2 乗の和を増やさない．
(ii) V_i の頂点の配置順序を変えない．

次に，V_i の頂点で，x 座標が $xmax_{i-1}$ より大きいものが存在するならば，次の操作 2 を実行する．

（操作 2） V_i の頂点で，$xmax_{i-1}$ より大きい x 座標をもつものを $v_{k'}, v_{k'+1}, \ldots, v_{n_i}$（$1 \leq k' \leq n_i$）とする．$V_i$ の頂点で，x 座標が $xmax_{i-1}$ のものが存在するならば，$l = k', k'+1, \ldots, n_i$ について $x(v_l) = xmax_{i-1} - k' + 1 + l$ とする．一方，x 座標が $xmax_{i-1}$ の頂点が存在しないならば，$l = k', k'+1, \ldots, n_i$ について $x(v_l) = xmax_{i-1} - k' + l$ とする．

操作 2 についても, 移動する頂点と V_{i-1} の頂点を結ぶ辺がある場合には，それらの各辺の辺長が短くなる．よって，この操作を実行しても条件 (i), (ii) に違反することはない．

操作 1 と 2 を実行した後の頂点配置が条件 1〜3 を満たすことは明らかである．以上より，V_i の最適解のうちに，条件 1〜3 を満たすものが存在することが分かる．そのような最適解において，仮に $x(v_{n_i}) = xmin_{i-1}$ であったとしても，$j = 1, 2, \ldots, n_i$ に対して $x(v_j)$ が $xmin_{i-1} - n_i + j$ より小さくなることはない．また，$x(v_1) = xmax_{i-1}$ であったとしても，$x(v_j)$ が $xmax_{i-1} + j - 1$ より大きくなることはない． □

補題 3.1 より，各頂点 v_j の x 座標を式 (3.13) の範囲に制限してよいことが分かる．以下では，さらなる制限が可能であることを示す．$el^D(v_j, t)$ について考える．$A^U(v_j) \neq \emptyset$ のとき，$el^D(v_j, t)$ を最小とする t の値は，v_j の重み付き上重心に最も近い整数である．これは，ある一つの整数 t^* となるか，連続した二つの整数 t^*, t^*+1 となる．前者の場合

$$lb_j = rb_j = t^* \tag{3.14}$$

とおき，後者の場合

$$lb_j = t^*,\ rb_j = t^* + 1 \tag{3.15}$$

とおく．例えば図 3.1 の場合，v_1, v_4 の重み付き上重心はそれぞれ $4.25, 7.5$ であるから，$lb_1 = rb_1 = 4$, $lb_4 = 7$, $rb_4 = 8$ となる．t が lb_j より小さくなるほど，あるいは rb_j より大きくなるほど，$el^D(v_j, t)$ は大きくなる．

$A^U(v_j) = \emptyset$ であれば，t の値に関わらず $el^D(v_j, t) = 0$ となる．この場合には，補題 3.1 を考慮して，以下のように lb_j, rb_j を定める．まず，v_j の現在の x 座標 $x(v_j)$ が $xmin_{i-1} - n_i + j$ 以下であれば

$$lb_j = rb_j = xmin_{i-1} - n_i + j \tag{3.16}$$

とし，$xmin_{i-1} - n_i + j < x(v_j) < xmax_{i-1} + j - 1$ であれば

$$lb_j = rb_j = x(v_j) \tag{3.17}$$

とする．そして，$xmax_{i-1} + j - 1 \leq x(v_j)$ であれば

$$lb_j = rb_j = xmax_{i-1} + j - 1 \tag{3.18}$$

とする．式 (3.14)〜(3.18) のいずれの場合でも

$$xmin_{i-1} - n_i + j \leq lb_j \leq rb_j \leq xmax_{i-1} + j - 1 \tag{3.19}$$

が成り立つ．

各頂点 $v_j \in V_i$ に対して，2 種類の値 $minr_j, maxr_j$ を以下のように定義する．

$$minr_j = \begin{cases} \min\{lb_j, minr_{j+1} - 1\} & (j < n_i) \\ lb_j & (j = n_i) \end{cases} \tag{3.20}$$

$$maxr_j = \begin{cases} rb_j & (j = 1) \\ \max\{rb_j, maxr_{j-1} + 1\} & (j > 1) \end{cases} \tag{3.21}$$

例えば，図 3.1 の場合，

$$lb_1 = 4,\ lb_2 = 4,\ lb_3 = 6,\ lb_4 = 7$$

であるから，式 (3.20) に従って $minr_4, \ldots, minr_1$ をこの順に求めると次のようになる．

$$minr_4 = 7,\ minr_3 = 6,\ minr_2 = 4,\ minr_1 = 3.$$

また，

$$rb_1 = 4,\ rb_2 = 4,\ rb_3 = 6,\ rb_4 = 8$$

であるから，式 (3.21) に従って $maxr_1, \ldots, maxr_4$ をこの順に求めると次のようになる．

$$maxr_1 = 4,\ maxr_2 = 5,\ maxr_3 = 6,\ maxr_4 = 8.$$

式 (3.20), (3.21) より，

$$minr_1 < minr_2 < \cdots < minr_{n_i}, \tag{3.22}$$

$$maxr_1 < maxr_2 < \cdots < maxr_{n_i} \tag{3.23}$$

である．各頂点 v_j に対して，式 (3.20) より $minr_j \leq lb_j$ であり，式 (3.21) より $rb_j \leq maxr_j$ である．さらに，式 (3.19) より $lb_j \leq rb_j$ であるから，v_j に対して次式が成立する．

$$minr_j \leq maxr_j \tag{3.24}$$

ここで，集合 $R(v_j)$ を次のように定義する．

$$R(v_j) = \{minr_j, minr_j + 1, \ldots, maxr_j\}. \tag{3.25}$$

このとき，次の定理が成立する．

定理 3.1 V_i の最適解のうちに，各頂点 $v_j \in V_i$ に対して $x(v_i) \in R(v_j)$ であるものが存在する．

（証明）$n_i = 1$ のとき，式 (3.10), (3.12) より

$$optmel^D(v_1) = \min_{t \in Z} el^D(v_1, t)$$

が成り立つ．よって，$lb_1 \leq x(v_1) \leq rb_1$ となるように $x(v_1)$ を定めると最適解になる．$n_i = 1$ のとき，式 (3.20), (3.21) より $minr_1 = lb_1$, $maxr_1 = rb_1$ であるから，$x(v_1) \in R(v_1)$ である．

$n_i > 1$ の場合について述べる．

V_i の任意の最適解に対して，次の操作 3 を実行するものとする．

（操作 3）$x(v_j) > maxr_j$ である頂点 v_j が存在するとき，そのような頂点のうちで最も左にある（添字が最小の）ものを v_k とする．v_k を $x(v_k) = maxr_k$ となるように左に移動する．

操作 3 によって頂点 v_k を移動したものとする．$A^U(v_k) \neq \emptyset$ であるとき $x(v_k)$ が rb_k より大きくなるほど $el^D(v_k, x(v_k))$ が大きくなることと，$rb_k \leq maxr_k$ であることから，この移動により次の (i) に違反することはない．

(i) v_k と V_{i-1} の頂点を結ぶ辺の重み付き辺長の 2 乗の総和を増やさない．

$k = 1$ であれば，次の (ii) にも違反しない．

(ii) V_i の頂点の配置順序を変えない．

$k \geq 2$ である場合，v_{k-1} は v_k より左にあるので，$x(v_{k-1}) \leq maxr_{k-1}$ である．式 (3.23) より $maxr_{k-1} < maxr_k$ であるから，$x(v_{k-1}) < maxr_k$ が成立する．よって，この場合も，$x(v_k) = maxr_k$ となるように v_k を移動したとき，(ii) に違反しない．

$x(v_j) > maxr_j$ なる頂点 v_j が存在する間，操作 3 を繰返し実行することにより，V_i の最適解で，次の $(*)$ を満たすものが得られる．

$(*)$ $x(v_j) > maxr_j$ である頂点 v_j をもたない．

次に，$(*)$ を満たす任意の最適解に対して，次の操作 4 を実行するものとする．

（操作 4）$x(v_j) < minr_j$ である頂点 v_j が存在するとき，そのような頂点のうちで最も右にある（添字が最大の）ものを v_k とする．v_k を $x(v_k) = minr_k$ となるように右に移動する．

操作 4 によって頂点 v_k を移動したものとする．$A^U(v_k) \neq \emptyset$ であるとき $x(v_k)$ が lb_k より小さくなるほど $el^D(v_k, x(v_k))$ が大きくなることと，$minr_k \leq lb_k$ であることから，この移動により上記の (i) に違反することはない．また，式 (3.24) より $minr_k \leq maxr_k$ であるから，次の (iii) にも違反しない．

(iii) $x(v_j) > maxr_j$ となる頂点 v_j を作らない．

$k = n_i$ であれば，$x(v_k) = minr_k$ となるように v_k を移動しても，上記の (ii) に違反しない．$k \leq n_i - 1$ である場合，頂点 v_{k+1} は v_k より右にあるので，$minr_{k+1} \leq x(v_{k+1})$ である．式 (3.22) より $minr_k < minr_{k+1}$ であるから，$minr_k < x(v_{k+1})$ が成立する．よって，この場合も，v_k を移動したとき (ii) に違反しない．

$(*)$ を満たす任意の最適解に対し，$x(v_j) < minr_j$ である頂点 v_j が存在する間，操作 4 を繰返し実行することにより，各頂点 v_j について $x(v_j) \in R(v_j)$ であるような最適解が得られる． □

定理 3.1 より，V_i の各頂点 v_j の x 座標として，$R(v_i)$ の要素だけを考えればよいことになる．Down 過程において V_i の頂点配置を更新する際には，以下の手順を実行する．

(a′) 各頂点 $v_j \in V_i$ に対して，lb_j, rb_j を計算した後，$minr_j$ と $maxr_j$ を求める．その後，各頂点 $v_j \in V_i$ と各整数 $t \in R(v_j)$ に対して，$el^D(v_j, t)$ を求める．

(b′) $j = 1, 2, \ldots, n_i$ の順に, 各 $t \in R(v_j)$ に対して, 式 (3.10), (3.11) に従って $mel^D(v_j, t)$ を計算する．ただし式 (3.11) では，s として $R(v_{j-1})$ に属する値だけを考える．

このような動的計画法によって $optmel^D(v_{n_i})$ まで求めた後，逆追跡を行うことにより，V_i の最適解を求める．この逆追跡のために，上記の (b′) で $mel^D(v_j, t)$（$j > 1$）を計算する際，$mel^D(v_{j-1}, s)$ （$s \in R(v_{j-1})$）を最小とする s の値を，$smin(v_j, t)$ として記録しておく．そのような s の値が複数存在する場合には，それらのうちで最大の値を選ぶものとする．

以下，**3.3.2.2** と **3.3.2.3** において，上記の (a′), (b′) での処理と時間計算量について述べる．その後 **3.3.2.4** において，Down 過程全体の実行時間について述べる．

3.3.2.2 $el^D(v_j, t)$ の計算

まず各頂点 $v_j \in V_i$ に対して lb_j と rb_j を求める．そのために，$A^U(v_j) \neq \emptyset$ なる各頂点 v_j に対して重み付き上重心を計算するが，これは $O(|A^U(v_j)|)$ 時間で実行できる．$\sum_{j=1}^{n_i} |A^U(v_j)| = |E^U(i)|$ であるから，V_i の全頂点についての lb_j, rb_j の計算は $O(n_i + |E^U(i)|)$ 時間で実行できる．また，式 (3.20) に従って，$j = n_i, n_i - 1, \ldots, 1$ の順に $minr_j$ を計算し，その後式 (3.21) に従って，$j = 1, 2, \ldots, n_i$ の順に $maxr_j$ を求めることは，明らかに $O(n_i)$ 時間で実行できる．

次に，$A^U(v_j) \neq \emptyset$ なる各頂点 v_j に対して，以下の三つの値を計算する．

$$X_j = \sum_{w \in A^U(v_j)} \omega(v_j, w)^2, \tag{3.26}$$

$$Y_j = 2 \cdot \sum_{w \in A^U(v_j)} \omega(v_j, w)^2 \cdot x(w), \tag{3.27}$$

$$Z_j = \sum_{w \in A^U(v_j)} \omega(v_j, w)^2 \cdot x(w)^2. \tag{3.28}$$

これらの値が分かっていれば,

$$el^D(v_j, t) = X_j \cdot t^2 - Y_j \cdot t + Z_j \tag{3.29}$$

であるから，各整数 $t \in R(v_j)$ に対して $el^D(v_j, t)$ を $O(1)$ 時間で求めることができる．したがって，頂点 $v_j \in V_i$ と整数 $t \in R(v_j)$ のすべての組合せに対して $el^D(v_j, t)$ を求めることは，X_j, Y_j, Z_j の計算を含めて，$O(|E^U(i)| + \sum_{j=1}^{n_i} |R(v_j)|)$ 時間で可能である．

以上より次の補題を得る．

補題 3.2 各階層 V_i に対して，(a′) での処理は $O(|E^U(i)| + \sum_{j=1}^{n_i} |R(v_j)|)$ 時間で実行できる． □

3.3.2.3　$mel^D(v_j, t)$ の計算

図 3.2 に，V_i 上の全頂点 v_j とすべての $t \in R(v_j)$ に対して，$mel^D(v_j, t)$ と後の逆追跡で用いる値 $smin(v_j, t)$ を計算する手続き compMEL(i) を示す．この手続きでは，各 v_j と各 t に対して，

$$mm(v_j, t) = \min_{minr_j \leq k \leq t} mel^D(v_j, k)$$

なる値を計算し，この最小値を与える k の値を $pm(v_j,\ t)$ として記録している．

頂点 v_1 については，各 $t \in R(v_1)$ に対して，compMEL(i) の 2 行目で $mel^D(v_1, t)$ を求めている．各頂点 v_j（$j = 2, 3, \ldots, n_i$）については，14 行目において，式 (3.11) に従って $mel^D(v_j, t)$ を計算している．式 (3.11) における s としては，$s < t$ かつ $s \in R(v_{j-1})$ を満たす値だけを調べればよいので，$tt = \min\{t-1, maxr_{j-1}\}$ なる値をまず求め，$mel^D(v_j, t)$ の計算に $mm(v_{j-1}, tt)$ を，$smax(v_j, t)$ の決定に $pm(v_{j-1}, tt)$ を，それぞれ用いている．

compMEL(i) の 1〜10 行目は $O(|R(v_1)|)$ 時間で実行できる．また，各整数 $j = 2, 3, \ldots, n_i$ に対し，12〜23 行目に要する時間は $O(|R(v_j)|)$ である．よって，次の補題が成立する．

補題 3.3 各階層 V_i に対して，(b′) での処理は $O(\sum_{j=1}^{n_i} |R(v_j)|)$ 時間で実行できる． □

for $t = minr_1$ to $maxr_1$ **do**
  $mel^D(v_1, t) \leftarrow el^D(v_1, t)$;
  if $(t > minr_1)$ **and** $(mm(v_1, t-1) < mel^D(v_1, t))$ **then**
    $mm(v_1, t) \leftarrow mm(v_1, t-1)$;
    $pm(v_1, t) \leftarrow pm(v_1, t-1)$;
  else
    $mm(v_1, t) \leftarrow mel^D(v_1, t)$;
    $pm(v_1, t) \leftarrow t$;
  end if
end for
for $j = 2$ to n_i **do**
  for $t = minr_j$ to $maxr_j$ **do**
    $tt \leftarrow \min\{t-1, maxr_{j-1}\}$;
    $mel^D(v_j, t) \leftarrow mm(v_{j-1}, tt) + el^D(v_j, t)$;
    $smin(v_j, t) \leftarrow pm(v_{j-1}, tt)$;
    if $(t > minr_j)$ **and** $(mm(v_j, t-1) < mel^D(v_j, t))$ **then**
      $mm(v_j, t) \leftarrow mm(v_j, t-1)$;
      $pm(v_j, t) \leftarrow pm(v_j, t-1)$;
    else
      $mm(v_j, t) \leftarrow mel^D(v_j, t)$;
      $pm(v_j, t) \leftarrow t$;
    end if
  end for
end for

図 3.2: 手続き compMEL(i)

3.3.2.4 Down **過程及び** Up **過程の時間計算量**

以下では，Down 過程の実行時間を評価する．まず，次の補題が成立する．

補題 3.4 任意の階層 V_i（$i \geq 2$）とその任意の頂点 v_j について，$|R(v_j)| = O(n)$ である．

（証明）V_i の頂点を左から順に $v_1, v_2, \ldots, v_{n_i}$ とする．$minr_j = lb_j$ となる頂点 v_j のうち，添字が最小のものを v_k とする（$minr_{n_i} = lb_{n_i}$ であることより，このような添字 k は必ず存在する）．式 (3.19) より $minr_k \geq xmin_{i-1} - n_i + k$ である．もし $k > 1$ であれ

ば，$j = 1, 2, \ldots, k-1$ について $minr_j = minr_{j+1} - 1$ である．よって，k の値によらず

$$\begin{aligned} minr_1 &= minr_k - k + 1 \\ &\geq xmin_{i-1} - n_i + k - k + 1 \\ &= xmin_{i-1} - n_i + 1 \end{aligned} \tag{3.30}$$

が成立する．同様に考えれば,

$$maxr_{n_i} \leq xmax_{i-1} + n_i - 1 \tag{3.31}$$

が導かれる．よって，任意の $v_j \in V_i$ について,

$$\begin{aligned} |R(v_j)| &= maxr_j - minr_j + 1 \\ &\leq maxr_{n_i} - minr_1 + 1 \\ &\leq xmax_{i-1} - xmin_{i-1} + 2n_i - 1 \end{aligned} \tag{3.32}$$

が成立する．したがって，$xmax_{i-1} - xmin_{i-1} = O(n)$ であれば，$|R(v_j)| = O(n)$ が導かれる．

V_i の頂点配置の更新後を考えると，$x(v_1) \geq minr_1$, $x(v_{n_i}) \leq maxr_{n_i}$ となるから，式 (3.30), (3.31) より

$$xmin_i = x(v_1) \geq xmin_{i-1} - n_i + 1, \tag{3.33}$$

$$xmax_i = x(v_{n_i}) \leq xmax_{i-1} + n_i - 1 \tag{3.34}$$

である．これらの式より，$i = 2, 3, \ldots, h$ について

$$xmin_i \geq xmin_1 - \sum_{l=2}^{i} (n_l - 1), \tag{3.35}$$

$$xmax_i \leq xmax_1 + \sum_{l=2}^{i} (n_l - 1) \tag{3.36}$$

となる．$xmin_1 = 1$, $xmax_1 = n_1$ であるから,

$$xmax_i - xmin_i \leq n_1 - 1 + 2\sum_{l=2}^{i} (n_l - 1) \tag{3.37}$$

が成立する．よって，$i = 1, 2, \ldots, h$ について $xmax_i - xmin_i = O(n)$ であり，補題 3.4 が導かれる． □

系 3.1 Down 過程終了時における描画幅は $O(n)$ である．

（証明）$i = 2, 3, \ldots, h$ について式 (3.35) が成立することより,

$$\begin{aligned} \min_{2 \leq i \leq h} xmin_i &\geq xmin_1 - \sum_{l=2}^{h} (n_l - 1) \\ &> xmin_1 - n \end{aligned} \tag{3.38}$$

であるから，Down 過程終了時において，頂点の x 座標の最小値は $xmin_1 - n$ を下回らない．同様に，頂点の x 座標の最大値が $xmax_1 + n$ を上回らないので．Down 過程終了時の描画幅は $xmax_1 - xmin_1 + 2n$ を超えず，よって $O(n)$ である． □

補題 3.2〜3.4 を用いて，次の補題が証明できる．

補題 3.5 Down 過程を 1 回実行するのに要する時間は $O(n^2)$ である．

（証明）$i = 2, 3, \ldots, h$ に対し，前述の手順 (a$'$), (b$'$) の実行時間の和は，補題 3.2, 3.3 より $O(|E^U(i)| + \sum_{j=1}^{n_i} |R(v_j)|)$ である．逆追跡に要する時間はこの値を超えず，補題 3.4 より各 $|R(v_j)|$ は $O(n)$ であるから，階層 V_i に対する処理時間は $O(|E^U(i)| + n \cdot n_i)$ となる．Down 過程に要する時間は，この値の全階層についての和であるから $O(n^2)$ である．

□

Up 過程についての説明は省略するが，1 回の実行に要する時間は，Down 過程と同じく $O(n^2)$ である．また，Up 過程終了時における描画幅は $O(n)$ である．

3.3.3 BDown 過程と BUp 過程

BDown 過程では，$i = 2, 3, \ldots, h-1$ について，V_{i-1} と V_{i+1} の頂点座標を一旦固定して，V_i の頂点配置を更新する．Down 過程の場合と同様，各頂点 $v_j \in V_i$ に対して，重み付き重心に最も近い整数 lb_j^B, rb_j^B を計算し，式 (3.20), (3.21) と同様に定義される値 $minr_j^B, maxr_j^B$ を求める．そして，集合 $R^B(v_j)$ を $\{minr_j^B, minr_j^B + 1, \ldots, maxr_j^B\}$ とする．その後，各 v_j と各 $t \in R^B(v_j)$ に対して，v_j の x 座標を t としたときの

- $V_{i-1} \cup V_{i+1}$ の頂点と v_j を結ぶ辺の重み付き辺長の 2 乗の和 $el^B(v_j, t)$,
- $V_{i-1} \cup V_{i+1}$ の頂点と $v_1, v_2, \ldots, v_j$ を結ぶ辺の重み付き辺長の 2 乗の和の最小値 $mel^B(v_j, t)$

を計算する．v_j の上下両方の階層の隣接頂点を考慮することを除けば，これらの計算方法は **3.3.2** で述べたのと同様のものでよい．$\min_{t \in R^B(v_{n_i})} mel^B(v_{n_i}, t)$ の値を求めた後，逆追跡を行うことにより，$Lwel2^D(i) + Lwel2^U(i)$ の値を最小とするような V_i の頂点配置を決定する．その際，ある j, t に対して $mel^B(v_{j-1}, s)$ を最小とする s の値が複数存在するならば，それらのうち最大の値を選ぶ．

BDown 過程では，$i = h$ に関しては，Down 過程と同じ処理を行う．

BUp 過程における処理については，BDown 過程と同様であるので，説明を省略する．

Up 過程終了時に描画幅は $O(n)$ であること，及び，BDown 過程と BUp 過程を高々定数回しか実行しないことに留意すれば，次の補題は，**3.3.2** におけるのと同様の議論により証明できる（詳細は省略する）．

補題 3.6 BDown 過程と BUp 過程の 1 回の実行時間は $O(n^2)$ である．また，これらの繰返しが終了した時点の描画幅は $O(n)$ である． □

3.3.4 後処理

BDown 過程と BUp 過程がすべて終了した時点において，頂点の x 座標の最小値を $Xmin$，最大値を $Xmax$ とする．各階層 V_i（$i = 1, 2, \ldots, h$）に対応する水平線上で，x 座標が整数 k（$Xmin \leq k \leq Xmax$）である点を $p(i, k)$ と表す．点 $p(i, k)$ に G の頂点が配置されていないとき，それを空白点と呼ぶことにする．例えば，描画中に図 3.3(a) のような部分があったとすると，空白点は同図 (b) 中に白丸で示したようになる．

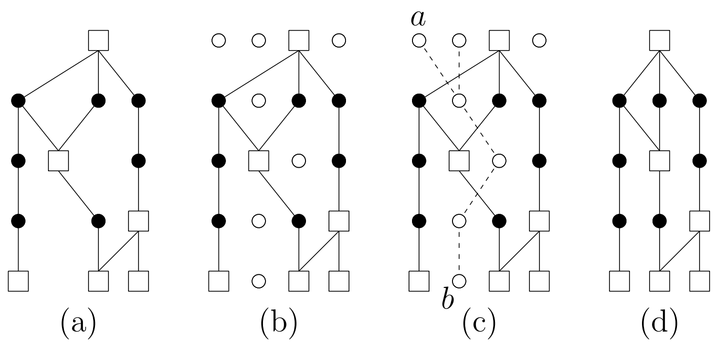


図 3.3: 後処理の実行例

すべての空白点からなる集合を頂点集合としたグラフ G_{sp} を考える．G_{sp} では，二つの空白点 $p(i, k), p(i', k')$ に対し，$|i - i'| \leq 1$，$|k - k'| \leq 1$ であるとき且つそのときに限り，辺を引くものとする．図 3.3 の例では，同図 (c) に破線で示した 5 本の辺ができることになる．

G_{sp} における，V_1 上の空白点から V_h 上の空白点への道で，それより左に G の頂点が存在するものを空白道と呼ぶ．提案法 1 の後処理では，最も左にある空白道を見つけ，それより左に存在する G のすべての頂点の x 座標を 1 大きくするという処理を，空白道がなくなるまで行う．例えば図 3.3(c) の場合，空白点 a から b への空白道を見つけ，その左にある全頂点を右に移動することにより，同図 (d) の描画を得る．この処理により，G 中のどの辺の辺長も増えることはなく，$WEL2$ の値が改善される．

空白道を見つけるために，BDown 過程と BUp 過程がすべて終了した直後の描画において，まず空白点の個数が最も少ない階層を一つ選び，V_{i^*} とする．この階層上の空白点を左から順に $p(i^*, k_1), p(i^*, k_2), \ldots, p(i^*, k_t)$ としたとき，$j = 1, 2, \ldots, t$ の順に，$p(i^*, k_j)$ を含む空白道を探す．そのために，$p(i^*, k_j)$ から出発して，まず上の階層に向けて空白点をたどっていく．その際，ある空白点から進むことのできる空白点が複数あれば，そ

れらのうち最も左にあるものを選択する．そして，V_1 上のある空白点まで到達すれば，$p(i^*, k_j)$ に戻り，下の階層に向けて同様の探索を行う．このようにして $p(i^*, k_j)$ を含む空白道を見つけることができれば，上述のように，その道より左にある全頂点の移動を行う．

空白道の探索を始める前に，すべての空白点 w に対して $mark(w) = -1$ とする．この -1 という値は，その空白点が未訪問であることを示す．V_{i^*} 上の空白点 $p(i^*, k_j)$ から上の階層に向けて探索をするとき，以下に述べる再帰的手続き $\mathrm{up}(w)$ を $w = p(i^*, k_j)$ として実行する．この手続き $\mathrm{up}(w)$ は，空白点 w から V_1 上の空白点まで到達できるか否かを判定するものであり，到達可能であれば $mark(w) = 1$ とし，到達不可能であれば $mark(w) = 0$ とする．$\mathrm{up}(w)$ で行う具体的な処理は以下のとおりである．なお，G_{sp} における w の隣接頂点で，w より一つ上の階層にあるものを，w の上隣接空白点と呼ぶことにする．

(a) w が V_1 上にあれば，$mark(w) = 1$ として，$\mathrm{up}(w)$ を終了する．w が V_1 上になければ (b) へ進む．

(b) w が上隣接空白点をもたなければ，$mark(w) = 0$ として，$\mathrm{up}(w)$ を終了する．w が上隣接空白点をもてば (c) へ進む．

(c) w の上隣接空白点を左から順に $z_1, \ldots, z_m$ （$1 \leq m \leq 3$）とする．$l = 1, \ldots, m$ の順に，$mark(z_l)$ が -1 であれば，G_{sp} における辺 (w, z_l) をたどって z_l を訪問し，$\mathrm{up}(z_l)$ を実行する．ただし，ある l' （$1 \leq l' < m$）に対して $mark(z_{l'}) = 1$ であることが分かれば，$\mathrm{up}(z_{l'+1})$ 以降を実行せずに，$mark(w)$ を 1 として，$\mathrm{up}(w)$ を終了する．また，すべての l （$1 \leq l \leq m$）に対して $mark(z_l) = 0$ となれば，$mark(w) = 0$ として，$\mathrm{up}(w)$ を終了する．

手続き $\mathrm{up}(w)$ は $mark(w) = -1$ のときのみ呼び出され，$\mathrm{up}(w)$ の終了後，$mark(w)$ は $1, 0$ のいずれかとなる．このことから，G_{sp} 中の同じ辺を 2 回以上たどらないこと，及び，同じ空白点 w に対して $\mathrm{up}(w)$ を 2 回以上実行しないことが分かる．V_{i^*} 上のある空白点 w に対して $mark(w) = 1$ となれば，w から $mark$ が 1 の上隣接空白点を順にたどることによって，V_1 上の空白点への道を得ることができる．

V_{i^*} 上の空白点から下の階層に向けて探索することも，$\mathrm{up}(w)$ と同様の手続きを考えれば実行できる．以上述べた後処理の実行時間に関して，次の補題が成立する．

補題 3.7 提案法 1 の後処理は $O(n^2)$ 時間で実行できる．

（証明）補題 3.6 より $Xmax - Xmin = O(n)$ であるから，空白点の個数は $O(n \cdot h)$ であり，これは $O(n^2)$ を超えない．G_{sp} において，各頂点（空白点）の次数は 6 以下であるから，G_{sp} の辺の本数も $O(n^2)$ である．

空白道を探す処理は，本質的に G_{sp}（の部分グラフ）を探索するものであるが，その際同じ辺を 2 度以上たどる必要はなく，よって，この処理に要する時間は後処理全体で $O(n^2)$ である．次に，G の頂点を移動する処理についてであるが，見つける空白道の本数はたかだか $O(n)$ 本であり，各空白道に対して頂点移動に要する時間も $O(n)$ であるから，この処理の実行時間も，後処理全体での和が $O(n^2)$ である．□

既に述べたように，提案法 1 における各過程の 1 回あたりの実行時間は $O(n^2)$ である．各過程の実行回数はたかだか定数回であり，また後処理の実行時間が $O(n^2)$ であるから，次の定理が成立する．

定理 3.2 提案法 1 は $O(n^2)$ 時間で実行可能である．□

3.4 計算機実験

以下の 3 通りの計算機実験により，優先度法と提案法 1 の比較を行った．
（実験 1-1）階層数が 2，頂点数が 20，辺の本数が 20, 40 あるいは 60 の連結グラフをランダムに 200 個ずつ作成した．各階層上の頂点順序は重心法 [1],[3] を用いて決定した．このような各データに対し，優先度法と提案法 1 のそれぞれを適用し，評価値 $EL2$ と $BAL2$ 及び実行時間を求めた．
（実験 1-2）階層数が 4，頂点の初期個数が 20，辺の初期本数が 20, 40 あるいは 60 の連結グラフをランダムに 200 個ずつ作成した（多階層グラフでは一般にダミー頂点が導入されるため，最終的な頂点数及び辺数はより多くなる）．実験 1-1 と同様，各階層上の頂点順序は重心法を用いて決定した．このような各データに対して，優先度法と提案法 1 のそれぞれを実行し，評価値 $EL2$，$DL2$，$BAL2$ 及び実行時間を求めた．
（実験 1-3）階層数が 8，頂点の初期個数が 40，辺の初期本数が 40, 80 あるいは 120 の連結グラフをランダムに 200 個ずつ作成し，実験 1-2 と同様の実験を行った．

3.3.1 で述べた定数 N の値は 5 とした．実験に用いた計算機の CPU は Core i7 870 (2.93GHz)，OS は Linux 2.6，プログラミング言語は Java 5.0 である．

実験結果を表 3.1 に示す．表中の各実験値は，200 個のデータに対する平均値である．

表 3.1 を見ると，2 階層グラフを用いた実験 1-1 では，優先度法に比べ実行時間が長くなっているものの，提案法 1 の方が $EL2, BAL2$ ともによい値を示している．多階層グラフを用いた実験 1-2, 1-3 でも，優先度法より実行時間は長くなっているが，評価値に関しては提案法 1 の方がよくなっている．特に，階層数がより多い実験 1-3 では，提案法 1 は，優先度法に比べ，3 種類すべての評価値を大幅に改善することができている．

階層数が 8，頂点数 40，辺の本数 40 の階層グラフに対し，優先度法による階層描画の例を図 3.4(a) に，提案法 1 により得られた階層描画の例を図 3.4(b) に示す．($EL2$，$DL2$，

$BAL2$) の値は図 3.4(a) では (271, 235, 107.90) であり，図 3.4(b) では (229, 144, 55.27) である．$EL2$，$DL2$, $BAL2$ ともに提案法 1 の方がよい値であることがわかる．

表 3.1: 優先度法との比較実験

(a) 実験 1-1

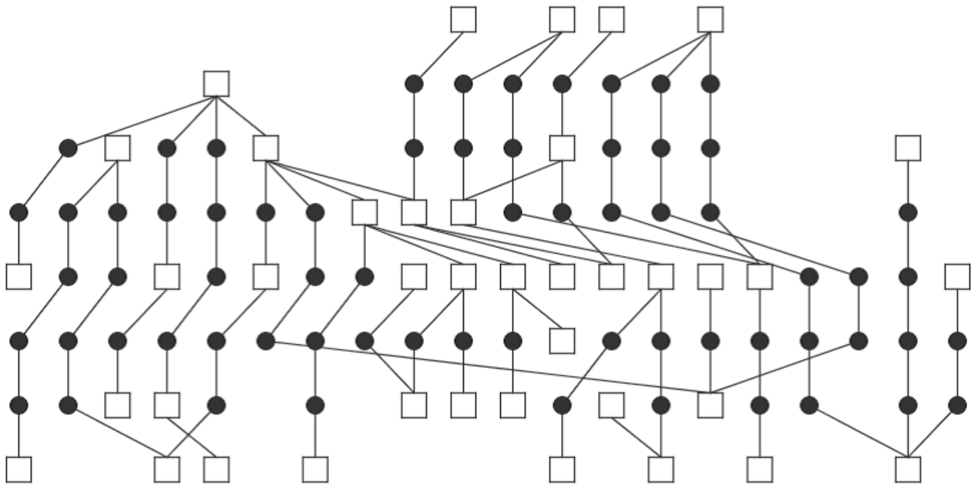
辺の本数	手 法	$EL2$	$BAL2$	時間 [ms]
20	優先度法	66.74	22.23	0.56
	提案法 1	61.59	18.01	1.35
40	優先度法	287.12	60.64	0.47
	提案法 1	268.91	51.84	0.77
60	優先度法	619.63	97.63	0.42
	提案法 1	584.81	85.85	0.83

(b) 実験 1-2

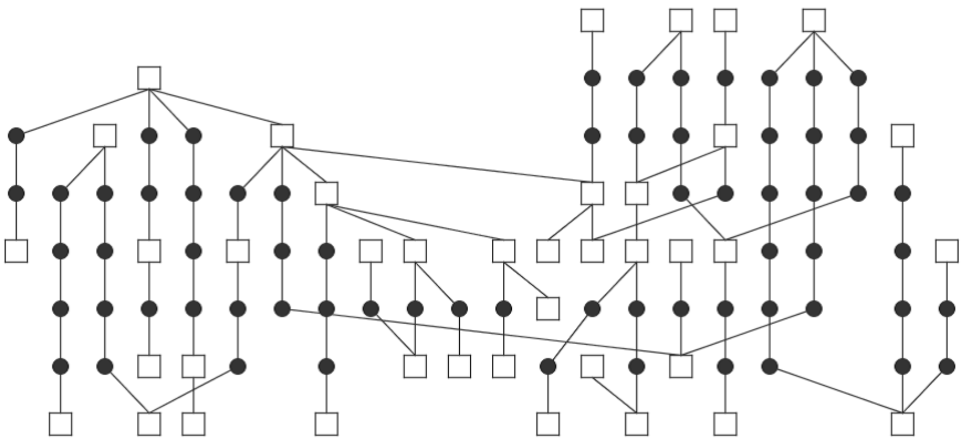
辺の本数	手 法	$EL2$	$DL2$	$BAL2$	時間 [ms]
20	優先度法	86.74	60.14	28.18	0.69
	提案法 1	80.55	44.16	21.40	2.70
40	優先度法	788.67	537.72	196.90	0.54
	提案法 1	759.98	434.61	141.33	3.01
60	優先度法	2552.84	1807.25	600.23	0.92
	提案法 1	2548.02	1578.11	459.56	5.85

(c) 実験 1-3

辺の本数	手 法	$EL2$	$DL2$	$BAL2$	時間 [ms]
40	優先度法	482.09	489.25	192.54	1.76
	提案法 1	338.52	266.25	85.21	6.03
80	優先度法	6677.14	6632.05	2090.99	4.54
	提案法 1	4891.26	3973.28	995.53	12.69
120	優先度法	26226.58	26333.16	7587.96	10.46
	提案法 1	19631.82	16043.82	3923.33	26.42



(a) 優先度法



(b) 提案法 1

図 3.4: 優先度法と提案法 1 の描画例

3.5 結言

本章では，各階層上の頂点の配置順序が指定された階層グラフを描画する際に，頂点の座標を決定する新しい方法を提案した．提案法 1 は，描画中の重み付き辺長の 2 乗の総和（$WEL2$）を小さくすることを目的としており，グラフの各階層を順に見ていき，それぞれの頂点の配置を動的計画法を用いて求めるものである．優先度法との比較実験を行ったところ，提案法 1 は，三つの評価基準 $EL2, DL2, BAL2$ のすべてに関して，より優れた描画を求めることができた．

提案法 1 では最小化したい目的関数を $WEL2$ としたが，各階層を順に見ていき，それぞれの頂点の配置を動的計画法を用いて求めるという方法は，描画中の辺長の総和を目的関数とした場合や，$\alpha \cdot EL2 + \beta \cdot DL2 + \gamma \cdot BAL2$ （α, β, γ：任意の定数）を目的関数とした場合などでも用いることができる [12, 13].

提案法 1 のさらなる高速化について検討することは今後の課題である．

第 4 章

階層グラフ描画におけるダミー頂点の共有

4.1　緒言

階層描画アルゴリズムでは，ダミー頂点の導入により，グラフ中の各辺が，連続した階層の頂点間を結ぶことになる．Sugiyama らの方法は，頂点座標の決定後，各辺を直線で描くが，このような描画を直線描画と呼ぶ．これに対し，各辺を垂直・水平線分からなる経路として描いた描画を直交描画と呼ぶ．直交描画を求める方法もいくつか提案されている．これらの方法では，同一頂点に接続する辺が垂直・水平線分の一部を共有することを許している．

直線描画では辺交差数が多くなるグラフでも，直交描画では辺交差数が大幅に少なくなることがある．これは，異なる辺に垂直・水平線分を共有させることにより，描画をより簡潔にできるためである．本章では，ダミー頂点についても，複数の辺に共有させることを考える．このような処理は，文献 [8] のにもみられるが，そこでの方法は，長辺を順に見ていき，先に作成したダミー頂点を利用できる場合に共有させるという極めて簡単なものであった．

ダミー頂点を複数の辺に共有させる処理は，直交描画のみならず，直線描画においても，描画を簡潔にするために有効であると考えられる．本章では，まずその有効性を示すため，ダミー頂点の共有処理を行う単純なアルゴリズムを提案し，計算機実験の結果を示す．描画をより簡潔にするためには，一般に，共有処理後のグラフの頂点数や辺数をできるだけ少なくすることが望ましいが，本論文では，頂点数を最小化する問題，及び辺数を最小化する問題が NP 困難であることも示す．

階層描画アルゴリズムに関する従来の研究の中に，辺のバンドリングと呼ぶ処理を行うものがある [14]．この処理は，ダミー頂点の座標を整数値に限定しないものとして，ある

条件を満たすダミー頂点を近づけて配置することにより，描画幅を削減するものである．この処理は，頂点間の隣接関係を変更することはなく，本章で述べるダミー頂点の共有化とは本質的に異なる．

本章の構成は以下の通りである．まず **4.2** においていくつかの定義をする．**4.3** では，ダミー頂点の共有処理を行うアルゴリズムを提案し，計算機実験の結果を示す．**4.4** と **4.5** では，それぞれ，頂点数を最小化する問題と辺数を最小化する問題が NP 困難であることを証明する．

4.2　ダミー頂点の共有化に関する問題の定義

通常，階層グラフ $G = (V, E)$ において，G のどの階層 V_i も空ではないので，$h \leq |V|$ である．よって，G の任意の辺のスパンは $|V| - 1$ 以下であり，通常の方法で作られるダミー頂点の総数は $(|V| - 2) \cdot |E|$ を超えない．また，各ダミー頂点の次数は 2 となり，各実頂点の次数はダミー頂点作成前と同じである．よって，頂点の次数の総和は $2(|V| - 2) \cdot |E| + 2|E|$ を超えず，ダミー頂点作成後の辺の本数が $(|V| - 1) \cdot |E|$ を超えないことが分かる．

ダミー頂点をもつ階層グラフにおいて，ある実頂点 v から出発し，一つ下の階層の隣接頂点をたどっていって別の実頂点 w に到達する道で，v, w 以外の実頂点を含まないものを，v から w へのダミー道と呼ぶことにする．本論文では，複数の辺がダミー頂点を共有することを許す．そして，ダミー頂点作成前の階層グラフ $G = (V, E)$ から，次の (A), (B) を満たす階層グラフ $G^* = (V^*, E^*)$ を作ることを考える．

(A) G^* におけるダミー頂点の集合を DS としたとき，$V^* = V \cup DS$ である．G^* も，G と同じく h 個の階層をもつ．G^* の階層を上から順に $V_1^*, V_2^*, \ldots, V_h^*$ としたとき，G において第 i 階層 V_i に属していた実頂点は，G^* においても第 i 階層 V_i^* に属する．

(B) G が辺 (v, w)（$v \in V_i,\ w \in V_j,\ i < j$）をもつとき，かつそのときに限り，$G^*$ において v から w へのダミー道が存在する．

このようなグラフ G^* を G の実現と呼ぶことにする．頂点数 $|V^*|$ がある自然数 K_V 以下の実現が存在するとき G は K_V 点実現可能であるといい，辺数 $|E^*|$ がある自然数 K_E 以下の実現が存在するとき K_E 辺実現可能であるという．G^* が条件 (B) を満たすことにより，G^* から G を一意に復元することができる．

図 4.1 に実現の例を示す．同図 (a) の 4 階層グラフに対し，通常の方法でダミー頂点を設けたのが同図 (b) である．このグラフは，(a) のグラフの実現の一つである．同図 (c) のような実現が存在するので，(a) のグラフは 12 点実現可能であり，15 辺実現可能である．

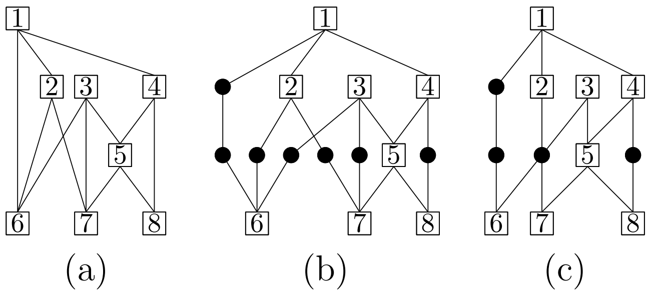


図 4.1: ダミー頂点の共有の例

本論文では，従来の多くの研究と同様，一般に階層描画では，辺の交差数が少ないこと，描画幅（頂点の x 座標の最大値と最小値の差）が小さいことなどが望まれる [2, 4]．描画対象の階層グラフ G が複雑で，辺交差数や描画幅を小さくすることができない場合，一つの方法として，G の適切な実現を作成し，それを描画することが考えられる．

頂点数や辺数を少なく抑えることは，一般に辺交差数と描画幅の削減につながる．よって，ダミー頂点の共有を許す場合，頂点数や辺数が少ない実現を求めることが望まれる．頂点数を最小化する問題と辺数を最小化する問題を判定問題にしたものを次に示す．

[問題 MNV]

入力： 階層グラフ $G = (V, E)$ と自然数 K_V．ただし $|V| \leq K_V \leq |V| + (|V| - 2) \cdot |E|$ とする．

問題： G は K_V 点実現可能か． □

[問題 MNE]

入力： 階層グラフ $G = (V, E)$ と自然数 K_E．ただし $|E| \leq K_E \leq (|V| - 1) \cdot |E|$ とする．

問題： G は K_E 辺実現可能か． □

4.3 ダミー頂点の共有を行うアルゴリズム

本節では，ダミー頂点の共有を行う単純なアルゴリズムを提案し，計算機実験の結果を示す．この方法は，問題 MNV 及び MNE に対する発見的手法として用いることができる．

4.3.1 提案法 2

ダミー頂点をもたない h 階層グラフ $G=(V,E)$ が与えられたとき，提案法 2 は，ダミー頂点共有処理を 2 段階に分けて行う．

第一段階の処理を図 4.2(a) に示す．ここでは，上の階層から順に実頂点 v を見ていき，v から下の階層の実頂点へ向かう全ての辺を，図 4.3(a) に示すように，ダミー頂点を含む一つの木で置き換える．この木に含まれるすべての辺のスパンは 1 になる．

$i=1,2,\ldots,h-2$ に対して次の (1) を実行する．

(1) 各実頂点 $v \in V_i$ に対して，次の (1a) を実行する．

(1a) $V_{i+2} \cup V_{i+3} \cup \cdots \cup V_h$ に存在し，v に隣接する実頂点を，最も下の階層にあるものから順に $v_1, v_2, \ldots, v_p$ とする．$p \geq 1$ であり，且つ辺 (v, v_1) のスパン t が 2 以上であれば，次の (1a1), (1a2) を実行する．

(1a1) ダミー頂点 $d_1^v, d_2^v, \ldots, d_{t-1}^v$ を設け，各 d_k^v を第 $i+k$ 階層に置く．辺 (v, v_1) を道 $[v, d_1^v, d_2^v, \ldots, d_{t-1}^v, v_1]$ で置き換える．

(1a2) $j=2,3,\ldots,p$ に対して次の $(*)$ を実行する．

$(*)$ v_j が存在する階層を V_{i+q} としたとき，$q \geq 2$ ならば，辺 (v, v_j) を (d_{q-1}^v, v_j) で置き換える．

(a) 第一段階

$i=h-1,h-2,\ldots,2$ に対して次の (2) を実行する．

(2) 第 i 階層に，$A^D(d) \subseteq A^D(d')$ かつ $|A^D(d)| \geq |A^U(d')|$ であるようなダミー頂点 d, d' が存在しなくなるまで，次の処理 (2a)〜(2d) を実行する．

(2a) $A^D(d) \subseteq A^D(d')$ かつ $|A^D(d)| \geq |A^U(d')|$ であるダミー頂点 d, d' を求める．ただし，条件を満たす頂点対が複数あるならば，$|A^D(d)|$ が最大となる d, d' を任意に選ぶ．

(2b) 各頂点 $u \in A^U(d')$ に対して，辺 (u, d) を加える．

(2c) 各頂点 $v \in A^D(d)$ に対して，辺 (d', v) を削除する．

(2d) $A^D(d') = \emptyset$ となれば，d' を削除する．

(b) 第二段階

図 4.2: 提案法 2

第一段階終了後について，各頂点 v に対し，第 2 章 **3.2** で定義したように一つ上の階層にある隣接頂点の集合を $A^U(v)$，一つ下の階層にある隣接頂点の集合を $A^D(v)$

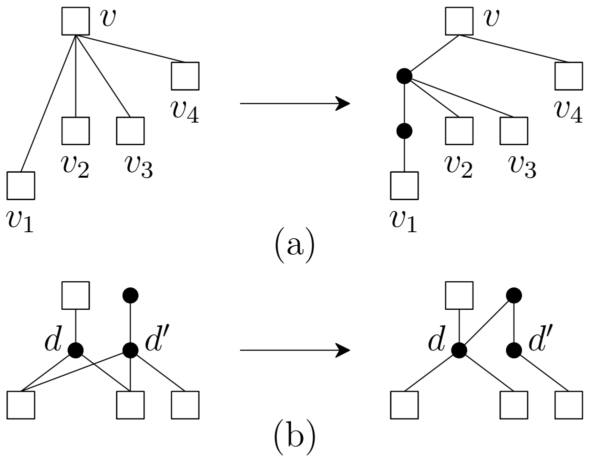


図 4.3: 提案法 2 でのダミー頂点共有

と表す．第二段階の処理を図 4.2(b) に示す．ここでは，下の階層から順に見ていき，$A^D(d) \subseteq A^D(d')$ かつ $|A^D(d)| \geq |A^U(d')|$ であるダミー頂点 d, d' が存在すれば，図 4.3(b) に示すような辺のつなぎ換えを行う．ここで，$|A^D(d)| \geq |A^U(d')|$ という条件は，つなぎ換えにより辺数が増えないようにするためのものである．

図 4.4(a) は，通常の方法で作られるダミー頂点の例を示したものである．この場合，第一段階を実行すると，同図 (b) のようにダミー頂点は 4 個になる．また，第二段階を実行すると，同図 (c) に示すように，辺数がさらに減り，ダミー頂点も 3 個だけになる．

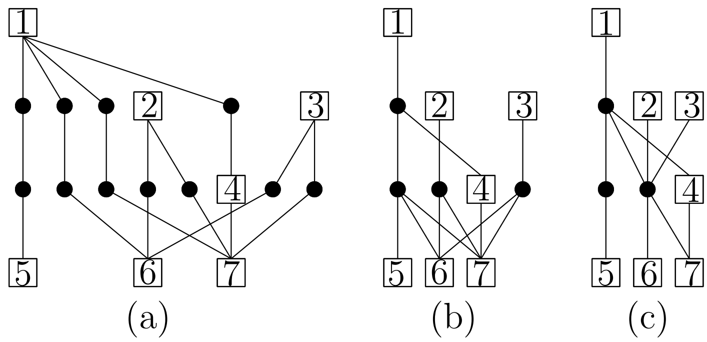


図 4.4: 提案法 2 の実行例

4.3.2 計算機実験

以下の 2 通りの計算機実験を行った．

（実験 2-1）階層数 h が 4，頂点の初期個数 $|V|$ が 20，辺の初期本数 $|E|$ が 20, 40 あるいは 60 の連結グラフ (V, E) をランダムに 200 個ずつ作成した．そして，通常の方法でダミー頂点を作成した場合と，提案法 2 を実行した場合について，頂点数，辺数，同一階層上の頂点数の最大値 $maxvi$，及び実行時間を求めた．また，$h = 8$，$|V| = 40$，$|E| = 40, 80$ あるいは 120 の連結グラフ各 200 個に対しても，同様の実験を行った．

（実験 2-2）実験 2-1 と同じデータに対して，以下の二つの方法によってグラフ描画を作成し，描画幅，辺交差数及び実行時間を求めた．

方法 a： 通常の方法でダミー頂点を作成した後，重心法 [2, 3] を実行し，さらに文献 [15] の辺交差削減法を実行して各階層上の頂点配置順序を決定する．次に，第 3 章で述べたの頂点座標決定アルゴリズムを用いて，各頂点の x 座標を決定する．最後に，各辺を直線で描く．

方法 b： 提案法 2 を実行した後，方法 a と同様の手順によって，各階層上の頂点順序の決定，各頂点の座標の決定を行って，直線描画を求める．

使用計算機の CPU は Core i7 870（2.93GHz），OS は Linux 2.6，プログラミング言語は Java 5.0 である．

実験結果を表 4.1 に示す．実験結果の値は全てグラフ 200 個に対する平均値である．実験 2-1 の結果を見ると，どの場合についても，提案法 2 を実行した場合の頂点数，辺数及び $maxvi$ の値は，通常の方法でダミー頂点を作成した場合に比べ，大幅に小さくなっている．また実験 2-2 の結果を見ると，提案法 2 を実行している方法 b では，$maxvi$ の減少に伴い，描画幅が方法 a より大幅に小さくなっている（描画幅は $maxvi - 1$ 以上になる）．同様に，方法 b による描画の辺交差数は，方法 a に比べてかなり小さくなっている．

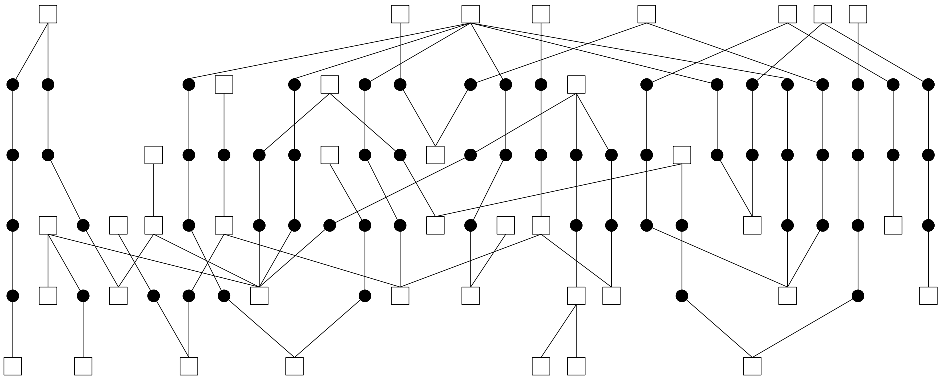
表 4.1: ダミー頂点の共有化アルゴリズムに関する実験結果

(a) 実験 2-1

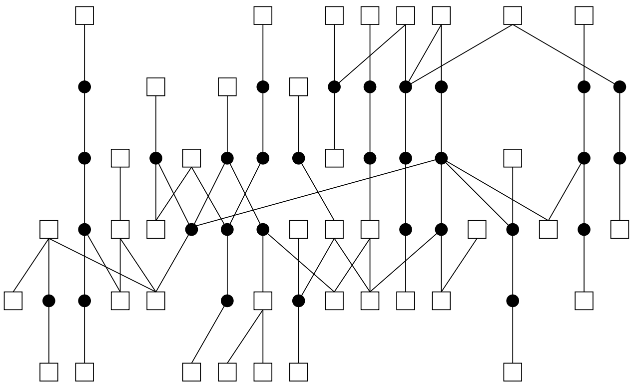
h	$\|V\|$	$\|E\|$	手法	頂点数	辺数	$maxvi$	時間 [ms]
4	20	20	通常の方法	33.34	33.34	12.96	0.02
			提案法 2	27.31	27.29	9.82	0.10
4	20	40	通常の方法	46.66	66.66	19.93	0.05
			提案法 2	29.79	48.34	11.14	0.11
4	20	60	通常の方法	61.09	101.09	27.35	0.05
			提案法 2	30.82	65.70	11.81	0.15
8	40	40	通常の方法	119.07	119.07	24.01	0.08
			提案法 2	83.16	83.13	16.09	0.58
8	40	80	通常の方法	199.78	239.78	41.42	0.21
			提案法 2	96.88	133.72	19.81	1.72
8	40	120	通常の方法	281.84	361.84	58.98	0.42
			提案法 2	105.37	176.19	22.27	2.95

(b) 実験 2-2

h	$\|V\|$	$\|E\|$	手 法	描画幅	辺交差数	時間 [ms]
4	20	20	方法 a	12.43	5.59	3.52
			方法 b	9.36	4.67	2.22
4	20	40	方法 a	19.25	70.17	13.36
			方法 b	10.25	55.39	3.64
4	20	60	方法 a	26.65	213.37	35.27
			方法 b	10.85	138.59	4.71
8	40	40	方法 a	24.50	24.00	20.88
			方法 b	16.44	16.34	9.91
8	40	80	方法 a	42.05	270.08	146.34
			方法 b	19.16	164.99	23.34
8	40	120	方法 a	60.26	780.26	490.43
			方法 b	21.46	393.08	37.06



(a) 方法 a



(b) 方法 b

図 4.5: 描画例

$|V| = 40, |E| = 40$ の 6 階層グラフに対し，方法 a,b を実行して得られた描画の例を図 4.5 に示す．描画幅は，方法 a で 26，方法 b で 17 であった．また，辺交差数は，方法 a で 46，方法 b で 20 であった．

以上の実験結果より，ダミー頂点の共有処理は，階層描画の描画幅と辺交差数を削減するために有効であることが分かる．

4.4　問題 MNV の NP 完全性

本節では，**2.2** で示した問題 MNV が NP 完全であることを証明する．この問題が NP に属することを示すのは容易である．そのためには，MNV に対する非決定性多項式時間アルゴリズムとして，ダミー頂点の集合とそれらの階層，及び各頂点の隣接頂点の集合を推測し，前述の条件 (A), (B) を満たすこと，及び頂点数が K_V 以下であることを確認するような方法を考えればよい．

次の問題は NP 完全であることが知られている [16].

[問題 SET BASIS]

入力： 有限集合 $S = \{s_1, s_2, \ldots, s_n\}$ とその部分集合の族 $C = \{c_1, c_2, \ldots, c_m\}$ 及び自然数 $k_{SB} \leq m$.

問題： S の部分集合の族 B で，次の (i), (ii) を満たすものが存在するか.

(i) $|B| = k_{SB}$.

(ii) 任意の $c_j \in C$ に対して, B のある部分集合で，その和集合が c_j に等しくなるものが存在する. □

例えば

$$S = \{a, b, c, d, e, f\},$$
$$C = \{\{a, b, e\}, \{a, b, c, d, e\}, \{c, d, e\}, \{a, b, f\}, \{e, f\}\}$$

であり，$k_{SB} = 4$ であるものとすると，

$$B = \{\{a, b\}, \{c, d\}, \{e\}, \{f\}\}$$

が条件 (i), (ii) を満たすから，上記の問題に対する答えは “yes” となる.

以下，問題 SET BASIS から MNV への変換 f_V を示す．SET BASIS の入力の S, C, k_{SB} から，MNV の入力として，3 階層連結グラフ $G = (V, E)$ と自然数 K_V を作る．G の第 1〜3 階層の頂点の集合を，それぞれ，V_1, V_2, V_3 と表す.

まず，各 $s_i \in S$ に対応して頂点を作り，それを V_3 の要素とする．次に，各 $c_j \in C$ に対応して頂点を作り，それを V_1 の要素とする．簡単のため，s_i, c_j に対応する頂点も，それぞれ，s_i, c_j と表すことにする．V_2 は 1 頂点 Y のみを含むものとする.

$$V = V_1 \ \cup \ V_2 \ \cup \ V_3, \tag{4.1}$$
$$V_1 = \{c_j \mid c_j \in C\}, \tag{4.2}$$
$$V_2 = \{Y\}, \tag{4.3}$$
$$V_3 = \{s_i \mid s_i \in S\} \tag{4.4}$$

である.

各 $c_j \in C$ に対し，その要素に対応する V_3 の頂点と頂点 $c_j \in V_1$ との間に辺を加える．また，頂点 Y と V_3 の全頂点との間にも辺を加える.

$$E = \bigcup_{j=1}^{m} \{(c_j, s_i) \mid s_i \in c_j\} \cup \{(Y, v) \mid v \in V_3\} \tag{4.5}$$

である．頂点 Y 及びそれに接続する辺は，$V_2 \neq \emptyset$ とし，G の連結性を保証するためのものである.

最後に，自然数 K_V を次のように定める．

$$K_V = |V| + k_{SB} = n + m + 1 + k_{SB}. \tag{4.6}$$

先に示した S, C の例に対しては，G は図 4.6(a) のようになる．また $k_{SB} = 4$ であれば，$K_V = 16$ となる．

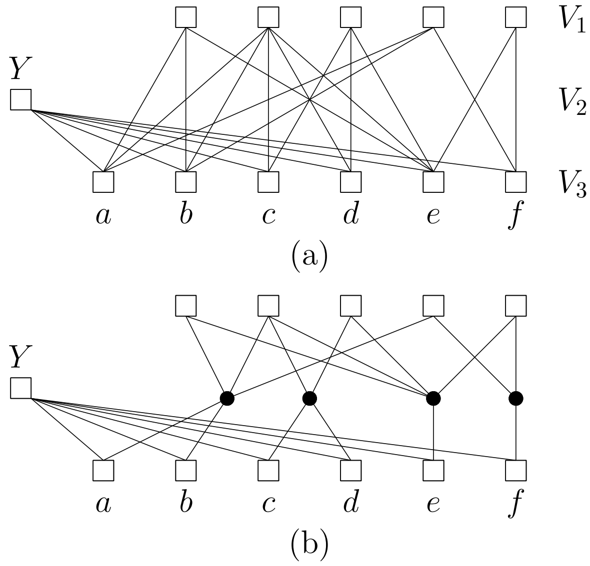


図 4.6: SET BASIS から MNV への変換の例

次の定理が成立する．

定理 4.1 問題 MNV は NP 完全である．

（証明）$|V| = n + m + 1$，$|E| \leq n \cdot (m + 1)$ であるから，SET BASIS から MNV への変換 f_V は，n, m に関する多項式時間で実行可能である．

S, C, k_{SB} に対して，前述の条件 (i), (ii) を満たす集合族 $B = \{b_1, b_2, \ldots, b_{k_{SB}}\}$ が存在するものとする．このとき，各集合 $b_l \in B$ に対応したダミー頂点を作り，それらと V を併せたものを V^* とする．次に，各 b_l について，対応するダミー頂点と $s_i \in b_l$ なる各頂点 $s_i \in V_3$ とを辺でつなぐ．また，各 c_j に対して，$c_j = \cup_{b_l \in B_j} b_l$ となる集合 $B_j \subseteq B$ が存在するので，B_j の各要素 b_l に対応するダミー頂点と頂点 $c_j \in V_1$ との間に辺を引く．これらの辺に，Y に接続する n 本の辺を加えたものを E^* とする．明らかに，グラフ (V^*, E^*) は G の実現であり，$|V^*| = |V| + k_{SB} = K_V$ である（図 4.6(b) に例を示す）．よって，S, C, k_{SB} に対して条件 (i), (ii) を満たす集合族 B が存在するならば，G は K_V 点実現可能である．

逆に，G が K_V 点実現可能であるときに，S, C, k_{SB} に対して条件 (i), (ii) を満たす集合族 B が存在することも容易に示すことができる．したがって，SET BASIS は MNV

に多項式変換可能である．

前述のように，SET BASIS は NP 完全である．さらに，MNV は NP に属するから，MNV は NP 完全である． □

上記の変換 f_V より，問題 MNV は，入力のグラフ G を連結な 3 階層グラフに限定しても NP 完全である．

4.5 問題 MNE の NP 完全性

本節では，問題 MNE が NP 完全であることを証明する．MNV と同様，問題 MNE が NP に属することを示すのは容易であるので，説明を省略する．

代表的な NP 完全問題の一つとして頂点被覆問題が知られている [16]．次の問題 VC は，通常の頂点被覆問題の入力にある制約を加えたものである．

[問題 VC]

入力： どの頂点の次数も 2 以上である連結単純グラフ $G_H = (V(H), E(H))$ と自然数 k_{VC}（$\leq |V(H)|$）．

問題： G_H は要素数 k_{VC} 以下の頂点被覆をもつか． □

頂点被覆問題の NP 完全性の証明は文献 [16] に示されている．それを若干変更することによって，VC の NP 完全性も示すことができる（詳細は省略する）．

以下，問題 VC から MNE への変換 f_E を示す．グラフ G_H において，$V(H) = \{v_1, v_2, \cdots, v_n\}$，$E(H) = \{e_1, e_2, \cdots, e_m\}$ であるものとする．G_H と自然数 k_{VC} から，MNE の入力として，3 階層グラフ $G = (V, E)$ と自然数 K_E を作る．G の第 1〜3 階層の頂点の集合を，それぞれ，V_1, V_2, V_3 と表す．

まず，各頂点 $v_i \in V(H)$ に対応して頂点を作り，それを V_3 の要素とする．V_3 には，これらとは異なる頂点 X も加える．次に，各辺 $e_j \in E(H)$ に対応して頂点を作り，それを V_1 の要素とする．簡単のため，v_i, e_j に対応する（G の）頂点も，それぞれ，v_i, e_j と表すことにする．V_2 は 1 頂点 Y だけを含むものとする．

$$V = V_1 \ \cup \ V_2 \ \cup \ V_3, \tag{4.7}$$
$$V_1 = \{e_j \mid e_j \in E(H)\}, \tag{4.8}$$
$$V_2 = \{Y\}, \tag{4.9}$$
$$V_3 = \{v_i \mid v_i \in V(H)\} \cup \{X\} \tag{4.10}$$

である．

G_H において，各辺 e_j の端点を a_j, b_j とする．このとき，e_j に対し，3 本の辺

(e_j, a_j), (e_j, b_j), (e_j, X) からなる集合 $E^{(j)}$ を考える．G の辺集合 E は

$$E = \bigcup_{j=1}^{m} E^{(j)} \ \cup \ \{(Y, v) \mid v \in V_3\} \tag{4.11}$$

とする．最後に，自然数 K_E を次のように定める．

$$K_E = 2m + 2n + k_{VC} + 1. \tag{4.12}$$

例えば, 図 4.7(a) のグラフを G_H とすると, G は同図 (b) のようになる．また, $k_{VC} = 2$ であれば $K_E = 21$ となる．

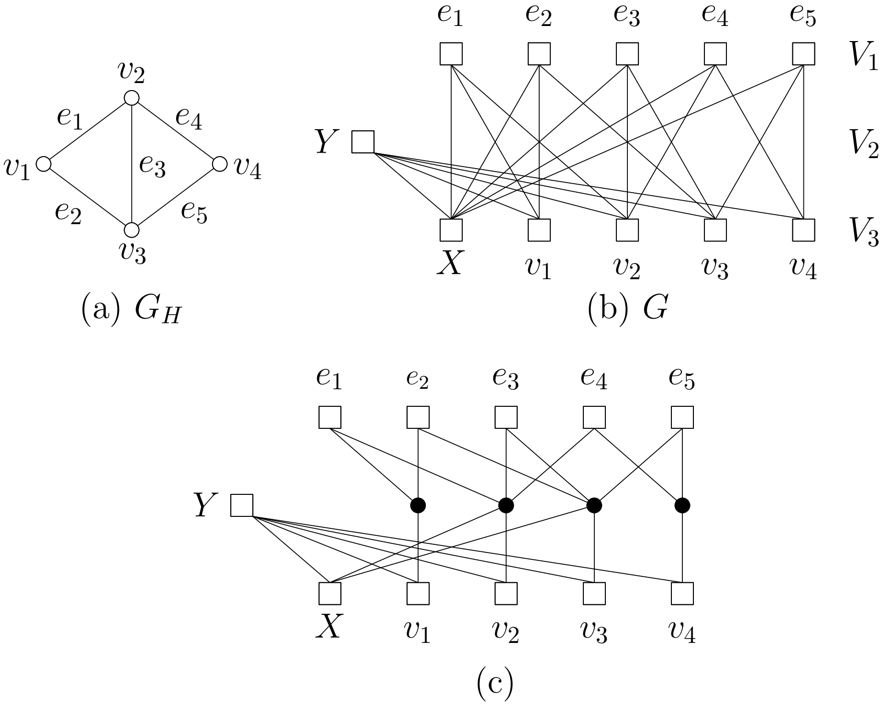


図 4.7: VC から MNE への変換の例

$|V| = n + m + 2$, $|E| = 3m + n + 1$ であるから，上記の変換 f_E は，n, m に関する多項式時間で実行可能である．次の補題が成立する．

補題 4.1 グラフ G_H が要素数 k_{VC} 以下の頂点被覆をもつならば，G は K_E 辺実現可能である．

（証明）G_H が要素数 k_{VC} 以下の頂点被覆 CV をもつものとする．G に対し，X 以外の各頂点 $v_i \in V_3$ に対応してダミー頂点 $d(v_i)$ を作る．G_H の各辺 $e_j = (a_j, b_j)$ に対して，G の各頂点 $e_j \in V_1$ を $d(a_j)$ 及び $d(b_j)$ とつなぐ．各ダミー頂点 $d(v_i)$ は $v_i \in V_3$ とつなぎ，さらに $v_i \in CV$ であれば $X \in V_3$ ともつなぐ．CV は G_H の頂点被覆であるから，

このようにして得られるグラフが G の実現であることは明らかである．辺数は,

$$2m + n + |CV| + n + 1 \leq K_E \tag{4.13}$$

であるから，G は K_E 辺実現可能である． □

例えば，$k_{VC} = 2$ とすると，図 4.7(a) のグラフ G_H は要素数 k_{VC} の頂点被覆 $\{v_2, v_3\}$ をもつ．これに対応して，補題 4.1 証明で述べた方法によりダミー頂点と辺を定めれば，図 4.7(c) のようになり，辺を $K_E = 21$ 本にすることができる．

以下では，G が K_E 辺実現可能であるときに，G_H が要素数 k_{VC} 以下の頂点被覆をもつことを示す．G の任意の実現において，第 2 階層に作られた各ダミー頂点 d に対し，連続する第 3 階層の頂点の集合を A_d と表す．また，V_1 の各頂点 e_j に対して，それに隣接するダミー頂点のことを，e_j がもつダミー頂点と呼ぶ．以下，G が K_E 辺実現可能であるときに，次の条件 1〜4 を満たして K_E 辺実現可能であることを順に示す．

(条件 1)　V_1 の任意の頂点 e_j がダミー頂点 d, d' をもつとき，$A_d, A_{d'}$ のどちらも他方を包含していない．

(条件 2)　V_1 のどの頂点 e_j も三つ以上のダミー頂点をもたない．

(条件 3)　V_1 の各頂点 e_j が二つのダミー頂点をもつ．

(条件 4)　G_H の任意の辺 $e_j = (a_j, b_j)$ に対し，頂点 $e_j \in V_1$ が $A_d = \{a_j, b_j\}$ となるダミー頂点をもたない．

補題 4.2 グラフ G が K_E 辺実現可能であれば，条件 1 を満たして K_E 辺実現可能である．

（証明）辺数が K_E 以下の G の任意の実現を考える．ある頂点 $e_j = (a_j, b_j) \in V_1$ が $A_d \supseteq A_{d'}$ なるダミー頂点 d, d' をもつとき，**2.2** で述べた条件 (A), (B) に違反することなく，辺 (e_j, d') を削除することができる（図 4.8 に一例を示す）．条件 1 に違反する頂点 $e_j \in V_1$ が存在する限りこの処理を繰返せば，辺数を増やすことなく，条件 1 を満たした実現が得られる． □

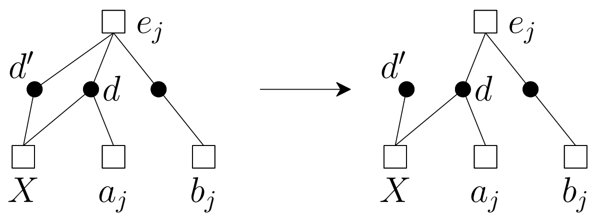


図 4.8: 補題 4.2 証明に対する例

補題 4.3 グラフ G が K_E 辺実現可能であれば，条件 1, 2 を満たして K_E 辺実現可能である．

（証明）辺数が K_E 以下で，条件 1 を満たした任意の実現を考える．V_1 のどの頂点 $e_j = (a_j, b_j)$ も四つ以上のダミー頂点をもっていないことは容易に確認することができる．ある頂点 e_j が三つのダミー頂点をもつとき，条件 1 より，それらに接続する辺の状態は図 4.9(a) 左, (b) 左のいずれかである．それぞれについて，図中の右側のように変更を行う．これらの変更によって，辺数を増やさず，かつ条件 (B) に違反することなく，条件 2 に対する e_j での違反をなくすことができる．このような処理を必要に応じて繰返すことにより，条件 1, 2 を満たした実現が得られる． □

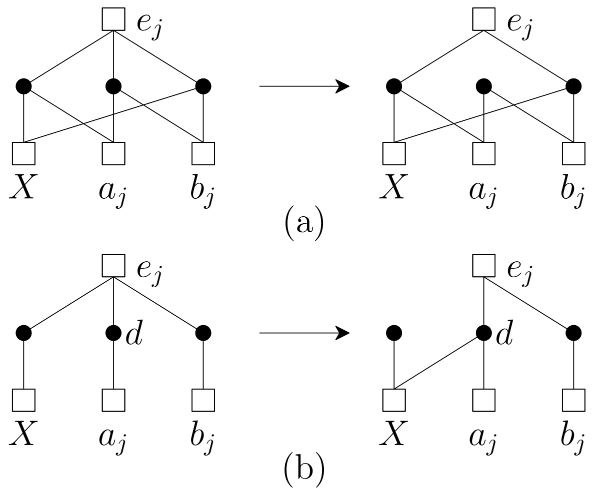


図 4.9: 補題 4.3 証明の説明図

補題 4.4 グラフ G が K_E 辺実現可能であれば，条件 1〜3 を満たして K_E 辺実現可能である．

（証明）辺数が K_E 以下で，条件 1, 2 を満たした任意の実現 RL を考える．RL において，頂点 $e_j = (a_j, b_j) \in V_1$ が，ただ一つのダミー頂点 d をもつものとする．$A_d = \{a_j, b_j, X\}$ である．

G_H では任意の頂点の次数が 2 以上であったため，頂点 a_j に接続する別の辺 $e_k = (a_j, b_k)$ が存在する．G_H は単純であったので，$b_j \neq b_k$ である．RL において，頂点 e_k がもつダミー頂点の状態は次のいずれかである．

(i) e_k は二つのダミー頂点をもち，それらの一方 d' に対して $A_{d'}$ が $\{a_j\}$ あるいは $\{a_j, X\}$ に等しい．

(ii) e_k は二つのダミー頂点をもつ．それらの一方 d' に対して $A_{d'} = \{a_j, b_k\}$ であり，もう一方 d'' に対して $A_{d''}$ は $\{X\}$ あるいは $\{b_k, X\}$ である．

(iii) e_k は一つのダミー頂点 d' のみをもち，$A_{d'} = \{a_j, b_k, X\}$ である．

(i) の場合，図 4.10(a), (b) のように，e_j と d' の間に辺を加え，$A_{d'}$ の頂点と d との間の辺を削除する．(ii) の場合，図 4.10(c) のように，新しいダミー頂点 dd を設け，それを頂点 e_j, e_k, a_j 及び X につなぐ．そして，辺 $(d, a_j), (d, X), (d', a_j)$ 及び (e_k, d'') を削除する．(iii) の場合，(ii) の場合と同様，新しいダミー頂点 dd を設け，それを頂点 e_j, e_k, a_j 及び X につないだ後，d, d' と a_j, X をつないでいた辺を削除する．図 4.10(d) 参照．G_H が多重辺をもたず，かつ RL が条件 (B) を満たしていたことより，(i) の場合 d に隣接していた V_1 の頂点は e_j のみであり，(ii), (iii) の場合 d, d' に隣接していた V_1 の頂点は e_j, e_k のみである．したがって，上記のようにしても条件 (B) に違反することはない．

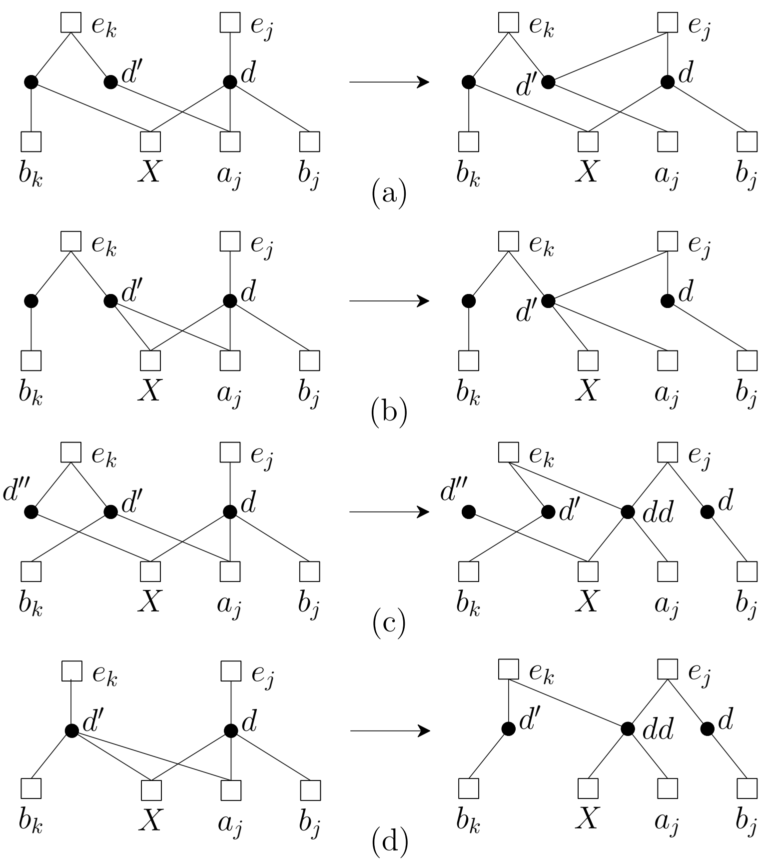


図 4.10: 補題 4.4 証明の説明図

(i)〜(iii) のいずれの場合も，辺数を増やすことなく，条件 1, 2 を満たしたまま，条件 3 に対する e_j での違反をなくすことができる．したがって，このような処理を必要に応じて繰返すことにより，条件 1〜3 を満たした実現が得られる． □

補題 4.5 グラフ G が K_E 辺実現可能であれば，条件 1〜4 を満たして K_E 辺実現可能である．

（証明）辺数が K_E 以下で，条件 1〜3 を満たした任意の実現 RL を考える．RL において，頂点 $e_j = (a_j, b_j) \in V_1$ が $A_d = \{a_j, b_j\}$ なるダミー頂点 d をもつものとする．条件 3 より，e_j はもう一つのダミー頂点 d' をもつ．A_d と $A_{d'}$ の間に包含関係がないから，$A_{d'}$ は $\{a_j, X\}$, $\{b_j, X\}$, $\{X\}$ のいずれかである．

$A_{d'} = \{a_j, X\}$ であれば，図 4.11(a) に示すように辺 (d, a_j) を削除する，また $A_{d'} = \{b_j, X\}$ であれば辺 (d, b_j) を削除する．d に隣接していた V_1 の頂点は e_j だけであるから，いずれの場合も条件 (B) に違反することはない．よって，$A_{d'}$ が $\{a_j, X\}$ あるいは $\{b_j, X\}$ であれば，条件 1〜3 を満たしたまま，かつ辺数を増やすことなく，条件 4 に対する e_j での違反をなくすことができる．

$A_{d'} = \{X\}$ であるものとする．グラフ G_H では，頂点 a_j に接続する別の辺 $e_k = (a_j, b_k)$ が存在する．RL において，頂点 e_k がもつ二つのダミー頂点のうち，頂点 a_j に隣接しているものを d'' とする（e_k がもつ二つのダミー頂点が共に a_j に隣接している場合には，任意に一方を選ぶ）．$A_{d''}$ は，$\{a_j\}, \{a_j, X\}, \{a_j, b_k\}$ のいずれかである．$A_{d''} = \{a_j\}$ の場合，図 4.11(b) のように，d'' を頂点 e_j, X とつなぎ，辺 (e_j, d') 及び (d, a_j) を削除する．また，$A_{d''} = \{a_j, X\}$ の場合，図 4.11(c) のように，d'' を e_j とつなぎ，やはり辺 (e_j, d') 及び (d, a_j) を削除する．これらの場合，辺 (d, a_j) を削除しているが，d に隣接していた V_1 の頂点は e_j のみであるから，条件 (B) に違反することはない．一方，$A_{d''} = \{a_j, b_k\}$ の場合には，新しいダミー頂点 dd を作る．そして，辺 (e_k, d') が存在していなかった場合には図 4.11(d)，存在していた場合には同図 (e) のように辺のつなぎ換えを行う（前者の場合，頂点 d'' の削除も行う）．いずれの場合も d, d'' に隣接していた V_1 の頂点は e_j, e_k のみであるから，このような処理によって条件 (B) に違反することはない．以上より，$A_{d''}$ が $\{a_j\}, \{a_j, X\}, \{a_j, b_k\}$ のいずれであった場合も，条件 1〜3 を満たしたまま，辺数を増やすことなく，条件 4 に対する e_j での違反をなくすことができる．

実現 RL に対し，以上述べた処理を必要に応じて実行することにより，条件 1〜4 を満たした実現が得られる． □

補題 4.5 より次の補題を得る．

補題 4.6 グラフ G が K_E 辺実現可能ならば，グラフ G_H は要素数 k_{VC} 以下の頂点被覆をもつ．

（証明）グラフ G が K_E 辺実現可能ならば，補題 5 より，条件 1〜4 を満たして K_E 辺実現可能である．そのような任意の実現において，V_1 の頂点に隣接していないダミー頂点をすべて削除したものを RL とする．明らかに RL も G の実現であり，辺数は K_E 以下

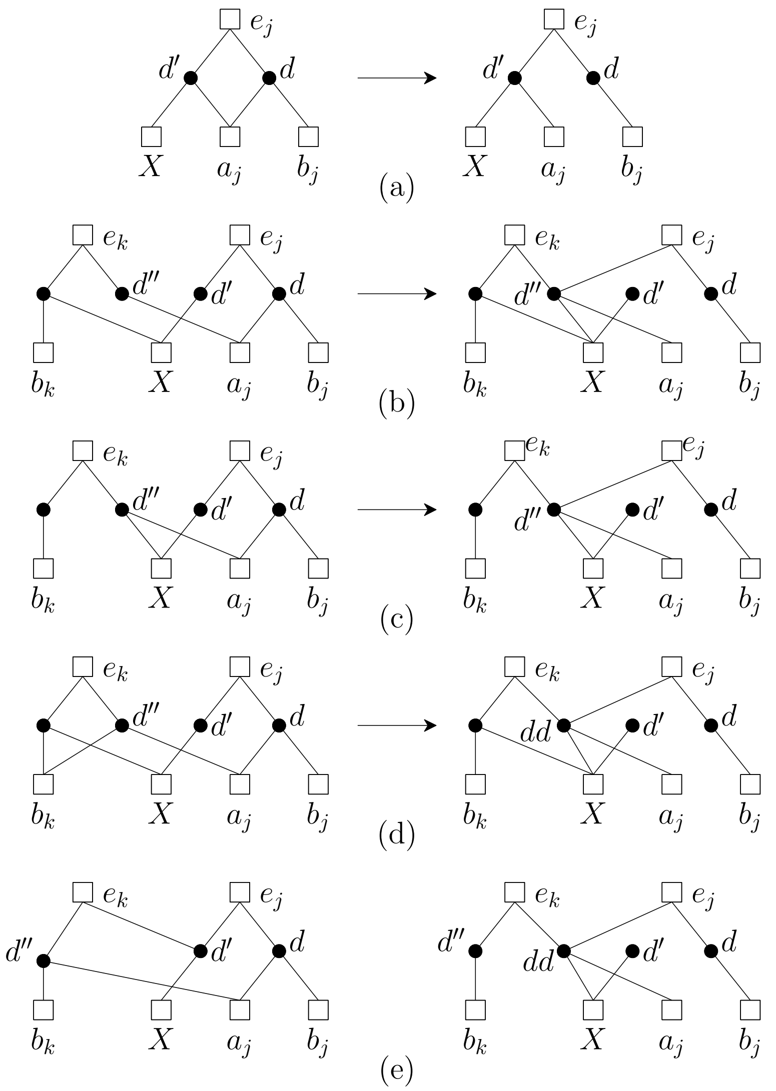


図 4.11: 補題 4.5 証明の説明図

である．

RL では，各頂点 $e_j = (a_j, b_j) \in V_1$ がもつダミー頂点は二つであり，それらの A_d は $\{a_j\}$, $\{b_j\}$, $\{a_j, X\}$, $\{b_j, X\}$ のいずれかである．A_d が X を含むようなダミー頂点すべてからなる集合を S^* とおき，それら以外のダミー頂点の集合を $\bar{S}^*$ とおく．明らかに

$$|S^*| + |\bar{S}^*| \geq n \tag{4.14}$$

が成立する．各 $d \in S^*$ に対しては $|A_d| = 2$ であり，各 $d \in \bar{S}^*$ に対しては $|A_d| = 1$ である．V_1 の頂点及び Y に接続する辺も考えると，RL における辺の総本数は $2|S^*| + |\bar{S}^*| + 2m + n + 1$ である．この値が $K_E = 2m + 2n + k_{VC} + 1$ を超えないので，

$$2|S^*| + |\bar{S}^*| \leq n + k_{VC} \tag{4.15}$$

が成立する．ここで式 (4.14) より $n+|S^*| \leq 2|S^*|+|\bar{S}^*|$ であるから，$|S^*| \leq k_{VC}$ が導かれる．

S^* に対応して，G_H の頂点の集合 S を

$$\{v \in V(H) \mid A_d = \{v, X\} \text{ なる } d \text{ が } S^* \text{に存在 } \}$$

と定める．RL において，V_1 の各頂点 e_j が S^* の少なくとも一つの頂点に隣接していることより，S は G_H の頂点被覆である．$|S| \leq |S^*| \leq k_{VC}$ であるから，G_H は要素数 k_{VC} 以下の頂点被覆をもつことになる． □

前述のように問題 MNE はクラス NP に属する．変換 f_E は n, m に関する多項式時間で実行可能であるから，補題 4.1, 4.6 より，問題 VC は MNE に多項式変換可能である．VC は NP 完全であるから，次の定理が成立する．

定理 4.2 問題 MNE は NP 完全である． □

変換 f_E より，問題 MNE は，入力のグラフ G を連結な 3 階層グラフに限定しても NP 完全である．

4.6 結言

本章では，階層描画において，複数の辺にダミー頂点の共有をさせる単純なアルゴリズムを提案した．計算機実験を行ったところ，通常の方法でダミー頂点を作成する場合に比べ，提案法 2 は頂点数と辺数を大幅に少なくすることができた．また，ダミー頂点を共有させることにより，最終的に得られる描画の描画幅と辺交差数を大きく削減し得ることを示した．次に，グラフの頂点数を最小にする問題，及び辺数を最小にする問題が，いずれも，入力を連結 3 階層グラフに限定しても NP 困難であることを証明した．**4.3.1** で示した提案法 2 はこれらの問題に対する発見的手法として用いることができるが，さらに有効なアルゴリズムについて検討することは今後の課題である．

第 5 章

階層グラフの直交描画アルゴリズム

5.1　緒言

Sugiyama ら [2, 3] の方法のように，各辺を直線で描く直線描画では，グラフが密になると辺交差数が非常に多くなり，グラフの構造が把握しづらくなることがある．本研究では，辺交差数を減らすための一つの方法として，ダミー頂点導入後のグラフの各辺を垂直・水平線分からなる経路として描く直交描画 [9]〜[11] に注目する．そして，本章では，階層グラフの直交描画を求める新しいアルゴリズムを提案する．

ダミー頂点導入後の階層グラフにおいて，ある階層の頂点のある集合 VS とすぐ下の階層の頂点のある集合 VS' に対し，$VS \cup VS'$ から誘導される部分グラフ G' が完全 2 部グラフをなすとき，G' の辺集合を完全 2 部辺集合と呼ぶことにする．階層グラフの直交描画アルゴリズム [9]〜[11] では，連続する 2 階層ごとにそれらの間の辺の集合を完全 2 部辺集合に分割し，同じ完全 2 部辺集合に属する辺の経路が垂直・水平線分の一部を共有することを許して，各完全 2 部辺集合を辺の交差なしに描く．分割された完全 2 部辺集合のそれぞれを高階辺と呼ぶ．以下，高階辺を，それが含む辺の端点の集合 $VS \cup VS'$ で表すことがある．

従来の直交描画法 [9]〜[11] では，高階辺の頂点数や描画形態に以下に述べるような制限を加えている．文献 [9] は，同一高階辺に含まれるものであっても，異なる 2 辺が水平線分の一部を共有することを禁止している．文献 [10] は，各高階辺がもつ上階層の頂点数 $|VS|$ が 1 に限るという制限を設けている．また，文献 [11] は，$|VS| \geq 2$ である高階辺も考えているが，各高階辺の描画に用い得る水平線分の本数を 1 以下に制限している．これらに対し，本研究では，$|VS|, |VS'|$ の値に制限を設けず，各高階辺の描画に最大 2 本の水平線分を用い得るものとする．図 5.1 に高階辺の描画の例を示す．同図 (a) 右は，2 本の水平線分 l, L を用いることにより，直線描画をする場合に比べ辺の交差を大きく減らし得る例であるが，文献 [9]〜[11] の方法ではこのような描画をすることはできない．

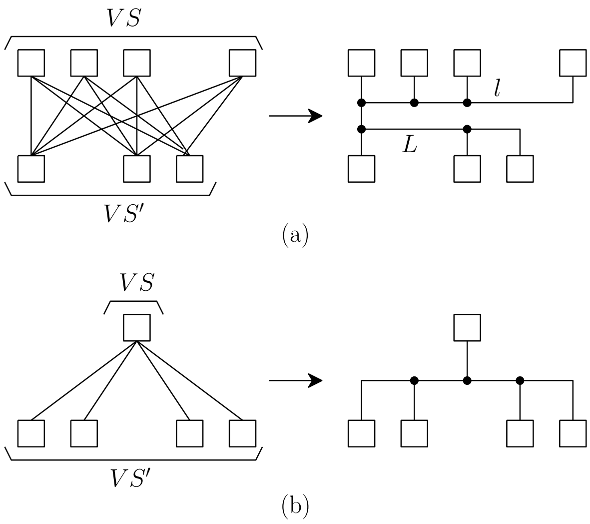


図 5.1: 高階辺の描画

階層グラフの直交描画では，グラフの構造が明確であり，辺を表す経路をたどりやすいことが望まれる．一般に，辺の交差数が非常に多い描画は，グラフの構造が不明確である．また，水平線分の y 座標の個数が多くて水平線分間の間隔が小さい直交描画や，長い水平線分が多く存在する直交描画は，辺を表す経路がたどりにくく見づらい．直交描画法に関する従来の研究には，高階辺の垂直・水平線分の配置を決定する際に，水平線分の y 座標の総数を少なくすることを目指したもの [9] と，辺交差数を少なくすることを目的としたもの [10, 11] がある．文献 [11] は，グラフの階層数が 2 であり，各高階辺がもつ上階層の頂点数が 1 に制限されている場合でさえ，辺交差数を最小にするように線分を配置する問題が NP 困難であることを示している．本章で提案する手法では，各階層における頂点の配置順序を定めた後，全高階辺の水平線分の長さの総和が小さくなるように頂点の x 座標を決定する．そして，その後，辺の交差を少なくし，その上で水平線分の y 座標の総数も小さくすることを目的として垂直・水平線分の配置を決定する．

本章の構成は以下のとおりである．まず，**5.2** において直交描画に関する諸定義を行い．次に，**5.3** では直交描画アルゴリズム全体の概略を示す．そして，**5.4** では高階辺の決定方法について，**5.5** では各頂点の x 座標の決定方法に関して，**5.6** では直交描画における各線分の配置の決定方法についてそれぞれ述べる．**5.7** では計算機実験の結果を示す．

5.2　直交描画に関する諸定義

最初に，有向グラフに関するいくつかの定義を示す．任意の有向グラフを $G = (V, E)$ とする．頂点 v から w に向かう有向辺を $(v \to w)$ と表す．G の任意の頂点 v に対し，v から出る辺を射出辺と呼び，その集合を $E^+_G(v)$ と表す．また，v に入る辺を射入辺と呼

び，その集合を $E_G^-(v)$ と表す．射入辺をもたない頂点をソースと呼び，射出辺をもたない頂点をシンクと呼ぶ．頂点の任意の集合 $V' \subseteq V$ に対して，G から V' の全ての頂点（及びそれらに接続する辺）を削除して得られるグラフを $G - V'$ と表す．また，辺の任意の集合 $E' \subseteq E$ に対して，G から E' の全ての辺を削除して得られるグラフを $G - E'$ と表す．

E_R を E の任意の部分集合とする．もし，G の任意の有向閉路に対して，E_R がその上の辺を 1 本以上含むならば，E_R は G の帰還辺集合であるという．E_R が帰還辺集合であれば，$G - E_R$ はアサイクリックになる．G の全ての帰還辺集合の中で，辺数が最小のものを最小帰還辺集合と呼ぶ．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 [16].

ダミー頂点を導入後の階層グラフを G^* と表し，その階層を上から順に $V_1^*, V_2^*, \ldots, V_h^*$ とする．本研究で求める直交描画 $\tilde{G}$ は，次の描画条件を満たすものである．

（描画条件） G^* において辺 (v, w) （$v \in V_i^*$, $w \in V_j^*$, $i < j$）が存在するときかつそのときに限り，$\tilde{G}$ において，v から w への経路で，以下の制約 (i), (ii) を満たすものが存在する．

(i) 水平線分は左右どちらの方向にたどってもよい．

(ii) 垂直線分は上から下の方向にのみたどってよい．

G^* において，ある階層 V_i^* の頂点のある集合 VS とすぐ下の階層 V_{i+1}^* の頂点のある集合 VS' に対し，$VS \cup VS'$ から誘導される部分グラフ G' が完全 2 部グラフをなすとき，G' の辺集合を完全 2 部辺集合と呼ぶことにする．階層グラフの直交描画アルゴリズムでは，連続する 2 階層ごとにそれらの間の辺の集合を完全 2 部辺集合に分割し，同じ完全 2 部辺集合に属する辺の経路が垂直・水平線分の一部を共有することを許して，各完全 2 部辺集合を線分の交差なしに描く．分割された完全 2 部辺集合のそれぞれを高階辺と呼ぶ．高階辺は辺の集合ではあるが，辺の端点の集合を決めれば一意に定まる．そこで，高階辺を頂点の集合 $VS \cup VS'$ で表すことがある．例えば，図 1.1(d) の直交描画 (図 5.2 に再掲) では，5 本の高階辺 $\{1, d_1\}, \{1, 2, 3, d_2\}, \{d_1, 6\}, \{4, d_2, 7, 8\}, \{5, 8\}$ が作られている．

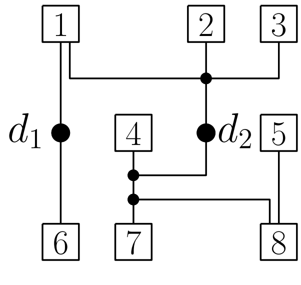


図 5.2: 直交描画の例

本研究では，各高階辺 $e = VS \cup VS'$（$VS \subseteq V_i^*, VS' \subseteq V_{i+1}^*$）の描画は，基本的に VS の頂点を垂直線分とたかだか 1 本の水平線分を用いてつなぎ，VS' の頂点を垂直線分とたかだか 1 本の水平線分を用いてつないで，それらの水平線分間を垂直線分で接続したものとする．VS の頂点をつなぐための水平線分を e の上水平線分と呼び，VS' の頂点をつなぐための水平線分を下水平線分と呼ぶ．図 5.1(a) 右の例では，l が上水平線分であり，L が下水平線分である．図 5.1(b) の例では，$|VS| = 1$ であり，上水平線分を作っていない．

直交描画において，異なる高階辺の線分が交差する回数を辺交差数と呼ぶ．例えば，図 5.3(a), (b) の描画の辺交差数は，それぞれ，$8, 2$ である．図 5.3(a) の例のように，$|VS|, |VS'|$ が共に 2 以上の高階辺でも水平線分を 1 本だけ用いて描画することが可能である．しかし，全ての高階辺に対してそのようにすると，辺交差数がかなり大きくなる場合がある．そこで本研究では，各高階辺に対して上下水平線分を使用できるものとしている．

直交描画において，高階辺に対して作られる任意の水平線分 l の左右の端点は，その高階辺に含まれるある頂点に垂直線分で接続される．l の左端点が頂点 v に接続されるとき，$x(v)$ を l の左仮座標と呼ぶ．同様に，l の右端点が頂点 w に接続されるとき，$x(w)$ を l の右仮座標と呼ぶ．2 階層グラフの直交描画の例を図 5.4 に示す．ここで，例えば水平線分 L_2 の左仮座標は 1 であり，右仮座標は 7 である．実際に水平線分 l を描画するときの左右端点の x 座標のことを実座標と呼ぶ．水平線分の左右端点の実座標は左右仮座標に一致するとは限らない．例えば，ある水平線分 l の右端点が垂直線分を介して実頂点 v に接続されるとき，l の右仮座標は $x(v)$ であるが，右端点の実座標は $x(v) - W_R/2$ 以上 $x(v) + W_R/2$ 以下の範囲で定める．

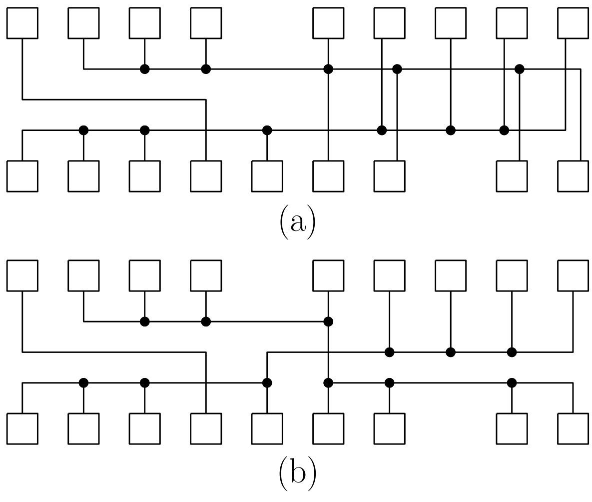


図 5.3: 上下水平線分の使用により辺交差が減少する例

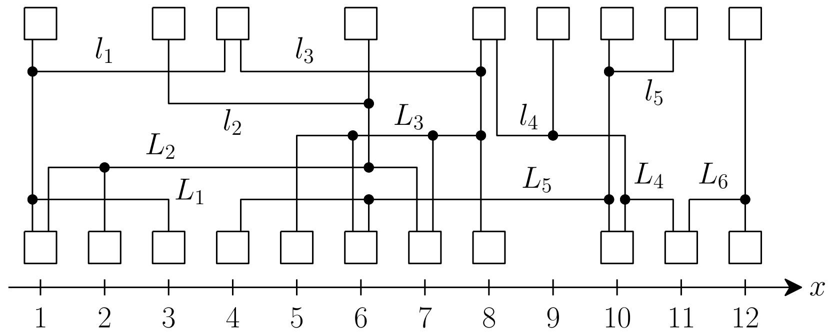


図 5.4: 2 階層グラフの直交描画の例

各水平線分 l に対し，その左仮座標を $left(l)$，右仮座標を $right(l)$ と表す．l の線分長を $right(l) - left(l)$ と定義する．更に各高階辺 e について，上下水平線分の線分長の和を，e の水平線分長和と呼び，$sl(e)$ と表す．ただし，e の上水平線分が存在しないとき，その線分長は 0 として計算する．e の下水平線分がないときも同様である．例えば図 5.4 では，l_1 の線分長は 3，L_1 の線分長は 2 であり，これらをもつ高階辺の水平線分長和は 5 である．直交描画中の全ての高階辺の水平線分長和の総和を総水平線分長和と呼ぶ．

5.3 直交描画アルゴリズムの概略

本章では，単純な階層グラフ G が与えられたときに，その直交描画を求めるアルゴリズムを提案する．**5.1** で述べたように，提案法 3 は，高階辺の垂直・水平線分の配置を決定する際，辺の交差を少なくし，その上で水平線分の y 座標の個数も小さくすることを目的としている．提案法 3 は以下の 5 段階よりなる．

第 1 段階：　ダミー頂点の集合の決定
第 2 段階：　高階辺の集合の決定
第 3 段階：　各階層における頂点の配置順序の決定
第 4 段階：　頂点の x 座標の決定
第 5 段階：　各高階辺の各垂直・水平線分の座標の決定と高階辺の描画

第 1 段階では，与えられた階層グラフ G に対し，第 4 章の **4.3.1** で示したダミー頂点の共有化を行うの方法を実行することにより，ダミー頂点数が比較的少ない階層グラフを求める．得られたグラフを $G^* = (V^*, E^*)$ と表し，その階層を上から順に $V_1^*, V_2^*, \ldots, V_h^*$ と表すことにする．

第 2 段階では，最終的に求める直交描画の辺交差数を小さくすることを意図して，高階辺の集合 E^{HP} を決定する．この段階の詳細は **5.4** で述べる．

第 3 段階では，G^* に対して，重心法 [2, 3] と文献 [15] の手法を実行して，各階層における頂点の順序を定める．これらの方法は辺交差数が少ない直線描画を求めるための頂点順序決定法である．現時点ではこれらの方法を用いているが，今後，直交描画により適した方法について検討していく予定である．

第 4 段階では，総水平線分長和を最小とするように V^* の頂点の x 座標を定める．提案法 3 では，この問題を線形計画問題に帰着し，ソルバー SCIP[24] を用いて解いている．この段階の詳細は **5.5** で説明する．

第 5 段階では，辺交差数をできるだけ小さくし，その上で水平線分の y 座標の個数を少なくすることを目的として，高階辺の各垂直・水平線分の座標を決定する．この段階の詳細は **5.6** で説明する．

5.4 高階辺の決定

提案法 3 の第 2 段階では，各整数 $i = 1, 2, \ldots, h-1$ について，V_i^* と V_{i+1}^* の頂点を結ぶ高階辺の集合 E_i^{HP} を決定する．集合 $V_i^* \cup V_{i+1}^*$ から誘導される G^* の部分グラフを G_i^* と書き，このグラフにおける各頂点 v の隣接頂点の集合を N_v と表すことにする．辺交差数を少なく抑えるためには，G_i^* において，できるだけ辺の多い完全 2 部部分グラフを見つけ，その辺集合を高階辺にすることが望ましい．しかし，2 部グラフにおいて，辺数最大の完全 2 部部分グラフを見つける問題は一般に NP 困難である [17]．そこで本研究では，辺数ができるだけ大きい極大完全 2 部部分グラフを発見的手法により見つけ，対応する高階辺を設けた後，その部分グラフの辺を G_i^* から削除するという処理を繰返すことにより，高階辺の集合 E_i^{HP} を決定する．

E_i^{HP} を決定する手続きの概略を図 5.5 に示す．この手続きの (2) では，辺数が多い極大完全 2 部部分グラフで 3 頂点以上を含むものを見つけ，対応する高階辺を作っている．この処理を可能な限り繰返した後，(3) では 2 頂点からなる高階辺の決定をしている．

集合 $E_1^{HP} \cup E_2^{HP} \cup \cdots \cup E_{h-1}^{HP}$ を E^{HP} とする．

5.5 頂点の座標の決定

階層グラフ G^* に対して，高階辺の集合 E^{HP} を決定し，各階層 V_i^* $(i = 1, 2, \ldots, h)$ における頂点の順序を定めたものとする．以下，直線描画に対する文献 [5] と同様，頂点の x 座標を決定する問題を線形計画問題に帰着する．

各階層 V_i^* において左から j 番目（$1 \leq j \leq |V_i^*|$）に置く頂点を v_i^j と表す．提案法 3 の第 4 段階では，

(a) 各 $i = 1, 2, \ldots, h,\ j = 1, 2, \ldots, |V_i^*|$ に対して，$x(v_i^j)$ は非負整数．

(1) $E_i^{HP} \leftarrow \emptyset$ とする．
(2) G_i^* に次数 2 以上の頂点がなくなるまで，次の (2a)∼(2c) を実行する．
  (2a) $num\_edges \leftarrow 0,\ M_1 \leftarrow \emptyset,\ M_2 \leftarrow \emptyset$ とする．
  (2b) 同一階層（V_i^* あるいは V_{i+1}^*）の全ての異なる 2 頂点 v, w に対して，次の (2b1)∼(2b3) を実行する．
    (2b1) $VS \leftarrow \{v, w\}$, $VS' \leftarrow N_v \cap N_w$ とする．
    (2b2) VS の要素以外に VS' の全要素に隣接している頂点が存在すれば，それらを全て VS に加える．
    (2b3) $|VS| \cdot |VS'| > num\_edges$ ならば，$M_1 \leftarrow VS$,$M_2 \leftarrow VS'$, $num\_edges \leftarrow |VS| \cdot |VS'|$ とする．
  (2c) 高階辺 $M_1 \cup M_2$ を E_i^{HP} に加え，M_1 の頂点と M_2 の頂点を結ぶ辺を全て G_i^* から削除する．
(3) G_i^* に残った各辺 (v, w) に対して，高階辺 $\{v, w\}$ を E_i^{HP} に加える．

図 5.5: 高階辺の集合 E_i^{HP} を定める手続きの概略

(b) 各 $i = 1, 2, \ldots, h$ に対して，$x(v_i^1) < x(v_i^2) < \cdots < x(v_i^{|V_i^*|})$.

という制約条件の下で，総水平線分長和が最小となるように全変数 $x(v_i^j)$ の値を決定する問題を扱う．

各高階辺 $e \in E_i^{HP}$ に対して，階層 V_i^* の頂点で e に含まれるもののうち，最も左のものを v_i^{j1}，最も右のものを v_i^{j2} とする．更に，階層 V_{i+1}^* の頂点で e に含まれるもののうち，最も左のものを v_{i+1}^{j3}，最も右のものを v_{i+1}^{j4} とする．e の上下水平線分は 1 本の垂直線分で接続することになるから，$x(v_i^{j2}) < x(v_{i+1}^{j3})$ あるいは $x(v_{i+1}^{j4}) < x(v_i^{j1})$ である場合には，上下水平線分の少なくとも一方を延長する必要がある．その延長分の長さを含めて，$sl(e)$ は次式で与えられる [22]..

$$\begin{aligned} sl(e) = &(x(v_i^{j2}) - x(v_i^{j1})) + (x(v_{i+1}^{j4}) - x(v_{i+1}^{j3})) \\ &+ \max\{x(v_{i+1}^{j3}) - x(v_i^{j2}), 0\} \\ &+ \max\{x(v_i^{j1}) - x(v_{i+1}^{j4}), 0\} \end{aligned} \tag{5.1}$$

$\max\{x(v_{i+1}^{j3}) - x(v_i^{j2}), 0\}$ を a_e，$\max\{x(v_i^{j1}) - x(v_{i+1}^{j4}), 0\}$ を b_e とそれぞれおき，次の (c), (d) を制約条件に追加する．

(c) 各高階辺 $e \in E^{HP}$ について，a_e, b_e は非負整数.
(d) 各高階辺 $e \in E^{HP}$ について，$a_e \geq x(v_{i+1}^{j3}) - x(v_i^{j2})$ かつ $b_e \geq x(v_i^{j1}) - x(v_{i+1}^{j4})$.

このようにして，総水平線分長和を最小化する問題は，$\{v_i^j \mid i = 1,2,\ldots,h,\ j = 1,2,\ldots,|V_i^*|\} \cup \{a_e, b_e \mid e \in E^{HP}\}$ を変数とし，制約条件 (a)〜(d) の下で $\sum_{e\in E^{HP}} sl(e)$ を最小化する整数計画問題として定式化できる．この問題から整数条件を外した線形計画問題の最適解が整数条件を満たすことを，文献 [23] におけるのと同様の議論により証明することができる（詳細は省略する）．提案法 3 の第 4 段階では，この線形計画問題をソルバーSCIP[24] を用いて解くことにより G^* の全頂点の x 座標を決定する．

5.6 高階辺の線分の座標決定

第 4 段階で V^* の各頂点の座標を決定した後，直交描画アルゴリズムの第 5 段階では，各高階辺の水平・垂直線分の座標を決定し，高階辺の描画を行う．以下，まず **5.6.1** においてその方法の概略を示し，各ステップの詳細について **5.6.2** 以降で説明する．

5.6.1 概略

提案する方法は，$i = 1,2,\ldots,h-1$ の順に，高階辺の集合 E_i^{HP} に対して，以下のステップ 1〜5 を実行するものである．

ステップ 1： 各高階辺を描画する際に用いる水平線分の集合と，各水平線分の左右仮座標を決定する．

ステップ 2： 各高階辺に対し，それが上下水平線分をもつならば，それらを結ぶ垂直線分に仮の x 座標を割り当てる．

ステップ 3： 水平線分の集合を水平線分群に分割し，それらの（上から下への）順序を定めた後，各水平線分に y 座標を割り当てる．

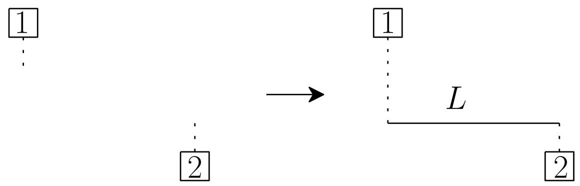
ステップ 4： 各高階辺の上下水平線分を結ぶ垂直線分の x 座標を変更することにより，辺交差数の削減を試みる．

ステップ 5： 各水平線分の左右端点の実座標を決定し，垂直線分を加えて，V_i^* と V_{i+1}^* の間の描画を行う．

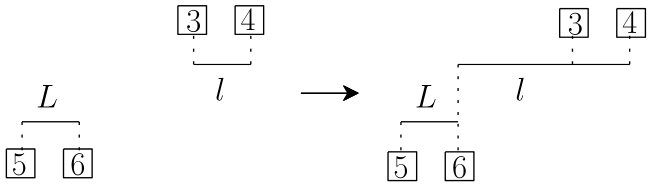
各ステップの実行方法について，次節以降で述べる．

5.6.2 ステップ 1

ステップ 1 では，E_i^{HP} 中の各高階辺 $e = VS \cup VS'$（$VS \subseteq V_i^*, VS' \subseteq V_{i+1}^*$）に対して水平線分を作る．まず，$VS$ の頂点をつなぐ上水平線分 l と VS' の頂点をつなぐ下水平線分 L を作る．ただし $|VS| = 1$ であれば，便宜上 l は長さ 0 の水平線分とする．同様に，$|VS'| = 1$ であれば L は長さ 0 の線分とする．l の左仮座標 $left(l)$ の値は，VS に



(a) $e = \{1, 2\}$



(b) $e = \{3, 4, 5, 6\}$

図 5.6: 水平線分の延長

属する頂点のうち最も左にあるものの x 座標とし，右仮座標 $right(l)$ の値は VS 中の最も右にある頂点の x 座標とする．L の左右仮座標も同様に定める．長さ 0 の水平線分については，左右の仮座標が同じ値になる．このとき，もし $right(L) < left(l)$ 若しくは $right(l) < left(L)$ であれば，l 若しくは L を以下のように延長し，左右仮座標の更新を行う．

- l の長さが 0 であれば，VS が含む唯一の頂点の x 座標まで L を延長する．図 5.6(a) に例を示す．
- l の長さが 0 でなければ，L の近い方の端点の x 座標まで l を延長する．図 5.6(b) に例を示す．

5.6.3　ステップ 2

ステップ 1 で定めた各水平線分 l に対して，開区間 $(left(l), right(l))$ を OI_l と書くことにする．ステップ 2 では，各高階辺 $e \in E_i^{HP}$ に対し，上水平線分 l と下水平線分 L を結ぶ垂直線分に仮の x 座標 tx_e を割当てる．tx_e は，l と L の左右仮座標の中から，以下の単純な規則に従って定める．

(i) l の左仮座標が L の左右いずれかの仮座標に等しいとき，$tx_e = left(l)$ とする．図 5.7(a), (b) 参照．

(ii) (i) の条件を満たさず，l の右仮座標が L の左右いずれかの仮座標に等しいとき，$tx_e = right(l)$ とする．図 5.7(c), (d) 参照．

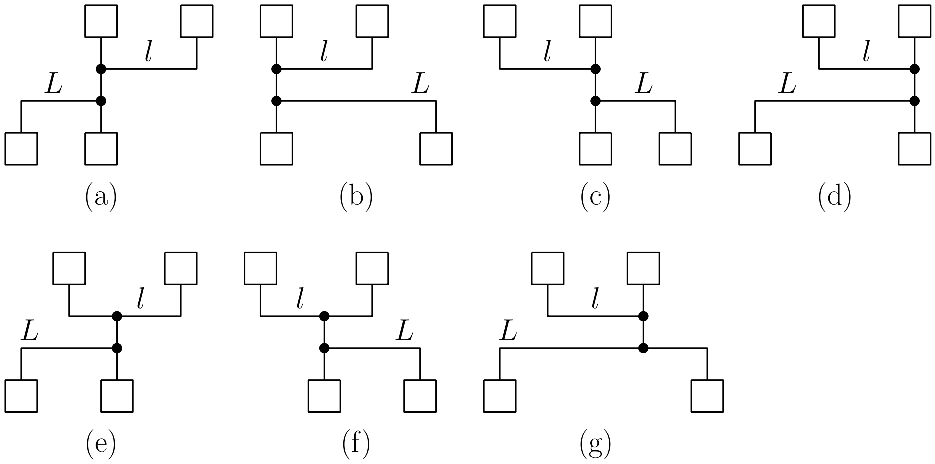


図 5.7: 上下水平線分の接続

(iii) (i), (ii) の条件を満たさず，L の右仮座標を OI_l が含むとき，$tx_e = right(L)$ とする．図 5.7(e) 参照.

(iv) (i)∼(iii) の条件を満たさず，L の左仮座標を OI_l が含むとき，$tx_e = left(L)$ とする．図 5.7(f) 参照.

(v) (i)∼(iv) の条件を満たさないとき，tx_e を，上下水平線分の右仮座標の小さい方の値とする．図 5.7(g) 参照.

以上の処理の終了後，長さ 0 の水平線分を全て削除する．残った上水平線分全てからなる集合を US_i，下水平線分全てからなる集合を LS_i とする．

5.6.4 ステップ 3

5.6.4.1 水平線分の配置に関する制約

$US_i \cup LS_i$ に属し，かつ $OI_l \cap OI_{l'} = \emptyset$ である任意の 2 本の水平線分 l, l' に対し，それらに接続する垂直線分のことを考慮しなければ，l, l' の端点の実座標を適切に定めることによって，l と l' が互いに重ならないようにしながら，同じ y 座標を割当てることができる．しかし，次の条件を満たす高階辺 $e, e' \in E_i^{HP}$ が存在する場合には，水平線分の y 座標の決定の際に注意が必要である．
（クロス条件）高階辺 e, e' が共に線分長 1 以上の上下水平線分をもつものとする．e の上下水平線分を l_e, L_e とし，e' の上下水平線分を $l_{e'}, L_{e'}$ としたとき，

$$right(l_e) = left(L_e) = left(l_{e'}) = right(L_{e'}) \tag{5.2}$$

が成立する．図 5.8 参照.

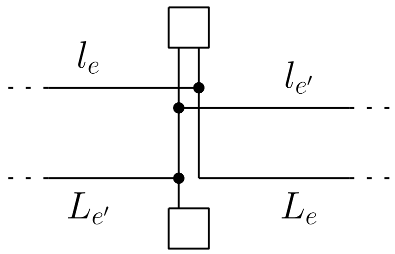


図 5.8: クロス条件

高階辺 e, e' がクロス条件を満たすとき，それらの上水平線分に異なる y 座標を与えるか，下水平線分に異なる y 座標を与えるかのどちらかが必要となる．提案法 3 では，このような場合，上水平線分に異なる y 座標を与えるものとする．

2 本の水平線分 l, l' に対して以下の (i), (ii) が成立するとき，l と l' は互いに独立であるということにする．

(i) $OI_l \cap OI_{l'} = \emptyset$.

(ii) l, l' はクロス条件を満たす 2 本の高階辺の上水平線分ではない．

図 5.4 において，例えば L_1 は l_1, L_2 以外の水平線分と独立であり，L_6 は他の全ての水平線分と独立である．l_4 と l_5 はクロス条件を満たす高階辺の上水平線分であるので，互いに独立でない．任意の $HS \subseteq US_i \cup LS_i$ に対し，その任意の 2 本の水平線分が独立であるならば，HS は独立線分集合であるという．

直交描画を行う際，同じ高階辺の下水平線分を上水平線分より上に描くと，**5.2** で述べた描画条件に違反することになる．本研究では，水平線分の配置に関して，以下の二つの制約を設けることにする．

制約 1： 同じ高階辺が上下水平線分をもつ場合，下水平線分を上水平線分より上に配置しない．

制約 2： 同じ高階辺の水平線分であるか否かによらず，互いに独立でない水平線分には同じ y 座標を割当てない．特に，上水平線分 l と下水平線分 L が互いに独立でないならば，l を L より上に描く．

これらの制約は，同じ高階辺の上下水平線分が独立であるときに，それらに同じ y 座標を割当てることを禁止していない．

E_i^{HP} の高階辺に含まれる頂点の x 座標全てからなる集合を XS_i とする．更に，任意の $t \in XS_i$ に対し，V_i^* と V_{i+1}^* の間に存在し，x 座標が t の頂点に接続する垂直線分全てからなる集合を VL_t とする（ここでは，2 本以上の垂直線分が分岐点で接続している場合，それらをまとめて 1 本の垂直線分とみなす）．垂直線分の x 座標を定めれば，接続す

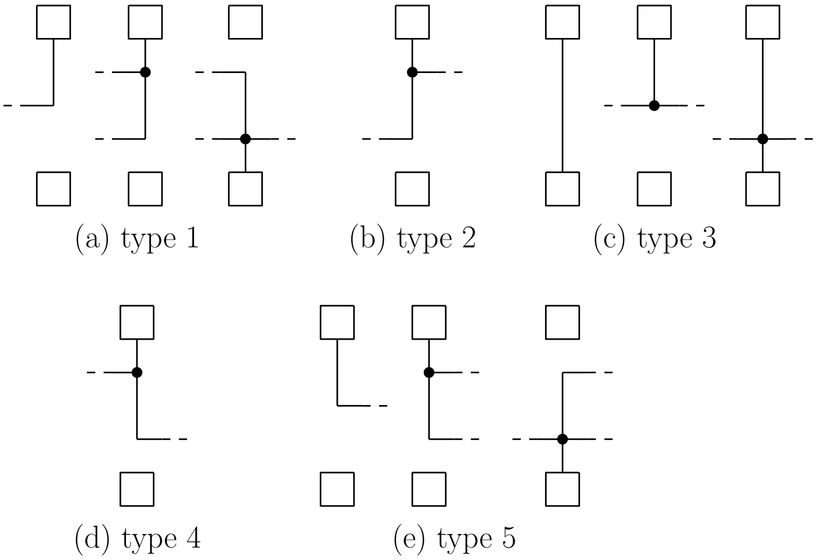


図 5.9: 垂直線分のタイプ

る水平線分の端点の実座標も定まることになる．

VL_t に属する任意の垂直線分 vl に対して，右仮座標が t の水平線分 l が接続しているとき l は左方向であるといい，左仮座標が t の水平線分 l が接続しているとき l は右方向であるという．また，分岐点で vl に接続し，左右に伸びている水平線分 l は両方向であるという．各 $t \in XS_i$ に対し，VL_t に属する垂直線分を，接続する水平線分（たかだか 2 本）の方向によって以下の五つのタイプに分類する．図 5.9 に例を示す．垂直線分 vl のタイプを $type(vl)$ と表す．

タイプ 1： 左方向の水平線分のみ，あるいは左方向の水平線分と両方向の水平線分が接続している．

タイプ 2： 左方向の下水平線分と右方向の上水平線分が接続している．

タイプ 3： 左方向あるいは右方向の水平線分が接続していない．

タイプ 4： 左方向の上水平線分と右方向の下水平線分が接続している．

タイプ 5： 右方向の水平線分のみ，あるいは右方向の水平線分と両方向の水平線分が接続している．

次の定理は，ステップ 3 及びステップ 5 の基礎となるものである．

定理 5.1 $LS_i \cup US_i$ の全ての水平線分に対して，制約 1，2 を満たす範囲で任意に y 座標を割当てたものとする．このとき，各垂直線分の x 座標を適切に定めることによって，異なる水平線分同士，あるいは垂直線分同士が重ならないようにすることができる．

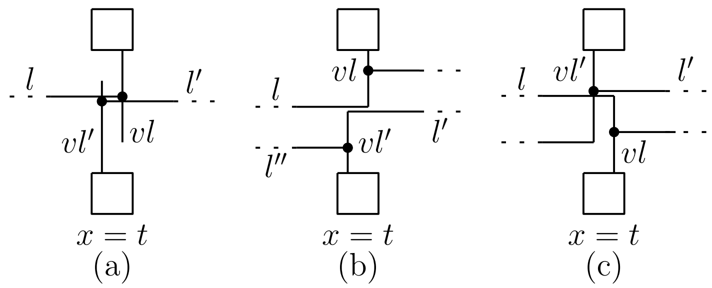


図 5.10: 定理 5.1 証明の説明図

（証明）$LS_i \cup US_i$ の全ての水平線分に対して，制約 1，2 を満たす範囲で任意に y 座標を割当てた後，各 $t \in XS_i$ に対して，VL_t の垂直線分を，タイプ $1, 2, \ldots, 5$ の順に左から並べ，（x 座標が t の頂点に接続する範囲で）x 座標を定めるものとする．ただし，ここでは，VL_t の全ての垂直線分に異なる x 座標を与えるものとする．また，タイプ 1 の垂直線分が複数ある場合はそれらの順序は任意とする．タイプ 3，タイプ 5 の垂直線分がそれぞれ複数ある場合も同様である．タイプ 2 の垂直線分が複数ある場合には，接続する下水平線分がより上に配置されているものをより左に置く．タイプ 4 の垂直線分が複数ある場合には，接続する上水平線分がより上に配置されているものをより右に置く．

以上のようにしたとき，異なるどの 2 本の垂直線分も重ならないことは明らかである．同じ y 座標をもつ 2 本の水平線分 l, l' が互いに重なったものと仮定する．水平線分への y 座標の割当てが制約 2 に従ったものであることより，l と l' は互いに独立であり，それらの一方は左方向，他方は右方向である．一般性を失うことなく，l が左方向，l' が右方向であるものとする．l が接続する垂直線分を vl，l' が接続する垂直線分を vl' とする．l と l' が重なることより，vl' が vl より左に置かれている．図 5.10(a) 参照（図を見やすくするため，l と l' の y 座標を少しずらして描いている）．

vl' は右方向の l' に接続しているから $type(vl')$ は 2, 4, 5 のいずれかであり，vl は左方向の l に接続しているから $type(vl)$ は 1, 2, 4 のいずれかである．$type(vl') \leq type(vl)$ であるから，$type(vl'), type(vl)$ は共に 2 あるいは 4 であることになる．

$type(vl') = type(vl) = 2$ であるとすると，l は下水平線分，l' は上水平線分であることになる．vl' に接続している下水平線分を l'' とする．制約 1 より，l' は l'' より下に置かれていないから，l が l'' より上に置かれていることになる．図 5.10(b) 参照．これは，下水平線分がより上に配置されているものをより左に置くとしたタイプ 2 の垂直線分の並べ方に矛盾する．よって，vl', vl のいずれかはタイプ 2 ではない．vl', vl のいずれかがタイプ 4 でないことも同様にして証明できる．

$type(vl') = 2, type(vl) = 4$ であるものとする．vl' が vl より左にあるので，vl', vl に接続する下水平線分は重なり得ない．よって，l', l は共に上水平線分である．しかし，l', l

をもつ高階辺はクロス条件を満たしているから，l' と l は独立ではなく，それらが同じ y 座標をもつことは制約 2 に反する．よって，$type(vl') = 2, type(vl) = 4$ であることもない． $\square$

直交描画において，同一の y 座標をもつ水平線分の極大な集合のそれぞれを水平線分群と呼ぶ．図 5.4 の描画では，11 本の水平線分が五つの水平線分群 $\{l_1, l_3, l_5\}$, $\{l_2\}, \{L_3, l_4\}, \{L_2\}, \{L_1, L_4, L_5, L_6\}$ に分割されている．提案法 3 では，連続する 2 階層の間に k 個の水平線分群を作る場合, 連続する水平線分群間の距離を $(Dist - W_R)/(k+1)$ とする．この値が小さすぎると描画が見づらくなるため，本研究では，辺交差数をできるだけ少なくした上で，水平線分群数（水平線分の y 座標の個数に等しい）を小さくすることを目的の一つとしている．

ステップ 3 では，互いに独立でない水平線分の上下関係を制約 1, 2 の下で定めた後，水平線分群を決定して，それらに y 座標を割当てる．文献 [9] が示している水平線分配置法は，水平線分群数の最小化を目的としたもので，本質的に，各水平線分を区間と見なして，対応する区間グラフ [18] の彩色問題を解いている．また，文献 [10] は，辺交差数を最小とする問題を，区間グラフをもとにしたある有向グラフの帰還辺集合 [19] を求める問題として定式化している（ただし，文献 [10] は，その問題を解くための具体的な方法について，十分な説明を与えていない）．本研究でも，区間グラフをもとにした有向グラフのある帰還辺集合を求める問題を考える．以下，**5.6.4.2** でその問題について述べた後，**5.6.4.3** で水平線分の y 座標の決定方法を示す．

5.6.4.2　問題の定式化

任意の有向グラフを $G_B = (V_B, E_B)$ とする．任意の 2 頂点 $v, w \in V_B$ に対し，$(v \to w) \in E_B$ 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_B を双方向グラフと呼ぶ．双方向グラフ G_B 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．また，以下の 2 条件を満たす有向グラフ $G_{CB} = (V_{CB}, E_{CB})$ を複合双方向グラフと呼ぶことにする．

(a) $V_{CB} = V_1 \cup V_2$（$V_1 \cap V_2 = \emptyset$），$E_{CB} = E_1 \cup E_2 \cup E_3$ 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．

水平線分の集合 $US_i \cup LS_i$ から，各辺に重み，長さという 2 種類の値を与えた有向グラフ $H = (V_H, E_H)$ を作成する．H の頂点は水平線分と一対一に対応する．簡単のため，

H の各頂点を，対応する水平線分と同じ記号で表すことにする．

$$V_H = V_H^U \cup V_H^L, \tag{5.3}$$
$$E_H = E_H^U \cup E_H^L \cup E_H^B \tag{5.4}$$

である．ここで，

$$V_H^U = \{l \mid l \in US_i\}, \tag{5.5}$$
$$V_H^L = \{L \mid L \in LS_i\}, \tag{5.6}$$
$$E_H^U = \{(l \to l'), (l' \to l) \mid l, l' \in US_i \text{ かつ } l \text{ と } l' \text{は互いに独立でない}\}, \tag{5.7}$$
$$E_H^L = \{(L \to L'), (L' \to L) \mid L, L' \in LS_i \text{ かつ } L \text{ と } L' \text{は互いに独立でない}\}, \tag{5.8}$$
$$E_H^B = \{(l \to L) \mid l \in US_i, L \in LS_i \text{ かつ } l \text{ と } L \text{ は同じ高階辺の水平線分であるか，互いに独立でない}\} \tag{5.9}$$

である．H の部分グラフ (V_H^L, E_H^L)，(V_H^U, E_H^U) はそれぞれ双方向グラフであり，H は複合双方向グラフである．(V_H^L, E_H^L) は，区間集合 $\{OI_L \mid L \in LS_i\}$ に対する区間グラフの各辺を逆並行辺対で置き換えたものであり，(V_H^U, E_H^U) は，$\{OI_l \mid l \in US_i\}$ に対する区間グラフにクロス条件を考慮した辺を加えた後，各辺を逆並行辺対で置き換えたものである．また E_H^B は，制約 1, 2 に従って水平線分の上下関係を制限するためのものである．例として，図 5.4（図 5.11(a) に再掲）の水平線分の集合に対する複合双方向グラフ H を図 5.11(b) に示す．図中，$E_H^U \cup E_H^L$ の各辺を実線で，E_H^B の各辺を破線で，それぞれ示している．この図では，一部の辺についてのみ，重みと長さを示している．辺の近傍にカッコなしで示している値が重みであり，カッコ付きで示している値（2 箇所のみ）が長さである．

H の各辺の重みは以下のように定める．まず，E_H^B の辺の重みは全て 0 とする（図 5.11(b) では E_H^B の辺の重みは省略している）．E_H^U の各辺 $(l \to l')$ については，水平線分 l を l' より上に置いたとき，l, l' 及びそれらに接続する垂直線分が作る辺交差の個数を重みとして与える．l と l' が上階層 V_i^* の同じ頂点 v につながっている場合，v に接続する垂直線分の x 座標の大小は，辺交差の個数が小さくなるように一時的に定めるものとする．E_H^L の各辺についても同様である．例えば図 5.11(a) の L_1 と L_2 に注目すると，これらはいずれも下階層の左端の頂点につながっている．L_2 を L_1 より上に置く場合，L_1, L_2 のいずれの左端点をより左に置いても辺交差の個数は 2 であるから，辺 $(L_2 \to L_1)$ の重みは 2 とする．一方，L_1 を L_2 より上に置く場合，図 5.11(c) に示すように，L_1 の左端点を L_2 の左端点より左にすると辺交差は 1 箇所，右にすると辺交差は 2 箇所となる．よって，$(L_1 \to L_2)$ の重みは 1 とする．

前述のように，ステップ 3 では，まず，互いに独立でない水平線分の上下関係を定める．そのため，複合双方向グラフ H の辺の重みは，水平線分の上下関係によって変わる

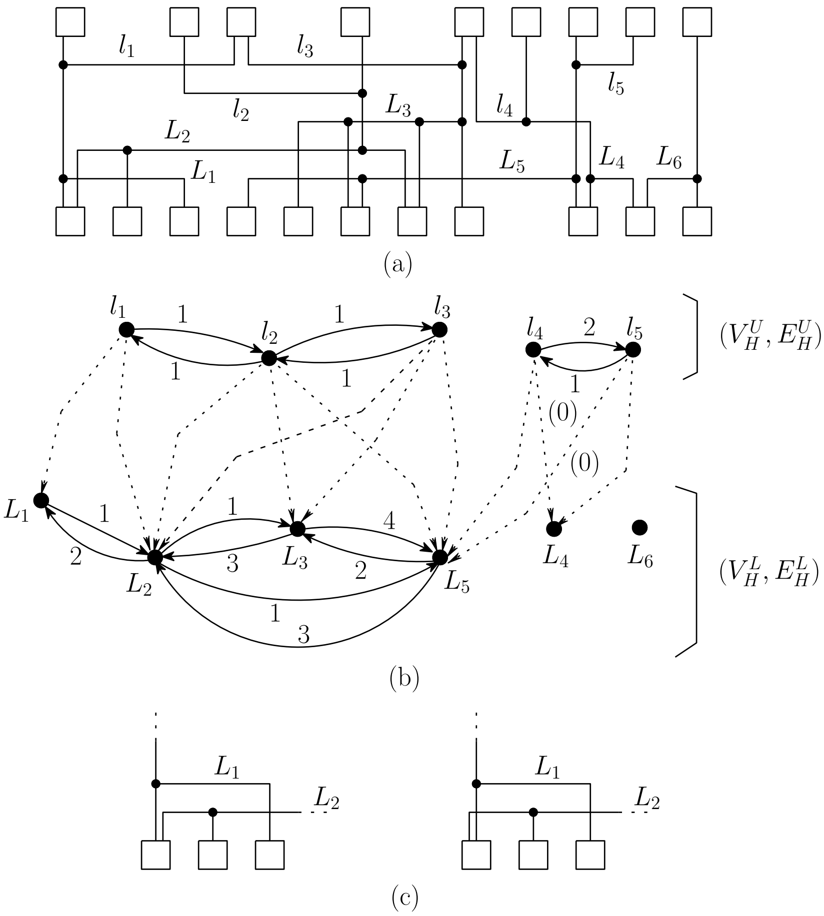


図 5.11: 複合双方向グラフ H の例

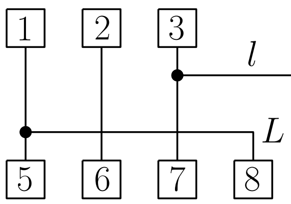


図 5.12: 複合双方向グラフ H では考慮していない辺の交差の例

可能性のある辺交差の個数のみを反映している．例えば図 5.12 において，高階辺 $\{2, 6\}$，$\{3, 4, 7\}$ はいずれも下水平線分をもっていない．このような高階辺の垂直線分が，上水平線分をもたない高階辺の下水平線分（図 5.12 では L）と交差する回数は，制約 1, 2 の下では水平線分の順序によらない（図 5.12 において，l を L より下に配置すれば辺交差数が変わるが，制約 2 に違反する）．よって，複合双方向グラフ H では，図 5.12 のような交差の個数を考慮していない．同様に，上水平線分をもたない高階辺の垂直線分が，下水平線分をもたない高階辺の上水平線分と交差する回数も考慮していない．

H の各辺の長さは以下のように定める．$E_H^U \cup E_H^L$ の辺の長さは全て 1 とする．E_H^B の各辺 $(l \to L)$ については，l と L が同一高階辺の上下水平線分で，互いに独立であるときのみ長さを 0 とし，それ以外の場合は長さを 1 とする．図 5.11(b) では，長さが 1 の辺の長さは全て省略している．

制約 1, 2 より，E_H^B の各辺 $e = (l \to L)$ に対して，その長さが 1 であれば，直交描画において水平線分 l を L より上に配置する必要がある．一方，e の長さが 0 であれば，水平線分 l を L より上に置くか，l と L に同じ y 座標をもたせる必要がある．

有向グラフ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，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．ステップ 3 では，以下の問題を発見的手法により解く．

［問題］複合双方向グラフ H の最大重み最小帰還辺集合 E_R のうち，$H - E_R$ における最長道の長さが最小となるものを求めよ． □

図 5.11(b) の複合双方向グラフ H の場合，この問題の最適解の一つは

$$E_R = \{(l_2 \to l_1),\ (l_2 \to l_3),\ (L_2 \to L_1), (L_3 \to L_2), (L_3 \to L_4),\ (L_4 \to L_2)\}$$

である．この E_R に対する $H - E_R$ を図 5.13(a) に示すのグラフに残った辺の重みの総和は 8 であり，最長道（例えば $[l_1, l_2, L_2, L_5, L_3]$）の長さは 4 である．残った辺の向きに従い，**5.6.4.3** で後述する方法によって水平線分群とそれらの y 座標を定めることにより，最終的に，図 5.13(b) のように，辺交差数が 8，水平線分群数が (最長道長) $+ 1 = 5$ の直交描画を得ることができる．

5.6.4.3　**水平線分群と順序の決定**

以下では，**5.6.4.2** で示した問題に対する発見的手法（MWFAS と呼ぶ）を提案した後，水平線分群とそれらの y 座標を決める方法について述べる．

最小帰還辺集合問題に対する文献 [19] の発見的手法をもとにして，寺本ら [20] は，全ての辺の長さが 1 である有向グラフ (V', E') が与えられたときに，要素数ができるだけ少ない帰還辺集合 F のうち，グラフ $(V', E' - F \cup \{(v \to w) \mid (w \to v) \in F\})$ の最長道長ができるだけ小さくなるものを求めるという方法を示している．MWFAS は文献 [20] の方法をもとにしたものであり，グラフ $H - E_R$ の最長道長を小さく抑えるために，文献 [20] と同様の工夫を用いている．

5.6.4.2 で定義した複合双方向グラフ H に関して，次の補題が成立する．

補題 5.1 複合双方向グラフ H の任意の最小帰還辺集合 E_R に関して，次の (i), (ii) が成立する．

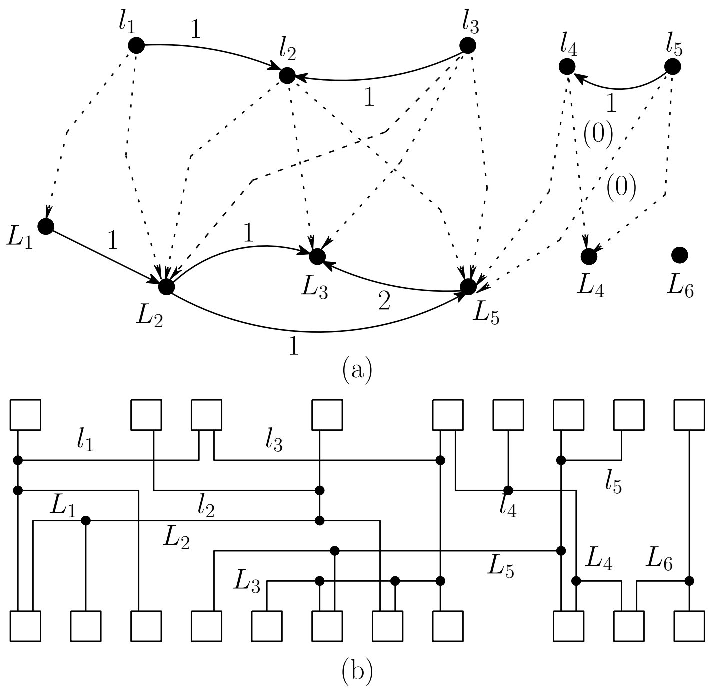


図 5.13: グラフ $H - E_R$ とそれから得られる直交描画

(i) $E_R \cap E_H^B = \emptyset$.
(ii) E_R は，各逆並行辺対からちょうど 1 本の辺を含む.

（証明）H には V_H^L の頂点から V_H^U の頂点に向かう辺がないので，(i) は明らかである．任意の逆並行辺対 p に対して，E_R は p 中の辺を少なくとも 1 本は含む．ある逆並行辺対 $p = \{(l \to l'), (l' \to l)\}$ に対して，E_R が p の両方の辺を含むものと仮定する．グラフ $H - E_R$ はアサイクリックである．このグラフが頂点 l から l' への有向道をもつ場合，$H - E_R$ に辺 $(l \to l')$ を加えてもアサイクリックであるから，$E_R - \{(l \to l')\}$ が H の帰還辺集合になる．逆に，$H - E_R$ が l から l' への有向道をもたない場合には，$E_R - \{(l' \to l)\}$ が H の帰還辺集合になる．いずれの場合も E_R が最小帰還辺集合であることに矛盾するから，E_R は p 中の辺をちょうど 1 本含む. □

系 5.1 H の任意の最小帰還辺集合は $|E_H^U \cup E_H^L|/2$ 本の辺を含む. □

アルゴリズム MWFAS は，E_R を空集合とした後，それが H の最小帰還辺集合になるまで辺を加えていく．それと共に，E_R に加えないと決めた辺を，別の集合 E^X に加えていく（グラフ (V_H, E^X) は常にアサイクリックであるようにする)．最初 $E^X = E_H^B$ とし，$E_H^U \cup E_H^L$ 中のある辺 $(l \to l') \in$ を E_R に加えたときには $(l' \to l)$ を E^X に加える.

MWFAS は，E_R, E^X 以外に，あるグラフ H' を管理する．H' は，最初 $H - E_H^B$ とし，(図 5.11(b) 中の L_5 のような）孤立点をもつ場合にはそれを削除する．その後，H' のあ

る頂点 v^* を選択し，$E^-_{H'}(v^*)$ の辺を E_R に加え，$E^+_{H'}(v^*)$ の辺を E^X に加える．そして，$H' \leftarrow H' - \{v^*\}$ とし，孤立点ができれば削除する．このような処理を，$H' = (\emptyset, \emptyset)$ となるまで繰返す．

MWFAS 実行中の各時点において，H' の頂点のうち，(射入辺の重みの総和) − (射出辺の重みの総和) が最大のものを重み差最大点と呼ぶ．MWFAS では頂点 v^* の射入辺を E_R に加えるので，E_R の辺の重みの総和を大きくするために，重み差最大点の集合 P を求め，その中から v^* を選択する．

この選択を行う際には，グラフ (V_H, E^X)（最終的に $H - E_R$ になる）の最長道長，及び各頂点を含む有向道の長さの最大値をできるだけ増やさないことを考える．そのために，文献 [20] と同様に，各頂点 v に対して二つの値 f_v, b_v を保持する．MWFAS 実行中の各時点の (V_H, E^X) において，f_v はソースから v への有向道の長さの最大値である（グラフ (V_H, E^X) はアサイクリックであるから，v がソースであるか，あるソース w（$\neq v$）から v への有向道が存在する）．また b_v は v からシンクへの有向道の長さの最大値である．f_v を v の F ランク，b_v を B ランクと呼び，$f_v + b_v$ をランク和と呼ぶ．また，全頂点のランク和のうちの最大値を最大ランク和と呼ぶ．(V_H, E^X) において，各頂点 v のランク和は v を通る有向道の長さの最大値であり，最大ランク和は最長道長である．各頂点の F ランク及び B ランクの計算と更新は，文献 [20] に述べられているのと同様の方法により行う．

P の中から v^* を選択する際には，各頂点 $v \in P$ について，それを v^* とした後の最大ランク和を計算し，その値が最小となる頂点の集合 P' を求める．そして，（文献 [20] で示されている方法 C と同様，）P' の中で F ランクが最小の頂点のうち，B ランクが最大のものを v^* とする．

図 5.14 に，アルゴリズム MWFAS の概略を示す（F ランク及び B ランクの計算に関する部分は省略している）．この記述において，deliv とは，H' から全ての孤立点を削除する手続きである．

(1) $E_R \leftarrow \emptyset$，$E^X \leftarrow E^B_H$ とする．
(2) $H' \leftarrow H - E^B_H$ とし，deliv を実行する．
(3) $H' = (\emptyset, \emptyset)$ となるまで，次の (3a)∼(3d) を実行する．
- (3a) $P \leftarrow$ (H'中の重み差最大点全てからなる集合) とする．
- (3b) 集合 P' を求め，P' の中で F ランクが最小の頂点のうち，B ランクが最大のものを v^* とする．
- (3c) $E_R \leftarrow E_R \cup E^-_{H'}(v^*)$，$E^X \leftarrow E^X \cup E^+_{H'}(v^*)$ とする．
- (3d) $H' \leftarrow H' - \{v^*\}$ とし，deliv を実行する．

図 5.14: アルゴリズム MWFAS の概略

補題 5.2 アルゴリズム MWFAS が求める集合 E_R は，複合双方向グラフ H の最小帰還辺集合である．

（証明）H がもつ任意の有向閉路を C とする．C は E_H^B の辺を含まないから，C 上の全ての辺は最初 H' に含まれている．MWFAS の終了時に $H' = (\emptyset, \emptyset)$ となること，及び，H' から辺を削除する操作が図 5.14(3d) の $H' \leftarrow H' - \{v^*\}$ のみであることを考えると，MWFAS の実行中に C 上のある頂点が v^* として選択され，C 上の少なくとも 1 本の辺が E_R に加えられることが分かる．よって，E_R は H の帰還辺集合になる．MWFAS は，ある辺 $(l \to l') \in E_H^U \cup E_H^L$ を E_R に加えたとき，$(l' \to l)$ を E^X に加える．よって最終的に $|E_R| = |E_H^U \cup E_H^L|/2$ となるから，系 1 より，E_R は H の最小帰還辺集合になる．□

MWFAS の実行終了時の (V_H, E^X) における最大ランク和に 1 を加えた値を k とする．以下では，この時点の各頂点のランクの値に基づいて，水平線分を k 個の水平線分群に分割する方法を説明する．上水平線分は上階層 V_i^* の近くに，下水平線分は下階層 V_{i+1}^* の近くに置いた方が，垂直線分の長さの総和が短くなって望ましい．このことも考慮して，次のように，水平線分の集合 HS_j（$j = 1, 2, \ldots, k$）を作成する．

$$HS_j = \{l \in US_i \mid f_l = j - 1\} \cup \{L \in LS_i \mid b_L = k - j\}. \tag{5.10}$$

次の二つの補題が成立する．

補題 5.3 2 本の上水平線分 l, l' が互いに独立でないならば，それらは $HS_1, HS_2, \ldots, HS_k$ のうちの異なる集合に属する．互いに独立でない下水平線分 L, L' についても同様である．

（証明）上水平線分 l, l' が独立でないならば，複合双方向グラフ H は頂点 l, l' 間に逆並行辺対をもち，MWFAS 実行終了時の (V_H, E^X) においても，l と l' の間には長さ 1 の辺が残る．よって，F ランクの定義より $f_l \neq f_{l'}$ となるから，l, l' は $HS_1, HS_2, \ldots, HS_k$ のうちの異なる集合に属する．互いに独立でない下水平線分 L, L' についても同様である．

□

補題 5.4 上水平線分 l が HS_j に，下水平線分 L が $HS_{j'}$ に属するものとする．l と L が独立でなければ $j < j'$ である．また，l と L が同一高階辺の水平線分であり，互いに独立であるならば，$j \leq j'$ である．

（証明）l, L が独立でないならば，MWFAS 実行終了時の (V_H, E^X) において，長さ 1 の辺 $(l \to L)$ が存在する．よって $f_l + 1 \leq f_L$ である．$j = f_l + 1$ より，

$$j \leq f_L \tag{5.11}$$

が成立する．最大ランク和が $k - 1$ であることより，$f_L + b_L \leq k - 1$ である．$b_L = k - j'$ であるので，

$$f_L \leq j' - 1 \tag{5.12}$$

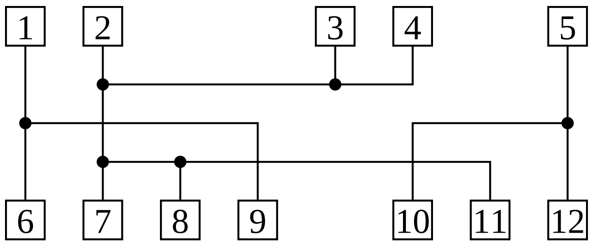


図 5.15: 垂直線分の移動による辺交差数が削減できる例

が成立し，式 (5.11) より $j < j'$ が導かれる．

l と L が同一高階辺の水平線分であり，互いに独立であるならば，辺 $(l \to L)$ の長さは 0 であるから，$f_l \leq f_L$ となる．$f_l = j-1$ であるから $j-1 \leq f_L$ となり，式 (5.12) より $j \leq j'$ が導かれる． □

補題 5.3, 5.4 より，次の定理が導かれる．

定理 5.2 $j = 1, 2, \ldots, k$ について HS_j は独立線分集合である．さらに，$HS_1, HS_2, \ldots,$ HS_k を水平線分群とし，この順に上から下へと配置したものは制約 1, 2 を満たす． □

階層 V_i^* の頂点の y 座標を Y_i，階層 V_{i+1}^* の頂点の y 座標を $Y_i + Dist$ とする．定理 5.2 後半に従い，$j = 1, 2, \ldots, k$ について，HS_j の各水平線分の y 座標を次のように定める．

$$Y_i + \frac{W_R}{2} + j \cdot \frac{Dist - W_R}{k+1}.$$

5.6.5 **ステップ 4**

各高階辺 $e \in E_i^{HP}$ について，上下水平線分をつなぐ垂直線分の仮の x 座標 tx_e をステップ 2 で定め，それを用いて，ステップ 3 で各水平線分の y 座標を決定した．しかし，この仮の x 座標を変更することにより，辺交差数を削減できる場合がある．例えば図 5.15 では，高階辺 $\{2, 3, 4, 7, 8, 11\}$ の垂直線分の仮の x 座標を，頂点 3 の x 座標と同じにすることによって，辺の交差を 1 減らすことができる．

ステップ 4 では，E_i^{HP} の高階辺 e を任意の順に見ていき，tx_e の値を変更することによって辺交差数を削減できるか否かを調べ，削減できる場合にのみそのような変更を行う．その際，水平線分の y 座標は変更せず，水平線分の延長も行わないものとする．また，tx_e は，注目している高階辺 e に含まれるいずれかの頂点の x 座標と同じ値にする．

5.6.6 ステップ 5

ステップ 5 では，各頂点に接続する垂直線分の x 座標を決定する．これにより，各水平線分の左右の端点の実座標が決まることになる．各高階辺 $e \in E_i^{HP}$ の上下水平線分をつなぐための垂直線分に関しては，x 座標が tx_e の頂点と e の水平線分をつなぐ線分が存在するから，それと同じ x 座標を与える．以上により，V_i^* と V_{i+1}^* の間の描画が完成する．

集合 XS_i, VL_t（$t \in XS_i$）及び垂直線分のタイプを，**5.6.4.1** で述べたように定義する．任意の垂直線分 $vl \in VL_t$ に対し，上側の端点の y 座標を $top(vl)$，下側の端点の y 座標を $bottom(vl)$ と表し，閉区間 $[top(vl), bottom(vl)]$ を CI_{vl} と表すことにする．更に，任意の $vl, vl' \in VL_t$ に対し，$CI_{vl} \cap CI_{vl'} = \emptyset$ であるとき，vl と vl' は独立であるということにする．互いに独立でない垂直線分には同じ x 座標を割当てることはできない．

ステップ 5 では，各 $t \in XS_i$ に対し，定理 5.1 の証明を考慮して，水平線分同士，垂直線分同士の重なりが起こらないように VL_t の垂直線分を並べる．まず，互いに独立でなく，タイプが異なる 2 本の垂直線分は，タイプの値がより小さいものを左に置く．互いに独立でなく，タイプが同じである垂直線分の順序は，そのタイプが 1, 3 あるいは 5 であれば，辺の交差数が少なくなるように定める．一方，タイプが 2 あるいは 4 である場合には，定理 5.1 の証明で述べたように，接続する水平線分の y 座標に従って順序を定める．これらのことに加え，垂直線分に割当てる x 座標の個数を少なくすることも考える．そのために，各 $t \in XS_i$ ごとに有向グラフ $H_t = (V_{H_t}, E_{H_t})$ を作成し，**5.6.4.3** で示したアルゴリズム MWFAS を適用する．

グラフ H_t は，VL_t の垂直線分に一対一に対応した頂点をもつ．簡単のため，H_t の各頂点を，対応する垂直線分と同じ記号で表すことにする．異なる垂直線分 vl, vl' が互いに独立なとき，それらのタイプに関わらず，頂点 vl, vl' 間に辺を設けない．vl, vl' が独立でないときには，以下のようにして辺を作る．

(i) $type(vl) < type(vl')$ である場合，辺 $(vl \to vl')$ を作る．この辺の重みは（辺交差の回数に関係なく）0 とする．図 5.16(a) に例を示す．

(ii) $type(vl) = type(vl')$ かつこれらの値が 1, 3, 5 のいずれかである場合，2 辺 $(vl \to vl')$, $(vl' \to vl)$ を作る．$(vl \to vl')$ には，vl を vl' より左に置いたときに，vl, vl' 及びそれらに接続する水平線分が作る辺交差の個数を重みとして付ける．$(vl' \to vl)$ の重みは，vl を vl' より右に置いたときの辺交差の個数とする．vl, vl' が共にタイプ 1 であるときの例を図 5.16(b) に示す．

(iii) $type(vl) = type(vl') = 2$ である場合，接続する下水平線分がより上にある垂直線分に対応する頂点から，もう一方の頂点に向かう辺を作る．図 5.16(c) に例を示す．$type(vl) = type(vl') = 4$ である場合には，接続する上水平線分がより下にある垂直

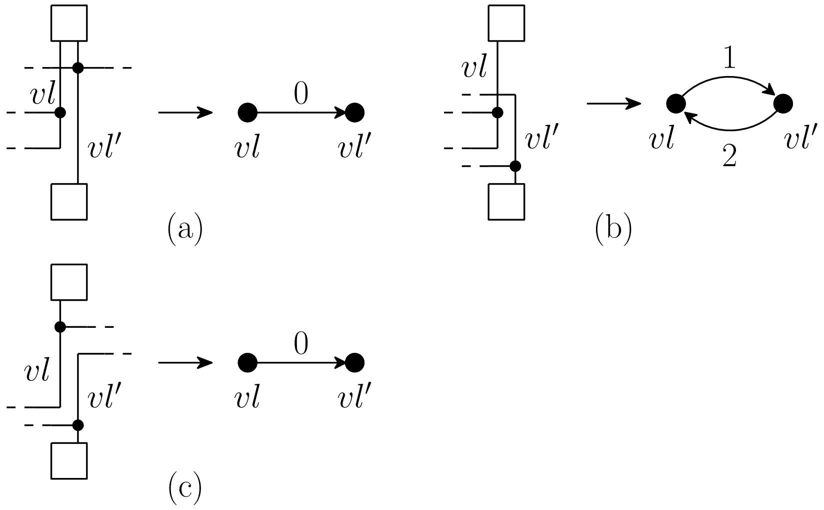


図 5.16: グラフ H_t の辺の作成

線分に対応する頂点から，もう一方の頂点に向かう辺を作る．これらの辺の重みは（辺交差の個数に関係なく）0 とする．

H_t の全ての辺の長さを 1 とする．上記の (i) 及び (iii) で作った辺全てからなる集合を $E^B_{H_t}$ とする．H_t では，$type(vl) > type(vl')$ なる辺 $(vl \to vl')$ は存在しない．よって，次の (a) が成立する．

(a) H_t には $E^B_{H_t}$ の辺を含む有向閉路は存在しない．

この (a) を用いれば，補題 5.1 の証明と同様の議論により，次の (b) が導かれる．

(b) H_t の任意の最小帰還辺集合は $|E_{H_t} - E^B_{H_t}|/2$ 本の辺を含む．

H を H_t，E^B_H を $E^B_{H_t}$ とそれぞれ読み替えて図 5.14 のアルゴリズム MWFAS を実行することにより，H_t の帰還辺集合 E_R で，$H_t - E_R$ の辺の重みの和，及び $H_t - E_R$ における最長道長が小さいものが得られる．この E_R が H_t の最小帰還辺集合であることは，上記の (a), (b) 及び補題 5.2 の証明と同様の議論により証明できる．

MWFAS の実行終了時の (V_{H_t}, E^X) における最大ランク和に 1 を加えた値を k' とする．この時点の各頂点の F ランクの値に基づいて，VL_t を k' 個の部分集合 $VLS_1, VLS_2, \ldots, VLS_{k'}$ に分割する．$j = 1, 2, \ldots, k'$ について，VLS_j は，F ランクの値が $j-1$ の頂点に対応する垂直線分からなる集合とする．**5.6.4.3** に示した補題 5.3 と同様，次の補題が成立する．

補題 5.5 2 本の垂直線分 $vl, vl' \in VL_t$ が互いに独立でないならば，それらは $VLS_1, VLS_2, \ldots, VLS_{k'}$ のうちの異なる集合に属する． □

最後に，x 座標が t の頂点の幅の範囲内で k' 通りの x 座標を定める．そして，各 VLS_j に属する垂直線分のそれぞれに，左から j 番目の x 座標を与える．

5.7 計算機実験

提案法 3 の有効性を確認するために計算機実験を行った．提案法 3 では，第 5 段階のステップ 3 において，複合双方向グラフ H の最大重み最小帰還辺集合 E_R のうち，$H - E_R$ における最長道の長さが最小となるものを求める問題を発見的手法 MWFAS により解いている．この部分において，複合双方向グラフ H の最大重み最小帰還辺集合を厳密解法により求める場合との比較実験も行った．以下，**5.7.1** において，その厳密解法について述べ，その後，**5.7.2** において提案法 3 の評価実験の方法と結果について述べる．

5.7.1 複合双方向グラフの最大重み最小帰還辺集合を求める厳密解法

各辺が非負の重みをもつ双方向グラフ $G_B = (V_B, E_B)$ が与えられるものとする．$|E_B| = 2z$ とし，$E_B = \{e_1, e_2, \ldots, e_{2z}\}$ であるものとする．$j = 1, 2, \ldots, 2z$ に対し，辺 e_j の重みを wt_j と表す．$j = 1, 2, \ldots, z$ 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．

G_B の最大重最小帰還辺集合を求めるための発見的手法として，アルゴリズム MWFAS と同様に，重み差最大の頂点の選択を繰返すという方法が考えられる．以下に，そのようなアルゴリズム find_FAS を示す．

［アルゴリズム find_FAS］

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

これ以降，双方向グラフ G_B の最大重み最小帰還辺集合を求める問題に対して，最適解を求めるアルゴリズム Opt を提案する．アルゴリズム Opt は，G_B に対し，E_B の部分集合で，各逆平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} を計算し，G_B の最大重み最小帰還辺集合として $E_B - F_{min}$ を出力する．

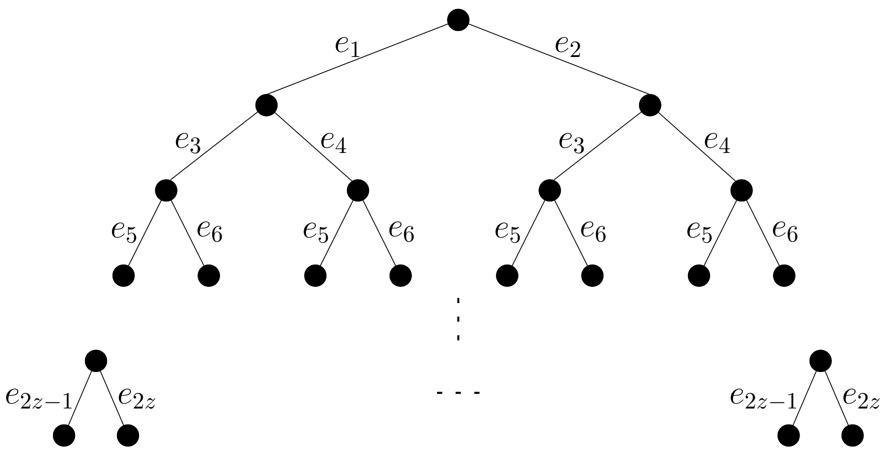


図 5.17: 探索木

Opt は，簡単な分枝限定法 [25] によるものである．図 5.17 に，Opt が用いる探索木を示す．この木の高さは z であり，各葉の深さも z である．図 5.17 中に示したように，$j = 0, 1, \ldots, z-1$ 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)[26] で構成していく．

Opt は，まず前処理として，次の (a), (b) を実行する．

(a) 各整数 $j = 1, 2, \ldots, z$ について，

$$wt_{2j-1} = 0,\ wt_{2j} = wt_{2j} - wt_{2j-1} \tag{5.13}$$

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_B に発見的手法 find_FAS 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，$W_{min} = 0$ であれば，find_FAS の求めた集合 X が最適解（G_B 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_B 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_B 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_B の）辺が $e_k = (v \to w)$ 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，分枝限定法により双方向グラフの最大重み最小帰還辺集合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．複合双方向グラフ H に対して最大重み最小帰還辺集合の最適解を求めるためには，H の双方向グラフ (V_H^L, E_H^L)，(V_H^U, E_H^U) それぞれに対して Opt を実行し，得られた集合の和集合を出力すればよい．

5.7.2 提案法 3 の評価実験

ランダムに作成した単純な階層グラフ $G = (V, E)$ を用いて計算機実験を行った．G の階層数 h，頂点数 $|V|$，辺数 $|E|$ の組合せは $(h, |V|, |E|) = (2, 40, 40)$, $(2, 40, 60)$, $(8, 40, 80)$, $(8, 40, 120)$ の 4 通りとし，それぞれについて 200 個のグラフを用意した．このようなデータに対し，まず提案法 3 と次の方法 1 を実行した．

方法 1： 提案法 3 の第 1, 3, 4 段階を実行した後，各辺を直線で描く方法.

更に，提案法 3 の第 5 段階のステップ 3 で行っている処理の効果を確認するため，以下の三つの方法も実行した.

方法 2： 第 5 段階のステップ 3 で v^* を決定する際（図 5.14 の (3b)），H' の頂点の中から任意に選ぶもの.

方法 3： 第 5 段階のステップ 3 で v^* を決定する際，重み差最大点の集合 P の中から任意に選ぶもの.

方法 4： 第 5 段階のステップ 3 でアルゴリズム MWFAS を用いずに，複合双方向グラフ H の最大重み最小帰還辺集合 E_R を厳密解法 Opt を用いて求めたもの．この方法では，$H - E_R$ の最長道の長さは考慮していない.

求めた値は以下の (i)〜(v) である．

(i) 第 1 段階により得られたグラフ G^* の頂点数 $|V^*|$ と辺数 $|E^*|$
(ii) 第 2 段階により得られた高階辺の本数 $|E^{HP}|$
(iii) 各方法で得られた描画における辺交差数
(iv) 各方法で得られた描画における総水平線分群数（方法 1 を除く）
(v) 各方法の実行時間（第 1 段階の開始から描画を得るまでの時間）

使用計算機の 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．

上記の項目のうち，(i), (ii) の結果を表 5.1(a) に，(iii)〜(v) の結果を表 5.1(b) にそれぞれ示す．表中の各実験値は 200 個のグラフに対する平均値である．

表 5.1(b) を見ると，直線描画を求める方法 1 に比べ，提案法 3 は辺交差数を大幅に減らすことができている．

方法 2 と提案法 3 の辺交差数を比較すると，提案法 3 の方が値がかなり小さくなっている．このことから，H' の重み差最大点の集合 P を作り，その要素の中から v^* を選択する処理が有効であることが分かる．

また，方法 3 と提案法 3 の平均水平線分群数を比較すると，提案法 3 の方が若干値がよくなっている．全 800 個のデータに対する結果を個々に調べたところ，提案法 3 による水平線分群数が方法 3 より小さかったデータが 395 個あり，逆に大きかったデータはわずかに 3 個だけであった．これらのことから，(V_H, E^X) の最長道長を考慮して v^* を選択する処理に多少の効果があったことが分かる．

最後に，方法 4 と提案法 3 を比較すると，辺交差数に関して，H の最大重み最小帰還辺集合を厳密に求めている方法 4 の方がよい値になることは明らかであり，実験結果の表からも確認できる．提案法 3 と方法 4 の辺交差数の差はさほど大きくなく，特に辺数が少ないときには差が小さくなっている．また，方法 4 は水平線分群数のことを考慮していないため，提案法 3 による水平線分群数は方法 4 より小さくなっている．方法 4 で用いている Opt は指数時間アルゴリズムであり，データによっては実行時間がかなり長くなることがある．実際，今回の実験でも，階層数 8，頂点数 $|V| = 40$，辺数 $|E| = 120$ のデータの中に，約 38 分の時間がかかったものが存在した．

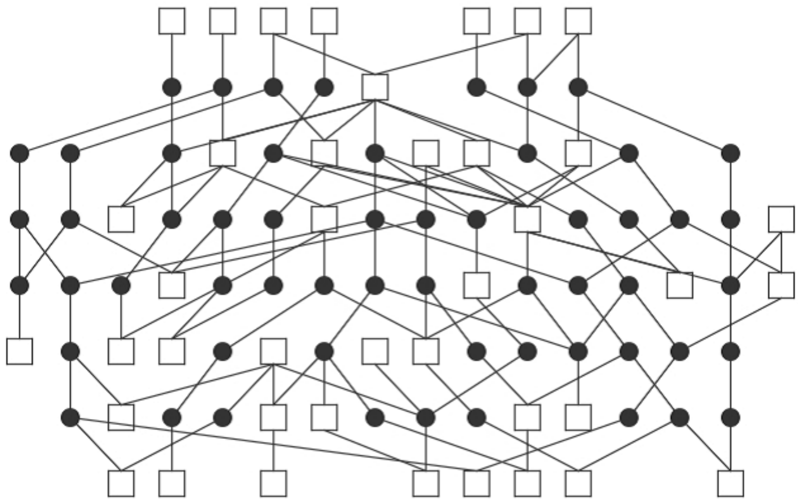
表 5.1: 直交描画に関する実験結果

(a) 高階辺の本数

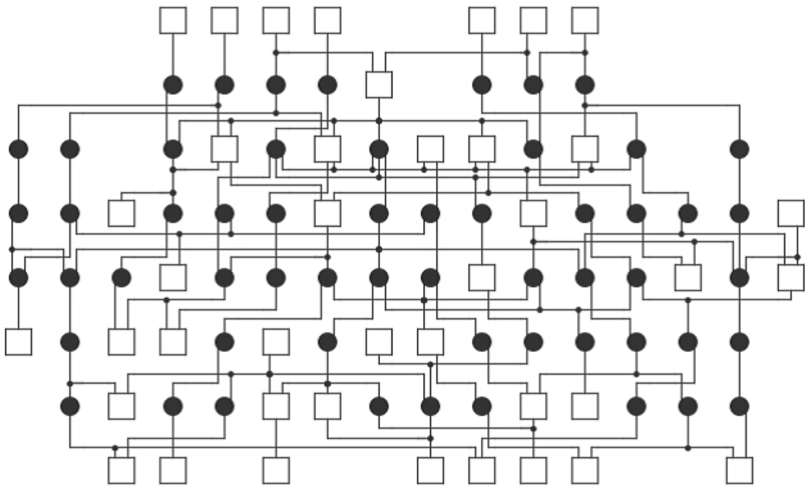
h	$\|V\|$	$\|E\|$	$\|V^*\|$	$\|E^*\|$	$\|E^{HP}\|$
2	40	40	40	40	16.10
2	40	60	40	60	19.01
8	40	80	96.88	133.72	79.08
8	40	120	104.82	175.51	88.68

(b) 各方法の比較

h	$\|V\|$	$\|E\|$	手法	辺交差数	水平線分群数	時間 [ms]
2	40	40	方法 1	46.64	− −	28.32
			方法 2	31.02	4.80	30.85
			方法 3	25.85	4.96	31.93
			方法 4	25.39	5.21	29.43
			提案法 3	25.77	4.65	32.61
2	40	60	方法 1	238.98	− −	31.48
			方法 2	132.17	10.97	36.62
			方法 3	106.75	11.28	39.64
			方法 4	103.52	12.49	532.07
			提案法 3	106.58	10.62	47.01
8	40	80	方法 1	153.33	− −	78.19
			方法 2	131.53	26.36	82.93
			方法 3	109.42	26.49	86.37
			方法 4	107.61	27.70	94.57
			提案法 3	109.44	25.03	91.51
8	40	120	方法 1	381.61	− −	85.09
			方法 2	290.12	39.73	97.66
			方法 3	234.48	40.09	98.00
			方法 4	228.37	42.78	37845.79
			提案法 3	233.89	38.22	102.37



(a) 方法 1



(b) 提案法 3

図 5.18: 方法 1 と提案法 3 の描画例

階層数 8，$|V| = 40$，$|E| = 80$ の階層グラフに対する描画例を図 5.18 に示す．図 5.18(a) は方法 1, (b) が提案法 3 の描画であり，辺交差数は (a) が 103，(b) が 77 である．

5.8 結言

本章では，提案法 3 として階層グラフの直交描画アルゴリズムを示し，高階辺の決定方法，各頂点の x 座標の決定方法，及び各高階辺を垂直・水平線分を用いて描く方法を提案した．高階辺を垂直・水平線分を用いて描く方法は，辺交差数を少なくすることを主な目

的としたものであるが，水平線分群数を少なく抑えるための工夫もしている．

提案法 3 の各段階を実行する方法について，現在も検討を続けている．特に，第 3 段階に関して直交描画により適した方法を構築すること，及び，第 4 段階に関して，辺交差数削減を考慮した頂点座標決定法を作成することは今後の課題である．

第 6 章

結論

本論文では，グラフの階層描画に関して，動的計画法を用いた新しい頂点座標決定法の提案と，複数の長辺にダミー頂点を共有させることにより，ダミー頂点の数を減らす方法，そして，新しい直交描画アルゴリズムの提案を行った．

第 3 章では，各階層上の頂点の配置順序が指定された階層グラフを描画する際に，頂点の座標を決定する新しい方法を提案した．提案法 1 は，描画中の重み付き辺長の 2 乗の総和を小さくすることを目的としており，グラフの各階層を順に見ていき，それぞれの頂点の配置を動的計画法を用いて求めるものである．優先度法との比較実験を行ったところ，提案法 1 は，近接性，直線性，バランス性という三つの評価基準のすべてに関して，より優れた描画を求めることができた．提案法 1 のさらなる高速化について検討することは今後の課題である．

第 4 章では，階層描画において，複数の辺にダミー頂点の共有をさせる単純なアルゴリズムを提案した．計算機実験を行ったところ，通常の方法でダミー頂点を作成する場合に比べ，提案法 2 は頂点数と辺数を大幅に少なくすることができた．また，ダミー頂点を共有させることにより，最終的に得られる描画の描画幅と辺交差数を大きく削減し得ることを示した．また，グラフの頂点数を最小にする問題，及び辺数を最小にする問題が，いずれも，入力を連結 3 階層グラフに限定しても NP 困難であることを証明した．提案法 2 はこれらの問題に対する発見的手法として用いることができるが，さらに有効なアルゴリズムについて検討を続けている [27].

第 5 章では，提案法 3 として，階層グラフの新しい直交描画アルゴリズムを示した．このアルゴリズムは

第 1 段階：　ダミー頂点の集合の決定
第 2 段階：　高階辺の集合の決定
第 3 段階：　各階層における頂点の配置順序の決定

第 4 段階：　頂点の x 座標の決定
第 5 段階：　各高階辺の各垂直・水平線分の座標の決定と高階辺の描画，高階辺の決定

という 5 つの処理を順に行うものである．第 5 章では，第 2 段階，第 4 段階，第 5 段階について新たな手法を提案した．第 5 段階の手法は，辺交差数を少なくすることを主な目的としたものであるが，水平線分群数を少なく抑えるための工夫も行っている．提案法 3 は直線描画と比べて，大幅に辺交差数を減らすことができることを計算機実験で示した．第 3 段階に関して直交描画により適した方法を構築すること，及び，第 4 段階に関して，辺交差数削減を考慮した頂点座標決定法を作成することは今後の課題である．

謝辞

本研究を進めるにあたり，６年半の長きに渡り熱心な御指導と大変多く知識を御教授いただいた，神戸大学大学院工学研究科，増田澄男教授に心より感謝いたします.

貴重なご意見ならびに御助言を頂きました，神戸大学大学院工学研究科，森井昌克教授に心より感謝いたします.

貴重なご意見ならびに御助言を頂きました，神戸大学大学院工学研究科，八坂保能教授に心より感謝いたします.

本研究を進めるにあたり，有意義な討論と御指導をいただいた，神戸大学大学院工学研究科，山口一章准教授に心より感謝いたします.

本研究を進めるにあたり，有意義な討論と御指導をいただいた，神戸大学大学院工学研究科，斎藤寿樹助教に心より感謝いたします.

また，様々な御指導，御指摘や助言をしていただいた，アルゴリズム研究室の院生，ならびに学部生の方々に深く感謝いたします.

参考文献

[1] G. D. Battista, P. Eades, R. Tamassia, and Ioannis G. Tollis, "Graph Drawing - Algorithms for the Visualization of Graphs" , Prentice-Hall, Upper Saddle River, NJ, 1999.

[2] K. Sugiyama, Graph Drawing and Applications - For Software and Knowledge Engineering, World Scientific, Singapore, 2002.

[3] K. Sugiyama, S. Tagawa, and M. Toda, "Methods for visual understanding of hierarchical system structures," IEEE Trans. Systems, Man, and Cybernetics, vol.SMC-11, no.2, pp.109-125, 1981.

[4] O. Bastert and C. Matuszewski,"Layered drawings of digraphs," Drawing Graphs (M. Kaufman and D. Wagner eds.), Lecture Notes in Computer Science, vol.2025, pp.87-120, Springer, Berlin, 2001.

[5] E.R. Gansner, E. Koutsofios, S.C. North and K.-P. Vo, "A technique for drawing directed graphs," IEEE Trans. Software Engineering, vol.19, no.3, pp.214-230, 1993.

[6] U. Brandes and B. Köpf, "Fast and simple horizontal coordinate assignment," Proc. 9th Int'l Symp. on Graph Drawing (P. Mutzel, M. Jünger and S. Leipert eds.), Lecture Notes in Computer Science, vol.2265, pp.31-44, Springer, Berlin, 2002.

[7] C. Buchheim, M. Jünger and S. Leipert, "A fast layout algorithm for k-level graphs," Proc. 8th Int'l Symp. on Graph Drawing (J. Marks ed.), Lecture Notes in Computer Science, vol.1984, pp.229-240, Springer, Berlin, 2001.

[8] M. Spönemann, H. Fuhrmann, R. Hanxleden, and P. Mutzel, "Port constraints in hierarchical layout of data flow diagrams," Proc. 17th Int'l Symp. on Graph Drawing (GD2009), Lecture Notes in Computer Science, vol.5849, pp.135-146, Springer, Berlin, 2010.

[9] G. Sander, "A fast heuristic for hierarchical Manhattan layout," Proc. Symp. on

Graph Drawing (GD'95), Lecture Notes in Computer Science, vol.1027, pp.447-458, Springer, Berlin, 1996.

[10] 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.

[11] T. Eschbach, W. Gunther, and B. Becker, "Orthogonal hypergraph drawing for improved visibility," J. Graph Algorithms and Applications, vol.10, no.2, pp.141-157, 2006.

[12] 荒木徹也, 増田澄男, 山口一章, 村田将太, "階層グラフ描画における頂点座標決定アルゴリズム," 電子情報通信学会論文誌 (A), vol.J94-A, no.12, pp.960-973, 2011.

[13] 的場郁典, 増田澄男, 山口一章, "階層グラフ描画の頂点座標決定法に関する一考察," 平成 24 年電気関係学会関西支部連合大会, 9amT-9, 2012.

[14] S. Pupyrev, L. Nachmanson, and M. Kaufmann, "Improving layered graph layouts with edge bundling," Proc. 18th Int'l Symp. on Graph Drawing (GD2010), Lecture Notes in Computer Science, vol.6502, pp.329-340, Springer, Berlin, 2011.

[15] 田守健太郎, 山口一章, 増田澄男, "局所探索法による階層的描画の辺交差数削減," 電子情報通信学会論文誌 (A), vol.J92-A, no.1, pp.55-61, 2009.

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

[17] R. Peeters, "The maximum edge biclique problem is NP-complete," Discrete Applied Mathematics, vol.131, pp.651-654, 2003.

[18] U.L. Gupta, D.T. Lee, and J.Y.-T. Leung, "Efficint algorithms for interval graphs and circular-arc graph," Networks, vol.12, pp.459-467, 1982.

[19] 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.

[20] 寺本正幸, 増田澄男, 山口一章, "有向グラフ描画アルゴリズムにおける閉路削除法の改良," 電子情報通信学会論文誌 (A), vol.J92-A, no.6, pp.434-439, 2009.

[21] 松井泰子, "Integer Programming and Gröbner Bases," 数理解析研究所講究録, vol.1175, pp.72-76, 2000.

[22] 荒木徹也, 山口一章, 増田澄男 "グラフの階層直交描画における頂点座標決定アルゴリズム," 平成 23 年度情報処理学会関西支部支部大会講演論文集, B–02, 2011.

[23] P.B. Miltersen, Mixed Integer Linear Programs, http://www.daimi.au.dk/dOpt/ilp.pdf

[24] 宮代隆平, "整数計画ソルバー入門," オペレーションズ・リサーチ, vol.57, no.4, pp.183-189, 2012.

[25] 平田富夫, アルゴリズムとデータ構造 – 改訂 C 言語版，森北出版，1990.

[26] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

[27] 堀尾明久, 荒木徹也, 増田澄男, 山口一章, “階層グラフ描画におけるダミー頂点共有化処理の改良,” 平成 24 年電気関係学会関西支部連合大会, 9amT-11, 2012.

関連発表

学術論文誌

[1] 荒木徹也, 増田澄男, 山口一章, “階層グラフ描画におけるダミー頂点の共有,” 電子情報通信学会論文誌 (A), vol.J94-A, no.12, pp.950-959，2011.

[2] 荒木徹也, 増田澄男, 山口一章, 村田将太, “階層グラフ描画における頂点座標決定アルゴリズム,” 電子情報通信学会論文誌 (A), vol.J94-A, no.12, pp.960-973，2011.

[3] 荒木徹也, 増田澄男, 的場郁典, 山口一章, 斎藤寿樹, “階層グラフの直交描画アルゴリズム,” 電子情報通信学会論文誌 (A)，投稿中 (条件付採録).

研究機関の紀要

[1] 荒木徹也, 増田澄男, 斎藤寿樹, 山口一章, “双方向グラフの最大重み最小帰還辺集合問題について,” 神戸大学大学院工学研究科・システム情報学研究科紀要，投稿中.

学術講演

[1] 荒木徹也, 増田澄男, 山口一章, “階層グラフ描画アルゴリズムにおける辺の形状の決定法,” 平成 20 年電気関係学会関西支部連合大会, G10-9, 2008.

[2] 荒木徹也, 増田澄男, 山口一章, “グラフの階層的直交描画アルゴリズムの改良,” 平成 21 年電気関係学会関西支部連合大会, G10-10, 2009.

[3] 荒木徹也, 山口一章, 増田澄男, 村田将太, “階層グラフ描画における頂点座標決定アルゴリズム,” 平成 22 年度情報処理学会関西支部支部大会, B-01, 2010.

[4] 村田將太, 荒木徹也, 増田澄男, 山口一章, “頂点が幅をもつ階層グラフ描画の頂点座標決定法,” 平成 22 年電気関係学会関西支部連合大会, 3A303-16, 2010.

[5] 荒木徹也, 山口一章, 増田澄男, “グラフの階層直交描画における頂点座標決定アルゴリズム,” 平成 23 年度情報処理学会関西支部支部大会, B-02, 2011.

[6] 荒木徹也, 増田澄男, 山口一章, “グラフの階層直交描画における線分座標決定,” 平成 24 年度情報処理学会関西支部支部大会, B-03, 2012.

[7] 堀尾明久, 荒木徹也, 増田澄男, 山口一章, “階層グラフ描画におけるダミー頂点共有化処理の改良,” 平成 24 年電気関係学会関西支部連合大会, 9amT-11, 2012.

[8] 荒木徹也, 増田澄男, 的場郁典, 山口一章, 斎藤寿樹 “階層グラフの直交描画アルゴリズム,” 電子情報通信学会コンピュテーション研究会, COMP13-7, 2013.