



ラベルをもつ無向グラフの描画アルゴリズムに関する研究

阿部, 昇

(Degree)

博士 (工学)

(Date of Degree)

2005-03-25

(Date of Publication)

2008-05-22

(Resource Type)

doctoral thesis

(Report Number)

甲3236

(URL)

<https://hdl.handle.net/20.500.14094/D1003236>

※ 当コンテンツは神戸大学の学術成果です。無断複製・不正使用等を禁じます。著作権法で認められている範囲内で、適切にご利用ください。



博士論文

ラベルをもつ無向グラフの
描画アルゴリズムに関する研究

平成17年1月

神戸大学大学院自然科学研究科

阿部 昇

内容梗概

本論文では、頂点や辺がラベルをもつ無向グラフの描画に関する2つの問題について考察する。グラフは様々な構造を表現するために利用可能であり、グラフの適切な描画は、その構造を把握するのに非常に有効である。しかし、グラフ描画を人手で作成することは、一般に手間のかかる作業である。そのため、近年、グラフの自動描画法について多くの研究がなされている。

G を連結無向グラフとし、その各頂点あるいは各辺が、あらかじめ決められたサイズのラベルをもつものとする。このようなグラフを描くための方法として、従来、まずラベルの情報を特に考慮せずにグラフ描画を定め、次にその描画中にラベルを配置するというアプローチ（分離型アプローチと呼ぶ）が主に検討されてきた。また、描画中にラベルを配置する方法としては、最初に各頂点あるいは各辺に対してラベルを配置する位置の候補（ラベル候補と呼ぶ）を多数作成し、それらの中から最終的なラベル位置を選択するという方法が主流であった。

本論文の前半では、分離型アプローチを用いるものとして、 G の描画 Γ が与えられたとき、どのラベルも頂点や辺及び他のラベルと重ならないようにして、できるだけ多くのラベルを Γ 中に配置するという問題について考える。

この問題に対し様々な研究が行われてきているが、最近、Kakoulis-Tollis は、まず各頂点あるいは各辺のラベル候補を複数個定め、次にそれらの中から最終的なラベル位置を選択するというアルゴリズムを提案した。この方法は高いラベル配置率を達成できるものとされている。しかし、周囲にある頂点や辺の位置を特に考慮せずに、機械的にラベル候補を定めるため、実際にはラベルを配置するのに適した位置があっても、それを見逃してしまうことがあるという問題点がある。また、各ラベル候補から最終的なラベルの配置位置を選択する部分にも改善の余地があると考えられる。よって本研究では、ラベル配置率の改善を主な目的として、Kakoulis-Tollis のラベル配置法の改良を行う。まず、グラフの辺のみあるいは頂点のみがラベルをもつ場合について、以下のような特徴をもったアルゴリズムを提案する。

- (1) 周囲の頂点や辺の状況を考慮してラベル候補を作成する。
- (2) 各ラベル候補の位置を最初に決めたものに固定するのではなく、他のラベル候補の位置を考慮して移動することを許す。

そして、これらの方法によるラベル配置率が、Kakoulis-Tollis の方法より大幅に高い値になること、及び実験的に求めた（緩い）上限値に近い値になることを示す。

グラフの頂点と辺の両方がラベルをもつ場合については、辺のみあるいは頂点のみがラベルをもつ場合のラベル配置法を組合せ、それに改良を加えたアルゴリズムを提案する。グラフ描画中にラベルを配置する問題に深く関連した問題として、地名などを表すラベルを地図中に配置する問題があり、Wagner らによる方法、Kameda-Imai による方法などが知られている。本研究では、これらの地図ラベル配置アルゴリズムをグラフ描画へのラベル配置に適用できるよう変更し、それらの方法と比較しても、提案手法がより高いラベル配置率を達成できることを計算機実験により示す。

分離型アプローチでは、グラフの描画を求める段階でラベルの存在を考慮していないため、得られる描画において、ラベルを配置したい頂点や辺の周囲に十分な領域が確保されとは限らない。このため、特にグラフが密である場合やラベルのサイズが大きい場合には、配置できないラベルが多く残ることがある。よって、本論文の後半では、各頂点あるいは各辺のラベルのサイズを考慮しながら頂点や辺の位置を決定する描画法を提案する。提案手法は、頂点や辺がラベルをもたないグラフを対象として開発された川西らのグラフ描画アルゴリズムをもとにしている。この方法はいわゆる力指向アプローチによるもので、頂点間に働く力及び頂点と辺の間に働く力を定義し、各頂点をそれに働く力に従って移動させるという処理を繰り返すものである。これらの力に加え、提案手法では、各ラベルに対してその想定位置を表す仮頂点を導入し、仮頂点とその付近の辺などに対して、ラベルの大きさを考慮した斥力を働かせている（この斥力は、ラベルの配置位置を確保するためのものである）。そして、この方法で得られる描画に対し、前半で提案したラベル配置アルゴリズムを適用することによって、ラベル配置率がさらに改善されることを計算機実験により示す。

目次

第1章	緒論	1
第2章	従来の方法とその問題点	4
2.1	緒言	4
2.2	グラフ描画アルゴリズム	4
2.2.1	Eades の描画アルゴリズム	4
2.2.2	Kamada-Kawai の描画アルゴリズム	5
2.2.3	川西らの描画アルゴリズム	6
2.3	ラベル配置アルゴリズム	7
2.3.1	Kamada のラベル配置アルゴリズム	7
2.3.2	Kakoulis-Tollis のラベル配置アルゴリズム	7
2.3.3	Wagner らの地図ラベル配置アルゴリズム	9
2.3.4	Kameda-Imai の地図ラベル配置アルゴリズム	11
2.4	従来の方法の問題点	12
2.4.1	ラベル配置アルゴリズムに関する問題点	12
2.4.2	分離型アプローチの問題点	12
2.5	結言	13
第3章	グラフ描画へのラベル配置アルゴリズム	14
3.1	緒言	14
3.2	辺のみがラベルをもつ場合のラベル配置アルゴリズム	14
3.2.1	アルゴリズムの説明	14
3.2.2	計算機実験	20
3.3	頂点のみがラベルをもつ場合のラベル配置アルゴリズム	23
3.3.1	アルゴリズムの説明	24
3.3.2	計算機実験	28
3.4	頂点と辺の両方がラベルをもつ場合のラベル配置アルゴリズム	30
3.4.1	提案手法 3-1	30
3.4.2	提案手法 3-2	31

3.4.3	計算機実験	32
3.5	結言	35
第4章	ラベルの存在を考慮したグラフ描画アルゴリズム	36
4.1	緒言	36
4.2	辺ラベルの存在を考慮したグラフ描画アルゴリズム	36
4.2.1	アルゴリズムの説明	37
4.2.2	計算機実験	50
4.3	頂点ラベルの存在を考慮したグラフ描画アルゴリズム	53
4.3.1	アルゴリズムの説明	54
4.3.2	計算機実験	63
4.4	結言	66
第5章	結論	67
	謝辞	69
	参考文献	70
	関連発表	73

目 次

図 1.1	グラフの描画とラベル付き描画の例	2
図 2.1	頂点と辺との間に働く力 ($f''_{i,k} < 0$)	6
図 2.2	KT 法におけるラベル候補の例	8
図 2.3	1つの地点に対するラベル候補の例 ($c = 4$)	10
図 2.4	ルール L1 と L3 の例	10
図 2.5	KT 法により生成される辺のラベル候補 (ラベル候補数 250)	13
図 3.1	提案手法 1 におけるラベル候補の作成法	15
図 3.2	提案手法 1 により生成されるラベル候補 (ラベル候補数 47)	16
図 3.3	辺の削除により 2 つの完全クラスタが作られる例	17
図 3.4	ラベル候補のシフトの例	18
図 3.5	一方のラベル候補のシフトでは重なりを避けられない例	18
図 3.6	ラベル付き描画の例 (辺の本数: 40, ラベルの高さ: 0.20)	24
図 3.7	ラベル空間とラベル候補の作成	25
図 3.8	$CLS(\lambda)$ と $CCLS(\lambda)$	25
図 3.9	ラベル候補の移動	26
図 3.10	手続き $\text{shift}(C)$ の実行例	26
図 3.11	KT 法において作成するラベル候補数を増やした場合の実験結果	29
図 3.12	KI 法におけるラベル空間の修正	32
図 4.1	仮頂点	37
図 4.2	仮頂点の位置の候補	39
図 4.3	cp_k が影響を受ける範囲 H_{cp_k}	40
図 4.4	距離 d の決定	40
図 4.5	力 $F^{(1)}(cp_k, e')$ の決定	41
図 4.6	辺 e' が cp_k と e の間を通過している場合	41
図 4.7	$F^{(1)}(cp_k, e'_1)$ と $F^{(1)}(cp_k, e'_2)$ が打ち消しあう例	41
図 4.8	仮頂点の位置の調整のための 2 種類の力	42
図 4.9	$F(i)$ の向きの変更	43

図 4.10	力 F_e	44
図 4.11	力 $F^{(4)}(i, e')$	45
図 4.12	力 $F^{(5)}(i, p_{e'})$	46
図 4.13	頂点 i のまわりの 8 個の点	47
図 4.14	繰り返し回数 M による配置率の変化	50
図 4.15	ラベル付き描画の例 (辺の本数 : 40 , ラベルの高さ : 0.30)	54
図 4.16	仮頂点の位置の候補	56
図 4.17	力 $F^{(1)}(cp_k, e')$	57
図 4.18	力 $F^{(2)}(p_i, e')$	58
図 4.19	力 F' と $F^{(3)}(p_i, p_j)$	58
図 4.20	p_i の位置の調整	59
図 4.21	頂点 i の位置の改善	61
図 4.22	ラベル付き描画の例	66

表 目 次

表 3.1	ラベル候補数の比較	21
表 3.2	配置率の比較	22
表 3.3	実行時間の比較	22
表 3.4	ラベル配置率	28
表 3.5	実行時間	29
表 3.6	ラベル配置率 [%]	33
表 3.7	実行時間 [ms]	34
表 3.8	ラベルを拡大したときの，頂点及び辺との重なりの個数	34
表 3.9	ラベルを拡大したときの，他のラベルとの重なりの個数	34
表 4.1	描画の望ましさに関する評価値の比較	52
表 4.2	実行時間の比較	53
表 4.3	ラベル配置率の比較	53
表 4.4	描画の望ましさに関する評価値の比較	65
表 4.5	実行時間	65
表 4.6	ラベル配置率	65

第1章 緒論

グラフは、様々な構造を表現するために用いることができる。その際、通常、グラフの各頂点はその構造の構成要素に対応し、各辺はそれらの構成要素間にある関係が存在することを示す。各構成要素の名称あるいは属性をグラフ中に明示したいとき、それらは頂点のラベルとして表わされる。一方、構成要素間の関係に関する情報は、辺のラベルとして表現される。

本論文では、グラフの頂点と辺のみを平面上に描いたものをグラフ描画、あるいは単に描画と呼び、これにラベルを配置したものをラベル付き描画と呼ぶことにする。

グラフは、辺の向きの有無により、有向グラフと無向グラフに分けられるが、本研究で扱うグラフは辺に向きのない無向グラフである。さらに、グラフには、頂点が階層構造をもつ階層グラフなどがあるが、本研究では頂点の配置位置に関して特に制約のない一般グラフを扱う。また、グラフを描画する際に、全ての辺を水平あるいは垂直線分で描く直交描画を用いたり、辺を折れ線や曲線で描いたりする方法もあるが、本研究では、辺は角度に制約のない直線で描くものとする。

グラフのサイズが小さい場合を除いて、適切なグラフ描画を手で作成することは一般に非常に時間のかかる作業である。よって、計算機を用いたグラフ描画の自動化を目指して、近年、グラフの描画アルゴリズムに関して広く研究が行われている [1]。頂点や辺がラベルをもたない無向グラフを描画対象としたアルゴリズムとして、Eades [2]、Kamada-Kawai [3]、角ら [4]、川西ら [5] による方法などが知られている。

頂点や辺がラベルをもつ無向グラフのラベル付き描画を求める方法としては、まず、ラベルの情報を特に考慮せずにグラフの描画を求め、次に、その描画の適切な位置にラベルを配置するというアプローチ（分離型アプローチと呼ぶ）が主に検討されてきた。描画中の適切な位置に頂点のラベルを配置する問題については文献 [6] ~ [9] の方法が、辺のラベルを配置する問題については文献 [8] ~ [11] などの方法があげられる。

本論文の前半では、文献 [6] ~ [12] と同様、無向グラフの描画 Γ が与えられ、各頂点あるいは各辺がもつラベルの大きさが指定されているときに、以下の 3 条件を満たすようにして、できるだけ多くのラベルを Γ 中に配置する問題について考察する。

- (a) 頂点のラベルは、付随する頂点からある一定距離 $\epsilon (\geq 0)$ だけ離れている。
- (b) 辺のラベルは付随する辺に接する。

- (c) どのラベルも，頂点を中心とする半径 ϵ の円（付随する辺以外の）辺，あるいは他のラベルと重ならない．

図 1.1(a) にグラフの描画の例を，同図 (b) にラベル付き描画の例を，それぞれ示す（破線は，各頂点を中心とする半径 ϵ の円を示している）．本論文では，ラベルを図示する際，同図 (b) のように軸平行な長方形で表す．



図 1.1: グラフの描画とラベル付き描画の例

頂点に対するラベル配置問題や辺に対するラベル配置問題は，NP 困難であることが知られている [12], [13] . Kakoulis-Tollis [8] は，最近，一般のグラフの描画に対して，頂点，辺及び領域のラベル付けを行うためのアルゴリズムを提案した．この方法は高いラベル配置率（配置できたラベルの数をラベルの総数で割った値）を達成できるものとされているが，頂点や辺のラベルを配置する場合，ラベルの配置位置の候補を設定する方法，及び，それらの候補の中から実際の配置位置を選択する方法のいずれについても改善の余地があると考えられる．そこで本論文の前半では，ラベル配置率の改善を主な目的として，このアルゴリズムの改良を行い，その有効性を計算機実験により評価する．

グラフ描画中にラベルを配置する問題に深く関連した問題として，地名などを表すラベルを地図中に配置する問題がある [14] ~ [21] . Christensen ら [14] は，焼きなまし法により地図中の地点にラベルを配置する方法を提案した．この方法は高いラベル配置率を実現できるものとされているが，Wagner ら [18], [19] は，Christensen らの方法に近いラベル配置率を達成する高速なアルゴリズムを提案した．また，Kameda-Imai [15] は，地図中の地点及び折れ線（道路や鉄道，川などに対応）にラベルを配置する方法を示した．本研究では，グラフの頂点と辺の両方がラベルをもつ場合について，Wagner ら及び Kameda-Imai の方法をグラフ描画へのラベル配置に適用できるよう変更し，それらの方法と提案方法との比較実験も行う．

分離型アプローチでは，グラフの描画を求める段階でラベルの存在を考慮していないため，グラフが密である場合やラベルのサイズが大きい場合には，描画中にラベルを配置する際に配置できないラベルが多く残ることがある．分離型アプローチ以外のアプローチによりラベル付き描画を求めるアルゴリズムとしては，Klau ら [22] 及び Binucci ら [23] によるものが知られている．しかし，これらの方法は直交描画を求める場合にのみ適用可能であり，また入力

として、グラフの頂点集合と辺集合以外に、いわゆる直交表現 (orthogonal representation) を必要としている。

そこで、本論文の後半では、さらにラベル配置率を改善するため、グラフ G が与えられたとき、各ラベルのサイズを考慮して、ラベルの配置に適した G の描画を求める問題について考察する。この問題では、多くのラベルを配置できる描画を求めることが目的であるが、同時に描画自体がある程度以上望ましいものであることも必要であるものとする。描画の望ましさの評価は、文献 [4], [5] などで行われている複数の基準 (辺の交差数、頂点と辺の近接の個数など) に基づいて行うこととする。

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、これまでに提案されたグラフ描画アルゴリズムとラベル配置アルゴリズムの代表的なものいくつかについて説明する。第 3 章ではグラフの描画中に頂点や辺のラベルを配置するアルゴリズムを、第 4 章ではラベルのサイズを考慮したグラフ描画法を、それぞれ提案する。最後に第 5 章において、本論文の結果をまとめ、今後の課題について述べる。

第2章 従来の方法とその問題点

2.1 緒言

本章では、まず 2.2 において、頂点や辺がラベルをもたないグラフを対象とした既存のグラフ描画アルゴリズムのいくつかについて説明する。

ラベルをもつグラフに対しては、前述のように、これまで分離型アプローチをとることが主に検討されてきた。このアプローチでは、与えられたグラフの描画をまず決定し、その後その描画中にラベルを配置することになるが、その後半で用いることができる 2 つのラベル配置方法について 2.3 で説明する。また、併せて、地図中にラベルを配置するための 2 つの方法も紹介する。最後に 2.4 において、2.3 で説明したラベル配置アルゴリズム及び分離型アプローチの問題点について考える。

2.2 グラフ描画アルゴリズム

本節では、Eades [2]、Kamada-Kawai [3]、及び川西ら [5] のグラフ描画アルゴリズムについて簡単に説明する。これらのアルゴリズムは、頂点や辺がラベルをもたない無向グラフを描画対象として設計されたものである。いずれのアルゴリズムも、頂点に働く力を考え、その力がつり合う位置に頂点を配置する、あるいは、描画全体のエネルギーを定義し、それが極小になるように頂点を配置する、といういわゆる力指向アプローチに基づいている。

2.2.1 Eades の描画アルゴリズム

Eades [2] の描画アルゴリズムでは、まず描画すべきグラフ G の頂点を適当に初期配置する。そして、その後、各頂点の移動をある定数 M 回繰り返し、各辺を直線分で描く。各頂点 i の移動を行う際には、他の頂点から i に働く力を求める。そして、合力の向きに、その大きさの C_1 倍だけ、 i を移動させる (C_1 : 正の定数)。

Eades は、隣接頂点 i, j 間に働く力 $f_{i,j}$ 及び非隣接頂点 i, j 間に働く力 $f'_{i,j}$ を以下のように定めている (力は方向と大きさをもったベクトル量であるから、厳密には、以下の式の右辺に力の方向に応じた単位ベクトルをかけるべきであるが、簡単のため省略している。これ以

降の力を表す式でも同様である.)

$$f_{i,j} = C_2 \log \left(\frac{d_{i,j}}{C_3} \right) \quad (2.1)$$

$$f'_{i,j} = \frac{C_4}{d_{i,j}^2} \quad (2.2)$$

ここで, $d_{i,j}$ は頂点 i, j 間の描画中の距離である. $C_2 \sim C_4$ は正の定数であり, 特に, C_3 は隣接頂点間の理想距離と呼ばれる値である. $f_{i,j}$ は, $d_{i,j} > C_3$ であれば $f_{i,j} > 0$ より引力として, $d_{i,j} < C_3$ であれば $f_{i,j} < 0$ より斥力として働く. これに対し, $f'_{i,j}$ は常に斥力として働く.

2.2.2 Kamada-Kawai の描画アルゴリズム

グラフ $G = (V, E)$ の全ての 2 頂点 i, j について理想距離を定義し, 更にそれを用いて描画全体のエネルギー E_t を定義する. Kamada-Kawai [3] のアルゴリズムは, このエネルギーが極小となるように頂点を配置した後, 各辺を直線分で描くというものである. G の描画に対してエネルギー E_t は次のように定義される.

$$E_t = \sum_{i < j} \frac{1}{2} k_{i,j} (d_{i,j} - \ell_{i,j})^2 \quad (2.3)$$

ここで, $d_{i,j}$ は 2 頂点 i, j 間の描画中の距離を表す. $\ell_{i,j}$ は頂点 i, j 間の理想距離であり, G 上の i, j 間の最短パスの長さ (パス上の辺の数) として定義される. また $k_{i,j}$ はバネ定数と呼ばれるもので, $\ell_{i,j}^2$ に反比例した値をもつ.

各頂点 i の座標を (x_i, y_i) で表すと, エネルギー E_t が極小になる条件は以下のようになる.

$$\frac{\partial E_t}{\partial x_i} = \frac{\partial E_t}{\partial y_i} = 0 \quad (i \in V) \quad (2.4)$$

上式を直接解くのは困難であるため, 適当な位置に頂点を初期配置した後, 1 つの頂点 m のみを動かすというステップを繰り返すことで式 (2.4) を満たすようにする. 各ステップにおいて, 移動させる頂点 m は, 次に定義される i が最大となる頂点とする.

$$i = \sqrt{\left\{ \frac{\partial E_t}{\partial x_i} \right\}^2 + \left\{ \frac{\partial E_t}{\partial y_i} \right\}^2} \quad (2.5)$$

選択された頂点 m の新しい位置は, 2 次元の Newton-Raphson 法を用いて求める. 具体的には, 次の 2 式を満たす δ_x, δ_y を計算し, m を x 方向に δ_x , y 方向に δ_y 動かすという処理を, m があるしきい値 ϵ を下回るまで繰り返し実行する.

$$\frac{\partial^2 E_t}{\partial x_m^2} (x_m, y_m) \delta_x + \frac{\partial^2 E_t}{\partial x_m \partial y_m} (x_m, y_m) \delta_y = -\frac{\partial E_t}{\partial x_m} (x_m, y_m) \quad (2.6)$$

$$\frac{\partial^2 E_t}{\partial y_m \partial x_m} (x_m, y_m) \delta_x + \frac{\partial^2 E_t}{\partial y_m^2} (x_m, y_m) \delta_y = -\frac{\partial E_t}{\partial y_m} (x_m, y_m) \quad (2.7)$$

以上のような処理を繰り返し, 全ての $i \in V$ に対して $i \leq \epsilon$ となればアルゴリズムは停止する.

2.2.3 川西らの描画アルゴリズム

川西ら [5] の描画法は、Eades の描画法と同様、 G の頂点を初期配置した後、頂点に働く力に基づいて各頂点 i の移動をある定数 M 回繰り返すという手法である。隣接頂点 i, j 間に働く力 $f_{i,j}$ 、及び非隣接頂点 i, j 間に働く力 $f'_{i,j}$ は、次のように定めている。

$$f_{i,j} = C_1 \left(\sqrt{\frac{d_{i,j}}{l_{i,j}}} - \frac{l_{i,j}}{d_{i,j}} \right) \quad (2.8)$$

$$f'_{i,j} = \frac{C_2}{l_{i,j}} \left(\sqrt{\frac{d_{i,j}}{l_{i,j}}} - \frac{l_{i,j}}{d_{i,j}} \right) \quad (2.9)$$

ここで C_1, C_2 は正の定数であり、 $d_{i,j}$ は頂点 i, j 間の描画中の距離である。また、 $l_{i,j}$ は i, j 間の理想距離であり、 G 上の i, j 間の最短パスの長さ（パス上の辺の数）として定義される。

川西らの描画法の後半では、頂点と辺の近接を減らすため、辺の midpoint から頂点に働く力も定義している。まず任意の頂点 i と、それに接続していない辺 $e = (a, b)$ に対して、 e の midpoint k と i との間の理想距離 $l'_{i,k}$ を次のように定めている。

$$l'_{i,k} = \frac{l_{i,a} + l_{i,b}}{2} \quad (2.10)$$

そして k と i の間に働く力を次のように定めている。

$$f''_{i,k} = \frac{C_3}{d'_{i,k}} \log \left(\frac{d'_{i,k}}{l'_{i,k}} \right) \quad (2.11)$$

ここで、 C_3 は正の定数であり、 $d'_{i,k}$ は i と k との間の描画中の距離である。この力は、頂点 i のみに、 i から k の方向に働く（ $f''_{i,k} > 0$ ならば引力、 $f''_{i,k} < 0$ ならば斥力となる）。ただし、図 2.1 に示すように、 $\angle iab$ と $\angle iba$ が共に鋭角となる場合にのみ働くものとしている。

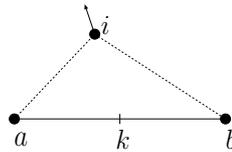


図 2.1: 頂点と辺との間に働く力 ($f''_{i,k} < 0$)

川西らの描画法の概略を以下に示す。なお、描画領域は 1 辺が D の正方形領域に固定されているものとする。

- (1) 全ての 2 頂点 i, j について、理想距離 $l_{i,j}$ を求める。
- (2) 各頂点 i と、それに接続していない各辺 e とについて、 e の midpoint k と i との間の理想距離 $l'_{i,k}$ を求める。

- (3) 頂点を半径 $D/2$ の円の周上に等間隔に初期配置する .
- (4) $I = 1, 2, \dots, M$ について , 次の (4-a) を実行する .
 - (4-a) 各頂点 i に対して ,
 - (4-a-1) i 以外の各頂点 j に対して $f_{i,j}$ あるいは $f'_{i,j}$ を計算し , それらの合力を求めて F_i とする .
 - (4-a-2) $I \geq M/2$ であれば , 各辺の中点から i に働く力 $f''_{i,k}$ を計算し , その合力を F_i に加える .
 - (4-a-3) F_i の向きに , その大きさの C_4 倍だけ i を移動させる .
- (5) 各辺を直線で描く .

2.3 ラベル配置アルゴリズム

本節では , グラフ描画を対象とした Kamada [6] , Kakoulis-Tollis [8] のラベル配置アルゴリズム , 及び Wagner ら [18], [19] , Kameda-Imai [15] の地図ラベル配置アルゴリズムについて簡単に説明する .

2.3.1 Kamada のラベル配置アルゴリズム

Kamada [6] は , グラフ描画中の頂点に対してラベルを配置する非常に単純な方法を提案している . 本研究では , 以下 , この方法を KM 法と呼ぶことにする . 各頂点 $v \in V$ に対し , この方法は , v のラベルを , v に接続する隣り合う辺のうち , それらのなす角度が最大である 2 辺の中央に配置する (ラベルが頂点や辺 , あるいはすでに配置されているラベルと重なるときは , そのラベルは配置されない) .

KM 法は非常に単純な方法であるため高速であるが , ラベルのサイズが十分に小さくない限り , 多くのラベルを配置することはできない .

2.3.2 Kakoulis-Tollis のラベル配置アルゴリズム

以下 , Kakoulis-Tollis [8] のアルゴリズムのことを KT 法と呼ぶ . KT 法は , 各頂点及び各辺に対してラベル配置位置の複数の候補 (ラベル候補と呼ぶ) を作り , それらの中から最終的なラベル位置を選択するアルゴリズムである . この方法は , 大きく分けて以下の 3 つのステップからなる .

ステップ1: ラベル候補の作成

図 2.2 のように、各頂点及び各辺に対して複数のラベル候補を作成する（同図 (a) は、頂点に対して 6 個のラベル候補を作成する場合の例を、同図 (b) は辺に対するラベル候補の例をそれぞれ示している）。そして、作成したラベル候補のうち、他の辺や頂点と重なるものは削除する（他のラベル候補とは重なっていてもよい）。

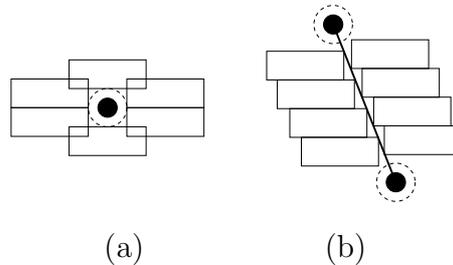


図 2.2: KT 法におけるラベル候補の例

ステップ2: 完全クラスタの生成

ラベル候補に 1 対 1 に対応して頂点を設け、異なる 2 つのラベル候補が重なっているとき対応する頂点間に辺を与えることによって、グラフ G_R を作る。このグラフの各連結成分をクラスタと呼び、クリークをなすクラスタを完全クラスタと呼ぶ。クリークをなしていないクラスタは非完全クラスタと呼ぶことにする。また、簡単のため、任意のラベル候補 λ に対して、対応する頂点の次数のことを λ の次数と呼ぶことにする。

このステップでは、全てのクラスタが完全クラスタとなるように、ラベル候補を削除していく。具体的には、全てのクラスタが完全クラスタになるまで次のような処理を繰り返す。

- (*) G_R の各非完全クラスタ C において、最も次数の高い（最も多くのラベル候補と重なっている）頂点（対応するラベル候補が、非常に少数のラベル候補のみをもつ G の頂点あるいは辺のものである場合には、 C の中で次数が次に高い頂点）を 1 つ削除する（対応するラベル候補も同時に削除する）。

ステップ3: ラベルの位置の決定

まず、2 部グラフ $G_M = (V_C, V_G, E_M)$ を作る。ここで、 V_C はステップ 2 でできた完全クラスタに 1 対 1 に対応する頂点の集合であり、 V_G は G の頂点及び辺に 1 対 1 に対応する頂点の集合である。ある完全クラスタ C が G の頂点あるいは辺に対するラベル候補を含むときかつそのときに限り、それらに対応した頂点間に辺が存在する。このグラフ G_M のことをマッチンググラフと呼ぶ。

ラベル位置の決定は、 G_M に対して最大マッチング問題 [24] を解くことによって行う¹。具

¹ラベル候補の *cost* を考えている場合には、要素数最大のマッチングの中で、辺の重みの総和が最小のものを求める問題に帰着される [8]。

体的には、以下の手順を実行する。

1. G_M のある最大マッチング M を求める。
2. 各辺 $(c, g) \in M$ ($c \in V_C, g \in V_G$) に対して、以下の (a), (b) を実行する (以下では G の頂点のラベルを配置する場合について説明しているが、辺の場合でも同様である。)
 - (a) c に対応したクラスタを C , g に対応した頂点を v とし, v のラベル候補のうち, C に含まれるものを λ とする。
 - (b) v のラベルを, λ の位置に配置する。

以上が、KT法の概略である。なお、文献 [8] では、ステップ 3 の後で次のような後処理をするとよいと書かれているが、具体的な処理内容は述べられていない。

- (i) 新しいラベルを配置できるように、配置済のラベルを局所的に移動する。
- (ii) ラベルが付随する頂点や辺、あるいは他のラベルと重なることを許して、より多くのラベルを配置する。

2.3.3 Wagner らの地図ラベル配置アルゴリズム

Wagner ら [18], [19] のアルゴリズムは、もともと地図中の地点に対し地名に相当するラベルを配置する方法として提案されたものである。KT法と同様、各地点に対し複数のラベル候補を作成した後、それらの中から最終的なラベル位置を選択するアルゴリズムである。

任意のラベル候補 λ について、それをもつ地点を $site(\lambda)$ と表すことにする。また、地点 p がもつラベル候補の集合を $LC(p)$ と表す。異なる地点に対する 2 つのラベル候補同士が互いに重なりあっているとき、それらはコンフリクトしているという。各ラベル候補 λ について、それにコンフリクトしているラベル候補全てからなる集合を X_λ と表し、 $|X_\lambda|$ なる数のことを λ の次数という。ラベル候補のある集合 S に対し、それに属する任意の 2 つのラベル候補がコンフリクトしている場合、 S はクリークをなしているという。

Wagner らのアルゴリズムは、フェーズ 1, フェーズ 2 の 2 段階から構成されている。

(1) フェーズ 1

フェーズ 1 では、まず、与えられた n 個の地点のそれぞれに対し、定数 c 個のラベル候補を設ける。 $c = 4$ の場合のラベル候補の例を図 2.3 に示す (図中の黒丸は地点を表している)。ただし、他の地点と重なっているラベル候補は消去する。次に、ラベル候補同士のコンフリクトの状況を調べておく。そして、各地点に対して、以下に述べる 4 つのルール L1 ~ L3 を可能な限り繰り返し適用し、各地点のラベル候補を減らしていく。

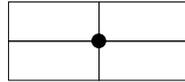


図 2.3: 1つの地点に対するラベル候補の例 ($c = 4$)

・ルール L1

ある地点 p に対して、他のどの地点のラベル候補ともコンフリクトしていない、すなわち、次数 0 のラベル候補 λ_{p_i} が存在する場合、 p のラベル位置を λ_{p_i} に決定し、 $LC(p) - \{\lambda_{p_i}\}$ のラベル候補を全て消去する。図 2.4(a) に例を示す ($c = 4$ の場合の例である。図中、ルールの適用によりラベル位置に決定するラベル候補を太線で、消去されるラベル候補を破線で示している)。

・ルール L2

ある 2 つの地点 p, q と、それぞれのあるラベル候補 $\lambda_{p_i}, \lambda_{q_j}$ に対し、 λ_{p_i} がコンフリクトしているのは λ_{q_j} 以外の q のラベル候補のみであり、かつ λ_{q_j} がコンフリクトしているのは λ_{p_i} 以外の p のラベル候補のみである場合、 p のラベル位置を λ_{p_i} に、 q のラベル位置を λ_{q_j} に決定し、 $LC(p) - \{\lambda_{p_i}\}, LC(q) - \{\lambda_{q_j}\}$ に属するラベル候補を消去する。

・ルール L3

ある地点 p に対し、 $LC(p) = \{\lambda_{p_i}\}$ であり、 $X_{\lambda_{p_i}}$ がクリークをなしている場合、 p のラベル位置を λ_{p_i} に決定し、 $X_{\lambda_{p_i}}$ のラベル候補を全て消去する。図 2.4(b) に例を示す。

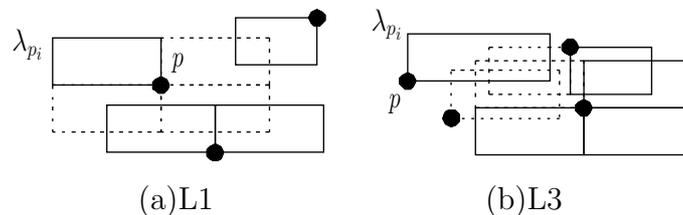


図 2.4: ルール L1 と L3 の例

(2) フェーズ 2

フェーズ 1 により次の条件が成立することになれば、Wagner らのアルゴリズムは停止する。そうならない場合には、このフェーズ 2 を実行する。

(停止条件) 全ての地点のラベル候補が高々 1 つにまで減り、かつそれらのどの 2 つも重なり合っていない。

フェーズ 2 では、まず全ての地点の中で、最も多くラベル候補が残っている地点 p を見つける。そして、 $LC(p)$ の中で、最も次数が大きいラベル候補を 1 つ消去する。この結果、ルール L1 ~ L3 が新たに適用できることになれば、それらを可能な限り繰り返して実行する。

以上の処理を，上記の停止条件が成立するまで繰り返す．このようにして残ったラベル候補は，ラベル配置問題の解を形成している．

Wagner らのアルゴリズムは以上のようなアルゴリズムである．平面上の点にラベル配置をする場合，この方法は KT 法より高い配置率を達成できるとの報告がある [19]．

グラフ描画中の頂点及び辺に対して KT 法と同様の方法でラベル候補を作成することにより，Wagner らのアルゴリズムはグラフ描画へのラベル配置にも用いることができる．以下では，そのように拡張した方法を WW 法と呼ぶことにする．

2.3.4 Kameda-Imai の地図ラベル配置アルゴリズム

Kameda-Imai [15] のアルゴリズムは，地図中の地点及び折れ線（道路や鉄道，川などに対応）に対して地名などに相当するラベルを配置するアルゴリズムである．このアルゴリズムを KI 法と呼ぶことにする．具体的には以下のような手順を実行する（以下の記述におけるラベル空間とは，地点や折れ線と重なることなくラベルを配置できる範囲のことであり，詳細は第 3 章で述べる）．

- a) 各地点に対してラベル空間を求める．
- b) 各折れ線に対してラベル空間を求める．
- c) 各地点及び各折れ線に対し，実際に作成されたラベル空間の大きさと，周囲に障害物が全くない場合に作成されるラベル空間の大きさとの比（ラベル移動比率）を求める．
- d) 全ての地点及び折れ線にラベルが配置されるまで，あるいは，ラベルが未配置の地点及び折れ線のラベル移動比率が全て 0 になるまで以下の処理を繰り返す．
 - d-1) ラベルが未配置で，かつラベル移動比率が 0 でない地点及び折れ線のうち，ラベル移動比率が最小のものを選ぶ．
 - d-2) 選んだ地点あるいは折れ線の最も広いラベル空間の中央にラベルを配置する．ここで配置したラベルを l とする．
 - d-3) l と重なる各ラベル空間を， l と重ならないように更新する（ラベル移動比率も更新する）．

KI 法では，折れ線のラベルを配置する際，ラベルの各文字を折れ線にそって 1 文字ずつ配置する．

2.4 従来の方法の問題点

本節では、2.3 で説明したラベル配置アルゴリズムと分離型アプローチの問題点について考える。

2.4.1 ラベル配置アルゴリズムに関する問題点

2.3 で述べた4つのラベル配置法のうち、KM法については、各頂点のラベル配置位置を独立に決定しようとしているため、配置できないラベルが多く残ることがある。またKI法は、そのままではグラフ描画へのラベル配置に用いることができない（グラフ描画に適用するための変更については3.4.3.1で後述する）。

KT法とWW法に関しては、以下のような問題点があげられる。

問題点1：これらの方法では、描画における各頂点及び各辺の周囲の状況を特に考慮せずにラベル候補を生成し、それらの中から最終的なラベルの位置を選択している。しかし、このような手順では、ラベルを配置できる位置が存在しているにもかかわらず、それを見逃すことがある。

問題点2：特にラベルのサイズが比較的小さい場合、不要なラベル候補が多く生成されてしまう。

次数が0のラベル候補、すなわち他のラベル候補と重なっていないラベル候補のことを、0次候補と呼ぶことにする。図2.5は、頂点数20、辺の数50のグラフのある描画に対して、KT法により生成される辺のラベル候補を全て示したものである。この図を見ると、0次候補が数多く作られていることがわかる。

0次候補は、他の辺のラベル候補を犠牲にすることなくラベルの配置ができるので歓迎すべきものではあるが、1本の辺にこのように多くの0次候補があってもあまり意味がなく、ステップ2以降の処理の実行時間を不必要に増やしてしまうことになる。

2.4.2 分離型アプローチの問題点

分離型アプローチでは、グラフの描画を求める段階でラベルの存在を考慮していないため、得られた描画において、ラベルを配置したい頂点や辺の周囲に十分な領域が確保されるとは限らない。このため、特にグラフが密である場合やラベルのサイズが大きい場合には、ラベル候補すら作成されない頂点や辺が多くでき、満足できるラベル配置率が得られないことがある。

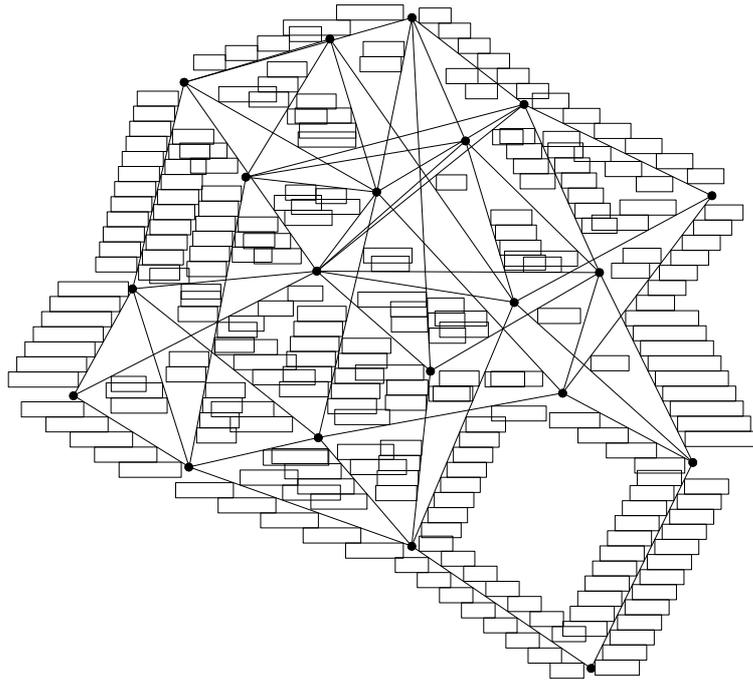


図 2.5: KT 法により生成される辺のラベル候補 (ラベル候補数 250)

2.5 結言

本章では、頂点や辺がラベルをもたないグラフを対象とした3つのグラフ描画アルゴリズムについて説明した後、Kamada [6]、Kakoulis-Tollis [8]のラベル配置法、及びWagnerら [18], [19]、Kameda-Imai [15]の地図ラベル配置法について述べた。さらに、これらのラベル配置法と分離型アプローチの問題点を指摘した。

第3章では、2.4.1であげた問題点に対する改善を行うことにより、新しいラベル配置法を提案する。また第4章では、頂点と辺の位置の決定段階でラベルの存在を考慮するグラフ描画法を提案する。

第3章 グラフ描画へのラベル配置アルゴリズム

3.1 緒言

本章では、与えられたグラフ描画中にラベルを配置する問題を扱う。2.4.1 で述べたように、KT 法及び WW 法は、以下の 2 つの問題点をもっている。

問題点 1：描画における各頂点及び各辺の周囲の状況を特に考慮せずに機械的にラベル候補を生成するため、ラベルを配置できる位置が存在しているにもかかわらず、それを見逃すことがある。

問題点 2：特にラベルのサイズが比較的小さい場合、不要なラベル候補が多く生成されてしまうことがある。

本章では、これらについての改善を KT 法に対して行うことにより、より高いラベル配置率を実現するラベル配置法を提案する。

以下では、辺のみがラベルをもつ場合、及び頂点のみがラベルをもつ場合のアルゴリズムを、それぞれ、3.2 及び 3.3 において示す。そして 3.4 において頂点と辺両方がラベルをもつ場合のアルゴリズムを提案し、2.3 で述べた各手法との比較を行う。

3.2 辺のみがラベルをもつ場合のラベル配置アルゴリズム

本節では、辺のみがラベルをもつ場合のラベル配置アルゴリズムについて説明する。以下この方法を提案手法 1 と呼ぶことにする。以下では、3.2.1 においてアルゴリズムの説明を、3.2.2 において計算機実験の結果をそれぞれ示す。

3.2.1 アルゴリズムの説明

提案手法 1 は、KT 法を基に設計したものであり、次のような特徴をもつ。

- (i) 各辺の周囲の状況を考慮してラベル候補を作成する。これにより、ラベル候補の集合を KT 法より大幅に小さなものとしている。

- (ii) 各ラベル候補の位置を最初に決めたものに固定するのではなく、他のラベル候補の位置を考慮して移動することを許している。
- (iii) グラフ G_R から完全クラスタを生成するための操作として、頂点の削除のみではなく、辺の削除も行っている。このことは、ラベル候補の移動を許しているために可能となっている。

以下、3.2.1.1で提案手法1におけるラベル候補の作成法について述べる。そして、3.2.1.2において提案手法1で行っている0次候補の選択と呼ぶ処理について説明する。次に、3.2.1.3及び3.2.1.4では、ラベル候補の2種類の移動操作について説明する。3.2.1.5では、本手法で行っている簡単な後処理について述べる。最後に、3.2.1.6において、提案手法1全体の簡単な記述を示す。

3.2.1.1 ラベル候補の作成

提案手法1におけるラベル候補の作成法について、例を用いて説明する。まず、各辺について、一方の端点から他方の端点まで、辺の両側でラベルを移動させたときにラベルが通過する範囲を求める。図3.1(a)参照。次に、その範囲を、図3.1(b)に示すように、辺や頂点と重ならない部分に分割する。以下、分割された部分のそれぞれのことを(辺ラベルに対する)ラベル空間と呼ぶことにする。ラベルを配置できるだけの大きさをもたないラベル空間は削除する。その後、図3.1(c)に示すように、残った各ラベル空間の中央に1つつラベル候補を作成する。

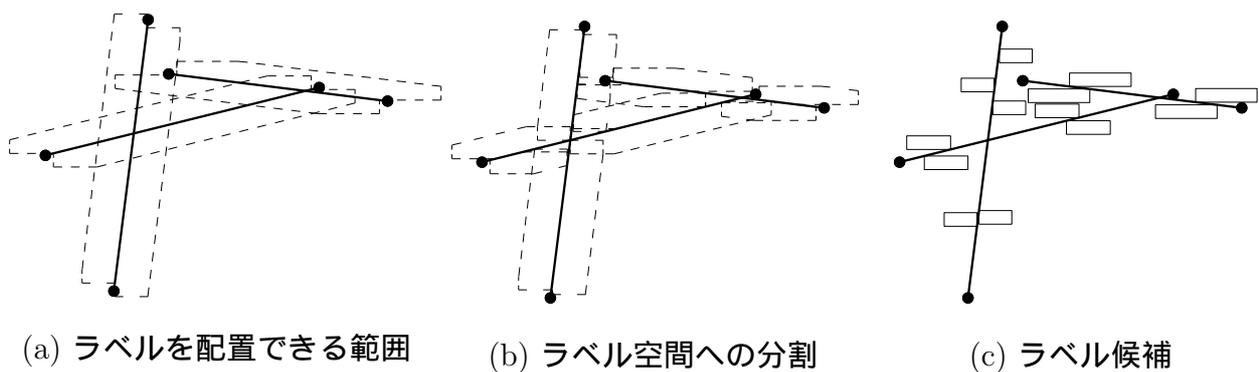


図 3.1: 提案手法1におけるラベル候補の作成法

3.2.1.2 0次候補の選択

提案手法1では0次候補の選択と呼ぶ処理を行っている。辺が0次候補をもつなら、その候補の位置に、他の辺に影響を与えることなくラベルを配置することができる。そこで、3.2.1.1

で述べたような方法でラベル候補を作成した後，次のような処理を繰り返し実行することにする．

- (**) 辺 e が 0 次候補を 1 つ以上もつなら，それらの中で最も広いラベル空間内にあるもの 1 つを残し，他の e のラベル候補を全て削除する．（残されたラベル候補には，後にラベルが配置されることになる．）

ラベル空間が広いラベル候補の場合，その周囲に存在する頂点や辺が少ない場合が多い．したがって，そのようなラベル候補のみを残すことにより，単にラベル候補数が減るだけでなく，最終的に得られるラベル付き描画において，ラベルと付随する辺の対応が明確になると期待できる．

この操作を 1 回実行して，いくつかのラベル候補を削除したとき，新たに 0 次候補が生じることがある．よって，0 次候補の選択は，0 次候補をもち，かつ 2 つ以上のラベル候補をもつ辺が残っている限り実行するものとする．

図 2.5 の例に対して，以上の処理により生成されるラベル候補を図 3.2 に示す．

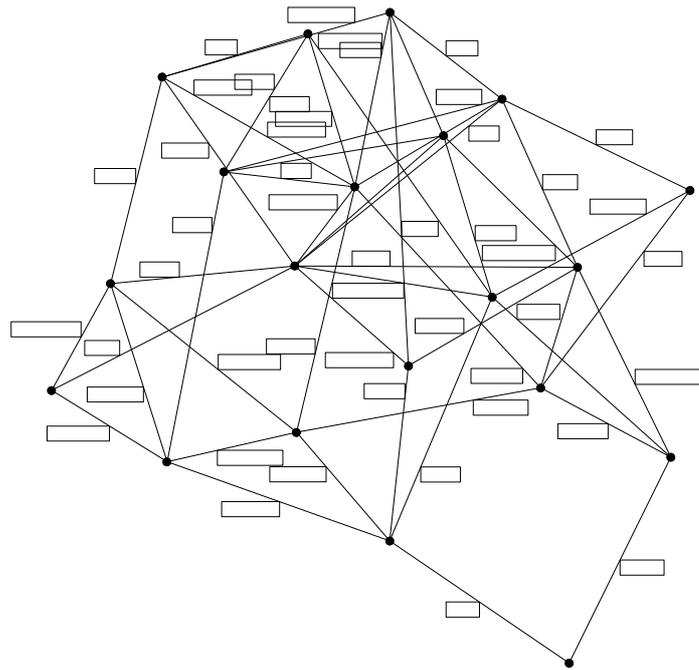


図 3.2: 提案手法 1 により生成されるラベル候補（ラベル候補数 47）

3.2.1.3 ラベル候補のシフト

KT 法のステップ 2 では，グラフ G_R から完全クラスタを作るための操作として，ラベル候補（頂点に対応）の削除しか行っていないが，クラスタ内の辺を削除することでも，完全

クラスタに分解し得る．図 3.3 にそのような例を示す．クラスタ内の辺はラベル候補同士が重なっているときに存在するから，ラベル候補をシフトさせてその重なりをなくすことが辺の削除に対応する．

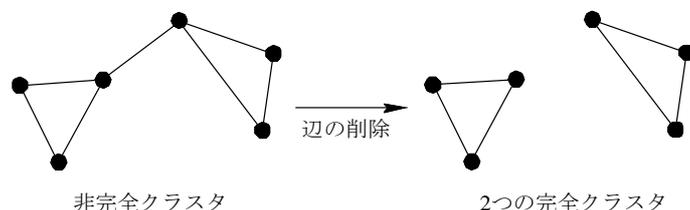


図 3.3: 辺の削除により 2 つの完全クラスタが作られる例

要素数が 2 以上のクラスタ C に対して，ラベル候補のシフトを行う手続き $\text{shift}(C)$ を次に示す．この手続きは，要素数が 2 以上の全てのクラスタに対して，ラベル候補の移動が行われなくなるまで繰り返し実行する．

手続き $\text{shift}(C)$

1. C の頂点に対応するラベル候補のうち，最も上にあるもの（長方形の上側の辺が最も上にあるもの）を λ_1 とする． λ_1 が付随している辺を e とする．
2. λ_1 を辺 e にそって上下いずれかにシフトする（ e が水平な辺の場合は左右いずれかにシフトする）．この移動は， λ_1 のラベル空間の内部のみで，今まで重なっていなかった他のラベル候補と重ならないようにして行う．移動距離は，次数の減少を最大とするような距離の中で最小のものとする．もし，次数を全く減らすことができない場合には， λ_1 の移動は行わない．
3. C の頂点に対応するラベル候補のうち，最も下にあるものを λ_2 とする． λ_2 の次数が 0 でないならば，(2) と同様にして， λ_2 を上下いずれかにシフトする（付随する辺が水平な場合は左右いずれかにシフトする）．

図 3.4(a)~(c) に 3 つの例を示す．まず，同図 (a) 左のように，辺が密集して，かつ各辺のラベル候補が少ない部分では，KT 法を適用してもラベルを配置できない辺が残ることが多い．手続き shift は，同図 (a) 右のようにラベル候補をシフトすることによって，同時に配置できるラベル候補の個数を増やしている．この例のように，手続き shift は完全クラスタに対しても実行する．

次に，図 3.4(b) 左では，3 つのラベル候補のうちの高々 1 つにしかラベルが配置できないが，手続き shift の実行後（同図 (b) 右）では 2 つに配置できるようになっている．この例において，一番下のラベル候補は移動していないが，これはシフトしても次数を減らすことができないためである．

図 3.4(a), (b) の例のように，クラスタ中で一番上にあるラベル候補は上向きにシフトされることが多いが，そうでない場合もある．図 3.4(c) 左における 2 つのラベル候補のうち，より上にあるものについて考える．このラベル候補は，上下いずれにシフトしても次数を 0 にすることができるが，移動距離の短い方が選択されて，下向きにシフトされることになる（同図 (c) 右）．

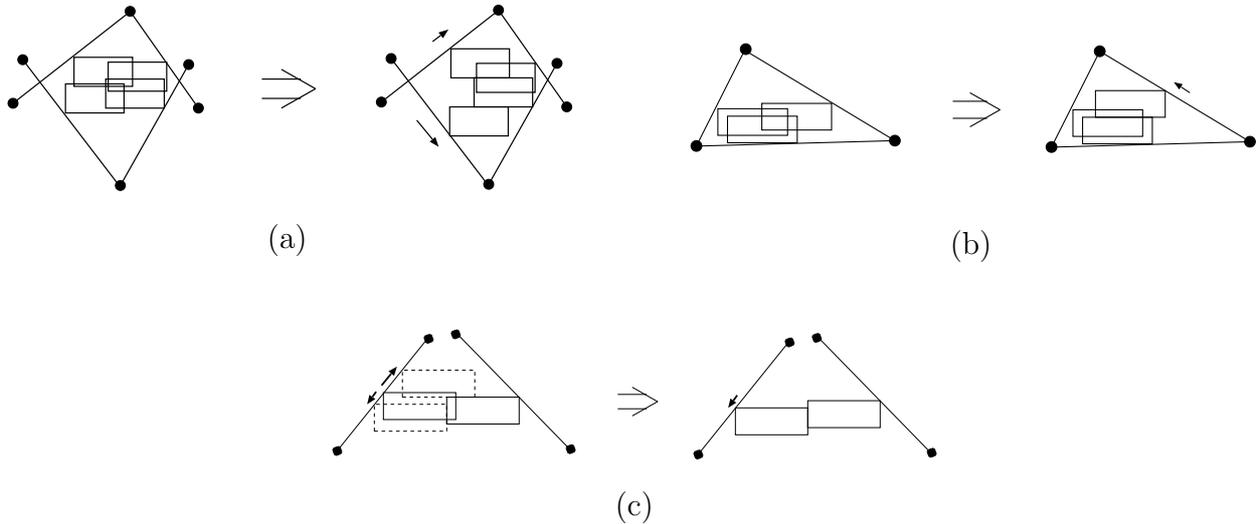


図 3.4: ラベル候補のシフトの例

3.2.1.4 要素数が 2 のクラスタに対するラベル候補のシフト

提案手法 1 では，各ラベル候補をそのラベル空間の中央に設けているため，要素数が 2 のクラスタにおいて，図 3.5 左のような状況が生じることがある．この図中の 2 つのラベル候補は，両方同時にシフトすれば重なりをなくすることができるが，片方ずつのシフトではそれができない．このような場合には，2 つのラベル候補を最も離れる位置に一旦シフトする．そして，図 3.5 右のように，それらの間隔の $1/3$ ずつ各々を近づけた位置に，ラベル候補を再移動する（このようにすれば，ラベル空間の端からの距離とラベル候補間隔が等しくなる）．

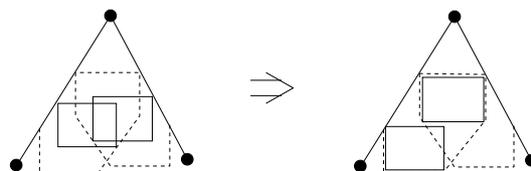


図 3.5: 一方のラベル候補のシフトでは重なりを避けられない例

3.2.1.5 後処理

マッチンググラフ G_M の最大マッチングを求め、ラベル配置を行った後、提案手法 1 では、非常に接近して配置されているラベルに対し、両者を離すような移動を行う。具体的には、まずそのような 2 つのラベル λ_1, λ_2 のそれぞれを、ラベルの接近を回避できる方向へ、それぞれが付随している辺にそって、ラベルの高さだけ移動させる（例えば、上向きに動かす場合、ラベルを表す長方形の移動前の上側の辺と、移動後の下側の辺の y 座標が同じになるようにする）。ただし、傾きが非常に緩やかで水平に近いような辺の場合は、ラベルを移動させる距離が非常に大きくなることがあるので、 x 座標、 y 座標ともに変化量がラベルの高さをこえないようにする。次に、移動後の λ_1, λ_2 の状態が次のいずれであるかを調べる。

状態 1：ラベル空間からはみ出しているか、他のラベルと重なっている。

状態 2：状態 1 ではないが、ラベルを上下左右にラベルの高さの半分ずつ拡大したとき、ラベル空間の端や他のラベルと重なる。

状態 3：状態 1, 2 のいずれでもない。

そして、 λ_1, λ_2 の状態に応じて、以下の処理 (i)~(iv) のうちのいずれか 1 つを実行する。

- (i) λ_1, λ_2 とともに状態 1 ではない場合：(移動前の両者間の距離 + 各々を移動させた距離)/2 だけそれぞれのラベルを移動させて、両者を近づける。
- (ii) λ_1, λ_2 の一方が状態 1、他方が状態 3 である場合：状態 1 のラベルを移動前の位置に戻す。
- (iii) λ_1, λ_2 の一方が状態 1、他方が状態 2 である場合：一般性を失うことなく、状態 1 のラベルが λ_1 であるものとする。この場合、 λ_1 は移動前の位置に戻す。 λ_2 については（ラベル空間の端及び他のラベルのうち）最も接近しているものとの距離と、 λ_1 との距離が等しくなる位置に移動する。
- (iv) λ_1, λ_2 とともに状態 1 である場合：それぞれを、ラベルとの重なりやラベル空間からはみ出しを回避できるところまで戻す。そして、その時の両者の間の距離を求め、その $1/3$ だけそれぞれを近づける。

以上に述べた処理を、非常に接近して配置されているラベルの対のそれぞれに対して実行する。なお、この後処理は、ラベルの混雑を緩和するためのものであって、ラベル配置率を向上させるためのものではない。

3.2.1.6 提案手法1の記述

提案手法1の手順をまとめたものを以下に示す。

提案手法1

- (1) G の各辺に対し, 3.2.1.1で述べた方法でラベル候補を作成する。
- (2) 0次候補と, それ以外に少なくとも1つのラベル候補をもつ辺が残っている限り, 3.2.1.2で述べた0次候補の選択処理(**)を実行する。
- (3) クラスタを作成する。
- (4) 要素数が2以上の各クラスタ C に対し, ラベル候補の移動が行われなくなるまで, 手続き $\text{shift}(C)$ を実行する。
- (5) 要素数が2のクラスタに対するラベル候補のシフト処理(3.2.1.4)を実行する。(4)と(5)で, 1つでもラベル候補を移動させることができれば(2)に戻る。
- (6) 2.3.2で述べた処理(*)を, 各非完全クラスタ C に対し1度だけ実行する。1つでもラベル候補を消去できれば(2)に戻る。
- (7) マッチンググラフ G_M を作り, その最大マッチングを求めて, ラベル配置を行う。
- (8) 3.2.1.5で述べた後処理を実行する。

3.2.2 計算機実験

KT法と提案手法1を比較するため, 計算機実験を行った。本節ではその結果について述べる。なお, ここでのKT法は, 辺のみをラベル配置の対象としている。

3.2.2.1 実験方法

実験データとして, 頂点数が20, 辺の数が30, 40若しくは50のグラフを, それぞれランダムに50個ずつ作成した。そして, それぞれのグラフに対して, Eades [2], Kamada-Kawai [3], 川西ら [5]の3通りのアルゴリズムにより描画を求めた。Kamada-Kawaiのアルゴリズムに関しては, 頂点数30で辺の数40, 50, 60のグラフも用いた。全ての場合について, 描画は1辺の長さが10の正方形領域内で行った。

辺のラベルの高さとしては, 0.10, 0.15, 0.20, 0.25, 0.30の5通りの値を試した。ただし, 1つの描画内ではラベルの高さは全て同じとした。一方, ラベルの幅は高さの2~5倍の範囲で辺ごとに定めた。

以上の実験データに対し, KT法と提案手法1とを実行し,

- ステップ 1 終了時におけるラベル候補の個数
- ラベル配置率
- 実行時間

を求めた．使用計算機の CPU は Celeron 500MHz，使用計算機言語は C 言語である．

3.2.2.2 実験結果

以下では，頂点数が 20 のグラフに対して Kamada-Kawai の描画法を用いた場合の実験結果のみを示すが，他の描画法を用いた場合や，頂点数が 30 の時にも同様の結果が得られた．

最初に，ステップ 1 終了時におけるラベル候補数の平均値を表 3.1 に示す．ここで「比率」とは，KT 法によるラベル候補数を，提案手法 1 によるラベル候補数で割った値である．この表より，提案手法 1 によるラベル候補数が，KT 法に比べて非常に少ないこと，特にラベルの高さが小さいときにその差が顕著であることが分かる．

表 3.1: ラベル候補数の比較

辺の数	ラベルの高さ	従来手法	提案手法 1	比率
30	0.10	627.10	30.00	20.90
	0.15	372.72	29.96	12.44
	0.20	248.20	29.98	8.28
	0.25	177.06	29.54	5.99
	0.30	133.52	28.64	4.66
40	0.10	821.24	39.78	20.64
	0.15	443.20	39.40	11.25
	0.20	274.62	38.86	7.07
	0.25	184.10	37.02	4.97
	0.30	131.10	34.76	3.77
50	0.10	1021.90	49.58	20.61
	0.15	509.08	48.70	10.45
	0.20	294.96	46.92	6.29
	0.25	186.34	44.64	4.17
	0.30	127.16	39.94	3.18

次に，KT 法と提案手法 1 による配置率の平均値を表 3.2 に示す．辺の数が多い場合ほど，またラベルの高さが大きい場合ほど，KT 法に比べ，大きな改善が見られる．KT 法は，各辺のラベル候補を最初に決めたものに限定しているため，望ましいラベル位置を見逃す可能性

表 3.2: 配置率の比較

辺の数	ラベルの高さ	従来手法	提案手法 1	上限値
30	0.10	99.9 %	100.0 %	100.0 %
	0.15	99.1 %	99.9 %	99.9 %
	0.20	96.9 %	99.1 %	99.2 %
	0.25	94.2 %	96.7 %	97.0 %
	0.30	89.2 %	94.0 %	94.4 %
40	0.10	99.3 %	99.5 %	99.5 %
	0.15	96.8 %	98.3 %	98.3 %
	0.20	91.3 %	95.3 %	95.5 %
	0.25	81.6 %	88.5 %	89.6 %
	0.30	71.8 %	81.7 %	82.9 %
50	0.10	99.0 %	99.2 %	99.2 %
	0.15	95.0 %	96.8 %	97.0 %
	0.20	84.6 %	91.2 %	91.9 %
	0.25	71.0 %	81.5 %	83.2 %
	0.30	59.2 %	71.2 %	73.4 %

表 3.3: 実行時間の比較

辺の数	ラベルの高さ	従来手法	提案手法 1
30	0.10	358.2 msec	23.2 msec
	0.15	140.0 msec	21.6 msec
	0.20	70.2 msec	22.0 msec
	0.25	43.0 msec	23.0 msec
	0.30	29.6 msec	21.6 msec
40	0.10	913.6 msec	42.6 msec
	0.15	249.2 msec	37.4 msec
	0.20	104.6 msec	37.0 msec
	0.25	54.4 msec	36.8 msec
	0.30	37.2 msec	36.8 msec
50	0.10	1576.0 msec	66.8 msec
	0.15	321.2 msec	61.6 msec
	0.20	123.8 msec	55.4 msec
	0.25	67.8 msec	55.8 msec
	0.30	42.4 msec	58.0 msec

がある．これに対し，提案手法 1 では，ラベル候補のシフトを許しているため，他のラベル候補の位置を考慮した位置にラベル候補を定めることができる．この理由により，表 3.2 のような大きな差が現れたのだと考えられる．

提案手法 1 では，最初に頂点や辺の存在を考慮して，各辺についてラベル空間を設けている．その手順より明らかに，任意の辺に対し，このステップでラベル候補が作られることが，最終的にラベルが配置できることの必要条件となる．したがって，次の値をラベル配置率の（緩い）上限値として用いることができる．

$$\frac{\text{ラベル候補が作成可能な辺の数}}{\text{辺の総数}}$$

表 3.2 に示した「上限値」とは，このようにして計算した値である．提案手法 1 の配置率とこの上限値を比較すると，特にラベルの高さが小さいときには，非常に近い値になっていることが分かる．

表 3.3 には，KT 法と提案手法 1 の実行時間の平均値を示す．グラフ G_M の最大マッチングを求める際には，Hopcroft-Karp [26] のアルゴリズムを用いた．KT 法の実行時間はラベルのサイズによって大きく変化するが，提案手法 1 では，おおむねグラフの辺の本数で決まっていることが分かる．また，表 3.1 と見比べると，ラベルの高さが小さく，KT 法で多くのラベル候補が作られるときに，提案手法 1 の実行時間が KT 法より大きく減っていることが分かる．ラベルの高さが比較的大きいときには，KT 法との差が小さくなっているが，そのような場合ほど，ラベル配置率に関して提案手法 1 の方が優れている．

KT 法と提案手法 1 によりラベルの配置を行った例を，図 3.6 に示す．この例の場合，配置できたラベルの個数は 2 つの手法で等しいのであるが，提案手法 1 の結果である (b) の方が，ラベルが適度に分散して配置され，辺とラベルとの対応関係も分かりやすくなっている．これは，提案手法 1 において，各ラベル空間の中央にラベル候補を作成していることや，0 次候補の選択の処理において最もラベル空間の広いものを選んでいること，及び 3.2.1.5 で述べた後処理を行っていることによると考えられる．

3.3 頂点のみがラベルをもつ場合のラベル配置アルゴリズム

本節では，頂点のみがラベルをもつ場合のアルゴリズムについて説明する．以下この方法を提案手法 2 と呼ぶことにする．以下では，3.3.1 においてアルゴリズムの説明を，3.3.2 において計算機実験の結果をそれぞれ示す．

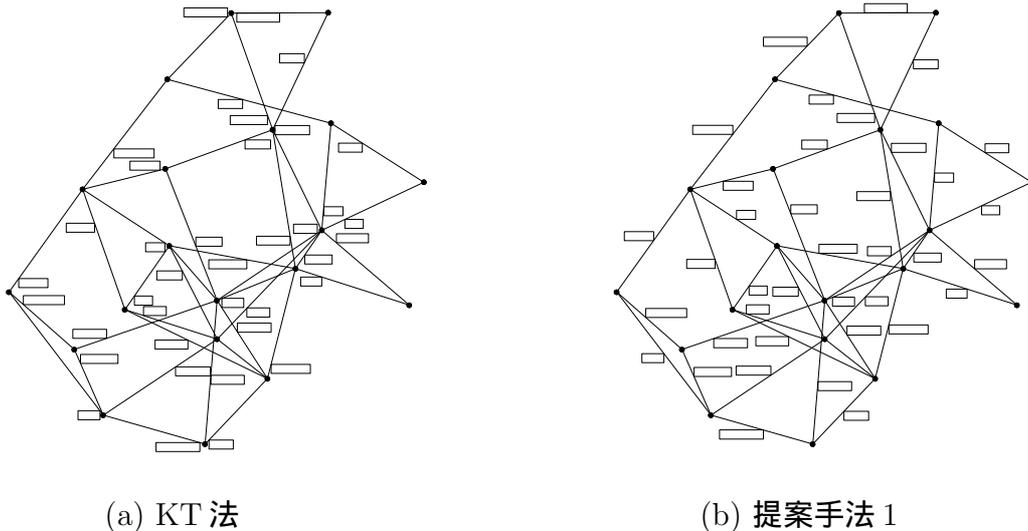


図 3.6: ラベル付き描画の例 (辺の本数: 40, ラベルの高さ: 0.20)

3.3.1 アルゴリズムの説明

提案手法 2 は, 3.2 で述べた辺のラベル配置を行うアルゴリズムと同じく, KT 法を基にしたアルゴリズムであり, 次のような特徴をもつ.

- (i) 周囲の頂点や辺の位置を考慮してラベル候補を作成する.
- (ii) 各ラベル候補の位置を最初に決めたものに固定するのではなく, 他のラベル候補の位置を考慮して移動することを許す.

提案手法 2 における処理の大まかな流れは提案手法 1 と同様のものであるが, ラベル配置を行う対象が辺ではなく頂点であるため, 各部分の処理内容は大きく異なる. 以下, 3.3.1.1 において提案手法 2 におけるラベル候補の作成方法について述べる. 次に, 3.3.1.2 において, 0 次候補の選択処理について説明する. さらに, 3.3.1.3 においてラベル候補の移動方法について述べる. 最後に, 3.3.1.4 においてアルゴリズム全体の記述を示し, 後処理について説明する.

3.3.1.1 ラベル候補の作成

各頂点 v に対し, 次のような方法でラベル候補を作成する.

- (1) 図 3.7(a) に示すように, v のラベルを, v を中心とする半径 ϵ の円にそって一周させる. このとき, ラベルが通過する範囲を R_v とする.
- (2) R_v を他の頂点や辺と重ならないような範囲に分割する. 図 3.7(b) 参照. このようにして得られた各範囲のことを (頂点ラベルに対する) ラベル空間と呼ぶことにする.

(3) 各ラベル空間の中央に1つずつラベル候補を作成する．図 3.7(b) 参照．

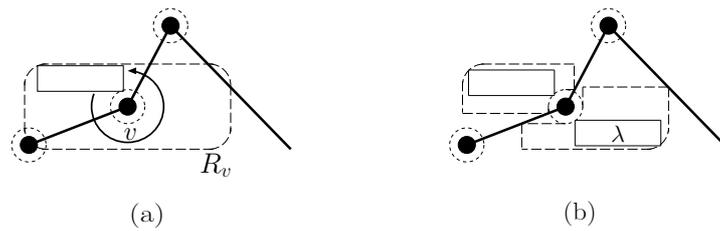


図 3.7: ラベル空間とラベル候補の作成

v に対して1つ以上のラベル候補が作られているものとする．各ラベル候補 λ に対し，対応するラベル空間を $LS(\lambda)$ と呼ぶことにする．また， $LS(\lambda)$ 中で v のラベルを時計回り (反時計回り) に移動させたときに通過する範囲を $CLS(\lambda)$ ($CCLS(\lambda)$) と定義する．図 3.8 に，図 3.7(b) の λ に対する $CLS(\lambda)$ と $CCLS(\lambda)$ を示す．



図 3.8: $CLS(\lambda)$ と $CCLS(\lambda)$

3.3.1.2 0次候補の選択

提案手法1と同様，提案手法2でも0次候補の選択処理を行う．3.3.1.1で述べたような方法でラベル候補を作成した後，3.2.1.2と同様，次のような処理を繰り返し実行する．

(**) 頂点 v が0次候補を1つ以上もつなら，それらの中で最も広いラベル空間内にあるもの1つを残し，他の v のラベル候補を全て削除する．

この処理は，0次候補をもち，かつ2つ以上のラベル候補をもつ頂点が残っている限り実行する．

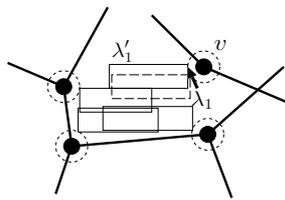
3.3.1.3 ラベル候補のシフト

提案手法2では，ラベル候補の重なりを減らすため，2つ以上の要素をもつクラスタ C に対し，下に示す手続き $\text{shift}(C)$ を行う．

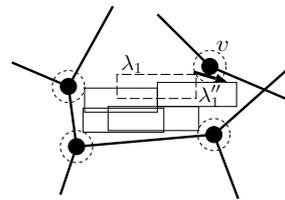
手続き $\text{shift}(C)$

1. A_C を， C 中の頂点に対応する全てのラベル候補の集合とする．

2. λ_1 を, A_C 中で最も上にある (ラベル候補の中心が最も上にある) 候補とする. 同様に, λ_2 を A_C 中で最も下にある候補とする.
3. λ_1 に対し以下を実行する.
 - (a) λ_1 をラベル候補としてもつ G 中の頂点を v とする.
 - (b) 以下の条件を満たす位置の中で最も λ_1 に近い位置 λ'_1 を見つける.
 - (i) $CLS(\lambda_1)$ の内部にある.
 - (ii) λ_1 と重なっている候補以外のどの候補とも重ならない.
 - (iii) (i) と (ii) を満たす位置の中で, 次数が最も小さくなる.
 図 3.9(a) に例を示す. λ'_1 は λ_1 と一致してもよい.
 - (c) (b) と同様にして, $CCLS(\lambda_1)$ 中の位置 λ''_1 を見つける. 図 3.9(b) 参照.
 - (d) n' (n'') を λ'_1 (λ''_1) の次数とする. $n' < n''$ である, あるいは $n' = n''$ かつ λ_1 が λ''_1 より λ'_1 に近いなら, v のラベル候補を λ_1 から λ'_1 へ移動させる. そうでなければ, λ''_1 へ移動させる.
4. λ_2 の次数が 0 でないなら, 3. と同様にして λ_2 の移動を行う.



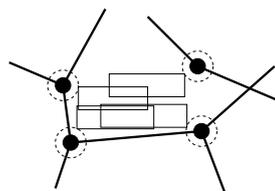
(a) 時計回りの方向への移動



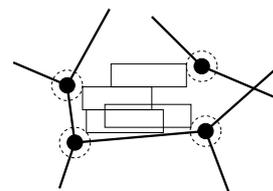
(b) 反時計回りの方向への移動

図 3.9: ラベル候補の移動

図 3.10 にこの手続きの実行例を示す. この例では, シフト前には同時にラベルを配置できるラベル候補は最大で 2 つであるのに対し, シフト後は同時に 3 つの候補にラベルを配置できる. この例のように, 一般に, 手続き $\text{shift}(C)$ によって同時にラベルを配置できるラベル候補の組み合わせを増加させることができる.



(a) シフト前



(b) シフト後

図 3.10: 手続き $\text{shift}(C)$ の実行例

3.3.1.4 アルゴリズムの記述と後処理

提案手法 2 全体の記述を以下に示す .

提案手法 2

- (1) G の各頂点に対し , 3.3.1.1 で述べた方法でラベル候補を作成する .
- (2) 0 次候補と , それ以外に少なくとも 1 つのラベル候補をもつ頂点が残っている限り , 3.3.1.2 で述べた 0 次候補の選択処理 (**) を実行する .
- (3) クラスタを作成する .
- (4) 要素数が 2 以上の各クラスタ C に対し , ラベル候補の移動が行われなくなるまで , 手続き $\text{shift}(C)$ を実行する . 1 つでもラベル候補を移動させることができれば (2) に戻る .
- (5) 2.3.2 で述べた処理 (*) を , 各非完全クラスタ C に対し 1 度だけ実行する . 1 つでもラベル候補を消去できれば (2) に戻る .
- (6) マッチンググラフ G_M を作り , その最大マッチングを求めて , ラベル配置を行う .
- (7) 次に述べる後処理を実行する .

(7) において , 非常に接近して配置されているラベル λ_1 と λ_2 に対して以下の手続きを実行する . この後処理は配置される頂点ラベルの数を増やすものではない .

手続き $\text{separate}(\lambda_1, \lambda_2)$

1. 一般性を失うことなく , λ_1 は λ_2 よりも上にあるものとする .
2. ht_{λ_1} を λ_1 の高さとする . 移動後の位置 λ'_1 が次の条件を全て満たすなら , λ_1 を $ht_{\lambda_1}/5$ だけ上に移動させる .
 - (a) λ'_1 は対応するラベル空間からはみ出ない .
 - (b) λ'_1 は他のラベルと重ならない .
 - (c) α (α') を λ_1 (λ'_1) と非常に接近しているラベルの数とする . このとき , $\alpha \geq \alpha'$ を満たす .
 - (d) β (β') を λ_1 (λ'_1) と非常に接近している頂点と辺の数とする . このとき , $\beta \geq \beta'$ を満たす .

上記の条件のうち 1 つでも満たせないものがあるなら , 同じ条件のもとで λ_1 を上へ $ht_{\lambda_1}/10$ だけ移動できないか試みる .

3. 2. と同様にして λ_2 を下へ移動させる .

3.3.2 計算機実験

ラベル配置率と実行時間に関して，提案手法 2 と KM 法 (2.3.1) 及び KT 法を比較するため計算機実験を行った．本節ではその結果について述べる．なお，ここでの KT 法は，頂点のみをラベル配置の対象としている．

3.3.2.1 実験方法

頂点数，辺の数を様々に変えて多数のグラフを作成し，Kamada-Kawai [3] のグラフ描画法を用いて描画を行った．描画領域は 1 辺が 10 の正方形領域と定めた．各頂点ラベルの高さは 1 つの描画中で全て同じであるものとし，0.1, 0.15, 0.2, 0.25, 0.3 の 5 通りの値を用いた．各ラベルの幅はラベルの高さの 2 ~ 5 倍の範囲で個々に定めた．使用計算機の CPU は Celeron 500MHz，使用計算機言語は C 言語である．

3.3.2.2 実験結果

表 3.4 と表 3.5 に，頂点数 40，辺の数 60 の場合の結果を示す．表中の各値は 100 個のグラフに対する平均値である．KT 法のステップ 1 において，各頂点に対して作成したラベル候補数は 10 である．KT 法と提案手法 2 において，グラフ G_M の最大マッチングを求める際には，Hopcroft-Karp [26] のアルゴリズムを用いた．

表 3.4: ラベル配置率

ラベルの高さ	KM 法	KT 法	提案手法 2	上限値
0.10	83.50%	95.95 %	98.95 %	98.95 %
0.15	74.90%	87.83 %	94.80 %	95.08 %
0.20	67.80%	78.70 %	88.05 %	88.55 %
0.25	61.33%	71.42 %	80.23 %	81.05 %
0.30	56.02%	65.10 %	73.15 %	74.25 %

KM 法は単純であるため，実行時間は非常に高速である．しかし，KT 法及び提案手法 2 も十分高速であるといえる．提案手法 2 は全ての場合において，KM 法及び KT 法より高いラベル配置率を達成できているが，特にラベルのサイズが大きい場合には，その差が顕著になっている．

提案手法 2 においてラベル空間が作成できない頂点に対しては，そのラベルを描画中に配

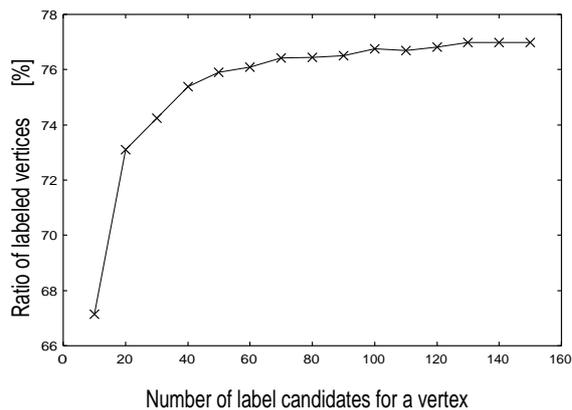
表 3.5: 実行時間

ラベルの高さ	KM 法	KT 法	提案手法 2
0.10	19.0 msec	30.8 msec	26.9 msec
0.15	17.4 msec	30.8 msec	27.6 msec
0.20	18.4 msec	28.2 msec	26.9 msec
0.25	17.1 msec	27.9 msec	26.6 msec
0.30	16.0 msec	26.4 msec	26.6 msec

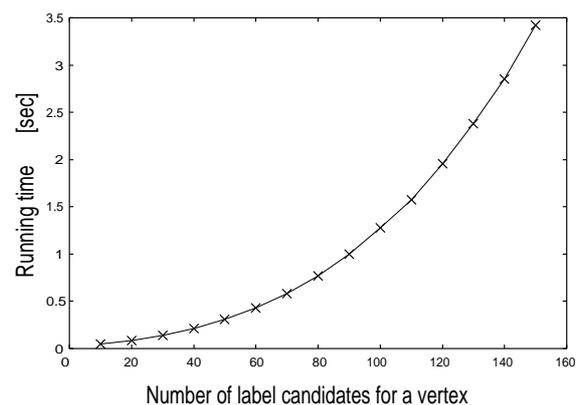
置ることができない．よって，次の値をラベル配置率の上限値として用いることができる．

$$\frac{\text{ラベル候補が作成可能な頂点の数}}{\text{頂点の総数}}$$

表 3.4 の「上限値」は，このようにして得られた値である．この上限値はやや緩いものであるが，提案手法 2 で得られるラベル配置率はこの値に非常に近いものになっている．



(a) ラベル配置率



(b) 実行時間

図 3.11: KT 法において作成するラベル候補数を増やした場合の実験結果

KT 法は，各頂点に対して作成するラベル候補数を増加させることでより多くのラベルを配置することができる．図 3.11 は，そのようにした場合のラベル配置率及び実行時間の变化を示している（頂点や辺と重なるラベル候補も含めてカウントしている）．この図は，頂点数 50，辺の数 70 のグラフ 100 個を用いた場合の結果である（ラベルの高さは 0.25 である）．

同じ入力に対し，提案手法 2 で得られる配置率は 77.54% であり，実行時間は 35.6msec である．KT 法において，1 つの頂点に対し 150 個のラベル候補を作成した場合でも，提案手法 2 のラベル配置率を上回ることができなかった．さらに，そのように多くのラベル候補を作成した場合の KT 法の実行時間は，提案手法 2 の及び 96 倍に達している．

3.4 頂点と辺の両方がラベルをもつ場合のラベル配置アルゴリズム

本節では，頂点と辺の両方がラベルをもつ場合のラベル配置アルゴリズムを2つ示す．以下これらの方法を提案手法3-1及び提案手法3-2と呼ぶことにする．以下では，3.4.1，3.4.2において，提案手法3-1及び提案手法3-2について説明する．3.4.3では，まず，グラフ描画へのラベル配置に適用できるように，Kameda-Imaiのアルゴリズムを変更する方法について述べる．その後，その方法及びKT法，WW法(2.3.3)と提案手法との比較実験の結果を示す．

3.4.1 提案手法3-1

提案手法3-1は，基本的に，3.2で述べた辺ラベル配置法と3.3で述べた頂点ラベル配置法を組み合わせたものである．ただし，

- 辺のラベルを，頂点を中心とする半径 ϵ の円と重ならないように配置するものとしているため，辺に対するラベル空間をそのような円と重ならないように作成する．
- 後処理には，3.3.1.4で述べた手続き $\text{separate}(\lambda_1, \lambda_2)$ を用いる．

提案手法3-1の手順をまとめたものを以下に示す．

提案手法3-1

- (1) G の各辺及び各頂点に対し，3.2.1.1，3.3.1.1で述べた方法でラベル候補を作成する．
- (2) 0次候補と，それ以外に少なくとも1つのラベル候補をもつ辺あるいは頂点が残っている限り，3.2.1.2及び3.3.1.2で述べた0次候補の選択処理(**)を実行する．
- (3) クラスタを作成する．
- (4) 要素数が2以上の各クラスタ C に対し，ラベル候補の移動が行われなくなるまで，手続き $\text{shift}(C)$ を実行する（辺に対するラベル候補の移動には3.2.1.3で述べた処理を，頂点に対するラベル候補の移動には3.3.1.3で述べた処理を，それぞれ行う）．1つでもラベル候補を移動させることができれば(2)に戻る．
- (5) 2.3.2で述べた処理(*)を，各非完全クラスタ C に対し1度だけ実行する．1つでもラベル候補を消去できれば(2)に戻る．
- (6) マッチンググラフ G_M を作り，その最大マッチングを求めて，ラベル配置を行う．
- (7) 後処理（手続き $\text{separate}(\lambda_1, \lambda_2)$ ）を実行する．

3.4.2 提案手法 3-2

提案手法 3-1 には、以下のように改善の余地がある。

- 狭いラベル空間は一般に混雑した場所に作られることが多いため、狭いラベル空間内でのラベル配置を減らすことによって、各ラベルがどの頂点あるいは辺のものなのかがより分かりやすくなると考えられる。
- 頂点と辺の両方のラベルを配置する場合、頂点のみあるいは辺のみの場合に比べ、ラベル候補がかなり多くなり、グラフ G_R のクラスタ中の頂点数が増える。そのため、各クラスタの頂点に対応するラベル候補のうち、最も上あるいは下のものだけにシフトを試みても、ラベル候補同士の重なりを大きく減らすことができず、ラベル配置率をあまり向上させることができない。
- G_R の各クラスタがクリークをなしたときに、マッチンググラフの最大マッチングを求めることによりラベル配置を求めると、各クリークの頂点に対応するラベル候補のうち高々1つにしかラベルを配置することができない。 G_R の全てのクラスタがクリークとなった後にも、次数の高い頂点と対応するラベル候補の削除、及びラベル候補のシフトを繰り返すことによって、より多くのラベルを配置できる可能性がある。

提案手法 3-2 は、以上のことを考慮して、提案手法 3-1 に以下のような改良を施したものである。

- 3.4.1 で述べた提案手法 3-1 の手順における (2) において、選ばれた 0 次候補があるしきい値 th より狭いラベル空間内にある場合、ラベル候補の削除は行わない。
 - 3.4.1 で述べた提案手法 3-1 の手順における (4), (5), (6) を以下の処理と置き換える。
- (4) ラベル候補のシフトを行うことにより、ラベル候補同士の重なりを回避を試みる。ここでは、他の候補と重なっている全ての候補に対してシフトを試みる。もし重なりを減らすことができれば (2) へ戻る。
 - (5) G_R が頂点数 2 以上のクラスタをもたなければ (6) へ進む。そうでなければ、 G_R の頂点数 2 以上の各クラスタから、最も次数の高い頂点（対応するラベル候補が、非常に少数のラベル候補のみをもつ G の頂点あるいは辺のものである場合には、次数が次に高い頂点。また、該当する頂点が複数ある場合には、それらのうちで最も狭いラベル空間内にあるラベル候補に対応する頂点）1 つを削除し、(2) に戻る。
 - (6) G の各頂点及び各辺に対し、2 つ以上のラベル候補が残されていれば、最も広いラベル空間内にあるラベル候補のみを残し、他の（その頂点あるいは辺の）ラベル候補を削除する。そして、残されたラベル候補の位置に各ラベルを配置する。

3.4.3 計算機実験

3.4.3.1 Kameda-Imai のアルゴリズムに対する変更

グラフ描画へのラベル配置に適用できるようにするため，2.3.4 で述べた KI 法に以下のような変更を加える．

- 折れ線に対してではなく，辺に対してラベルを配置する．その際，辺に沿って 1 文字ずつ配置するのではなく，全ての文字を横 1 行に並べて配置する．
- 頂点及び辺のラベルを，第 1 章で示したラベル配置問題中の条件 (a)~(c) を満たすように配置する．

さらに，解の質を向上させるため，以下のような改良を行う．

- KI 法では，図 3.12(a) のような場合，同図 (b) 及び (c) の斜線部分のように 2 つのラベル空間があるものと見なしている．ラベルと頂点との距離が ϵ になるように変更すると，2 つのラベル空間は，例えば，同図 (d), (e) のようになる．本研究では，これらの代わりに同図 (f) のようなラベル空間 1 つだけを作るものとする（本研究における予備実験によれば，このようにした方がラベル配置率が若干改善された）
- ラベル同士の近接を減らすため，提案手法 3-1 及び提案手法 3-2 と同じ後処理を行う．

以下，この方法を MKI 法と呼ぶことにする．

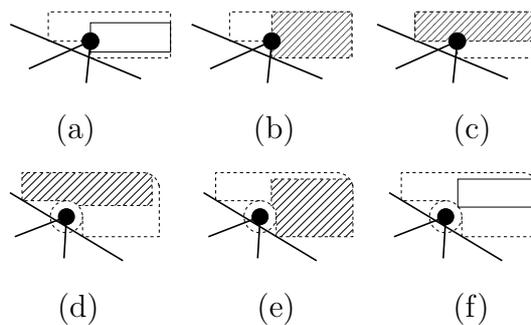


図 3.12: KI 法におけるラベル空間の修正

3.4.3.2 実験方法

計算機実験により，KT 法，WW 法，提案手法 3-1，提案手法 3-2 及び MKI 法を比較した．なお，KT 法及び WW 法では，各頂点に対して 10 個のラベル候補を作成した．

頂点数，辺の数の様々な組み合わせについて，グラフを100個ずつランダムに作成し，それらを Kamada-Kawai [3] の描画法で描画したものを入力データとして，ラベル配置率の平均値と実行時間の平均値を求めた．なお，全ての場合について，描画は1辺の長さ10の正方形領域内で行った．ラベルの高さは全ての頂点及び辺で等しいものとし，その値は0.2若しくは0.3とした．また，ラベルの幅は高さの2~5倍の範囲で頂点，辺ごとに定めた．使用計算機のCPUはDron 800MHz，使用計算機言語はC言語である．

3.4.3.3 実験結果

ラベル配置率の結果を表3.6に示す．この表から，以下のことが分かる．

- 前述のように，平面上の点にラベル配置をする場合，WW法はKT法より高い配置率を達成できるとの報告があるが[19]，グラフ描画中に頂点と辺のラベルの両方を配置する場合も同様である．
- ラベル候補を機械的に作る方法（KT法，WW法）に比べ，頂点や辺の周囲の状況を考慮した方法（提案手法3-1，提案手法3-2，MKI法）の配置率は大幅に高くなっている．後者の3つの方法ではあまり差はないが，提案手法3-1及び3-2はMKI法より若干高い配置率を示しており，特にグラフが疎である場合，その差が少し大きくなっている．これは，グラフが疎である場合，提案手法3-1及び3-2において，ラベル候補をシフトできる範囲が広がるためであると考えられる．

表 3.6: ラベル配置率 [%]

頂点数 辺の数	30				40				50			
	35		50		45		60		55		70	
ラベルの高さ	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3
KT法	90.23	74.65	78.74	58.69	83.86	63.74	73.03	51.65	77.96	55.12	67.02	45.36
WW法	90.91	75.97	79.15	59.19	85.05	65.73	73.51	52.32	79.41	57.47	67.67	46.13
提案手法3-1	96.38	84.97	87.51	68.53	91.66	75.88	82.58	60.88	87.27	67.21	76.82	54.52
提案手法3-2	96.38	85.25	87.55	68.69	91.78	76.41	82.60	61.22	87.32	68.21	77.08	54.85
MKI法	95.78	83.91	86.89	68.20	90.89	74.34	82.07	60.47	86.28	65.91	76.35	54.15

次に，実行時間の比較を表3.7に示す．この表より，いずれの方法も十分高速であることが分かる．ラベル配置率が高かった3つの方法を比べてみると，貪欲算法であるMKI法が最も高速であるが，提案手法3-1及び3-2との差はさほど大きくはない．

複数の頂点及び辺に接近して配置されているラベルは，どの頂点あるいは辺のものであるのかが分かりづらい．そこで，各ラベルがどの頂点あるいは辺のものであるのかが，どの程度分かりやすくなっているかを評価するため，各ラベルを0.025, 0.05, 0.075若しくは0.1だけ上下左右に拡大したとき（そのラベルをもつ頂点，辺以外の）頂点や辺と重なる個数の平均値を調べた．また，異なるラベル同士が近付きすぎていることも一般に望ましくないため，上と同様

表 3.7: 実行時間 [ms]

頂点数	30				40				50			
辺の数	35		50		45		60		55		70	
ラベルの高さ	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3
KT 法	33.4	20.4	48.0	23.6	44.6	28.3	60.7	31.8	61.1	37.7	72.3	40.3
WW 法	20.6	14.6	30.0	17.9	30.3	21.7	40.1	25.8	43.1	32.8	50.1	33.1
提案手法 3-1	15.9	17.2	21.7	21.6	25.4	31.2	32.6	29.9	41.9	47.7	44.7	40.1
提案手法 3-2	18.3	19.4	24.2	22.4	29.0	30.7	35.8	32.4	42.4	46.4	48.8	40.8
MKI 法	15.0	15.0	23.1	19.9	23.3	22.0	31.8	26.8	33.2	31.5	39.6	33.2

に、各ラベルを 0.025, 0.05, 0.075 若しくは 0.1 だけ上下左右に拡大したとき、他のラベルと重なる個数の平均値も調べた。これらの値は配置されているラベル数に大きく影響されるため、配置率が高かった 3 つの方法についてのみ比較することとした。0.05 だけ拡大したときのその結果を表 3.8 及び表 3.9 に示す。これらの表を見ると、頂点及び辺との重なりについてはわずかながら提案手法 3-2 が、ラベルとの重なりに関しては KI 法が、それぞれ優れている。提案手法 3-1 と提案手法 3-2 とを比較すると、提案手法 3-2 の方がラベル配置数が若干多いのにも関わらず、頂点及び辺との重なりに関して提案手法 3-1 より少ない値を示している。これは、3.4.2 で述べた一つ目の改良によるものと考えられる。

表 3.8: ラベルを拡大したときの、頂点及び辺との重なり個数

頂点数	30				40				50			
辺の数	35		50		45		60		55		70	
ラベルの高さ	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3
提案手法 3-1	0.232	0.286	0.347	0.364	0.296	0.354	0.416	0.398	0.353	0.405	0.472	0.445
提案手法 3-2	0.207	0.263	0.308	0.304	0.281	0.329	0.379	0.344	0.340	0.390	0.426	0.387
MKI 法	0.194	0.267	0.315	0.361	0.259	0.336	0.388	0.405	0.322	0.421	0.450	0.475

表 3.9: ラベルを拡大したときの、他のラベルとの重なり個数

頂点数	30				40				50			
辺の数	35		50		45		60		55		70	
ラベルの高さ	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3
提案手法 3-1	0.202	0.272	0.191	0.212	0.334	0.421	0.269	0.284	0.459	0.484	0.348	0.327
提案手法 3-2	0.216	0.297	0.210	0.248	0.355	0.459	0.300	0.318	0.477	0.542	0.374	0.374
MKI 法	0.065	0.156	0.082	0.118	0.157	0.249	0.136	0.160	0.241	0.322	0.192	0.206

3.5 結言

本章では、あらかじめ定められたグラフ描画中に、どのラベルも頂点や辺及び他のラベルと重ならないようにして、できるだけ多くのラベルを配置するという問題について考察を行った。そして、KT法を改良することにより、辺のみあるいは頂点のみがラベルをもつ場合のアルゴリズムをまず提案し、次に辺と頂点の両方がラベルをもつ場合のアルゴリズムを示した。

辺のみあるいは頂点のみがラベルをもつ場合に対しては、KT法に比べ、ラベル配置率が大幅に改善でき、かつ実行時間の短縮も可能になることを計算機実験により示した。さらに、ラベル配置率の(緩い)上限値を実験的に求め、提案手法によるラベル配置率が非常に近い値になることを示した。頂点と辺両方がラベルをもつ場合に対しては、そのような上限値の計算はしなかったが、KT法以外に、地図ラベル配置アルゴリズムとして開発されたWW法、及びKameda-Imaiの方法を変更したMKI法との比較実験を行った。そして、提案手法がこれらの方法より優れたラベル配置率を達成できることを示した。

以上の計算機実験で用いたグラフ描画は、与えられたグラフに対し、ラベルの情報を考慮することなく求めたものである。表3.2や表3.4より分かるように、グラフが密である場合や、ラベルのサイズが大きい場合、ラベル配置率の上限値ですら十分に大きい値にならないことがある。これは、配置率のさらなる改善のためには、分離型アプローチとは別のアプローチが必要であることを示している。したがって、本研究では、次章において、ラベルの情報を考慮してグラフ描画を求める問題を考える。

第4章 ラベルの存在を考慮したグラフ描画アルゴリズム

4.1 緒言

前章では，グラフの描画が与えられるものとして，そこにラベルを配置する問題について考えた．しかし，表 3.2 や表 3.4 より分かるように，グラフが密である場合や，ラベルのサイズが大きい場合，ラベル配置率の上限値ですら十分に大きい値にはならないことがある．したがってこのような場合には，ラベルの存在を考慮せずにグラフ描画を求め，その後ラベル配置を行うという従来の方法では，配置できないラベルが多く残ることになる．そこで，本章では，頂点と辺の位置の決定段階でラベルの存在を考慮することにより，ラベルの配置により適したグラフ描画を求める方法を提案する．

本章で提案する各アルゴリズムは，川西ら [5] のグラフ描画アルゴリズムをもとにして設計したものである．川西らの描画法を選択したのは，この方法が他の方法に比べ，次のような長所をもつためである．

- 文献 [5] では 5 つの評価基準を用いて，川西らの描画法を他の方法と比較しているが，4 つの評価基準に関して，川西らの方法が最も優れている．
- 川西らの描画法は，頂点と辺の近接を避けるための工夫をしているため，他の方法よりも頂点や辺のラベルを配置するのに適していると考えられる．

本章の構成は次の通りである．まず，4.2 において辺ラベルの存在を考慮したグラフ描画アルゴリズムを提案し，計算機実験により，これまでの方法よりも辺ラベルの配置に適した描画が得られることを示す．頂点ラベルに対しても同様に，4.3 においてアルゴリズムの提案と計算機実験を行う．

4.2 辺ラベルの存在を考慮したグラフ描画アルゴリズム

本節では，辺ラベルの存在を考慮したグラフ描画アルゴリズムについて説明する．以下この方法を提案手法 4 と呼ぶことにする．

4.2.1 アルゴリズムの説明

4.2.1.1 アルゴリズムの概略

前述のように，提案手法 4 は川西らの描画法をもとにしたものであり，前半（繰り返し回数 I がある定数 M_1 未満のとき）では，2.2.3 で述べた力 $f_{i,j}$ 及び $f'_{i,j}$ を頂点の移動に用いる．一方，後半（ $I \geq M_1$ のとき）では，これらの力に $f''_{i,k}$ を加え，さらに各辺のラベルの配置位置を想定しながら頂点の移動を行う．そのため，各辺に対しラベルの想定位置を表す仮頂点（図 4.1 のようにラベルの中央に対応する）を導入する．

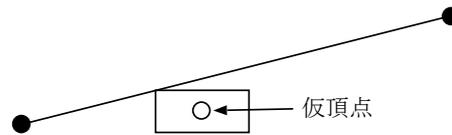


図 4.1: 仮頂点

辺のラベルを配置する際，周辺に他の辺があれば障害になる．そこで，提案手法 4 では，仮頂点と付近を通る（その仮頂点をもつ辺以外の）辺との間に斥力を働かせる．そして，その力に従って辺の端点を移動させ，ラベルを配置するための領域を確保するという方針をとっている．

描画領域は，1 辺の長さが D の正方形領域に固定されているものとする．提案手法 4 の概略は次の通りである．

- (1) 全ての 2 頂点 i, j について，理想距離 $l_{i,j}$ を求める．
- (2) 各頂点 i と，それに接続していない各辺 e とについて， e の中点 k と i との間の理想距離 $l'_{i,k}$ を求める．
- (3) 頂点を半径 $D/2$ の円の周上に等間隔に初期配置する．
- (4) $I = 1, 2, \dots, M_1 - 1$ について，以下の (4-a) を実行する．
 - (4-a) 各頂点 i に対して，以下を実行する．
 - (4-a-1) i 以外の各頂点 j に対して $f_{i,j}$ あるいは $f'_{i,j}$ を計算し，それらの合力を求めて F_i とする．
 - (4-a-2) F_i の向きに，その大きさの C_4 倍だけ i を移動させる．
- (5) $I = M_1, M_1 + 1, \dots, M$ について，以下の (5-a) ~ (5-c) を実行する．
 - (5-a) 各頂点 i に対して，以下を実行する．

- (5-a-1) i 以外の各頂点 j に対して $f_{i,j}$ あるいは $f'_{i,j}$ を計算し, それらの合力を求めて F_i とする.
- (5-a-2) 各辺の中点から i に働く力 $f''_{i,k}$ を計算し, その合力を F_i に加える.
- (5-a-3) F_i の向きに, その大きさの C_4 倍だけ i を移動させる.
- (5-b) 描画範囲の大きさを調べ, その結果に基づいてあるパラメータ r の値を計算する.
- (5-c) 各頂点 i に対して, 以下を実行する.
 - (5-c-1) i に接続する各辺 e について, 以下を実行する.
 - (5-c-1-1) $I = M_1$ であるか, I がある小さな自然数 C_5 (定数) の倍数であるならば, e の仮頂点の位置を決定する.
 - (5-c-1-2) e の仮頂点の位置を調整する.
 - (5-c-1-3) e 及びその仮頂点に働く力を求める.
 - (5-c-2) i に接続する各辺 e 及び仮頂点に対し (5-c-1-3) で求めた力に基づいて, i の座標を更新する.
- (6) 頂点と辺の近接を緩和する処理を行う.
- (7) 各辺を直線で描く.

上記の手順のうち, (1) ~ (3) は, 2.2.3 で述べた川西らの描画法と同様, 前処理及び頂点の初期配置を行う部分である (初期配置を行う方法も川西らの描画法と同様である). (4) は提案手法4の前半部分であり, 頂点間に働く力のみに従って頂点の移動を行っている. (5)以降が後半部分であり, (5-c)において, 仮頂点の決定や, 仮頂点と辺の間の力に基づく頂点の移動などを行っている.

以下では, まず 4.2.1.2 において, 各辺の仮頂点の位置の決定方法 (上記の手順の (5-c-1-1)) について述べる. 4.2.1.3 では, 仮頂点をより望ましい位置に移動させる方法 ((5-c-1-2)) について説明する. 4.2.1.4 では, ラベルの配置位置を確保するための力 ((5-c-1-3)) を定義し, (5-c-2) における頂点の移動方法について述べる. 4.2.1.5 では, 描画範囲 (実際に描画が占める領域) の大きさに基づいて行うあるパラメータの計算 ((5-b)) について述べる. 4.2.1.6 では, 描画における頂点と辺の近接を減らすために, (6) で実行している方法について説明する. 最後に 4.2.1.7 において, 提案手法4全体のより詳しい記述を示す.

4.2.1.2 仮頂点の位置の決定

(a) ラベル空間による仮頂点の位置の決定

提案手法4の (5-c-1-1) では, 各辺に対して仮頂点の位置の決定 (あるいは再決定) を行う.

3.2.1.1 で述べたラベル空間が作成可能な辺については，最も広いラベル空間の中央に仮頂点を作る．一方，ラベル空間が作成できない辺については，仮頂点の位置の候補をいくつか定め，それらのうち最も望ましいと思われるものを選択することによって，仮頂点の位置を決定する．これらの処理の詳細について次の (b) で述べる．

(b) ラベル空間が作成できない場合

辺 $e = (i, j)$ に対して，ラベル空間が作成できない場合，仮頂点の位置の候補を以下のようにして作る．

最初に， e に対して仮頂点が作られていない場合には， e の中央の両側に候補をおく．一方，仮頂点が既に作られている場合には，仮頂点の位置の候補として，現在の仮頂点の位置と，辺を挟んでその反対側の位置を考える．例えば図 4.2(a) の辺 e に対して，現在の仮頂点の位置が cp_1 であったとすると， cp_1 とその反対側の cp_2 が候補となる．

次に e と他の辺との交点を全て求める．そして図 4.2(b) の $cp_3 \sim cp_8$ のように， e の端点と交点に挟まれた各部分の左右（あるいは上下）両側の中央の位置を候補に加える．

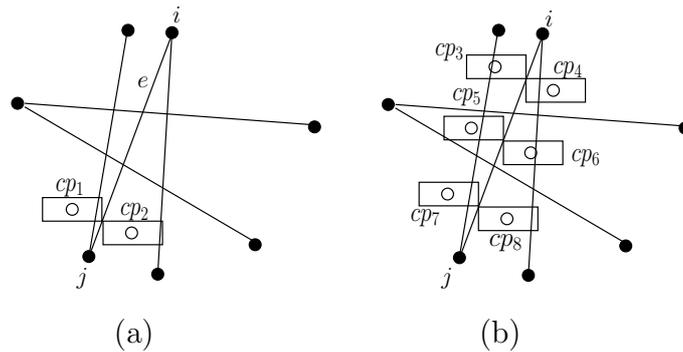


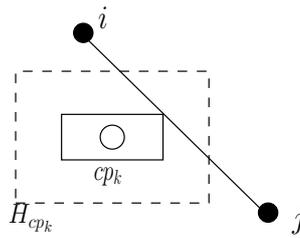
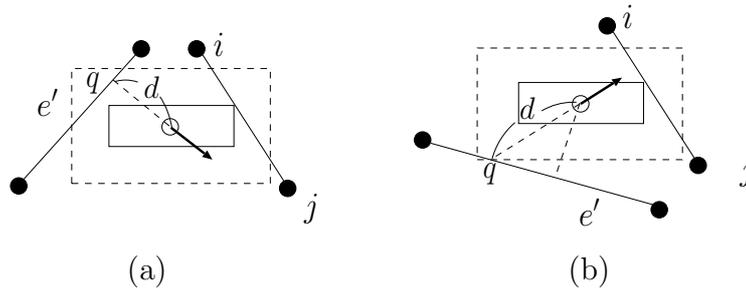
図 4.2: 仮頂点の位置の候補

このようにして定めた各候補 cp_k にラベルを置いたとき，それと重なる辺が最も少ないものを，仮頂点の位置として選ぶことにする．そのような候補が複数存在する場合には，それらの候補が周囲の各辺 e' から受ける力 $F^{(1)}(cp_k, e')$ を求め，その大きさの和 $\sum_{e'} |F^{(1)}(cp_k, e')|$ が最も小さい位置を選択する．ただし，大きく離れた辺からの力は意味がないので，図 4.3 の破線のように，その候補にラベルを置き，ラベルを上下左右にラベルの高さだけ拡大した範囲 (H_{cp_k} と表す) を考え，それと重なる辺のみから力を受けるものとする．

さて，仮頂点あるいは候補 cp_k が辺 e' から受ける力 $F^{(1)}(cp_k, e')$ の定義であるが，まず次のような力 F を求める．

$$F = C_6 / C_7^d \quad (4.1)$$

ここで C_6, C_7 は正の定数であり， $C_7 > 1$ である． d は， H_{cp_k} の内部（周上を含む）にある

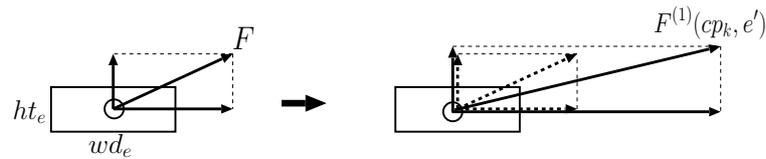
図 4.3: cp_k が影響を受ける範囲 H_{cp_k} 図 4.4: 距離 d の決定

e' 上の点のうち cp_k に最も近いもの q と cp_k との距離である．図 4.4 に 2 つの例を示す． F の向きは，図 4.4 に矢印で示したように， q から cp_k に向かう方向である．

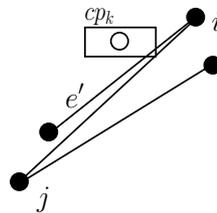
なお，ここでは F を式 (4.1) のように定めたが，(Eades [2] の描画法における非隣接頂点間の力のように，) $F = C/d^2$ (C : 定数) のような関数を用いることも考えられる．しかし，この関数を用いると， d が非常に小さい場合， F が非常に大きな力になってしまう．後述するように，提案手法 4 では頂点の移動も F をもとに行っているため，この力が大きくなりすぎる場合があることは望ましくない．そこで， d がどのような値になっても大きさが C_6 をこえることのない，式 (4.1) の関数を用いることにした．

ここで，ラベルの形状に着目する．現在注目している仮頂点あるいは候補 cp_k をもつ辺 e のラベルの幅を wd_e ，高さを ht_e としたとき，これらの値を力 $F^{(1)}(cp_k, e')$ に反映させるため，まず図 4.5 左側に示すように， F を x 成分， y 成分に分解する．次に同図右に示すように， x 成分は $C_8 \cdot wd_e$ 倍， y 成分は $C_8 \cdot ht_e$ 倍し (C_8 : 正の定数)，それらを合成した力を求めて $F^{(1)}(cp_k, e')$ とする．この計算では，ラベルが横長であれば x 成分の拡大の度合がより大きく，縦長ならば y 成分の拡大の度合がより大きくなる．このように， $F^{(1)}(cp_k, e')$ はラベルの形状をある程度反映した力となっている．以後，図 4.5 に示した処理を形状反映処理と呼ぶことにする．

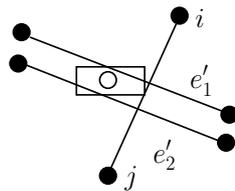
力 $F^{(1)}(cp_k, e')$ の決定法は原則として以上のとおりであるが，例外を 1 つ設けている．図 4.6

図 4.5: 力 $F^{(1)}(cp_k, e')$ の決定

のような場合， $F^{(1)}(cp_k, e')$ はそれほど大きな力にならないが，ラベルの位置としては望ましくない．このような位置にラベルを配置しようとするとき，通常頂点を大きく移動させることが必要になり，描画に与える影響が大きくなるからである．そこで，この図のように，辺 e' が cp_k と e の間を通るときには，ペナルティとして， $|F^{(1)}(cp_k, e')|$ をやや大きな値 $pena$ にする．

図 4.6: 辺 e' が cp_k と e の間を通過している場合

範囲 H_{cp_k} に重なる各辺 e' に対して $F^{(1)}(cp_k, e')$ を計算した後，前述のように，それらの大きさの和 $\sum_{e'} |F^{(1)}(cp_k, e')|$ を求める．ここで，ベクトルとして和を求めているのは，図 4.7 の e'_1, e'_2 のような辺があった場合，それぞれから受ける力が打ち消しあってしまい，この図に示したような候補が仮頂点の位置として選ばれやすくなってしまふからである．

図 4.7: $F^{(1)}(cp_k, e'_1)$ と $F^{(1)}(cp_k, e'_2)$ が打ち消しあう例

(c) 仮頂点の削除と復活

提案手法 4 における仮頂点の位置の選択方法は (a) と (b) で述べたとおりであるが，これから述べる仮頂点の削除の処理を，アルゴリズムの後半で 1 回だけ行うことにしている．この処理は，辺 e に対してラベル空間が作成できず，更に e の仮頂点のどの候補 cp_k についても $\sum_{e'} |F^{(1)}(cp_k, e')|$ があるしきい値 th_1 よりも大きいとき， e の仮頂点を削除するという

ものである．この条件を満たす場合， e のラベルを配置することは困難であると考えられる．更に，そのような辺の仮頂点を残しておくこと，それが周囲の辺に力を及ぼし，他の複数の辺のラベル配置を阻害してしまうことがある．仮頂点の削除は，このような理由によるものである．

なお，仮頂点が削除されている辺 e について，後に，ラベル空間が作成できた，若しくは (b) で述べたようにして仮頂点の候補を作ったとき $\sum_{e'} |F^{(1)}(cp_k, e')|$ の最小値があるしきい値 th_2 よりも小さくなったとする．これらのことは， e の周囲の状況が大きく変わり， e のラベルを配置することがさほど困難ではなくなっていることを意味する．よって，このような場合には，(a) 及び (b) で述べた方法により， e の仮頂点を作ることにする．一度仮頂点を削除した辺に対して再び仮頂点を作るため，この処理を仮頂点の復活と呼ぶ．

4.2.1.3 仮頂点の位置の調整

辺 $e = (i, j)$ の仮頂点を p_e と表す．本節では， p_e を e 上のより望ましい位置に移動させるための 2 種類の力と，それらに基づいた移動方法について述べる．

(a) 周囲の辺から仮頂点に働く力

仮頂点が他の辺に近づくことを防ぐため，周囲の辺 e' から仮頂点に働く力 $F^{(2)}(p_e, e')$ を定義する．

まず，4.2.1.2(b) で述べた方法により，仮頂点 p_e が周囲の各辺 e' から受ける力 $F^{(1)}(p_e, e')$ を求める．ただし，力 F を計算する際には，式 (4.1) の C_6, C_7 の代わりに，それぞれ別の定数 C_9, C_{10} を用いる (d は前述のように定める)．更に，図 4.6 のような場合でもペナルティを課することはしない．

次に，図 4.8(a) に示すように，各 $F^{(1)}(p_e, e')$ に対して e に平行な向き成分を求め，これを $F^{(2)}(p_e, e')$ とする．

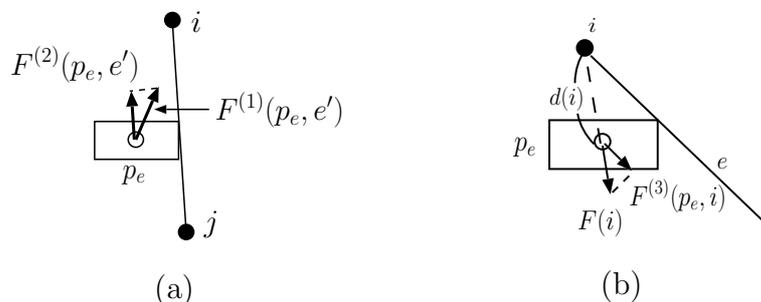


図 4.8: 仮頂点の位置の調整のための 2 種類の力

(b) 辺の端点から仮頂点に働く力

仮頂点が辺の端点に近づきすぎるのはあまり望ましいことではない．その位置にラベルを

配置すると、同じ頂点に接続する他の辺のラベルと見誤りやすくなるためである。また、本研究では頂点のラベルについて考えていないが、一般には頂点と辺の両方がラベルをもつこともあり、そのような場合に、ラベルが頂点、辺いずれのものなのかが分かりにくくなる。そこで、 e の両方の端点 i, j から p_e に斥力を働かせることにする。

まず、以下のような力を求める。

$$F(i) = C_9/C_{10}^{d(i)} \quad (4.2)$$

$$F(j) = C_9/C_{10}^{d(j)} \quad (4.3)$$

ここで、 C_9, C_{10} は (a) で述べたのと同じ定数であり、 $d(i)$ は i と p_e との距離、 $d(j)$ は j と p_e との距離である。 $F(i), F(j)$ の向きは、それぞれ i, j から p_e に向かう方向である。そして、これらの力 $F(i), F(j)$ に対し形状反映処理を実行し、更に e に平行な向きの成分を求め、 p_e に i から働く力 $F^{(3)}(p_e, i)$ 、 p_e に j から働く力 $F^{(3)}(p_e, j)$ として定める。図 4.8(b) 参照（図を簡単にするため、ここでは形状反映処理を省略している）。

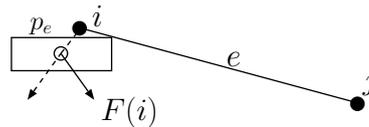


図 4.9: $F(i)$ の向きの変更

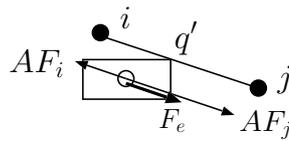
p_e の x 座標が i, j いずれの x 座標よりも小さい（あるいは大きい）ような特別な場合には、 $F(i), F(j)$ の求め方を変更する必要がある。例えば、図 4.9 の場合には i から j へ向かう方向の力を p_e に加えたいところであるが、上で定義した $F(i)$ は図中の破線の矢印のような力になり、 e に平行な成分を求めると、 j から i に向かう方向の力になってしまう。このようなことを避けるため、 p_e の x 座標が i, j いずれの x 座標よりも小さい（あるいは大きい）場合、以下に述べる処理 (*) によって、力の方向を変えることにする。

- (*) 一般性を失うことなく、 p_e との x 座標の差が小さい方の端点を i とする。このとき、 $F(i)$ の x 成分を逆向きにし、 y 成分をそのままにした力（図 4.9 に実線の矢印で示す）を $F(i)$ として用いる。

p_e の y 座標の値が i, j いずれよりも小さい（あるいは大きい）場合も同様に、（ p_e との y 座標の差が小さい方の端点を i としたとき、） $F(i)$ の y 成分を逆向きにした力を新たに $F(i)$ として用いる。

(c) 仮頂点の移動

各辺 $e = (i, j)$ に対し、仮頂点 p_e のより望ましい位置への移動は次の手順で行う。

図 4.10: 力 F_e

- (i) p_e が周囲の各辺 e' から受ける力 $F^{(1)}(p_e, e')$ を求め、更にそれから $F^{(2)}(p_e, e')$ を計算する。
- (ii) p_e が 端点 i 及び j から受ける力 $F^{(3)}(p_e, i)$, $F^{(3)}(p_e, j)$ を求める。
- (iii) 各辺 e' に対して求めた $F^{(2)}(p_e, e')$, 及び $F^{(3)}(p_e, i)$, $F^{(3)}(p_e, j)$ のうち, i から j に向かう方向の最大の力を AF_j とし, 逆方向の最大の力を AF_i とする. このとき, AF_j と AF_i の合力を F_e とする. 図 4.10 参照.
- (iv) 仮頂点 p_e を, 力 F_e の向きに, その大きさの C_4 倍だけ移動させる. ただし, p_e の位置にラベルを配置したとき, ラベルが辺と接する点 (図 4.10 における点 q') が e の端点をこえてはみ出すことになれば, その端点の少し手前まで戻す.

4.2.1.4 頂点の移動のための力

4.2.1.1 で概略を示したように, 提案手法 4 の後半 ($I \geq M_1$ のとき) では, 変数 I の各値について, 各頂点 i の移動を通常 2 回行う ((5-a-3) と (5-c-2)). 1 回目の移動は, 川西らの描画法におけるのと同様, 頂点間の力及び頂点と辺の間の力によるものである. 一方, 2 回目の移動は, ラベルの配置場所の確保を目的とするもので, i に接続する辺の仮頂点が周囲の辺から受ける力, 及び, i に接続する辺が周囲の仮頂点から受ける力による. 本節では, これら 2 種類の力について説明した後, これらの力に従ってどのように i を移動させるかについて述べる.

(a) 周囲の辺から働く力

ここでは, 辺 $e = (i, j)$ の仮頂点 p_e に周囲の各辺 e' から働く力に基づいて, 頂点 i に働く力 $F^{(4)}(i, e')$ を定義する. まず p_e と e' に対して, 4.2.1.2(b) で定義した力 $F^{(1)}(p_e, e')$ を求める (力 F は式 (4.1) によって定める. 図 4.6 のような場合でもペナルティを課することはしない).

e' が e と端点を共有する場合, $F^{(1)}(p_e, e')$ が, i, j のうち e' の端点ではない方の頂点にかかるものとする. 一方, e' が e と端点を共有しない場合には, e のラベルが e と接する点 (図 4.11 の q') に近い方の端点にかかるものとする ($F^{(1)}(p_e, e')$ は実際には形状反映処理の実行後のものを用いるが, 簡単のため図 4.11 では省略している). $F^{(1)}(p_e, e')$ が端点 i に

かかるとき, e に垂直な方向の成分を求め, $F^{(4)}(i, e')$ とする. $F^{(1)}(p_e, e')$ が j にかかるときには, $F^{(4)}(i, e')$ は大きさが 0 の力であるものとする.

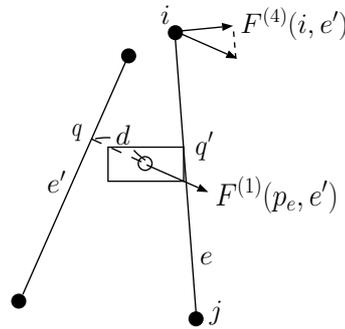


図 4.11: 力 $F^{(4)}(i, e')$

(b) 周囲の辺の仮頂点から働く力

ここでは, 辺 $e = (i, j)$ が周囲の辺 e' の仮頂点 $p_{e'}$ から受ける力を求め, それに基づいて頂点 i に働く力 $F^{(5)}(i, p_{e'})$ を定義する. この力は, e' のラベルを $p_{e'}$ の位置に置き, 図 4.3 のようにそのラベルを拡大したとき, その範囲 $H_{p_{e'}}$ と e が重なるときにのみ働くものとする.

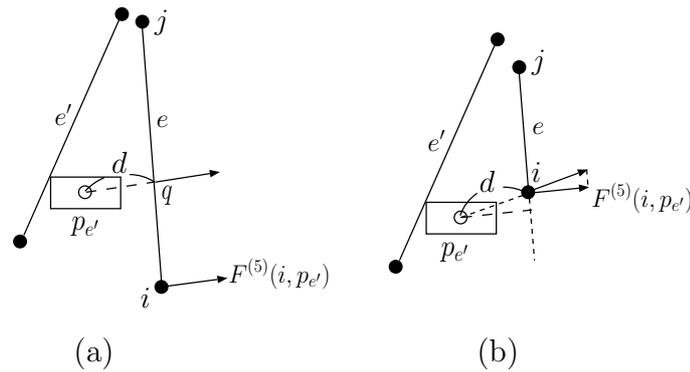
まず, 4.2.1.2(b) で述べた $F^{(1)}(p_e, e')$ と同様にして, $p_{e'}$ と e の間の力を求める. ただし, 式 (4.1) における d は, $H_{p_{e'}}$ の内部 (周上を含む) にある e 上の点のうち $p_{e'}$ に最も近いもの q と $p_{e'}$ との距離とする. また, 形状反映処理を行う際には, 式 (4.1) で求めた力の x 成分, y 成分に, それぞれ, e' のラベルの幅の C_8 倍, 高さの C_8 倍を掛ける. この力の向きは, 図 4.12 のように, $p_{e'}$ から q へ向かう方向であるものとする (ここでも, 図 4.11 と同様, 形状反映処理は省略している).

e と e' が端点を共有する場合, この力は, i, j のうち e' の端点ではない方の頂点にかかるものとする. 一方, e と e' が端点を共有しない場合には, i, j のうち q に近いほうの頂点にかかるものとする. そして, i にかかる場合は, その力の, e と垂直な方向の成分を求め, $F^{(5)}(i, p_{e'})$ とする. j にかかる場合には, $F^{(5)}(i, p_{e'})$ は大きさが 0 の力であるものとする.

(c) 頂点の移動

各頂点 i について, ラベルの配置場所の確保を目的とする移動は次の手順で行う.

- (i) i に接続する各辺 e について, その仮頂点 p_e が周囲の各辺 e' から受ける力 $F^{(1)}(p_e, e')$ を求める. さらに, それらの力から $\sum_{e'} F^{(4)}(i, e')$ (ベクトルとして足し合わせる) を計算し, これを $F_{i,e}$ とおく.
- (ii) i に接続する全ての辺 e についての $F_{i,e}$ の和を計算し, F_i とおく.
- (iii) i に接続する各辺 e について, それが周囲の各辺 e' の仮頂点 $p_{e'}$ から受ける力を求める. 更に, それらの力から $\sum_{e'} F^{(5)}(i, p_{e'})$ を計算し, これを $F'_{i,e}$ とおく.

図 4.12: 力 $F^{(5)}(i, p_{e'})$

(iv) i に接続する全ての辺 e についての $F'_{i,e}$ の和を計算し, それを F_i に加える.

(v) 頂点 i を, 力 F_i の向きにその大きさの C_4 倍だけ移動させる.

上の (v) で i を移動させる際には, i に接続する各辺 e の仮頂点 p_e も同時に移動させるものとする. $e = (i, j)$ であるものとする. また, p_e の位置に e のラベルを置いたとき, それと接する点を q' とする. このとき, p_e の移動は, $(i, q'$ 間の距離) と $(j, q'$ 間の距離) の比が, i の移動の前後で変わらないように行うものとする.

4.2.1.5 描画範囲の大きさによるパラメータの計算

本研究では, 描画領域を1辺の長さが D の正方形領域内に固定している. そのため, 4.2.2 で述べる計算機実験では, 得られた描画がこの領域内にちょうど納まるように縮小若しくは拡大した後, ラベル配置を行うこととしている. しかし, 提案手法4の実行途中でも, 描画が1辺 D の正方形領域に入りきらないこともあれば, もっと小さな正方形領域に入り得ることもある(実行途中では辺は描いていないが, 頂点の座標から描画範囲は容易に分かる). このような状況下では, 図4.3のような範囲 H_{p_e} をそのまま求めても, 各時点の描画に対しては大きすぎたり小さすぎたりすることになってしまう. また, 提案手法4では仮頂点と辺や端点との距離を何度も計算するが, それらも最終的に描画が拡大若しくは縮小されることを考えると, そのままの値を用いることがよいとはいえない. そこで各繰り返しにおいて, その時点の描画がどれぐらいの大きさの正方形領域に入るかを求め, その値を, H_{p_e} を求めるときに用いるラベルのサイズ, また各々の処理で用いる距離 d に反映させることにする.

具体的には, 4.2.1.1 で述べた概略の (5-b) において, その時点の描画が1辺 K の正方形領域に入るものとしたとき, K/D なる値を求めて r とする. そして, (5-c) では, 描画中の各辺 e のラベルの幅 wd_e , ラベルの高さ ht_e , 及び各々の処理で用いている距離 d を以下のようにする. なお, もともとの(描画領域の1辺を D としたときの)ラベルの幅及び高さを

それぞれ wd'_e, ht'_e とし, d' を現在の描画上での距離とする.

$$wd_e = wd'_e \times r \quad (4.4)$$

$$ht_e = ht'_e \times r \quad (4.5)$$

$$d = d'/r \quad (4.6)$$

4.2.1.6 近接回避処理

グラフ描画において, 頂点と辺の近接を避けることは, 頂点間の隣接関係を明確にするために重要である. 提案手法 4 では, 川西らの描画法と同様, 頂点と辺の間に力 $f''(i, k)$ を働かせている. しかし, ラベルの配置場所を確保する力も働かせているため, これだけでは十分に近接を避けられないことが多い. そこで, 描画を求める最後の段階において, 文献 [25] に述べられている簡単な近接回避処理を実行することにする. これは, 以下の (i), (ii) を 5 回繰り返すというものである.

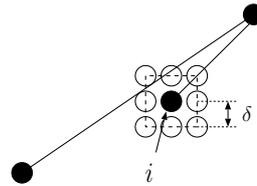


図 4.13: 頂点 i のまわりの 8 個の点

- (i) 近接を起こしている各頂点 i に対して, そのまわりに, 図 4.13 のように, 8 個の点を作る.
- (ii) 現在の i の位置と 8 個の点のうち, 近接の個数が最も少なくなる位置へ i を移動させる.

なお, ここでは, 図 4.13 中の δ は隣接頂点間の理想距離の $1/10$ とする. また, ある頂点と辺との間の距離が, 描画中の最も短い辺の長さの $1/5$ 以下ならば, 近接を起こしていると判断することにする.

4.2.1.7 提案手法 4 の記述

提案手法 4 の記述を以下に示す.

- (1) 全ての 2 頂点 i, j について, 理想距離 $l_{i,j}$ を求める.

- (2) 各頂点 i と、それに接続していない各辺 e について、 e の中点 k と i との間の理想距離 $\ell_{i,k}$ を求める。
- (3) 頂点を半径 $D/2$ の円の周上に等間隔に初期配置する。
- (4) $I = 1, 2, \dots, M_1 - 1$ について、以下の (4-a) を実行する。
- (4-a) 各頂点 i に対して、以下を実行する。
- (4-a-1) i 以外の各頂点 j に対して $f_{i,j}$ あるいは $f'_{i,j}$ を計算し、それらの合力を求めて F_i とする。
- (4-a-2) F_i の向きに、その大きさの C_4 倍だけ i を移動させる。
- (5) $I = M_1, M_1 + 1, \dots, M$ について、以下の (5-a) ~ (5-c) を実行する。
- (5-a) 各頂点 i に対して、以下を実行する。
- (5-a-1) i 以外の各頂点 j に対して $f_{i,j}$ あるいは $f'_{i,j}$ を計算し、それらの合力を求めて F_i とする。ただし、 $f_{i,j}$ の計算で用いる定数 C'_1 (C_1 は提案手法4と川西らの描画法で共通の定数であるが、後述するように両方で用いられている値が異なるため、このように表記している) は、別のより小さな定数 C''_1 で置き換えることとする。
- (5-a-2) 各辺の中点から i に働く力 $f''_{i,k}$ を計算し、その合力を F_i に加える。
- (5-a-3) F_i の向きに、その大きさの C_4 倍だけ i を移動させる。
- (5-b) 4.2.1.5 で述べた値 r を求める。
- (5-c) 各頂点 i に対して、以下を実行する。
- (5-c-1) i に接続する各辺 e について、以下を実行する。
- (5-c-1-1) $I = M_1$ を満たすか、あるいは I がある小さな自然数 C_5 の倍数であるならば、4.2.1.2(a), (b) で述べた処理によって e の仮頂点の位置を決定する。もし e の仮頂点削除されており、4.2.1.2(c) で述べた条件が成立するならば、仮頂点の復活を行う。
- (5-c-1-2) $I = M - 50$ の場合、仮頂点の削除をする条件が成立しているかどうか判定し、成立しているならば e の仮頂点を除去する。
- (5-c-1-3) e の仮頂点 p_e が削除されていなければ、周囲の各辺 e' から p_e に働く力 $F^{(2)}(p_e, e')$ 、及び、 e の端点から p_e に働く力を求める。これらのうち、 i からもう一方の (e の) 端点に向かう方向の最大の力と、逆方向の最大の力を選び、それらの合力を F_e とする。仮頂点 p_e を F_e の向きに、その大きさの C_4 倍だけ移動させる。

(5-c-1-4) e の仮頂点 p_e が削除されていないならば，周囲の各辺 e' から p_e に働く力より $F^{(4)}(i, e')$ を求め，それらの合力を $F_{i,e}$ とする．

(5-c-1-5) 周囲の各辺 e' の仮頂点 $p_{e'}$ から e に働く力より $F^{(5)}(i, p_{e'})$ を求め，それらの合力を $F_{i,e}$ に加える．

(5-c-2) i に接続する全ての辺に対する $F_{i,e}$ の合力を求め， F_i とする．頂点 i を， F_i の向きに，その大きさの C_4 倍だけ移動させる．

(6) 4.2.1.6 で述べた近接回避処理を実行する．

(7) 各辺を直線で描く．

提案手法 4 で用いている定数の値は，頂点数 20，辺の数 50 のグラフ 1000 個を用いて，

(i) 提案手法 4 により描画を求め，

(ii) 得られた描画に対し，3.2 で述べた提案手法 1 を用いてラベル配置を行う

という予備実験を行い，様々な値の組み合わせを試した結果をもとに定めたものである．なお，描画領域の 1 辺の長さ D は 10，各ラベルの高さは 0.30 とし，ラベルの幅は高さの 2~5 倍の範囲で辺ごとに定めた（本研究では，川西らの描画法，提案手法 4 とともに，隣接頂点間の理想距離，すなわち，辺の長さの理想距離を 1 としている． D の値が小さすぎると，各ラベルのサイズを小さくするだけでなく，この理想距離の値もより小さくする必要があるが，本研究で扱っている程度の頂点数，辺数のグラフであれば， $D = 10$ としていて特に問題はない．）

この予備実験では，まず以下の (a)~(c) のようなことを考慮して，各定数について値の範囲を広めに設定した．そして，各定数の値を設定範囲内のランダムな値とするような（定数値の）組合せを 500 通り定めて，それぞれについて提案手法 4 を実行した．その後，得られた結果（描画の質，ラベル配置率の両方を考慮）に基づいて，より適切と思われる範囲を各定数ごとに設定しなおし，また同様の実験を繰り返す，という処理を 5 回繰り返した．このようにして，各定数の値の範囲を絞りこんでいき，最後に適切と考えられる値を選択した．

(a) 川西らの描画法でも用いられていた定数のうち， C_4 については，川西らの描画法での値 0.1 と同様の値を用いる．一方， $C_1 \sim C_3$ については，川西らの描画法と同じ値（ $C_1 = 2.0$ ， $C_2 = 1.0$ ， $C_3 = 0.05$ ）を用いると，力 $f_{i,j}$ ， $f'_{i,j}$ 及び $f''_{i,k}$ がラベルの配置場所の確保を妨げるように働くことが多いため，川西らの描画法より小さな値 $C'_1 \sim C'_3$ を用いる．特に，力 $f_{i,j}$ は影響が大きいため，提案手法 4 の後半で用いる C_1 の値（ C''_1 とする）は，前半で用いる値 C'_1 よりさらに小さくする．

(b) 定数 C_5 は，仮頂点の位置を適切に定めるために，小さな自然数とする．

- (c) 提案手法4では、仮頂点の候補の決定(4.2.1.2(b))及び頂点の移動(4.2.1.4)のために、 C_6/C_7^d という大きさの力を計算している。一方、仮頂点の位置の調整(4.2.1.3)のためには、 C_6, C_7 の代わりに別の定数 C_9, C_{10} を用いた力を計算している。どちらの場合でも、注目している仮頂点と近くにある辺や端点との間の力を計算するのであるが、仮頂点の位置の微妙な調整を行うためには、あまり大きな力を働かせないようにし、また同時に、仮頂点のすぐ近くでない辺及び端点による影響を小さく抑えるのがよいと考えられる。そのため、 C_9 は C_6 より小さな値とし、 C_{10} は C_7 より大きな値にする。

最終的に定めた各定数の値を以下に示す。

$$C'_1 = 1.5, C''_1 = 0.27, C'_2 = 0.3, C'_3 = 0.01, C_4 = 0.1,$$

$$C_5 = 5, C_6 = 3.6, C_7 = 17, C_8 = 1.0, C_9 = 3.3, C_{10} = 85,$$

$$pena = 3.24, th_1 = \sqrt{3.4}, th_2 = \sqrt{0.3}, M = 250, M_1 = 100.$$

なお、繰り返し回数 M に関しては、図4.14に示すように、値を大きくするに従ってラベル配置率も増加する傾向が見られた。しかし M を大きくすると当然実行時間も長くなってしまいうため、配置率と実行時間の兼ね合いから $M = 250$ とした。

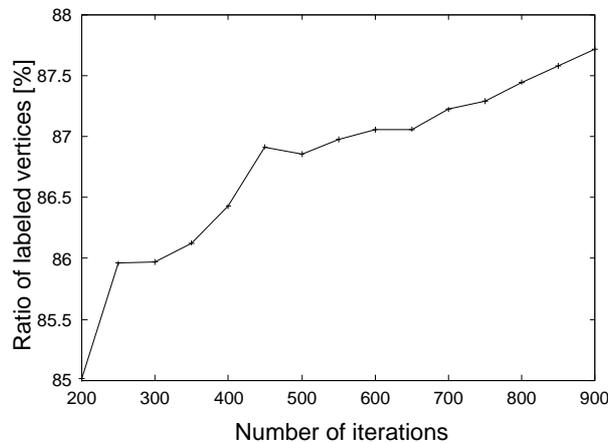


図 4.14: 繰り返し回数 M による配置率の変化

なお、川西らの描画法における各定数の値は、文献[5]と同様、 $C_1 = 2.0, C_2 = 1.0, C_3 = 0.05, C_4 = 0.1, M = 200$ とした。

4.2.2 計算機実験

提案手法4の有効性を確認するため、計算機実験により川西らの描画法との比較を行った。本節では、その実験結果を示す。

4.2.2.1 実験方法

頂点数が 20, 辺の数が 30, 40 あるいは 50 のグラフを乱数を用いて 1000 個ずつ作成し, 川西らの描画法と提案手法 4 で描画を求めた. 全ての場合について, 描画は 1 辺が $D = 10$ の正方形領域内で行った (川西らの描画法及び提案手法 4 で得られた描画を, 1 辺が 10 の正方形領域に納まるように縮小若しくは拡大した). 各ラベルの高さとしては 0.25, 0.30 の 2 通りを試した. また, ラベルの幅は高さの 2~5 倍の範囲で辺ごとに定めた.

以上のようにして得られた描画に対し, まず描画の望ましさを比較するため, 以下の 4 種類の値を求めた. これらはいずれも, 値が小さいほうが望ましいとされている.

- 辺の交差数
- 辺の長さの分散: 辺の長さの一様性に関して描画を評価するためのもの.
- 辺の長さの総和
- 頂点と辺の近接数: ある頂点とその頂点に接続しない辺との間の距離が, 0.15, 0.20, あるいは 0.25 以下になっている個数を調べた. 0.15 以下の近接が特に避けられるべきものである.

さらに,

- 実行時間 (描画を求めるのに要した時間)

及び

- 得られた描画に対して, 3.2 で述べた提案手法 1 を実行したときのラベル配置率

に関して川西らの描画法と提案手法 4 の比較を行った.

使用計算機の CPU は Athlon XP 1700+, 使用計算機言語は C 言語である.

4.2.2.2 実験結果

まず, 描画の望ましさに関する 4 種類の評価値についての比較結果を表 4.1 に示す. 表中の各値は 1000 個の描画に対する平均値である. この表の結果はラベルの高さが 0.25 のときのものであるが, 高さが 0.30 のときも同様の傾向の結果が得られた.

提案手法 4 では, 川西らの描画法に比べて, 頂点と辺の近接数以外の値がやや大きくなっている. 提案手法 4 では, ラベル配置場所を確保するための力を導入しているため, 描画が若干拡大し, 辺長分散及び辺長総和の値が悪化したのだと考えられる. しかし, 非隣接頂点間の理想距離に基づいた力も用いているため, これらの評価値が, Eades [2] の描画法のよう

に極端に悪くなる例は見られなかった．一方，頂点と辺の近接数については，川西らの描画法と比べ，ほとんどの場合で改善された．特に0.15以下の近接の個数は大きく改善されており，4.2.1.6で述べた近接回避法の有効性が見てとれる．

表 4.1: 描画の望ましさに関する評価値の比較

辺の数	評価基準	川西らの描画法	提案手法 4
30	辺交差数	6.43	7.45
	辺長分散	0.28	0.87
	辺長総和	67.47	72.48
	近接数 (0.15)	0.061	0.030
	" (0.20)	0.109	0.073
	" (0.25)	0.157	0.196
40	辺交差数	23.63	27.51
	辺長分散	0.69	1.79
	辺長総和	113.24	123.97
	近接数 (0.15)	0.478	0.101
	" (0.20)	0.674	0.300
	" (0.25)	0.907	0.769
50	辺交差数	54.49	64.13
	辺長分散	1.09	2.76
	辺長総和	160.23	180.00
	近接数 (0.15)	1.226	0.346
	" (0.20)	1.688	0.918
	" (0.25)	2.238	2.050

次に，川西らの描画法と提案手法 4 の実行時間を表 4.2 に示す．

これらの各値は，1つの描画を求めるのに要した時間の平均値である（ラベル配置に要した時間は含んでいない）．提案手法 4 では，辺数の増加に伴い，実行時間も大きくなっている．これは，提案手法 4 で用いたいくつかの処理の実行時間が辺数に依存しているためである（例えば 4.2.1.3 で述べた仮頂点の位置の調整の際，1つの仮頂点に対し，その仮頂点が周囲の辺から受ける力を求めている．4.2.1.2 で述べた仮頂点の位置の決定や，4.2.1.4 で述べた頂点の移動のための力を求める処理も同様である）．また，提案手法 4 の場合，ラベルが大きくなれば実行時間が増加しているが，これは，範囲 H_{pe} が拡大して，仮頂点あるいは辺にかかる力を計算する際に，より多くの辺を考慮しなければならなくなるためであると考えられる．

最後に，川西らの描画法及び提案手法 4 による描画に対して，提案手法 1 を用いてラベル

表 4.2: 実行時間の比較

辺の数	ラベルの高さ	川西らの描画法	提案手法 4
30	0.25	0.041 sec	0.363 sec
	0.30		0.382 sec
40	0.25	0.040 sec	0.588 sec
	0.30		0.626 sec
50	0.25	0.043 sec	0.893 sec
	0.30		0.961 sec

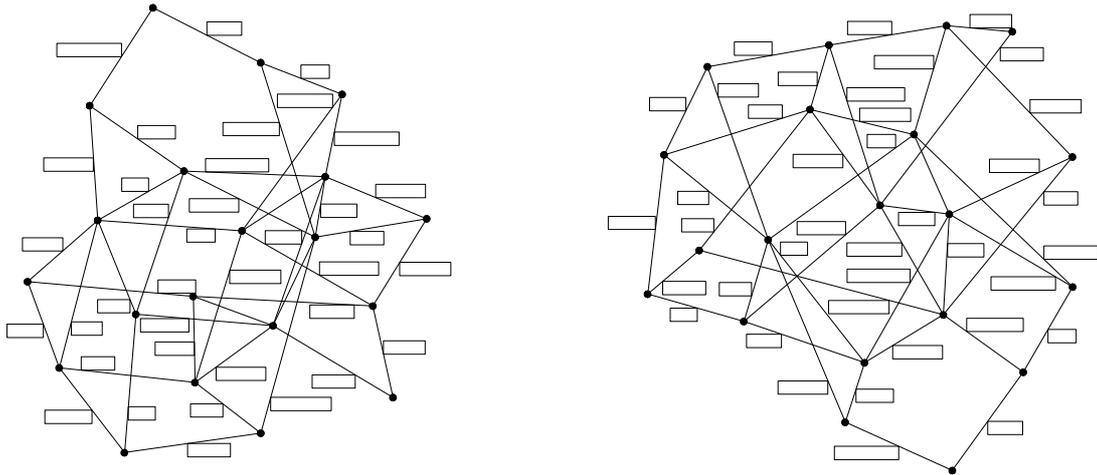
配置を行ったときの配置率を表 4.3 に示す．ここでも，各数値は 1000 個の描画に対する平均値である．全ての場合において改善が見られるが，特に辺の数が多いほど，またラベルのサイズが大きいほど，大きな改善が見られた．図 4.15 に，2 つの描画法による描画にラベル配置を行った例を示す．

表 4.3: ラベル配置率の比較

辺の数	ラベルの高さ	川西らの描画法	提案手法 4
30	0.25	98.18 %	99.55 %
	0.30		98.69 %
40	0.25	94.39 %	97.66 %
	0.30		94.43 %
50	0.25	86.64 %	93.22 %
	0.30		85.96 %

4.3 頂点ラベルの存在を考慮したグラフ描画アルゴリズム

本節では，頂点ラベルの存在を考慮したグラフ描画アルゴリズムについて説明する．以下，この方法を提案手法 5 と呼ぶことにする．



(a) 川西らの描画法

(配置できたラベルの数 : 37)

(b) 提案手法4

(配置できたラベルの数 : 39)

図 4.15: ラベル付き描画の例 (辺の本数 : 40, ラベルの高さ : 0.30)

4.3.1 アルゴリズムの説明

4.3.1.1 アルゴリズムの概略

4.2 で述べた提案手法4と同様, 提案手法5の前半 (繰り返し回数 I がある定数 M_1 未満のとき) では, 2.2.3 で述べた川西らの描画法における力 $f_{i,j}$ 及び $f'_{i,j}$ を頂点の移動に用いる. 後半 ($I \geq M_1$ のとき) も提案手法4と同様, これらの力に $f''_{i,k}$ を加え, さらに各頂点のラベルの想定位置を表す仮頂点 (ラベルの中央に対応) に働く力を考える. 仮頂点と, その近くを通る辺などとの間に斥力を働かせる点も同様であるが, 仮頂点の作成方法や仮頂点の位置の調整方法などが異なっている.

描画領域は, 1 辺の長さが D の正方形領域に固定されているものとする. 提案手法5の概略は以下の通りである.

- (1) 全ての2頂点 i, j について, 理想距離 $l_{i,j}$ を求める.
- (2) 各頂点 i と, それに接続していない各辺 e とについて, e の中点 k と i との間の理想距離 $l'_{i,k}$ を求める.
- (3) 頂点を半径 $D/2$ の円の周上に等間隔に初期配置する.
- (4) $I = 1, 2, \dots, M_1 - 1$ について, 以下の (4-a) を実行する.
 - (4-a) 各頂点 i に対して, 以下を実行する.
 - (4-a-1) i 以外の各頂点 j に対して $f_{i,j}$ あるいは $f'_{i,j}$ を計算し, それらの合力を求めて F_i とする.

- (4-a-2) F_i の向きに, その大きさの C_4 倍だけ i を移動させる .
- (5) $I = M_1, M_1 + 1, \dots, M$ について, 以下の (5-a) ~ (5-c) を実行する .
- (5-a) 各頂点 i に対して, 以下を実行する .
- (5-a-1) i 以外の各頂点 j に対して $f_{i,j}$ あるいは $f'_{i,j}$ を計算し, それらの合力を求めて F_i とする .
- (5-a-2) 各辺の中点から i に働く力 $f''_{i,k}$ を計算し, その合力を F_i に加える .
- (5-a-3) F_i の向きに, その大きさの C_4 倍だけ i を移動させる . i の仮頂点が作成されているなら, 仮頂点も i と同じ方向に同じ距離だけ移動させる .
- (5-b) 描画範囲の大きさを調べ, その結果に基づいてあるパラメータ r の値を計算する .
- (5-c) 各頂点 i に対して, 以下を実行する .
- (5-c-1) $I = M_1$ であるか, I がある小さな自然数 C_5 (定数) の倍数であるならば, i の仮頂点の位置を決定する .
- (5-c-2) i の仮頂点の位置を調整する .
- (5-c-3) 付近にある辺から i の仮頂点に働く力を求める .
- (5-c-4) 付近にある別の頂点の仮頂点から i の仮頂点に働く力を求める .
- (5-c-5) i に接続する各辺 e に対し, 付近にある仮頂点から働く力を求める .
- (5-c-6) (5-c-3) ~ (5-c-5) で求めた力に基づいて, i に働く力 F'_i を求める .
- (5-c-7) F'_i の向きに, その大きさの C_4 倍だけ i を移動させる . i の仮頂点が存在するなら, 仮頂点も i と同じ方向に同じ距離だけ移動させる .
- (6) 頂点と辺の近接を緩和する処理と, 後に述べる後処理を行う .
- (7) 各辺を直線で描く .

以下では, まず 4.3.1.2 において, 各頂点の仮頂点の位置の決定方法 (上記の手順の (5-c-1)) について述べる . 4.3.1.3 では, 仮頂点をより望ましい位置に移動させる方法 ((5-c-2)) について説明する . 4.3.1.4 では, ラベルの配置位置を確保するための力 ((5-c-3) ~ (5-c-6)) を定義し, (5-c-7) における頂点の移動方法について述べる . 4.3.1.5 では, 描画範囲 (実際に描画が占める領域) の大きさに基づいて行うあるパラメータの計算 ((5-b)) について述べる . 4.3.1.6 では, (6) の描画における頂点と辺の近接を減らすための処理と後処理について説明する . 最後に 4.3.1.7 において, 提案手法 5 全体のより詳しい記述を示す .

4.3.1.2 仮頂点の位置の決定

(a) ラベル空間による仮頂点の位置の決定

提案手法5の(5-c-1)では、各頂点に対して仮頂点の位置の決定(あるいは再決定)を行う。

3.3.1.1で述べたラベル空間が作成可能な頂点に対しては、最も広いラベル空間の中央に仮頂点を作る。一方、ラベル空間が作成できない頂点については、仮頂点の位置の候補をいくつか定め、それらのうち最も望ましいと思われるものを選択することによって、仮頂点の位置を決定する。これらの処理の詳細について次の(b)で述べる。

(b) ラベル空間が作成できない場合

頂点*i*に対して、ラベル空間が作成できない場合、仮頂点の位置の候補を以下のようにして作る。

まず、*i*に対して既に仮頂点が作られている場合には、現在の仮頂点の位置を候補の1つ cp_1 とする。仮頂点が作られていない場合には、候補の位置にラベルを配置した場合、ラベルの中央が*i*の真上に来るように候補 cp_1 を作成する。次に、*i*が1本の辺のみに接続している場合には、候補の位置にラベルを配置した場合、その辺と反対の位置に、ラベルの角(*i*と最も近くなるもの)がくるように候補 cp_2 を作成する。図4.16(a)参照。一方、*i*が2本以上の辺と接続している場合には、*i*に接続する隣り合う2辺の間に1つずつ仮頂点の位置の候補 cp_k を作成する。図4.16(b)参照。

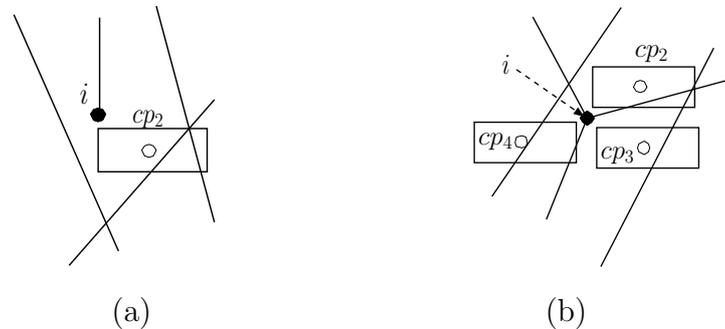


図 4.16: 仮頂点の位置の候補

このようにして作成された各候補 cp_k に対し、それらの候補が周囲の各辺 e' から受ける力 $F^{(1)}(cp_k, e')$ を求め、その大きさの和 $\sum_{e'} |F^{(1)}(cp_k, e')|$ が最も小さい候補の位置を仮頂点の位置と定める。ただし、大きく離れた辺からの力は意味がないので、提案手法4と同様、ラベルを上下左右にラベルの高さだけ拡大した範囲 H_{cp_k} (4.2.1.2(b)参照)を考え、それと重なる辺のみから力を受けるものとする。

仮頂点あるいは候補 cp_k が辺 e' から受ける力 $F^{(1)}(cp_k, e')$ は、提案手法4で用いたものと同様である。まず次のような力 F を求める。

$$F = C_6 / C_7^d \quad (4.7)$$

C_6, C_7 は正の定数であり, $C_7 > 1$ である. d は, H_{cp_k} の内部 (周上を含む) にある e' 上の点のうち cp_k に最も近いもの q と cp_k との距離である. この力 F に対し, 4.2.1.2(b) で述べた形状反映処理を施したものを $F^{(1)}(cp_k, e')$ とする. 図 4.17 に例を示す. 4.2.1.2(b) では,

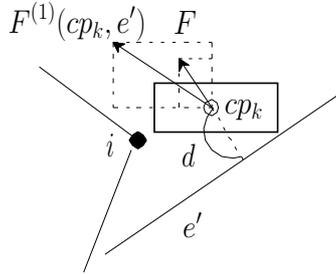


図 4.17: 力 $F^{(1)}(cp_k, e')$

例外を 1 つ設け, 条件を満たす場合には $F^{(1)}(cp_k, e')$ をやや大きな値 $pena$ に置き換えていたが, 提案手法 5 ではこのような例外は設けていない.

範囲 H_{cp_k} に重なる各辺 e' に対して $F^{(1)}(cp_k, e')$ を計算した後, 前述のように, それらの大きさの和 $\sum_{e'} |F^{(1)}(cp_k, e')|$ を求め, この値の最も小さくなる候補の位置を仮頂点の位置と定める.

(c) 仮頂点の削除と復活

提案手法 5 では仮頂点の削除の処理を, アルゴリズムの後半で 1 回だけ行う. この処理は, 4.2.1.2(c) で述べたものと同様, 頂点 i に対してラベル空間が作成できず, 更に i の仮頂点のどの候補 cp_k についても $\sum_{e'} |F^{(1)}(cp_k, e')|$ があるしきい値 th_1 よりも大きいとき, i の仮頂点を削除するというものである.

また, 仮頂点が削除されている頂点 i について, 後に, ラベル空間が作成できた, 若しくは (b) で述べたようにして仮頂点の候補を作ったとき $\sum_{e'} |F^{(1)}(cp_k, e')|$ の最小値があるしきい値 th_2 よりも小さくなったとする. このような場合に i の仮頂点を復活させる点も, 4.2.1.2(c) と同様である.

4.3.1.3 仮頂点の位置の調整

頂点 i の仮頂点を p_i と表す. 本節では, p_i をより望ましい位置に移動させるための 2 種類の力と, それらに基づいた移動方法について述べる.

(a) 周囲の辺から仮頂点に働く力

仮頂点が他の辺に近づくことを防ぐため, 周囲の辺 e' から仮頂点に働く力 $F^{(2)}(p_i, e')$ を定義する.

まず, 4.3.1.2(b) で述べた方法により, 仮頂点 p_i が周囲の各辺 e' から受ける力 $F^{(1)}(p_i, e')$

を求める．ただし，力 F を計算する際には，式 (4.7) の C_6, C_7 の代わりに，それぞれ別の定数 C_9, C_{10} を用いる (d は前述のように定める) ．

図 4.18 に示すように， i のラベルを i の周りを一周させたときに p_i が通過する軌跡を L_i と定める．そして，同図に示すように，各 $F^{(1)}(p_i, e')$ に対して L_i との接線成分を求め，これを $F^{(2)}(p_i, e')$ とする ．

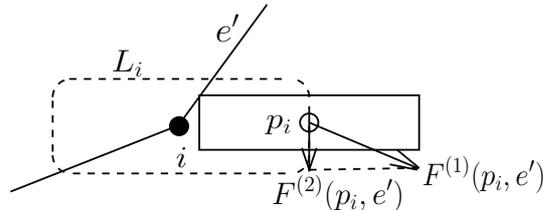


図 4.18: 力 $F^{(2)}(p_i, e')$

(b) 周囲の仮頂点から仮頂点に働く力

ある頂点 i の仮頂点 p_i が，別の頂点 j の仮頂点 p_j に近づきすぎるのはあまり望ましいことではない． p_i と p_j の位置にそれぞれのラベルを配置すると両者が重なり合ってしまう，いずれか一方にしかラベルを配置できなくなってしまうためである．そこで， p_j から p_i に斥力を働かせることにする ．

まず， p_j の位置に j のラベルを配置したとき，そのラベルが H_{p_i} と重なるような各 p_j に対し，以下のような力 F' を求める ．

$$F' = C_{11}/C_{10}^d \quad (4.8)$$

ここで， C_{11} は定数であり， d は p_i と p_j との距離である．また，力の方向は p_j から p_i に向かう方向であるものとする． F' に形状反映処理を施したものを $F^{(3)}(p_i, p_j)$ とする．図 4.19 参照．さらに，(a) と同様， $F^{(3)}(p_i, p_j)$ の L_i との接線成分を $F^{(4)}(p_i, p_j)$ と表すことにする ．

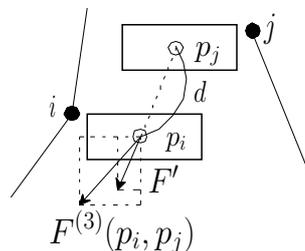
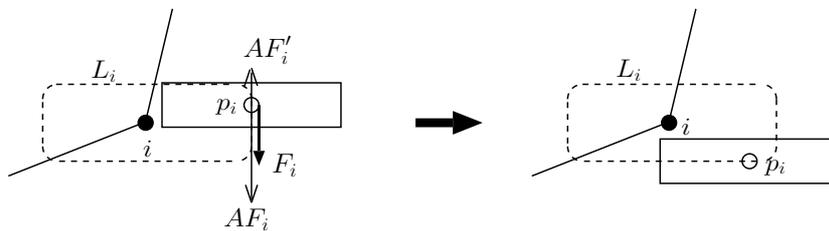


図 4.19: 力 F' と $F^{(3)}(p_i, p_j)$

(c) 仮頂点の移動

各頂点 i に対し，仮頂点 p_i のより望ましい位置への移動は次の手順で行う ．

- (i) p_i が周囲の各辺 e' から受ける力 $F^{(1)}(p_i, e')$ を求め、さらにそれから $F^{(2)}(p_i, e')$ を計算する。
- (ii) p_i が周囲の仮頂点 p_j から受ける力 $F^{(3)}(p_i, p_j)$ を求め、さらにそれから $F^{(4)}(p_i, p_j)$ を計算する。
- (iii) 各 $F^{(2)}(p_i, e')$ 及び各 $F^{(4)}(p_i, p_j)$ のうち、時計回りの方向の最大の力を AF_i とし、逆方向の最大の力を AF'_i とする。このとき、 AF_i と AF'_i の合力を F_i とする。図 4.20 左参照。
- (iv) 仮頂点 p_i を、力 F_i の向きに、その大きさの C_4 倍だけ軌跡 L_i にそって移動させる。図 4.20 右参照。

図 4.20: p_i の位置の調整

4.3.1.4 頂点の移動のための力

4.3.1.1 で概略を示したように、提案手法 5 の後半 ($I \geq M_1$ のとき) では、変数 I の各値について、各頂点 i の移動を通常 2 回行う ((5-a-3) と (5-c-7))。1 回目の移動は、川西らの描画法におけるのと同様、頂点間の力及び頂点と辺の間の力によるものである。一方、2 回目の移動は、ラベルの配置場所の確保を目的とするもので、 i の仮頂点が周囲の辺から受ける力、 i の仮頂点が周囲の仮頂点から受ける力、及び i に接続する辺が周囲の仮頂点から受ける力による。本節では、これら 3 種類の力について説明した後、これらの力に従ってどのように i を移動させるかについて述べる。

(a) 周囲の辺から i の仮頂点に働く力

頂点 i の仮頂点 p_i と周囲の各辺 e' に対して、4.3.1.2(b) で定義した力 $F^{(1)}(p_i, e')$ を求める (力 F は式 (4.7) によって定める)。そして、この力 $F^{(1)}(p_i, e')$ を i がそのまま受けるものとする。

(b) 周囲の仮頂点から i の仮頂点に働く力

頂点 i の仮頂点 p_i と別の頂点 j の仮頂点 p_j (p_i の付近にあるもの) に対して、4.3.1.3(b) で定義した力 $F^{(3)}(p_i, p_j)$ を求める。そして、この力 $F^{(3)}(p_i, p_j)$ を i がそのまま受けるものとする。

(c) 周囲の仮頂点から i に接続する辺に働く力

4.2.1.4 と同様、辺 $e = (i, j)$ が周囲の頂点 v の仮頂点 p_v から受ける力を求め、それに基づいて頂点 i に働く力 $F^{(5)}(i, p_v)$ を定義する。この力は、 v のラベルを p_v の位置に置き、そのラベルを拡大したとき、その範囲 H_{p_v} と e が重なるときにのみ働くものとする。

まず、4.3.1.2(b) で述べた $F^{(1)}(p_i, e')$ と同様にして、 p_v と e の間の力を求める。ただし、式 (4.7) の C_6 の代わりに別の定数 C_{12} を用いる。また、 d は、 H_{p_v} の内部（周上を含む）にある e 上の点のうち p_v に最も近いもの q と p_v との距離とする。また、形状反映処理を行う際には、式 (4.7) で求めた力の x 成分、 y 成分に、それぞれ、 v のラベルの幅の C_8 倍、高さの C_8 倍を掛ける。

e の端点 i, j のいずれかが v と一致する場合、この力は、 v と一致しない方の頂点にかかるものとする。一方、 v が i, j いずれとも一致しない場合には、 i, j のうち q に近いほうの頂点にかかるものとする。そして、 i にかかる場合は、その力の、 e と垂直な方向の成分を求め、 $F^{(5)}(i, p_v)$ とする。 j にかかる場合には、 $F^{(5)}(i, p_v)$ は大きさが 0 の力であるものとする。

(d) 頂点の移動

各頂点 i について、ラベルの配置場所の確保を目的とする移動は次の手順で行う。

- (i) i の仮頂点 p_i が周囲の各辺 e' から受ける力 $F^{(1)}(p_i, e')$ を求める。そしてその合力 $\sum_{e'} F^{(1)}(p_i, e')$ (ベクトルとして足し合わせる) を計算し、これを F_i とおく。
- (ii) i の仮頂点 p_i が周囲の別の頂点 j の仮頂点 p_j から受ける力 $F^{(3)}(p_i, p_j)$ を求める。そしてその合力 $\sum_{p_j} F^{(3)}(p_i, p_j)$ (ベクトルとして足し合わせる) を計算し、これを F_i に加える。
- (iii) i に接続する各辺 e について、それが周囲の頂点 v の仮頂点 p_v から受ける力を求める。さらに、それらの力から $\sum_{p_v} F^{(5)}(i, p_v)$ を計算し、これを $F'_{i,e}$ とおく。
- (iv) i に接続する全ての辺 e についての $F'_{i,e}$ の和を計算し、それを F_i に加える。
- (v) 頂点 i を、力 F_i の向きにその大きさの C_4 倍だけ移動させる。

上の (v) で i を移動させる際には、 i の仮頂点 p_i も同じ方向へ同じ距離だけ移動させるものとする。

4.3.1.5 描画範囲の大きさによるパラメータの計算

提案手法 5 では、提案手法 4 と同様に、各繰り返しにおいて、その時点の描画がどれぐらいの大きさの正方形領域に入るかを求め、その値をラベルのサイズと、各々の処理で用いる距離 d に反映させている。

4.3.1.1 で述べた概略の (5-b) において，その時点の描画が 1 辺 K の正方形領域に入るものとしたとき， K/D なる値を求めて r とする．そして，(5-c) では描画中の各頂点 i のラベルの幅 wd_i ，ラベルの高さ ht_i ，及び各々の処理で用いている距離 d を以下のようにする．なお，もともとの（描画領域の 1 辺を D としたときの）ラベルの幅及び高さをそれぞれ wd'_i ， ht'_i とし， d' を現在の描画上での距離とする．

$$wd_i = wd'_i \times r \quad (4.9)$$

$$ht_i = ht'_i \times r \quad (4.10)$$

$$d = d' / r \quad (4.11)$$

4.3.1.6 後処理

提案手法 5 では，4.3.1.1 で述べた概略の (6) において，頂点と辺の近接を避けるため，簡単な近接回避処理を実行する．この処理は 4.2.1.6 で述べたものと全く同じものである．

この処理の後で提案手法 5 では，より多くのラベルを配置できるようにするための後処理を行う．図 4.21(a) において， i のラベルを p_i の位置に配置すると，そのラベルは辺と重なってしまう．しかし，この重なりは，同図 (b) のように i と p_i を少しだけ下に移動させることで回避できる．



図 4.21: 頂点 i の位置の改善

よって，提案手法 5 では，次のような条件を満たす場合， i を δ だけ下（あるいは上）に移動させることにしている．

1. p_i が i よりも下（上）にある．
2. p_i の位置に i のラベルを配置したとき以下の条件を満たす．
 - (a) ラベルは i 以外の頂点（を中心とする半径 ϵ の円）と重ならない．
 - (b) p_i より上側（下側）でラベルと重なる辺がある．
 - (c) p_i より下側（上側）でラベルと重なる辺はない．

上記の方法で全ての頂点に対して上下方向への移動を試みた後，同様にして左右への頂点の移動も試みることにしている．

4.3.1.7 提案手法5の記述

提案手法5の記述を以下に示す。

- (1) 全ての2頂点 i, j について、理想距離 $l_{i,j}$ を求める。
- (2) 各頂点 i と、それに接続していない各辺 e とについて、 e の中点 k と i との間の理想距離 $l'_{i,k}$ を求める。
- (3) 頂点を半径 $D/2$ の円の周上に等間隔に初期配置する。
- (4) $I = 1, 2, \dots, M_1 - 1$ について、以下の(4-a)を実行する。
 - (4-a) 各頂点 i に対して、以下を実行する。
 - (4-a-1) i 以外の各頂点 j に対して $f_{i,j}$ あるいは $f'_{i,j}$ を計算し、それらの合力を求めて F_i とする。
 - (4-a-2) F_i の向きに、その大きさの C_4 倍だけ i を移動させる。
- (5) $I = M_1, M_1 + 1, \dots, M$ について、以下の(5-a)~(5-c)を実行する。
 - (5-a) 各頂点 i に対して、以下を実行する。
 - (5-a-1) i 以外の各頂点 j に対して $f_{i,j}$ あるいは $f'_{i,j}$ を計算し、それらの合力を求めて F_i とする。ただし、 $f_{i,j}$ の計算で用いる定数 C'_1 (C_1 は提案手法5と川西らの描画法で共通の定数であるが、後述するように両方で用いられている値が異なるため、このように表記している) は、別のより小さな定数 C''_1 で置き換えることとする。
 - (5-a-2) 各辺の中点から i に働く力 $f''_{i,k}$ を計算し、その合力を F_i に加える。
 - (5-a-3) F_i の向きに、その大きさの C_4 倍だけ i を移動させる。 i の仮頂点が作成されているなら、仮頂点も i と同じ方向に同じ距離だけ移動させる。
 - (5-b) 4.3.1.5 で述べた値 r を求める。
 - (5-c) 各頂点 i に対して、以下を実行する。
 - (5-c-1) $I = M_1$ を満たすか、あるいは I がある小さな自然数 C_5 の倍数であるならば、4.3.1.2(a), (b) で述べた処理によって i の仮頂点の位置を決定する。もし i の仮頂点が削除されており、4.3.1.2(c) で述べた条件が成立するならば、仮頂点の復活を行う。
 - (5-c-2) $I = M - 50$ の場合、仮頂点の削除をする条件が成立しているかどうか判定し、成立しているならば i の仮頂点を除去する。

- (5-c-3) i の仮頂点 p_i が削除されていなければ, 周囲の各辺 e' から p_i に働く力 $F^{(2)}(p_i, e')$, 及び, 周囲にある別の頂点 j の仮頂点 p_j から受ける力 $F^{(4)}(p_i, p_j)$ を求める. これらのうち, 時計周りの最大の力と, 逆方向の最大の力を選び, それらの合力を F_i とする. そして, p_i を, F_i の向きに, その大きさの C_4 倍だけ 4.3.1.3(a) で述べた軌跡 L_i にそって移動させる.
- (5-c-4) i の仮頂点 p_i が削除されていなければ, 周囲の各辺 e' から p_i に働く力 $F^{(1)}(p_i, e')$ を求め, それらの合力を F_i とする.
- (5-c-5) i の仮頂点 p_i が削除されていなければ, 周囲の別の頂点 j の仮頂点 p_j から受ける力 $F^{(3)}(p_i, p_j)$ を求め, それらの合力を F_i に加える.
- (5-c-6) i に接続する各辺 e について, それが周囲の頂点 v の仮頂点 p_v から受ける力を求め, その力から $F^{(5)}(i, p_v)$ を計算する. それらの合力を $F'_{i,e}$ とする.
- (5-c-7) i に接続する全ての辺 e についての $F'_{i,e}$ の和を計算し, それを F_i に加える.
- (5-c-8) 頂点 i を, 力 F_i の向きにその大きさの C_4 倍だけ移動させる. i の仮頂点が存在するなら, 仮頂点も i と同じ方向に同じ距離だけ移動させる.

(6) 4.3.1.6 で述べた近接回避処理と後処理を実行する.

(7) 各辺を直線で描く.

提案手法 5 で用いている各定数の値は予備実験によって定めた. 予備実験の実験条件や各定数の値の定め方は, 4.2.1.7 で述べた提案手法 4 における予備実験とほぼ同様である.

最終的に定めた各定数の値は以下の通りである.

$$C'_1 = 2.4, C''_1 = 0.26, C'_2 = 0.6, C'_3 = 0.2, C_4 = 0.1,$$

$$C_5 = 5, C_6 = 2.9, C_7 = 9, C_8 = 1.0, C_9 = 3.3, C_{10} = 64, C_{11} = 1.5, C_{12} = 3.4,$$

$$th_1 = \sqrt{5.9}, th_2 = \sqrt{0.3}, \delta = 0.25, M = 250, M_1 = 100.$$

なお, 川西らの描画法における各定数の値は, 文献 [5] と同様, $C_1 = 2.0, C_2 = 1.0, C_3 = 0.05, C_4 = 0.1, M = 200$ とした.

4.3.2 計算機実験

提案手法 5 の有効性を確認するため, 計算機実験により川西らの描画法との比較を行った. 本節では, その実験結果を示す.

4.3.2.1 実験方法

頂点数，辺の数の様々な組み合わせについて，乱数を用いて1000個ずつグラフを作成し，川西らの描画法と提案手法5で描画を求めた．全ての場合について，描画は1辺が $D = 10$ の正方形領域内で行った（川西らの描画法及び提案手法5で得られた描画を，1辺が10の正方形領域に納まるように縮小若しくは拡大した）．各ラベルの高さとしては0.25, 0.30の2通りを試した．また，ラベルの幅は高さの2~5倍の範囲で辺ごとに定めた．

以上のようにして得られた描画に対し，まず描画の望ましさを比較するため，4.2での計算機実験と同様，以下の4種類の値を求めた．

- 辺の交差数
- 辺の長さの分散
- 辺の長さの総和
- 頂点と辺の近接数

さらに，

- 実行時間（描画を求めるのに要した時間）

及び

- 得られた描画に対して，3.3で述べた提案手法2を実行したときのラベル配置率

に関して川西らの描画法と提案手法5の比較を行った．

使用計算機のCPUはAthlon XP 1700+，使用計算機言語はC言語である．

4.3.2.2 実験結果

まず，描画の望ましさに関する4種類の評価値についての比較結果を表4.4に示す．表中の各値は，頂点数40，辺の数60のグラフ1000個の描画に対する平均値である．この表の結果はラベルの高さが0.30のときのものであるが，高さが0.25のときも同様の傾向の結果が得られた．

提案手法5では，ラベル配置場所を確保するための力が働くため，川西らの描画法に比べ，辺長分散や辺長総和の値が悪くなっている．

次に，表4.5に実行時間の比較を示す．この表もラベルの高さが0.30のときの結果を示している．

表 4.4: 描画の望ましさに関する評価値の比較

	川西らの描画法	提案手法 5
辺交差数	24.36	30.37
辺長分散	0.259	0.940
辺長総和	101.7	116.1
近接数 (0.20)	1.766	2.035

表 4.5: 実行時間

頂点数	辺の数	川西らの描画法	提案手法 5
30	50	0.108 sec	0.442 sec
40	60	0.191 sec	0.692 sec
50	70	0.301 sec	1.016 sec

これらの各値は、1つの描画を求めるのに要した時間の平均値である（ラベル配置に要した時間は含んでいない）。提案手法 5 では仮頂点を導入しているため、川西らの描画法よりも実行時間が大きくなっている。

表 4.6 に、川西らの描画法及び提案手法 5 による描画に対して、提案手法 2 を用いてラベル配置を行ったときの配置率を示す。この表より、提案手法 5 で得られる描画は、川西らの描画法で得られる描画よりも頂点ラベルの配置に適したものになっていることがわかる。

表 4.6: ラベル配置率

頂点数	辺の数	川西らの描画法	提案手法 5
30	50	83.28%	91.44%
40	60	82.31%	91.29%
50	70	80.37%	89.11%

最後に、図 4.22 に、頂点数 40、辺の数 60 のグラフに対し、2つの描画法による描画にラベル配置を行った例を示す（ラベルの高さは 0.30 である）。川西らの描画法で得られた描画に対しては 30 個の、提案手法 5 で得られた描画に対しては 37 個の頂点ラベルを配置することができている。

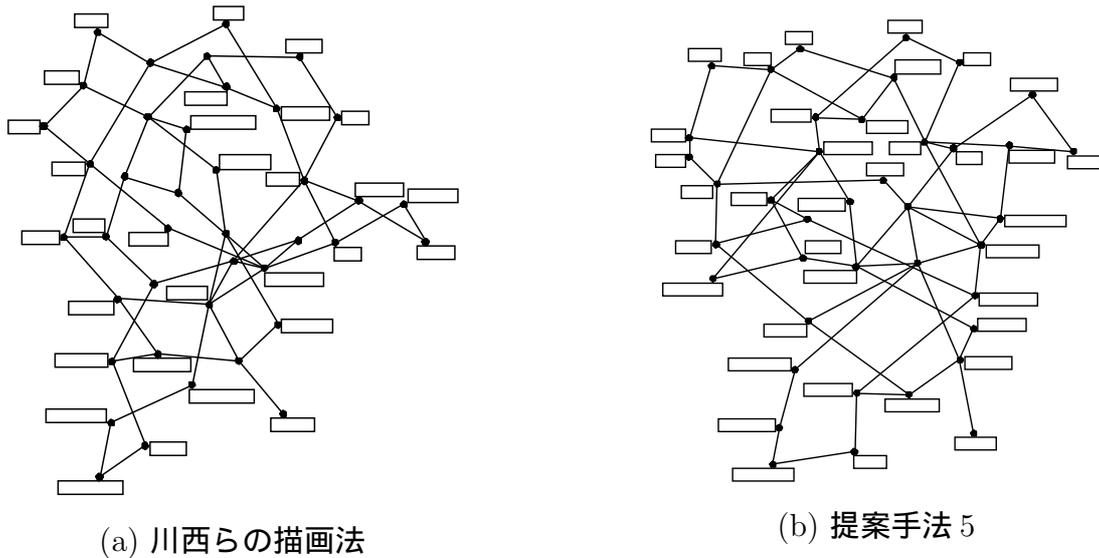


図 4.22: ラベル付き描画の例

4.4 結言

ラベルの情報を考慮せずにグラフ描画を求めると、グラフが密である場合やラベルのサイズが大きい場合には、ラベル配置率の上限値ですら十分に大きい値にはならないことがある。そこで、本章では、各頂点あるいは各辺のラベルのサイズを考慮しながら頂点や辺の位置を決定するというグラフ描画法を提案した。そして、これらのアルゴリズムによる描画に、第3章で提案したラベル配置アルゴリズムを適用することによって、ラベル配置率が大きく改善できることを計算機実験により示した。

本章では、辺のみあるいは頂点のみがラベルをもつグラフに対して描画アルゴリズムを設計したが、辺と頂点の両方がラベルをもつ場合について詳細に検討することは今後の課題である。

第5章 結論

本論文では、頂点や辺がラベルをもつグラフの描画に関する2つの問題について考察を行った。

本論文の第3章では、あらかじめ定められたグラフの描画中に、どのラベルも頂点や辺及び他のラベルと重ならないようにして、できるだけ多くのラベルを配置するという問題について考察した。この問題に対し、Kakoulis-Tollis [8] は、まず各頂点あるいは各辺のラベルを配置する位置の候補を複数個定め、次にそれらの中から最終的なラベル位置を選択するというアルゴリズムを提案していた。この方法は高いラベル配置率を達成できるものとされていたが、ラベル配置位置の候補を定める部分、及び各候補から最終的なラベルの配置位置を選択する部分の両方に改善の余地があった。そこで、本研究では、ラベル配置率の改善を主な目的として、このアルゴリズムの改良を行った。まず、辺のみあるいは頂点のみがラベルをもつ場合についてのアルゴリズムを提案し、それらの方法のラベル配置率が、Kakoulisらの方法より大幅に高い値になること、及び実験的に求めた（緩い）上限値に近い値になることを示した。次に、辺と頂点の両方がラベルをもつ場合に対するアルゴリズムを提案し、Kakoulisらの方法だけでなく、地図ラベル配置アルゴリズムとして開発されていた方法よりも高いラベル配置率を達成できることを示した。

無向グラフのラベル付き描画を得るための方法として、従来、まずラベルの情報を特に考慮せずにグラフの描画を求め、次に得られた描画に対してラベルを配置する、という分離型アプローチが主に検討されていた。しかし、このような方法では、特にグラフが密である場合やラベルのサイズが大きい場合には、配置できないラベルが多く残ることがあった。そこで、本研究では、第4章において、辺のみあるいは頂点のみがラベルをもつ場合のそれぞれについて、各ラベルのサイズを考慮しながら頂点や辺の位置を決定するというグラフ描画アルゴリズムを提案した。これらの方法は、各ラベルの想定位置を示すための仮頂点とそれに働く力を定義することによって、川西らによるグラフ描画アルゴリズムを拡張したものである。そして、これらのアルゴリズムによる描画と、第3章で提案したラベル配置アルゴリズムを用いることによって、ラベル配置率をさらに改善できることを計算機実験により示した。

第3章で提案したアルゴリズムでは、各辺あるいは頂点に対するラベル候補の作成方法を工夫しているが、この手法は、グラフ描画へのラベル配置だけではなく、地図へのラベル配置にも適用することができる [27]。今後、他の可能性も含めて、この手法の応用について検

討を続けていく予定である．

第4章で提案した2つの方法の実行時間を短縮すること，及びラベル配置率がさらに改善できるような描画を求め得るよう改良することは今後の課題である．辺と頂点の両方がラベルをもつグラフについては，これらの2つの方法を組み合わせた描画法がまず考えられるが，今後，より詳細に検討していく必要がある．

謝辞

本研究の機会を賜わり，また学部4年から6年間に渡り熱心に御指導頂きました神戸大学工学部電気電子工学科・増田澄男教授に深く感謝致します．

有意義な御教示ならびに御助言を頂きました神戸大学工学部電気電子工学科・田中初一教授に深く感謝致します．

貴重な御意見ならびに御助言を頂きました神戸大学工学部電気電子工学科・沼昌宏教授に深く感謝致します．

また，本研究をすすめるにあたり，適切な御指導及び御助言をくださった神戸大学工学部電気電子工学科・山口一章助手に深く感謝いたします．さらに，いろいろ手助けして頂いた電子情報数理研究室の皆様に深く感謝いたします．

参考文献

- [1] K. Sugiyama, Graph Drawing and Applications – For Software and Knowledge Engineering, World Scientific, Singapore, 2002.
- [2] P. Eades, “A heuristic for graph drawing,” *Congressus Numerantium*, vol.42, pp.149-160, 1984.
- [3] T. Kamada and S. Kawai, “An algorithm for drawing general undirected graphs,” *Information Processing Letters*, vol.31, no.1, pp.7-15, 1989.
- [4] 角 浩二, 田中寿俊, 榎原博之, 中野秀男, “グラフ描画アルゴリズムの性能評価,” *電子情報通信学会論文誌 (A)*, vol.J79-A,no.3 pp.680-686, 1996.
- [5] 川西和人, 増田澄男, 山口一章, “2 種類の理想距離による Eades のグラフ描画法の改良,” *電子情報通信学会論文誌 (A)*, vol.J83-A, no.9, pp.1117-1121, 2000.
- [6] T. Kamada, *Visualizing Abstract Objects and Relations – A Constraint-Based Approach*, World Scientific, Singapore, 1989.
- [7] 菊地玄人, 増田澄男, 山口一章, “グラフ描画における頂点ラベルの配置法,” *電子情報通信学会論文誌 (A)*, vol.J85-A, no.1, pp.141-147, 2002.
- [8] K.G. Kakoulis and I.G. Tollis, “A unied approach to labeling graphical features,” *Proc. of the 14th Annual Symposium on Computational Geometry*, pp.347-356, Minneapolis, 1998.
- [9] 亀田貴之, 今井桂子, “ラベル配置問題に対する実験的評価,” *情報処理学会研究報告, アルゴリズム 76-9*, 2001.
- [10] K.G. Kakoulis and I.G. Tollis, “An algorithm for labeling edges of hierarchical drawings,” *Proc. of the 5th International Symposium on Graph Drawing, Lecture Notes in Computer Science*, vol.1353, pp. 169-180, Springer, Berlin, 1997.

- [11] U. Dgrusöz, K.G. Kakoulis, B. Madden, and I.G. Tollis, "Edge labeling in the Graph Layout Toolkit," Proc. of the 6th International Symposium on Graph Drawing, Lecture Notes in Computer Science, vol.1547, pp.356-363, Springer, Berlin, 1998.
- [12] K.G. Kakoulis and I.G. Tollis, "On the edge label placement problem," Proc. of the 5th International Symposium on Graph Drawing, Lecture Notes in Computer Science, vol.1190, pp.241-256, Springer, Berlin, 1997.
- [13] M. Forman and F. Wagner, "A packing problem with applications to lettering of maps," Proc. of the 7th Annual Symposium on Computational Geometry, pp.281-288, 1991.
- [14] J. Christensen, J. Marks, and S. Shieber, "An empirical study of algorithms for point-feature label placement," ACM Transaction on Graphics, vol.14, no.3, pp.203-232, 1995.
- [15] T. Kameda and K. Imai, "Map label placement for points and curves," IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences, vol.E86-A, no.4, pp.835-840, 2003.
- [16] S. Nakano, T. Nishizeki, T. Tokuyama, and S. Watanabe, "Labeling points with rectangles of various shapes," Proc. of the 8th International Symposium on Graph Drawing, Lecture Notes in Computer Science, vol.1984, pp.91-102, Springer, Berlin, 2001.
- [17] M. van Kreveld, T. Strijk, and A. Wol, "Point set labeling with sliding labels," Proc. of the 14th Annual Symposium on Computational Geometry, Minneapolis, pp.337-346, 1998.
- [18] F. Wagner and A. Wol, "A combinatorial framework for map labeling problem," Proc. of the 6th International Symposium on Graph Drawing, Lecture Notes in Computer Sciences, vol.1547, pp.316-331, Springer, Berlin, 1998.
- [19] F. Wagner, A. Wol, V. Kapoor and T. Strijk, "Three rules suce for good label placement," *Algorithmica*, vol.30, no.2, pp.334-349, 2001.
- [20] A. Wol, The Map-Labeling Bibliography,
<http://i11www.ira.uka.de/map-labeling/bibliography/>.
- [21] S.A. Hirsch, "An algorithm for automatic name placement around point data," *The American Cartographer*, vol.9, no.1, pp.5-17, 1982.
- [22] G.W. Klau and P. Mutzel, "Combining graph labeling and compaction," Proc. of the 7th International Symposium on Graph Drawing, Lecture Notes in Computer Science, vol.1731, pp.27-37, Springer, Berlin, 1999.

- [23] C. Binucci, W. Didimo, G. Liotta, and M. Nonato, “Labeling heuristics for orthogonal drawings” Proc. of the 10th International Symposium on Graph Drawing, Lecture Notes in Computer Science, vol.2265, pp.27-37, Springer, Berlin, 2002.
- [24] 浅野孝夫, 情報の構造 [上] データ構造とグラフアルゴリズム, 日本評論社, 1994.
- [25] 高谷佳男, 局所探索処理によるグラフ描画アルゴリズムの改良, 神戸大学工学部卒業論文, 2002.
- [26] J.E. Hopcroft and R.M. Karp, “An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs,” SIAM J. Comput., vol.2, no.4, pp.225-231, 1973.
- [27] 阿部 昇, 舟川国男, 増田澄男, 山口一章, “地名配置におけるラベル候補作成法,” 電気関係学会関西支部連合大会講演論文集, G11-1, 2003.

関連発表

学術論文

- [1] 阿部 昇, 増田澄男, 山口一章, “グラフ描画における辺のラベルの配置法,” 電子情報通信学会論文誌 (A), vol.J85-A, no.3, pp.306-314, 2002.
- [2] 阿部 昇, 増田澄男, 山口一章, “辺がラベルをもつ無向グラフの描画法,” 電子情報通信学会論文誌 (A), vol.J86-A, no.8, pp.848-859, 2003.
- [3] N. Abe, S. Masuda and K. Yamaguchi, “Placement of vertex labels in a graph drawing,” IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences, vol.E87-A, no.10, pp.2774-2779, 2004.
- [4] 阿部 昇, 山口一章, 増田澄男, “グラフ描画へのラベル配置アルゴリズムの実験的評価,” 電子情報通信学会論文誌 (A), 採録決定 .

国際学会

- [1] Noboru Abe, Sumio Masuda and Kazuaki Yamaguchi, “Placement of Vertex Labels in a Graph Drawing,” Proc. of the 3rd Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications, pp.69-78, Tokyo, Japan, 2003.
- [2] Noboru Abe, Sumio Masuda and Kazuaki Yamaguchi, “An Algorithm for Drawing a Graph with Vertex Labels,” Proc. of the 7th Japan-Korea Joint Workshop on Algorithms and Computation, pp.267-275, Sendai, Japan, 2003.

学術講演

- [1] 阿部 昇, 増田澄男, 山口一章, “グラフ描画における辺のラベルの配置法,” 電子情報通信学会技術研究報告, COMP2001-2, 2001.

- [2] 阿部 昇, 増田澄男, 山口一章, “辺がラベルをもつグラフの描画アルゴリズム,” 電子情報通信学会総合大会講演論文集, D-1-2, 2002.
- [3] 阿部 昇, 増田澄男, 山口一章, “グラフ描画への頂点ラベルの配置,” 電気関係学会関西支部連合大会講演論文集, G11-8, 2002.
- [4] 阿部 昇, 増田澄男, 山口一章, “頂点がラベルをもつグラフの描画アルゴリズム,” 情報処理学会第 65 回全国大会講演論文集, 3C-2, 2003.
- [5] 阿部 昇, 舟川国男, 増田澄男, 山口一章, “地名配置におけるラベル候補作成法,” 電気関係学会関西支部連合大会講演論文集, G11-1, 2003.
- [6] 阿部 昇, 増田澄男, 山口一章, “グラフ描画への頂点ラベル及び辺ラベルの配置,” 電気関係学会関西支部連合大会講演論文集, G11-3, 2003.