

筑波大学大学院博士課程

システム情報工学研究科修士論文

クラスタ2部グラフの
階層的な円周描画手法の開発

伊藤 隆朗

(コンピュータサイエンス専攻)

指導教員 三末 和男

2010年3月

概要

データ間の関係に着目した分析は広く行われている．異なる二つの集合間に渡る関係性に注目した情報分析の要求も多い．このような関係を 2 部グラフとして表現し，視覚的に描くことは関係の把握を支援する．この際，全体を俯瞰したのちに一部分をより詳細に見ていくプロセスが望まれるが，既存の 2 部グラフ描画手法では全体の俯瞰の支援は行えるものの，詳細な部分に踏み込んでいくような段階的な情報探索の支援を十分に行えていない．

本研究では，実世界の情報の多くが階層構造をなして整理されていることに着目し，2 部グラフと階層構造を同時に表現できる描画手法を開発した．この手法では，2 部グラフのうち片方のノード集合が階層構造を持つグラフ構造をクラスタ 2 部グラフと呼び，階層構造を持つノード群を階層的に円周状に配置する．階層構造を考慮した情報提示を行うことによって，全体を俯瞰したのちに一部分をより詳細に見ていく情報探索の支援が可能になると考えられる．

開発した手法について美的基準に基づくレイアウトの評価を行った．その結果，本手法は設定した美的基準を満足し視覚的な混雑を軽減できることが分かった．加えて，具体的なデータの可視化を行い，本手法が段階的な情報探索の支援に有効であることが分かった．

目次

第1章	はじめに	1
1.1	情報可視化と Graph Drawing	1
1.2	2部グラフ	1
1.3	2部グラフの表現方法	2
1.4	大規模な情報分析時の問題点	2
1.5	本研究の目的とアプローチ	3
1.6	本論文の構成	4
第2章	関連研究	5
2.1	2部グラフ描画手法	5
2.2	円周配置手法	5
2.3	大規模グラフの描画手法	6
2.4	複合グラフの描画手法	6
2.5	エッジ描画	7
第3章	クラスタ2部グラフ	8
3.1	定義と記法	8
3.2	クラスタ2部グラフ描画の用途	9
3.3	クラスタ2部グラフ描画への要求	10
第4章	階層型アンカーマップ	11
4.1	アンカーマップ	11
4.2	階層型アンカーマップの表現スタイル	11
4.3	用語	14
4.4	美的基準	15
第5章	ノードの配置手法	18
5.1	技術的な課題	18
5.2	ノード配置の概要	20
5.3	子要素の配置角度の決定	21
5.4	半径と初期距離の決定	22
5.5	クラスタマップの向きの決定	24
5.6	クラスタマップの配置	27

5.7	フリーノードの配置	30
第6章	エッジの描画手法	32
6.1	曲線を用いたエッジ描画	32
6.2	円周配線	33
6.3	ローカルバンドル	34
第7章	階層情報付加のためのインタフェース	36
7.1	インタラクティブなグラフ描画	36
7.2	クラスタマップ化部分の指定インタフェース	36
7.3	関係の弱い部分の検出	37
第8章	描画手法の評価と考察	40
8.1	クラスタマップの配置スタイルの評価	40
8.1.1	概観の維持の評価	41
8.1.2	空間効率の評価	42
8.1.3	考察	44
8.2	クラスタマップの向き決定方法の評価	46
8.2.1	評価結果	47
8.2.2	考察	47
第9章	応用例	53
9.1	階層情報を持つデータへの応用	53
9.2	階層情報を持たないデータへの応用	55
第10章	結論	61
	謝辞	62
	参考文献	63

目次

1.1	2部グラフの表現手法	3
3.1	クラスタ2部グラフのイメージ	8
4.1	アンカーマップによるWebページと訪問者の描画例	12
4.2	階層型アンカーマップのノード配置スタイル	13
4.3	円周配線	13
4.4	ローカルバンドル	14
4.5	子要素の配置角度とクラスタマップの向き	15
5.1	4種類の配置スタイル	19
5.2	クラスタマップの向きによる可読性の違い	20
5.3	クラスタはひとつのアンカーとして考える	21
5.4	子要素の配置角度	22
5.5	内接スタイル	23
5.6	向きを求める際の兄弟クラスタマップの状態	25
5.7	指標の計算方法	26
5.8	単純にスプリングモデルを適用した際に起こる問題点	30
5.9	フリーノードのレイアウト方法	31
6.1	円弧の半径と角度	33
6.2	ローカルバンドルと円周配線の組み合わせ	34
6.3	ローカルバンドルの描画手順	35
7.1	自由曲線による切れ目指定	37
7.2	アンカーを囲まない切れ目指定	37
7.3	アイコンをクリックするとクラスタマップ化	38
8.1	ベースアンカーマップの生成イメージ	41
8.2	概観の維持の評価結果	42
8.3	配置スタイルに関する実験の描画結果	43
8.4	空間効率の評価結果	44
8.5	弦に接するスタイルでアンカーが近づく例	45

8.6	向きの評価に用いたグラフの構造	48
8.7	向きに関する実験の描画結果 構造 1~3	49
8.8	向きに関する実験の描画結果 構造 4~6	50
9.1	アクセスログの描画例	54
9.2	アンカーマップによる商品と購入者の関係の可視化例	56
9.3	階層型アンカーマップによる商品と購入者の関係の可視化例	57
9.4	アンカーマップによる論文と著者の関係の可視化	59
9.5	階層型アンカーマップによる論文と著者の関係の可視化	60
9.6	子クラスタマップの拡大図	60

表目次

8.1	実験に用いたグラフの規模	40
8.2	概観の維持の評価結果	42
8.3	空間効率の評価結果	42
8.4	エッジとクラスタマップ交差の評価結果	52
8.5	B方法・C方法の順位	52

第1章 はじめに

1.1 情報可視化と Graph Drawing

情報可視化 (Information Visualization) とは, コンピュータグラフィックスを活用して抽象的なデータを人間が直接見ることができる形で表現し, 人間のデータ理解を支援することを目的とした研究分野である.

コンピュータサイエンスにおける可視化の研究には, 科学技術計算の結果をコンピュータグラフィックスで表現する科学的可視化 (Scientific Visualization) と呼ばれる研究分野がある. この技術を科学技術分野だけではなく, より広い抽象的なデータを対象にした研究分野が情報可視化である.

抽象的なデータの関係や構造を表現するためにグラフがよく使われる. グラフを適切に可視化することにより, 抽象的なデータの関係を人間が理解しやすいように表現することができる. しかしながら, データ量が多くなると人の手だけでレイアウトを行うことは困難となる. そのため, グラフを計算機によって人間が理解しやすいように自動的に描く手法が求められており, 情報可視化の中でも特にグラフ描画問題 (Graph Drawing) と呼ばれている. 本研究は, このグラフ描画問題を対象としている.

1.2 2部グラフ

顧客とその顧客の購入した商品, 論文と著者, Web ページとそのページの閲覧者など, 実社会の様々なところで2つの集合間の関係が現れる. このような2つの集合間の関係構造は2部グラフとして表現することができる.

2部グラフとは, ノードを二つの排他的な集合に分割し, 各集合内のノード同士にエッジが無いようにできるグラフのことであり, 以下のように表すことができる.

グラフ: $G = (V, E)$

ノード: $V = A \cup B$ ただし, $A \cap B = \emptyset$

エッジ: $E \subseteq A \times B$

顧客とその顧客の購入した商品を例に挙げると, 顧客を集合 A の要素, 商品を集合 B の要素とし, 顧客に対応するノードとその顧客の購入した商品に対応するノードの間をエッジでつなぐことで, 顧客と商品との関係を2部グラフとして表現することができる.

1.3 2部グラフの表現方法

2部グラフを描くことで、商品と顧客や Web ページと訪問者のような多対多の関係構造の把握を支援することができる。ここで、2部グラフの代表的な表現方法を紹介する。図 1.1 は商品と購買者の関係を 4 つの表現方法を用いて表した例である。

図 1.1(a) は行列形式である。購買者が買った商品の欄に購入した数を記入している。売り上げ数など細かな情報を読み取ることができるが、商品間の関係など関係構造の直感的な把握に向いているとは言い難い。

次に、図 1.1(b) のような 2 層形式で表現したものが挙げられる。2 層形式はネットワーク図の表現でよく用いられる網図形式の中でも 2 部グラフに特化した表現である。片方の列に購買者を、もう片方の列に商品を並べて書き、購入者と買った商品を直線でつないだものである。商品と購買者の描かれる領域がはっきりと分かれているため、視覚的な区別がしやすいと言える。2 部グラフの表現としてはオーソドックスなものである [1]。

図 1.1(c) は網図形式で表現したものである。示した図は、網図の描画手法としてよく用いられる力指向アルゴリズム [2, 3] によって描画したものである。この表現は全体的な関係構造が視覚的に把握しやすいと言える。しかし、商品と購買者のノードが混在しているためそれらの区別が困難である。

最後に、Misue の提案したアンカーマップ表現 [4, 5] がある (図 1.1(d))。この表現では、2 部グラフの片方の集合のノードを円周上に等間隔に配置し、もう片方の集合のノードを力指向アルゴリズムを利用して配置する。片方の集合を円周上に配置する表現は、アンカーマップに限らず 2 部グラフの描画ではよく用いられる [6, 7, 8, 9]。この表現の特徴は、ノードの区別が明確であり、なおかつ商品同士の関連性など全体的な構造も視覚的に把握しやすい点である。アンカーマップの特徴については、4.1 節で詳しく述べる。

1.4 大規模な情報分析時の問題点

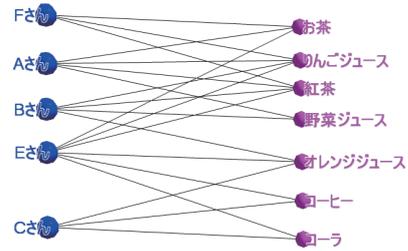
可視化を用いて情報の探索を行うプロセスの指針として、Shneiderman が提唱した「Overview first, zoom and filter, then details on demand」という方針がある [10]。

先に挙げた表現手法の場合、比較的小規模なグラフであれば大まかな傾向から細かな接続関係までを読み取ることができるため、2 部グラフの分析を支援することができる。しかしながら、実世界では大規模なデータを持つことが多い。例えば、著者の運営する Web サイト¹には約 50 のページがあり、1 週間で 1000 以上の訪問者 (ユニークユーザ) が訪れている。このような情報を先に挙げた手法で可視化した場合、大まかな傾向はつかめるものの、さらに踏み込んだ情報の把握が困難となると考えられる。

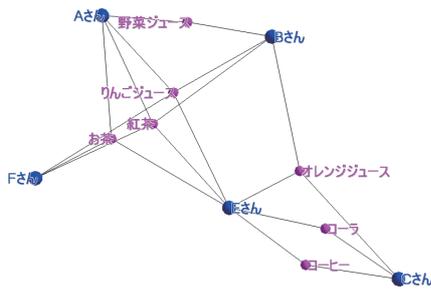
¹<http://www.clks.jp/>

	オレンジ ジュース	紅茶	コーラ	りんご ジュース	お茶	野菜 ジュース	コーヒー
Aさん	0	1	0	1	2	1	0
Bさん	3	2	0	4	0	3	0
Cさん	2	0	11	0	0	0	5
Eさん	4	2	2	1	2	0	1
Fさん	0	2	0	2	3	0	0

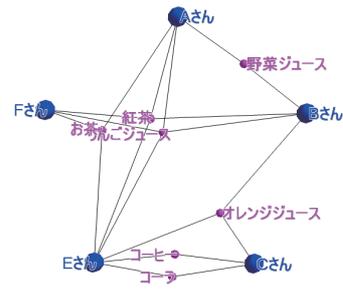
(a) 行列表現



(b) 2層形式



(c) 網図形式 (スプリングモデル)



(d) アンカーマップ表現

図 1.1: 2部グラフの表現手法

1.5 本研究の目的とアプローチ

本研究は、大規模な2部グラフの情報分析支援を目的とし、大まかな傾向を把握した後に局所的な情報の把握を行うような段階的な情報探索を支援する描画手法の開発を目指す。

この目標を達成するため、本研究では大規模な情報の多くが階層構造をなして [11, 12] 整理されていることに着目した。例えば、Web ページと訪問者では、Web ページは内容に応じて設計されたディレクトリ構造を持っており、訪問者は国や使用ブラウザ、言語などの分類情報を持っている。分類情報は深さ1の階層情報とみなすことができる。商品と購入者の場合、商品は階層化された種別情報を持っており、購入者も年代や性別などの情報を持っている。階層情報は、情報を分析する際の有用な手がかりとなると考えられる。

本研究では、2部グラフ構造と階層構造の両方を持ったグラフ構造のうち、特に、2部グラフの片方のノード集合が階層構造を持つグラフ構造「クラスタ2部グラフ」を対象とし、2部グラフと階層情報を同時に表現できる描画手法の開発する。この手法では、階層情報を元に2部グラフをアンカーマップ表現を用いて階層的に配置していく。この描画手法は、全体の大まかな傾向と局所的に構成される傾向の両方をアンカーマップによって提示することができる。これは Shneiderman が提唱した方針に沿っており、目標として掲げた段階的な情報探索の支援が可能になると考えられる。この描画手法を「階層型アンカーマップ」と呼ぶ。

1.6 本論文の構成

本章では，2部グラフの表現手法と問題点を説明し，研究の目的とアプローチを述べた．第2章では関連研究を述べ，第3章では本研究で対象とするクラスタ2部グラフの導入と定義を行う．第4章では開発したレイアウト手法の概要を述べ，第5章でノードの配置手法，第6章でエッジの描画方法を述べる．また，第7章では，インタラクティブなグラフ描画のためのインタフェースについて述べる．第8章では，開発した配置手法に関する評価について述べ，第9章で応用例を示し，第10章で結論を述べる．

第2章 関連研究

2.1 2部グラフ描画手法

2部グラフを2層形式やそれを拡張したモデルでの描画アルゴリズムやエッジ交差最小化に関する定理の研究がなされてきた。Newton[1]らは、2層形式描画において、エッジ交差最小化を目指したヒューリスティックアルゴリズムを提案した。Zheng[13]らは、2部グラフの片方のノード群を直線状に配置し、もう片方のノード群を直線の片側に配置するOLOCモデルと、両方のノード群を別の領域内に配置するTCモデルを提案し、それら2種類にモデルにおける定理の証明を行った。Giacomo[8]らは、エッジが交差しないよう2部グラフを2つの曲線上に配置する描画手法を提案した。以上の研究は、主にエッジ交差最小化に着目した研究である。エッジ交差の最小化は、視覚的な混雑を抑え、個々のノードの接続関係を読み取るためには重要な要因であるが、大規模なグラフを描画して大まかな傾向を読み取る用途には適さない可能性がある。

一方、大規模な2部グラフから大まかな傾向を把握するための描画手法として、Misue[4, 5]の提案した「アンカーマップ」がある。この手法では、2部グラフの片方のノード集合を円周上に等間隔に配置し、もう片方のノード集合を力指向アルゴリズム[2]によって配置する。同様のスタイルは、Thielら[6]や、Donovanら[7]のシステムでも用いられている。この表現形式は、大規模な2部グラフを俯瞰し、大きな傾向は読み取れるものの、その後のより詳細な情報探索を行うのは困難なものである。本研究はこの点を解消し、段階的な情報探索を可能とするためにアンカーマップの拡張を行なったものである。

アンカーマップに類似したスタイルとして、Naudらの3D-SE Viewer[9]がある。この手法では、2部グラフのノードを3D空間の同心球上に配置する。また、著者らはアンカーマップを3D空間に拡張し、片方のノード群を球面上に配置する手法「スフィアアンカーマップ」[14]を開発している。これらの手法は3D拡張によってグラフのスケラビリティを向上させているが、閲覧の際に3D構造の把握が必要となる。この把握には、回転させるなど操作を行う必要があり、情報の把握に時間を要してしまう[15]。そのため、本手法では2D上での効果的なレイアウト手法の開発を目指している。

2.2 円周配置手法

一般グラフを、円周上に配置するグラフ描画アルゴリズムの研究が多くなされている。円周配置研究は、大きく、一つの円を対象とするものと複数の円を配置するものに分けられる。

まず，単一の円を対象とした研究について述べる．Sixらは，biconnected graph をエッジ交差を抑えながら高速に円周配置する手法 CIRCULAR[16] を提案した．Gansnerら [17] は，単一の円周配置方法として，エッジ長を短く配置する手法，円周外にエッジを引き回す手法，エッジバンドルの3つの技術を提案した．Misueは，アンカーマップにおいて，同じノードを共有する円周上のノードが，できるだけ近くに配置されるような並び順を決定するアルゴリズムを提案した．

次に，複数の円を対象とした研究について述べる．Sixらは，CIRCULARを拡張し，複数の円を用いて一般グラフを配置するレイアウト手法 [16] を提案した．また，彼らは力指向レイアウトと円周レイアウトを統合した手法を含む，ユーザがグルーピングを行う円周描画手法のためのフレームワークの提案も行った [18]．Kaufmann[19]らは，Sixのフレームワークをユーザインタラクションに適合するように拡張を行った．本研究は，複数の円を対象とした研究であるが，2部グラフの配置に着目している点で先のものとは異なる．

2.3 大規模グラフの描画手法

大規模グラフの描画手法として，木構造の描画手法の研究が数多くなされてきた [20]．ノードリンクダイアグラムスタイルとしては，3D描画手法のCone Trees[21]や，MunznerのH3[22]が提案された．近年の研究では，TreeMapスタイル [23]の研究が多くなされている．

一般大規模グラフの描画手法の研究としては，力指向アルゴリズムをマルチレベル法を用いて高速化するFM³[24]や，GPUを用いて高速化を行う手法 [25]が提案された．また，グラフをクラスタリングし，TreeMapをベースとしたレイアウト [26]や，階層をトラバースして空間充填曲線に並べていく [27]ことで高速に大規模なグラフのレイアウトを行う手法が提案された．

大規模なグラフ構造の把握支援のために，オーバービューと詳細に別の描画手法を用いるアプローチをとっている研究がある．Baurら [28]はマイクロ構造に関して円周配置を行い，マクロ構造は一般的な力指向方式 [2]を用いてレイアウトを行うことで概観の把握と詳細情報の把握を支援する手法を提案した．Dwyerら [29]はオーバービュー構造に高速なグラフィックレイアウトシステムを，詳細なサブグラフに質の高いグラフィックレイアウトシステムを使ってレイアウトを行い，インタラクションによって情報探索を行えるようにした．Henryら [30]は，密な連結関係にある部分に行列表現を用いる網図とのハイブリット形式によって大規模グラフの把握を支援する手法を提案した．本研究では，これらの研究とは異なり，段階的な情報探索を可能とするために各部分グラフに同一の表現を用いている．

2.4 複合グラフの描画手法

2種類以上の構造を合わせて同時に提示する手法の研究もおこなわれている．Eades[31]らは，一般グラフ構造と木構造の両方を持つクラスタグラフの描画手法を提案した．Frishmanら [32]は，動的なクラスタグラフに対応した自動レイアウト手法を提案した．また，Hoら

[11] はクラスタグラフの 3D 描画手法を提案した．この手法では，Cone Trees[21] のようなレイアウトを用いてクラスタの表現を行っている．

Omote ら [33] は，クラスタグラフを拡張し，一般グラフ構造とクラスタ間で葉の共有を許す木構造を持つインターセクティングクラスタグラフの描画手法を提案した．さらに，高 [34] は，この種類のグラフの空間効率の良い描画方法とソーシャルネットワーク分析への応用を示した．Yingxin ら [35] は，各ノードが多次元の属性を持った情報を，spherical SOM と円周配置を用いて可視化する手法を提案した．Itoh ら [36] は，一般的なグラフと複数カテゴリに所属する構造を持つ大規模グラフを描画する手法として，力指向手法と Tree Map のような space-filling 手法のハイブリットな手法を提案した．また，Collins ら [37] は任意の 2 つ以上の可視化結果を結合させるシステム VisLink を提案した．

2 部グラフと別の構造を同時に提示する研究としては，佐藤の ClusteredAnchorViz[38] がある．この手法では，可読性の向上を目的として，アンカーマップを用いてレイアウトされた 2 部グラフに対し，スプリングモデルを用いて配置される方のノード群をクラスタリングしてノード類似度情報を付加表示した．これに対し，本研究では段階的な情報探索を目的としており，アンカーマップで円周に配置される方のノード群のレイアウト方法に改良を加えている．また，改良を加えているノード群が異なるため，本手法は佐藤の手法と組み合わせることが可能である．この他に，Xu ら [39] は，2 部グラフのうち片方のノード集合内にもエッジが存在する半 2 部グラフの描画手法を提案した．

2.5 エッジ描画

ノードの配置方法を工夫することで，描画結果の可読性を向上させる研究がある一方で，エッジの描画方法を工夫することで描画結果の可読性を向上させる研究がある．

Holten[12] は，クラスタグラフのエッジを階層構造に沿って結束して描くことによって可読性を向上させる手法を提案した．エッジを結束して可読性を向上させる手法はエッジバンドルと呼ばれている．また，Cui ら [40] は，描画されたグラフ構造を解析しエッジバンドルを行うことのできる手法を提案した．

Baur ら [28] は，サブグラフ間のエッジ接続のために円を周回するエッジの配線方法を提案した．

本研究でも，提案する描画スタイルに合ったエッジ配線方法の工夫をおこなっており，Holten[12] や Baur ら [28] の効果を本手法に適用できるように改良を行っている．

第3章 クラスタ2部グラフ

3.1 定義と記法

2部グラフと階層構造やカテゴリ情報を同時に提示する手法は、佐藤 [38] や石原 [41] によって既に試みられているが、このようなグラフの形式的な定義がなされていない。そこで、描画手法の開発を行う前に、本研究で取り扱うクラスタ2部グラフの定義をおこなう。

クラスタ2部グラフを $G = (A, F, C, E, T)$ とする。 $G_B = (A \cup F, E)$ は2部グラフであり、 A と F は有限のノード集合を表し、これらは互いに排他である。 E は有限のエッジ集合であり、 $A \times F$ の部分集合である。 $G_T = (A \cup C, T)$ は木構造であり、 C は葉ではないノードの集合である。 C の要素をクラスタと呼ぶ。また、 G_T の葉は A と完全に一致する。図 3.1 にクラスタ2部グラフのイメージを示す。

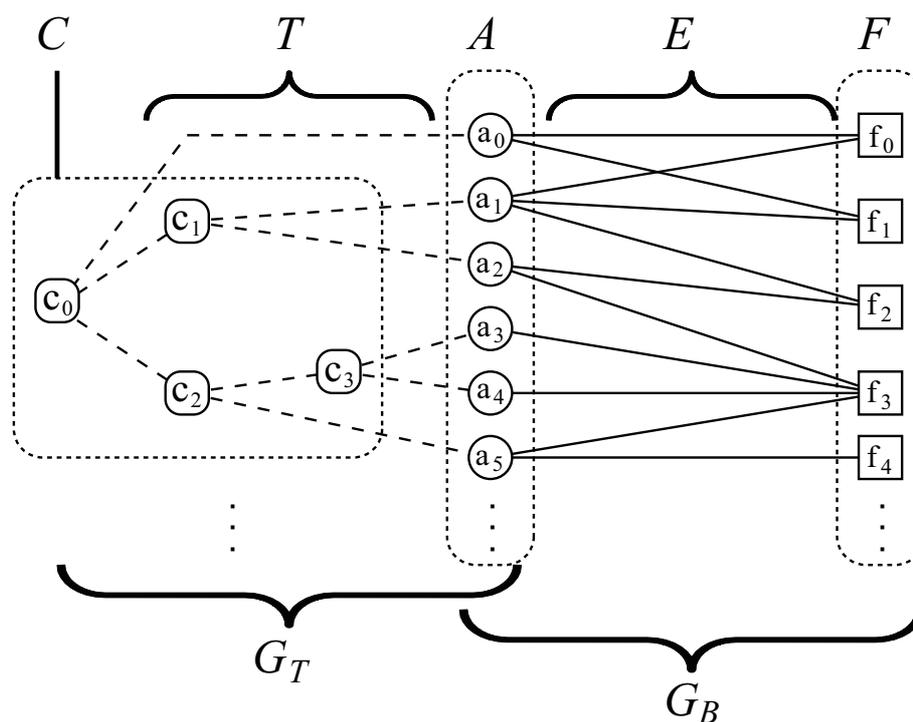


図 3.1: クラスタ2部グラフのイメージ

次に、本論文で用いるクラスタ2部グラフに関する記法を述べる。 $Ch(c)$ をクラスタ $c \in C$ の子の集合とし、 $De(c)$ をクラスタ c の子孫のノードの集合とする。さらに、 $A'(c)$ を $De(c)$ 内のすべての葉となるノードの集合、すなわち $A'(c) = De(c) \cap A$ とし、 $C'(c)$ を $De(c)$ 内のすべてのクラスタとなるノードの集合、すなわち $C'(c) = De(c) \cap C$ とする。 $E'(c)$ を、 $A'(c)$ 内のノードと接続するエッジの集合、すなわち $E'(c) = \{(a,x) \in E | a \in A'(c)\}$ とする。また、 $F'(c)$ を $A'(c)$ 内のノードと連結するノードの集合、すなわち $F'(c) = \{x \in F | (a,x) \in E'(c)\}$ とする。

フリーノード f と接続するアンカーの集合を $A''(f)$ とする。さらに、この $A''(f)$ をすべて含むクラスタ、すなわち $A''(f) \subset A'(c)$ となるクラスタの集合を $C_b(f)$ とする。 $C_b(f)$ をフリーノード f の所属クラスタの集合と呼び、その要素を所属クラスタと呼ぶ。また、フリーノードの所属クラスタ群 $C_b(f)$ のうち最も下にあるクラスタを最下所属クラスタと呼び $c_b(f)$ とする。

3.2 クラスタ2部グラフ描画の用途

クラスタ2部グラフは、木の根に近い部分では全体的な構造を表し、各クラスタごとに局所的な2部グラフを構成する構造となっている。この構造を描くことによって、大規模な情報を分析する際に、全体を俯瞰したのちに詳細に知りたい部分を分析する段階的な情報分析を支援することができると考えられる。クラスタ2部グラフ描画の用途としては、つぎの2通りが考えられる。

ひとつは、2部グラフ構造とは別に木構造が取得でき、その木構造をもとに構成したクラスタ2部グラフを描く用途である。例として、Webサイトのアクセスログや商品の購買情報が挙げられる。Webサイトのアクセスログの場合、Webページとアクセスした訪問者として構成される2部グラフ構造と、Webページのディレクトリ構造の2つの構造が取得できる。Webページのディレクトリ構造は、ページの内容によって分けられていることが多いため、この情報を用いてクラスタ2部グラフを構成することで、意味を持ってまとまったページ群と訪問者の関係性や、ディレクトリ内でのページ群と訪問者との関係性を表すことができると考えられる。商品と購買情報の場合、商品と購入者として構成される2部グラフ構造と、商品のカテゴリで構成される階層構造が取得できる。商品のカテゴリ情報を用いてクラスタ2部グラフを構成することで、カテゴリと購入者間の関係性やカテゴリ内での購入傾向を表すことができると考えられる。この他、2部グラフ構造から木構造を生成し、それを元にクラスタ2部グラフを構成することも可能である。たとえば、2部グラフをクラスタリングすることで階層構造を付加することができる。

もうひとつは、2部グラフ構造を何らかの手法で描画し、その結果を見ながら階層情報を付加していくことで情報分析を進めていく用途が考えられる。

3.3 クラスタ2部グラフ描画への要求

先に述べた2通りの用途を想定し、本研究ではクラスタ2部グラフの描画において以下のような要求を考慮した。括弧内には、商品の購買情報を用いた例を示す。

Req.1 木構造を持たないノードとクラスタによって形作られる2部グラフ構造の提示
(例、カテゴリと購入者の関係の提示)

Req.2 各クラスタ内で形成される2部グラフ構造の提示
(例、各カテゴリ内での商品と購入者の関係や、カテゴリ内での小カテゴリと購入者の関係の提示)

Req.3 クラスタの階層構造の提示
(例、カテゴリの階層構造の提示)

クラスタ2部グラフの場合、全体を俯瞰する際に提示すべき大局的な関係性は、クラスタ間で形成される関係性にあため Req.1 を考慮した。また、詳細に知りたい部分を分析する際に提示すべき局所的な関係性は、あるクラスタ内で形成される関係性に当たるため Req.2 を考慮した。加えて、着目している局所的な関係性が全体の中でどこに位置しているのかを提示することも情報把握には重要であると考え Req.3 を考慮した。

第4章 階層型アンカーマップ

4.1 アンカーマップ

アンカーマップは、Misue によって提案された大規模 2 部グラフ向けの描画手法である。この手法では、2 部グラフの片方の集合のノードを円周上に等間隔に配置固定し、もう片方の集合のノードを力指向のレイアウト手法によって配置する。固定された方のノードをアンカー、力指向によって配置される方のノードをフリーノードと呼ぶ。Misue はこのレイアウトスタイルに加え、同じフリーノードを共有するアンカー同士ができるだけ近くに配置されるようなアンカーの並び順決定アルゴリズムも提案した。

アンカーマップの利点の一つは、フリーノードの集合の規模や位置によって 2 部グラフの大まかな傾向を読み取ることができることである。図 4.1 に Web ページをアンカー、訪問者をフリーノードとして描いた例を示す。この図では上と右にフリーノードが集中していることが分かる。このことから、「どの Web ページが多く見られているのか」や「上のページ群と右のページ群を見ているユーザが異なる」といった、大まかな傾向を把握することができる。このほかに、中央付近にいくらかのフリーノードが見て取れる。このノードのドメインを調べたところ、そのほとんどがクローラーであることが分かった。以上のように、2 部グラフをアンカーマップを用いて描画することによって、2 部グラフの大まかな傾向の把握や、特異なノードの発見を支援することができる [5]。

4.2 階層型アンカーマップの表現スタイル

クラスタ 2 部グラフの描画手法を開発するに際し、前章で述べたアンカーマップの特徴を生かして 2 部グラフの傾向の読み取りやすさを維持しつつ階層構造を提示することを考えた。基本アイデアは、アンカーマップを階層構造に従って再帰的に配置していくことである。

図 4.2 に、Web ページと訪問者を例とした本手法のノード配置スタイルを示す。例として用いたデータには、13 のページと 3 つのディレクトリ「/a」「/b」「/b/c」がある。このディレクトリ構造に従って、各ディレクトリ下のページ・ディレクトリによって一つのアンカーマップを形成するように描く。これによって、Req.1 や Req.2 を満足できると考えられる。また、階層的な配置によって Req.3 を満たすことができ、はじめにクラスタ間で形成される 2 部グラフの傾向を把握したのち詳細に調べたい部分の局所的な関係性の把握するプロセスを支援することができる。このように、階層的にアンカーマップを配置していく表現手法を「階層型アンカーマップ」と呼ぶ。

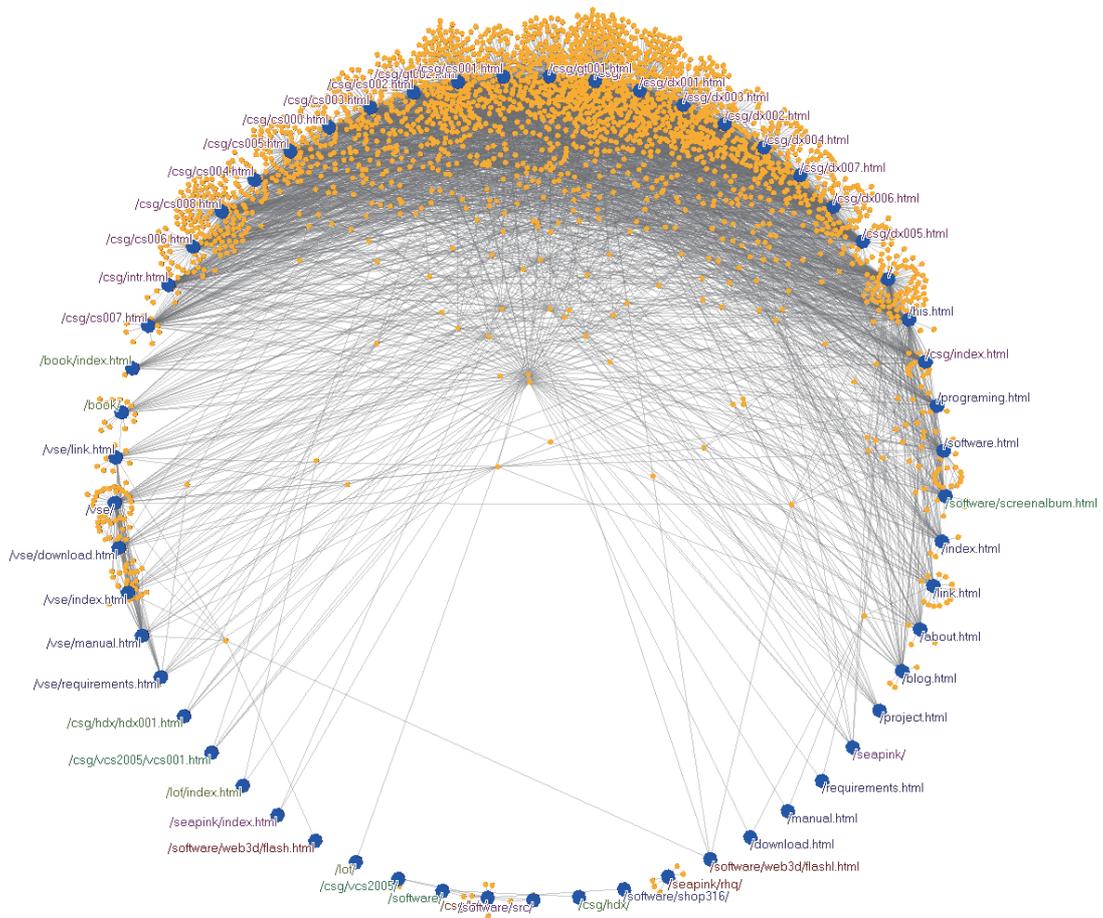


図 4.1: アンカーマップによる Web ページと訪問者の描画例

Web ページ

`/index.html`
`/r1.html`
`/r2.html`
`/a/a1.html`
`/a/a2.html`
`/a/a3.html`
`/a/a4.html`
`/b/b1.html`
`/b/b2.html`
`/b/b3.html`
`/b/c/c1.html`
`/b/c/c2.html`
`/b/c/c3.html`

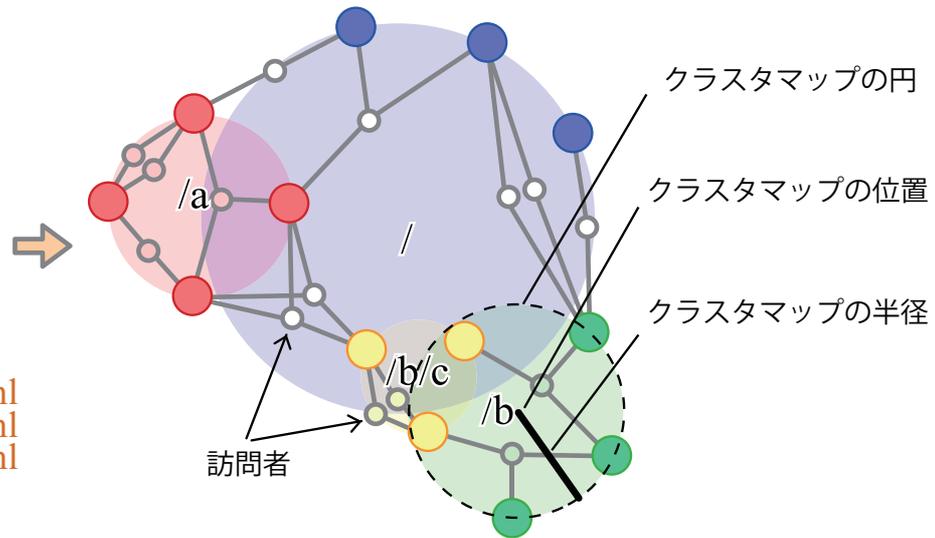


図 4.2: 階層型アンカーマップのノード配置スタイル

階層型アンカーマップのノード配置スタイルに合わせ、エッジ描画の方法にも改良を行う。本手法のノード配置スタイルでは、図 4.3(a) のような場合にマップとエッジの交差が発生してしまう。これを避けるため、図 4.3(b) のようにマップの周囲を迂回するように配線する。これを円周配線と呼ぶ。この方法は、エッジとマップの交差を避けるだけでなく、クラスタ間の接続関係をより強調できるようになると考えられる (図 4.3(c))。

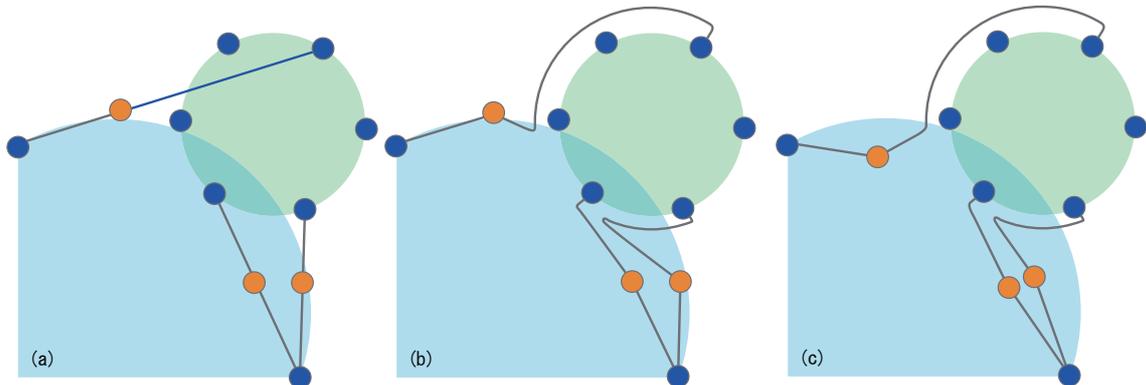


図 4.3: 円周配線

本研究では Req.2 の「各クラスタ内で形成される 2 部グラフ構造の提示」も目標としているが、ノードの配置スタイルは Req.2 よりも Req.1 を重視した配置となっており、Req.2 の表現には不十分な点がある。図 4.4(a) に具体例を示す。図上のフリーノード [a] は、緑のマップ内

のアンカーとしかつながっていないことと、どのアンカーと接続してるかの両方、つまり、全体の中での関係と局所的な関係をノードの配置のみで提示することができている。一方、図下のフリーノード [b] は青いマップのアンカーと緑のマップの両方のアンカーと接続している。このようなノードの場合、青いマップ内での関係は提示できるものの、緑のマップ内の関係 (図 4.4(b)) は十分に提示できておらず、Req.2 を十分に満たしていないと考えられる。

そこで、本研究では (図 4.4(c)) のようにエッジの描画方法を工夫することによって、あるフリーノードの局所的な関係を表すようにした。これをローカルバンドルと呼ぶ。

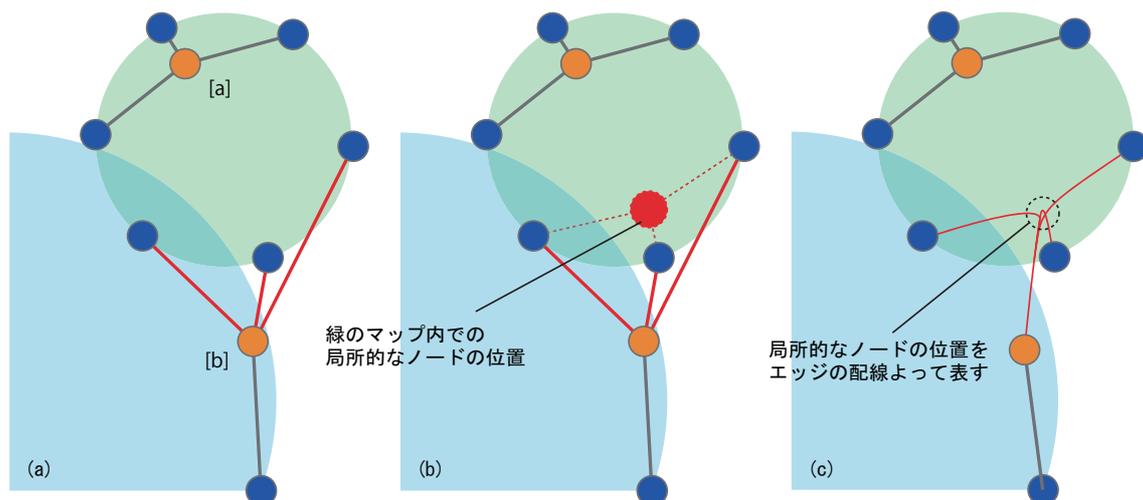


図 4.4: ローカルバンドル

4.3 用語

階層アンカーマップにおける、用語を説明する。各クラスタ下のアンカー、および子クラスタで構成されるマップをクラスタマップと呼ぶ。特に、最上位のクラスタマップをルートクラスタマップと呼ぶ。

クラスタ 2 部グラフの各クラスタは親子関係を持っており、同様にクラスタマップやアンカーも親子関係を持つ。あるクラスタの子であるクラスタによって形成されるクラスタマップを子クラスタマップ、逆にあるクラスタの親であるクラスタによって形成されるクラスタマップを親クラスタマップと呼ぶ。また、アンカーが配置されるクラスタマップもそのアンカーの親クラスタマップと呼ぶ。加えて、あるクラスタの子であるアンカー及び子クラスタマップをまとめて子要素と呼ぶ。図 4.2 の例では、赤いクラスタマップ「/a」は青いクラスタマップ「/」の子クラスタマップであり、逆に青いクラスタマップ「/」は赤いクラスタマップ「/a」の親クラスタマップである。この他、黄色いクラスタマップ「/b/c」は赤いクラスタマップ「/a」の子クラスタマップであり、青いクラスタマップ「/」に対しては孫にあたるため青いクラスタマップの孫クラスタマップと呼ぶ。

階層型アンカーマップでは、アンカーと子クラスタマップは親クラスタマップの円周上に配置される。この円を、クラスタマップの円と呼び、円の中心をクラスタマップの位置、半径をクラスタマップの半径と呼ぶ(図 4.2)。

親クラスタマップ内で、子要素が親に対してどの方向に配置されるかを表す角度を配置角度と呼び、この角度の基準をクラスタマップの向きと呼ぶ(図 4.5)。

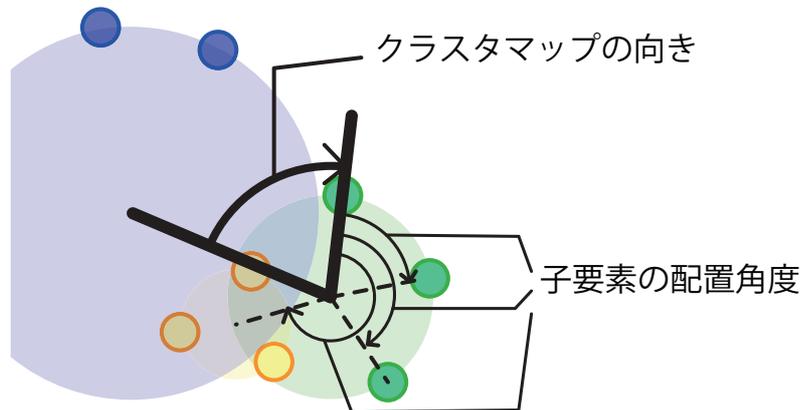


図 4.5: 子要素の配置角度とクラスタマップの向き

4.4 美的基準

階層型アンカーマップの描画手法を開発するにあたり、ノードの配置に関して以下のような美的基準を定めた。これらの基準は、先に挙げたクラスタ2部グラフ描画に対する用途や要求を考慮した上で定めたものである。

- E1 フリーノードが関係のないクラスタマップの内部にできるだけ配置されないようにする
- E2 エッジで構成される線分が関係のないクラスタマップとできるだけ交差しないようにする
- E3 同一のフリーノードを共有するアンカー同士をできるだけ近くに配置する
- E4 2部グラフの概観をできるだけ維持する
- E5 空間効率をできるだけ良くする

E1, E2 は Req.2 を満足するために有用な基準であると考え、美的基準の中で最も優先する基準として設定した。

E3 は、Misue のアンカーマップでも採用されている基準であり、本手法においても Req.1 と Req.2 にある 2部グラフの大まかな傾向を提示するために有用な基準であると考え採用した。

E4 は、2部グラフの大まかな傾向を提示する際に重要と考え採用した基準である。

E5 は一般的なグラフ描画でも採用される美的基準であり，情報分析を行う際に有用であると考え採用した．

美的基準は，描画手法の良し悪しを評価する基準として利用するため形式的な定義を行った．まず，E1 と E2 に用いる「あるフリーノードと関係のないクラスタマップ」と「あるエッジと関係のないクラスタマップ」について述べる．フリーノード f の所属クラスタの集合 $C_b(f)$ の要素によって形成されるクラスタマップの集合を f と関係のあるクラスタマップ群 $M_b(f)$ とする．「あるフリーノードと関係のないクラスタマップ」とは， $M_b(f)$ 以外のすべてのクラスタマップのことである．すべてのクラスタマップの集合を M としたとき，フリーノード f と関係のないクラスタマップの集合は $M - M_b(f)$ と定義する．また，エッジ e に接続するフリーノードを $f(e)$ とするとき「あるエッジと関係のないクラスタマップ」は $M - M_b(f(e))$ と定義する．

これを元に，E1 と E2 を定義する．あるクラスタマップもしくはノードを n としたとき， n の位置を $P(n)$ ，クラスタマップ m の半径を $R(m)$ とすると，E1 は以下のように定義される．

$$\text{minimize } \sum_{f \in F} \sum_{m \in M - M_b(f)} b \left(\text{ただし } \begin{cases} b = 1 & |P(f) - P(m)| < R(m) \\ b = 0 & \text{else} \end{cases} \right) \quad (4.1)$$

エッジ e の両端のノードを端点とする線分とクラスタマップ m が交差する場合に交差した部分の線分の長さを返し，交差しない場合は 0 を返す関数を $\text{IntersectLength}(e, m)$ としたとき，E2 は以下のように定義される．

$$\text{minimize } \sum_{e \in E} \sum_{m \in M - M_b(f)} \text{IntersectLength}(e, m) \quad (4.2)$$

E3 は Misue のアンカーマップでも用いられているが，そこでの定義はアンカーがひとつの円周上に配置されていることを前提に定式化されているため，本研究では新たに定義を行った．アンカー a と接続するフリーノードの集合を $F''(a)$ としたとき，E3 は以下のように定義する．

$$\text{minimize } \sum_{a, a' \in A, a \neq a'} |P(a) - P(a')| |F''(a) \cap F''(a')| \quad (4.3)$$

E4 は，アンカーマップで得られる 2 部グラフの大まかな傾向提示をできるだけ維持することを目的とした基準である．アンカーマップでは，フリーノードの分布を見ることによって 2 部グラフの大まかな傾向を読み取ることができる．そこで，本研究では，子クラスタマップがないアンカーマップと子クラスタマップのある階層型アンカーマップとでフリーノードの変化量をすくなくすることを 2 部グラフの概観の維持としてこの美的基準を定義する．元となるアンカーマップのことをベースアンカーマップと呼び，同じクラスタに属するアンカー

同士が隣接して配置されるアンカーマップとする．ベースアンカーマップ上でのノード n の位置を $P_0(n)$ としたとき，E4 は以下のように定義する．

$$\text{minimize } \sum_{f \in F} |P_0(f) - P(f)| \quad (4.4)$$

最後に，E5 を以下のように定義する．

$$\text{minimize } \frac{\max_{a_1 \in A, a_2 \in A, a_1 \neq a_2} |a_1 - a_2|}{\min_{a_1 \in A, a_2 \in A, a_1 \neq a_2} |a_1 - a_2|} \quad (4.5)$$

第5章 ノードの配置手法

5.1 技術的な課題

Misue のアンカーマップでは，ひとつの円周上でのアンカーの並び順のみを決定すればよい．しかし，階層型アンカーマップでアンカーの位置を決定するためには，並び順のほかにクラスタマップの位置，半径，向きを求める必要がある．本研究では，この3つの決定方法の開発を新たに行なった．

クラスタマップの位置，半径は，美的基準のE4，E5に影響を与えられられる．これに関しては，4種類の配置スタイルを考案，開発し，検討を行うこととした．図5.1は，各配置スタイルについての説明である．

[A] 外接スタイル 子クラスタマップを親クラスタマップに外接させるスタイルである．クラスタマップの大きさに関しては，子クラスタマップと親クラスタマップ上でのアンカー間の距離が同じになるようにした．

[B] 内接スタイル balloon layout スタイル [20] を用い，子クラスタマップを親クラスタマップの内部に配置するスタイルである．この場合，E1を満足するために子クラスタマップの半径の設定に工夫が必要となると考えられる．これに関しては，5.4章で詳細に述べる．このスタイルは，クラスタマップがおおよそベースアンカーマップでのアンカーの位置の重心に配置されるため，美的基準E4をよく満足するスタイルと考えられる．

[C] 弦に接するスタイル 子クラスタマップをできるだけ親クラスタマップの内側に配置するスタイルである．このスタイルでは，子クラスタマップを親クラスタマップの弦に接するように配置し，半径に関しては，[A]と同様に子クラスタマップと親クラスタマップのアンカー間の距離が同じになるようにする．クラスタマップの大きさを保ったまま[A]よりも空間効率がよく，E5を最もよく満たす考えられるが，子クラスタマップが親クラスタマップの内側に入り込み，美的基準E4を十分に満足できない可能性がある．

[D] 円周上スタイル アンカー同様に子クラスタマップを親クラスタマップの円周上に配置するスタイルである．このスタイルは，空間効率は[C]に劣ると考えられるが，[C]よりもE4を満足するスタイルと考えられる．

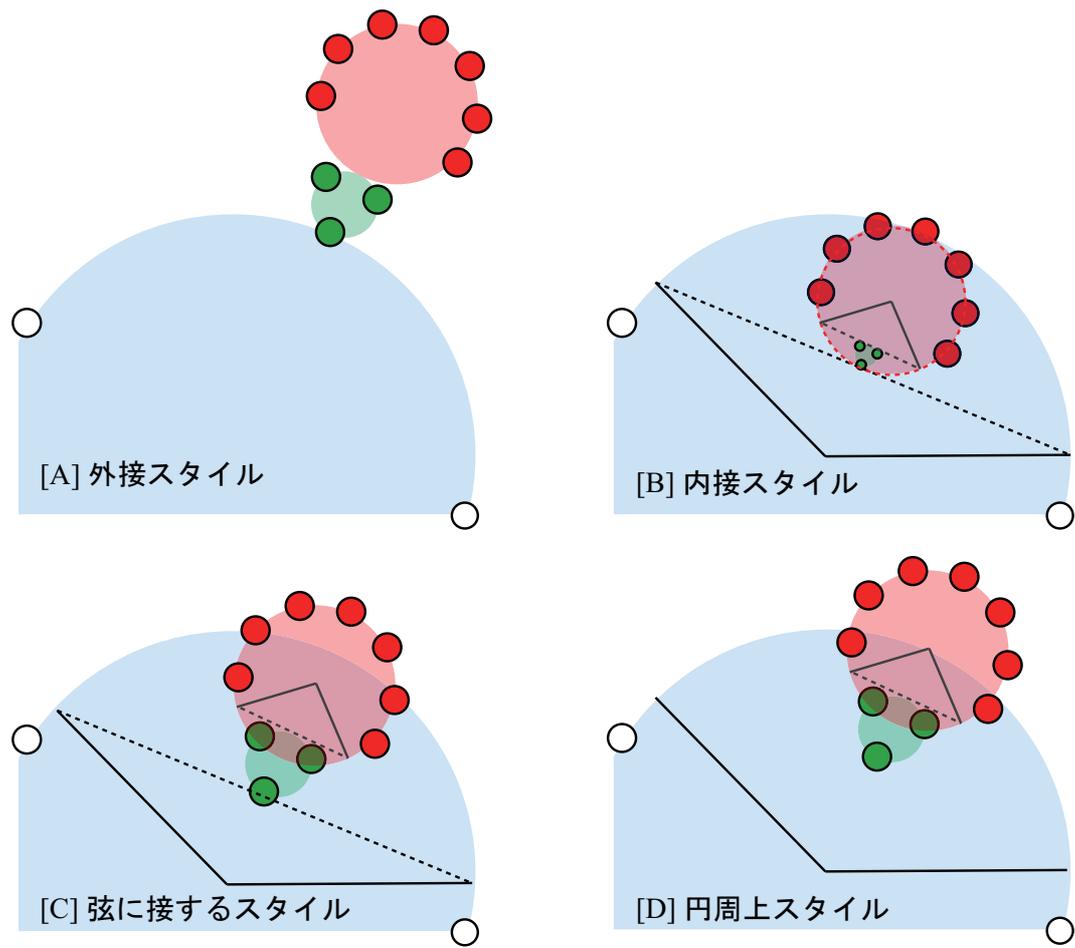


図 5.1: 4 種類の配置スタイル

クラスタマップの向きは、美的基準の E2 に大きく影響を与えられられる。図 5.2 に、向きの違う 2 種類のクラスタマップの描画例を示す。図 5.2(a) の場合、クラスタ間を接続するエッジの多くがクラスタマップと交差してしまい、視覚的な混雑を引き起こしてしまっているが、図 5.2(b) の場合、この混雑が軽減されていることが分かる。このように、クラスタマップの向きは描画結果の可読性に大きく影響を与えるため、できるだけ可読性の高い向きを求める必要がある。

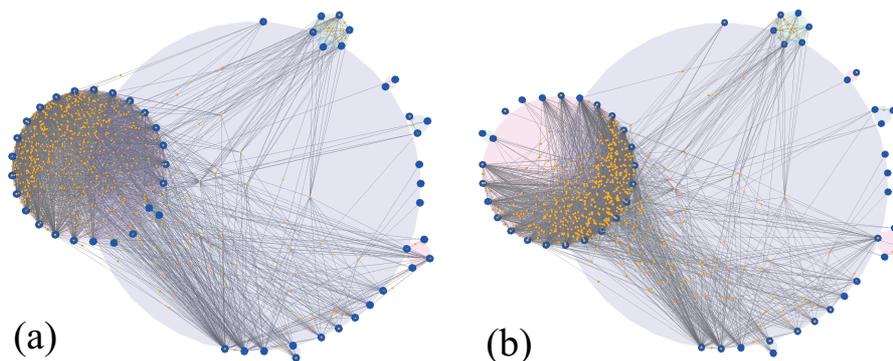


図 5.2: クラスタマップの向きによる可読性の違い

5.2 ノード配置の概要

階層型アンカーマップの配置手法では、アンカーの位置を決定した後、フリーノードの配置をおこなう。

まず、アンカーの位置を以下の手順で求める。

Step1 各クラスタマップ内での子要素の配置角度を求める

Step2 各クラスタマップの半径と親クラスタマップの中心からの初期距離を求める

Step3 各クラスタマップの向きを求める

Step4 クラスタマップの配置を行う

子要素の位置は、子要素の配置角度、クラスタマップの半径と向き、親クラスタマップとの距離によって再帰的に決定していく。配置角度に関しては、先に述べた 4 種類のスタイルによる違いはないが、クラスタマップの半径と親クラスタマップとの距離はスタイルによって異なる。また、向きもスタイルによって結果が異なる可能性がある。そのため、Step2 で初期距離を決めた後に向きを求め、その後求めた向きを元に最終的に距離を求める手順を取る。

4種類の配置スタイルでは、Step2とStep4で用いる関数に違いがあるものの、同じ手順を用いてレイアウトを求めることができる。

以上の手順でアンカーの配置を決定した後、フリーノードの配置を行う。

5.3 子要素の配置角度の決定

子要素の配置角度は、まず、Misueのアンカー並び順決定アルゴリズムによって配置順序を求めた後、子要素の含むアンカーの数に応じたスペースを割り当てていくことで決定する。

Misueのアルゴリズムを用いる際、子要素を一つのアンカーとして取り扱う。つまり、子クラスタマップも一つのアンカーと見なし順序を求める。また、クラスタ下で形成される局所的な関係性を提示するために、クラスタマップ内の要素につながったエッジのみを評価対象とする(図5.3)。

クラスタ c で形成されるクラスタマップ内の子要素の並び順を求めるとき、2部グラフ $G'_B(c) = (Ch(c) \cup F'(c), E_x(c))$ を Misue のアルゴリズムに入力する。 $E_x(c)$ は $E'(c)$ から生成される新しいエッジの集合で、 $e \in E'(c)$ が $Ch(c)$ の要素に接続していない場合、 $Ch(c)$ 内の祖先のノードに接続するものとしたエッジの集合である。ノード集合 $Ch(c)$ はアンカーと子クラスタの両方を含むが、クラスタは一つのアンカーとみなす。図3.1で示したグラフを例として挙げると、 c_2 によって形作られるクラスタマップ内の子要素の順序を求める場合、 $G'_B(c_2) = (\{c_3, a_5\} \cup \{f_3, f_4\}, \{(c_3, f_3), (a_5, f_3), (a_5, f_4)\})$ が Misue のアルゴリズムに入力される。

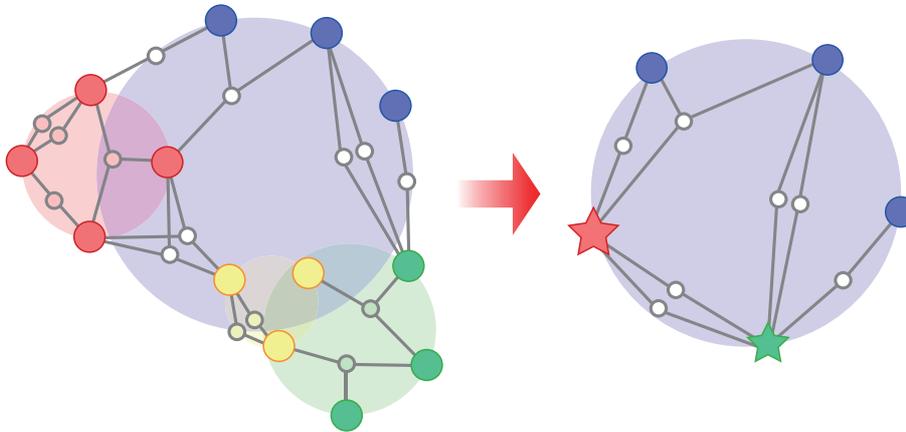


図 5.3: クラスタはひとつのアンカーとして考える

次に、図5.4に示すように、子クラスタマップの含むアンカーの数(リーフの数)に応じてスペースを割り当てる。クラスタ c で形成されるクラスタマップの子要素の配置角度を求めるとき、Misueのアルゴリズムによって得られた子要素の列を $W = \{w_0, w_1, w_2, \dots, w_h\}$ とすると、子要素 w_i に対して割り当てられるスペース $Space(w_i)$ と配置角度 $Angle(w_i)$ は以下のように求められる。

$$\text{Space}(w_i) = \begin{cases} \pi \frac{|A'(w_i)|}{|A'(c)|} & (w_i \in C) \\ \pi \frac{1}{|A'(c)|} & (w_i \in A) \end{cases} \quad (5.1)$$

$$\text{Angle}(w_i) = \sum_{j < i} \text{Space}(w_j) + \frac{1}{2} \text{Space}(w_i) \quad (5.2)$$

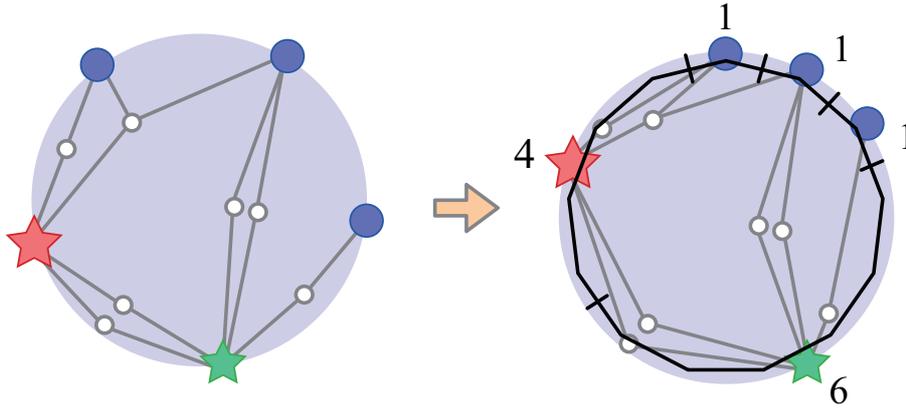


図 5.4: 子要素の配置角度

5.4 半径と初期距離の決定

このステップでは 4 種類の配置スタイルによって決め方が異なるため、スタイルごとに決定方法を述べる。ただし、以下に述べる方法はルートクラスタマップより下のマップのみに適応されるものである。

外接スタイル

外接スタイルの半径は、子クラスタマップと親クラスタマップとで、円弧上での隣り合うアンカーの距離が同じになるようにする。クラスタ c で形成されるクラスタマップを m_c 、 c の親クラスタ p で形成される m_c の親クラスタマップを m_p 、 m_p の半径を $R(m_p)$ としたとき、 m_c の半径 $R(m_c)$ は以下のように求められる。

$$R(m_c) = \frac{|A'(c)|}{|A'(p)|} R(m_p) \quad (5.3)$$

クラスタマップ m_c とその親クラスタマップ m_p 間の初期距離 $D_a(m_c)$ は以下のように求められる。

$$D_a(m_c) = R(m_p) + R(m_c) \quad (5.4)$$

内接スタイル

内接スタイルの半径は，外接スタイルと同様に半径を求めた場合，子クラスタマップ内に関係のないフリーノードがレイアウトされる可能性があるため，美的基準の E1 を満足できなくなる (図 5.5 左) .

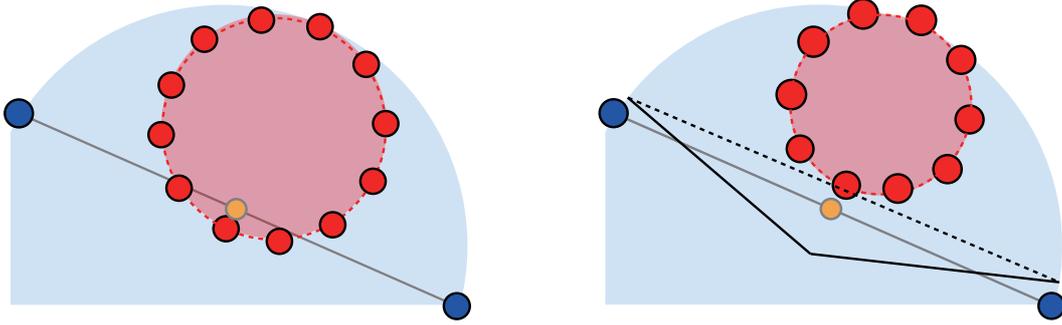


図 5.5: 内接スタイル

そこで，この配置スタイルでは，子クラスタマップの直径を円周から弦までの長さとした (図 5.5 右) . 内接スタイルの m_c の半径 $R(m_c)$ は以下のように求められる .

$$R(m_c) = \frac{1}{2} \left(1 - \cos \left(\frac{|A'(c)|}{|A'(p)|} \pi \right) \right) R(m_p) \quad (5.5)$$

そして，クラスタマップ m_c とその親クラスタマップ m_p 間の初期距離 $D_b(m_c)$ は以下のように求められる .

$$D_b(m_c) = R(m_p) - R(m_c) \quad (5.6)$$

弦に接するスタイル

子クラスタマップは図 5.5 の弦のところまで近づけることができると考えられ，これを基準としたものが弦に接するスタイルである . 半径は外接スタイルと同じものを用いるため，式 5.3 により求められる . また，クラスタマップ m_c とその親クラスタマップ m_p 間の初期距離 $D_c(m_c)$ は以下のように求められる .

$$D_c(m_c) = \cos \left(\frac{|A'(c)|}{|A'(p)|} \pi \right) R(m_p) + R(m_c) \quad (5.7)$$

円周上スタイル

円周上スタイルの半径も外接スタイルと同じものを用いるため、式 5.3 により求められる。初期距離に関しては、基本的には円周上に配置するような親クラスタマップとの距離を用いるが、そのままでは内接スタイルのときと同様、子クラスタマップ内にフリーノードがレイアウトされる可能性があるため、そのような場合には弦に接するスタイルと同おなじように求める。クラスタマップ m_c とその親クラスタマップ m_p 間の初期距離 $D_d(m_c)$ は以下のように求められる。

$$D_d(m_c) = \begin{cases} R(m_p) & (D_c(m_c) < R(m_p)) \\ D_c(m_c) & (D_c(m_c) \geq R(m_p)) \end{cases} \quad (5.8)$$

アンカーの初期距離

アンカーは、配置スタイルに関わらず親クラスタマップの円周上に配置する。アンカー a とその親クラスタマップ m_p 間の初期距離 $D(a)$ は以下のように求められる。

$$D(a) = R(m_p) \quad (5.9)$$

5.5 クラスタマップの向き決定

初期距離をベースとして、E2 を満足するようなクラスタマップの向きを求める。E2 を指標とした場合、あるクラスタマップの最適な向きは他のクラスタマップの向きの影響を受ける。そのため、E2 を最小化するクラスタマップの向きを求めるにはこの影響を考慮する必要がある。しかし、例えば、クラスタマップの数を n 、クラスタマップ一つあたりのパターン数を u としたとき、 u^n パターンを探索する必要があり、多くの計算時間を要する。

本研究では他のクラスタマップの向きの影響は全体の可読性には影響が少ないと考え、この影響を無視した計算量の少ない手法の開発を行った。他のクラスタマップの影響を無視することで、全体で un パターンの探索で済む。

クラスタマップの向きは、ルートクラスタマップから下のクラスタマップへと順に向きを決定していく。この際、向きを求めているクラスタマップとその祖先のクラスタマップは Step2 で求めた半径と初期距離を用いて配置する。それ以外のクラスタマップ（向きを求めているクラスタマップの兄弟、子孫のクラスタマップ）は半径を 0 として配置を行う。半径を 0 にすることで同じ深さのクラスタマップ間での探索順序を考慮する必要がなくなる。図 5.6 は、緑のクラスタマップの向きを求める際の状態を示している。緑とその親の青いクラスタマップの半径、初期位置は Step2 で求めたものを用い、兄弟の赤いクラスタマップと子の黄色いクラスタマップの半径は 0 としている。また、赤と黄色のクラスタマップの含むアンカーは親の中心に配置する。

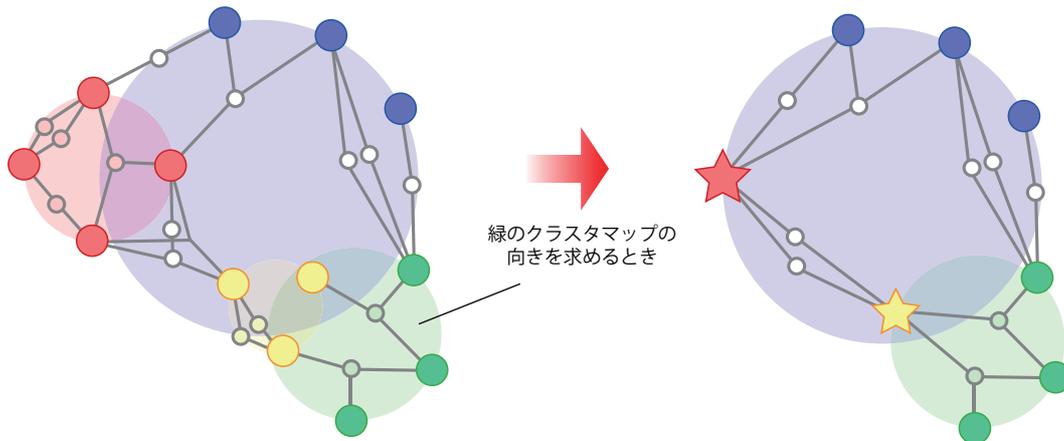


図 5.6: 向きを求める際の兄弟クラスタマップの状態

次に、クラスタマップごとの向きの求め方について述べる。

まず、向きを求める際に用いる指標の定義を行う。クラスタマップの向きは E2 をできるだけ満足するものを求めるため、E2 をベースとして指標の定義を行った。

クラスタ c で形成されるクラスタマップを m_c 、 m_c の子クラスタマップの集合を $M_c(m_c)$ とするとき、指標 I は以下の式によって求められる。ただし、すべてのアンカーは図 5.6 のように配置されるが、関数 IntersectLength では Step2 の求めた半径を使って計算を行う。また、フリーノードを 5.7 節で述べる仮想的なアンカーの位置の重心に配置する。

$$I = \sum_{e \in E} \sum_{m \in \{m_c\} \cup M_c(m_c)} \text{IntersectLength}(e, m) \quad (5.10)$$

図 5.7 は指標 I の計算方法を説明したものである。緑のクラスタマップの向きを求めるとき、図の赤い部分の長さの合計が指標 I となる。向きを求めるクラスタマップの子の半径を考慮しない場合を試したところ、子クラスタマップ多くのエッジが交差し、E2 の評価値と可読性が落ちてしまったため、子の半径を考慮するようにした。また、子の向きが確定していないため孫の位置は定まっておらず、孫との交差は計算できない。そのため、考慮するのは子の半径までとした。

この指標 I の値をできるだけ小さくするような向きを求める。これには、2 種類の方法を開発した。

はじめに開発した方法は、クラスタマップを少しずつ回転させながら I を計算し、1 周分探索したらクラスタマップを裏返して（子要素の並びを逆順にして）もう 1 周回転させ、 I が最も小さかった向きを採用する方法である。回転させる角度を $2\pi/|A'(c)|$ (c は向きを求めるクラスタマップの形成元のクラスタ) とした場合、クラスタマップ一つあたり $|A'(c)| \times 2$ パターンを探索することで向きを求めることができる。

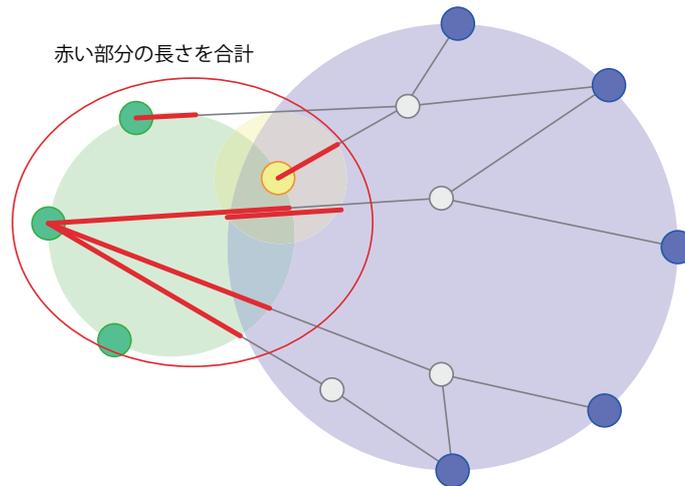


図 5.7: 指標の計算方法

次に、先の方法をさらに最適化し、バイナリサーチのように向きを探索する方法を開発した。最初に、ある向きと π 違う向きの 2 つの向きについて I を計算し、 I の小さい方の向きを選ぶ。この向きを d_0 とする。次に、 $d_0 \pm \pi/2$ の 2 つの向きについて I を計算し、 $d_0, d_0 + \pi/2, d_0 - \pi/2$ の中で最も I の小さい向きを選び、これを d_1 とする。さらに、 $d_1 \pm \pi/4$ の 2 つの向きについて I を計算し、 $d_1, d_1 + \pi/4, d_1 - \pi/4$ の中で最も I の小さい向きを選ぶ。このように、選んだ向きからの幅を半分にしていきながら探索していき、この幅が閾値（例えば、 $\pi/|A'(c)|$ ）より小さくなった時点で探索を打ち切る。そして、先の方法と同様に子要素の並びを逆順にして再度探索をおこない、 I が最も小さかった向きを採用する。この方法は、先の方法と比べて精度が落ちる可能性があるものの、クラスタマップ一つあたり $\log |A'(c)| \times 2$ パターンの探索回数で済む。

バイナリサーチ式の探索アルゴリズムを Algorithm 1 に示す。アルゴリズム中の `Direction` は、クラスタ c で形成されるクラスタマップの向きを示し、`Index()` はその向きでの I の値を返す関数、`ReverseChildren()` は子要素の並び順を逆にする関数であり、 $PI = \pi$ 、 $AnchorNum = |A'(c)|$ とする。

Algorithm 1

```
float[] minE5 = {float.MaxValue, float.MaxValue};
float[] minDir = {0,0};

for (int rv = 0; rv < 2; rv++) {
    float d = PI;
    while (d > PI / AnchorNum) {
        float f0 = minDir[rv] - d;
        Direction = f0;
    }
}
```

```

    p0 = Index();

    float f1 = minDir[rv] + d;
    Direction = f1;
    p1 = Index();

    if ( p0 < minlen || p1 < minlen)
    {
        if (p0 < p1){
            minDir[rv] = f0;
            minE5[rv] = p0;
        } else {
            minDir[rv] = f1;
            minE5[rv] = p1;
        }
    }
    d /= 2;
}
if (rv == 0) ReverseChildren();
}

if (minlen[0] < minlen[1]) {
    ReverseChildren();
    Direction = minDir[0];
} else {
    Direction = minDir[1];
}

```

5.6 クラスタマップの配置

これまでで求めた，親との初期距離，半径，向き，配置角度を用いて最終的なクラスタマップの位置を求める．しかし，外接スタイル・弦に接するスタイル・円周上スタイルの3つの配置スタイルでは，初期距離をそのまま適応すると孫クラスタマップが親クラスタマップの内部に入り込みすぎてしまう可能性がある．そこで，基準を満たすまで子クラスタマップを少しずつ遠くにずらしていくアルゴリズムを用いて位置の修正をおこなう．アルゴリズムで用いる基準はスタイルによって異なる．

まず，位置修正のアルゴリズムを Algorithm 2 に示す．アルゴリズム中の $Rule(m)$ はクラス

タマップ m のスタイルの基準値を返す関数, `LayoutAnchor()` はアンカーの配置を行う関数, `GetChildClusters(m)` はの子クラスタマップを取得する関数, `Dist(m)` は m の親からの距離, `R(m)` は m の半径, `root` はルートクラスタマップである.

Algorithm 2

```
void UpDistance(){
    bool intersect;
    do
    {
        intersect = StepDistance(root);
        LayoutAnchor();

    } while (intersect);
}
bool StepDistance(parent)
{
    bool intersectCircle = false;

    foreach (child in GetChildClusters(parent))
    {
        intersectCircle |= StepDistance(child);

        float d0 = 0;

        foreach (g in GetChildClusters(child))
        {
            float d = Dist(g);
            float l = Rule(child) + R(g);

            if (d < l)
            {
                float df = l - d;
                if (df > ddd)
                {
                    ddd = df;
                }
                if (df > 0)
                {
```

```

        intersectCircle = true;
    }
}
}
P(child) += d0;
}
return intersectCircle;
}

```

次に、アルゴリズム内で利用する $\text{Rule}(m)$ の返す値を定義する。クラスタ c によって形成されるクラスタマップを m_c 、 c の親クラスタ p によって形成される親クラスタマップを m_p としたとき、各配置スタイルの $\text{Rule}(m)$ の返す値 $\text{Rule}(m_c)$ は以下のように定義する。

外接スタイル

$$\text{Rule}(m_c) = \text{R}(m_p) \quad (5.11)$$

弦に接するスタイル

$$\text{Rule}(m_c) = \text{R}(m_p) \cos\left(\pi \frac{|A'(c)|}{|A'(p)|}\right) \quad (5.12)$$

円周上スタイル

$$\text{Rule}(m_c) = \begin{cases} \text{R}(m_p) \cos\left(\pi \frac{|A'(c)|}{|A'(p)|}\right) & (b > 0) \\ \text{R}(m_p) - \text{R}(m_c) & (b \leq 0) \end{cases} \quad (5.13)$$

ただし、 $b = \text{R}(m_p) \cos\left(\pi \frac{|A'(c)|}{|A'(p)|}\right) - (\text{R}(m_p) - \text{R}(m_c))$ とする。

5.7 フリーノードの配置

Misue のアンカーマップでは、フリーノードの配置にスプリングモデル等の力指向アルゴリズム [2, 3] を用いる。この手法によって、フリーノードはアンカーとの接続関係において適切な位置に配置されるとともに、関係の近いフリーノード同士がかたまりを形成するような配置が得られる。階層型アンカーマップにおいても、アンカーマップと同様にフリーノードをアンカーやクラスタマップとの接続関係において適当な位置に配置することを目標とするが、単純にスプリングモデルを適用しただけでは次のような問題が起きる。

スプリングモデルを用いた場合、フリーノードは接続しているアンカーの重心周辺に配置される。そのため、図 5.8 のように子クラスタマップと多く接続しているようなフリーノードは子クラスタマップに強く引っ張られ内部に配置されてしまう可能性がある。これは最も重要視する美的基準 E1 に反する。

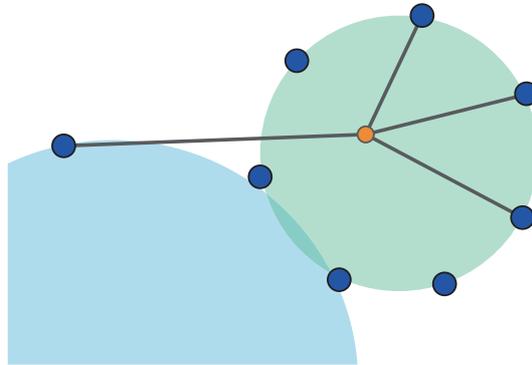


図 5.8: 単純にスプリングモデルを適用した際に起こる問題点

そこで、アンカーを仮想的に親クラスタマップの方へと移動させその座標を用いたスプリングモデルにより配置を行う (図 5.9)。これにより、フリーノードが子クラスタマップ内へ配置されることを避けると同時に、Req.1 の「フリーノードとクラスタによって形作られる 2 部グラフ構造の提示」をより強調することができると考えられる。

フリーノードの最下所属クラスタの子ではないアンカーを、最下所属クラスタで形成されるクラスタマップの方へと仮想的に移動させる。フリーノードを f 、フリーノードの最下所属クラスタを c_{bf} 、アンカー a の移動に関するクラスタを c_{af} 、あるクラスタを元に形成されるクラスタマップ及びアンカー n の位置を $P(n)$ とする。アンカー a の移動に関するクラスタを c_{af} とは、 c_{bf} の子のうち子孫に a を含むクラスタのことで、これは必ず 1 つ存在する。また、 $D(c)$ と $R(c)$ をクラスタ c で形成されるクラスタマップの親との距離と半径とする。ただし、 $D(c)$ にはノード配置方法の Step2 で用いた式を利用する。このとき、アンカー a の仮想的な位置 P_v は以下のように定義される。

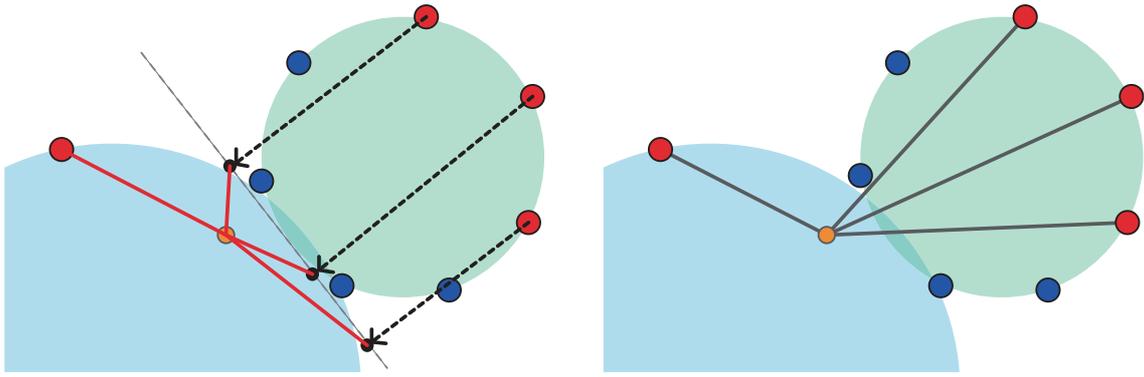


図 5.9: フリーノードのレイアウト方法

$$P_v = P(a) + \left((P(a) - P_c) \cdot n_c \right) n_c$$

$$P_c = P(c_{af}) + (D(c_{af}) - R(c_{af})) n_c \quad (5.14)$$

$$n_c = \frac{P(c_{bf}) - P(c_{af})}{|P(c_{bf}) - P(c_{af})|}$$

クラスタマップの向きを求める際に用いられるフリーノードの位置も同様の位置を用いて求められる。ただし、スプリングモデルを用いるのではなく、フリーノードと接続するアンカーの仮想的な位置の重心を用いる。

第6章 エッジの描画手法

6.1 曲線を用いたエッジ描画

エッジの描画にはローカルバンドルと円周配線の2つの方法を用いる。そのため、まず、そのエッジがどのクラスタを縦断しているかを調べ、それに応じてどちらの方法を用いるかを判断しながら描画をおこなっていく。

まず、エッジに接続しているアンカーの親クラスタと、接続しているフリーノードの最下所属クラスタを調べる。このとき、アンカーの親クラスタは必ずフリーノードの最下所属クラスタと同一かその子孫となっている。次に、アンカーの親クラスタからフリーノードの最下所属クラスタまで順にクラスタをたどっていき、クラスタによってローカルバンドルを行うか円周配線を行うかを判断しながら描画を行っていく。エッジ描画の手順を以下に示す。

1. アンカー a の親クラスタ c_a とフリーノード f の最下所属クラスタ c_{fu} を調べる
2. c_a から c_{fu} まで順にたどっていき、クラスタの列 $C_e = \{c_0, c_1, c_2, \dots, c_k\}$ を取得する
3. a の位置を開始点とする
4. c_0 から c_k までを

フリーノード f と接続する c_i の子要素が1つ以下

$i = 0$ 円周配線で描画

フリーノード f と接続する c_{i-1} の子要素が2つ以上 円周配線で描画

フリーノード f と接続する c_{i-1} の子要素が1つ以下 直線で描画

フリーノード f と接続する子要素が2つ以上 ローカルバンドルと円周配線で描画

5. c_k の終了点と f までを直線で描画する

フリーノードと接続する子要素とは、そのフリーノードと接続しているアンカーが接続しているアンカーを含むクラスタマップのことである。フリーノードと接続する子要素が1つ以下のクラスタが連続した場合、円周配線を行うとクラスタマップの周りをエッジが回り冗長となるため、次にローカルバンドルがあるまで直線による描画を行う。

6.2 円周配線

円周配線は、クラスタマップとエッジの交差を避けることと同時にクラスタ間の接続関係をより強調することを目的にしている。そのため、クラスタマップを迂回するエッジはクラスタマップよりも少し大きい半径の弧を描き、親クラスタマップに近いところで円周配線を終了するような描画を行う(図 6.1)。

円弧の半径は親クラスタマップから離れるほど大きくし、角度は離れるほど小さくする。これにより、円周配線同士の交差を避けることができる。クラスタ c で形成されるクラスタマップ m_c の円周配線を行うとき、 c の親クラスタ p の親クラスタマップを m_p 、クラスタマップ m の位置を $P(m)$ 、開始点を p_b とすると、円弧に用いる半径 R_a と角度 ArcAngle は以下のように定義される。ただし、 u, v は定数とする。

$$d = \arccos\left(\frac{(P(m_p) - P(m_c)) \cdot (p_b - P(m_c))}{|P(m_p) - P(m_c)| |p_b - P(m_c)|}\right) \quad (6.1)$$

$$R_a = R(m_c) + ud \quad (6.2)$$

$$\text{ArcAngle} = d - vd \quad (6.3)$$

図 6.1 は円弧の半径および角度の説明である。

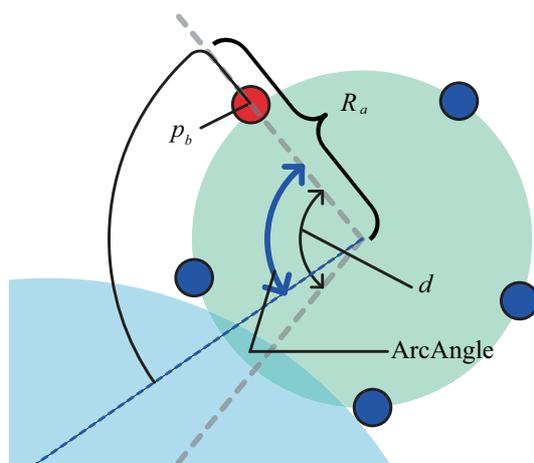


図 6.1: 円弧の半径と角度

円弧は、開始点から最も近いところから描き始める。これを円弧の開始点と呼ぶこととする。円周配線は、開始点から円弧の開始点周辺までを直線で描き、そこからベジェ曲線を用いて円弧の開始点までを描き、最後に円弧を描くという手順で描く。円弧の描画が終わった点が終了点となり、次のクラスタでの描画の開始点となる。また、同じ経路を通るエッジが重なって描画されないようにエッジをランダムに少し移動させて描画する。

6.3 ローカルバンドル

ローカルバンドルは、Req.2の「各クラスタ内で形成される2部グラフ構造の提示」を目的とした、ある子クラスタ内でのフリーノードの局所的な関係性を提示するためのエッジ描画方法である。フリーノードと接続する子要素が1つ以下の場合には、フリーノードがクラスタマップ内に配置されないため、2つ以上の場合のみローカルバンドルをおこなう。

ローカルバンドルでは、あるクラスタのみを取り出したとき、フリーノードが配置されるであろう位置の周辺をエッジが通るような描画をおこなう(図6.2(a))。フリーノードが配置されるであろう位置をローカル位置と呼ぶ。クラスタ c 内でのフリーノードのローカル位置は、フリーノードと接続しているアンカーのうちクラスタ c に含まれているアンカーのみを対象として5.7節で述べたアンカーの仮想位置の重心を用いる。

ローカルバンドルだけでは、ローカル位置以降の線がクラスタマップと交差して可読性が低くなる可能性があるため(図6.2(b))、ローカル位置と最も近い位置から円周配線をおこなう(図6.2(c))。

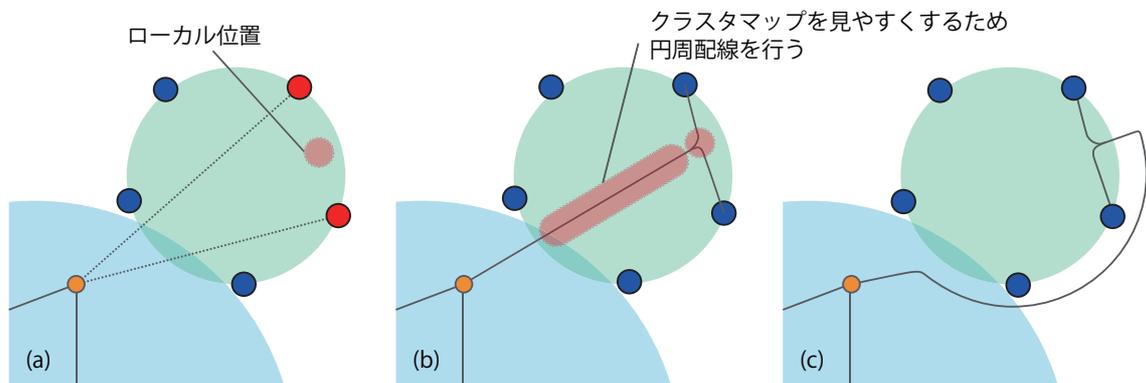


図 6.2: ローカルバンドルと円周配線の組み合わせ

ローカルバンドルでは、開始点からローカル位置周辺までは直線、ローカル位置周辺はベジェ曲線、ローカル位置周辺から円周配線の開始点までは再び直線を用いて描画をおこなう(図 6.3)。また、同じ経路を通るエッジが重なって描画されないようにエッジをランダムに少し移動させて描画する。

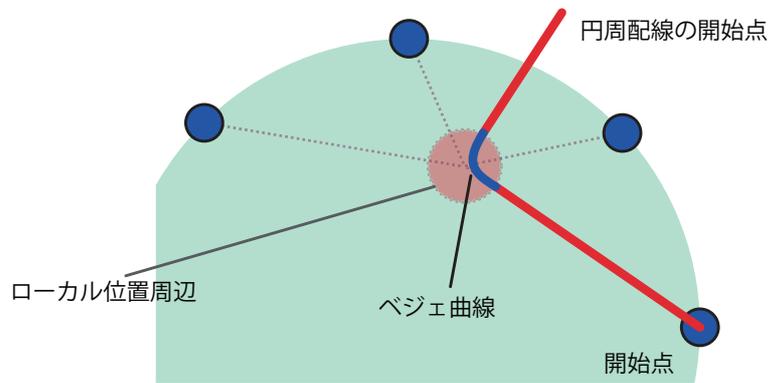


図 6.3: ローカルバンドルの描画手順

第7章 階層情報付加のためのインタフェース

7.1 インタラクティブなグラフ描画

本研究では、描画結果を観察しながら構造を整理していくことは閲覧者の情報分析の支援につながる [42] と考え、描画結果を観察しながら階層情報を付加していく用途を想定している。

Misue のアンカーの並び順決定アルゴリズムでは、同じフリーノードを共有するアンカー同士ができるだけ近くに配置されるような順序が求められる。この結果、関係の強いアンカー同士が集まり、ある種のアンカーのクラスタを形成する。アンカーの並びは1次元の配列のため、関係の強いアンカー同士が隣接している部分と、関係の弱いアンカー同士が隣接している部分が存在する。そのため、関係の弱い部分を切れ目として関係の強いアンカー同士が隣接している部分を取り出すことで、2部グラフに階層情報を付加していくことができると考えられる。しかしながら、このクラスタの切れ目は明確なものではなく一意に定めることは難しい。クラスタの切れ目を自動的に検出する技術の研究もなされている [43] が、本研究はこれとは異なり描画結果をもとに人の判断で切れ目を指定していくことで、閲覧者にとって適切な階層構造の付加が行えると考え、インタフェースを用いた階層情報付加のアプローチを取る。

階層情報を持たない2部グラフをアンカーマップによって描画し、その結果を見ながら階層情報を付加して階層型アンカーマップによって描き、その結果を見てさらに階層情報を付加していくような用途を想定したインタラクティブなグラフ描画への対応を行う。そのためには、階層情報付加のためのインタフェースと部分的なレイアウトの変更方法が必要となる。階層型アンカーマップのレイアウト方法は階層構造に関して再帰的にレイアウトを行うことができるため、部分的なレイアウト変更には容易に対応することができる。そのため、本章では階層情報付加のためのインタフェースについて述べる。

7.2 クラスタマップ化部分の指定インタフェース

描画されたアンカーマップ/階層型アンカーマップを観察することによって、閲覧者はおおよその切れ目の位置の検討をつけることができる考えられるが、アンカーの数が多い場合には適切な切れ目を正確に指定することは難しいと考えられる。そこで、閲覧者は切れ目としたい部分の範囲を指定し、システムがその範囲内で最も子要素（アンカー）間の関係の弱い部分を切れ目とするインタフェースを開発した。

閲覧者は、図 7.1 のように切れ目としたい部分のアンカーを左ドラッグで囲む。すると、囲まれたアンカーとその両側のアンカーで作られる間の中から次節で述べる方法で切れ目が決定される。

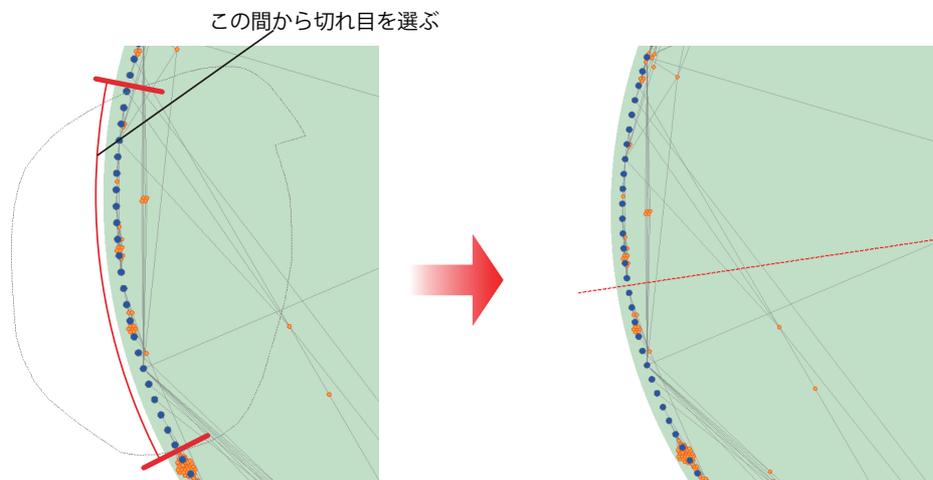


図 7.1: 自由曲線による切れ目指定

先の手法の場合、閲覧者は必ず 1 つ以上のアンカーを囲む必要がある。また、2 つ以上の切れ目の曖昧性が生じることとなり、1 箇所の間を指定することができない。そこで、閲覧者がひとつのアンカーも囲まなかった場合は、左ドラッグによって形成された閉曲線の重心に最も近いアンカーの間が切れ目となるようにした (図 7.2)。

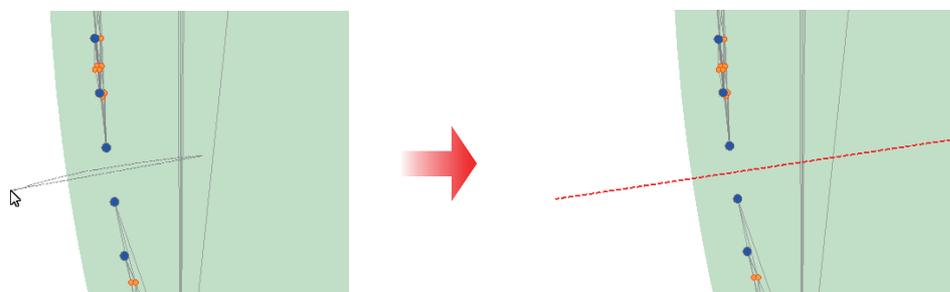


図 7.2: アンカーを囲まない切れ目指定

閲覧者が 2 つ以上の切れ目を指定すると、クラスタマップ化のアイコンが表示される (図 7.3)。このアイコンをクリックすると、切れ目に囲まれた部分がクラスタマップ化される。

7.3 関係の弱い部分の検出

関係の弱い部分を求めるために、Algorithm 3 によって隣り合う子要素間の関係の強さを計算する。アルゴリズム中の Children は配置順に子要素が格納されている配列、 $W(w, w')$ は子

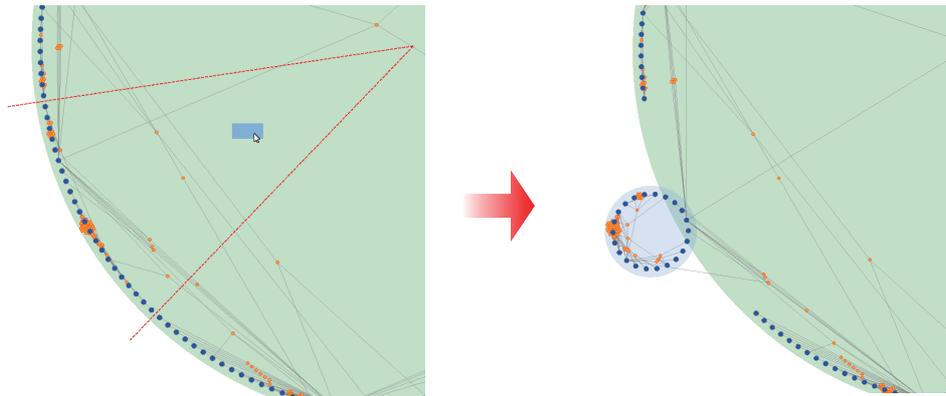


図 7.3: アイコンをクリックするとクラスタマップ化

要素 w と w' が共有するフリーノード数を返す関数, pow は定数とする. アルゴリズムが終了すると, 隣り合う子要素間の関係の強さを格納した配列が返される. この配列の i 番目の要素には, $Children[i]$ と $Children[i+1]$ の要素間の関係の強さが格納されている.

Algorithm 3

```
float[] ComputeRelationLevel()
{
    float[] relLevel = new float[|Children|];

    for (int j = 0; j < |Children|; j++)
    for (int k = j+1; k < |Children|; k++)
    {
        bool m = ture;
        int len = Angle(Children[k]) - Angle(Children[j]);
        if (len > 2*PI)
        {
            len = 2*PI - len;
            m = false;
        }
        if ( addLevel < 0) addLevel = 0;

        addLevel = Pow(addLevel, pow) * W(Children[k], Children[j]);

        if (m)
        {
```

```

        for (int i = j; i < k; i++)
        {
            relLevel[i] += addLevel;
        }
    }
    else
    {
        for (int i = k; i < |Children|; i++)
        {
            relLevel[i] += addLevel;
        }

        for (int i = 0; i < j; i++)
        {
            relLevel[i] += addLevel;
        }
    }
}

return relLevel;
}

```

求められた子要素間の関係のうち、指定された間の中で最も値の小さいところを切れ目とする。

第8章 描画手法の評価と考察

8.1 クラスタマップの配置スタイルの評価

4種類の配置スタイルに対し、美的基準のE4とE5を基準とした評価結果と描画結果を示し、考察を行う。

評価には実世界のデータから抽出した3つのグラフG1, G2, G3を用いる。G1は購入者と商品との関係を表す2部グラフである。データは研究室内で運営している買い物システムの2006年6月から2009年5月までの購買履歴である。商品をアンカー、購入者をフリーノードとしている。G2はWebページと訪問者を表すもので、著者のWebページの2009年9月1日～2009年9月7日までの7日間分のアクセスログから抽出したものである。G3もG2同様Webページのアクセスログであり、ソフトイーサ株式会社¹の一日のアクセスログのデータから抽出したグラフである。G2, G3はWebページをアンカー、訪問者をフリーノードとしている。評価に用いたグラフの規模を表8.1に示す。

表 8.1: 実験に用いたグラフの規模

グラフ名	データ	A	F	E	C	最大の階層の深さ
G1	購入者と商品	578	41	2303	27	2
G2	Webページと訪問者	59	1354	4986	18	2
G3	Webページと訪問者	445	2152	5524	112	6

3つのグラフには次のような特徴がある。G1は、フリーノードが多くのアンカーにつながっているグラフであり、ほとんどのフリーノードの最下所属クラスタがルートのクラスタとなっている。階層型アンカーマップでレイアウトすると、フリーノードの大半がルートクラスタマップの内部に配置される。また、フリーノード数に比べてアンカー数が圧倒的に多い。G2は、子クラスタマップ内にもフリーノードが配置される傾向にあるグラフであり、アンカーに比べてフリーノード数が多いグラフである。G3は他のグラフに比べて階層が深く、G2同様アンカーに比べてフリーノード数が多いグラフであるが、G1のようにクラスタ間を横断するようなフリーノードが多いグラフである。

¹<http://www.softether.co.jp/jp/>

8.1.1 概観の維持の評価

E4 を計算するに当たりベースアンカーマップが必要となる．この実験ではあらかじめ階層構造が与えられているグラフを用いるため，以下に述べる手法を用いてベースアンカーマップをレイアウトして評価を行った．

ベースアンカーマップのレイアウト方法

ベースアンカーマップのレイアウトでは，まず，階層型アンカーマップを配置し，次に，下のクラスタから順にアンカーを親のクラスタマップへ埋め込んでいき，最終的に一つ円周上にアンカーが並ぶアンカーマップを生成する．

親クラスタマップへの埋め込み方法は，子クラスタマップの一部を切り，親アンカーマップの中で割り当てられたスペースに順番に並べていくものである．切り込みを入れる部分は，親から最も遠い部分とした．図 8.1 にベースアンカーマップの生成手順を示す．

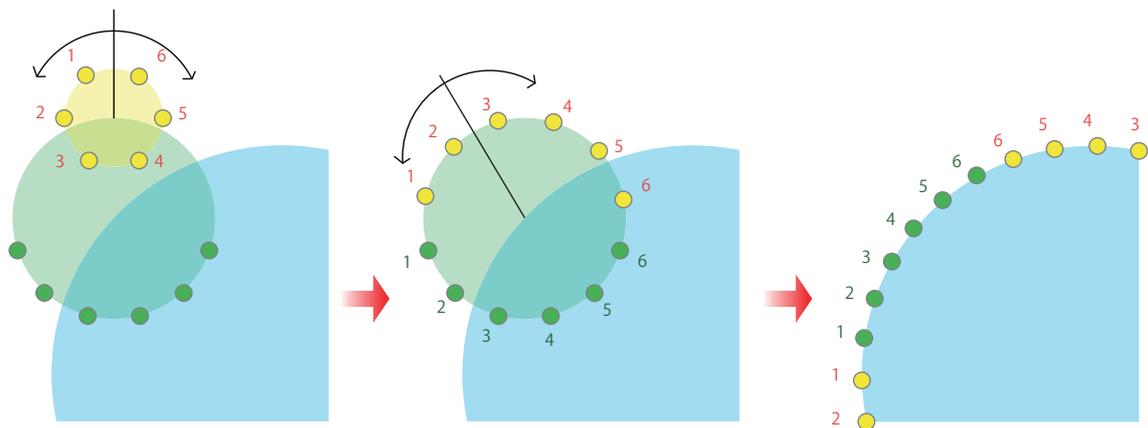


図 8.1: ベースアンカーマップの生成イメージ

評価結果

美的基準 E4 の評価結果を表 8.2 と図 8.2 に示す．表中の値は，ルートクラスタマップの半径を 1 として E4 を計算し，その値をフリーノード数で割ったものである．これはノードの移動量の平均値を表わしており，値が小さいほど良い結果である．

評価の結果，G1 では内接スタイルが，G2 では円周上スタイルが，G3 では弦に接するスタイルが最も良い値であった．値を比べると，G1 では 4 つのスタイルでほとんど差がなく，G2 では円周上スタイルが他の 3 スタイルよりも特に良い成績であった．一方，G3 は外接スタイル以外の 3 スタイルはほぼ同等の成績であった．

表 8.2: 概観の維持の評価結果

グラフ名	外接	内接	弦接	円周上
G1	0.146	<u>0.137</u>	0.148	0.144
G2	0.842	0.857	0.863	<u>0.774</u>
G3	0.347	0.281	<u>0.266</u>	0.268

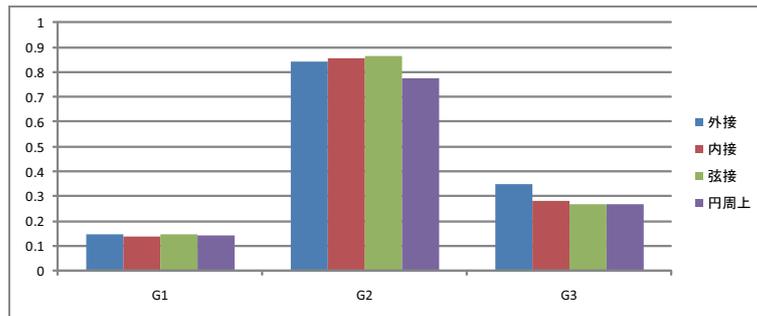


図 8.2: 概観の維持の評価結果

実験で得られた G1, G2, G3 のそれぞれの配置スタイル及びベースアンカーマップの描画結果を図 8.3 に示す。

8.1.2 空間効率の評価

空間効率の評価結果を表 8.3 と図 8.4 に示す。ただし、内接スタイルの評価結果は明らかに悪いので、図 8.4 では省略している。表中の値は、ルートクラスタマップの半径を 1 として美的基準 E5 を計算した値であり、値が小さいほど良い結果を示している。

評価の結果、弦に接するスタイルと円周上スタイルの結果が良く、内接スタイルが非常に悪い結果であった。

表 8.3: 空間効率の評価結果

グラフ名	外接	内接	弦に隣接	円周上
G1	336.792	7406.644	<u>276.187</u>	282.234
G2	45.925	352.907	<u>33.122</u>	38.635
G3	424.434	20059.202	579.548	<u>319.929</u>

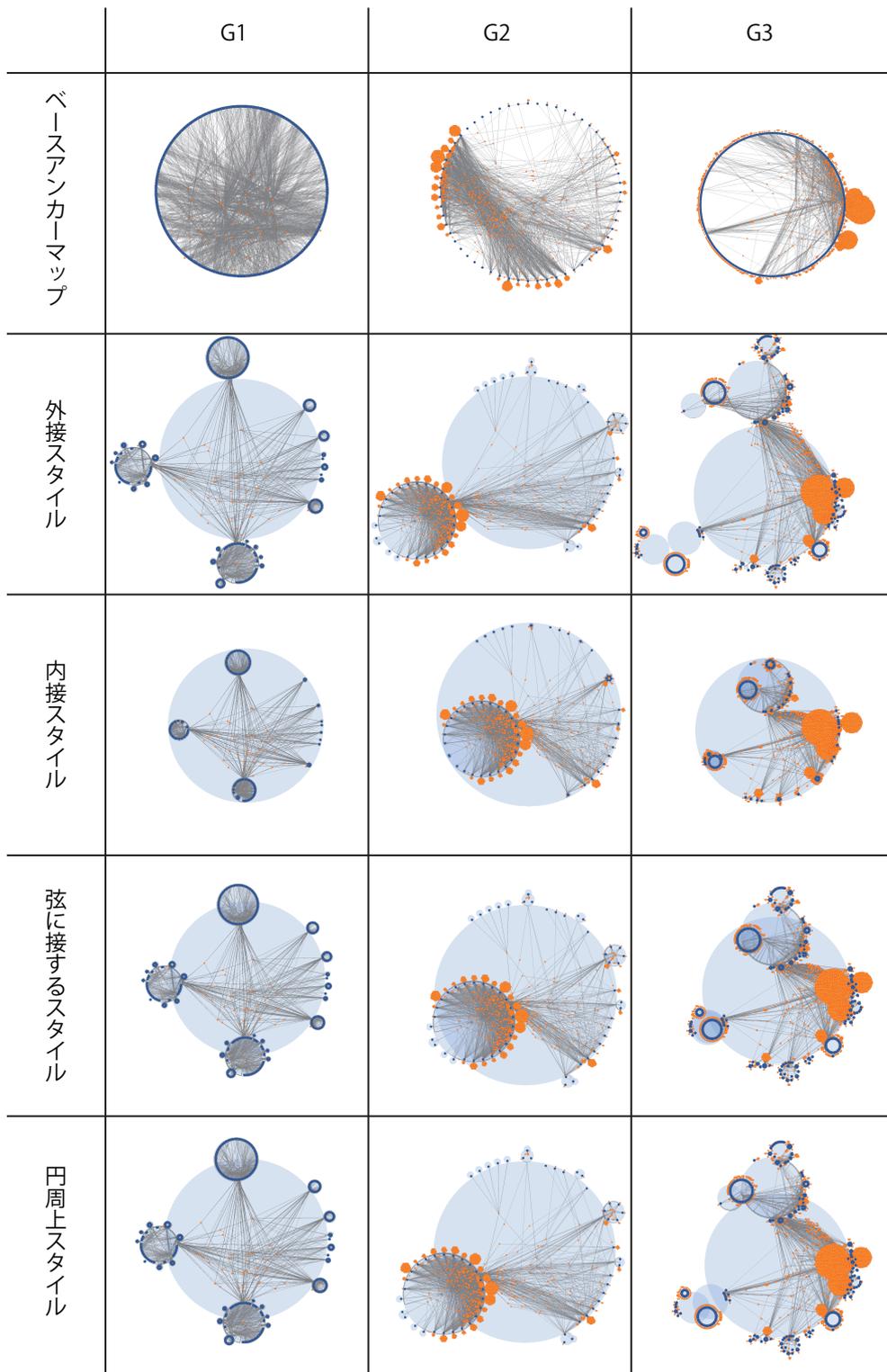


図 8.3: 配置スタイルに関する実験の描画結果

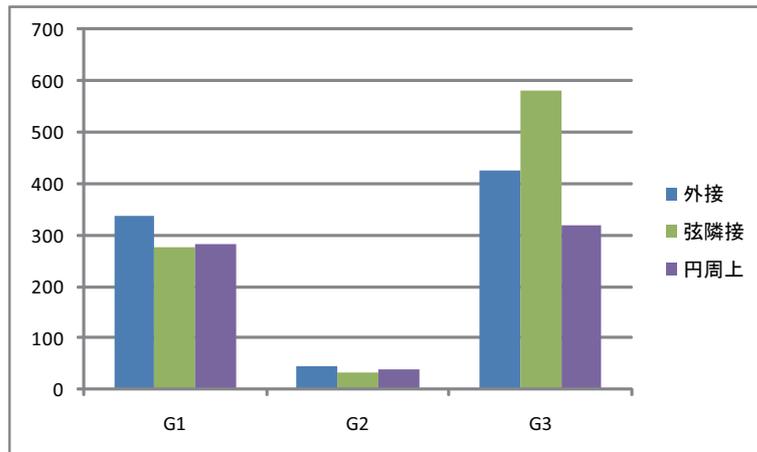


図 8.4: 空間効率の評価結果

8.1.3 考察

まず、美的基準 E4 の評価結果についての考察を述べる。G1 では内接スタイル、G2 では円周上スタイル、G3 では弦に接するスタイルが最も良く、グラフごとに全て別の結果となった。しかしながら、G1 では3つのスタイルの差は0.01以内、G3 では弦に接するスタイルと円周上スタイルの差は0.02と小さな差であった。描画結果を見ると、G1 と G3 は子クラスマップに配置されるフリーノードが少ないグラフであり、子クラスマップ間を接続するフリーノードが大半を占めている。このようなクラス間を横断するフリーノードが多いグラフでは、内接スタイル・円周上スタイル・弦に接するスタイルのどのスタイルを用いても美的基準 E4 については大きな差は生じないと考えられる。

一方、G2 では円周上スタイルが他のスタイルと比べて特に成績が良かった。G2 の描画結果を見ると、最も大きい子クラスマップ内に多くのフリーノードが配置されていることが分かる。そのような子クラスマップを内接スタイルや弦に接するスタイルで配置すると親の内部に張り込んでしまうため、成績が悪くなってしまったと考えられる。G2 のように多くのフリーノードが子クラスマップ内に配置されるようなグラフは、円周上スタイルを用いるのが適当であると考えられる。

次に、美的基準 E5 の評価結果についての考察を述べる。

4つの配置スタイルの中でも、内接スタイルは特に悪い結果となった。この原因は、内接スタイルでは階層が深くなるにしたがって半径が他のスタイルと比べて小さくなっていきアンカー間の距離が狭くなっていくためと考えられる。E5 では、アンカー間の最小距離を基準として空間効率を計算しているため、アンカー間の距離が狭くなる影響が顕著に出たと考えられる。

内接スタイル以外の3つのスタイルは、階層が深くなってもアンカー間が狭くなることはないため、外接スタイルが空間効率が悪く、円周上スタイルが中間、弦に接するスタイルが

最も良いという結果となったと考えられる。しかしながら，G3 では外接スタイルよりも悪い結果となっている。これは，図 8.5 に示すように，クラスタマップの大部分のアンカーが子クラスタマップを作る場合に子クラスタマップと親クラスタマップの位置がほぼ同じ位置に配置されてしまい，アンカーが近づいてしまうことが原因と考えられる。一方，円周上スタイルでは，子クラスタマップの中心が親クラスタマップの内部に配置されることはないため，弦に接するスタイルのような問題が起こらず良い結果となったと考えられる。G1 と G2 では，円周上スタイルと弦に接するスタイルの結果がほぼ同じであることを考えると，美的基準 E5 を指標とした場合，円周上スタイルが最も安定して利用できると考えられる。

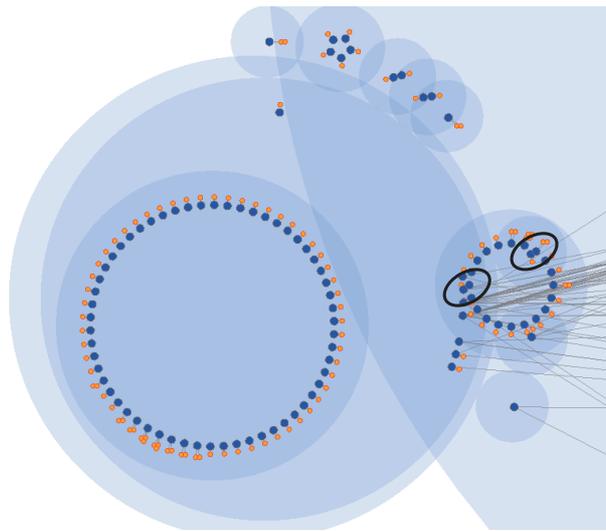


図 8.5: 弦に接するスタイルでアンカーが近づく例

実験の結果，円周上スタイルが美的基準を満足した上で安定的に利用できる配置スタイルであることが分かった。

8.2 クラスタマップの向き決定方法の評価

クラスタマップの向きの求め方は、美的基準の E2 をできるだけ満足することを目標とした方法である。E2 は他のクラスタマップの向きに影響を受けると考えられるが、開発した決定方法ではこの影響を考慮しないことで高速化を行っている。そこで、他のクラスタマップを考慮する探索方法と比べて開発した向きの決定方法が E2 をどの程度満たすかについての評価を行った。加えて、開発した 2 つの手法の精度の比較を行なった。

比較したのは、他のクラスタマップの向きに影響を考慮した方法と、開発した 2 つの方法の合計 3 つの方法である。便宜上、他のクラスタマップの向きに影響を考慮した方法を A 方法、クラスタマップを少しずつ回転させながら向きを探索する方法を B 方法、バイナリサーチのように向きを探索する方法を C 方法と呼ぶ。

A 方法は、 $2\pi/\text{アンカー数}$ の精度でアンカーが逆順の場合を考慮し、他のクラスタマップの向きも考慮したパターンの探索を行う方法である。そのため、 $\prod_{c \in Ch(c_0)} 2|A(c)|$ (c_0 はルートクラスタ) パターンの探索を行うこととなる。

評価には、6 種類のクラスタ 2 部グラフを用いた。図 8.6 に階層型アンカーマップを用いて描いた図を示す。この図は、子要素の並び順や半径・位置は開発した手法で求めたものだが、向きはランダムに割り当てたものである。

1 つ目は、ルートクラスタに 6 つの子クラスタがあり、子クラスタがそれぞれ 6 つのアンカーを持つ構造である (図 8.6(a))。このグラフは、子クラスタが均等に配置されている階層の深さが 1 の構造で、構造 1 と呼ぶ。

2 つ目は、ルートクラスタに 6 つのアンカーと 6 つの子クラスタがあり、子クラスタがそれぞれ 12, 8, 6, 4, 3, 3 個のアンカーを持つ構造である (図 8.6(b))。このグラフは、子クラスタが不均等に配置されている階層の深さが 1 の構造で、構造 2 と呼ぶ。

3 つ目は、ルートクラスタに 3 つのアンカーと 3 つの子クラスタがあり、各子クラスタが 3 個のアンカーと 3 つのアンカーを持つ子クラスタを 1 つ持つ構造である (図 8.6(c))。このグラフは、子クラスタが均等に配置されている階層の深さが 2 の構造で、構造 3 と呼ぶ。

4 つ目は、ルートクラスタに 3 つのアンカーと 3 つの子クラスタがあり、子クラスタのうち 1 つは 3 つのアンカーと 3 つのアンカーを持つ子クラスタを 1 つ持ち、1 つは 9 つのアンカーと 3 つのアンカーを持つ子クラスタを 2 つ持ち、残りの一つはアンカーを 4 つ持つ子クラスタである (図 8.6(d))。このグラフは、子クラスタが不均等に配置されている階層の深さが 2 の構造で、構造 4 と呼ぶ。

5 つ目は、ルートクラスタに 6 つのアンカーと 2 つの子クラスタがあり、深さ 1 のクラスタが 4 つのアンカーと 1 つの子クラスタ、深さ 2 のクラスタが 3 つアンカーと 1 つの子クラスタ、深さ 3 のクラスタが 3 つのアンカーを持つ構造である (図 8.6(e))。このグラフは、子クラスタが均等に配置されている階層の深さが 3 の構造で、構造 5 と呼ぶ。

6 つ目は、構造 4 のうち最も大きな子クラスタを、9 つのアンカーと 1 つの子クラスタを持ち、その子クラスタが 3 つのアンカーと 3 つのアンカーを持つ子クラスタを 1 つ持つクラスタと置き換えた構造である (図 8.6(f))。このグラフは、子クラスタが不均等に配置されている

階層の深さが3の構造で、構造6と呼ぶ。

この6種類の構造のアンカーに対し、次数2のフリーノードを40個、次数3のフリーノードを30個、次数4のフリーノードを20個、次数5のフリーノードを10個の合計100個のフリーノード、300本のエッジをランダムに追加し、各構造ごとに5つつつ、30個のクラスタ2部グラフを生成した。ただし、最低5つのアンカーはどのノードとも接続しないように制約した。

レイアウトは、ルートアンカーマップの大きさを1とし、円周上スタイルを用いて行った。

8.2.1 評価結果

4つのグラフについて向きを求め、E2を計算した結果を表8.4に示す。

実験の結果、構造1と2ではB・C方法ともにA方法と同程度かそれ以上の成績の向きの探索が行えた。A方法よりもC方法の方が良い理由は、C方法は $\pi/2^n$ 単位での探索しか行えず、A方法で行うパターンとは多少のずれがあるためである。深さが2以上となる構造3~6では構造1と2ほど良い結果は得られなかった。

次に、B、C方法がA方法で求めた全パターンの中でどの位の順位に位置するかを表8.5に示す。括弧内は、全順位の中で上位何%に位置しているかを示した値である。ただし、Aよりも良い結果であった場合は0と表記する。

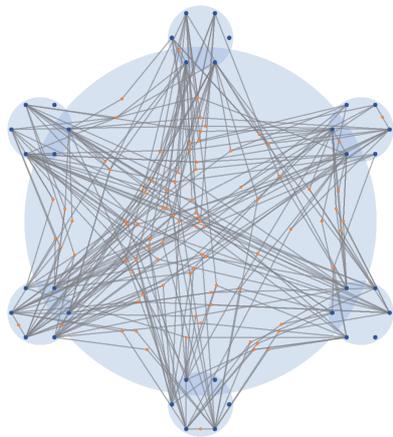
表より、両手法ともグラフで概ね上位1%程度、悪くとも5%以内の結果の探索が行えていることがわかる。特に階層の深さが1の構造1と2では、ほぼ最高の向きの探索が行えている。

最後に、図8.7と図8.8に各構造・各手法について、ひとつずつ描画結果を示す。構造1~4は表中で1番目のグラフの描画結果であり、構造5は成績の悪かった表中で2番目のグラフ、構造6はB方法・C方法ともに1%以下と成績の悪かった表中で3番目のグラフの描画結果である。

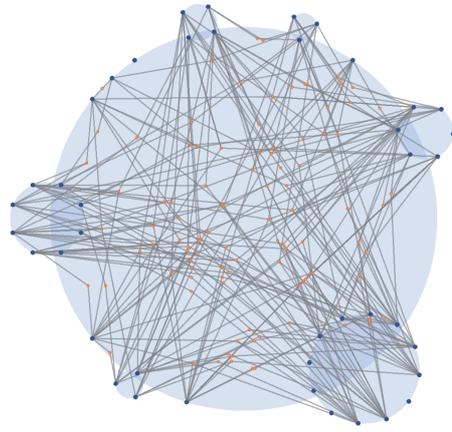
8.2.2 考察

評価の結果、開発した向きの決定方法はすべてのパターンを探索するのに比べて概ね上位1%程度の向きを探索できることが分かった。ランダムに向きを求めた図8.6と比べると、B方法、C方法ともに子クラスタマップとエッジの交差が少ないすっきりとした配置が得られたことが分かる。また、階層の深さが1の場合には、他のクラスタマップの向きの影響は非常に少ないことが分かった。

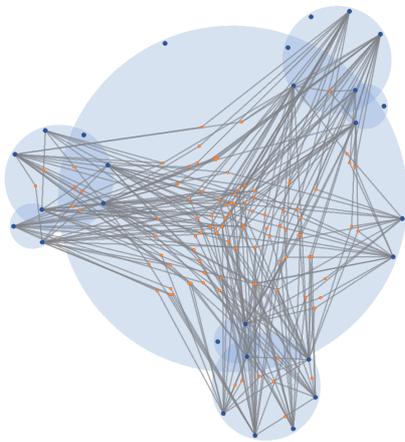
階層の深さ2以上となる構造3~6で、構造1・2よりも結果が悪くなった理由は、孫のクラスタマップにエッジが交差することを避けきれていないことが原因と考えられる。開発した方法では、兄弟のクラスタマップの半径を0としているが、最終的には半径を持つため、それに応じてフリーノードの位置やエッジの方向が変わる。これは孫のクラスタマップがない場合には影響はないが、孫のクラスタマップがある場合、半径が0のときには孫クラスタマッ



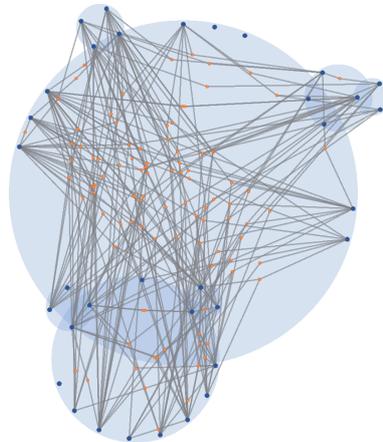
(a) 構造 1



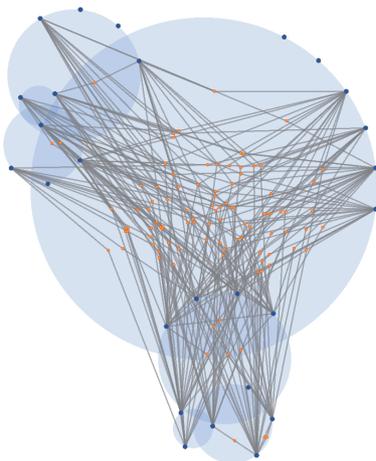
(b) 構造 2



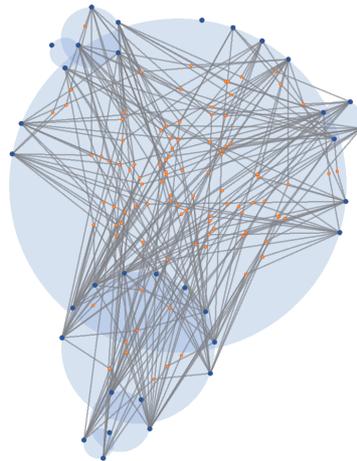
(c) 構造 3



(d) 構造 4



(e) 構造 5



(f) 構造 6

図 8.6: 向きの評価に用いたグラフの構造

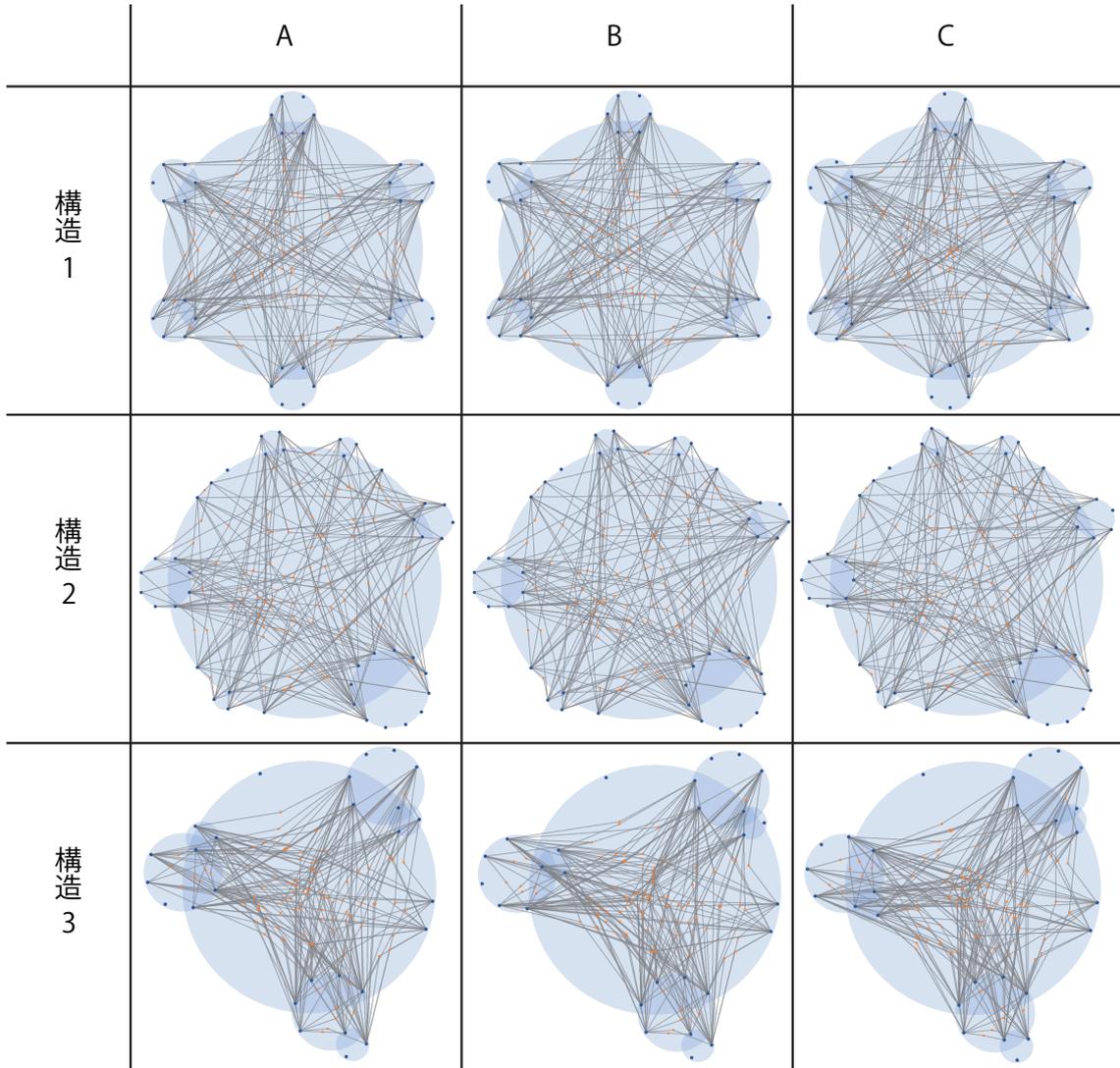


図 8.7: 向きに関する実験の描画結果 構造 1 ~ 3

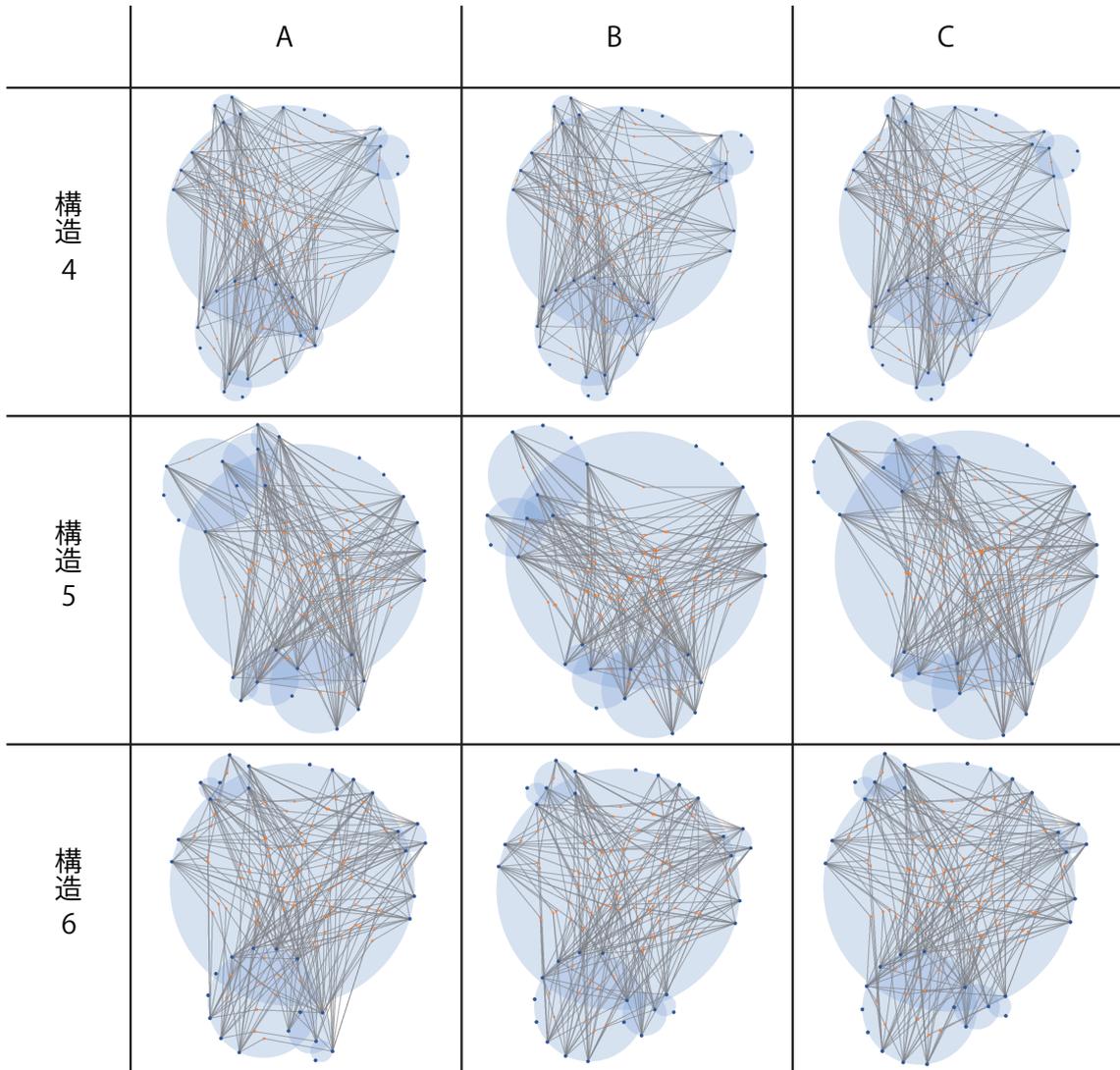


図 8.8: 向きに関する実験の描画結果 構造 4~6

ブと交差しなかったエッジが最終的な描画結果では少し方向が変わることで交差してしまい、結果が悪くなってしまったと考えられる。この点については、指標 I を計算する際に、エッジの方向が変わることを考慮し、子クラスタマップの半径を大きめに見積もることで改善できると考えている。

次に、B 方法と C 方法の結果について考察を行う。結論としては、C 方法は B 方法と比べて計算量が少なく精度も遜色のない方法であることが分かった。他のクラスタマップの影響が少ない構造 1 と 2 の結果、多少の違いはあるもののほぼ同じ精度での探索が行えていることから、一つのクラスタマップの向きを求めるには十分な精度を持っていることが分かる。グラフによって C 方法と B 方法で差が出る理由は、この 2 つの方法の違いよりも先に述べた半径を 0 とした探索の影響が大きいと考えられる。そのため、クラスタマップの向きを求める方法としては C 方法を用いるのが適切といえる。

表 8.4: エッジとクラスタマップ交差の評価結果

	A 方法	B 方法	C 方法
構造 1	21.49	21.49	21.56
	19.29	19.29	18.83
	21.77	21.9	21.49
	19.96	20.19	19.74
	20.38	20.45	19.84
構造 2	15.12	15.16	14.33
	24.97	25.04	24.64
	21.17	21.16	20.83
	17.97	18.20	16.61
	21.12	21.13	20.56
構造 3	42.21	46.22	46.30
	40.42	42.83	43.07
	39.24	46.07	44.27
	36.70	50.29	50.25
	41.56	49.35	49.06
構造 4	41.60	44.47	43.33
	30.29	36.35	31.54
	37.99	39.65	40.52
	38.82	41.92	44.88
	40.14	45.27	45.17
構造 5	50.22	61.10	64.37
	68.53	88.85	82.76
	51.80	72.79	66.28
	50.22	61.10	64.37
	45.71	55.41	59.04
構造 6	50.03	56.39	59.68
	40.48	44.88	44.38
	38.65	46.51	50.30
	29.44	29.52	32.20
	43.13	49.77	45.87

表 8.5: B 方法・C 方法の順位

	B 方法	C 方法
構造 1	1 (0.00)	2 (0.00)
	1 (0.00)	0 (0.00)
	4 (0.00)	0 (0.00)
	4 (0.00)	0 (0.00)
	2 (0.00)	0 (0.00)
構造 2	3 (0.00)	0 (0.00)
	4 (0.00)	0 (0.00)
	2 (0.00)	0 (0.00)
	57 (0.00)	0 (0.00)
	2 (0.00)	0 (0.00)
構造 3	543 (0.06)	585 (0.09)
	309 (0.03)	428 (0.05)
	7288 (0.82)	2194 (0.25)
	34946 (3.95)	34404 (3.88)
	6602 (0.75)	5801 (0.75)
構造 4	824 (0.09)	116 (0.01)
	7184 (0.81)	62 (0.01)
	254 (0.03)	1092 (0.12)
	430 (0.04)	4683 (0.53)
	9046 (1.02)	8405 (0.95)
構造 5	10779 (0.52)	26498 (1.28)
	113423 (5.47)	24562 (1.18)
	116701 (5.63)	24204 (1.17)
	10779 (0.52)	26498 (1.28)
	2895 (0.14)	10271 (0.50)
構造 6	6239 (0.51)	33250 (2.67)
	12659 (1.02)	7969 (0.64)
	19958 (1.60)	28506 (2.29)
	3 (0.00)	276 (0.02)
	14586 (1.12)	1166 (0.09)

第9章 応用例

9.1 階層情報を持つデータへの応用

まず、著者の Web サイトのアクセスログの描画例を示す。図 9.1(a) はアンカーマップを用いた描画例である。Web ページをアンカー、訪問者をフリーノードとして描いたものである。この図から、4.1 節で述べたような、アクセス統計などでは見ることの難しい複数ページにわたるサイトの閲覧傾向を見ることができる。

図 9.1(b) は同じデータを階層型アンカーマップを用いて描画したものである。アンカーの階層化には Web ページのディレクトリ構造を用いている。まず、図全体を見渡すことで、上の方のディレクトリ「/csg」内のページを見た訪問者、図右下の方のページ群（トップページや更新履歴のページ群）を見た訪問者、左下のディレクトリ「/vse」内のページを見た訪問者の 3 種類の訪問者がいることが分かる。また、これら 3 つのページ群のすべてを見た訪問者は多いものの、「/vse」と右下の方のページ群の 2 つを見た訪問者は少ないことが分かる。

次に「/vse」の分析を行うためにそのクラスタマップに着目する。「/vse」は著者の開発したフリーウェアのサイトが置かれているディレクトリである。このクラスタマップを観察すると、「/vse/」と「/vse/download.html」の 2 ページを見た訪問者、「/vse」と「/vse/manual.html」の 2 ページを見た訪問者、ディレクトリ内のほとんどのページを見た訪問者の 3 つのタイプが多いことが分かる。ローカルバンドルを見ることで、このディレクトリのページ以外のページを見ている訪問者は、ディレクトリ内のほとんどのページを巡回していることがわかる。これらの訪問者が「/vse」以外のどのようなページを見ていたかを調べるために、再び親クラスタマップを見る。すると、これらの訪問者のほとんどが他の多くのページを巡回していることが分かる。そこで、これらの訪問者の IP アドレスを調べたところ、そのほとんどがクローラーであったことが分かった。このように階層型アンカーマップを観察することで、「/vse」内のサイトの閲覧傾向や他のディレクトリとの関係性を分析することができる。

ここで着目した「/vse」をアンカーマップで描いた場合、図 9.1(a) の拡大図のように直線上にアンカーが配置されてしまい、先に述べたような特徴を読み取ることは難しいと考えられる。このことから、一部を円周上に並べる階層型アンカーマップのスタイルが従来手法に比べ局所的な関係性の分析支援に有効であることが分かった。

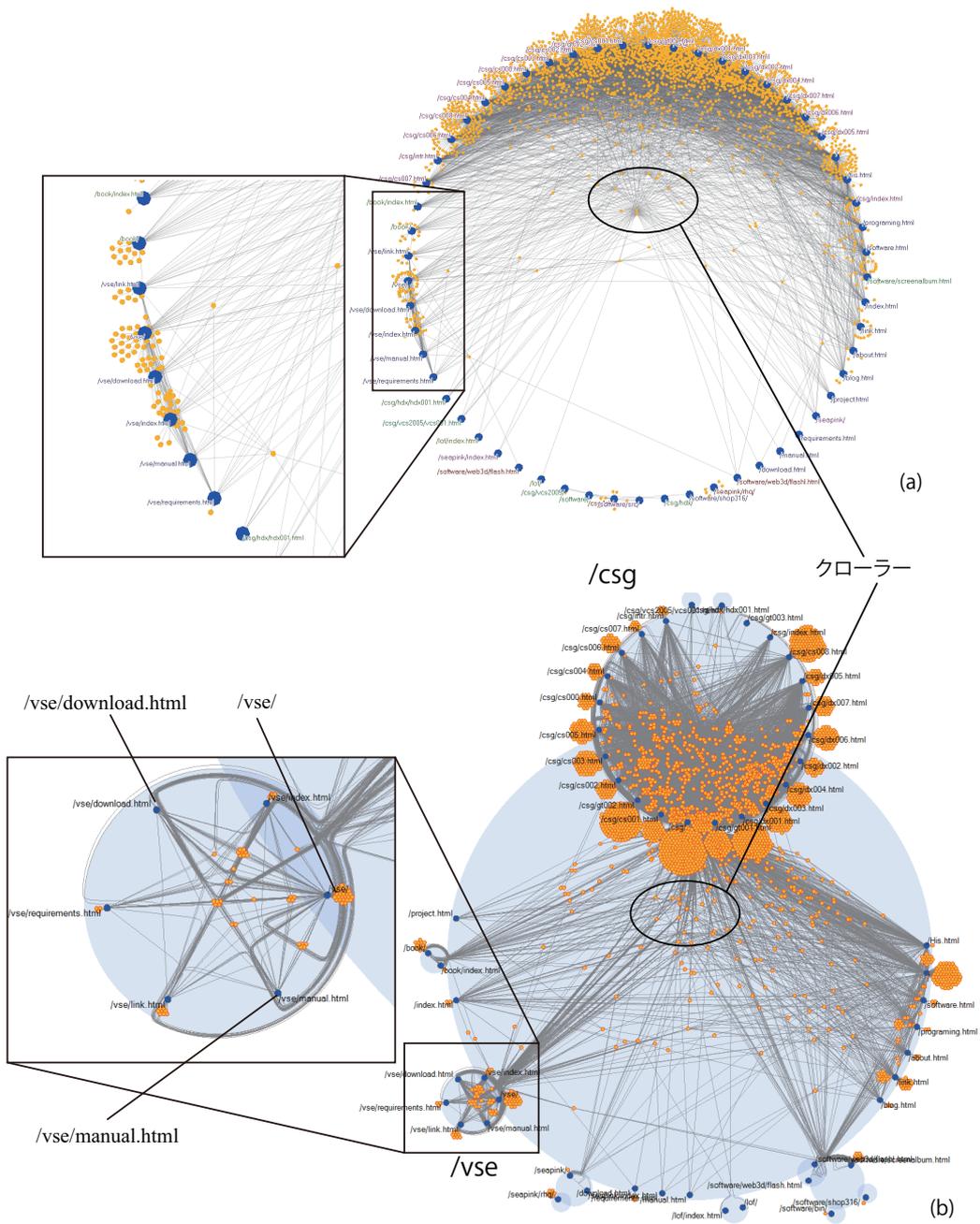


図 9.1: アクセスログの描画例

次の例は、商品と顧客との関係を可視化したものである。グラフは実験で用いたグラフ G2 である。これは実際の店舗のデータではないものの、実際の購買情報である。このデータには、9111 の売り上げ情報が記録されており、商品が 578 種類、購入者が 41 名いる。商品は「お茶」「お菓子」「清涼飲料」などの 11 種類のカテゴリに分けられている。さらに「お菓子」と「清涼飲料」は数が多いため「ポテトチップス」などの同じ種類の商品にカテゴリ分けされている。

このグラフを、アンカーマップを用いて描画したのが図 9.2 である。フリーノードの配置を行う際、購入した数に応じてスプリングモデルのばねの強さを強くしている。この図から、購入者がいくらかのクラスタに分かれていることが見える。しかし、フリーノードが非常に多くのアンカーに接続しているため、アンカーのラベルを読み取って購入者の傾向を読み取ることが困難である。

次に、同じデータを階層型アンカーマップで描いたのが図 9.3 である。まず、図全体を見渡し、3 つの購入者のグループ [A] [B] [C] に着目する。これらはそれぞれ赤、青、緑で色分けされており、そのフリーノードに接続しているエッジも同じ色で描いている。図全体を見渡すと、[B][C] の購入者は全てのカテゴリの商品を多く購入しているのに対し、[A] は清涼飲料水とお菓みに偏っている。そこでさらに、お菓子の子クラスタマップに注目する(図 9.3(a))。ここでは、[B] と [C] は同じような趣向を示しているが [A] の趣向は異なっている。例えば、[B] と [C] の購入者は「うまい棒」を好んで購入しているが、[A] の購入者は「うまい棒」をあまり購入していない。次に、清涼飲料水の子クラスタマップ(図 9.3(b))へ注目すると、先程とは異なり [A] と [B] は同じような趣向を示しているが [C] の趣向は異なっている。[A] と [B] は「紅茶」や「りんごジュース」を好んでいるが、[C] は「炭酸飲料」や「スポーツドリンク」を好んでいる。さらに、[C] に好まれている「炭酸飲料」に注目(図 9.3(c))すると、[C] と [A][B] では購入されている商品が異なっていることが分かる一方、「コカコーラ」は 3 グループの購入者に購入されていることも分かる。

以上の例から、階層情報を考慮した情報提示を行うことは、より詳細な情報分析の支援に有効であることが分かった。

9.2 階層情報を持たないデータへの応用

開発したインタフェースを用いて 2 部グラフの情報分析を行う例を示す。使用したデータは DBLP¹から「Kazuo Misue」の共著者とその共著者と論文を書いたことのある研究者を抽出したものである。図 9.4 は著者をアンカーとし、論文をフリーノードとしてアンカーマップで描いたものである。Misue の並び順決定アルゴリズムによってアンカーは並べられているため、同じ論文のノード(フリーノード)を共有する著者のノード(アンカー)同士が近くなるように配置されている。そのため、局所的なコミュニティが存在し、その中の著者で論文が書かれていたとすると、同じコミュニティ内の著者のノードは近くに配置され、論文の

¹<http://dblp.uni-trier.de/>

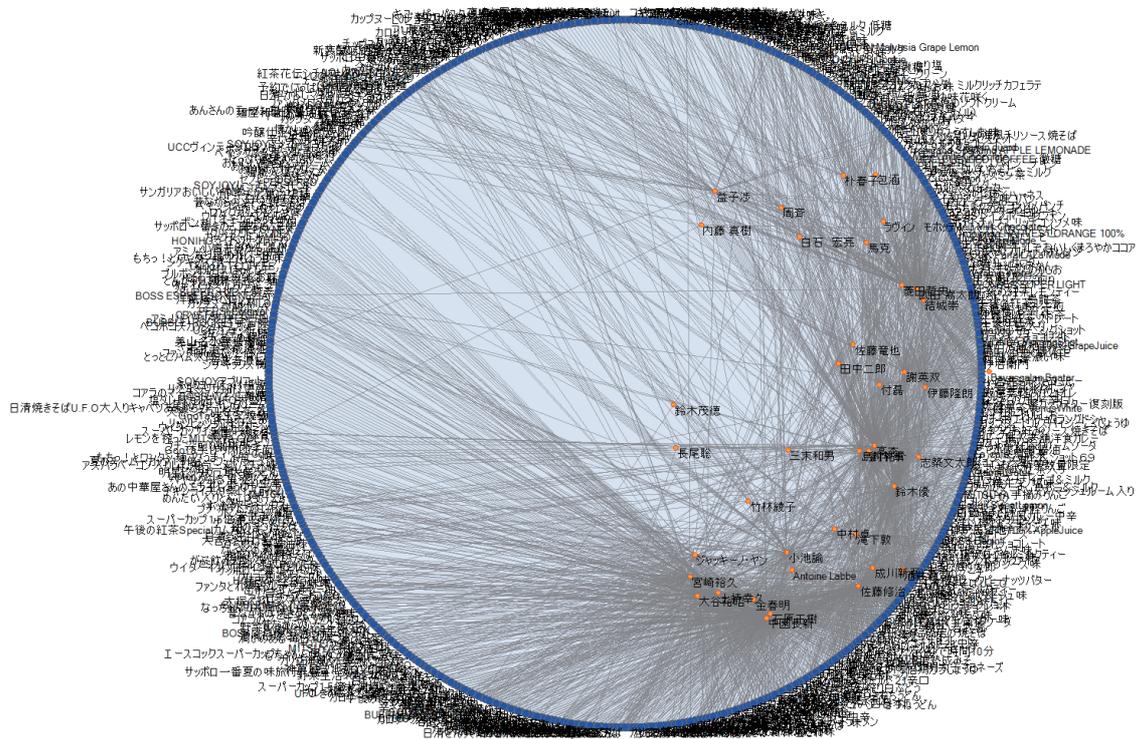
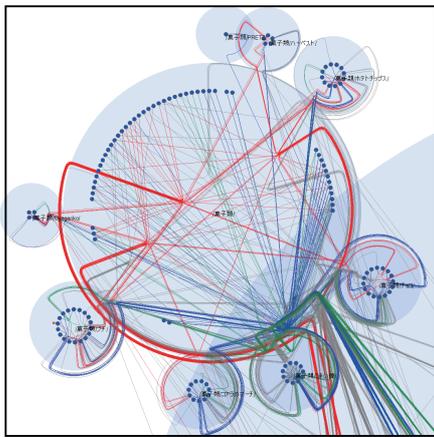
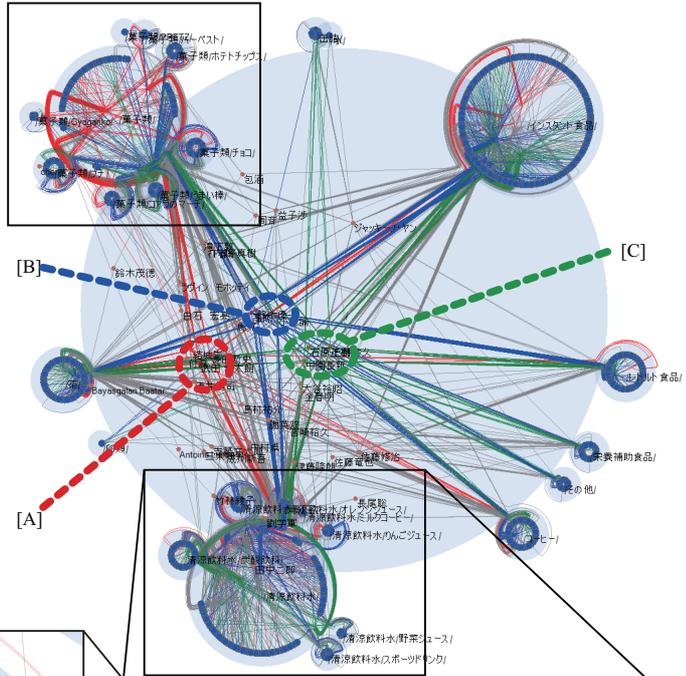


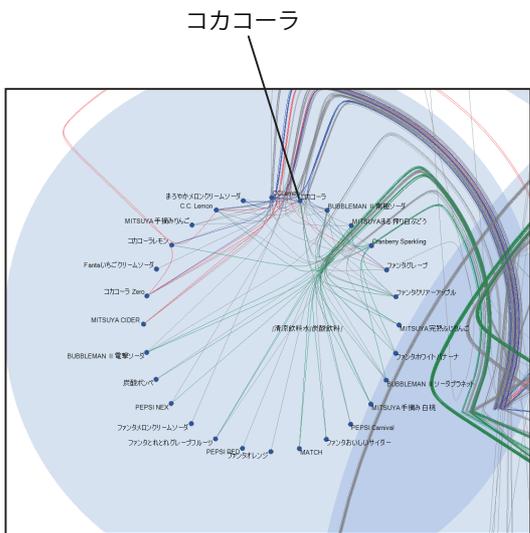
図 9.2: アンカーマップによる商品と購入者の関係の可視化例



(a) お菓子



(b) 清涼飲料水



(c) 清涼飲料水 / 炭酸飲料

図 9.3: 階層型アンカーマップによる商品と購入者の関係の可視化例

ノードは円周近くに配置されることとなる。

まず、図 9.4 を見ると、円周近くにフリーノードが多く配置されており、エッジがいくらかのノードから円周を沿うように放射状に広がっている。このため、このグラフには局所的なコミュニティが存在しているのではないかと考えられる。しかし、フリーノードやエッジが円周上に配置されているため接続関係の読み取りが困難である。

次に、開発したインタフェースを用いてクラスタマップを形成し、より詳細な分析を行う。クラスタマップを形成することで、円周付近に配置されていたフリーノードやその接続関係の把握が行い易くなると考えられる。先に示したアンカーマップを見ながらコミュニティの切れ目となりそうな部分を開発したインタフェースで 6 箇所囲い（図 9.4 の赤い丸）、検出された切れ目を使って 5 つの子クラスタマップを形成したものが図 9.5 である。この図を見ると、子クラスタマップ (a) と (c) ではあるノードから放射状にエッジがつながっているのがわかる。このうち (a) にズームした図が図 9.6(a) であり、「Jiro Tanaka」を中心としたコミュニティが形成されていることがわかる。実際、この子クラスタマップ内の著者のノードの多くが Jiro Tanaka の研究室に在籍している、もしくは在籍していた研究者である。今度は子クラスタマップ (a)(c) とは異なり、フリーノードが子クラスタマップの円周上に配置されている (b) に注目する。子クラスタマップ (b) を拡大したのが図 9.6(b) である。このクラスタマップでは、「Yu Suzuki」が [A], [B], それ以外の 3 つのコミュニティをつないでいることが分かる。そこで、3 つのコミュニティをつなぐキーパーソンではないかと考えコミュニティ間をつなぐ論文を調べたところ、「Yu Suzuki」という名前の別の研究者が少なくとも 3 人おり、[A], [B], それ以外のコミュニティで執筆した人物は異なっていた。つまり、全く別のコミュニティが 3 つ存在していることが分かった。

以上の例から、階層型アンカーマップと開発したインタフェースは、全体を俯瞰したのち、情報を細かな単位に分割し、より詳細な情報分析を行うプロセスを支援に有効であることが分かった。

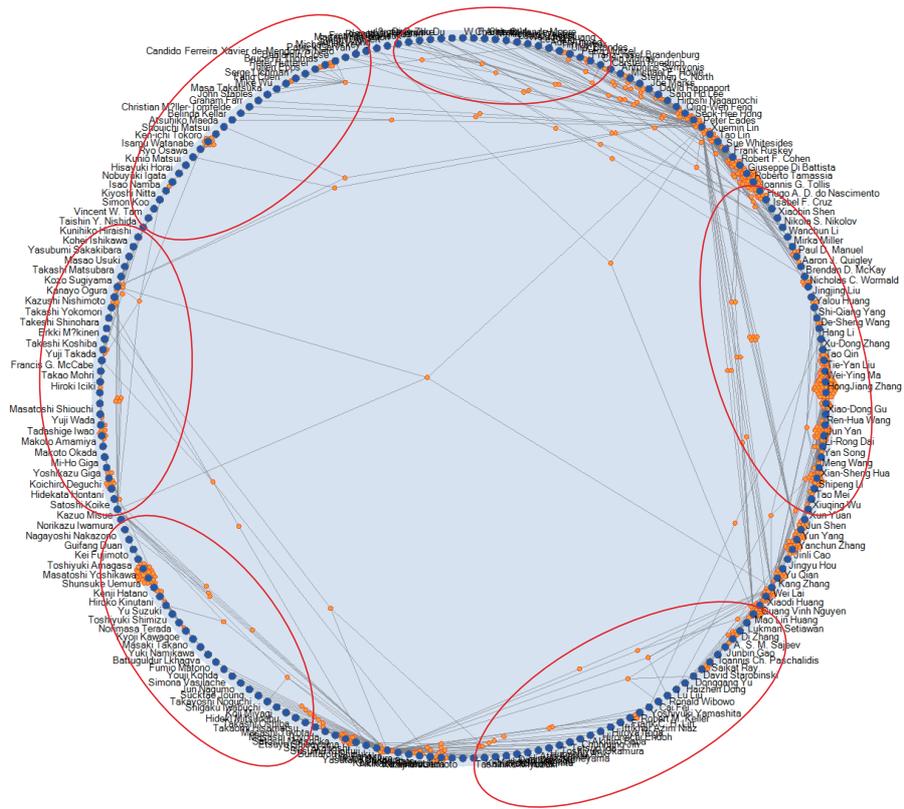


図 9.4: アンカーマップによる論文と著者の関係の可視化

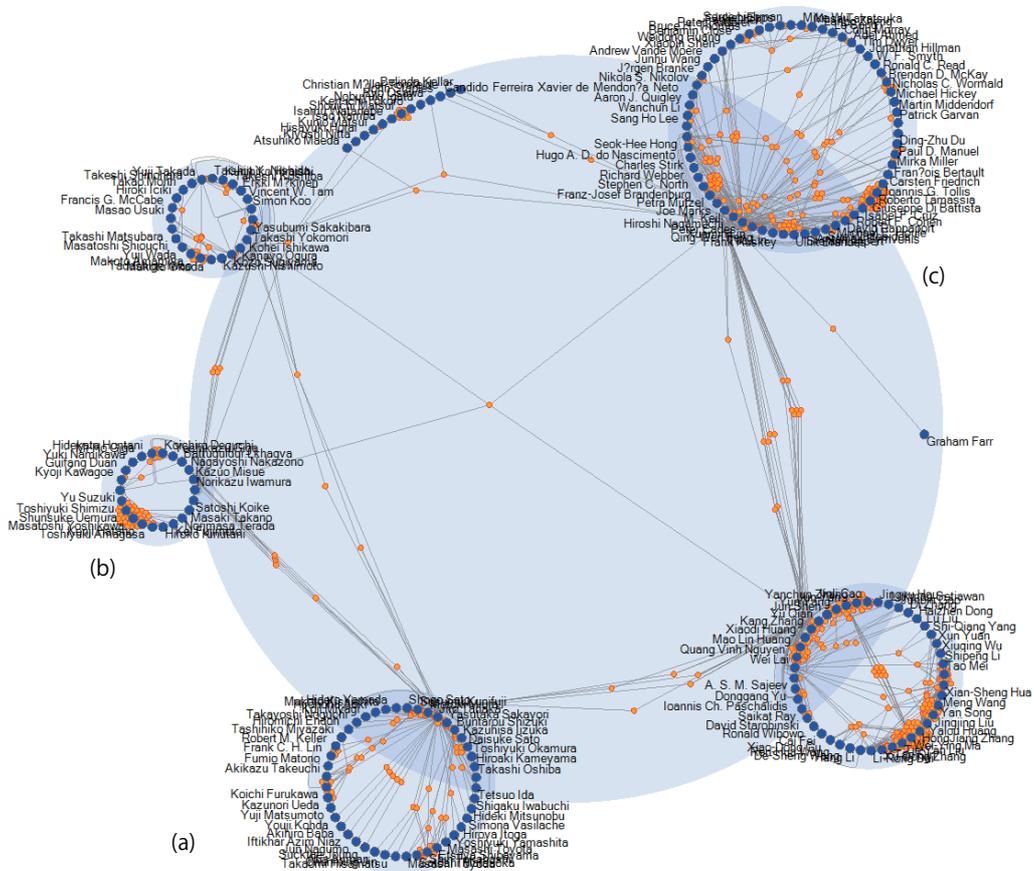


図 9.5: 階層型アンカーマップによる論文と著者の関係の可視化

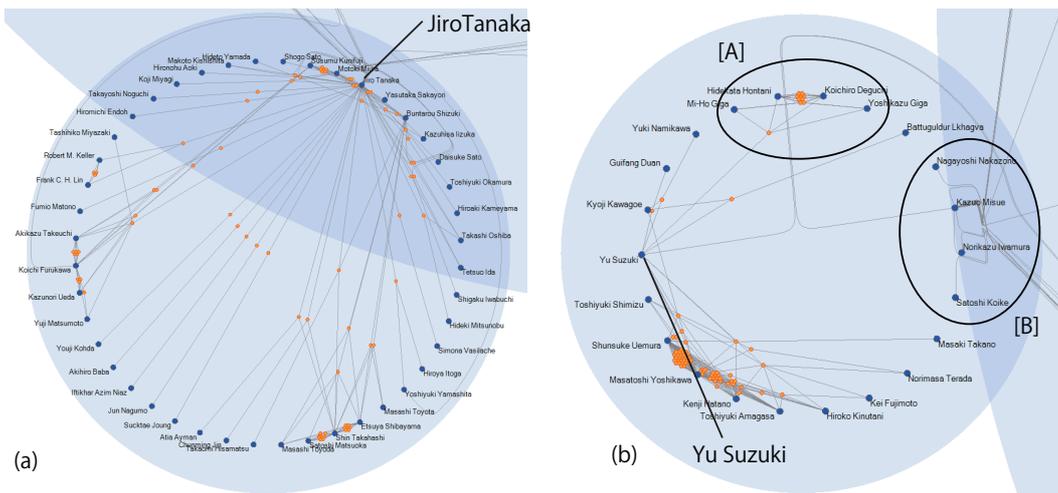


図 9.6: 子クラスタマップの拡大図

第10章 結論

本研究では，2部グラフの片方のノード集合が階層構造を持つグラフ構造「クラスタ2部グラフ」を導入し，その描画手法「階層型アンカーマップ」を開発した．

本研究の技術的な貢献は，階層型アンカーマップの描画手法を開発した点である．4種類の配置スタイルとクラスタマップの向きの決定方法を開発し，美的基準に基づくレイアウトの評価を行った．その結果，4種類の配置スタイルのうち円周上スタイルが美的基準を満足し安定して利用できることが分かった．また，向きの決定方法はエッジとクラスタマップの交差を減らし，視覚的な混雑を避ける向きを探索できることが分かった．

開発した描画手法が，大規模2部グラフの情報探索の際，全体の傾向を俯瞰した後により詳細な部分を分析する段階的な情報分析の支援に有効なことを具体例を用いて示した．また，開発した描画手法及びインタフェースが，従来の描画手法と比べより深い2部グラフ構造の情報の分析の支援を行えることを具体例を用いて示した．

階層型アンカーマップは，2部グラフの片方のノード集合が階層構造を持つグラフ構造を対象とした描画手法である．しかし，実世界では2部グラフの両方のノード集合に階層構造を持つグラフも存在している．今後の課題として，既に提案されているフリーノード側に階層構造がある場合の表現手法 [38] と組み合わせることで，両方に階層構造を持つグラフへの応用も考えられる．

謝辞

本論文を執筆するにあたり，指導教員である三末和男先生，副指導教員である田中二郎先生をはじめ，志築文太郎先生，高橋伸先生には研究方法から研究内容，論文執筆に至るまで丁寧なご指導と適切なお助言をいただき，心より感謝致します．インタラクティブプログラミング研究室の皆様にはゼミなどを通して貴重なご意見を数多くいただきました．特に，NAISチームの皆様にはチームミーティングでの意見のみならず研究生生活全体にわたって多くのご意見とご指摘をいただきました．ここに深く感謝致します．また，研究に従事することを承諾していただき，日頃より私を支えてくださいましたソフトイーサ株式会社の皆様に心より感謝いたします．最後に，様々な面で力になってくれた家族や多くの友人，大学生活でお世話になったすべての方々的心から感謝致します．

参考文献

- [1] Matthew Newton, Ondrej Sykora, and Imrich Vr' o. Two new heuristics for two-sided bipartite graph drawing. In *Graph Drawing 2002*, pp. 312–319, 2002.
- [2] Peter Eades. A heuristic for graph drawing. *Congressus Numerantium*, Vol. 42, pp. 149–160, 1984.
- [3] Thomas M. J., Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Softw. Pract. Exper.*, Vol. 21, No. 11, pp. 1129–1164, 1991.
- [4] Kazuo Misue. Drawing bipartite graphs as anchored maps. In *Asia-Pacific Symposium on Information Visualization (APVIS 2006)*, pp. 169–177, 2006.
- [5] Kazuo Misue. Anchored map: Graph drawing technique to support network mining. *IEICE Trans. Inf. & Syst.*, Vol. E91-D, No. 11, pp. 2599–2606, 2008.
- [6] Kilian Thiel, Fabian Dill, Tobias Kotter, and Michael R. Berthold. Towards visual exploration of topic shifts. In *IEEE International Conference on Systems, Man and Cybernetics 2007*, pp. 522–527, 2007.
- [7] John O'Donovan, Barry Smyth, Brynjar Gretarsson, Svetlin Bostandjiev, and Tobias Hollerer. Peerchooser: visual interactive recommendation. In *CHI '08*, pp. 1085–1088, 2008.
- [8] Emilio Di Giacomo, Luca Grilli, and Giuseppe Liotta. Drawing bipartite graphs on two curves. In *Graph Drawing 2006*, pp. 380–385, 2007.
- [9] Antoine P. Naud, Shiro Usui, Naonori Ueda, and Tatsuki Taniguchi. Visualization of documents and concepts in neuroinformatics with the 3d-se viewer. *Frontiers in neuroinformatics*, Vol. 1, No. 1, Nov 2007.
- [10] Ben Shneiderman and Maxine Cohen Catherine Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 2004.
- [11] Joshua Ho and Seok-Hee Hong. Drawing clustered graphs in three dimensions. In *Graph Drawing 2005*, pp. 492–502, 2006.

- [12] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 12, No. 5, pp. 741–748, 2006.
- [13] Lanbo Zheng, Le Song, and Peter Eades. Crossing minimization problems of drawing bipartite graphs in two clusters. In *Asia-Pacific Symposium on Information Visualization (APVIS 2005)*, pp. 33–38, 2005.
- [14] Takao Ito, Kazuo Misue, and Jiro Tanaka. Sphere anchored map: A visualization technique for bipartite graphs in 3d. In *HCI International 2009, LNCS 5618*, pp. 811–820, 2009.
- [15] Colin Ware and Glenn Franck. Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Transactions on Graphics*, Vol. 15, No. 2, pp. 121–140, 1996.
- [16] Janet M. Six and Ioannis G. Tollis. A framework for circular drawings of networks. In *Graph-Drawing '99*, pp. 107–116, 1999.
- [17] Emden R. Gansner and Yehuda Koren. Improved circular layouts. In *Graph Drawing 2006*, pp. 386–398, 2007.
- [18] Janet M. Six and Ioannis G. Tollis. A framework for user-grouped circular drawings. In *Graph Drawing 2003*, pp. 135–146, 2004.
- [19] Michael Kaufmann and Roland Wiese. Maintaining the mental map for circular drawings. In *Graph Drawing 2002*, pp. 12–22, 2002.
- [20] Ivan Herman, Guy Melancon, and M. Scott Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 6, No. 1, pp. 24–43, 2000.
- [21] George Robertson, Jock D. Mackinlay, and Stuart K. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *CHI '91*, pp. 189–194, 1991.
- [22] Tamara Munzner. H3: Laying out large directed graphs in 3d hyperbolic space. In *Proceedings of the 1997 IEEE Symposium on Information Visualization*, pp. 2–10, 1997.
- [23] Brian Johnson and Ben Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *VIS '91: Proceedings of the 2nd conference on Visualization '91*, pp. 284–291, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.
- [24] Stefan Hachul and Michael Jünger. Drawing large graphs with a potential-field-based multi-level algorithm (extended abstract). In *Graph Drawing 2004*, pp. 285–295, 2005.
- [25] Yaniv Frishman and Ayellet Tal. Multi-level graph layout on the gpu. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 13, pp. 1310–1319, 2007.

- [26] Chris Muelder and Kwan-Liu Ma. A treemap based method for rapid layout of large graphs. In *PacificVis 2008*, pp. 231–238, 2008.
- [27] Chris Muelder and Kwan-Liu Ma. Rapid graph layout using space filling curves. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 14, No. 6, pp. 1301–1308, 2008.
- [28] Michael Baur and Ulrik Brandes. Multi-circular layout of micro/macro graphs. In *Graph Drawing 2007*, pp. 255–267, 2008.
- [29] Tim Dwyer, Kim Marriott, Falk Schreiber, Peter J. Stuckey, Michael Woodward, and Michael Wybrow. Exploration of networks using overview+detail with constraintbased cooperative layout. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 14, No. 6, pp. 1293–1299, 2008.
- [30] Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin. Nodetrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 13, No. 6, pp. 1302–1309, 2007.
- [31] Peter Eades and Qing-Wen Feng. Multilevel visualization of clustered graphs. In *Graph Drawing 1996*, pp. 101–112. Springer, 1997.
- [32] Yaniv Frishman and Ayellet Tal. Dynamic drawing of clustered graphs. In *INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization*, pp. 191–198, 2004.
- [33] Hiroki Omote and Kozo Sugiyama. Method for drawing intersecting clustered graphs and its application to web ontology language. In *Asia-Pacific Symposium on Information Visualization (APVIS 2006)*, pp. 89–92, 2006.
- [34] 高杰. ソーシャルネットワーク分析のための視覚的インタフェースの構築. 筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻 2008 年度 修士論文, p. 50, 2009.
- [35] Yingxin Wu and Masahiro Takatsuka. Visualizing multivariate network on the surface of a sphere. In *Asia-Pacific Symposium on Information Visualization (APVIS 2006)*, pp. 77–83, 2006.
- [36] Takayuki Itoh, Chris Muelder, Kwan-Liu Ma, and Jun Sese. A hybrid space-filling and force-directed layout method for visualizing multiple-category graphs. In *PacificVis 2009*, pp. 121–128, 2009.
- [37] Christopher Collins and Sheelagh Carpendale. Vislink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 13, pp. 1192–1199, 2007.

- [38] 佐藤修治. クラスタ情報の可視化による大規模 2 部グラフの可読性向上手法. 筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻 2007 年度 修士論文, p. 69, 2008.
- [39] Kai Xu, Rohan Williams, Seok-Hee Hong, Qing Liu, and Ji Zhang. Semi-bipartite graph visualization for gene ontology networks. In *Graph Drawing 2009*, 2009.
- [40] Weiwei Cui, Hong Zhou, Huamin Qu, Pak Chung Wong, and Xiaoming Li. Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 14, No. 6, pp. 1277–1284, 2008.
- [41] 石原正樹. 動的ネットワークの成長過程と差分の可視化手法. 筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻 2006 年度 修士論文, p. 73, 2007.
- [42] Michael J. McGuffin and Igor Jurisica. Interaction techniques for selecting and manipulating subgraphs in network visualizations. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 15, No. 6, pp. 937–944, 2009.
- [43] Kenta Suzuki and Ken Wakita. Extracting multi-facet community structure from bipartite networks. *Computational Science and Engineering, IEEE International Conference on*, Vol. 4, pp. 312–319, 2009.