数理モデル化と問題解決　10－5
（１９９６．１１．２６）

結び目と平面グラフの最適埋蔵

落合豊行、鴨浩靖
奈良女子大学理学部情報科学科
土井綾乃、今藤紀子
奈良女子大学理学研究科情報科学専攻

平面グラフを平面上に美しく埋蔵する方法を紹介する．平面グラフは、全ての辺を直線で描くことができるので、平面グラフの埋蔵は、頂点の位置だけにより、一意的に決まる．ここで紹介する方法は、各頂点の隣接関係、外周路（非有界領域に実現される面), そして 簡単な行列の計算のみから、外周路上にない頂点の座標を求めるものである．具体的な配置のデータは、隣接関係における隣接度を調節することによって変更が可能である．この方法は、Pーデータと呼ばれる離散的データから、元の結び目を復元する場合にも利用できる．また、三次元においても同様の効果が得られ、トーラスなどの閉曲面上や、円柱の表面における三角形分割の描画など、さまざまな応用が期待できる．

Nicely embeddings of knots and planar graphs

Ochiai Mitsuyuki, Kamo Hiroyasu
Information and computer sciences, Faculty of sciences, Nara Women's University
Doi Ayano, Imafuji Noriko
Information and computer sciences, Graduate school, Nara Women's University

We present the way to nicely embed planar graphs on a plane. Since all of the edges of plane graphs can be drawn as the straight line segments, the embeddings of planar graphs are determined only by the coordinates of each verteces. Our way gets displacement coordinates of each verteces induced by the simple matrix computation associated with adjacency of each verteces and the initial coordinates of the verteces on the outermost domain.. Furthemore,the data of position can be changed to propery arrange of adjacency degree. This way is also available to the drawings of knots. Moreover it sure that we can get similar result even in 3-dimentional space, so we expect many kinds of applications such as the drawings of triangulated cylinders or closed orientable surfaces.

1　行列と外周路

グラフが平面の中に埋め込み可能である、あるいは平面的であるというのは、そのグラフを平面内に描く場合、２つの辺が交わるのはそれがともに接続している点においてのみであるように描くことができるときをいう．平面的グラフ G のそのような描画は G の平面への埋め込みと呼ばれる．また、平面グラフとは平面的グラフの平面への埋め込みのことである．

ここでは、コンピュータを用いて，平面的グラフを平面上に美しく埋め込む方法について考察する．

ただし，グラフは単純グラフ，すなわち，多重辺あるいはループをもたないものとする．

まず，グラフは頂点数と，その頂点間の隣接関係によって決定づけられている．この隣接関係は，隣接行列と呼ばれる抽象的なデータによって表現できる．

n 個の頂点 $v_1,\ldots,v_n$ をもつグラフ G の隣接行列とは，

$$a_{ij}=\begin{cases}1 & (v_i \text{ と } v_j \text{ が隣接しているとき})\\ 0 & (v_i \text{ と } v_j \text{ が隣接していないとき})\end{cases}$$

で定められる $n \times n$ 行列 $A = (a_{ij})$ のことである．

例えば，図1のグラフの隣接行列は，次の図の中のようになる．

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

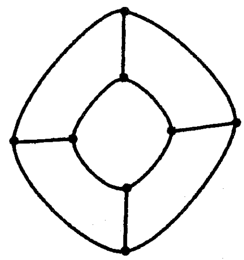


図 1:

平面グラフ G の隣接行列が与えられ，平面へ埋め込んだ場合の非有界な領域の境界となる閉路 (外周路) が指定されたとき，グラフ G をバランスよく描く方法について述べる．

外周路上には r 個の頂点があるとし，煩わしさを避けるため，それらの頂点を順に $v_1, v_2, \ldots, v_r$ とする．これらを外頂点と名づける．残りの頂点 $v_{r+1}, \ldots, v_n$ を内頂点と名づける．まず，外頂点を その順番通りに単位円周上に均等に配置する．そして，$n-r$ 個の内頂点を原点におく．次に，各内頂点を，その頂点がつながっている他の頂点の重心に移動させることにする．この移動を逐次繰り返して，各頂点間に平衡状態が生じたとき，バランスのよい平面グラフが生成できる．この移動は次に述べる変換行列 F を用いて行う．

重心への移動を k 回繰り返した後の，頂点 v_i の座標を (x_i^k, y_i^k) とし，$\mathbf{x}^k = (x_1^k, \ldots, x_n^k)$, $\mathbf{y}^k = (y_1^k, \ldots, y_n^k)$ とおく．

このとき，$\mathbf{x}^k$ から $\mathbf{x}^{k+1}$ への変換と $\mathbf{y}^k$ から $\mathbf{y}^{k+1}$ の変換は，同じ変換行列 F で

$$\mathbf{x}^{k+1} = F\mathbf{x}^k,$$
$$\mathbf{y}^{k+1} = F\mathbf{y}^k$$

と表される．ただし，ベクトルは縦ベクトルとみなしている．ここで，$F = (f_{ij})$ は次で定義される。

$$f_{ij} = \begin{cases} \delta_{ij} & (i = 1, \ldots, r) \\ a_{ij}/d(v_i) & (i = r+1, \ldots, n) \end{cases}$$

ここで，δ_{ij} はクロネッカーの δ 関数 ($i = j$ のとき 1，$i \neq j$ のとき 0)，a_{ij} は G の隣接行列の i, j 成分，$d(v_i)$ は頂点 v_i の次数 (頂点に接続する辺の数) である．

$$\mathbf{x} = (x_1, \ldots, x_n) = \lim_{k\to\infty} \mathbf{x}^k,$$
$$\mathbf{y} = (y_1, \ldots, y_n) = \lim_{k\to\infty} \mathbf{y}^k$$

が，われわれが目標とする平衡状態における頂点の座標である．ところが，コンピュータでこの計算をそのまま実行するには，変換を適当な回数で打ち切らざるをえず，計算量が多い上に，しかも打ち切り誤差が生じる．しかし，それを避ける巧妙な手段がある．まず，変換行列 F は

$$F = \begin{pmatrix} E & O \\ B & A \end{pmatrix}$$

の形をしている．ここで，E は r 次の単位行列，O は $r \times (n-r)$ 型の零行列，A は $(n-r) \times (n-r)$ 型の行列，B は $(n-r) \times r$ 型の行列である．また，$\hat{\mathbf{x}}^k = (x_1^k, \ldots, x_r^k)$, $\tilde{\mathbf{x}}^k = (x_{r+1}^k, \ldots, x_n^k)$ とおく

と，外頂点は動かないのであるから，当然 $\hat{\mathbf{x}}^k = \hat{\mathbf{x}}^0$ である．よって，$\tilde{\mathbf{x}}^{k+1} = A\tilde{\mathbf{x}}^k + B\hat{\mathbf{x}}^0$ となる．求めるべき極限の内頂点の x 座標を $\tilde{\mathbf{x}} = (x_{r+1}, \ldots, x_n)$ とおくと，これはこの変換で不変なはずであるから，$\tilde{\mathbf{x}} = A\tilde{\mathbf{x}} + B\hat{\mathbf{x}}^0$ を満たす．そこで，これを解いて，

$$\tilde{\mathbf{x}} = (E - A)^{-1} B \hat{\mathbf{x}}^0$$

を得る．ここで，E は A と同じサイズの単位行列である．また，y 座標に関しても，同様に

$$\tilde{\mathbf{y}} = (E - A)^{-1} B \hat{\mathbf{y}}^0$$

と求まる．

先の隣接行列の例から，もとのグラフの平面への描画を作成してみる．初めに外周路を指定しなければならない．ここでは，外周路を v_1,v_2, v_3,v_4,v_1 とする．そうすると，頂点移動の変換行列は

$$F = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 1/3 & 0 & 1/3 \\ 0 & 1/3 & 0 & 0 & 1/3 & 0 & 1/3 & 0 \\ 0 & 0 & 1/3 & 0 & 0 & 1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 1/3 & 1/3 & 0 & 1/3 & 0 \end{pmatrix}$$

となる．

逆行列を用いた方法で求めてみる．

$$A = \begin{pmatrix} 0 & 1/3 & 0 & 1/3 \\ 1/3 & 0 & 1/3 & 0 \\ 0 & 1/3 & 0 & 1/3 \\ 1/3 & 0 & 1/3 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1/3 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \\ 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1/3 \end{pmatrix},$$

$$(E - A)^{-1} = \begin{pmatrix} 7/5 & 3/5 & 2/5 & 3/5 \\ 3/5 & 7/5 & 3/5 & 2/5 \\ 2/5 & 3/5 & 7/5 & 3/5 \\ 3/5 & 2/5 & 3/5 & 7/5 \end{pmatrix},$$

$$\hat{\mathbf{x}}^0 = (0,1,0,1), \quad \hat{\mathbf{y}}^0 = (1,0,1,0)$$

より，

$$\begin{aligned} \tilde{\mathbf{x}} &= (E - A)^{-1} B \hat{\mathbf{x}}^0 = (0, 1/3, 0, 1/3), \\ \tilde{\mathbf{y}} &= (E - A)^{-1} B \hat{\mathbf{y}}^0 = (1/3, 0, 1/3, 0) \end{aligned}$$

が求まる．明らかに，この方法の方が計算量が少なく，しかも正確である．

これらの結果からグラフを描くと図 2のようになる．

当然のことであるが，外周路の指定を間違えて，埋め込みの 1 つの領域の境界になりえない閉路を指定した場合には，正しい埋め込みは得られない．

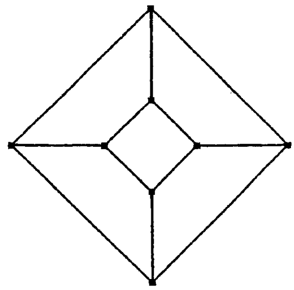


図 2:

この方法でよりバランスよく描画するためには、まず、面積の大きい領域を探して、その領域に頂点を設け、新たに隣接関係をつくる．加えた頂点を $v_m(r+1 \leq m \leq n)$ とすると、$F = (f_{ij})$ を

$$f_{ij} = \begin{cases} \delta_{ij} & (i = 1, \ldots, r) \\ a_{ij}/(td-1) & (i = r+1, \ldots, n, j \neq m) \\ t(d-1)/(td-1) & (j = m) \end{cases}$$

と定義する．ここで、t は $2 \leq t$ なる実数で、d は $d(v_i)$ のことである．このようにすると、よりバランス良く描画することができる．

図 3において、左側が隣接関係のみによる描画、中央が面積の大きい領域に頂点を設けることにより得た新たな隣接関係による描画、右側は加えた頂点の隣接度を 2 倍にして描いたものである．[1]

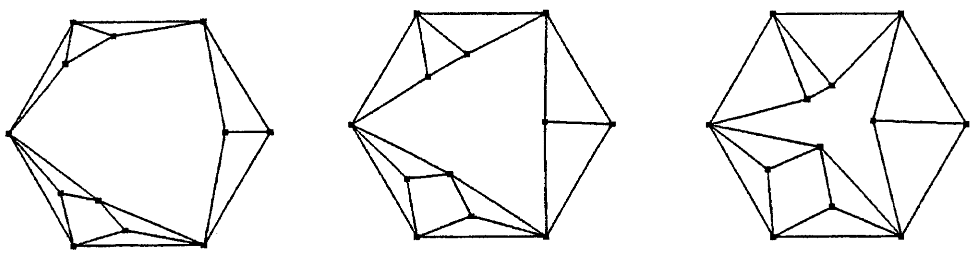


図 3:

この方法は面積の大きいところから隣接度を調整することにより、バランス良く描画することを目指しているが、最適な配置データに到達するためのアルゴリズムとしては完成していない．

2 正則射影の表示法

前節では，抽象的なデータにより表現されたグラフを平面上に美しく描く方法について考えた．本節では，絡み目を美しく描く方法を同様にして考察してみる．グラフの隣接行列のように，絡み目の正則表示を表現するデータ (P- データと呼ぶ) を構成する方法について述べる．

いま，各成分ごとに向きの指定された絡み目 L $(= K_1 \cup \ldots \cup K_n)$ が，ある正則表示として与えられているとする．また，各成分は $K_1, K_2, \ldots, K_n$ と順序が指定されているものとする．

まず，準備として，絡み目の各成分 $K_1, K_2, \ldots, K_n$ 上に，交点以外の任意の 1 点を選び，起点として固定する．

次に，最初の成分 K_1 から順に，それぞれの起点から指定された向きに沿って結び目を一周する．このとき，上方交点，下方交点に出会うごとに番号を 1 つずつ増やしながら番号づけをしていく．新しい

成分に移っても番号は1に戻さずそのまま増やしていくことにする．これにより各交点は，上方交点，下方交点ごとに番号づけされたことになる (図 4).

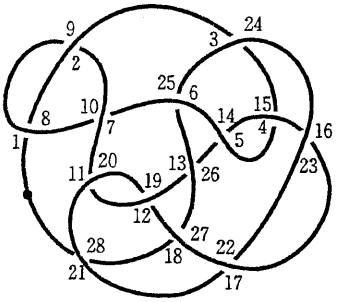


図 4:

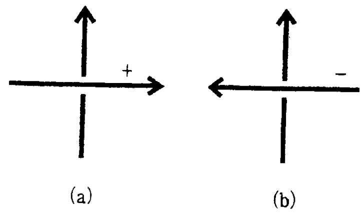


図 5:

ここで2つの数列を定義する．指定された順番における i 番目の成分の，上方または下方交点の数を並べたものを数列 m_i とする．もう1つの数列を作るために，もう一度1番目の成分の起点から出発し，絡み目を一周していく．このとき，上方交点を通過するたびに，対応する下方交点の番号を書き出していくことにする．その際，番号と一緒に交点の符号も書き出すようにする．交点の符号は交差の仕方によって図5のようになる．これにより長さが交点数と等しい数列が得られる．この数列を a_n とする．これで準備は完了した．

絡み目 (の正則表示) を表現するための P- データとは，3行にわたって記される．1行目は交点数の2倍の数 (上方交点と下方交点の合計の数) と絡み目の成分数を並べ，2行目には数列 m_i を，3行目には数列 a_n を順に並べる．

図4の例の P- データは，以下のようになる．

14	1												
14													
−9	−14	+25	−1	−7	+19	−4	−22	+27	+11	−28	−16	+3	+13

図 6:

上で定義した P- データをもとにして，各上下交点間の関係を表す完全 P- データと呼ばれるものを簡単に得ることができる．すなわち，

prd	1	2	3	4	5	6	7	8	9	10	11	12	13	14
a	8	9	24	15	14	25	10	1	2	7	20	19	26	5
b	2	1	4	2	1	3	2	1	2	1	4	3	4	2
c	1	1	1	1	1	1	1	1	1	1	1	1	1	1

15	16	17	18	19	20	21	22	23	24	25	26	27	28
4	23	22	27	12	11	28	17	16	3	6	13	18	21
1	2	1	3	4	3	1	2	1	3	4	3	4	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1

という形の表で，最上段の数字 i が i 番目の上方または下方交点を示し，その縦の列中の数字は

a : 対応する下方または上方交点の番号

b : 交点の属性 (1: 上方／符合 −, 2: 下方／符合 −, 3: 上方／符合 +,
4: 下方／符合 +)

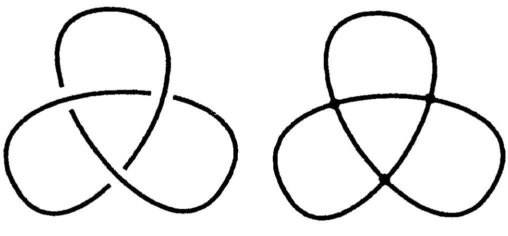
c : 属している成分の番号

を表している．上は図 4 の完全 P- データである．

3 P- データから正則射影へ

本節では，絡み目の P- データが与えられたとき，そのもとになる絡み目の正則射影を描く方法について考えてみることにする．

L を向きのついた絡み目，$\tilde{L}$ をその正則表示，n をその正則射影の交点数とする．すると $\tilde{L}$ によって，4- 正則 (有向) 平面グラフ，すなわち各頂点から常に 4 本の辺が出ている平面に埋め込まれたグラフが得られる．以下，このグラフを G_L と書くことにする．図 7は，クローバー結び目の例である．



(a) $\tilde{L}$　(b) G_L

図 7:

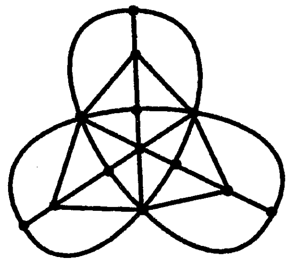


図 8:

P- データが与えられたとき，完全 P- データを作り，それから G_L を構成する．しかしそれらは 3 連結であるとは限らないので，2.1 節で示したような，グラフを平面上に描く方法を用いることができない．そこで，G_L の各面に対し，さらに重心細分を行うことにし，それによって得られるグラフを G_L^* とおく (重心細分とは，各辺および面について内部に新たに頂点 (重心) をとり，面の重心の頂点と，その面が含んでいる辺の重心の頂点および面が含んでいるもともとの頂点とを辺で結んだものである)．ただし，一番外側の面については細分をしない．クローバー結び目の例では図 8のようになる．

G_L^* のグラフをもとに変換行列 f が定義できる．図 8の f は 次式のようになる．

f を用いて頂点の収束先を前節の方法で求めることができ，コンピュータで描画するときは，このようにして得た座標をもとに，ベジエ曲線を用いて絡み目を滑らかになるように描き，交点の近くではどちらが上方または下方交点かがわかるように描けばよい．

変換行列 f を定めるには，前にも指摘したように，一番外側の面の周を形成する外周路を指定しなければならない．このためには，絡み目の正則表示 $\tilde{L}$ から定まる完全 P- データから，平面グラフ G_L がどのような面をもつかを決定する必要がある．正則表示において，絡み目を与えられた方向に沿って交点をたどったとき，それが進行方向右から左へか，その逆で交差するかの情報を完全 P- データは含んでいる．

$$
\begin{pmatrix}
1 & & & & & & & & & & & & \\
& 1 & & & & & & & & & & & \\
& & 1 & & & & & & & & & & \\
& & & 1 & & & & & & & & & \\
& & & & 1 & & & & & & & & \\
& & & & & 1 & & & & & & & \\
1/4 & 1/4 & & & & 1/4 & & 1/4 & & & & & \\
& 1/4 & & & & 1/4 & 1/4 & & 1/4 & & & & \\
& 1/6 & & 1/6 & & 1/6 & & 1/6 & & 1/6 & & 1/6 & \\
& 1/4 & & 1/4 & & & & & 1/4 & & 1/4 & & \\
& 1/4 & 1/4 & 1/4 & & & & & & 1/4 & & & \\
& & & 1/4 & & 1/4 & & & 1/4 & & & & 1/4 \\
& & & 1/4 & 1/4 & 1/4 & & & & & & 1/4 &
\end{pmatrix}
$$

図9の左側は、隣接関係のみで書いたものであり、右側は、2辺形の領域に対して、隣接度による調整を行なって得られたものである.

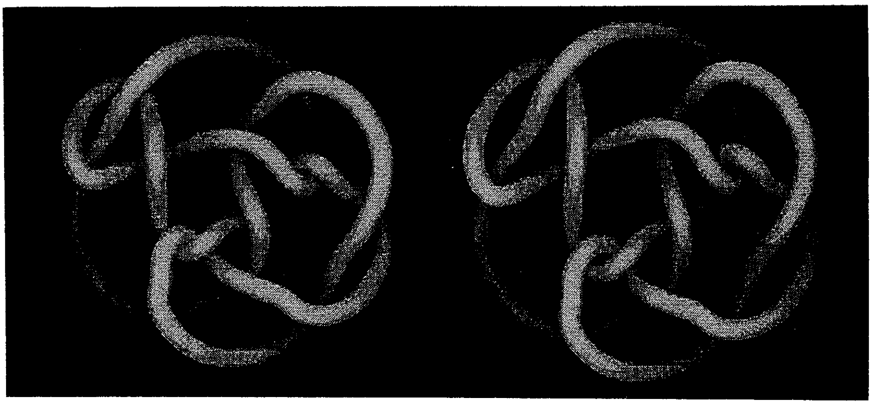


図 9:

また、三次元でも同様にして二つの円を不動点として指定することにより表面が三角形分割された立体が得られる. 例えば、図 10のような グラフを考える.

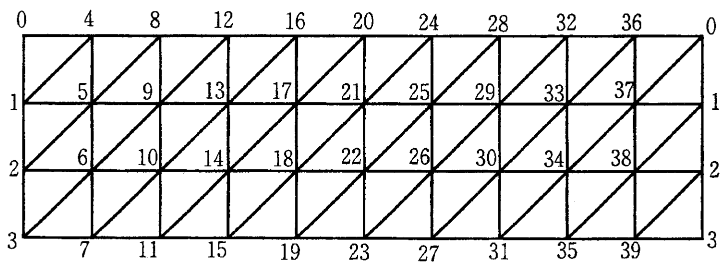


図 10:

サイクル 0, 4, 8,12,16,20,24,28,32,34,0 と サイクル 3,7,11,15,19,23,27,31,35,39,3 を 2 つの単位円周上に等間隔に配置しその他の頂点は 1 点に置く．この状態を、初期状態にして前述の行列を計算することによって、三角形分割された円柱の配置データが得られる．図 11がこの計算によって得られた図である．

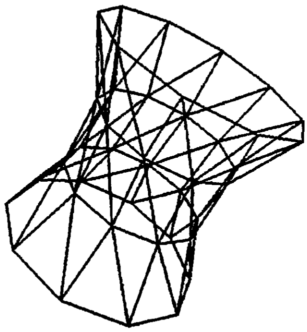


図 11:

4 応用

この論文で述べた方法は、次のようなことを行うために利用可能である.

- 結び目をバランス良く描画する
- 三角形分割された曲面を簡単に描画する
- 曲面をさらに滑らかに描画する

参考文献

[1] Norishige Chiba, Kazunori Onoguchi, Takao Nishizeki, “Drawing Plane Graphs Nicely” Acta Informatica 22, 187-201(1985)

[2] 落合豊行、山田修司、豊田英美子, “コンピュータによる結び目理論入門,” 牧野書店