6J-03　平面分割におけるグラフの最適配置*

6J－3

田山聖子　三浦孝夫
産能大学 †

1　前書き

何か物事を直感的にすばやく理解する方法として，図を用いることが多い．例えば，フローチャートやＰＥＲＴ，論理回路図，状態遷移図などがあげられる．これらは，各々が表現していることを整理したり，あるいは実際の動きを追ったりするときに役立つ．しかし，一定の表示範囲に対して，図の大きさが適正でなかったり，あるいは表現したいことと異なった図であったりすると（例えば，線が切れていたり，誤ったところへ結びついていたりするなど），かえって分かりにくくなってしまう．上記のようなことは手動によれば当然起こり得る可能性は高いと考えられる．よって，計算機により自動的に上記のようなことがないような適正に配置され図示された図を得ることは大変有効であると考える．

本稿では，対象として有向グラフをとりあげ，決められた範囲に適正に配置するアルゴリズムを提案する．

2　適正な配置の定義

適正な配置とは以下の４つの条件を満たすものとする．

(1) 始まり start と終わり end は各々，一番上と一番下の中央に配置する．
(2) 弧の数はできるだけ少なくする．
(3) 中心となる流れは，できるだけ中央になるようにする．
(4) バランスがとれていること．

本稿での有向グラフの適正な配置の定義は，有向グラフに代表されるような形の図から，いかにすばやく”流れ”を読み取ることができるかということを念頭においている．条件１は，始めと終わりを強調することによって，図（有向グラフ）の基本的な流れを視覚的に印象づけるために有効である．条件２は，画面（表示範囲）をすっきりさせ，見やすくする．弧が込み合うと，点と点のつながりが掴みにくくなる．条件３は，図がどのような用途に使われるかによって，点の位置が変わるということを仮定した上で考えた．例えば，先行弧を早道として考える。それは何に対して早道かというと，ある主要となる流れに対してである．図からできるだけすみやかに流れを捉えるためには，中心となる流れを上下方向に，しかもできるだけ中心にもってくることが大変有効であると考える．条件４は，点が配置される各レベルごとの点の数によって，分割幅を変化させることを意味し，できるだけ対称的に配置されるようにする．

3　入力グラフの定義

本稿のアルゴリズムにおいて入力可能なグラフは以下の条件をすべて満たすものである。

(1) 有向グラフである。
(2) 始まり start と終わり end が各々必ず１つのみ存在する。
(3) 任意の点 a から b への弧は高々一つ。
(4) 任意の点 a からそれ自身への弧は存在しない。

条件２については，複数の始まりがあれば，その各々の点を終点とし start を始点とする弧を加える。複数の終わりがある場合も同様に，その各々の点を始点とし，end を終点とする弧を作る。

以上のような条件を満たす有向グラフについて，点集合と弧集合 (a,b,l)，start と end のラベルを入力する。弧集合において，a,b は点，l は弧ラベルである。点も弧もラベルは正の整数で与える。本稿では，start のラベルは０に固定してある。弧ラベルは，まず，start から end へ到達する主要な経路を構成する弧に固定した値を与える．そして，各点において，左に配置したい弧ほど大きい値を，右に配置したい弧ほど小さい値を与える．固定値がついている弧があるならば，その固定値も含めて重み付けを与える。

*Best-suited Arrangement of Graphs by Plane Partition
†Sanno College

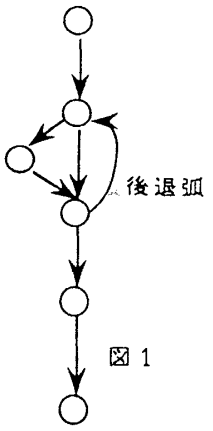

図１

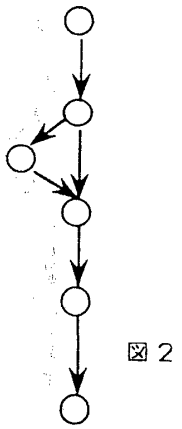
図２

4　有向グラフを用いた配置アルゴリズム

アルゴリズムの全体の流れは次のようになっている。入力として頂点集合 v，弧集合 (a,b,l),end を与える。a は始点，b は終点，l は弧の重みとする。最終的に，紙面，または画面への出力を得るものとする。

(1) 深さ優先探索を行う
(2) 後退弧の除去
(3) 先行弧に対する処理
(4) 分割･座標決定
(5) 弧配置･図示

4.1　前処理

ここでは，(1) と (2) の処理について述べる．あらかじめ，隣接行列が作成されているものとする．深さ優先探索は，固定値が与えられている経路を優先的にたどっていく．各頂点のたどりの順番は格納しておく．その順番において，大きい値から小さい値へ向かう弧は，交差弧と後退弧の２通りが考えられる．その中で，先祖へ向かう弧が後退弧で，そうでなければ交差弧である．各頂点 k に真の子孫の数をもたせることにし，以下を満たすとき，後退弧となる．ただし，a,b は弧の始点と終点であり，sison[k] には頂点 k の真の子孫の数，jyun[k] には頂点 k のたどりの順番が格納されている．

$$jyun[b] \leq jyun[a] \leq jyun[b] + sison[b]$$

これより，後退弧を除去する．後退弧の除去は，(3) の中で，各頂点において end までの最長経路を求めるとき閉路が存在してはアルゴリズム上問題があるため，必要となる．後退弧は，配置において外して考え，あらかじめ用意しておいたスペースに最後に付け加えるものとする．

図１は与えられた有向グラフで，図２は後退弧除去後の有向グラフである。

4.2　先行弧に対する処理

この処理は次の通りである。

(1) 各頂点において，end までの最長経路を求める
(2) 先行弧において，抜かしているレベルに仮の頂点を作成
(3) 深さ優先探索を行う

(1) は，(2) において，いくつのレベルを抜かしているか調べるために必要となる．全体で，最大の最長経路の値を記憶しておく (MXP)．(2) よりできたグラフは中抜きのない形となる。新たにできた弧の重みは元の先行弧の重みと同じにする。仮の頂点の存在は，各レベルの分割において，先行弧を含めて考えるのに有効となる．(3) は，start から弧の重みが大きい方から優先的にたどっていく．これにより，各頂点において，左に配置したい弧ほどたどりの順番は小さくなる．

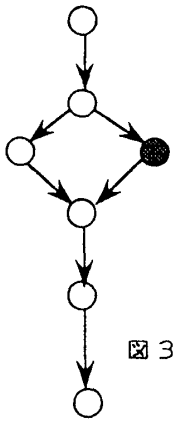

図3

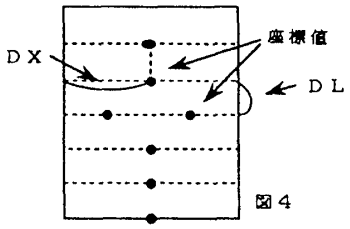

図4

図3は図2に上記の処理を施した結果である。

4.3 分割·座標決定

ここでは，各レベルでの頂点の配置を決定する。流れは次の通りである。

(1) P[1..k1] に start を格納する
(2) P[1..k1] が end となるまで以下を再帰的に繰り返す
(3) P[1..k1] の終点を Q[1..k2] へ格納する（重複しないようにする）
(4) Q[1..k2] をたどりの順番について小さい順にソートする
(5) ソートされた Q[1..k2] を順に，左から座標を割り当てる

ここでは，上から順に各レベルごとの頂点を配置していく．(3)では，各レベルにおける頂点を抜きだしている．ここでのグラフは，各弧における始点と終点の最長経路の差が1となっているため，単純に，終点を次の（1つ下の）レベルへ割り当てることが可能である．(4)においては，前小節(3)より，左へ配置したいものほど Q[] の始めの方となる．(4)では，具体的な座標が割り当てられる．表示範囲の縦（Y軸方向）の長さを MXP で割り，その分割幅をレベル幅 DL とする．横（X軸方向）の長さは Q[] の個数で割り，その分割幅を DX とする．調整幅 T には DX/2 を格納しておく．各頂点にあたえる座標は，(DL*現レベル，DX*k2-T) である．なお，表示範囲は，後退弧用のスペースとして，両脇に確保する分を除いた範囲としておく．以下，Q[] を現レベルでの頂点とし，再帰的に (2) から (5) を繰り返す。

図4は，図3において与えられる座標を示している。

4.4 弧配置·図示

この処理は次の通りである。

(1) 頂点を図示
(2) 前処理で得られたグラフ（隣接行列）における弧を図示
(3) 後退弧を図示

(1)の際には，start と end は各々，一番上のレベルと一番下のレベルに配置する．仮の頂点は図示しない．(2)においては，始めに，弧の数を減らすために，2こ以上の始点から2こ以上の終点の組について，各始点が全ての終点に対して弧をもつとき，図5のように図示する．次に，上記の場合に当てはまらない弧を図示し，(3)において，後退弧をできるだけ両脇を通るように配置する．

図5

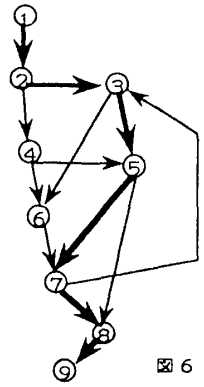

図6

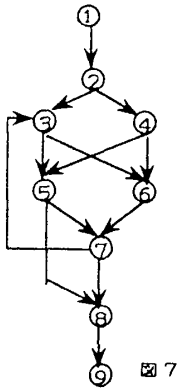

図7

5 評価

適正な配置の条件について，(1)は，明らかに満たされている．(2)については，図5のような形にすることにより減らすことを考えた．(3)については，重みを与えることにより実現すると考える．(4)についても，3.4において考慮している．これらの4つの条件をもとに配置を考えた結果，図6は図7のように清書される．図6において，太い矢印の弧が中心となる経路とする。明らかに図6より図7の方が流れが掴みやすく，見やすいと考えられる．このアルゴリズムの計算量は，頂点の数を v，弧の数を e とするとき，全体の流れ (1),(2),(3),(4),(5) の順で，O(v+ev),O(e),O(v^3+e),O(e+vlogv),O(v+v^3+e) である．

6 結び

本稿では，決められた表示範囲に有向グラフを適正に配置するアルゴリズムを提案した．

本稿において，交差を減らす問題について考慮していないのは，次の理由による。

(1) 流れの捉えやすさは，その図を見る人間の意図が大きく関係するため，その意図を配置に反映させたほうが現実的である
(2) 交差を最小にする問題はＮＰ完全である
(3) 発見的手法は考えられているが，弧の重みを用いているので，見る人間の意図を配置に反映させるために重み付けを考えることは，それほどの制限ではない

なお，発見的手法を用いて交差の問題を取り扱ったアルゴリズムについては，参考文献 [?] を参照。

よって，本稿では，図を使用する人間の意図を反映させることを考えた。その方法は，適正な配置の条件に基づいて考えた。今後の課題としては，後退弧を含めた配置である．

参考文献

[1] Emden R.Gansner.Eleftherios.Stephen C.North.and Kiem-phong Vo.A Technique for Drawing Directed Graph.IEEE Transactions on Software Engineering.vol.19.no.3.March 1993