グラフィクスとＣＡＤ　50－4
（1991．5．24）

路線図としての要件を満たすグラフ描画の試み

酒井 恵光　　川合 慧

東京大学教養学部情報・図形科学教室

地下鉄などの路線図を描く場合、実際の路線のつながり方に関するグラフとしての構造だけでは正しい路線図を描くには不十分である。グラフ構造に加えて局所的な接続順序構造と路線網内にできる閉路の構造とに関する幾何学的条件を導入し、３者をまとめて路線図条件と呼ぶ。路線図条件を満たした上で、任意性の残る部分にさまざまな要望を反映させるような図生成システムの構成を行なった。作業は路線図条件を満たすだけの図を生成する「平面への埋め込み」と、物理モデル上の緩和を利用した「整形」の２段階による。図生成の制御は不完全だが、正しい路線図を描くことはできる。

PLANAR GRAPH DRAWING FOR ROUTE-MAP

Eko Sakai　　Satoru Kawai

Department of Graphics and Computer Science, College of Arts and Sciences,
The University of Tokyo

A set of geometric constraints called 'route-map condition' is abstracted out, which identifies a class of route-map-like graphs on a plane.

We also present an algorithm that generates a variety of route-maps specified by the 'route-map condition'.

The algorithm consists of two successive steps:

(1) Construction of a pseudo route-map which satisfies given 'route-map condition'. The pseudo route-map realizes a planar placement of the graph satisfying the condition in triangular representation.

(2) Geometric transformation to meet other requirements. This transformation keeps the 'route-map condition' be satisfied.

Some discussion on our prototype system are included.

1　目的

この研究では抽象的な情報を可視化する問題の一つとして、たとえば地下鉄の路線図を、その路線網の抽象的な構造に基きつつ、その他の多様な要求に沿うようなさまざまな形を生成することを目的とする。路線網の構造のもっとも基本的な部分は、無向グラフとしての構造である。ただし、この構造だけでは図1のように、路線図としての同一性を保証し得ない。路線図としての同一性を保証するためには、グラフとしての構造に加えて、路線図上の各点におけるある種の局所的な位置関係を規定しなければならない。

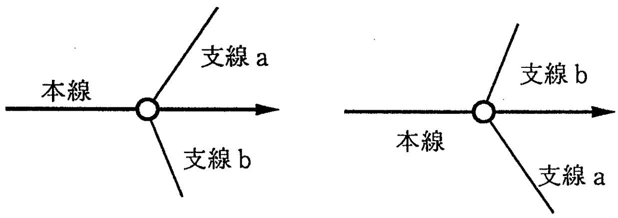

図 1: 局所的位置関係が保存されない例

本研究では、まず路線図としての同一性を保証する際に必要となる幾何学的位置関係について考察する。この幾何学的位置関係とグラフ構造を合わせて**路線図条件**と呼ぶ。局所的な位置関係の保存によって全体として路線図としての同一性が保証されるため、路線図条件の内容は局所的な条件づけの集合となる。

次に路線図条件に基づき、その他の種々の要望を満たす形で路線図を描く方法を考える。これによって、路線図としてはあくまで正当なものでありながら、たとえばある路線は起点から終点までまっすぐであるというような現実とは異なる主観を反映させた図を生成できる。

2　図構成の流れ

この研究では図2のように2段階の操作により描画を実現する。

まず第1段階で路線図条件だけを制約として駅と路線の配置を行なう。この操作は路線図条件を満たす平面上の配置を、有限領域内に三角形を敷き詰めるという形で実現するものである。この第1段階の操作を**平面への埋め込み**と呼ぶ。この段階で路線図条件を満たす図を一旦作り、中間生成図とする。なお、平面への埋め込みで一意な中間生成図ができるわけではない。中間生成図は三角形の敷き詰めの方法により多様な形で生成され得る。

中間生成図は路線図条件は満たすが一般にはわかりやすい図にはならない。そこで第2段階として図の整形を行ない、その際に種々の要求 — 例えばある特定の路線をまっすぐに表示するなど — を図に反映させる。この第2段階の操作によって最終的な路線図が生成される。

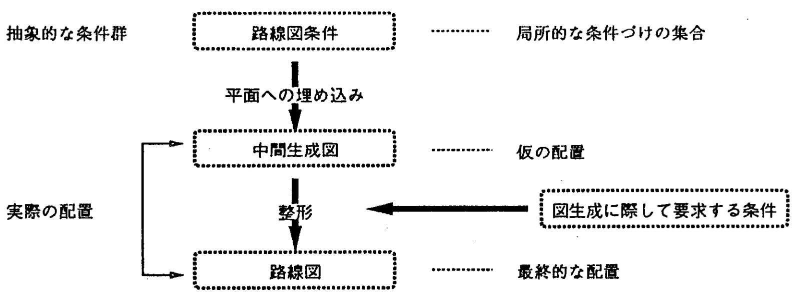

図 2: 路線図生成の枠組み

3 路線図条件

3.1 用語について

「同じ路線図である」ことを規定する路線図条件を説明する際に使用する用語・概念について明確にしておく。路線図の基本的な構造は無向グラフである。本研究では一般の無向グラフで頂点、辺と呼んでいるものをそれぞれ実駅、原始区間と呼ぶ（図3）。

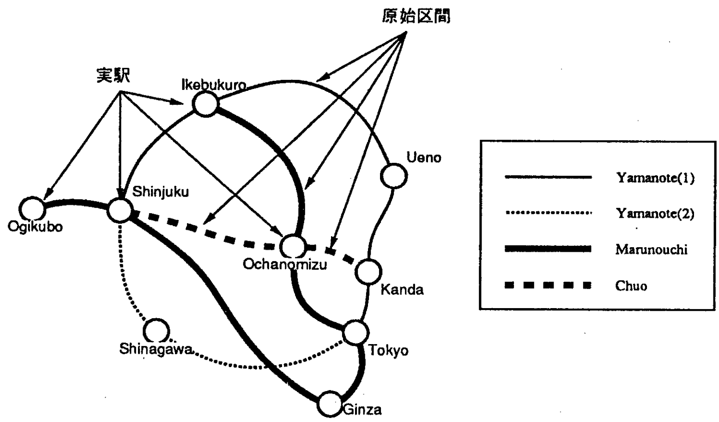

図 3: 実駅と原始区間

路線図を描く場合、実駅と原始区間を単位にして操作を行なえば良さそうだが、グラフとしては交わらない原始区間（辺）同士が図を描くと交わることになる、という場合にはこれでは不都合である。現実の路線網でいえば、立体交差しているところを表現する場合に相当する。

このような交差を処理するために、その交差地点に仮想駅というものを考える。ここでは処理を単純化するため、図表現した場合に交差が発生するすべての地点に仮想駅を設けることにする（図4）。そして、仮想駅の設置によって区切られた原始区間の各部分と、途中に仮想駅が設置されなかった原始区間とを合わせ、区間と呼ぶ。路線図条件を規定し、さらにそれに基づいて路線図を描くという作業に際し、上記のような交差に関する処理はすべて行なわれていることを今後では前提とする。そして、単に駅という場合、最初に述べた実駅と、原始区間の分割による区間の構成の際に設けた仮想駅とを合わせたものの総称とする。この駅と区間とが今後の処理の単位となる。区間構成の過程で路線図上の交わりのある箇所にはすべて駅が存在することになるため、駅を頂点、区間を辺とするグラフは平面的グラフとなる。

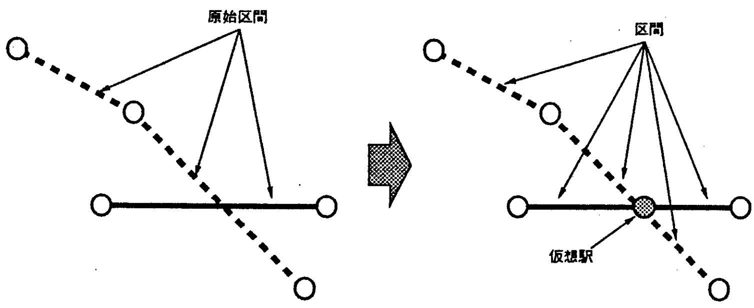

図 4: 仮想駅の設置と区間

3.2 路線図条件

この研究ではいくつかの路線図が「同じ」であることを規定する条件を路線図条件と呼んでいる。多分に感覚的な表現をとれば、ある路線図の駅、区間を連続的に移動させても、その過程で他の区間を「またぐ」ことがなければ、変形後の路線図は最初の路線図と同じ、すなわち路線図条件が一致している、とする。

路線図条件の内容を規定する上で例外的な処理をできるだけ排除するため次のような仮定をおく。

- 区間同士は交わらない。
- 区間、駅をそれぞれ辺、頂点とするグラフは連結である。
  連結でない場合には仮想的な区間を有限個導入して連結な場合に帰着する。
- 区間の両端は異なる駅である。

路線図としての同一性を規定する路線図条件として、グラフ構造に加えて次の2つについて考える。

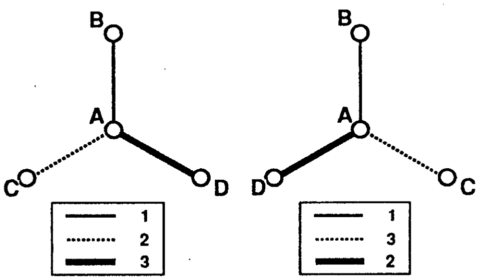

図 5: 接続順序の違う例

接続の順序　図5の左右の路線図が同じであるためには、グラフとしての構造が同一であることに加え、A駅に関してその近傍で、「Bへ向かう区間(AB)よりCへ向かう区間(AC)が左に」、「Cへ向かう区間(AC)よりDへ向かう区間(AD)が左に」、「Dへ向かう区間(AD)よりBへ向かう区間(AB)が左に」、それぞれなければならない。これらをまとめると区間ABから順に反時計回りに<AB, AC, AD>という順になる。この順序をAにおける接続の順序と呼ぶ。これが一致しなければ路線図として同一とはいえない。これによって図5の2つの路線図が区別される。もし路線図をグラフとして見た場合に木(tree)となる、つまり区間の組合せによって閉路ができないなら、この条件だけで路線図としての同一性を保証でき、異なる路線図を区別できる。

閉路の回る向き　図6のように路線図の中に（途中同じ駅を2度以上通ることのない）閉路ができる場合、前述の接続順序だけでは異なる路線図を区別できない。

本研究では閉路の回る「向き」に着目し、このような路線図を区別する。図6の例で言うと、A–B–C–AあるいはB–C–D–Bという閉路がIIでは左回りになっているのに対し、IIIでは右回りになっている。

すべての閉路の回る向きを記述するのは一般には冗長である。路線図全体のグラフとしての連結性を仮定しているため、生成される図の「もっとも外側」になる閉路の回る向きを示せば（逆回りになった時点で「もっとも外側」になり得なくなることもある）、後は接続順序条件から他の閉路の回る向きはすべて決定される。ただし、現段階で動作しているプロトタイプ・システムにおいてはインプリメント上の事情から、すべての閉路の回る向きを情報として与えている。

前述のように、路線図がグラフとして閉路を持たない場合には接続順序だけで、また閉路を持つ場合には接続順序および閉路の回る向きによって路線図としての同一性を保証できる。

1. グラフとしての構造 $(G = (V, E))$

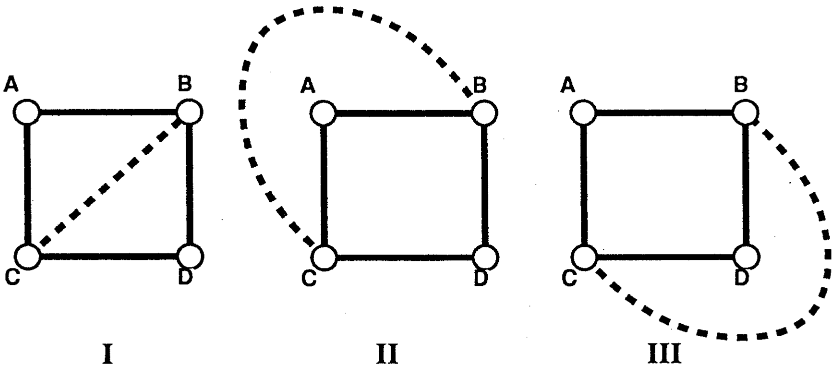

図 6: 接続順序による区別ができない例。I と II、I と III は区別できるが、II と III は区別できない。

2. 駅における接続の順序

3. 閉路の「向き」

をあわせて、路線図条件と呼ぶ。

4 平面への埋め込み — 最低限の配置

路線図条件に基づいてまず行なうのが平面への埋め込みである。グラフとしては同じものでもユークリッド平面 $\mathbf{R}^2$ 上への写し方によって幾何学的な違いを生じ得る。平面への埋め込みの際も路線図条件を考えた時と同様の仮定を置く。すなわち区間、駅によるグラフは連結な平面グラフとする。

平面への埋め込みは、区間とその両端となる駅を路線図条件を満たすように順次配置してゆくことで行なう。その際、一番最初に配置する区間は別として、二番目以降の区間は少くとも一方の端となる駅は既に配置されているものに限って順次配置することとする。つまり、グラフとしての連結性は配置処理の間保たれる。

まず平面上に多角形の領域を考え、この有限領域内に図を描くことにする。区間はすべて折れ線として配置すれば、駅、区間の曲がる点、描画領域を規定する多角形の頂点の相互間に互いに交わらない線分を引いて描画領域内全体を三角形に分割することができる。区間を表す折れ線を構成する線分を実辺、描画領域を三角形分割するために引いた線分を仮想辺と呼ぶ（図 7）。区間同士は交わらないから、区間を順次配置する際に実辺との交点が生じないようにする。

区間の具体的な配置の例として図 7の状態で区間 AB を配置することを考える。既に配置されている区間同様、AB も折れ線として配置する。A での接続順序は <AC, AB, AD>、B での接続順序は <BA, BC, BD> とする（どちらも反時計回り）。このとき、A の近傍で考えると（図 8） A における接続順序の条件を満たすために図 8中 right とある方に向かわなければならないことがわかる。区間 AB は実辺とは交わらないから、配置される折れ線 A...B は区間（折れ線） $\widehat{AC},\widehat{CB},\widehat{BD},\widehat{DA}$ を周（境界）とする多角形領域の内部に収まる。より具体的には図 9のようにする。つまり、A を頂点に持つ三角形から始めて、B を頂点に持つ三角形まで、仮想辺を共有する三角形を順にたどってゆく。そして、三角形をたどる際に手がかりとした仮想辺上の点 P_i を適当に（両端は除く）とり、$A, P_1, P_2, \ldots, P_n, B$ と順に線分で結んでゆく。最後に折れ線 $AP_1P_2\ldots P_nB$ の配置後の B での接続順序条件と、閉路条件を確認する。この例では遠回りする以外に A と B の他の結び方はないが、A,B での接続順序条件がそれぞれ <AC, AD, AB>, <BA, BD, BC> となっている場合、図 9の route 1 をとるか route 2 をとるかで閉路条件が違ってくる。

以上まとめて、区間配置のアルゴリズムは表 1のようになる。

次に図 7で一端 E がまだ配置されていない区間 AE を配置することを考える。A における接続順序条件は <AC,AE,AD> とする。この場合も区間（折れ線） AE が実辺が境界となる多角形領域内に収まるこ

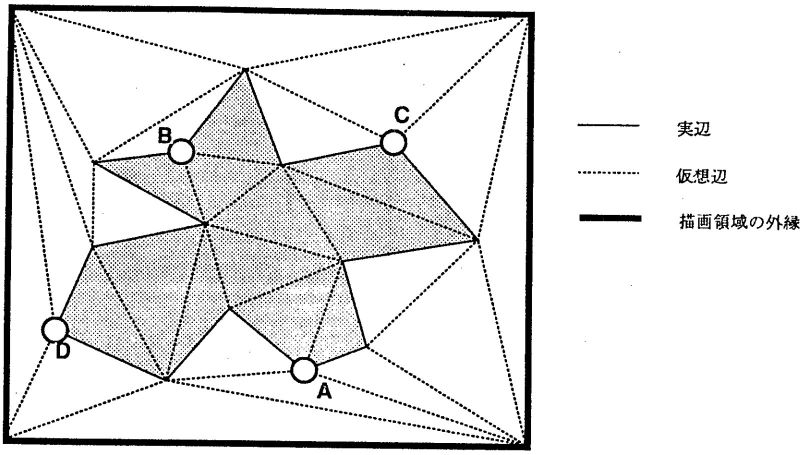

図 7: 描画領域の分割と実辺、仮想辺

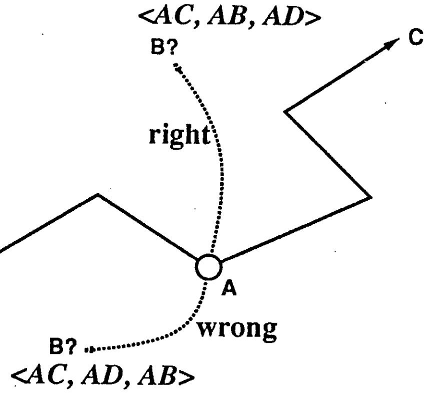

図 8: A の近傍での区間 AB

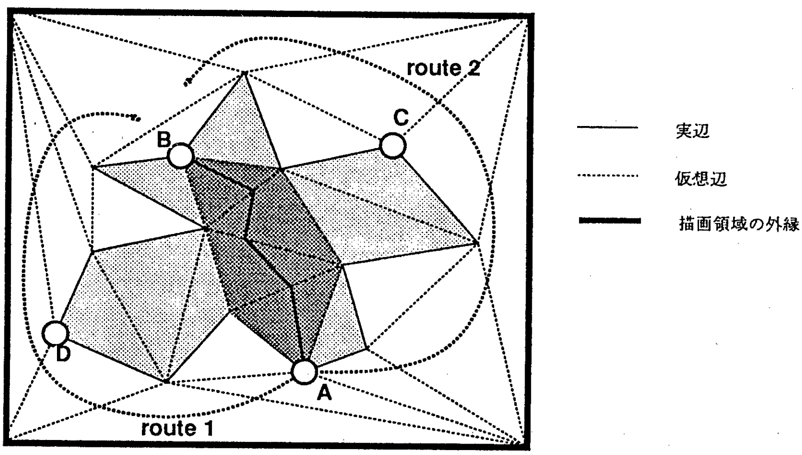

図 9: 区間の配置

ALGORITHM 区間の配置

Input: A: 区間の起点, B: 区間の終点, 実辺、仮想辺、領域の外縁とこれらを辺とする三角形, 路線図条件

Output: 折れ線 $AP_1P_2\ldots$

BEGIN

A を頂点に持つ三角形 T_1 をさがす。ただし、T_1 は内部の任意の点 X をとって AX を結ぶと A における接続順序が <AB, AX, AC> となる三角形。

T_1 の 3 辺を E_{11}, E_{12}, E_{13} とする。

FOR i := 1 **TO** 3 **DO BEGIN**

  IF E_{1i} が仮想辺 **THEN BEGIN**

    E_{1i} 上の両端以外のある点を P_1 とする。

    T_1 と E_{1i}、折れ線（線分）AP_1 を *PROCEDURE* 三角形探索に与えて折れ線 l を得る。

    IF l が A と B を結ぶ折れ線 **THEN EXIT**(l を返す)

  END

END

配置不可能！（不当な路線図条件）

END

PROCEDURE 三角形探索

Input: 三角形 T_i とその一辺 E_i、折れ線 $AP_1\ldots P_i$

Output: 折れ線 $AP_1\ldots$

BEGIN

T_i と仮想辺 E_i を共有する三角形を T_{i+1} とする。

T_{i+1} の E_i 以外の 2 辺を $E_{i+1,1}, E_{i+1,2}$ とする。

FOR j := 1 **TO** 2 **DO BEGIN**

  IF $E_{i+1,j}$ が仮想辺 **THEN BEGIN**

    IF 区間の終点 B が T_{i+1} の頂点 **THEN**

      IF 折れ線 $AP_1\ldots P_iB$ を配置した時、路線図条件が崩れない **THEN**

      EXIT($AP_1\ldots P_iB$ を返す)

    IF $P_1,\ldots,P_i$ がいずれも仮想辺 $E_{i+1,j}$ 上にない **THEN BEGIN**

      $E_{i+1,j}$ 上の両端以外の任意の点を P_{i+1} とする。

      T_{i+1} と $E_{i+1,j}$、折れ線 $AP_1\ldots P_iP_{i+1}$ を *PROCEDURE* 三角形探索に与えて折れ線 l を得る。

      IF l が A と B を結ぶ折れ線 **THEN EXIT**(l を返す)

    END

  END

END A と B を結ばない折れ線として A を返す。

END

表 1: 区間配置のアルゴリズム

とがわかるので、E をこの領域内の適当な点に置けば前述の例の場合に帰着される。
中間生成図の例を図 10に示す。

5 図の整形

三角形分割の一つの実現としての中間生成図は作り、それを整形することによって路線図を描くという方法をとっている。しかし中間生成図は路線図条件を満たすというだけのもので、またこの段階では図の制御は行なわれていない。そこで第2段階で図に対する要求を満たすよう図の整形を行なう。

整形は、各区間を弾性体のようなものでできた棒であるとみなした一種の物理モデルにおける緩和現象を利用する方法を採用している。現状では満足できる整形を行なうには至っていない。

整形の例を図 10に示す。

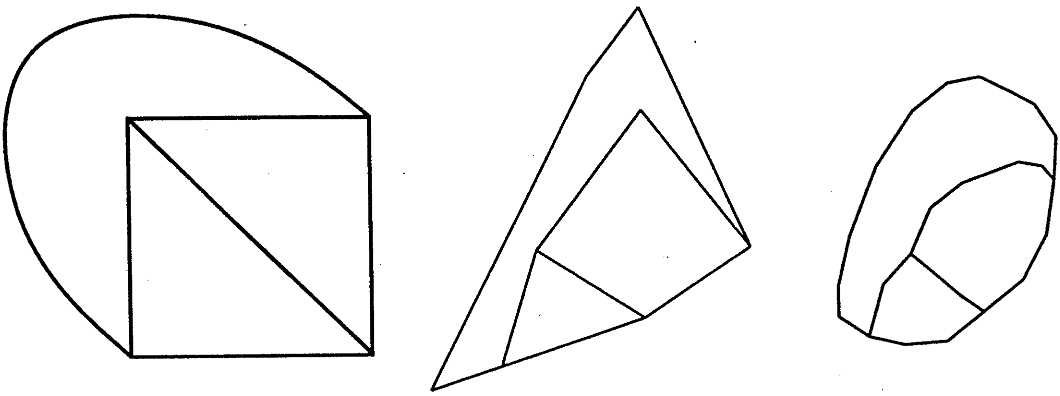

図 10: 路線図生成例 左から 概念図、中間生成図、整形した図

6 問題点

現在、路線図条件に基づく中間生成図まではほぼ問題なく作ることができる。ただし、路線図条件の扱いには以下のような問題点もある。

- 路線図条件の内容が繁雑になる。
- 路線図条件が条件として厳しく、「どちらでも良い」状況が扱えない。

研究の後半部を占める整形に関しては現状では不満足な結果しか得られていない。より有効な整形は今後の課題である。

参考文献

[1] Tomihisa Kamada : VISUALIZING ABSTRACT OBJECTS AND RELATIONS, Series in Computer Science – Vol.5, World Scientific (1989).

[2] Godfried Toussaint : An Output-Complexity-Sensitive Polygon Triangulation Algorithm, CG International '90, Springer-Verlag, pp.442-466 (1990).

[3] 酒井 恵光 : 制約条件に基づくグラフの視覚化 — 路線図の描画に関する研究, 東京大学 教養学部 基礎科学科第 2 卒業研究 I (1990).