

線分間結合に制約のある描線順序の近似最適化†

雄山真弓** 阿部健一†††

平面（または空間）に散在する複数の点あるいは線分を一巡する効率的な描線順序を定める際、一般には特定の点あるいは線分同士が結合できないなどの制約を考慮しなければならない。例えば、NCロボットによるプリント基板の効率的な作成のための数値制御データを求める問題などがそうである。本報告は、作図を構成する点や線分に種々の結合制約が与えられた場合の描線順序の効率化について、1) 結合制約を考慮した作図データの構成、2) 結合制約の意味および具体例と結合制約の表記法、3) 結合制約のある作図データの効率的描線法、4) その方法を用いた計算機実験について報告する。

1. ま え が き

平面または空間に散在する複数の点や線分をすべて1回通り、連結する軌跡のうちその軌跡が最短なものを求めることは、組合せ最適化の問題として扱うことができる。しかし、点や線分の数が増加するに従ってその組合せは増大し、一般には計算機を用いても処理に膨大な時間がかかる。したがって、最適解に近い近似解を有効的に求めるヒューリスティックな方法が研究されてきた^{1)~4)}。

平面の場合については、種々研究がなされ、諸手法の評価もなされている⁵⁾。著者は、空間へも拡張できる段階的クラスタ化法を報告してきた^{6), 7)}。さらに、その拡張として、点や線分を連結する過程で、結合上の制約条件がある場合の軌跡の最適化についても研究を行ってきた。結合上の制約条件を持つ軌跡の最適化は、その応用範囲が広いにもかかわらず、あまり報告がなされていない。これは、制約条件を持つデータを扱うことが軌跡の最適化を行うことをより複雑化することに起因する。

本報告では、線分や点に与えられる複雑な結合上の制約をとりあげ、それらを表現する作図データの構成法についてのべる。次に、これらの作図データを用いて効率的軌跡を求める方法について検討し、著者が先に提案した段階的クラスタ化法^{6), 7)}に制約条件の処理を組み入れた方法を提案する。さらに、この方法が制約条件を持つ様々な作図データの効率的軌跡生成に優れた結果を与えることを計算機実験の結果で示す。

2. 結合制約を考慮した作図データの構成

2.1 作図データの構成

図形Gはk本の直線分からなるものとする。以下、点も両端点が一致する線分の特殊な場合と考慮して、図形の構成要素に含める。端点の総数をnとする($n=2k$)。各端点は、座標値と直線分を構成する他端点の番号に相当するパラメータbとで次のように書き表す。

$$p_i = (x_i, y_i, z_i, b_i) \quad (1)$$

$$i = 1, 2, \dots, 2k$$

端点 p_i と p_j が1直線分の両端点を構成するなら

$$b_i = j \quad (2)$$

である。

2つの端点の結合距離をそれら2点を結ぶ直線分の長さあるいはその定数倍であるとする。そこで、図形Gの線分をすべて1回通り、端点間の結合距離の総和が最小となる描線順序を求めたい。このとき、端点間結合にそれぞれ固有の制約があり、端点の連結が全く無制限には行えないとする。

端点間結合の制約としては様々なものが考えられるが、本文では、それらの制約が、各端点 p_i に付帯する制約条件 $r_{ij}, j=1, 2, 3, \dots, u_i$ (u_i : 制約の数) で表されるものを扱う。その意味、具体例および表記法については、以下の節で述べる。

以上より、作図データは、端点を基準にして、

- (1) 入力順に与える端点のデータ番号: i
- (2) 座標値: x_i, y_i, z_i
- (3) 線分のお他端点のデータ番号: b_i
- (4) 制約条件: $r_{ij}, j=1, 2, 3, \dots, u_i$

で与えられる。

2.2 結合制約の意味および具体例

端点間の結合制約とは、端点を結合していく際に、ある端点同士が直接には結合できない、あるいは、端

† Approximate Optimization of Drawing Sequence on Restricted Connection between Segments or Points by MAYUMI OYAMA (Information Processing Research Center, Kwansai Gakuin University) and KENICHI ABE (Department of Information Technology, Toyohashi University of Technology).

** 関西学院大学情報処理研究センター

††† 豊橋技術科学大学情報工学系

点に属性値が与えられていて、2つの端点の属性値の組合せがある条件を満たさないと結合できないとか、2つの端点の属性値の組合せで端点間の結合距離が変化する、などを意味する。

例えば、移動ロボットを使い平面上に複数個の荷物を決められた位置に効率的に置いていくような問題を考える場合、ある決められた位置を始点または終点にしたいとか、平面上のある位置に障害物がある等が、作図上での制約となる。前者の場合は、他のどの点からも始点には結合できない。また、終点からは他のどの点からも結合できないという制約に言い換えることができる。後者の場合は、障害物を挟む2点間の結合の際に、障害物を迂回する経路の長さを距離としなければならない。

以下の例は、例えばプリント基板上に電子部品を自動装着するためのスケジューリング問題の単純化したモデルとなっている。後の章で計算機実験の際に例題として用いる。

いま、平面に3種類の部品(O, *, Δ)を決められた位置に1個ずつ置いていく作業を考える(図1)。図の各点の記号は、そこに置くべき部品の種類である。3種類の部品はそれぞれ必要な数ずつ、ある間隔で直線状にトレー(図の右下)に載っているとす。平面上をトレーを滑らせて(あるいは、下の台を左右、上下に動かして)すべての部品を次々に置いていく。このとき、図2に示すように同じ座標軸であっても、置くべき部品の種類により、その移動距離が最短なトレーの動きは異なる。

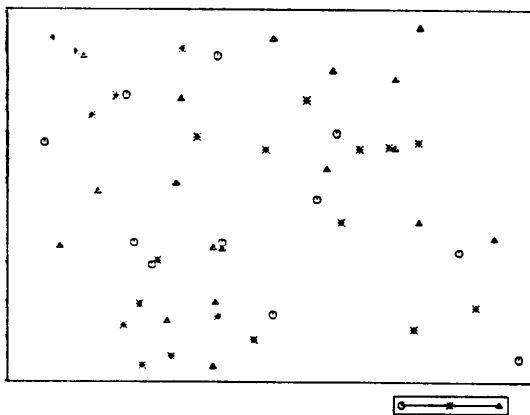


図1 3種類の部品(O, *, Δ)を指定の位置にトレー(図右下)を移動して配置するイメージ図

Fig. 1 The plane is composed by moving the tray at the lower right between the fixed positions of three kinds of elements.

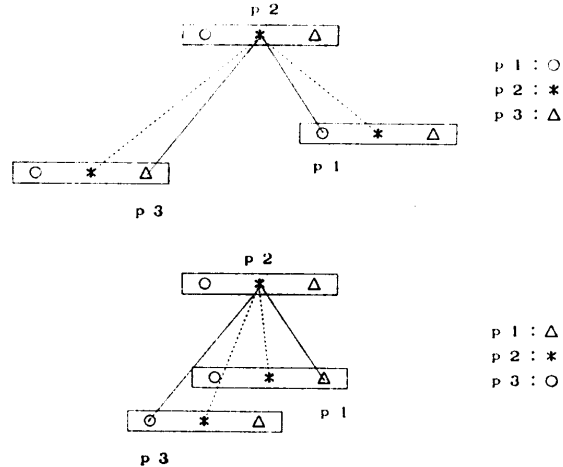


図2 同一座標値に異なる部品を配置した場合のトレーの動き

点線: トレーの動き 実線: 座標の位置関係
Fig. 2 The movement of the tray between 3 points in which each case is composed of different elements.

2.3 結合制約の表記法

ある線分の端点における1つの制約 r を

$$r = \{R, w_1, w_2, w_3, \dots\} \quad (3)$$

で表す。ここに、 R は結合の意味を表す識別子であり、 w_1, w_2, \dots は、 R に必要なパラメータである。例えば、 $R=01$ によって、端点 p_i からは直接結合できない端点であることを意味するとした場合、

$$r_i = \{01, j_1, j_2, \dots\} \quad (4)$$

によって、その端点からデータ番号 j_1, j_2, \dots 等の端点とは結合できないことを表す。後の計算機実験では、結合不可の条件が対称 (p_i から p_j への結合ができないとき、 p_j から p_i への結合も不可) な場合を扱う。

また、 $R=02$ によって、両端点のパラメータがある条件を満たさないと結合できないことを意味するとして。例えば、2点 p_i, p_j の制約 r_i, r_j を

$$\begin{aligned} r_i &= \{02, w_1, w\} \\ r_j &= \{02, w_2, w\} \end{aligned} \quad (5)$$

とする。また、 w_1, w_2, w は整数値をとるとする。そこで、

$$|w_1 - w_2| \leq w \quad (6)$$

のときのみ、 p_i と p_j は連結できるとする。このような制約は次のような作図データに適用できる。いま、 w_1, w_2, \dots を色を表す属性値とし、1から5の色で平面上を塗り分けていくとする。色替えに時間がかかるため、1の色が終わってから2の色へ、という順で、

それぞれの色が途中で混ざらないようにして塗っていく。そのためには、 $w=0$ とすればよい。

前節のトレーを使った部品置き例题は、結合距離の再計算を必要とする典型的な例である。トレー上の部品 \circ と $*$ 、および $*$ と Δ の間の間隔は等しいとし、それを d とする。また、トレーは、その移動に際し常に x 軸に平行であるとする。点 p_i についての制約 r_i を次式のように表す。

$$r_i = \{03, w_i\} \quad (7)$$

ここで、 R を仮に 03 とした。 w_i は、1, 2, 3 のいずれかの値をとり、それぞれの部品 \circ , $*$, Δ を表している。点 p_i で部品 w_i を置き、次に点 p_j にトレーを移動させ、そこで要求している部品 w_j を置くとする、 p_i の座標値を次のように変えて、 p_i との結合距離を計算する。

$$(x_i + d(w_j - w_i), y_i) \quad (8)$$

以上の制約条件の表記法によって、さまざまな制約が表せる。例えば、上記に述べた制約のほか、(1) 特定の線分間で結合できない場合、(2) 端点間結合に順序制限がある場合、(3) 端点間結合が与えられた角度の範囲でのみ許される場合、などが考えられるが、その詳細は省略する。

3. 結合制約を考慮した段階的クラスタ化法

3.1 段階的クラスタ化法の概略と結合制約の考慮

先に発表した段階的クラスタ化法^{6),7)}の概略を述べ、本方法に結合制約の処理をどのように組み込むかを示す。

段階的クラスタ化法では、まず、全端点をクラスタに分け、各クラスタごとに、その中に2端点が含まれる線分についてのみ総あたり法で最短結合を行い線分を連結する。連結していない端点をさらにクラスタ化し、同様に連結する。以上を端点数が少なくなるまで繰り返し、最後にクラスタ同士の接続順序を総あたり法で最短結合し、全体の描線順序を求める。

各段階でのクラスタは、次のように行う。空間(または平面)に散在する端点をすべて覆うように作図範囲を定め、これを総端点数 n にパラメータ λ をかけた $n \cdot \lambda$ 個のセル(形状は立方体)に分ける。セルを順次サーチして端点の有無を調べ、同一のセル内、あるいは隣り合うセルに含まれる端点同士は同じクラスタに属するとする。一般には、 λ の値を大きくすると、セルが小さくなって端点を含まないものが増えるため、クラスタの数が多くなる。反対に、 λ の値を小

さくするとクラスタ数は少なくなる。パラメータ λ の初期値は、端点数、端点の分散、各線分の長さなどを考慮して決定すれば良いが、計算量が増大するため経験的値を与える。 λ の初期値が適切でない場合、例えば1つのクラスタに含まれる端点数が多すぎるなどクラスタ化が適切でない場合は、後述のように λ の値を再設定し、クラスタ化をしない。

各クラスタでの最短結合は、線分の2端点と同じクラスタに含まれる場合とそうでない場合とで結合方法を変え、また軌跡がループを作らないように行う。このようにして連結できなかった端点に対し次段階クラスタ化を行う。

端点に結合制約がある場合は、クラスタ内端点の組合せにおいて結合の可否を調べる。クラスタ内で連結できない端点は、クラスタ化を段階的に行う中で順次連結する。

以上をアルゴリズムの形でまとめると次のようになる。ただし、結合制約の処理以外は簡単に記述している。詳しくは文献7)を参照されたい。

[1] 作図データの入力

作図を構成する端点のデータ番号: i , 座標値: x_i, y_i, z_i , 線分その他端点のデータ番号: b_i , 制約条件: $r_{ij}, j=1, 2, 3, \dots, u_i$, を入力する。

[2] 制約 r_{ij} の変換

各端点の $r_{ij} = \{R, w_1, w_2, \dots\}$ を R 値によって以下のように変換し[8]のための準備を行う。

1) 端点 p_i に結合できない端点のデータ番号 $c_{ij}, j=1, 2, \dots$ を求める。

2) 端点 p_i の結合可否が、組み合わせられる端点の属性値等で決定される場合、例えば、距離の再計算や各端点の持つ属性値の組合せによって判定を行ってから結合が決められる場合、それらの処理を行う処理ルーチンを呼び出す必要がある。ここでは、それぞれの処理ルーチンを呼び出すためのパラメータを生成する。これを $f_{ij}, j=1, 2, \dots$ とする。

処理の効率化を考慮して1), 2)で生成される c_{ij}, f_{ij} は、端点 p_i のレコードに組み込まれる。

[3] パラメータ λ の初期値の設定

[4] 作図範囲の決定

[5] クラスタ化のためのセルの決定

[6] クラスタ分けとクラスタ数の決定

[7] クラスタ化が不適切ならパラメータ λ を再設定(後で詳述)し、[5]へ、

[8] 各クラスタ内での最短結合

各クラスタ C_k ($k=1, 2, 3, \dots, q$) 中の線分の連結関係を調べる。 C_k に1線分の2端点が共にある端点を A グループ、1端点のみならば(他のクラスタに1端点がある) B グループとする。

端点の組合せは、以下のルールで行う。

(a) A グループの端点同士、A グループの端点と B グループの端点は連結できる。

(b) B グループの端点同士は結合できない。

(c) (a), (b) の条件で結合できる端点のうち、それぞれの端点をもつ c_{ij}, f_i の値を調べて結合する。

1) [2]の1)で求めた c_{ij} の扱い

結合組合せを行う2端点の p_i, p_j が c_{ij} , ($j=1, 2, \dots, u_i$), c_{ij} , ($j=1, 2, \dots, u_i$) を持つ場合、

$$c_{ij} \neq s, j=1, 2, \dots, u_i$$

かつ

$$c_{ij} \neq t, j=1, 2, \dots, u_i$$

ならば、 p_i, p_j は結合できる。

2) [2]の2)で求めた f_i の扱い

端点 p_i, p_j に f_i, f_j がある場合は、相当する処理ルーチンを用いて座標値を計算し端点間距離計算を行う。

以上のルールに従ってクラスタ内の端点結合を q 個のクラスタについて行う。

[9] 残る端点数の判定(残る端点数が十分少ないならば[10]へ、多い場合は[4]へ)

[10] クラスタ間結合

残る端点について総あたりで制約を考慮し、最短結合を行う。

[4]から[9]の間を1段階として[9]の条件によって繰り返す回数が段階数となる。[10]では未結合の2端点が残るまで端点結合を行う。結合制約を考慮した場合のアルゴリズムの違いは、第1は、入力時の端点データに制約値を含ませ、結合制約値を変換ルーチンを通して分類し、端点組合せ時に計算が複雑にならないように配慮していること。第2は、クラスタ内の端点組合せにおいて、結合制約を端点同士が直接結合できない場合と、距離や組合せの再計算を必要とするものに分けて行っていることである。制約がないデータでは、軌跡が必ず得られることが保証されるが、制約条件が非常に厳しい場合や、論理的に結合できないような制約を持つデータでは、段階化の途中でどれとも結合ができない状況になることも考えられる。この場合は、なんらかのチェックルーチンを用いて明らかにする必要がある。

[7]のパラメータ λ の再検討については、クラスタ分けを行って求めたクラスタの数 q が、1) 1に近い値、2) 端点数 n に近い値、3) 前段階のクラスタ化時の端点数 n が変わらず(結合が行われていない) q の数も変わらない、などの場合は、パラメータ λ を(9), (10), (11)式に表すように、再設定する必要がある。

$$n - q \leq \varepsilon \quad \text{ならば} \quad \lambda = \lambda - \alpha \quad (9)$$

$$q - 1 \leq \varepsilon' \quad \text{ならば} \quad \lambda = \lambda + \alpha \quad (10)$$

$$n' = n \quad \text{で} \quad q \geq q' \quad \text{ならば} \quad \lambda = \lambda - \alpha \quad (11)$$

ここで、

q : クラスタの数, n : 残る端点数

q' : 前段階のクラスタの数,

n' : 前段階の端点数

ε の値は、ほぼ0に近い値を与えるが、 ε' の値は、端点数 n が数千の場合は、数十のオーダーにする必要がある。端点数が数百の場合は、1~10を用いる。

α の値は、端点数の多い場合と少ない場合、端点の散らばり具合によって変化させる必要があるが、本方法では、 λ の値(実験では、しばしば $1 \leq \lambda \leq 5$ を用いた)を1割~5割増加した値となるようにした($0.1 \leq \alpha \leq 0.5$)。

3.2 本方法の利点

制約条件を考慮した段階的クラスタ化法の利点として以下のことを挙げることができる。

(1) 時間計算量の効率化

時間計算量は、入力データ n に対して少なくとも多項式、できれば $O(n)$, $O(n^2)$ 時間程度であることが必要である。ここで述べた段階的クラスタ化法は、分割統治法の応用であり、段階数を多くすれば、時間計算量は減少する。段階数の増加は、クラスタ化の回数を増加する。しかし、本方法でのクラスタ化は、端点の数にパラメータ λ を掛けた数のセルを探す手間で行うことができ、時間計算量の大幅な増加にならない。したがって段階数が少ない場合でも、 $O(n^2)$ のオーダーで処理できる。

制約条件が多くなると、制約条件に合うかどうかの組合せ計算時間が大きくなるが、端点同士の組合せ数を大幅に減少させたと同様に端点のもつそれぞれの制約についての比較数も、段階的処理を行うことによって、当然減少することは明らかである。

(2) 複雑な作図データの分類と領域計算量の節約

制約条件が複雑にならないように、データの入力時とクラスタ内組合せにおいて工夫を行っている。例え

ば、作図データの種々の結合制約について、入力データ側では、制約値 r として扱いやすくしている。さらに、入力後は、直接に結合できないデータを c_{ij} とし、変換ルーチンを用いて結合が決定されるものは、 f_{ij} として処理の簡潔化を図っている。 c_{ij} も f_{ij} も点データのレコードに組み込まれる。したがって、領域計算量は、端点の座標値をふくめて端点総数の数倍で処理できる。

(3) パラメータ λ によりクラスタ再編成が容易

クラスタを作る場合、端点座標の最大値と最小値を用いてその範囲内にある端点をクラスタ化している。端点の散らばりが均質でなく、極端な偏りを持つデータでは、各クラスタの中に含まれる端点数に大きな差ができる場合がありうる。このため端点数の多いクラスタをそのまま最短結合することは、組合せ時間が大きくなり効率的でない。そこで、パラメータ λ の値を

変えることによって、クラスタの再編成を行うことができるようにしている。

(4) 種々の結合制約への対応が可能

結合可否が、端点および結合する端点との関係のみで決定される場合は本方法で対応できる。また、クラスタ化処理を工夫することで、さらに複雑な結合制約の場合にも本方法が適用できる⁶⁾。

4. 計算機実験と結果

実験結果を評価する値は、最適化率 (P) と計算時間 (CPU-TIME)、最適化前後の軌跡表示がある。最適化率 P は(12)式で表される⁶⁾。

$$P = 100 \{(\sigma_{in} - \sigma_{out}) / \sigma_{in}\} \quad (12)$$

σ_{in} : 最適化を行う前の線分間移動距離 (作図部分を含まない) の合計

σ_{out} : 最適化を行った後の線分間移動距離の合計

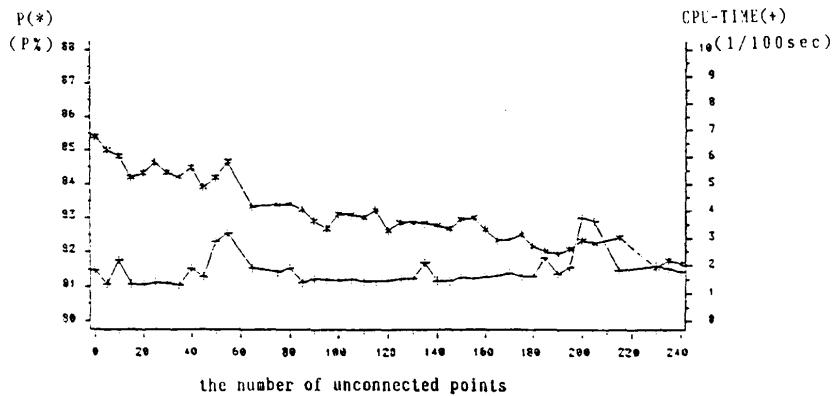


図 3 端点間結合不可数を増加させた場合の最適化率と CPU-時間の関係 (端点の総数=300 の 3 次元データ, 結合不可数=0~240)

Fig. 3 The relation between optimized percentage and CPU-times in the case that the number of unconnected points is augmented.

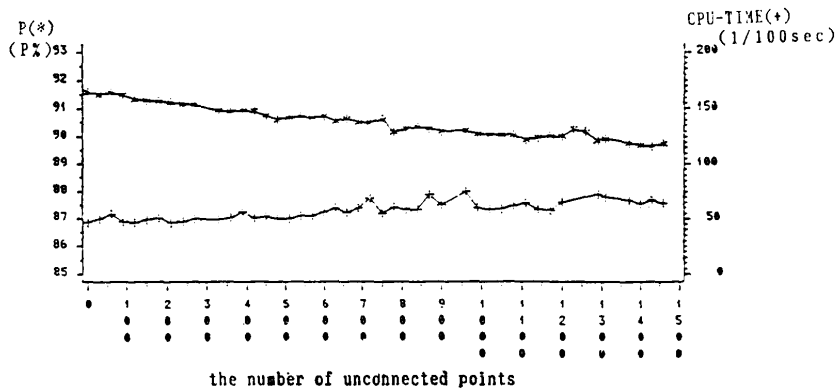


図 4 端点間結合不可数を増加させた場合の最適化率と CPU-時間の関係 (端点の総数=3,000 の 3 次元データ, 結合不可数=0~1,500)

Fig. 4 The relation between optimized percentage and CPU-times in the case that the number of unconnected points is augmented.

処理に使用した計算機は、HITAC M-680 D である。

4.1 結合制約 $R=01$ の作図データに対する実験結果

端点間に結合不可の条件を与え、結合不可数を順次

増加した場合の最適化率と計算時間の関係を図 3、図 4 に示している。図 3 は、端点数 300 の 3 次元データについて、端点間の結合不可数を横軸にとり、これを 0 から 240 まで増加した場合の最適化率 ($P\%$) と計算時間 (CPU-TIME (10^{-2} sec)) の関係と同じグラフ

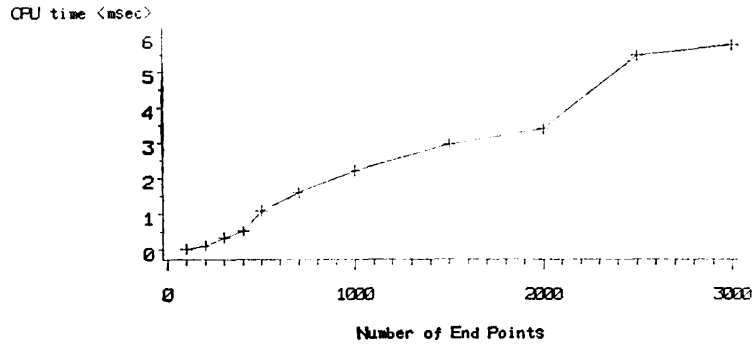


図 5 端点の数を増加した場合の処理時間の関係 ($R=01$ の結合制約を端点総数と同数与えた場合) ($\lambda=4.0$ の場合)
 Fig. 5 Relation between the number of end points and the CPU-times ($\lambda=4.0$).

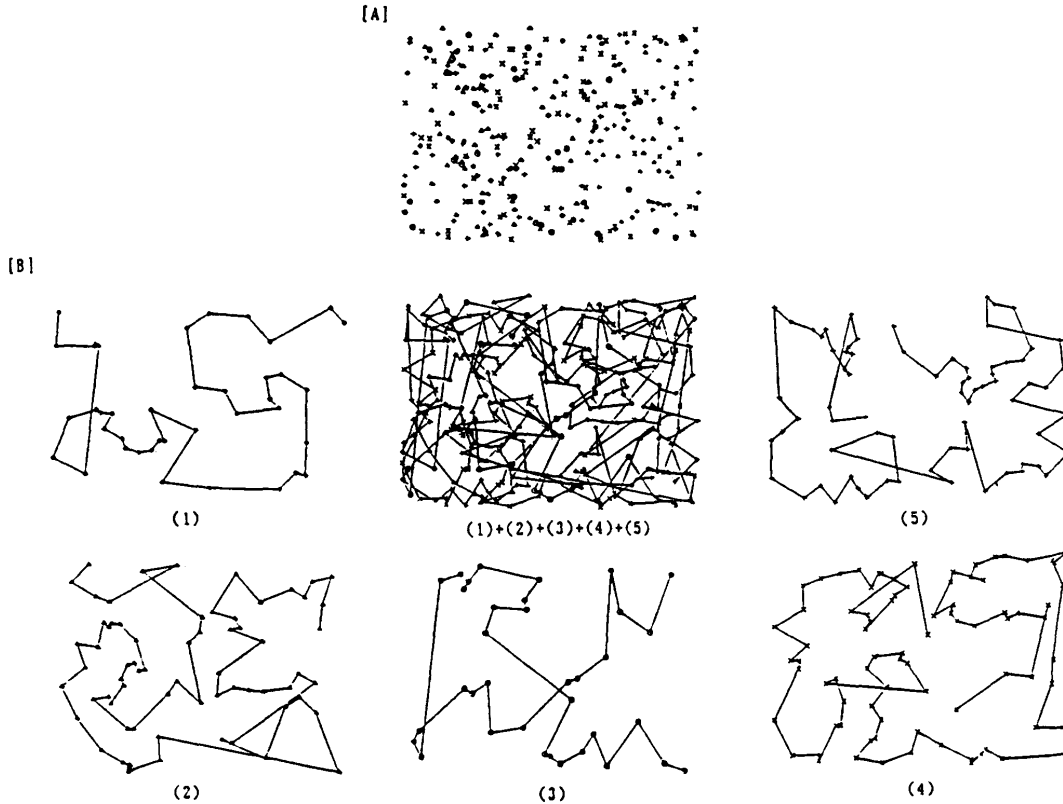


図 6 5 色で構成される点座標を色別に効率的に塗り分けた例
 [A]: 色属性を持つ入力点座標, [B]: 色別 (1)~(5) の描線軌跡
 (1)+(2)+(3)+(4)+(5) で全体の軌跡
 Fig. 6 An example of a plane composed of five colored elements in which the different colors are painted in the most effective method.
 [A]: The coordinates of the colored elements.
 [B]: The locus of the drawn for each individual color.

上に示している。結合不可数の増加によって最適化率は減少し、計算時間は、増加している。

一方、図4は、端点数3,000について結合不可数を0から1,500まで変化させて同様に行った結果である。端点数の増加によって、入力データの軌跡のうち、作図を行う線分間移動距離の合計 σ_{in} が大きい、このため P の増減が小さくなっている。

結合不可数が0のときに、最適化率が最大を示している。端点数を100から4,000まで種々に変えて測定したが、いずれも結合不可数と最適化率、計算時間の関係に同様な傾向がみられた。

結合不可の端点の組合せは、結合不可数が0で最適化率が最大となる端点組合せのうち、短い組合せを上位から選び出して行った。

図5は、端点数と同数の結合制約を各図形に与えた場合の処理時間の変化をプロットしたものである。横軸に端点総数(制約は端点と同数を与えている)、縦軸は処理時間(10⁻¹msec)を示している。パラメータ λ の初期値は4で行った。処理時間が直線関係で増加し、端点の増加による爆発的な増加が起こらないことを示している。

4.2 結合制約 $R=02$ の作図データに対する実験結果

図6は、 $x-y$ 平面上の300の点を、それぞれ決められた5種類の色で塗り分けた例を示す。同じ色同士を結ぶ効率的経路が求められ、これが5種類の色について行われている。一般には、データをあらかじめ色別に分けて行う方法も考えられるが、本方法は、全データに $r_{ij} = \{02, w_1, w\}$ を与え $w=0$ として処理を行っている。結果を見やすくするために、色ごとに分けた図と、重ねた場合の図で示す。点の数は300、縦、横の大きさは、2,000、3,000、属性値(色、部品など)の種類は5、計算時間は、 0.1056×10^{-1} secである。

4.3 結合制約 $R=03$ の作図データに対する実験結果

(a) 端点に与えた属性によって結合に影響する例

図7は、2.2節の具体例で示した3種類の部品(○, *, Δ)を指定した位置に効率的に置いて行く例の実験結果を示す。2.3節の結合制約の表記法で述べたように、 $r_{ij} = \{03, w_1\}$ である。 w_1 は3種類の部品を数字で与えている。

3種類の部品を持つ点データの配置を示す関係を図7[A]に示した。[B]は、入力データの軌跡を示している。[C]は、各点の属性値を満足しながらトレーの

移動距離の最適化を行った結果を示す。点の数は300、縦、横の大きさは、2,000、3,000、トレー上の3つの部品の間隔は300である。最適化率は86.75%、計算時間は 0.1285×10^{-1} secである。

(b) 結合制約が与えられなかった場合との比較

図8は、(a)の制約条件のあるデータと、同じ位置

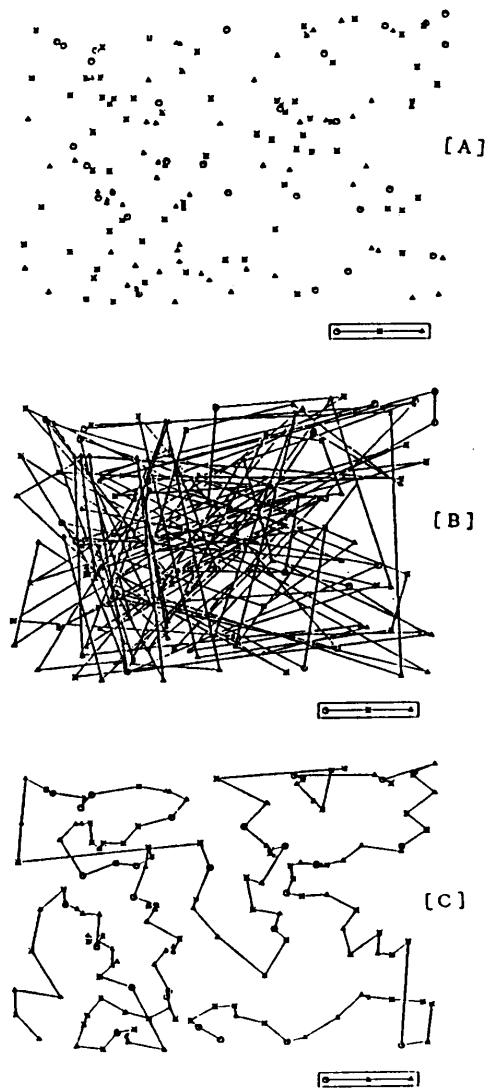


図7 端点結合距離に変化を与える結合制約($R=03$)の処理例

[A]点の配置と3種類の部品の関係、[B]入力データの点順序、[C]最適化を行った点順序。

Fig. 7 An example in which a restricted connection ($R=03$) is applied to the distance of end point connections.

[A]The relation between the coordinates of the 3 types of elements, [B]The sequence of input data, [C]Optimization of the resulting sequence.

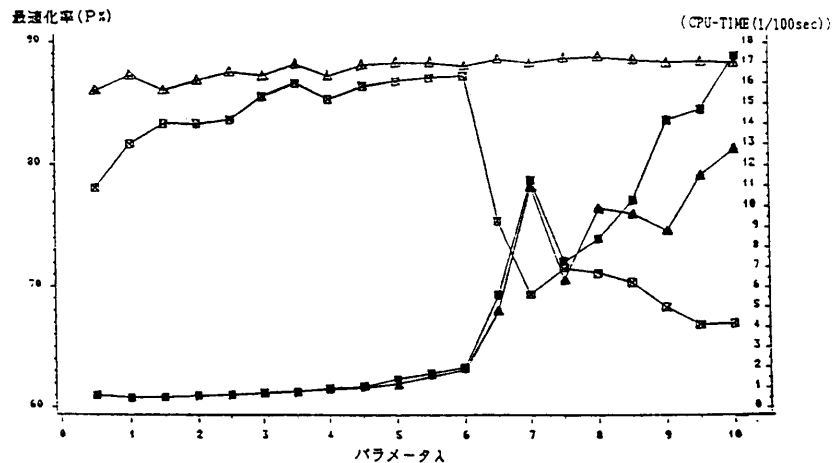


図 8 点に $R=03$ の結合制約を持つデータ (□, ■) と、持たないデータ (△, ▲) の比較
パラメータ λ の変化と最適化率, CPU-時間の関係
□, △: 最適化率 (P%), ■, ▲: CPU-時間 (1/100 sec).

Fig. 8 A comparison of the data in which the restriction of $R=03$ was applied and the data in which there was no restriction. The relation between the change in parameter λ and the rate of optimization and the CPU-time □, △ are the rate of optimization (P%) ■, ▲ are the CPU-time (1/100 sec).

にすべて同じ部品を置いていく場合 (制約条件を持たない場合について) の 2 種類のデータについて、パラメータ λ を変化させた場合の最適化率と処理時間の関係を比較したものである。パラメータ λ の値の増加によって、最適化率と、処理時間が、制約のあるなしで異なる傾向がみられることを示している。処理時間は、パラメータ λ を 6 以上にすると、いずれの場合も増加してくる。また、最適化率は、制約条件をもつ場合は、パラメータ λ を 6 以上に設定すると急激に減少している。これは、距離属性が端点の組合せの時点で結合距離を変化させることに起因する。 λ の増加によって各クラスタ中の端点数が減少し、最適でない組合せがおこった結果である。一方、制約を持たない場合は最適化率の大きな変化は見られない。この結果から間接制約をもつデータの処理では、 λ の値 (実験による結果では $1 \leq \lambda \leq 5$) を必要以上に多くとってはいけなことがわかった。

5. ま と め

軌跡に制約条件をもつことは、組合せの自由度は減るが、必要な比較時間は増大する。さらに、結合不可条件のみでなく、種々の制約条件のあるデータを扱うことは、最適化問題をより複雑化する。段階的クラスタ化法によって、制約条件のあるデータの扱いが、効率的、総合的に扱えたこと、クラスタ化の手法は、座標データのみでなく、制約条件などの多次元のデータ

を分類・処理できること、またクラスタによって組合せを減らし効率的処理を行うことができること、など従来にはない優れた結果がえられた。今後は、電子部品や機械組立て等における、3 次元の動作を伴う NC ロボット用データ作成等への応用が考えられる。

参 考 文 献

- 1) Avis, D.: Worst Case Bounds for the Euclidean Matching Problem, *Comput. Math. Appl.*, Vol. 7, pp. 251-257 (1981).
- 2) Iri, M., Murota, K. and Matsui, S.: Linear-Time Heuristics for the Minimum Weight Perfect Matching on a Plane with an Application to the Plotter Algorithm, Research Memorandum, RMI 81-07, University of Tokyo (1981).
- 3) Supowit, K. J., Plaisted, D. A. and Reingold, E. M.: Heuristics for Weighted Perfect Matching, *Proc. 12th Annual ACM Symp. on Theory of Computing*, pp. 398-419 (1980).
- 4) Supowit, K. J., Reingold, E. M. and Plaisted, D. A.: The Travelling Salesman Problem and Minimum Matching in the Unit Square, *SIAM J. Comput.*, Vol. 10, pp. 676-681 (1981).
- 5) Avis, D.: *A Survey of Heuristics for the Weighted Matching Problem*, School of Computer Science, McGill University, Technical Report No. SOCS=82.4 (Feb. 1982).
- 6) 雄山真弓: 段階的クラスタ化による効率的描線法, 情報処理学会論文誌, Vol. 28, No. 11,

pp. 1135-1145 (1987).

- 7) 雄山真弓: 段階的クラスタを用いた描線順序の効率化, 情報処理学会論文誌, Vol. 29, No. 12, pp. 1091-1100 (1988).
- 8) 雄山真弓, 阿部健一: 線分間結合に制約のある描線順序の近似最適化, 第40回情報処理学会全国大会論文集, pp. 1548-1549 (1990).

(平成元年11月7日受付)

(平成2年4月17日採録)



雄山 真弓 (正会員)

昭和16年生。昭和38年東北大学理学部化学科卒業。昭和42年関西学院大学理学部実験助手, 昭和51年同大学情報処理研究センター講師, 昭和55年助教授, 昭和64年教授, 現在に至る。作図処理の最適化, データ解析, 教育のためのソフトウェアの開発, 私大間ネットワークなどの研究, 情報処理教育に従事。IEEE学会, 応用統計学会, CAI学会各会員。



阿部 健一 (正会員)

昭和16年生。昭和39年東北大学工学部電気工学科卒業。昭和41年同大学院工学研究科修士課程修了。昭和44年同大学院工学研究科博士課程修了。工学博士。同年同大学工学部助手, 昭和49年同助教授, 昭和59年豊橋技術科学大学教授, 現在に至る。制御理論, 数式処理応用の研究に従事。電子情報通信学会, 計測自動制御学会, 日本オペレーションズ・リサーチ学会等の各会員。