

段階的クラスタ化法を用いた電子部品装着順序効率化のアルゴリズム

雄山 真弓

関西学院大学 情報処理研究センター

プリント基板に電子部品を効率的に装着する問題を段階的クラスタ化法を用いて行った。FA等で多く使われている電子部品装着機では、これに電子部品を実装する順序をあらかじめ決めるためのNCデータの作成が必要である。部品の数や種類の増加に伴い、複雑な組み合わせ問題となる。本論文は、1) 部品を所定の位置に配置するための基板の動き、2) 部品群が配置された棚の動き、3) 部品を取り上げてそれを棚から基板に移すためのロボットの動き、これらの3種の動作を最小にして処理時間を短くする効率的処理を、段階的クラスタ化法を用いたアルゴリズムで行った。効果を実証するため、専門家が行った場合との比較も行っている。

Application of Stepwise Clustering Method for the Determination of Efficient Assembly Sequence

Mayumi Oyama

Information Processing Research Center

Kwansei Gakuin University

Uegahara, Nishinomiya, 662, JAPAN

(E-mail: oyama@kgupyr.kwansei.ac.jp)

A robot is often used to assemble many parts on a board. When there exist large variety of part types, it is an important task to find a sequence by which the parts can be assembled in a short time. This problem can be regarded as an extension of the efficient drawing path problem. In this paper, the stepwise clustering method, which has been very effective for the latter problem, is extended to provide an efficient assembly sequence and it is applied to the assembly sequence of electronic circuit boards. The resulting sequences are shown to be efficient enough to give shorter assembly times than those by human experts.

1. はじめに

精密部品組立工場の多くは、ロボットを用いて電子部品の組立を行っている。一定時間内により多くの部品を基板上に配置することは、効率的な部品の装着順序をみつけることを意味する¹⁾。しかし、部品の数や種類が多くなると、効率的な配置順序をみつけることは、非常に複雑な問題となる。

この問題は、空間に存在する点や線分を全て一度だけ通り、その最短路を求める描線順序問題として考えることができる。つまり、ロボット腕の待ち時間の最小化は、部品棚と基板の動きで決められる空間における最短の描線軌跡を求める問題とみなすことができる。しかし、実際には最短路の距離を問題とするのではなく、最短の配置時間をもつ部品配置順序を決定することである。ここで、配置時間は、描線距離とは同じでないが、時間と距離軌跡の長さが、線形関係でしかも偏差が小さいならば、この問題は効率的な描線順序を求める問題²⁾⁻⁴⁾として扱うことができる。

本論文では、始めにこの問題について、明確な定義を与える。次にこの問題に適用できるように発展させた段階的クラスタ化法⁵⁾⁻⁹⁾のアルゴリズムを説明し、最後に電子部品の組立工程における実際のデータを使用して計算機実験を行なった結果を示す。更に、結果の有効性を評価するため熟練者が定めた組立順序との比較を行なう。

2. 効率的組立順序問題

図1は、典型的な組立順序問題を図式化したものである。ここでの処理の目的は、基板上の決められた位置に複数の部品を配置することである。装置は、部品棚とロボット、そして基板から構成されている¹⁰⁾。図1は、3種類からなる7つの部品が部品棚に置いてあることを示している。ここで、部品棚は左右に動くことができる。部品棚の位置により、ロボットが、どの部品を取るかが決定される。他方、基板は前後左右に動く。した

がって、部品棚の左右の動きと基板の前後左右の動き、さらに部品棚と基板間でのロボットの往復運動が組み合わされて、所定の位置に所定の部品を置き、基板上に部品を組立てていくことが出来る。

この例では、ロボットが部品棚と基板の間を移動する時間は、常に一定(t_r)とする。他方、部品棚が部品の種類を変えるために動く時間は、部品棚上の部品間の距離で d_p に比例する。ロボットの動きと並行して部品棚は同時に動くことができる。そこで、部品棚の移動時間が t_r よりも短い場合は、つぎの部品を取り上げるのに要するロボット腕の待ち時間はゼロとなる。

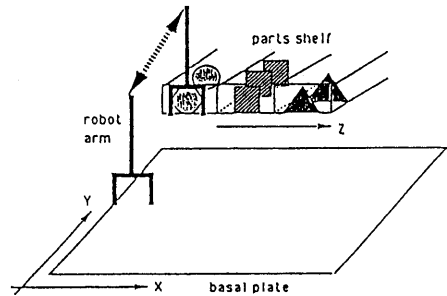


図1 基板組立順序問題のイメージ図

図2は、ロボットの待ち時間 t_p と部品棚の部品配置距離 d_p の関係をグラフに示したものである。このグラフは、もし部品棚の部品間距離 d_p が、限界値 d_p^0 よりも小さいならば、 t_p はゼロになり、 $d_p > d_p^0$ ならば、 t_p は t_p^0/d_p^0 の傾きで増加することを示す。同様な関係が、 x - y 平面上における基板の移動距離 d_b と、部品を基板上に

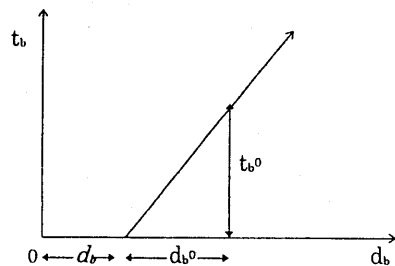
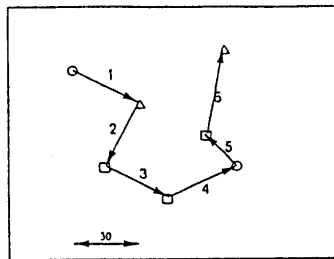


図2 基板移動と無駄時間発生の関係

置く動作を行うのに生じるロボットの待ち時間 t_b の間に存在する。ここで、 d_b は限界値であり、 t_b は $d_b > d_b^0$ の領域で t_b^0/d_b^0 の傾きで増加する。結局、部品配置順序が、部品棚と基板の移動距離を決定し、あわせてロボット腕の待ち時間を決めることとなる。

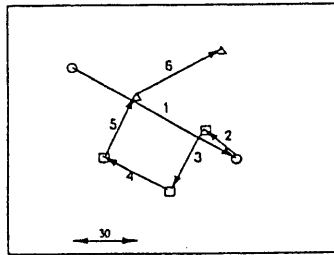
つぎに、基板上での部品の配置順序について説明する。図3は、部品配置順序についての3つの例を示している。図中の数字は、配置の順番を指示している。この例では、ロボットの待ち時間 t_p と t_b を計算するのに次の関係を用いている。



	t_b	t_p
1	0	30
2	0	0
3	0	0
4	0	0
5	0	0
6	10	0
sum	10	30

total waiting time=40

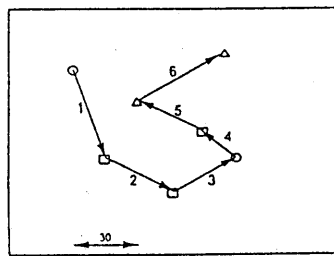
(a)



	t_b	t_p
1	45	0
2	0	0
3	0	0
4	0	0
5	0	0
6	10	0
sum	55	0

total waiting time=55

(b)



	t_b	t_p
1	15	0
2	0	0
3	0	0
4	0	0
5	0	0
6	10	0
sum	25	0

total waiting time=25

(c)

図3 部品の配置順序の関係

部品棚の動きによって生じる待ち時間は、部品の種類が、○から△、またはその逆の場合は、 $t_p = 30$ とし、その他の場合は、 $t_p = 0$ とする。さらに、基板の動きによって生じる待ち時間は、 $d_b < 30$ の場合は、 $t_b = 0$ とし、その他の場合は、 $t_b = d_b - 30$ とする。

図3 [A]と[B]は、それぞれ基板の動きと、部品棚の動きを最小にした場合の配置順序である。最後の配置[C]が、本来の目的と合致するような、ロボットの待ち時間を最小とする配置順序である。各図の右側には、これらの配置順序に伴うロボットの待ち時間を示している。 t_b は基板の動きによって生じたロボットの待ち時間であり、 t_p は部品棚の動きによって生じたロボットの待ち時間である。これらを合わせた時間が、組立処理に伴う全待ち時間を意味することとなる。

図1で提起した問題は、つぎのように一般化することができる。m種類の部品からなるn個の部品を配置することとする。ここで、それぞれの部品の座標値を (x_i, y_i) 、部品の部品棚上での配置位置を p_i で与えるとする。 t_{bi} は、基板上にi番目の部品を配置するのにかかる待ち時間、 t_{pi} は、部品棚が必要な部品位置にくるまでの待ち時間である。それぞれを(1),(2)式に示す。

$$t_{bi} = 0 \quad [\text{if } d_{bi} \leq d_b]$$

$$t_{bi} = (d_{bi} - d_b) \cdot t_b^0 / d_b^0 \quad [\text{if } d_{bi} > d_b] \quad (1)$$

$$t_{pi} = 0 \quad [\text{if } d_{pi} \leq d_p]$$

$$t_{pi} = (d_{pi} - d_p) \cdot t_p^0 / d_p^0 \quad [\text{if } d_{pi} > d_p] \quad (2)$$

ここで、

$$d_{bi} = \max\{|x_i - x_{i-1}|, |y_i - y_{i-1}|\}$$

$$d_{pi} = |p_i - p_{i-1}|$$

問題は、次式(3)で与えられる合計待ち時間 t を最小にする配置順序を求めることである。

$$t = \sum_{i=2}^n (t_{bi} + t_{pi}) \quad (3)$$

3. 効率的な配置順序のアルゴリズム

ここでは効率的な部品配置順序を与えるアルゴリズムを、段階的クラスタ化法の考え方に従って説明する。

図4は部品配置とその配置順序の例を示している。8種類からなる20個の部品を基板上に装着する問題である。図の点線は、段階的クラスタ化法を用いて求めた最も無駄時間の少ない部品の配置順序を示している。この問題で用いたパラメータの値を以下に示す。

$$d_p = 1, t_p^0 / d_p^0 = 1,$$

$$d_b = 200, t_b^0 / d_b^0 = 0.00125$$

部品棚上の部品位置の絶対距離が、0または1である場合は、待ち時間 t_p は0となり、また、基板上の2つの部品間の距離が200より小さい場合は、待ち時間 t_b は0となる。最大の待ち時間の t_p は、基板上の最長部品距離が1000ならば1.0、また、 t_p の場合については、部品の種類が、1から8に変わる場合で6.0の値をとる。

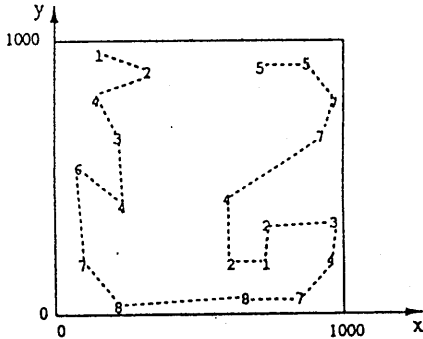


図4 最適化された組立行程の例

段階的クラスタ化法を用いて効率的配置順序を求める手順は以下のように表わされる。

[1] それぞれの部品を線分とみなす。この線分は、 (x_i, y_i) と p_i が作る空間にあり、2端点の座標値は同じとする。2端点は、以下に示すデータで与えられる。

端点 i : $(x_i, y_i, p_i, i+n)$

端点 $i+n$: (x_i, y_i, p_i, i)

ここで、4つの変数の最後の項は、線分を表す2端点間の接続関係を示すもので、他端点の番号を示している。

[2] 部品棚の位置を示す値 p_i を z 軸の座標値に変換する。これによって、各端点は、3次元空間中の点として表現される。座標値 z_i は、つぎの式により計算する。

$$z_i = \mu \cdot p_i \quad (4)$$

ここで、 μ は、経験値である。この例では、 μ に150を与える。 μ 値の決定法はこの節の後半で述べる。

[3] 線分の端点が散在する空間を、一片の大きさ h の立方体セルで分割する。 h の値は、パラメータ λ 、各軸における最大、最小座標値の差 x_w, y_w, z_w 、および、端点数 n を用いて、次の(5)式で決定する。

$$h = ((x_w \cdot y_w \cdot z_w) / 2n \cdot \lambda)^{1/3} \quad (5)$$

もし、 λ の値として1を用いた場合、1セルの中に存在する平均端点数は1である。あとで、 λ の値をいくつに設定すれば、適当なセルの大きさが決定できるかについて説明する。表示を見やすくするために、図4の例では、 $\lambda=12.5$ を用いて、空間を8³個の立方体のセルに分割している。図5の[A]は、セル分割後の様子を x - y 平面上へ投影したものである。さらに、[B]は、 z 軸を含む平面へ投影した場合の概念図を示す。[4] 各セルを順に調べて、端点を含むセルが隣り合っている場合、これらのセル中の端点は同じクラスタに属するものとする。端点が格子にある場合は、あらかじめ、その格子がどちらのセルに属するのか決めておく。この結果、それぞれの端点は、いずれかのクラスタに属することとなる。

[5] クラスタ内での全ての端点間の距離を比較し、最短距離の端点同士を連結する。このクラスタ内結合のプロセスは、連結すべき端点が無くなるまで繰り返す。但し、次の2つの場合、つまり、端点の連結によって閉路ができてしまう場合と、連結する線分のお他端点が共に他のクラスタに存在する場合は、連結しないものとする。

[6] 新しく生成された線分を対象として、ステ

ップ [3] ~ [5] を繰返す。

[7] 線分の数が十分少なくなると、繰返しは終了し、すべての空間が1つのクラスタ領域であるとみなされる。そして、最終的に1線分にまとめられるまで、クラスタ内の連結操作を行なう。

以上によって、すべての部品配置順序が決定されたことになる。図5の [C]、[D] に示す5つの線分に対して、待ち時間が最小となる連結を行ない、図4に示した結果が得られる。

段階的クラスタ化法を効率的に実行するためには、次の2つの要因が重要である。その1つは、

値を1に設定し、各クラスタに含まれる全端点数の最大値を調べて、その値が20より大きい場合と4より小さい場合に分けて新しい λ の値を(6)式に示すように再設定している。

$$\begin{aligned} \text{if } k \geq 20 \text{ then } & \lambda_{\text{new}} \leftarrow f \cdot \lambda_{\text{old}} \\ \text{if } k \leq 4 \text{ then } & \lambda_{\text{new}} \leftarrow (1/f) \cdot \lambda_{\text{old}} \quad (6) \end{aligned}$$

ここで、 k は、クラスタに含まれる最大端点数、 f はダンピングファクターであり、その値として2を用いている。

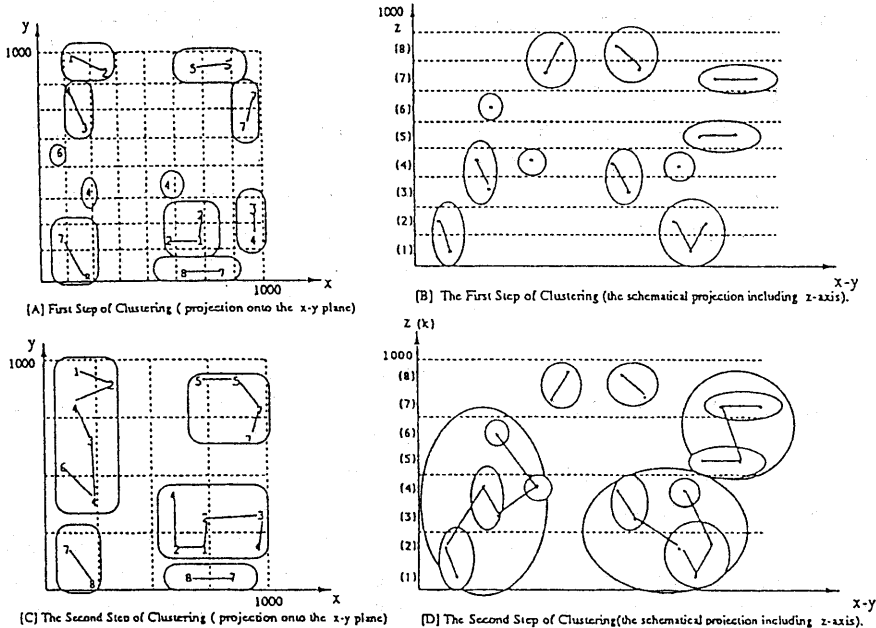


図5 段階的クラスタ化法による描線順序の一般化と連結過程の例

セルサイズをどう選ぶかという問題である。たとえば、セルサイズが大きすぎるとクラスタの数は少なくなり、極端な場合はすべての端点が1クラスタに属することになる。反対にセルサイズが小さすぎると、クラスタの数は多くなりすぎ、ほとんどのクラスタが1端点で構成されることになる。これらの場合には、効率的なクラスタ内線分結合は、期待出来ない。したがって、実際のプログラムでは、適切なセルサイズが得られるように、 λ の値を動的に調整している。すなわち、 λ の初期

次に、処理を行うのに重要なもう1種のパラメータ μ について説明を行う。パラメータ μ は、Z軸の大きさを決定する。この意味づけを行うため、待ち時間 Δt に相当する大きさの仮想的な立方体を考える。この場合、 Δt に相当するx, y, Z軸上での実際のセルの大きさは、次式(7)で与えられる。

$$\begin{aligned} \text{x, y 軸} : \Delta d_b &= d_b + (d_b^0 / t_b^0) \cdot \Delta t \\ \text{Z 軸} : \Delta d_p &= d_p + (d_p^0 / t_p^0) \cdot \Delta t \quad (7) \end{aligned}$$

パラメータ μ は、時間軸での大きさ Δt の立方体セルを与えるとすると、 x 、 y 軸と z 軸間の尺度比率 $\Delta d_b / \Delta d_p$ と見なすことができる。待時間 Δt が 0 の場合と無限大の場合の 2 つの極端な場合を考えるとすると、それぞれの場合のパラメータ μ の大きさは、(8)式で与えられる。

$$\begin{aligned} \Delta t = 0 & : \mu(0) = d_b / d_p \\ \Delta t = \infty & : \mu(\infty) = (d_b^0 \cdot t_p^0) / (d_p^0 \cdot t_b^0) \quad (8) \end{aligned}$$

μ 値の実際の選択は、問題の性質による。もし、待時間が殆ど無視できる状態での効率的部品配置順序をみつける場合には、 $\mu(0)$ を用いるべきであり、一方ロボット腕の動きが非常に速く、 d_b と d_p による効果が小さい場合は、 $\mu(\infty)$ を用いるのがよい。本プログラムでは、 $\mu = \mu(0)$ からスタートして、 μ の値を 2 倍増しにして計算を繰り返す行う。そして合計の待時間が 2 回連続して減少しなくなった時に計算を終了する。

図 5 の場合の例は作図表示を見やすくするため異なったパラメータを用いている。

4. 複雑な組み立て工程問題への対応

つぎに、複数の装置がコントロール可能であるような、より複雑な組立工程問題を取り上げ、段階的クラスタ化法によるアルゴリズムの拡張性について考察する。例えば、位置を速やかにかえることができる複数の部品棚があるとすると、アルゴリズムは、複数の部品棚の位置を示す他の座標軸を新たに加えるという単純な拡張によって問題をとくことができる。複数の基板がある場合についても、同様にアルゴリズムを拡張すればよい。

他方、複数のロボット腕がある場合は、複雑な問題となる。図 6 は、その状況を示している。この図では、(a) のロボット腕 $R1$ と $R2$ は同時に動作し、部品を掴んで基板上に置く操作を交互に行っている。いま、 x - y 平面での 2 つの腕間の距離が d で、これが部品棚の距離にして 3 部品に相当すると考えよう。この時、(b) の問題を図

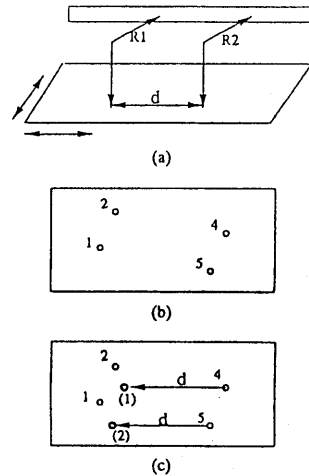


Fig. 7. Assembly problem with two robot arms.

(c) のものに置き換えることができる。ここで、タイプ 4 と 5 を持つ部品は、実際の位置から距離 d だけ離れたタイプ 1 とタイプ 2 をもつコピーされた部品として表現できる。もともとタイプ 1 と 2 を持つ 2 つの部品は、上記のコピーは持たない。なぜならば、 $R1$ は、 $R2$ が部品を配置している間、何もしないからである。このように問題を変換すれば、新しい問題 (c) に段階的クラスタ化法を適用することができる。ここで、多くの部品は、異なった位置とタイプをもって 2 回現れている。なお、アルゴリズムに 1 カ所次のような修正を加える必要がある。それは、部品がクラスタ内結合で連結された時、連結された部品のコピーは、ただちに問題から削除しなければならないことである。この種の問題における拘束条件は、ユークリッド空間における通常の描線軌跡問題として扱うには、制約が強すぎるといえよう。従って他の問題における程の効率性を期待することはできない。しかしながら、ここで述べた段階的クラスタ化法の拡張は、実際の問題に多く適応できるものであり、熟練者や通常の探索方法では到底解決できない問題に対する有効な方法であるといえる。

5. 電子部品装着順序への応用

表2 3種類のロボットの特性

	\bar{d}_b	d_b^0	t_p^0	\bar{d}_p	d_p^0	t_p^0	t_r
Robot1	750	5000	0.26	2	1	0.75	0.5
Robot2	800	5000	0.25	3	1	1.00	0.5
Robot3	300	5000	0.25	2	1	0.75	0.4

* The meanings of the variables are mentioned in Section II.

表1 電子基板のデータ構成

Board	Number of parts	Coordinates				Parts types p_i
		x_{max}	x_{min}	y_{max}	y_{min}	
Board 1	119	15 879	8650	23 035	18 140	1, 2, ..., 55, 56
Board 2	182	9870	10	5440	10	1, 2, ..., 85, 86
Board 3	375	29 105	10	1950	10	1, 2, ..., 77, 78

表3 組立順序において発生する無駄時間の変化

Board	Robot	Type	Σt_b^a	Σt_p^b	Total waiting time t	Optimized ratio (%)
[Board 1]	Robot1	REFERENCE	0.29	1335.75	1336.04	
		OUTPUT	5.52	0.00	5.52	99.59
	Robot2	REFERENCE	0.25	1685.00	1685.25	
		OUTPUT	4.99	0.00	4.99	99.70
	Robot3	REFERENCE	0.91	1335.75	1336.66	
		OUTPUT	7.09	0.00	7.09	99.47
[Board 2]	Robot1	REFERENCE	0.61	3634.50	3635.11	
		OUTPUT	12.47	0.75	13.22	99.64
	Robot2	REFERENCE	0.56	4685.00	4685.56	
		OUTPUT	10.98	0.00	10.98	99.77
	Robot3	REFERENCE	1.16	3634.50	3635.66	
		OUTPUT	15.09	0.75	15.84	99.56
[Board 3]	Robot1	REFERENCE	7.25	4003.50	4010.75	
		OUTPUT	42.35	4.50	46.85	98.83
	Robot2	REFERENCE	6.71	5144.00	5150.71	
		OUTPUT	31.67	10.00	41.67	99.19
	Robot3	REFERENCE	10.76	4003.50	4014.26	
		OUTPUT	40.25	6.75	47.00	98.83

* Σt_b is the sum of waiting times to place every part on the board.

^b Σt_p is that for picking up the parts from the shelf.

とりあげた電子部品実装の工程は、実際に松下電気産業株式会社の工場で行なわれているものである。表1に3つの電子回路のデータを示す。また、回路を組み立てるのに3種類のロボットが使われている。これらのロボットの特性を表2に示す。

Board1,2,3に示した組み立て工程を、それぞれ3種類のロボットが実行するとして、段階的クラスタ化法による計算機実験を行った。その結果、待ち時間が大幅に減少し、効率化が有効に行われることが判明した。ここでは、この効果を明確にするために、X-y平面上の動きのみを最小とする配置順序を入力データとして、(9)式に示す待時間の減少度を表す最適化率(%)を与える。

$$\text{最適化率}(\%) = 100 * (t_{ref} - t_{opt}) / t_{ref} \quad (9)$$

ここで、 t_{ref} と t_{opt} は、それぞれ入力データと最適化結果における待時間の合計である。表3は最適化前と最適化後の結果を示している。図7は、ロボット1とBoard1の組合せで最適化を行った結果のグラフ表示である。X-y平面上の点が各部品の位置を示す。TIMEで示されるZ軸は、それぞれの部品を取り上げてそれを基板上に置く際の待時間の合計を示す。

[A]と比べて[B]の図でZ軸の値が減少していることから、すべての部品で適性に効率化が行われたことがわかる。

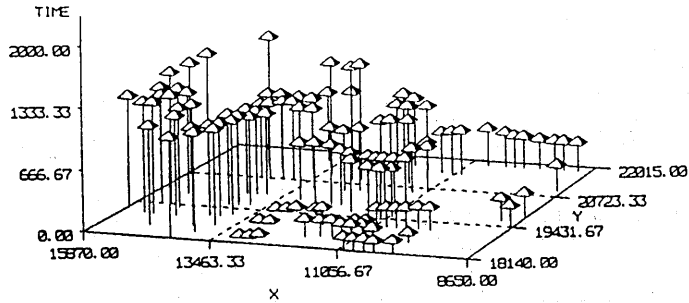
表4は、工場の熟練者と段階的クラスタ化法による部品配置の時間を比較している。結果は明らかに段階的クラスタ化法が、熟練者がおこなった配置順序に実用面で置き代わりうる可能性を示している。実際、最初のケースでは、組立時間は同じであるが、後の2つのケースでは、熟練者よりもはるかに優れた結果を示している。また、最初のケースよりも部品の数が多いことも注目したい。

表4 熟練者と段階的クラスタ化の処理時間の関係

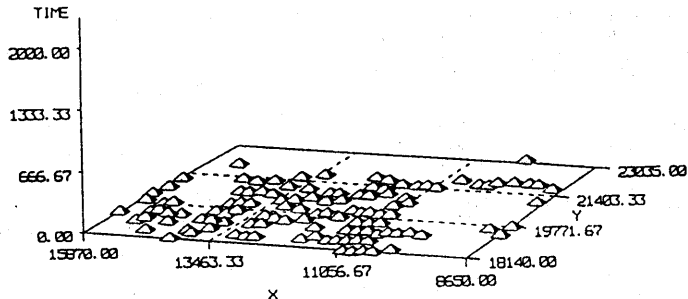
		$n \cdot t_r$	Σt_b	Σt_p	Total assembly time
Board 1	by EXPERT	59.5	5.11	0.00	64.61
	Robot 1 OUTPUT	59.5	5.52	0.00	65.02
Board 2	by EXPERT	91.0	15.03	0.00	106.03
	Robot2 OUTPUT	91.0	10.98	0.00	101.98
Board 3	by EXPERT	150.0	43.86	5.25	199.11
	Robot3 OUTPUT	150.0	40.25	6.75	197.00

6. おわりに

本論文は、段階的クラスタ化法が、FAの分野でロボットによる電子部品装着の順序決定に有効な手段となることを示した。実際、段階的クラスタ化法で得られた結果は、従来その作業を行ってきた熟練者の結果と比べても、同等かそれ以上



(a)



(b)

図7 電子部品組立における無駄時間の変化

上の効率性を与えている。最近は、多品種少量生産の時代であり、新たなNCデータの作成が頻繁に起ることが多く、工場では深刻な問題となっている。本方法は、工場の種々の部品配置の順序決定や工程最適化に、今後応用出来るであろう。

参考文献

- 1) Milner, D.A., "Manufacturing systems: an approach to machining process optimization," "IEEE Transactions on Manufacturing Technology, vol. MFT-3, No.1, pp. 18. (1974).
- 2) 伊理正夫: "計算幾何学と地理情報処理", bit 別冊, pp.114 (1986).
- 3) 茨木俊秀: "組合せ最適化の理論", 電子通信学会(1979).
- 4) 伊理正夫, 野崎昭弘, 野下浩幸編: "計算の効率化とその限界", 数学セミナー増刊, 入門現代の数学13, 日本評論社(1980).
- 5) "段階的クラスター化による効率的描線法", 雄山, 情報処理学会論文誌, Vol.28, No.11,

pp.1135-1145, (1987).

6) "段階的クラスターを用いた描線順序の効率化", 雄山, 情報処理学会論文誌, Vol.29, No.12, pp.1091-1100, (1988).

7) "線分間結合に制約条件のある描線軌跡の最適化", 雄山, 阿部, 情報処理学会論文誌, Vol.31, No.8, pp.1221-1229, (1990).

8) "Application of Stepwise Clustering Method for Efficient Drawing of Biopolymer Structures", Oyama, M., Sasaki, S. and Yoshida, M., Journal of Mathematical Chemistry, Vol.7, pp.217-227, (1991).

9) "Application of Stepwise Clustering for Assembly Sequence", Oyama, M., Abe, K., IEEE Transaction on Systems, Man, and Cybernetics. Vol. 22, No. 2, (1992)

10) 安田昌弘, 森俊洋, 西村一之: "最適実装指令テープ作成システム", National Technical Report, Vol.29, No.4, pp.123-131, (1983).