

部品装着機における ノズル割当を考慮した装着順序問題に対する ヒューリスティックな解法

山本圭輔[†] 太田秀典[†] 中森眞理雄[†]

本論文は、電子基板の生産効率向上を目的として、多機能型部品装着機における部品の装着順序問題の解法を論じている。この問題は装着機のライン全体の動作最適化問題の部分問題であり、配送計画問題 (Vehicle Routing Problem, VRP) の一種と解釈することができる。しかしながら、部品を装着するノズルの位置が経路に影響を及ぼすため、通常の VRP に比べてより複雑な問題となっている。本論文では装着順序問題を整数計画問題として定式化し、機械の特徴を考慮した高速なアルゴリズムを提案する。そして、提案アルゴリズムが、時間をかけて探索を行った解と比較しても遜色のない結果を、実用的な時間で与えることを計算機実験により示す。

A Heuristic Algorithm for Placement Sequence Problem with Nozzle Assignment in a Chip Mounting Machine

KEISUKE YAMAMOTO[†] HIDENORI OHTA[†]
MARIO NAKAMORI[†]

In this paper, we discuss algorithms for the placement sequence problem that significantly affects the performance of chip mounting machines. This problem is a kind of the Vehicle Routing Problem (VRP). However, distance between nozzles of a head in the chip mounting machine makes it more difficult than the typical VRP. We formulate the problem as an integer programming problem and propose several algorithms considering features of the machine. As a result of computational experiments, we obtain satisfactory solutions in practical time.

1. はじめに

基板生産時間の短縮を目的として、部品装着機の運用、動作最適化アルゴリズムの研究が盛んに行われている。部品装着機とは、電子基板生産の部品装着工程で用いられる、基板への部品装着を自動的に行う機械であり、今日ではロータリー型部品装着機 (ロータリー機) と多機能型部品装着機 (多機能機) の2種類の部品装着機が最も広く用いられている。これまで、ロータリー機を対象とした研究は多く発表されてきたが、比較的新しいタイプである多機能機に関しては、動作の自由度が大きいため、まだ研究が十分ではなく、動作最適化アルゴリズムについても改良の余地が残されている。

1.1 多機能型部品装着機

基板生産の現場では、部品装着機 (以下では装着機と略す) を複数台並べたラインを構成して生産を行う場合が多い。すなわち、予め各装着機が担当する部品を決めておき、1台の装着機に割当てられた部品の装着が終了したら、基板はコンベアにより次の装着機へ運ばれ、その装着機に割当てられた部品の装着が行われる。基板に全ての部品を装着するには、通常、ライン上にある全ての装着機を通過する必要がある。

多機能型装着機の簡略図を図1に示す。装着機にはヘッドと呼ばれる部位があり、ヘッドには部品を吸着・装着することができるノズルが複数本ついている。装着機の動作は、最初にヘッドが部品供給部上にある複数の部品を吸着し、それらの部品を吸着したまま基板上の1つの部品の装着地点まで移動してその部品を装着し、別の部品の装着地点までヘッドを移動してはその部品を装着するという動作を繰り返し行い、吸着した全ての部品を装着し終えたら、再び部品を吸着するためにヘッドは部品供給部上へと戻る。ヘッドが部品供給部で部品を装着してから、ヘッドが部品供給部へ戻るまでの一連の動作はターンと呼ばれる。1台の装着機に割当てられた全ての部品の装着が完了するまで、一般には、何回かのターンを行うことになる。

[†] 東京農工大学
Tokyo University of Agriculture and Technology

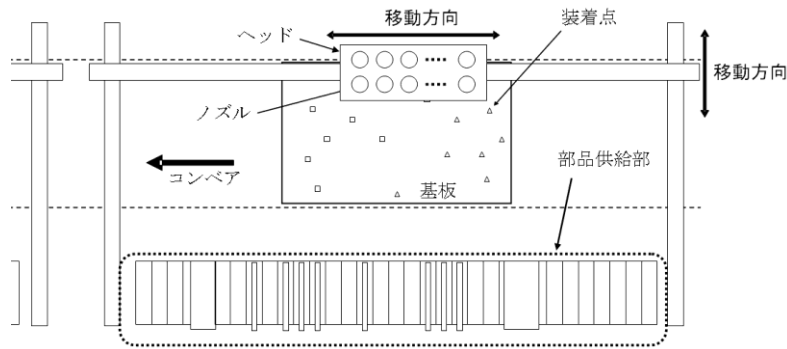


図 1 部品装着機の簡略図

1.2 部品装着機における諸問題

基板生産時間を短縮するためには、まず、ライン内の各装着機にどの部品を割り当てるかを考えなければならない[1]。そして部品を割り当てたあと、各装着機の動作の最適化を行い、各動作時間を算出してライン全体の性能を評価する。ライン全体の性能を評価するためには、部品の割り当て毎に各装着機のさまざまな最適化を行わなければならない。そのため、装着機の動作を高速に解くアルゴリズムが求められている。

1 台の装着機における動作最適化問題には、部品供給部での部品の並びを決定する「部品配置問題」、部品を吸着する順序を決定する「吸着順序問題」、基板上で部品を装着する順序を決定する「装着順序問題」がある。その中でも、「部品配置問題」、「吸着順序問題」に着目して最適化を行っている研究が多く[2][3]、「装着順序問題」に着目した研究、特にノズル位置による距離のずれを考慮した研究はあまり行われていない。しかしながら、基板生産時間を短縮するためには、「装着順序問題」についても考慮し、十分に最適化を行うことが重要である。そこで、本論文では多機能機の特性を考慮した装着順序問題に対するアルゴリズムを提案する。

2. 部品装着機における装着順序問題

装着動作に必要な時間はさまざまな要素により決定されるが、本論文では最も影響の大きいヘッドの移動距離を装着順序問題の評価値とする。

装着順序問題は、一般的に知られている配送計画問題 (Vehicle Routing Problem) の一種と考えることができる。しかしながら、多機能機においては装着点間の距離が最小となるように装着順序を求めても、ヘッドの移動距離が必ずしも最小になるわけ

はない。ノズル間の距離が装着点間の距離に対して無視できない大きさであるために、ヘッドの移動距離は各装着点とその装着点に部品を装着するノズルの組合せによって大きく異なる。例えば図2に示した装着点とノズルの組合せの場合には、移動距離は装着点間の距離に比べ大幅に大きくなってしまふ。これに対して、図3に示した装着点とノズルの組合せの場合には、装着点間の距離よりヘッド移動距離が短くなる。なお、各装着点とその装着点に部品を装着するノズルの組合せを、各装着点のノズル割当と呼ぶことにする。

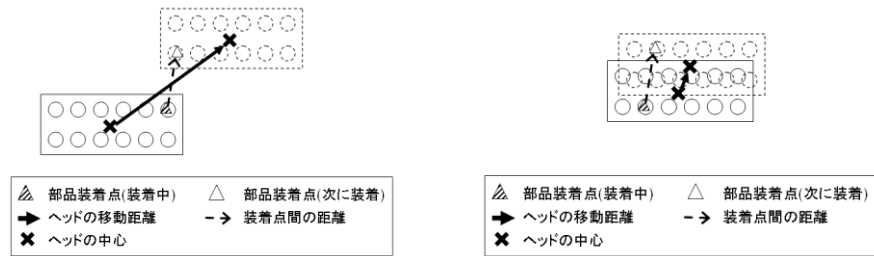


図 2 ヘッド移動距離 (増加)

図 3 ヘッド移動距離 2 (短縮)

本論文で扱う装着順序問題は、装着点数を c 、ヘッド上のノズル本数を n 、総ターン数を t として、以下のように定式化できる。なお、ヘッドは x 方向、 y 方向それぞれに独立して動くモータを持っているため、ヘッド移動距離はチェビシェフ距離によって定義することにする。また、実際の部品装着機ではノズルの種類によって吸着・装着可能な部品種類が限られている場合があるが、本稿では簡単のため、各ノズルがすべての種類の部品を吸着・装着できるものとする。

$$\text{minimize } \sum_{i=0}^c \sum_{j=0}^c \sum_{k=0}^n \sum_{m=1}^t d_{ijkl} x_{ijm} y_{ikm} y_{jlm} \quad (1)$$

subject to

$$\sum_{j=0}^c x_{ijm} = 1 \quad \text{for } i \in \{1, \dots, c\} \quad (2)$$

$$\sum_{i=0}^c x_{ijm} = 1 \quad \text{for } j \in \{1, \dots, c\} \quad (3)$$

$$\sum_{j=1}^c x_{0jm} = \sum_{i=1}^c x_{i0m} = t \quad (4)$$

$$\sum_{i \in C} \sum_{j \in V_m \setminus C} x_{ijm} \geq 1 \quad \text{for } C \subset V_m, m \in \{1, \dots, t\} \quad (5)$$

$$\sum_{i=1}^c y_{ikm} \leq 1 \quad \text{for } k \in \{1, \dots, n\}, m \in \{1, \dots, t\} \quad (6)$$

$$\sum_{k=0}^n y_{ikm} = 1 \quad \text{for } i \in \{1, \dots, c\} \quad (7)$$

$$y_{00m} = 1 \quad \text{for } m \in \{1, \dots, t\} \quad (8)$$

$$y_{i0m} = 0 \quad \text{for } i \in \{1, \dots, c\}, m \in \{1, \dots, t\} \quad (9)$$

- d_{ijkl} : 装着点*i*をノズル*k*で装着し、次に装着点*j*をノズル*l*で装着する場合のヘッド移動距離
 d_{0j0l} : 装着出発点から移動し、装着点*j*をノズル*l*で装着する場合のヘッド移動距離
 d_{i0k0} : 装着点*i*をノズル*k*で装着し、装着出発点に戻る場合のヘッド移動距離
 x_{ijm} : ターン*m*において装着点*i*から装着点*j*へのパスが存在するとき1, そうでなければ0
 y_{ikm} : ターン*m*において装着点*i*をノズル*k*で装着するとき1, そうでなければ0
 V_m : ターン*m*に含まれる装着点の集合

式(1)は装着順序問題の目的関数であり、ヘッド移動距離の最小化が目的となっている。式(2), (3)はすべての装着点は1つのターンで1度だけ装着されることを示し、式(4)はすべてのターンは装着出発点から始まり装着出発点へ戻ることを示している。また、式(5)は各ターンが一つの巡回路によって形成されていることを示す。式(6)は各ノズルが一度に1個までしか部品を装着することができないことを示し、式(8)は各装着点がいずれかのノズルによって装着されることを示している。最後に、式(8), (9)は式(1)において、装着出発点からのパスと装着出発点へのパスを成立させるためのものである。

3. アルゴリズム

装着順序問題は部品割当の度に発生するので高速に解く必要がある。そこで、本論文では装着順序問題を解くアルゴリズムとして局所探索法 (Local Search, LS) を用いる。

局所探索法の初期解生成法として2種類の手法を用意した。また、ターン間での装着点の割当を改善する近傍として、2-swap 近傍と 2-swap-insert 近傍の2種類を用意した。この2つの近傍をそれぞれ個別に用いた局所探索法によって、初期解の改善を行った。

3.1 初期解生成

初期解の生成として、配送計画問題等に広く用いられる最近近傍法 (Nearest Neighbor, NN) を改良し、ノズル割当についても考慮した手法 (以下、ex-NN と呼ぶ) と、マッチングにより各ターンの装着点とノズル割当を求める手法 (以下、Greedy Matching と呼ぶ) を提案する。

3.1.1 ex-NN

以下の手順で初期解を生成する。なお、各経路において1番目に装着を行う装着点を初期装着点と呼ぶことにする。

各ターンについて以下の(1)から(3)の手順を行う。

- (1) ノズル $k \in \{1, \dots, n\}$ について、以下のステップ a, b を行い、 n 通りの経路を生成する。
 - a 装着出発点から最も近い装着点を初期装着点とし、 k をその装着点に割当てるノズルとする。
 - b 割当可能なノズル、または現在の経路に含まれていない未決定な装着点が無くなるまで以下の操作を繰り返す。
 - 未割当のノズルと現在の経路に含まれていない未決定な装着点の組合せのうち、直前の装着からヘッド移動距離が最も短い組合せを次の装着点とそのノズル割当とする。
- (2) 生成された n 通りの経路のうち、ヘッド移動距離の最も短いものを採用し、現在のターンにおける装着順序とノズル割当を決定する。
- (3) 得られたターンに対して、2-opt 近傍による局所探索法を適用し、ターンの装着順序を改善する。

この手法において、初期装着点に割当てるノズルを n 通り全て試すのは、部品供給部

上の装着出発点と基板との距離が基板の横幅よりも大きく、チェビシエフ距離で考えた場合、初期装着点までのヘッド移動距離の最も短くなるノズル割当てが1通りに定まらないためである。なお、(3)で用いられる 2-opt 近傍は経路問題に対して非常に効果的で、規模の小さい問題であれば局所探索のみで最適解に近いところまで到達可能であることが、経験上よく知られている。

3.1.2 Greedy Matching

1つのターンにおいて、図4のような基板上的装着点の位置とヘッド上のノズル配置がまったく同じである場合、ヘッドの移動距離は小さくなる。図4の例ではヘッド移動距離は、装着出発点と基板の往復距離のみである。これより、図5のように適当な箇所をヘッドの中心としたとき、割当てられたノズルと装着点の間の距離が短いほどヘッドの移動距離が短くなる傾向にあると筆者らは予想した。この予想を基に、基板上に仮配置したノズルと装着点の間でマッチングを行い、初期解を生成する手法を提案する。

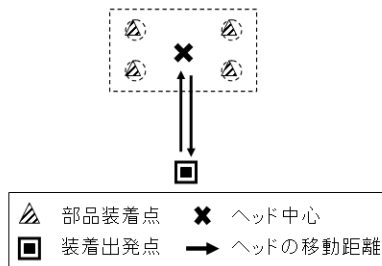


図4 理想的な装着点とノズル割当て

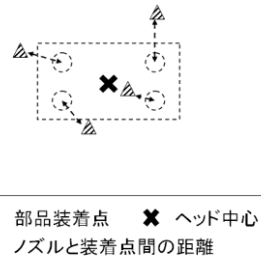


図5 ノズルと装着点の距離

Greedy Matching の手順は大きく2つに分けられる。

はじめに装着点に対してグループ分けを行い、各グループを $g_m (m=1, \dots, t)$ とする。各グループ g_m を求める手順は次のとおりである。

基板の四隅から最も距離の短い装着点をそれぞれ探し、この中で最も距離の短い装着点を c_m とする。基板の右上からの距離が最も短い場合には、ヘッド上の右上のノズルを選択する。同様に、基板の右下からの距離が最も短い場合は右下のノズル、左上からの場合には左上のノズル、左下からの場合には左下のノズルをそれぞれ選択する。選択したノズルで c_m に装着するときのヘッド位置を考慮し、このときの各ノズル位置から最も近い装着点をそれぞれ貪欲的に求め、これらの装着点をグループ g_m に割当てる。

残った装着点に対して同様の操作を繰り返し、すべての装着点をグループ分けする。次に、各グループに含まれる装着点の座標から重心を求め、そこをヘッド中心の座標としたときの各ノズルの座標をそれぞれ求める。求めた各ノズルの座標点をノズル点 $N_{mk} (m=1, \dots, t, k=1, \dots, n)$ とする (図6参照)。

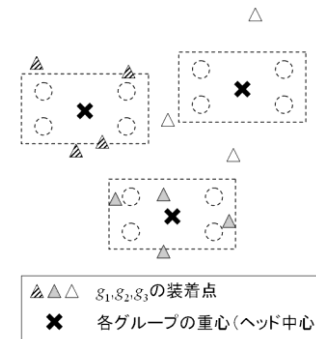


図6 グループとヘッド配置の例

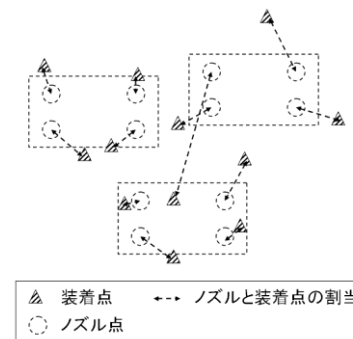


図7 マッチングの例

さらに、全ノズル点と全装着点の間で、次に述べる方法で最大マッチングを求め、得られた結果から、ノズル点 N_{mk} とマッチングした装着点をターン m にノズル k で装着させることにする。図7にマッチングの例を示す。なお、グループ分け通りに各装着点が振り分けられるわけではないことに注意されたい。

マッチングの評価値として各ノズル点と各装着点の距離 (ここではユークリッド距離) を用い、距離の短い組合せから貪欲的にマッチングを求めた。なお、**最小重み最大マッチング** を求めるアルゴリズムを利用することにより、すべてのマッチングの総距離を最小とするマッチングを多項式時間で求めることができるが、計算量が大いこと、事前実験を行った結果最終的な解のヘッド移動距離に大差がないことから、上記のように貪欲的にマッチングを求める手法を用いた。

この段階では各ターンに割当てられる装着点とそのノズル割当てしか決定されていないので、最後にターンごとに最近近傍法を適用して装着順序を求め、その後各ターンに対して、2-opt 近傍による局所探索法を適用し、各ターンの装着順序を改善する。

3.2 近傍解生成

近傍解の生成法として、2点の装着順序を交換する 2-swap 近傍が考えられる。また、各ターンへの装着点の割当てを改善するために筆者らのオリジナルな手法として 2-swap-insert 近傍を提案する。以下でそれぞれの近傍解生成法について記述する。

なお、配送計画問題等によく用いられる近傍解生成法として挿入近傍があるが、装着順序問題においては多くのターンにノズル本数分の装着点が割当てられており、挿入できる箇所が少ない。そのため、探索範囲が狭く効果的な手法とはならないので採用しなかった。

3.2.1 2-swap

2-swap 近傍は 2 つの装着点の装着順序を交換する近傍である。2 点在同一ターンに含まれている場合はノズル割当を変えずに装着順序のみを交換する。2 点が別のターンに含まれている場合には、ノズル割当は交換先の装着点に割当てられていたノズルを割当てる。

3.2.2 2-swap-insert

2-swap-insert 近傍は、交換した装着点をそれぞれの装着順序の最良（以下に述べる意味で）の位置に挿入する近傍である。

ターン T_1 で装着する装着点を a_1 、ターン T_2 で装着する装着点を a_2 とする。まず、 a_1 と a_2 の割当を交換し、 T_1 で a_2 を、 T_2 で a_1 を装着するものとする。なお、この際ノズル割当は交換先のターンでのノズル割当とする。次に、交換された a_1 、 a_2 を各ターンの経路に挿入したとき、もっともヘッド移動距離の短くなる箇所にそれぞれを挿入する。

初期解は 2-opt 近傍を用いた局所探索を先に行っているため、各ターンの装着順序がほぼ最良の状態になっている。また、装着順序が最良の状態である解について 2-swap-insert を用いて得られた近傍解も、交換のあとの挿入によって各ターンの装着順序をほぼ最良に保つことができる。つまり、2-swap-insert 近傍を用いることによって、各ターンへの装着点の割当とターンごとの装着順序を同時に最適化することが可能となる。

4. 計算機実験

提案した手法の性能を評価するために、計算機により比較実験を行った。入力データとして、それぞれランダムに装着点の座標を設定した 15 種類のデータを用いた。ヘッド上のノズル本数は 12 本で、図 8 に示すような配置のものを用いた。ノズル間の距離（図 8 の a , b ）は実際の部品装着機を参考に値を設定した。また、基板のサイズは横 240mm、縦 200mm とし、装着出発点の位置は x 座標を基板の x 軸方向の中央、 y 座標を基板から 140mm の位置とした（図 9 参照）。実験環境は CPU が Intel Core2 Duo 3.00GHz、メモリが 1.96GB、言語は C++を用いた。

局所探索には即時移動戦略を用い、各データに対してそれぞれのアルゴリズムを 5 回実行してその平均を求めている。

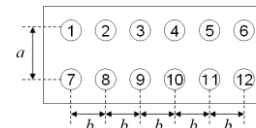


図 8 ヘッド上のノズル配置

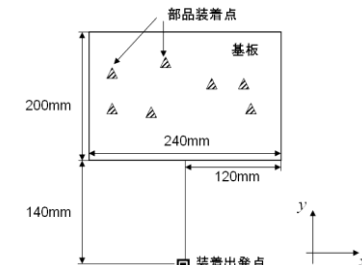


図 9 基板サイズと装着出発点

表 1 各アルゴリズムの結果

初期解 近傍	装着点数	ex-NN 2-swap		ex-NN 2-swap-insert		Greedy Matching 2-swap		Greedy Matching 2-swap-insert	
		time	cost	time	cost	time	cost	time	cost
1	250	0.2	10876.9	0.8	10740.9	0.1	10837.4	0.8	10707.3
2	250	0.2	11168.0	0.7	10958.2	0.2	11096.2	0.7	10922.3
3	250	0.2	11355.3	0.6	11170.7	0.1	11312.7	0.7	11010.1
4	250	0.2	11391.2	0.7	11175.5	0.1	11232.6	0.6	10998.2
5	250	0.2	11444.4	0.8	11231.3	0.1	11245.9	0.8	11003.6
6	500	0.9	22052.8	4.7	21697.5	0.7	21885.6	3.4	21618.5
7	500	0.7	21917.3	3.9	21598.7	0.7	21665.4	3.3	21403.5
8	500	0.8	22402.0	3.7	22047.5	0.7	22207.6	3.8	21887.3
9	500	0.8	21892.8	4.1	21469.6	0.8	21500.9	3.5	21209.7
10	500	0.8	21945.3	3.5	21652.4	0.6	21691.2	3.2	21397.4
11	1000	5.3	43223.3	25.5	42444.6	4.5	42867.1	19.4	42324.5
12	1000	4.2	42683.2	24.3	41967.3	4.3	42281.2	18.6	41831.5
13	1000	4.5	42554.0	21.3	41886.1	3.7	41919.6	17.9	41507.0
14	1000	5.2	41974.7	19.1	41323.6	3.9	41249.1	17.4	40962.7
15	1000	3.9	42277.6	20.3	41667.2	4.1	41904.4	18.3	41502.9

表 1 は局所探索に各アルゴリズムを用いたときの結果である。time は計算時間 (s)、cost はヘッドの総移動距離 (mm) を示す。

初期解ごとに総移動距離を比較すると、2 つの近傍のどちらを用いた場合でも、Greedy Matching を初期解生成に利用した方が、ex-NN を利用したものより平均 1% 程度良い結果となった。次に、2-swap 近傍と 2-swap-insert 近傍の性能について比較すると、2-swap-insert 近傍を用いたアルゴリズムの方が、2-swap 近傍を用いたものよりも平均 1~2% 程度良い結果となった。この結果から、2-swap-insert 近傍は装着順序問題

において非常に効果的な近傍であるといえる。これらの結果より、初期解生成に Greedy Matching, 近傍解生成に 2-swap-insert を用いたアルゴリズムが最も優秀であった。

計算時間について比較してみると、2-swap-insert 近傍を用いた場合は 2-swap 近傍を用いた場合より 5 倍程度の計算時間がかかっている。しかし、どちらも実用上問題ない計算時間であり、十分高速であるといえる。問題のサイズによって近傍を使い分けるという方法も考えられる。

表 2 局所探索法と焼きなまし法の比較

探索法	局所探索法	焼きなまし法	焼きなまし法	焼きなまし法	焼きなまし法	
	Greedy Matching	ex-NN	ex-NN	Greedy Matching	Greedy Matching	
初期解						
近傍	2-swap-insert	2-swap	2-swap-insert	2-swap	2-swap-insert	
装着点数	cost	cost	cost	cost	cost	
1	250	10707.3	10686.4	10498.4	10681.7	10486.1
2	250	10922.3	10851.3	10594.6	10858.6	10576.9
3	250	11010.1	10975.3	10742.8	10956.8	10681.5
4	250	10998.2	11028.5	10747.6	10987.2	10731.5
5	250	11003.6	11011.2	10790.9	11000.1	10749.4
6	500	21618.5	21581	21215.6	21592.3	21246.3
7	500	21403.5	21481.2	20932.4	21458.4	20987.5
8	500	21887.3	21950	21445.5	21877.6	21447.5
9	500	21209.7	21244.2	20754.5	21225	20801.3
10	500	21397.4	21370.7	20928	21401.8	20957.5
11	1000	42324.5	42071.6	41514.2	42161.2	41595.6
12	1000	41831.5	41578.1	40991.4	41721.3	41085.9
13	1000	41507.0	41409.8	40848.2	41466.5	40888.2
14	1000	40962.7	40944.1	40371.6	40970.8	40408.2
15	1000	41502.9	41327.8	40739.1	41415.1	40830.7

次に、局所探索法により得られた解の質を評価するため、提案した二つの近傍をそれぞれ用いた焼きなまし法 (Simulated Annealing, SA) により時間をかけて十分に改善を行ったときの解との比較を行った。表 2 は局所探索で最良となった初期解生成に Greedy Matching, 近傍解生成に 2-swap-insert を用いたアルゴリズムと、各種提案手法を焼きなまし法に適用したアルゴリズムの性能を比較したものである。局所探索による解は、2-swap-insert 近傍を用いた焼きなまし法による解と比較して平均 2% 程度の差となっており、十分質の良い解が得られていることがわかる。また、2-swap 近傍を用いた焼きなまし法による解と比較すると平均 0.15% の差しかなく、2-swap-insert 近傍が非常に効果的な近傍であるといえる。

5. まとめ

本研究では、部品装着機における装着順序問題の定式化を行い、問題の特徴を考慮したアルゴリズムを提案した。計算機実験により、提案アルゴリズムを用いて実用的な時間で、十分に時間をかけて探索を行った場合と比べても遜色ない結果を得ることができることを確認した。

今後の課題として、吸着動作も考慮した部品装着機全体に関する問題に提案アルゴリズムを適用させ、検証を行うことがあげられる。また、実用化のためには、ノズル種類や部品種類を考慮することや、ヘッド移動距離ではなくさまざまな要素を含めた実時間を評価値とすることなどが必要である。

参考文献

- 1) 山田剛史, 中森真理雄: 基板生産における生産ラインのラインバランシング問題に対するヒューリスティックな解法, 情報処理学会論文誌, 数理モデル化と応用, Vol. 1, No.1, pp.102-121 (2008)
- 2) Sun, D.S., Lee, T.E., Kim, K.H.: Component Allocation and Feeder Arrangement for a Dual-Gantry Multi-Head Surface Mounting Placement Tool. International Journal of Production Economics, Vol.95, pp.245-264 (2005).
- 3) Yamada, T., Miyashiro, R., and Nakamori, M.: An Algorithm of Feeder Arrangement and Pick up Sequencing of Component Placement Machine on Printed Circuit Board, Proc. International Conference on Parallel and Distributed Processing Techniques and Applications, pp.403-409 (2005).