

PCB 삽입 경로 및 부품함 위치 배정 연구

이상복* · 강석호**

A Study of Inserting Sequence and Feeder Allocation Problem on Printed Circuit Board

Sangbok REE* and Suk-HO KANG**

Abstract

In this paper, we try to raise the efficiency of PCB production by improving automated inserting process. This process is composed of part inserting sequence and feeder allocation problem. Until now, this problem is formulated by Traveling Salesman Problem(TSP) or Rural Postman Problem. They concentrated more on parts(chips) inserting. But in this paper, we concentrate on Feeder. We formulate 0-1 Integer program(MP2) easier than TSP. This program can be solved by using LP Package(i.e. LINDO, GAMS etc). We propose Heuristic algorithm ISFA which guarantees a local optimum and often gives optimal solution. We give some examples to prove ISFA algorithm.

1. 서 론

현대 산업에서 모든 자동화 기계 및 전자 제품에 필수품인 인쇄집적기판(Printed Circuit Board : PCB)의 생산은 기하급수적으로 늘어가고 있다. PCB를 생산할 때, 고려해야 하는

여러 가지 문제 중 구체적인 기술적 문제 외에, PCB상에 각 부품 위치를 결정하는 칩 위치 배정 문제(Chip Layout Problem)와 PCB에 각 부품(CHIP)을 해당 부품함에서 원하는 위치에 가져다 삽입하는 삽입 경로 문제 그리고 부품함 위치 배정 문제는 수학적 최적화 문제임을 알 수 있다. 칩 위치 배정 문제는 PCB

* 서경대학교 산업공학과

** 서울대학교 산업공학과

설계자가 각 부품간의 기술적 사항 및 생산 문제를 모두 고려해서 결정한다. 이 논문에서는 삽입 경로 문제와 부품합 위치 배정 문제를 동시에 고려해서 연구한다.

부품을 삽입하는 제공된 로봇의 기술적인 정도에 따라 문제 해법은 조금씩 달라진다. 로봇은 입력된 삽입 경로에 따라 작업한다. PCB에 부품들을 삽입하는 로봇으로 많이 사용되고 있는 로봇은 4가지 형식이 있다. 1) 로봇이 한번에 여러 부품을 동시에 집어서 기억된 순서대로 차례로 삽입하는 형식(타입 1) : 이런 로봇은 가격이 비싸고 속도는 빠르나 불량율이 높다. 현장에서 많이 사용되고 있지는 않다. 2) 로봇이 한번에 한 부품만 집어서 원하는 위치에 수직으로 먼저 움직이고 다음에 수평으로 움직이는 형식(타입 2) : 이런 로봇은 속도는 느리나 정확도가 높다. 3) 로봇이 한번에 한 부품만 집어서 수직과 수평으로 동시에 움직여서 원하는 위치로 이동하는 형식(타입 3) : 이런 로봇은 속도와 정확도는 중간이다. 4) 로봇이 한번에 한 부품만 집어서 원하는 위치로 직접 이동하는 형식(타입 4) : 이런 로봇은 속도는 빠르고 정확도는 중간이다.

PCB를 생산할 때, 전체적으로 고려되는 목적 함수는 생산성과 신뢰성이다. 이 논문에서는 PCB 생산시에 자동 삽입 공정은 삽입 경로 문제로 삽입 경로 문제의 해법을 개발함으로써 삽입 공정 시간을 단축하고자 한다. 삽입 경로 해법을 개선하고자 할 때는 부품합의 위치 배정 문제가 삽입 경로 문제에 중요한 결정 문제이므로 부품합 위치 배정 문제를 함께 개선하는 해법을 연구했다. 일반적인 삽입 경로 문제는 NP-Hard 문제로 증명됐다[1,2,5,8]. 이 논문에선 쉽게 이 문제를 풀 수 있는 해법

을 제시하여 실제 생산 현장에 유용하게 적용될 수 있도록 하였다.

II. 부품합의 위치가 주어졌을 때 삽입 경로 문제

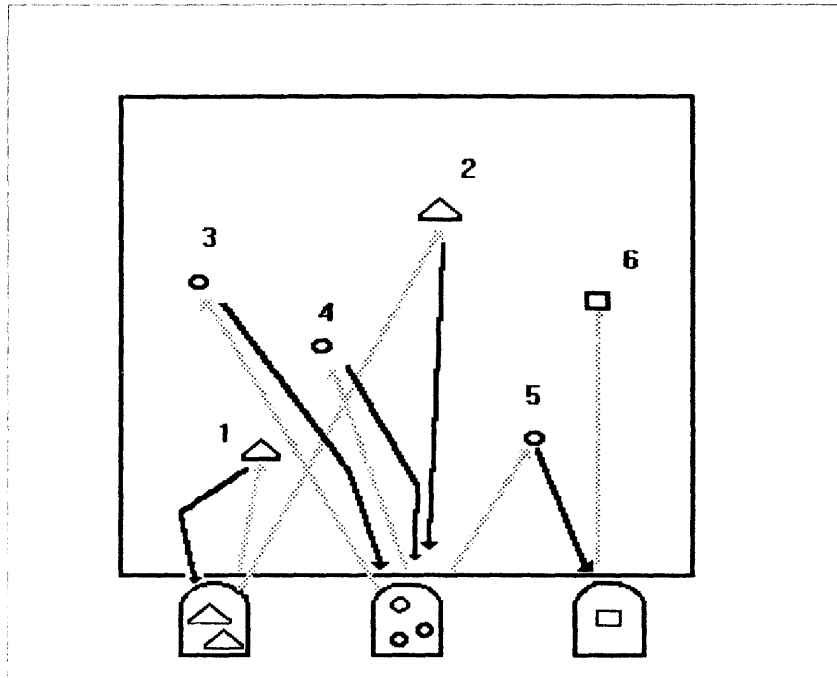
PCB상에 삽입 되어져야 할 부품의 위치와 부품합의 위치가 알려져 있을 때 로봇이 자동으로 부품을 삽입하는 문제를 생각해 보자. 부품이 삽입되는 PCB는 일정하게 똑같은 직사각형으로 되어 있다. 특수한 경우에는 모양이 일반 다각형 형태를 이룬다. PCB 둘레에는 부품이 들어있는 부품합이 놓여 있다. 제공된 로봇이 부품합에서 부품을 집어서 PCB상 해당 삽입 위치에 삽입하는 삽입 순서를 정하는 삽입 경로 문제를 풀어본다. 각 PCB가 자동적으로 로봇 아래로 옮겨지면, 이미 입력된 삽입 순서에 따라 로봇이 부품을 삽입한다. 로봇 팔은 처음엔 원점에서 부품합까지 가서 부품을 삽입하기 시작하고 마지막 부품이 삽입되고 나면 원점으로 다시 돌아온다. 마지막 부품이 삽입되면 삽입이 완료된 PCB는 다음 공정으로 옮겨지고, 새 PCB가 이동되어 위치한다. 이 작업은 2차원 평면에서 이루어진다. 부품들은 로봇 팔에 의해서 PCB 위에서 하나씩 차례로 삽입된다. 로봇 팔이 부품을 잡는 시간과 부품을 삽입하는 시간은 일정하므로 이동 시간에 포함시켜 생각할 수 있다.

각 부품 종류별로 부품합이 각각 있어 부품합은 한 종류의 부품만을 담고 있으며 용량은 충분하다. 부품합이 놓이는 위치는 일반적으로 PCB의 둘레로 PCB의 좌측, 우측, 위쪽, 아래쪽, 또는 이들의 조합에 위치한다.

이절의 목적은, 하나의 PCB상에 모든 부품을 삽입 하는데 로봇 팔이 움직인 총 비용을 최소화 하기 위한 부품 삽입 경로를 결정하는 것이다. 로봇 팔이 움직인 총 비용은 로봇 팔이 움직인 총 거리로 생각할 수 있으므로, 하나의 PCB상에 모든 부품을 삽입 하는데 로봇 팔이 움직인 총 거리를 최소화 하기 위한 부품 삽입 경로를 결정하는 것이다.

PCB 상에 부품을 삽입하는 문제를 정확히 이해하기 위하여 이미 알려진 교외 우체부 해법을 살펴 보자. [그림 1]을 보면, 유방향 네트워크 $G=(N, E)$ 에서, 각 호 $(i, j) \in E$ 에 관련된 비용을 c_{ij} 이라 하면, 각 호 E 는 서로 독립인 A (=비수요 호)와 R (=수요호)로 구성되어 있다. 여기서 A 는 PCB상의 부품에서 부품함

까지의 거리로 임의로 선택할 수 있으나, R 은 수요호로 부품함에서 PCB상의 부품까지의 거리로 필수적이다. 교외 우체부 문제는 네트워크 $G=(N, E)$, $E=R \cup A$ 안에서 R 의 각 요소를 꼭 한번씩 통과하면서 총비용 $(\sum_{k=1}^r c_{i_k, j_k})$ 을 최소로 하는 폐경로 $(i_1, j_1), (i_2, j_2), \dots, (i_r, j_r)$ 를 찾는 문제이다. 비수요 호(A)는 한번만 방문할 필요는 없다. PCB 부품 삽입 경로 문제는 교외 우체부 문제로 생각할 수 있다[2]. 교외 우체부 문제는 NP-Hard로 증명되었다[8]. 교외 우체부 문제로 제시된 해법은 각 단계별 복잡한 과정을 거친다. 각 단계별로 효과적인 알고리즘이 사용되더라도 전체 문제를 풀기에는 비효율적이다[2,5,7].



[그림 1] PCB상 각 부품들과 부품함의 수요호와 비수요호 및 삽입순서

로봇 팔의 움직임에 따라 여러가지 거리가 적용된다. 본 해법에서는 여러 형태의 거리에 대하여 전부 적용 가능하도록 정의하였다. $d(p_{ik}, C_i)$: 비수호의 거리 PCB상의 i 타입 k 번째 부품에서 i 타입 부품함까지 거리 $i=1, \dots, N$; $k=1, 2, \dots, D_i$. 로봇 팔이 움직이는 속도는 일정하다. 가속이나 감속은 없다. 로봇 팔이 움직이는 방식에 따라 거리함수를 아래와 같이 표현할 수 있다.

- 1) 직선 거리: 로봇 팔이 부품 $P_{ik}=(a_{ik}, b_{ik})$ 에서 부품함 $C_i=(x_i, y_i)$ 로 직접이동할 수 있을 때:

$$d(P_{ik}, C_i) = \sqrt{(a_{ik} - x_i)^2 + (b_{ik} - y_i)^2}$$

- 2) 체비세브 거리: 로봇 팔이 부품 $P_{ik}=(a_{ik}, b_{ik})$ 에서 부품함 $C_i=(x_i, y_i)$ 로 X 축과 Y 축으로 동시에 같은 속도로 움직일 때:
 $d(P_{ik}, C_i) = \text{Max}[|a_{ik} - x_i|, |b_{ik} - y_i|]$

- 3) 직교 거리: 로봇 팔이 부품 $P_{ik}=(a_{ik}, b_{ik})$ 에서 부품함 $C_i=(x_i, y_i)$ 로 X 축으로 이동한 다음 Y 축으로 이동할 때:
 $d(C_i, P_{ik}) = |a_{ik} - x_i| + |b_{ik} - y_i|$

그 외 다른 거리형태는 위와 같은 요령으로 정의하여 적용할 수 있다.

기존 해법이 부품에서 다음 부품으로 이동하는 경로에 관심을 두는데 비해 부품함에서 부품함으로의 경로에 관심을 두어 해를 찾을 때, 다음 제안을 만족해야 한다.

제안 1: (1) 각 PCB상의 부품들의 비수호 호는 하나의 부품함과만 연결된다. (2) 각 부품함에는 부품함이 갖고 있는 부품 수 만큼 PCB상의 부품들의 비수호호와 연결되어야 한

다. 부품함 입장에선 균형적이어야 한다. (3) 부품함이 같은 타입의 PCB상의 부품과 모두 연결되어선 안된다.

증명: (1) 비수호호가 두개의 부품함과 연결되었을 때는 로봇 팔이 부품을 갖지않고 다시 한번 부품까지 방문해야 한다. 이는 목적에 어긋난다. (2) 부품함이 갖고 있는 부품 수 만큼만 로봇 팔이 부품함을 방문 하는 것이 로봇 팔 운동 거리의 최소화에 적합하다. (3) 부품함이 같은 타입의 PCB 상의 부품과 연결되면 로봇 팔 운동에 부분환이 생긴다. 해당 부품을 전부 삽입한 후에 로봇 팔은 빈 손으로 다음 타입으로 간다. 이는 목적에 어긋난다. 각 타입의 부품함 엔 PCB상의 같은 타입의 부품이 적어도 하나씩 입력되어야 한다. ■

III. 부품함 위치 배정 문제

부품함 위치 배정 문제는 부품함의 위치를 결정 함으로써 PCB상의 모든 부품을 삽입하는 로봇 팔의 총 이동 경로의 거리 합이 최소가 되도록 부품함의 위치를 정하는 문제이다. 부품함 위치는 PCB 둘레의 모든 지역에서 가능하지 않다. 로봇 팔이 움직이는 영역을 고려할 때는 PCB 형태를 일반 다각형으로 해석할 수 있다. 이 절에서는 삽입 순서를 고려하지 않고 부품함 위치 배정 문제만을 다룬다. 모든 부품들에서 부품함 까지 거리의 합이 최소가 되는 부품함의 위치를 찾는다. 그러면 이 문제는 일반 위치 배정 문제가 된다. 삽입 되어야 할 총 부품들의 각 부품 별로 중앙값(Median)을 찾는 문제다. 여러 점에서 모든 거리의 합이 최소가 되는 점을 찾는 문제는 중앙값

찾는 문제가 된다. 위치 배정문제는 많은 사람이 다루어 왔다. 제공된 로봇의 운동에 따라 등가원이나 등가선을 이용하여 최적 위치를 언제나 제시할 수 있다[3,6]. 제시된 중앙점은 구체적으로 한점으로 주어진다. 제시된 위치는 정수가 아닌 실수이다. 그러나 언제나 PCB 둘레는 부품합이 놓이는 실제 상황을 고려해서 구획으로 나눌수 있고, 정수화할 수 있다[12]. PCB 둘레를 구획으로 나누어 생각할 때는 중앙점 찾는 방법과는 다르게 선형식으로 수식화할 수 있다[11]. 이 경우의 목적 함수도 PCB 상의 모든 부품에서 부품합까지의 거리의 합을 최소화 하는 것이다. 이 문제는 각 타입별 부품을 하나의 군으로 묶어, 가능한 위치중에서 각 타입 별로 최적 부품합 위치점을 찾는 문제이다. 이 문제는 배정 문제(Assignment Problem)로 모델링할 수 있다. 주어진 PCB은 사각형 $[0, a] \times [0, b] = R^2$ 이라 한다.

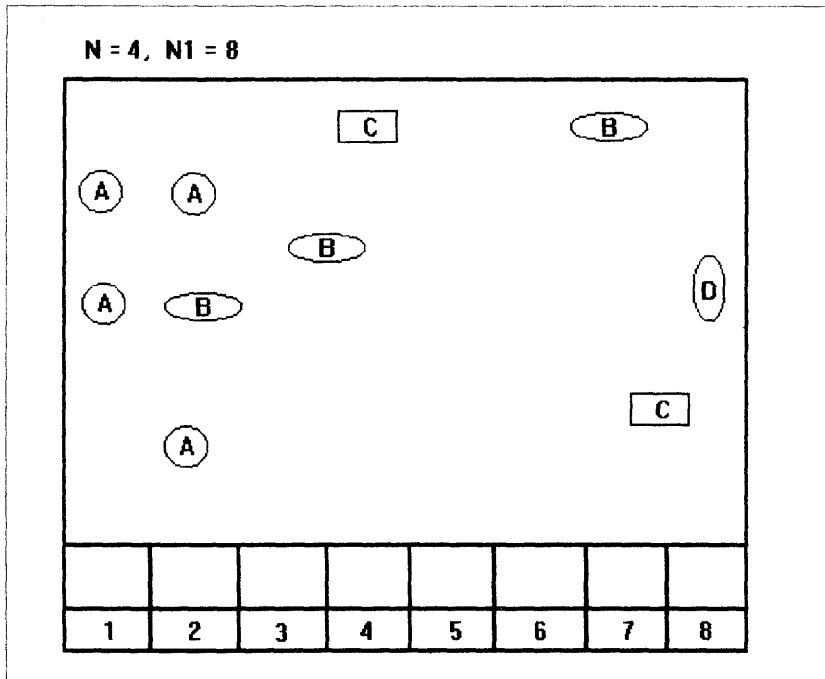
- N : 부품 종류 수 $n=1, \dots, N$
- N_i : PCB 둘레에 부품합이 놓일 수 있는 위치의 수
- M : 삽입되어야 할 총 부품 수
- D_i : 부품 타입 i 의 개수, $\sum_{i=1}^N D_i = M$
- $P_{ik} = (a_{ik}, b_{ik})$: 부품 타입 i 의 k 번째 부품의 PCB 상의 위치
- $C_i = (x_i, y_i)$: 부품 i 가 들어있는 i 타입 부품합의 위치
- $d(C_j, P_{ik})$: j 타입 부품합에서 PCB상 i 타입의 k 번째 부품까지의 거리

· 모델 MP 1

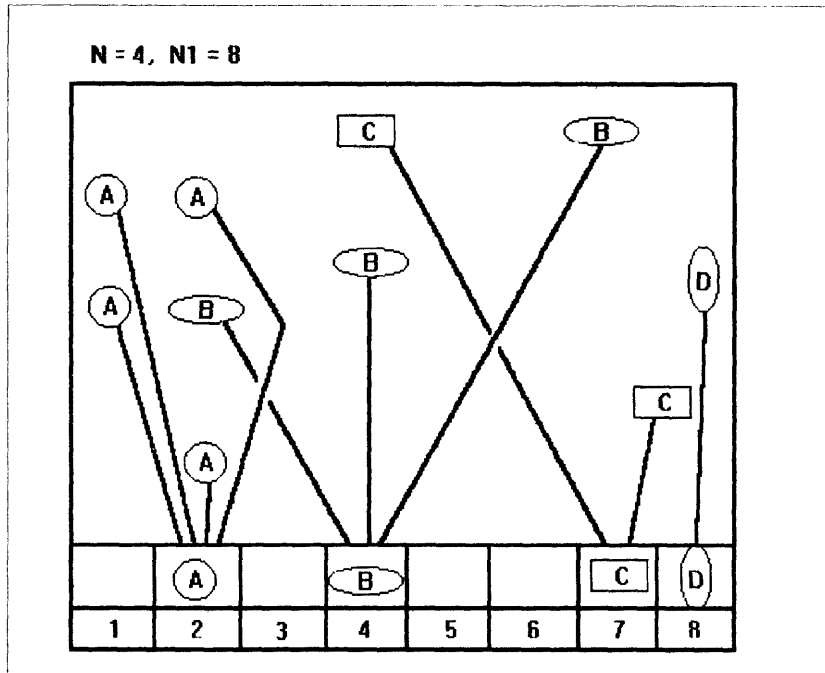
$$\begin{aligned} \text{Min} \quad & \sum_{j=1}^{N1} \sum_{i=1}^N Y(i,j) \cdot \left[\sum_{k=1}^{D_i} d(C_j, P_{ik}) \right] \quad (1) \\ \text{s.t.} \quad & \sum_{j=1}^{N1} Y(i,j) = 1 \quad i=1, \dots, N \quad (2) \\ & \sum_{i=1}^N Y(i,j) \leq 1 \quad j=1, \dots, N1 \quad (3) \\ & Y(i,j) = 0, 1 \quad \forall i, j \quad (4) \end{aligned}$$

- $Y(i,j)=1$: 부품 i 의 부품합이 j 의 위치에 있다.
- 0 : 부품 i 의 부품합이 j 의 위치에 있지 않다.
- $\sum_{k=1}^{D_i} d(C_j, P_{ik})$: 부품 i 의 모든 제품이 부품합 j 까지 거리의 합.

위 모델 MP1는 배정 문제이다. 위 식은 상용 패키지(LINDO, GAMS 등)를 이용해서 최적치를 구할 수 있다. [그림 2]와 같이 같이 각 부품이 주어졌을 때, 이를 모델화하여 GAMS 패키지를 이용하여 [그림 3]과 같은 결과를 얻었다.



[그림 2] 각 부품과 부품함 위치가 주어졌을 때



[그림 3] 주어진 각 부품에 대해 부품함 위치를 찾았을 때

IV. 삽입 경로와 위치 배정을 동시에 고려한 문제

부품함 위치를 고려한 경우가 고려하지 않은 경우보다는 전체 비용에서 개선됐음은 자명하다. 또한 각각 부품함 위치와 삽입순서를 나누어 풀것이 동시에 고려한 것보다 좋은 해는 아니다[5,12]. 삽입경로와 부품함 위치 문제를 동시에 고려한 Leipälä 와 Nevalainen[7]은 Quadratic assignment TSP으로 모델을 제시했다. Hamacher 등은 삽입 경로 문제와 부품

함 위치 배정 문제를 서로 보완하면서 해를 찾는 발견적 기법을 제시했다[4]. 이 절에서는 삽입 경로 문제는 부품함 위치 배정 문제를 동시에 풀 수 있게 모형화 해서 해법을 찾는다. 모형화는 II 절의 제안 1을 만족해야 한다. 부품함의 위치는 앞 장에서 본 바와 같이 언제나 구획으로 나누어 생각한다.

목적함수는 로봇트 팔의 움직인 총 거리의 합을 최소화 하는 것이다. 이는 부품들에서 부품함까지의 거리의 합과 부품함에서 부품들까지의 거리의 합을 합한 것이다. 이를 모형화 하면 다음 MP2식과 같다.

· 모델 MP2

$$Min \sum_{i=1}^N \sum_{k=1}^{D_i} \sum_{j=1}^{N1} d(C_j, P_{ik}) \cdot Y(ij) + \sum_{i=1}^N \sum_{j=1}^{N1} \sum_{k=1}^{D_i} d(P_i, C_j) \cdot X(ikj) \tag{5}$$

$$s.t. \sum_{j=1}^{N1} Y(ij) = 1, \quad i = 1, \dots, N \tag{6}$$

$$\sum_{i=1}^N Y(ij) \leq 1, \quad j = 1, \dots, N1 \tag{7}$$

$$\sum_{j=1}^{N1} X(ikj) = 1, \quad i = 1, \dots, N ; k = 1, \dots, D_i \tag{8}$$

$$\sum_{j=1}^N \sum_{k=1}^{D_i} X(ikj) = \sum_{i=1}^N Y(ij) \cdot D_i, \quad j = 1, \dots, N1 \tag{9}$$

$$\sum_{k=1}^{D_i} X(ikj) \leq Y_{ij}(D_i - 1) + MX(1 - Y_{ij})$$

$$MX = Max(D_i) + 1, \quad i = 1, \dots, N \quad j = 1, \dots, N1, \quad i = 1, \dots, N \tag{10}$$

$$X(skj) \leq \sum_{k=1}^N Y(ij), \quad j = 1, \dots, N1, \quad s = 1, \dots, N, \quad k = 1, \dots, D_i \tag{11}$$

$$Y(ij), X(ikj) = 1 \text{ or } 0 \quad i = 1, \dots, N, \quad j = 1, \dots, N1, \quad k = 1, \dots, D_i \tag{12}$$

• 부호

- $d(P_{ik}, C_q)$: 부품 i 의 k 번째 부품에서, q 위치에 있는 부품함까지의 거리.
- $d(C_j, P_{sq})$: j 위치에 있는 부품함에서, 부품 s 의 q 번째 부품까지의 거리.
- $X(ikj)$: 1 또는 0의 값을 가지는 결정변수
1=부품 i 의 k 번째 부품이 j 위치에 있는 부품함과 연결되었을 때
0=그 외 경우
- $Y(ij)$: 1 또는 0의 값을 가지는 결정변수
1=부품 i 가 담겨진 부품함이 j 위치에 있을 때
0=그 외 경우

• 제약식

- 식(7): 모든 부품함은 단 한 위치에만 놓여야 한다.
- 식(8): 부품함 후보 위치에 부품함이 놓일 수 있다.
- 식(9): 모든 부품들은 단지 하나의 부품함에 연결된다.

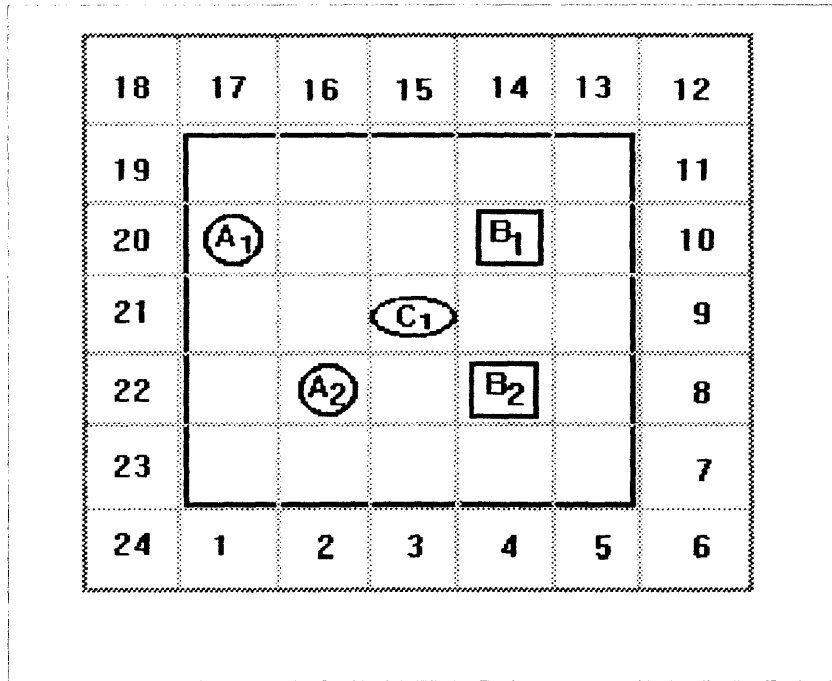
- 식(10): 모든 부품함에 든 부품들은 전부 삽입되어야 한다.
- 식(11): 같은 타입의 부품 모두가 같은 타입이 든 부품함에 연결 될 수는 없다.
- 식(12): $X(ikj)$ 변수는 $Y(ij)$ 변수에 종속적이다.
- 식(13): 모든 결정변수는 0 또는 1의 값을 가진다.

위 모델 MP2는 0-1 선형 정수 계획법 문제로서 기존의 패키지로 직접 풀 수 있다. 모델 MP2는 우체부 문제나 외판원 문제 보다는 훨씬 쉽다. MP2 식을 풀었을 때 부분환이 존재하지 않으면 이 때가 최적해이다. 부분환이 존재하면 최적해를 제시하진 못한다.

예제 1: 아래 [그림 4]와 같고, 부품 위치는 <표 1>과 같다. 즉 각 부품의 좌표는 A1(1,4), A2(2,2), B1(4,4), B2(4,2) 그리고 C1이 (3,3)에 주어졌다. 부품함 위치 가능 지역은 PCB 둘레에 가능하다. 각 둘레에 표시된 위치는 숫자로 표시했다.

<표 1> PCB 둘레의 부품함 가능 위치

| | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|
| PCB 둘레 번호 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 부품함 좌표 | 1,0 | 2,0 | 3,0 | 4,0 | 5,0 | 6,0 | 6,1 | 6,2 |
| PCB둘레번호 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 부품함 좌표 | 6,3 | 6,4 | 6,5 | 6,6 | 5,6 | 4,6 | 3,6 | 2,6 |
| PCB둘레번호 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 부품함 좌표 | 1,6 | 0,6 | 0,5 | 0,4 | 0,3 | 0,2 | 0,1 | 0,0 |



[그림 4] PCB에 부품의 위치가 주어졌을 때

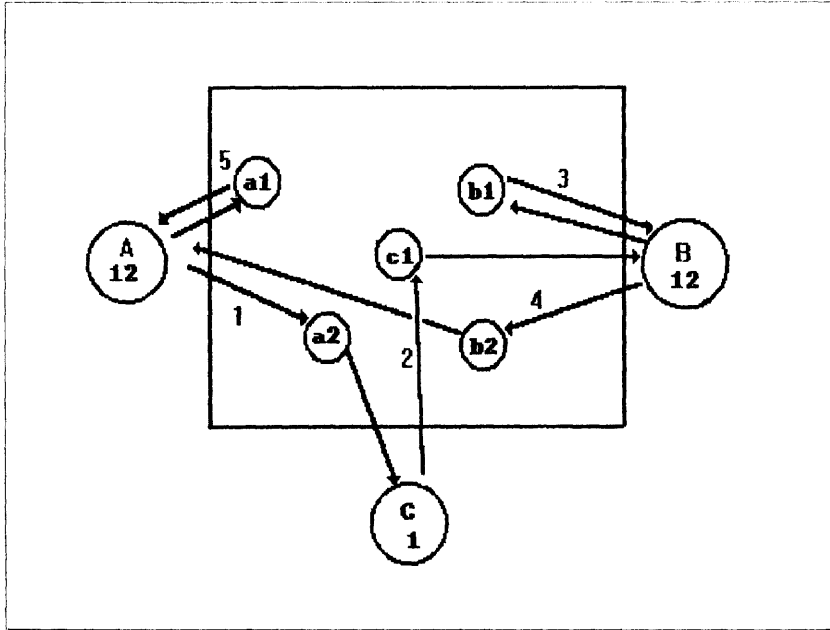
먼저 이 문제에 대해 부품합의 위치를 구한 후 삽입순서를 구해보자. 로봇트 운영방식 세 가지 경우 l_1 -norm, l_2 -norm, l_∞ -norm에 따라 전절에서 논의한 기법으로 부품합의 위치를 계산하면 세 경우 모두 똑같이 A 부품합 위치 좌표는 (0,3), B=(6,3), C=(3,0) 이다. 다음에 제시한 알고리즘 ISFA를 이용하여 삽입순서를 결정한다. 다음엔 이를 MP2를 이용

하여 3가지 운동방식 모두에 대해 풀었다. 이때 얻은 해는 모두 최적값이다. MP2를 풀 후에 부분환이 생기지 않았기 때문이다. 이때 얻은 부품합 위치를 번호로 나타내면 아래 <표 2>와 같다. [그림 5]~[그림 10]에서 기존의 방법과 개선된 알고리즘 ISFA 기법을 각각 보여 주었다.

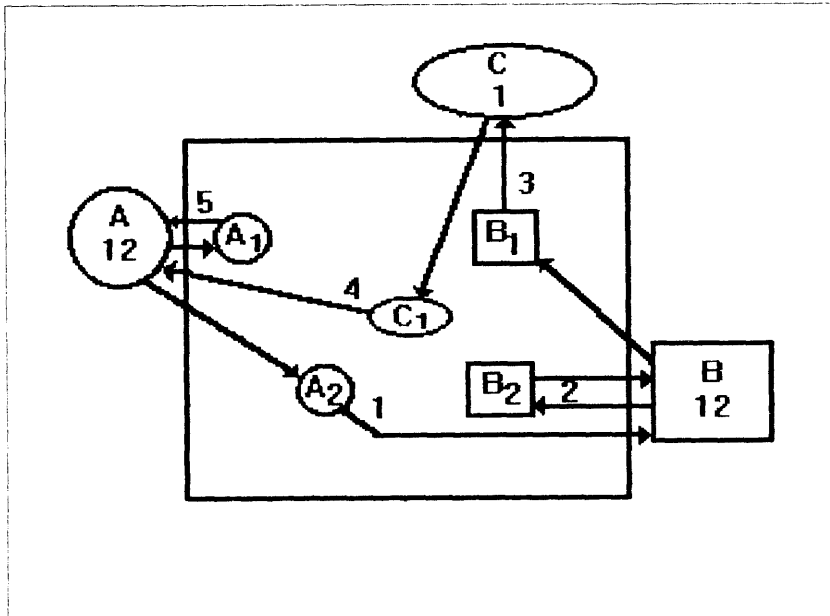
<표 2> MP2를 이용한 각 운동방식에 따른 부품합의 위치 번호

| | | l_1 -norm | l_2 -norm | l_∞ -norm |
|---|-------|-------------|-------------|------------------|
| A | 부 품 합 | 20 | 20 | 21 |
| B | 부 품 합 | 8 | 9 | 10 |
| C | 부 품 합 | 14 | 21 | 22 |

1) l_1 -norm 일 때

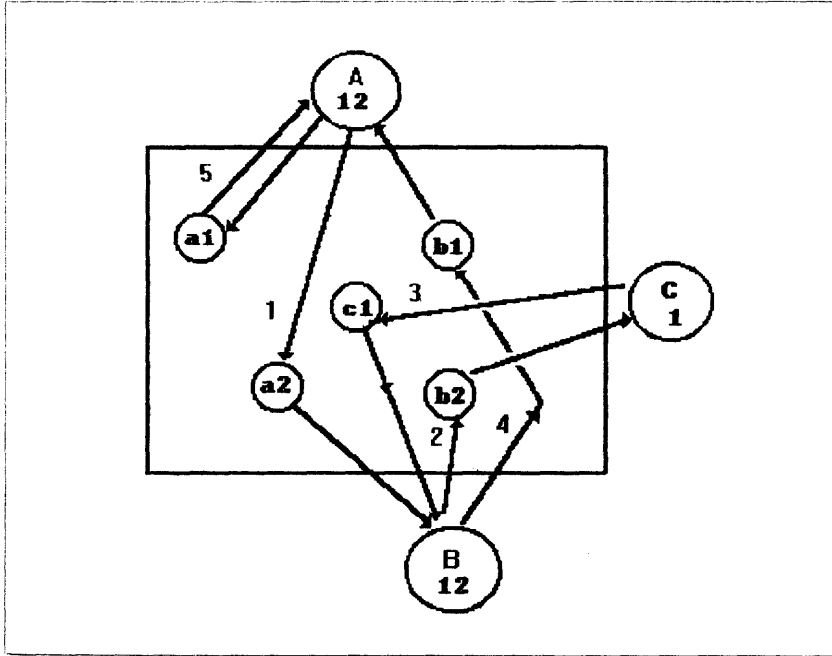


[그림 5] l_1 -norm로 기존의 방법으로 풀었을 때

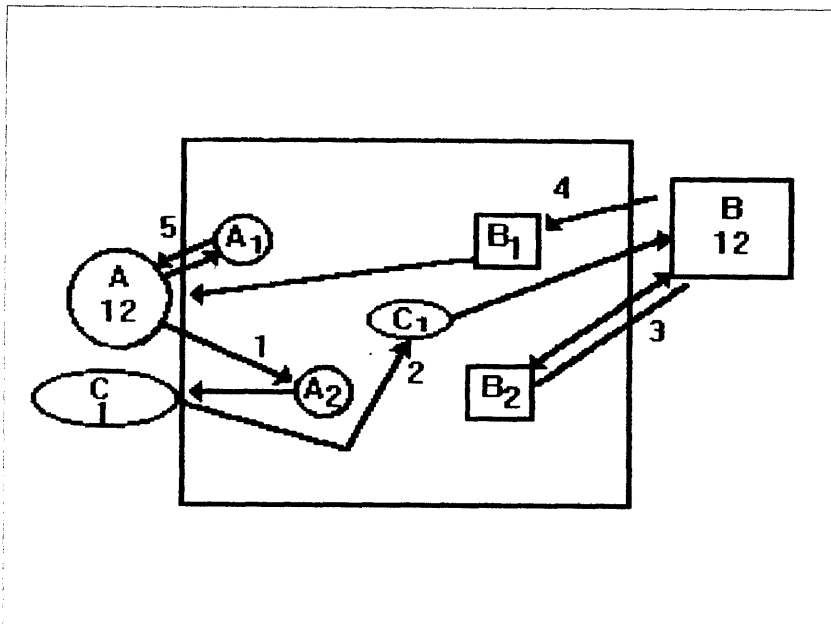


[그림 6] l_1 -norm로 ISFA 풀었을 때

2) l_2 -norm 일 때

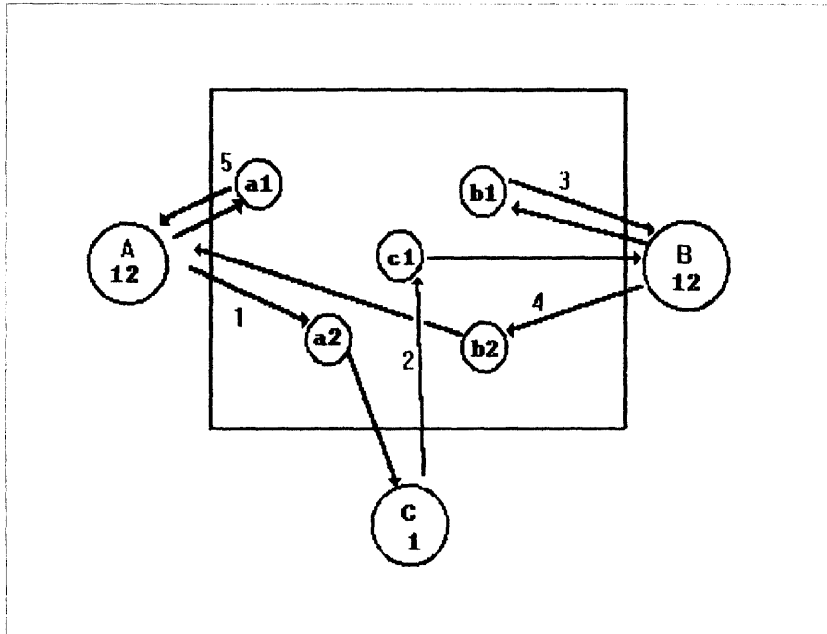


[그림 7] l_2 -norm로 기존의 방법으로 풀었을 때

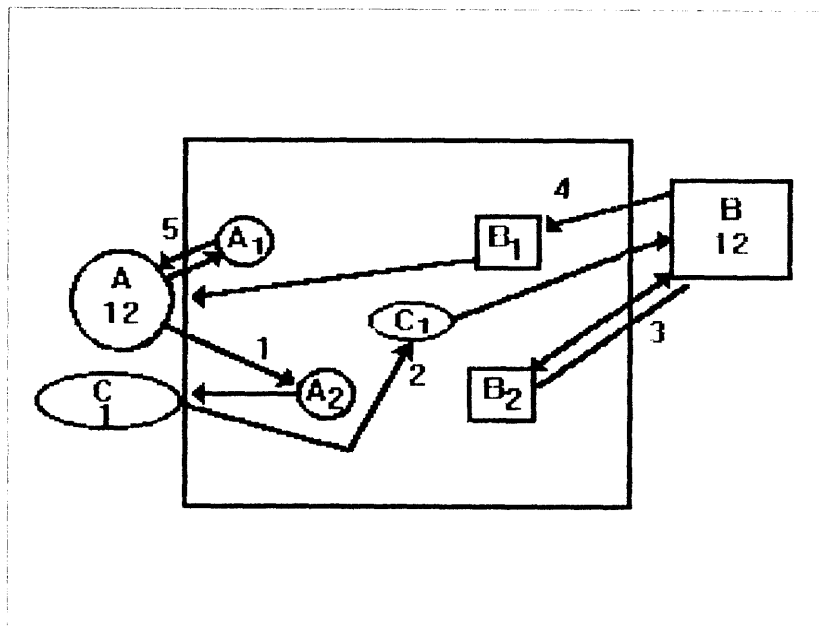


[그림 8] l_2 -norm로 ISFA 풀었을 때

3) l_x -norm 일 때



[그림 9] l_x -norm로 기존의 방법으로 풀었을 때



[그림 10] l_x -norm로 ISFA 풀었을 때

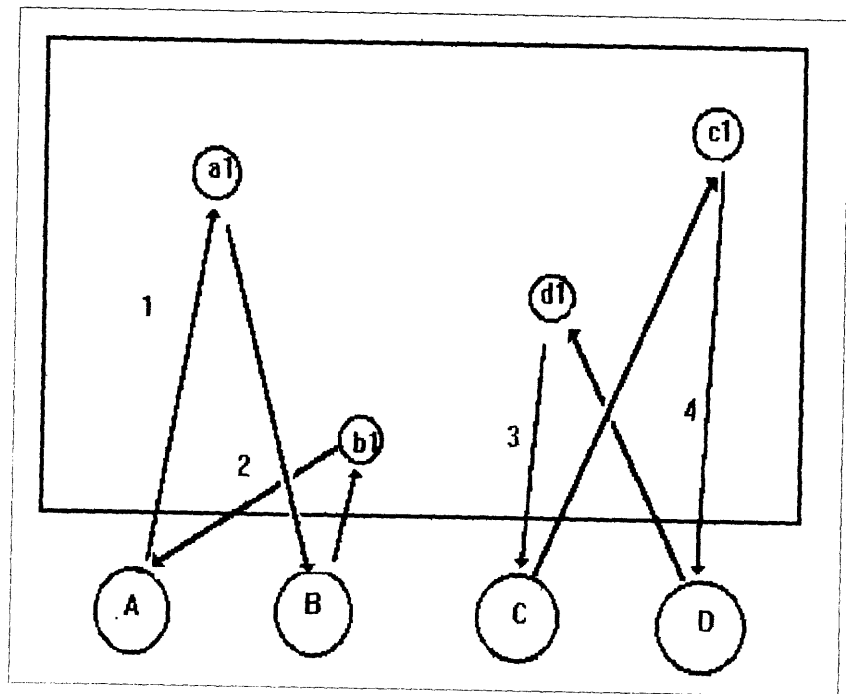
이를 정리하면 다음 <표 3>과 같다.

<표 3> 각 로봇트가 움직인 총 거리

| | l_1-norm | l_2-norm | $l_\infty-norm$ |
|---------------------------------------|------------|------------|-----------------|
| 부품함과 삽입순서를 따로 계산 했을 때 | 30 | 23.9 | 22 |
| MP2를 이용하여 부품함과 삽입순서를 함께 계산 했을 때.(최적해) | 28 | 23.7 | 22 |

MP2를 패키지로 푼 후에 부분환이 존재하는 지는 오일러 환을 찾으면 알 수 있다. 모든 점

이 오일러 환에 들어가면 부분환이 없는 것이고, 그렇지 않으면 부분환이 존재하는 것이다.



[그림 11] 부분환이 생기는 경우

부분환 제거 : 총거리(식5)를 최소로 증가시키는 조건 하에서, 주어진 제약 조건식을 만족하면서 부분환을 제거하는 방법을 찾는다. 부분환이 있을 때는 반드시 둘 이상이 있다.

· 부분환의 순서는 왼쪽부터 차례로 $S_i, i = 1, \dots, i$ 이라 하자. 즉 [그림 9] 부분환 $A \rightarrow a1 \rightarrow B \rightarrow b1 \rightarrow A$ 은 S_1 이다.

· 선택 부품함 : 어느 부분환에서 가장 가까이 있는 다른 부분환의 부품함. [그림 9]에서 부분환 S_i 의 선택 부품함은 C 이다. 이는 SC_i 이라 나타내자. $SC_i = C$ 이다. C 는 부분환 S_j 에 속해있다.

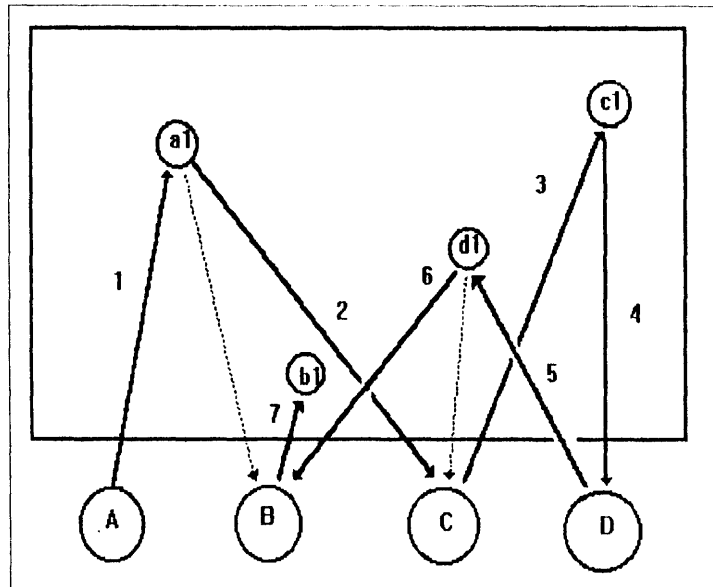
· 선택 경로 : 부분환을 제거하기 위해 부분환에서 선택된 경로다. 선택경로는 두 부분환을 연결하고자 이웃한 부분환의 선택 경로에서 하나씩 2개를 동시에 선택한다. 부분환 S_i 의 선택 부품함 SC_i 에 연결되는 경로 중에서 전체 거리를 가장 적게 증가시키는 경로를 $SC_{i,k}$ 에

로 연결한다. 동시에 부분환 $S_{i,k}$ 의 선택 부품함 $SC_{i,k}$ 에 연결되는 경로 중에서 전체 거리를 가장 적게 증가시키는 경로를 $SC_{i,k}$ 에 연결한다. 선택경로 SP_i 는 선택 부품함 SC_i 에서 선택된 경로이다. [그림 9]에 $SC_i = C$ 에 후보 선택 경로는 $d(d1,C)$ 와 $d(c1,C)$ 이고 이중 $d(d1,C)$ 이 더 적게 전체 거리를 증가시키므로 이를 선택한다. 즉 $SP_i = d(d1,C)$ 이다. 이를 수식으로 표현하면,

$$SP_i = \text{Min}\{d(P_{i,k}, SC_i) | d(P_{i,k}, SC_i) \text{ and } d(P_{i,k}, SC_i) \in S_j, i = 1, \dots, N, k = 1, \dots, D_i\}$$

$$SP_{i,k} = \text{Min}\{d(P_{i,k}, SC_{i,k}) | d(P_{i,k}, SC_{i,k}) \text{ and } d(P_{i,k}, SC_{i,k}) \in S_{j,k}, i = 1, \dots, N, k = 1, \dots, D_i\}$$

두 부분환 제거 : 선택 경로를 서로 교환한다. 즉 두 부분환을 제거한다.



[그림 12] 부분환 제거

부분환 제거 알고리즘

1. 두 부분환으로부터 두 선택 경로를 선택한다.
2. 두 선택 경로를 서로 교환한다.
3. 모든 부분환이 제거 될 때까지 계속 진행한다.

주어진 문제에서 부분환을 모두 제거하고, 삽입 순서를 찾음으로서 PCB상의 삽입 경로 문제는 끝난다. 오일러 환은 간단한 알고리즘 (Fleury's Algorithm)[9]을 이용함으로써 쉽게 구할 수 있다. PCB상에서 삽입 경로 문제 및 부품함 위치 배정 문제 알고리즘은 다음과 같다.

알고리즘 ISFA

- 단계 1 : 실제문제를 좌표문제로 변환한다.
 단계 2 : 모델 MP2를 푼다.
 단계 3 : 부분환을 찾는다.
 단계 4 : 만약 부분환이 없으면 최적이다.
 부분환 있으면 부분환 제거 알고리즘을 이용하여 부분환을 제거하고 끝낸다.

제안 2 : 알고리즘 ISFA를 사용하면, 부분 최적해를 제시한다. 전체 최적해에 비해서 최대 부분환 크기보다 작은 오차 안에서 해를 제시해 준다.

증명 : 알고리즘 ISFA는 모델 MP2를 푼 후에 부분환이 존재하지 않으면 최적해를 준다. 그러나 부분환이 생기면 부분환 제거 알고리즘을 이용해 강제로 부분환을 제거한다. 이 때 오차

가 생길 수 있으며 오차의 최대범위는 가장 큰 부분환을 잘못 제거하는 경우이다. ■

V. 결 론

본 연구에서는 실제 PCB 생산시에 삽입 작업을 개선함으로써 전체 생산 비용을 줄이고 생산율을 높이고자 한다. 삽입 작업 개선을 위하여 삽입 경로 문제와 이에 따른 부품함 위치 배정 문제에 대해서 살펴보았다. 기존의 방법이 부품에 중점을 둔데 비해, 부품함에 중점을 두어, 새로운 발견적 기법을 제시했다. 삽입 경로 문제와 부품함 위치배정 문제를 동시에 고려한 모델식 MP2를 제시하고 알고리즘 ISFA를 제시했다. 이 해법은 현장에서 쉽게 적용할 수 있다.

추후 연구과제로서는, 현장에서 사용될 로봇 팔이 여러 개의 부품을 잡고 움직이는 경우와 여러개의 로봇 팔이 동시에 움직이는 형식에 따른 새로운 알고리즘의 개발이 필요해졌다.

참 고 문 헌

[1] Ahmadi, J., S. Grotzinger and D. Johnson, "Evaluation of Concurrency in Circuit Card Assembly Machines", *Research Report RC-12161 IBM*, 1986.
 [2] Ball, M.O. and M.J. Magazine, "Sequencing of Insertions in Printed Circuit Board Assembly," *Opns. Res.*, 36

- (1988), 192-201.
- [3] Francis, R. L. and J.A. White, *Facility Layout and Location: An Analytical Approach*, Prentice-Hall, 1974, New-York.
- [4] Francis, R. L., H.W. Hamacher, C. Y. Lee, and S. Yeralen, "Finding Placement Sequence and Bin Locations for Cartesian Robots", *Research Report: DDT-8817840* 1991.
- [5] Foulds, L.R. and H.W. Hamacher, "Optimal Bin location and Sequencing in printed Circuit Boards assembly", *Research Report Dept. of Math Univ. KL* 1990.
- [6] Hamacher, H. W. and S. Nickel, "Restricted Planar Location Problems and Applications", *Technical Report. Dept. of Math Univ. KL* 1992.
- [7] Leipälä, Timo and Olli Nevalainen, "Optimization of the Movements of a Component Placement Machine", *European Jour. of O.R.* 38(1989), 167-177.
- [8] Lenstra, J.K. and A.H.G. Rinnooy Kan, "Complexity of Vehicle Routing and Scheduling Problem", *Network*, 11(1981), 221-227.
- [9] Nijenhuis, C.B. and H. Wilf, *Combinatorial Algorithms*, Academic Press, New York 1975.
- [10] Shevell, S.F., J.a. Busacott and M.J. Magazine, "Simulation and Analysis of a Circuit Board Manufacturing Facility", *In the Proceedings of the 1986 Winter Simulation Conference*. (December), 1986, Washington, D.C
- [11] Ree, Sangbok, "A Study of Optimal Multi part type Inserting Sequence and Container Allocation on Printed Circuit Board", *Maser thesis, University of Kaiserslautern, Germany* 1992.
- [12] Ree, Sangbok, "A Study of Optimal Inserting Sequence and Feeder Allocation Problem on Printed Circuit Board", *PH. D. thesis, Seoul National University* 1993.