

# PCB 조립 공정의 작업 투입 순서 및 부품함 배치 문제에 관한 연구

유성열\* · 이강배\*\*

\*부산가톨릭대학교 유통경영정보학부, \*\*동아대학교 경영정보과학부

## A Study on Job Sequence and Feeder Allocation Problem in PCB Assembly Line

Sung-Yeol Yu\* · Kang-Bae Lee\*\*

\*Division of Distribution Management & MIS, Catholic University of Pusan

\*\*Division of Management Information Science, Dong-A University

In this paper, we consider a planning problem arising from printed circuit board manufacturing industries. Given a set of several types of PCBs, component feeders and surface mounting machines in series in a PCB assembly line, the problem is to define the feeder allocation and job sequence with the objective of minimizing the total operation time of the line. We formulate the problem as a mathematical model. And, the problem is proven to be NP-hard, so a genetic algorithm is developed. Finally, we give test results to evaluate the performance of the genetic algorithm.

**Keywords** : printed circuit board, job sequence, feeder allocation, genetic algorithm

### 1. 서론

인쇄회로기판(PCB; Printed Circuit Board)은 전자 산업 분야에서 필수적인 부품으로 광범위한 분야에서 다양하게 사용되고 있다. 이러한 PCB의 생산은 전형적인 조립 공정을 통해서 이루어진다.

일반적으로, 단일 혹은 소수의 제품을 다루는 소품종 대량생산 환경의 경우에는 부품 장착 속도가 빠른 전용 실장기를 이용하여 작업을 진행하게 되고 이때의 공정 작업 시간은 각 PCB에 대한 부품 장착 순서와 부품함 배치에 영향을 받게 된다. 이와는 달리, 다품종 소량 생산 환경에서는 상대적으로 그 조립 속도가 느리더라도 거의 모든 종류의 부품을 다룰 수 있는 다기능 표면 실장기 (Multi-functional Surface Mounting Machine)를 요하며, 이 때의 공정 작업 시간에 영향을 끼치는 요인은 부품 장착 순서와 부품함 배치 뿐만 아니라, 각각의 PCB 투입 순서도 큰 영향을 미치게 된다[1].

본 연구에서는, 두 대의 다기능 표면 실장기로 이루어

진 PCB 조립 공정에서 다양한 종류의 PCB 유형들을 소량씩 생산하는 다품종 소량 생산 환경에서의 효율적인 PCB 투입 순서 및 부품함 배치 문제를 다룬다.

PCB 조립 공정에서 발생하는 작업 순서 혹은 부품함 배치에 관련된 문제를 다룬 연구로는 먼저, Carmon *et al.*[1]의 연구가 있다. 이들은 다양한 유형의 PCB를 생산하기 위한 공정에서 그룹 테크놀로지(group technology) 개념에 기초하여, 작업 준비 시간 감소를 위한 GSU(group set-up) 방법을 제안하였다. Maimon *et al.*[9]은 생산될 PCB 유형간의 부품 공통성을 기준으로 공통성이 큰 PCB 들을 연속적으로 투입함으로써 부품함 교체 시간을 줄이고자 하는 기법으로써 SDS(sequential dependent set-up) 방법을 제안하였다. GSU나 SDS 방법에서는 각 기계에서 사용할 부품함들의 배치는 사전에 미리 결정되어 있다고 가정하여 문제의 범위에는 포함시키지 않았다. Crama *et al.*[2] 와 Yu *et al.*[11] 등은 한 대의 기계로 제품을 생산하는 경우에 부품함 위치 결정 문제를 다루었다.

본 연구에서 다루고자 하는 문제인 두 대의 기계가

있는 조립 공정에서 PCB 투입 순서와 부품함 배치 문제를 동시에 다룬 연구로는 Gronalt & Zeller[5]의 연구가 있다. 이들은 기계 1과 기계 2의 작업 부하(work load) 차이와 PCB 유형별로 각 기계에 할당되는 부품함의 수를 측정 기준으로 설정하고, 전체 작업 시간을 최소화하기 위한 발견적 기법을 제시 하였다. 이들은 PCB 조립 공정의 성과 측정치(measure)로써, 두 기계간의 작업 부하의 차이와 전체 작업 공정 중에 두 기계에서 모두 사용되는 부품함의 수를 작업 투입 순서는 고려하지 않고 단순히 합한 값을 이용하였다. 그러나 실제 PCB 조립 공정에서는 투입 순서에 따라 공통으로 사용되는 부품함의 숫자는 달라질 수 있다. 따라서 본 연구에서는 작업에 투입되는 PCB 유형이 바뀔 때마다 교체되는 부품함의 수를 반영함으로써 보다 현실적인 상황을 묘사한다. 아래에 본 연구에서 다루고자 하는 문제에 대해 자세한 설명을 제시한다.

먼저, 생산되어야 할 PCB 유형들이 주어지 있다. 각 PCB 유형은 다수의 동일한 PCB로 이루어진다. 조립 라인을 구성하는 두 대의 실장기의 작업 속도는 서로 다를 수 있다. 또한 각 부품들은 서로 다른 부품함에 의해 공급된다. 각 실장기에는 부품함을 놓을 수 있는 슬롯(slot)의 숫자가 한정되어 있어, PCB가 요구하는 부품의 종류가 실장기의 슬롯 수 보다 많을 경우에는 두 대의 실장기 모두에서 장착 작업을 거쳐야 한다. 이 경우, 어떤 부품함을 어떤 실장기에 장착할 것인지가 해당 PCB 군의 조립 작업을 시작하기 전에 결정되어야 한다. 두 실장기 사이에 버퍼(buffer)는 존재하지 않으며, 따라서 첫 번째 실장기에서 부품 조립이 끝난 PCB는 바로 두 번째 실장기로 옮겨져 조립 작업이 진행된다. 주어진 PCB 유형의 모든 PCB에 대한 조립이 끝나면, 새로운 PCB 유형의 작업이 시작되기 전에 각 실장기에 장착된 부품함의 위치가 변경될 수 있다. 이 경우, 부품함을 교체한 시간만큼 전체 작업 시간은 길어지게 된다. 따라서 주어진 PCB 유형의 조립 작업을 모두 완료하는데 걸리는 시간은 PCB 유형들의 투입 순서와 각각의 PCB 조립에 필요한 부품함들을 어느 기계에 배치하였느냐에 따라 달라진다. 다음 장에서는 지금까지 설명한 PCB 조립 공정에서의 작업 투입 순서 및 부품함 배치 문제에 대한 수리 모형을 제시하고, 문제의 난이도에 대하여 논의한다.

## 2. 수리 모형

주어진 문제를 수리적으로 표현하기 위하여 사용될 기호를 다음과 같이 정의한다.

- $P$  : PCB 유형의 집합 ( $|P| = \alpha$ )
- $B$  : 부품함의 집합 ( $|B| = \beta$ )
- $M$  : 기계의 집합
- $B_i$  : PCB 유형  $i$ 에 사용되는 부품들을 공급하는 부품함들의 집합
- $s_k$  : 부품함  $k$ 가 차지하는 슬롯의 수
- $c^d$  : 기계  $d$ 에서 부품함 하나를 교체하는 데 걸리는 시간(차지하는 슬롯의 수에 비례함)
- $N^d$  : 기계  $d$ 의 총 슬롯의 수
- $n_{ik}$  : PCB 유형  $i$ 에 필요한 부품 중 부품함  $k$ 로부터 공급되는 부품의 수
- $o^d$  : 기계  $d$ 의 단위 시간당 부품 장착 수(작업 속도)
- $q_i$  : PCB 유형  $i$ 에 속한 PCB의 수
- $u_{ik}^d$  : 기계  $d$ 에서 PCB 유형  $i$ 에 속하는 모든 PCB에 부품함  $k$ 에 의해서 공급되는 모든 부품을 장착하는데 걸리는 시간 ( $= q_i n_{ik} / o^d$ )

모형에서 사용될 의사 결정 변수(decision variables)를 다음과 같이 정의한다.

- $z_{ij}$  : PCB 유형  $i$ 에 대한 작업이 완료된 후 바로 유형  $j$ 에 대한 작업이 시작될 경우 1, 아닌 경우 0.
- $y_{ik}^d$  : PCB 유형  $i$ 에 필요한 부품함  $k$ 가 기계  $d$ 에 장착될 경우 1, 아닌 경우 0.
- $w_i^d$  : 기계  $d$ 상에서 PCB 유형  $i$ 의 작업부하
- $W_i$  : PCB 유형  $i$ 의 두 기계간의 작업부하 차이

이상과 같은 기호 및 변수들을 이용하여, PCB 공정에서의 주어진 작업 순서 및 부품함 위치 결정 문제는 다음과 같은 수리 모형으로 정의 될 수 있다.

$$(P) \text{ Minimize } \sum_{i \in P} W_i + \sum_{i \in P} \sum_{j \in P} \sum_{d \in M} c^d (\max [0, h(i, j, d) - N^d]) z_{ij} \dots (1)$$

subject to

$$w_i^1 - w_i^2 \leq W_i, \quad \text{for all } i \in P \dots (2)$$

$$w_i^2 - w_i^1 \leq W_i, \quad \text{for all } i \in P \dots (3)$$

$$\sum_{k \in B_i} u_{ik}^d y_{ik}^d = w_i^d, \quad \text{for all } i \in P, d \in M \dots (4)$$

$$\sum_{d \in M} y_{ik}^d = 1, \quad \text{for all } i \in P, k \in B_i \dots (5)$$

$$\sum_{i \in P} z_{ij} = 1, \quad \text{for all } j \in P \dots\dots\dots (6)$$

$$\sum_{j \in P} z_{ij} = 1, \quad \text{for all } i \in P \dots\dots\dots (7)$$

$$\lambda_i - \lambda_j + (\alpha + 1)z_{ij} \leq \alpha, \quad \text{for all } i, j \in P \dots\dots\dots (8)$$

$$\sum_{k \in B_i} s_k y_{ik}^d \leq N^d, \quad \text{for all } i \in P, d \in M \dots\dots\dots (9)$$

$$z_{ij} \in \{0, 1\}, \quad \text{for all } i, j \in P \dots\dots\dots (10)$$

$$y_{ik}^d \in \{0, 1\}, \quad \text{for all } i \in P, k \in B_i, d \in M \dots\dots (11)$$

$$w_i^d \geq 0, \quad \text{for all } i \in P, d \in M \dots\dots\dots (12)$$

$$W_i \geq 0, \quad \text{for all } i \in P \dots\dots\dots (13)$$

목적함수 (1)은 두 가지 항목의 합으로 구성된다. 첫 번째 항은 두 기계간의 작업부하 차이를 나타내며, 두 번째 항은 각 기계에서 PCB 유형이 교체될 때 마다 발생하는 부품함 교체시간을 나타낸다. 이 두 항목은 모두 시간의 함수로 표시되므로 식(1)에서와 같이 합으로 나타낼 수 있다. 여기에서, 함수  $h(i, j, d)$  는 연속되는 두 PCB 유형  $i$  와  $j$  가 기계  $d$  상에서 필요로 하는 총 슬롯의 수를 나타내며, 다음 식에 의해 계산된다.

$$h(i, j, d) = \sum_{k \in B_i} s_k y_{ik}^d + \sum_{k \in B_j} s_k y_{jk}^d - \sum_{k \in B} s_k y_{ik}^d y_{jk}^d \dots (14)$$

식 (14)의 첫 번째 항은 기계  $d$  에서 PCB 유형  $i$  에 필요한 부품함들이 차지하는 슬롯의 수이며, 마찬가지로 두 번째 항은 유형  $j$  에 필요한 부품함들이 차지하는 슬롯의 수이다. 세 번째 항은 유형  $i$  와  $j$  에 공통으로 사용되는 부품함들이 차지하는 슬롯의 수이다.

제약식 (2)와 (3)은 두 기계의 작업부하 차이를 결정하는 식이다. 두 기계간의 작업 부하 차이  $W_i$ 는  $w_i^1$ 과  $w_i^2$ 의 차이, 즉  $W_i = |w_i^1 - w_i^2|$  로 계산될 수 있다. 이를 수식으로 표현한 것이 식 (2)와 (3)이다. 제약식 (4)는 각 기계별 PCB별 워크로드를 정의하기 위한 수식이다. 제약식 (5)는 어느 한 PCB 유형에 사용되는 동일한 부품들은 하나의 기계에서 모두 사용됨을 의미한다. 즉, 한 종류의 PCB 유형에 대하여 같은 부품들이 두 대의 기계에 나누어 할당되지 못함을 의미한다. 제약식 (6), (7), (8)은 어느 한 유형의 PCB에 대한 작업이 끝나면, 다른 PCB의 작업이 시작되며, 모든 PCB 유형들은 정확히 한 번씩만 작업이 진행됨을 나타낸다. 이 제약식들은 일반적인 우편배달부문제(TSP; Travelling Salesman Problem)에 관한 제약으로, 식 (8)은 TSP 문제의 서브투어(sub-tour)를 방지하기 위한 식으로  $\lambda_i (\lambda_j)$ 는 임의의 실

수이다. 제약식 (9)는 각 기계에 동시에 장착되는 부품함의 수가 해당 기계의 총 슬롯의 수를 초과할 수 없음을 나타내는 제약식이다. 제약식 (10)과 (11)은 두 변수  $z_{ij}$  와  $y_{ik}^d$ 은 0 또는 1의 값을 갖는 이진 변수(binary variables)임을 의미한다. 제약식 (12)와 (13)은 작업부하 및 두 기계에서의 작업부하 차이가 항상 0보다 큰 값을 가짐을 의미한다.

모형 (P)는 본 연구에서 다루고자 하는 PCB 투입 순서 및 부품함 배치 문제를 수학적으로 표현한 것이다. 모형 (P)는 목적함수가 비선형이며, 제약식에 TSP 제약과 배낭 문제(Knapsack Problem) 제약을 포함하고 있다. 따라서 본 문제는 최적해를 구하는 것이 매우 어려운 (NP-hard) 문제이다[10]. 이에, 본 연구에서는 합리적인 시간 내에 최적해에 가까운 근사해를 제시해 줄 수 있는 유전 알고리즘에 근거한 해법을 제시하고자 한다.

### 3. 유전 알고리즘

유전 알고리즘은 적자생존과 자연 도태의 원리를 바탕으로 집단을 구성하는 각 개체에 대하여 목적 함수 값과 제약 조건을 위반하는 정도에 따라 적합도 값을 구하고, 이 값이 큰 개체를 다음 세대로 진화시키는 기법을 통해 최적해에 가까운 근사해를 찾는 방법으로 알려져 있다. 이러한 유전 알고리즘은 일반적인 발견적 기법(heuristic)에 비해 다음과 같은 몇 가지 특징을 지니고 있는 것으로 알려져 있다[4]. 먼저 발견적 기법은 문제의 유형에 따른 고유한 해법을 일일이 개발해야 하나 유전 알고리즘은 거의 모든 유형의 문제에 대하여 일관성 있는 해법 절차를 적용한다. 또한, 발견적 기법에서는 주어진 상황이나 가정이 조금만 달라져도 종전의 해법이 더 이상 적용 불가능한 경우가 많으나, 유전 알고리즘에서는 변경된 사항에 따라 목적 함수 및 제약 조건의 코딩만 일부 수정하면 지속적인 사용이 가능하다.

이러한 유전 알고리즘은 이미 다양한 분야의 최적화 문제에 대하여 적용되어 오고 있다. 특히, 최근에는 PCB 공정과 관련된 여러 가지 유형의 문제에 대해서도 적용되고 있다. Maimon & Braha[8]는 한 대의 기계를 이용한 PCB 조립 공정에서의 일정 계획 수립 문제를 해결하기 위한 알고리즘을 제시하였는데, 그들의 알고리즘은 유전 알고리즘에 기반하고 있다. Khoo & Loh[7]의 경우에도 PCB 조립 공정의 공정 계획을 수립하기 위하여 유전 알고리즘 기반의 알고리즘을 제시하였다. Ji *et al.*[6]은 PCB 조립 공정에서의 최적 사이클 타임을 구하기 위한 유전 알고리즘 기법을 제시하였다. Deo *et al.*[3]은 하나의 PCB를 생산 하는 도중에 다른 PCB의 생산이 허용

되는 경우의 PCB 조립 공정 계획에 관한 문제의 해법으로 유전 알고리즘을 제시하였다.

기존의 PCB 공정에 대한 유전 알고리즘 해법들은 하나의 기계로 이루어진 공정에 관한 문제이거나, 여러 대의 기계로 이루어진 공정일지라도 준비시간에 대한 문제만을 다루거나, 혹은 부품함 배치에 대해서만 문제를 정의하고 다루고 있다. 이에 반해 본 연구에서 다루고자 하는 문제는 여러 대의 기계로 이루어진 PCB공정에서, PCB 투입 순서에 따른 부품함 교체 시간과 각 기계별 부품함 배치 문제를 동시에 다룬다. 이러한 문제의 유형은 보다 실질적인 적용성이 높은 문제라고 할 수 있다.

본 장에서는 본 연구에서 제시하고자 하는 유전 알고리즘에 대하여, 개체의 구조, 각 개체의 적합도 평가 함수 및 평가 절차, 유전 연산의 방법, 초기 개체의 구성 등에 대해 설명한 후, 전체 유전 알고리즘의 절차에 대해 제시한다.

### 3.1 개체 구조

본 연구에서 적용하는 개체의 형태는  $\alpha \times (\beta + 1)$  행렬  $X = \{x_{ik} : i = 1, 2, \dots, \alpha, k = 1, 2, \dots, \beta + 1\}$ 로 주어진다. 이 행렬은 두 개의 부행렬(sub-matrix)  $X_S$ 와  $X_A$ 로 이루어진다. 즉,  $X = (X_S, X_A)$ 를 의미하며, 이 때,  $X_S$ 는  $X$ 의 첫 번째 열(column)로만 구성된 부행렬이다.  $X_S$ 의 각 원소들은 첫 행(row)부터 차례대로 PCB 유형이 투입되는 순서를 의미한다.  $X_A$ 는 두 번째 열부터 마지막 열( $(\beta + 1)$ 번째 열)로 구성된 행렬이며, 어느 한  $x_{ik} \in X_A$ 는  $i$ 번째 행에 해당되는 PCB 유형 ( $X_S$ 의  $i$ 번째 행의 값)에 필요한 부품함 ( $k - 1$ )이 어느 기계에 배치되는가를 나타낸다. 만약 이 값이 1이면, 첫 번째 기계 배치됨을 의미하며, 이 값이 2이면 두 번째 기계에 배치됨을 뜻한다.

<그림 1>은 3가지 유형의 PCB와 4종류의 부품함이 주어진 경우에 나타날 수 있는 개체의 한 예이다.

$$X = (X_S, X_A) = \left( \begin{array}{c|cccc} 2 & 0 & 2 & 1 & 0 \\ 3 & 2 & 1 & 0 & 2 \\ 1 & 2 & 1 & 2 & 0 \end{array} \right)$$

<그림 1> 개체 구조의 예

<그림 1>에서 첫 번째 열 (부행렬  $X_S$ )의 2, 3, 1은 PCB 유형의 투입이 유형 2 → 유형 3 → 유형 1의 순으로 이루어짐을 나타낸다. 두 번째 열부터 5번째 열까지는 각 부품함이 어떤 기계에서 사용되는가를 나타낸다. 예를 들어, 첫 번째로 작업에 투입되는 PCB 유형 2

는 필요한 부품함이 2와 3이며, 이 때 부품함 2는 기계 2에 배치되고, 부품함 3은 기계 1에 배치됨을 뜻한다. 두 번째로 투입되는, PCB 유형 3은 사용되는 부품함이 1, 2, 4이며, 부품함 1은 기계 2에서, 부품함 2는 기계 1에서, 부품함 4는 기계 2에서 각각 사용됨을 뜻한다. 마지막 3번째 행에서 볼 수 있듯이, PCB 유형 1에서는 부품함 1, 2, 3이 필요하며, 부품함 1은 기계 2에서, 부품함 2는 기계 1에서, 부품함 3은 기계 2에서 각각 사용됨을 의미한다.

본 연구에서 사용하는 개체의 구조는 앞의 2장에서 제시한 수리 모형 (P)의 제약식들을 모두 만족하는 가능해(feasible solution)이다.

### 3.2 적합도 값의 평가

유전 알고리즘에서는 한 세대를 구성하는 모든 개체들의 적합도 값(fitness value)을 구한 후 이를 근거로 다음 세대의 새로운 개체들을 구성해 나간다. 따라서 적절한 적합도 함수의 선택은 전체 유전 알고리즘의 효과에 큰 영향을 미친다. 본 연구에서의 적합도 값은 모형 (P)의 목적함수인 식 (1)을 통해 쉽게 그 값을 구할 수 있다.

<그림 1>과 같은 개체가 주어질 경우의 적합도 값을 구하는 절차를 예를 통하여 설명하고자 한다. 먼저, 각 PCB 유형별로 생산해야 할 PCB가 각각 3개 ( $q_1 = q_2 = q_3 = 3$ )씩 주어질 것이라고 가정하자. 각 PCB 유형별로 필요로 하는 부품 수는 다음 <표 1>과 같이 주어질 것이라고 가정하자.

<표 1> PCB별 필요 부품수

부 품	PCB유형		
	1	2	3
1	1	0	1
2	3	1	2
3	1	2	0
4	0	0	1

또한, 각 기계에서 부품함 하나를 교체 하는데 필요한 준비 시간은 모두 1 ( $c^1 = c^2 = 1$ )이며, 기계 1의 작업 속도는 2 ( $o^1 = 2$ ), 기계 2의 작업 속도는 1 ( $o^2 = 1$ )이라고 가정하자. 그리고 모든 부품함에 대하여 차지하는 슬롯의 수는 모두 1 ( $s_1 = \dots = s_4 = 1$ )이라고 하자. 또한 각 기계의 총 슬롯의 수는 각각 1과 2 ( $N^1 = 1; N^2 = 2$ )로 주어질 것이라고 가정하자. 그러면, 먼저, <그

림 1>의 유전인자  $X$  와 <표 1>의 필요 부품수를 이용하여, 모든  $i, k, d$  에 대한  $u_{ik}^d$  를 다음과 같이 계산할 수 있다.

$$\begin{aligned} u_{11}^2 &= q_1 n_{11} / o^2 = 3 \cdot 1 / 1 = 3 \\ u_{12}^1 &= q_1 n_{12} / o^1 = 3 \cdot 3 / 2 = 4.5 \\ u_{13}^2 &= q_1 n_{13} / o^2 = 3 \cdot 1 / 1 = 3 \\ u_{22}^2 &= q_2 n_{22} / o^2 = 3 \cdot 1 / 1 = 3 \\ u_{23}^1 &= q_2 n_{23} / o^1 = 3 \cdot 2 / 2 = 3 \\ u_{31}^2 &= q_3 n_{31} / o^2 = 3 \cdot 1 / 1 = 3 \\ u_{32}^1 &= q_3 n_{32} / o^1 = 3 \cdot 2 / 2 = 3 \\ u_{34}^2 &= q_3 n_{34} / o^2 = 3 \cdot 1 / 1 = 3 \end{aligned}$$

이외의 나머지 모든  $u_{ik}^d = 0$ 의 값을 갖는다. 또한,  $y_{ik}^d$  는 <그림 1>의  $X_A$ 에 의해서

$$y_{11}^2 = y_{12}^1 = y_{13}^2 = y_{22}^2 = y_{23}^1 = y_{31}^2 = y_{32}^1 = y_{34}^2 = 1$$

이며, 이 외의 나머지 모든  $y_{ik}^d$ 는 0의 값을 갖는다. 이들  $u_{ik}^d$ 의  $y_{ik}^d$ 의 값들과 식 (4)를 이용하여 각 PCB들의 워크로드 차이를 다음과 같이 계산할 수 있다.

$$\begin{aligned} w_1^1 &= u_{12}^1 = 4.5, w_1^2 = u_{11}^2 + u_{13}^2 = 6 \\ w_2^1 &= u_{23}^1 = 3, w_2^2 = u_{22}^2 = 3 \\ w_3^1 &= u_{32}^1 = 3, w_3^2 = u_{31}^2 + u_{34}^2 = 6 \end{aligned}$$

따라서  $W_1 = 1.5$ ,  $W_2 = 0$ ,  $W_3 = 3$ 이며, 목적 함수 (1)의 첫 번째 항인 두 기계간의 워크로드 차이는

$$\sum_{i \in P} W_i = W_1 + W_2 + W_3 = 4.5 \text{ 가 된다.}$$

이제 목적 함수 (1)의 두 번째 항을 살펴보자. <그림 1>의 개체에서 PCB 유형의 투입 순서는 2, 3, 1 이므로, 투입 순서 변수  $z_{ij}$  는  $z_{23} = z_{31} = 1$ 이며, 나머지는 모두 0의 값을 갖는다. 먼저, 연속 투입되는 두 PCB 유형 PCB 2와 3에 대해서,

$$h(2, 3, 1) = 1 + 1 - 0 = 2, h(2, 3, 2) = 1 + 2 - 0 = 3$$

이고, PCB 3과 1에 대해서는

$$h(3, 1, 1) = 1 + 1 - 1 = 1, h(3, 1, 2) = 2 + 2 - 1 = 3$$

이다. 따라서 목적 함수 (1)의 두 번째 항은

$$\begin{aligned} & \max [0, h(2, 3, 1) - N^1] + \max [0, h(2, 3, 2) - N^2] \\ & + \max [0, h(3, 1, 1) - N^1] + \max [0, h(3, 1, 2) - N^2] \\ & = (2 - 1) + (3 - 2) + (1 - 1) + (3 - 2) = 3 \end{aligned}$$

과 같이 계산된다. 결론적으로 <그림 1>의 예와 같은 개체의 적합도 값은  $4.5 + 3 = 7.5$ 가 된다.

지금까지 예를 통해 하나의 개체에 대한 적합도 값을 구하는 절차를 살펴보았다. 이를 일반화시켜, 개체  $X = (X_S, X_A)$ 가 주어졌을 때,  $X$ 의 적합도를 평가 하는 절차는 다음과 같다.

#### <적합도 평가 절차>

step 1 : 주어진 입력 값들을 이용하여 모든  $i, k, d$  에 대한  $u_{ik}^d$ 를 다음 식에 의해 계산한다.

$$u_{ik}^d = q_i n_{ik} / o^d$$

step 2 :  $X_A$ 로부터  $y_{ik}^d$ 의 값이 0인지 1인지를 결정하고, 식(5)와  $u_{ik}^d$  값을 이용하여, 모든  $i, d$ 에 대한  $w_i^d$

$$\text{를 } w_i^d = \sum_{k \in B_i} u_{ik}^d \text{로부터 계산한다.}$$

step 3 : 적합도 함수(목적 함수)의 첫 번째 항인  $\sum_{i \in P} W_i$

$$\text{를 } \sum_{i \in P} W_i = \sum_{i \in P} |w_i^1 - w_i^2| \text{에 의해 계산한다.}$$

step 4 :  $X_S$ 로부터,  $z_{ij}$ 의 값이 0인지 1인지를 결정하고, 모든  $i, j, d$ 에 대한  $h(i, j, d)$ 를 식 (14)를 이용하여 구한다.

step 5 : 목적 함수 (1)의 두 번째 항에 step 4에서 구한 값들을 대입하여 두 번째 항의 값을 구한다.

step 6 : step 3에서 구한 값과 step 5에서 구한 값을 더해 적합도 값을 구한다.

### 3.3 유전 연산(genetic operations)

유전 알고리즘은 교차 연산(crossover) 혹은 돌연변이 연산(mutation)을 통해 다음 세대를 구성할 후보 개체들을 생성해 낸다. 먼저 본 연구에서 적용한 교차 연산 절차를 살펴본다.

한 세대의 크기를 *popsize* 라고 하고, 임의의 개체  $X$ 에 대한 적합도 함수 값을  $F(X)$ 라고 하자. 교차 연산을 위해서, 먼저 현 세대로부터 교차 연산을 할 개체 한 쌍을 임의로 선택하되, 각각의 개체의 적합도 값에 비례하여 선택될 확률에 가중치를 부여한다. 선택된 두 개체에 대해서, 개체 각각에서 임의의 두 행을 선택한다. 그 다음에, 선택한 두 행을 서로 교차하여 새로운 한 쌍의 개체를 생성한다. 한 세대에서의 교차 연산은 미리 정해진 수만큼 행해진다. 교차 연산의 횟수는 *popsize* 와, 미리 결정된 교차 확률 *crossprob*을 곱하여 결정한다.

이상의 교차 연산 과정을 정리하면 다음과 같다.

<교차 연산 절차>

step 1 : 현 세대의 모든 유전인자  $X_k$ , ( $k = 1, 2, \dots, popsize$ )에 대해서, 교차연산에 선택될 확률  $p_k$  와 누적확률  $cp_k$ 를 다음 식으로부터 구한다.

$$p_k = \frac{\sum_{j=1}^{popsize} F(X_j) - F(X_k)}{\sum_{j=1}^{popsize} F(X_j) * (popsize - 1)}, \quad cp_k = \sum_{j=1}^k p_j$$

step 2 : 0과 1사이의 난수(random number)  $r_1$ 을 발생시켜  $cp_k \leq r_1 \leq cp_{k+1}$ 을 만족하는 유전인자  $X_k$ 를 선택한다. 같은 방식으로 난수  $r_2$ 에 대해서  $cp_l \leq r_2 \leq cp_{l+1}$ 을 만족하는  $X_l$ 을 선택한다.

step 3 :  $X_k$ 에서  $x_{p1}^1 = r_1$ 과  $x_{q1}^1 = r_2$ 를 각각 만족하는 두 행  $x_p^1$  과  $x_q^1$  를 선택한다. 같은 방식으로,  $X_l$ 에서  $x_{s1}^2 = r_2$  와  $x_{t1}^2 = r_1$ 를 각각 만족하는 두 행  $x_s^2$  와  $x_t^2$  를 선택한다.

step 4 :  $Y_1 = X_k$ ,  $Y_2 = X_l$ .  $Y_1$ 의  $p$ 번째 행과  $q$ 번째 행을 각각  $x_s^2$  와  $x_t^2$  로 바꾼다. 마찬가지로,  $Y_2$ 의  $s$ 번째 행과  $t$ 번째 행을  $x_p^1$  과  $x_q^1$  으로 대체한다. 이렇게 결정된  $Y_1$ 과  $Y_2$ 는 새로운 세대의 후보가 된다.

step 5 : 현재까지의 총 교차연산의 횟수가 미리 정해진 교차 연산의 횟수 ( $crossprob \times popsize$ ) 이면, 끝낸다. 아니면 step 2로 돌아간다.

<교차 연산 절차>에 의해서 만들어진 자식 개체  $Y_1$  과  $Y_2$ 는 부모 개체  $X_k$ 와  $X_l$ 이 가능해(feasible solution)이면 항상 가능해이다. 따라서 교차 연산에 의해 새로운 세대를 구성해 나가는 과정은 초기해가 가능해일 경우 항상 가능해를 유지하게 된다.

두 번째 유전연산인 돌연변이 연산의 목적은 기존의 개체 구조와는 다른 새로운 형태의 개체가 생성 되도록 함으로써 해가 지역 최적해(local optimal solution)에 머무르는 현상을 방지하는 것이다. 본 연구에서는 다음 절에서 기술하게 될 초기 유전인자의 생성방법과 동일한 방법으로 돌연변이 연산을 수행하도록 하였다.

### 3.4 초기해의 구성

유전 알고리즘에서 초기 세대의 구성은 임의의 해를 이용하는 방법을 많이 적용한다. 그러나 부품 할당을 임의로 할 경우, 각 기계의 슬롯 용량을 초과하는 실행 불가능한 해가 만들어질 수 있다. 따라서 실행 가능한 유

전인자를 생성하기 위해 다음과 같은 방식으로 각각의 초기 개체를 생성한다.

<초기해 구성 절차>

step 1 : PCB 유형의 투입 순서를 임의로 결정한다.

step 2 : 각 PCB 유형에 대해서, 다음 과정을 수행한다.

step 2.1 : 기계1의 남아 있는 용량( $rc^1$ )과 기계2의 남아 있는 용량( $rc^2$ )을 다음과 같이 초기화한다.

$$rc^1 = N^1, \quad rc^2 = N^2$$

step 2.2 : 아직 어느 기계에도 할당되지 않은 하나의 부품함  $k$ 를 선택한 후, 1 혹은 2의 값을 갖는 난수  $r$ 을 발생시킨다.

step 2.3 : (i)  $r = 1$ 인 경우: 만약,  $rc^1 \geq s_k$ 이면 부품함  $k$ 를 기계 1에 할당하며,  $rc^1 = rc^1 - s_k$  로 수정한다. 만약,  $rc^1 < s_k$ 이면 부품함  $k$ 를 기계 2에 할당한다.

(ii)  $r = 2$ 인 경우: 만약,  $rc^2 \geq s_k$ 이면 부품함  $k$ 를 기계 2에 할당하며,  $rc^2 = rc^2 - s_k$  로 수정한다. 만약,  $rc^2 < s_k$ 이면 부품함  $k$ 를 기계 1에 할당한다.

step 2.4 : 모든 부품함에 대한 할당이 완료되었으면 끝내고, 아니면 step 2.2로 돌아간다.

### 3.5 유전 알고리즘 수행 과정

지금까지 개체의 구성과 적합도 함수, 그리고 교차 연산과 돌연변이 연산을 수행하는 방법, 또한 실행 가능한 초기 개체의 구성 방법을 살펴보았다. 여기에서는 이상의 과정들을 종합하여 본 연구에서 제시하고자 하는 유전 알고리즘의 전체적인 수행절차를 정리한다.

step 1 : 각 세대의 모집단 크기  $popsize$ , 총 수행 횟수  $iterno$  를 정의한다.

step 2 : <초기해 구성 절차>에 의해  $popsize$  개의 초기 개체를 생성한다.

step 3 : 각 개체에 대해서 적합도 값  $F(X_k)$  ( $k = 1, 2, \dots, popsize$ )를 계산한다.

step 4 : <교차 연산 절차>에 의해 ( $crossprob \times popsize$ ) 개의 새로운 개체를 생성한다.

step 5 : 돌연변이 연산을 통해 새로운  $mutno$  개의 새로운 유전인자를 생성한다.

step 6 : 기존 개체와 교차 연산을 통해 생성된 개체, 그리고 돌연변이연산을 통해 생성된 개체의 적합도 값을 비교하여, 그들의 적합도 값에 비례하

게 다음 세대에 선택될 확률을 부여하여 *popsiz*e 개의 개체들로 새로운 세대를 구성한다.

*step 7* : 가장 좋은 적합도 값을 갖는 유전인자를 결정한다.

*step 8* : 이 과정이 *iterno*에 이르면 멈춘다. 아니면, *step 3*으로 돌아간다.

## 4. 성과 측정

본 연구에서 제안한 유전 알고리즘의 성능을 평가하기 위하여, 사용되는 PCB 유형의 수( $\alpha$ ), 부품함의 수( $\beta$ ), PCB 유형별 작업할 PCB의 수( $q_i$ , for all  $i \in P$ ), 각 PCB 유형별로 필요한 부품함의 수( $|B_i|$ , for all  $i \in P$ ), 각 PCB 유형에 필요한 각각의 부품의 수( $n_{ik}$ , for all  $k \in B_i$ ,  $i \in P$ )를 변경시켜가면서, 다양한 문제를 생성하여 성능을 측정하였다. 또한, 알고리즘의 성과가 어느 정도인지를 파악하기 위해 각 문제에 대한 하한값(lower bound)을 계산하여 어느 정도의 성과를 나타내는지 제시한다. 제안된 유전 알고리즘은 펜티엄-IV 1.6Ghz 개인용 컴퓨터에서 C언어로 구현하였다.

### 4.1 실험 문제 생성 및 파라미터의 설정

PCB 유형의 수( $\alpha$ )는 10, 15, 20개, 부품함의 수( $\beta$ )는 50, 100, 200으로 정하여 문제를 생성하였으며, 따라서 총 9가지 문제를 생성하였다. 모든 문제에 대하여, PCB 유형별 작업할 PCB 수량( $q_i$ )은 U[1,30] 분포로부터 임의로 생성하였으며, 각 PCB 유형별로 필요로 하는 부품함의 수( $|B_i|$ )는 총 부품함 집합의 10 ~ 50%가 되도록 임의로 생성하였다. 또한 각 PCB 유형에 필요한 특정 부품의 수( $n_{ik}$ )는 U[1,10]의 분포로부터 임의로 생성하였다. 작업 도중 부품함 교체를 위한 시간( $c^d$ )은 두 기계 모두 1분으로 설정하였으며, 각 기계의 속도는 시간당 5000개의 부품을 장착할 수 있는 것으로 정의하였다. 본 실험에서는 두 기계의 작업 속도에 관한 파라미터를 같은 것으로 설정하였다. 실제 기계의 작업 속도는 서로 다를 수도 있으나, 이 경우 각 기계에 대한 파라미터만 바뀌면 된다. 따라서 본 실험에서는 기계의 속도를 같은 것으로 간주하고 실험을 진행하였다.

필요한 파라미터는 모든 문제에 대하여 공통적으로 교차연산확률(*crossprob*) 0.5, 돌연변이 확률(*mutprob*) 0.02, 세대수(*iterno*) 1000, 각 세대의 모집단 크기(*popsiz*e) 200으로 설정하였다.

### 4.2 실험 결과

유전 알고리즘의 성능은 주어진 문제의 최적해와 유전 알고리즘으로 구한 해의 값을 비교하여 어느 정도 성과를 나타내는지 비교할 수 있다. <표 2>에 나타나 있는 9가지 예제 문제들은 모두 실제 산업 현장에서 적용할 수 있는 크기의 실용적인 문제들로서 최적해를 구하는 것은 사실상 불가능하다. 따라서 최적해와 유전 알고리즘으로 구한 해를 비교하여 성능을 평가하는 것은 어렵다. 다른 한 가지 방법은 기존의 연구들 중에서 본 연구에서 다루고 있는 문제와 동일한 문제를 다른 알고리즘을 통해 다른 문제의 결과와 비교하는 것이다. Gronalt & Zeller[5]의 연구가 본 연구의 문제와 유사하나, 1장에서 언급한 바와 같이 이들의 목적 함수는 본 연구의 측정치와는 차이가 있어 직접적인 비교는 불가능하다. 또한, <표 2>의 문제1과 같이 PCB 유형 10가지, 부품 종류 50가지인 경우조차도 최적해를 구할 수 없는 문제로서, 본 연구에서 제시한 유전 알고리즘의 성능을 최적해와 비교하는 것도 불가능하다. 따라서 본 연구에서는 최적해의 하한값 (lower bound)을 구하여 이 하한값과 본 연구에서 제안한 유전 알고리즘에 의한 해를 비교하여 본 유전 알고리즘이 이상적인 해(하한값)에 얼마나 근접하는지를 파악함으로써 그 성능을 측정하고자 한다. 하한값을 구하는 과정은 먼저 각 PCB에 대하여 장착되어야 할 모든 부품들의 수를 계산한다. 이 부품들을 각 기계의 슬롯 용량을 고려하지 않고 각 기계의 작업 속도를 고려하여 두 기계에 동일한 작업시간이 되도록 나누어 배치한다. 다음은 부품함 교체에 대한 하한치로서, 사용되는 총 부품함의 수와 두 기계의 슬롯의 수의 합과의 차이로부터 구할 수 있다 이상의 하한값 계산 과정을 정리하면 다음과 같다.

#### <하한값 계산 절차>

*step 1* : 이상적인 최소 작업시간의 계산  $t$ 를 모든 PCB의 생산에 필요한 모든 부품들의 수라고 하자. 또한,  $t_1$ 과  $t_2$ 를 각각 기계 1과 기계 2에서 조립되는 부품의 수라고 하자. 그러면, 정의에 의하여  $t = \sum_{i \in P} \sum_{k \in B_i} q_i n_{ik}$  이며,  $t_1 + t_2 = t$ 가 된다. 또한, 기계 1에서 소요되는 작업 시간은  $t_1/o^1$ 이며, 기계 2에서 소요되는 작업시간은  $t_2/o^2$ 이다. 이상적인 최소 작업 시간은 기계 1과 기계2에서 소요되는 작업시간이 같은 경우, 즉  $t_1/o^1 = t_2/o^2$ 일 때이므로,  $t_1 = (o^1/o^1 + o^2)t$ 이다. 따라서 이상적인 최소작업시간은  $t/(o^1 + o^2)$  (단위 : 시간)이다.

*step 2* : 이상적인 최소 부품함 교체시간의 계산 최소 부품함 교체 횟수  $f$ 는 사용되는 부품함의 수를 두

기계의 슬롯의 수의 합계로 나눈 값이므로  $f = \lfloor \text{사용되는 부품함의 수} / (N_1 + N_2) \rfloor$  에 의해 계산된다. 이 때, 교체 시간은  $f/60$  (단위: 시간) 이다.

step 3 : 하한값 계산 step 1과 step 2에서 구한 시간의 합  $t/(o^1 + o^2) + f/60$  이 하한값이 된다.

<표 2>에 PCB수와 부품 수를 변화시켜가면서 수행한 9가지 서로 다른 크기의 문제에 대한 실험 결과가 나타나 있다. 이 표에서 '용량'은 기계 1과 기계 2의 슬롯의 수를 나타낸다. 'GA'는 유전 알고리즘에 의한 해를 나타내는데, 총 10회 수행한 결과의 평균값을 보여준다. 'LB'는 각 문제에 대한 최적해의 하한 값이다. 여기에서 'GA'와 'LB'의 값은 부품의 조립시간과 부품함 교체 시간의 합을 나타내는 것으로 단위는 분이다. 'GAP'는 유전 알고리즘에 의한 해와 하한값과의 차이를 백분율로 나타낸 것이다. 9가지 문제 중 하한값과의 차이가 가장 적은 문제 3의 경우 0.9%를 나타내고 있다. 평균적으로 약 7.6%의 차이가 존재하는 것으로 나타나 상당히 근접한 해를 구할 수 있음을 보여준다. 또한, 계산 시간에 있어서 모든 문제에 대해서 1분 이내에 해를 도출하였다.

<표 2> 실험 결과: 유전 알고리즘의 성능

문제	PCB 유형	부품 종류	용량1)	GA2)	LB3)	GAP4) (%)
1	10	50	20/20	56.67	54.76	3.5
2	10	100	35/35	347.90	311.70	11.6
3	10	200	70/70	409.62	405.61	0.9
4	15	50	20/20	146.44	126.30	15.9
5	15	100	35/35	245.86	225.59	8.9
6	15	200	70/70	463.01	427.79	8.2
7	20	50	20/20	202.85	180.86	12.1
8	20	100	35/35	401.98	367.70	9.3
9	20	200	70/70	521.32	497.97	4.7
평균						7.6

- 1) 기계1/기계2의 부품 슬롯의 수
- 2) 제안된 유전 알고리즘에 의한 해
- 3) 최적해의 하한값
- 4) (GA-LB)/LB\*100

## 5. 결 론

본 연구는 두 대의 표면실장기로 이루어진 PCB 조립 공정에서 다양한 유형의 PCB를 생산함에 있어서 PCB의 투입 순서와 부품함 배치를 결정하는 문제를 다룬다. 이 문제를 수리적으로 모형화하여 제시하였으며, 이를 통해 이 문제가 합리적인 시간 안에 최적해를 구하기 어려운 NP-hard에 속하는 문제임을 보여주었다.

문제의 해결을 위해 최적해에 근사한 해를 찾기 위한 방법으로써 유전 알고리즘에 기반을 둔 해법을 제시하였다. 또한 제한된 해법의 성능 평가를 위해 최적해의 하한값을 구하여 제안된 알고리즘에 의한 해와 하한값을 비교하였다. 그 결과, 비교적 좋은 성능의 해를 짧은 시간 안에 도출할 수 있음을 제시하였다.

본 연구에서 다룬 문제는 PCB 생산 현장에서 실질적으로 발생할 수 있는 매우 실용적인 문제로써 본 연구의 결과는 실제 산업현장에도 적용이 가능한 해법이라고 보인다. 뿐만 아니라, 본 연구에서 제안한 알고리즘은 표면 실장기의 수가 3대 이상일 경우에도 유전 인자의 형태를 그대로 유지하면서, 쉽게 확대 적용할 수 있어, 문제 상황 변화에도 유연하게 적용할 수 있는 방안이다.

## 참고문헌

- [1] Carmon, T. F., Maimon, O. Z., and Dal-El, E. M.; "Group set-up for printed circuit board assembly," International Journal of Production Research, 27(10) : 1795-1810, 1989.
- [2] Crama, Y., Kolen, A. W. J., Oerlemans, A. G. and Spijksma, F. C. R; "Throughput rate optimization in the automatic assembly of printed circuit boards," Annals of Operations Research, 26:455-480, 1990.
- [3] Deo, S., Javadpour, R., and Knapp, G. M.; "Multiple setup PCB assembly planning using genetic algorithms," Computers & Industrial Engineering, 42(1) : 1-16, 2002.
- [4] Goldberg, D. E.; Genetic Algorithm in Search, Optimization, and Machine Learning, Addison-Wesley. 1989
- [5] Gronalt, M., and Zeller, R.; "Component allocation and job sequencing for two-series connected SMD placement machines," International Journal of Production Research, 38(2):409-427, 2000.
- [6] Ji, P., Sze, M. T., and Lee W. B.; "A genetic algorithm of determining cycle time for printed circuit board assembly lines," European Journal of Opera-



- tional Research, 128(2):175-184, 2001
- [7] Khoo, L. P., and Loh, K. M.; "A genetic algorithms enhanced planning system for surface mount PCB assembly," *International Journal of Advanced Manufacturing Technology*, 16(4):289-296, 2000
  - [8] Maimon, O.Z., & Braha, D. (1998). A genetic algorithm approach to scheduling PCBs on a single machine. *International Journal of Production Research*, 36(3), 761-784.
  - [9] Maimon, O. Z., Dal-El, E. M., and Carmon, T. F.; "Set-up saving schemes for printed circuit board assembly," *European Journal of Operational Research*, 70(2):177-190, 1993.
  - [10] Nemhauser, G. L., and Wolsey, L. A.; *Integer and Combinatorial Optimization*. New York, John Wiley & Sons, 1988
  - [11] Yu, S., Sohn, J., Park, S. and Oh, B. J.; "Efficient operation of a multi-functional surface mounting device," *Computers & Industrial Engineering*, 33(3-4) : 797-800, 1997.