

## 다중피더배치를 고려한 칩마운터의 조립순서 최적화

### PCB Assembly Optimization of Chip Mounters for Multiple Feeder Assignment

김 경 민, 박 태 형\*

(Kyung-Min Kim and Tae-Hyoung Park)

**Abstract :** We propose an optimization method to reduce the assembly time of chip mounters. Feeder arrangement and assembly sequence are determined considering the multiple feeder assignment. The problem is divided into two sub-problems: feeder arrangement problem and assembly sequence problem. We present mathematical model for each sub-problem. The clustering algorithm and assignment algorithm are applied to solve the feeder arrangement problem. The assignment algorithm and connection algorithm are applied to solve the assembly sequence problem. Simulation results are then presented to verify the usefulness of the proposed method.

**Keywords :** PCB assembly, chip mounter, multiple feeder assignment, optimization

#### I. 서론

SMT(surface mount technology) 인라인(in-line) 시스템은 표면 실장형 부품(SMD: surface mount device)을 인쇄회로기판(PCB: printed circuit board)에 조립하는 전자조립시스템이다. 각종 전자 제품의 소형 경박화 추세에 따라 전자부품들은 기존의 삽입형에서 표면실장형으로 대체되고 있다. 특히 PCB의 홀에 부품을 삽입하여 조립하는 삽입형 부품과 비교하여, PCB 표면 위에 부품을 올려놓아 조립하는 표면실장형 부품은 자동 조립이 매우 수월하여 현재 대부분의 전자제품 제조라인에서 그 점유비율이 크게 증가하고 있다. SMT 인라인 시스템은 스크린프린터(screen printer) 공정, 칩마운터(chip mounter) 공정, 리플로우어(reflower) 공정 등으로 분류된다. 이와 같은 SMT 인라인 시스템의 여러 공정 중 칩마운터 공정은 전체 PCB조립 시간과 생산성에 가장 큰 영향을 준다. PCB조립 공정 중 칩마운터 장비는 통상 1대~4대 정도가 운용되며 수십 개에서 수천 개에 이르는 많은 부품을 실장한다[1].

칩마운터는 기본적으로 피더에서 부품을 흡착하여 PCB보드 상의 지정된 위치에 부품을 장착하는 메커니즘을 갖으며, 부품을 흡착 및 장착하는 조립헤드, 조립헤드를 이동시키는 프레임, 부품을 공급하는 피더, PCB를 이동시키는 반송부 등으로 구성된다. 칩마운터는 기구적 구조에 따라 겐트리형, 듀얼 겐트리형 및 로터리형으로 구분될 수 있으며[1], 본 논문에서는 그림 1과 같은 조립헤드의 수가 하나인 단일헤드 겐트리형 칩마운터를 다루고자 한다. 칩마운터의 전, 후열에 설치된 피더열에는 여러 개의 피더라인이 있으며, 각 피더라인에는 부품 피더가 설치된다. 겐트리형 XY 프레임에는 하나의 조립헤드가 설치되어 있으며, 조립헤드는 pick-and-placement의 사이클 운동을 수행한다.

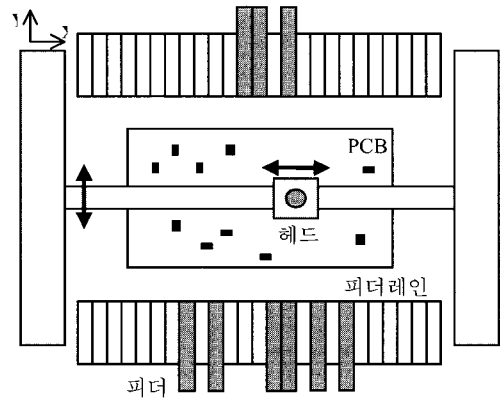


그림 1. 단일헤드 겐트리형 칩마운터.

Fig. 1. A gantry-type chip mounter with single head.

칩마운터의 최적화 문제는 조립시간을 최소화하는 피더배치 및 조립순서를 결정하는 문제이다. 이 문제를 동시에 통합하여 최적화 하는 것은 NP-complete의 최적화 문제로서, 최적해를 구하는 것이 계산적으로 매우 어렵다고 알려져 있다 [2]. 따라서 전체 문제를 몇 개의 문제로 분할하고 이를 계층적으로 구성하여 근사적 최적해를 구하는 발견적 기법(heuristic)에 의한 접근이 시도되고 있다[3-7]. 일반적으로 전체 문제를 피더배치 최적화 문제와 조립순서 최적화 문제 등으로 분리하며, 분리된 문제를 해결하기 위한 여러 가지 방법들이 제안되고 있다.

Kumar 등[2]은 단일헤드 겐트리형 칩마운터의 최적화 문제를 정수계획문제(integer programming)로 수식화하였으며, 피더배치 문제는 minimum-weight-maximum-matching 문제로, 조립순서 문제는 표준 TSP(traveling salesman problem) 문제로 각각 변환하였다. Ball 등[8]은 조립순서 문제를 RPP(rural postman problem) 문제로 모델링 하였으며, 근사적 최적해를 구하기 위한 발견적 알고리즘을 제안하였다. 박태형 등[9]은 조립순서 문제를 수송문제로 모델링하였으며, 헝가리안 알고리즘을 이용하여 해를 구하였다. 그러나 이러한 연구는 단일피더배치 경우만 고려하였다.

\* 책임저자(Corresponding Author)

논문접수 : 2004. 6. 17., 채택확정 : 2004. 9. 4.

김경민 : 충북대학교 제어계측공학과(semcocal@empal.com)

박태형 : 충북대학교 전기전자컴퓨터공학부

(taehpark@chungbuk.ac.kr)

※ 본 연구는 한국과학재단 목격기초연구(R05-2002-001164-0)의 지원에 의하여 수행되었습니다.

전체 조립시간을 단축하기 위해서 장착점이 많은 부품을 하나 이상의 피더에 배치하는 것이 유리하다. 이러한 다중피더배치는 장착 시간뿐만 아니라 피더 교환에 소모되는 시간도 단축할 수 있다[10][11]. DePuy 등[10]은 부품할당 문제를 정수계획문제로 모델링하였으며, 하나의 부품을 여러 개의 피더에 배치하는 것이 조립시간 단축에 유리하다는 실험적 결과를 보고하였다. Kazaz 등[11]은 다중피더배치 경우가 고려된 칩마운터의 최적화 문제를 정수계획문제로 모델링하였다. 피더배치 문제는 할당문제로, 조립순서 문제는 MDVRP (multi-depot vehicle routing problem) 문제로 변환하였다.

본 논문은 다중피더배치를 고려한 단일헤드 겐트리형 칩마운터의 최적화 문제를 위한 새로운 방법을 제시한다. 칩마운터의 최적화 문제를 정수계획문제로 모델링하며, 전향경로 최적화 문제와 후향경로 최적화 문제로 분리하여 구성한다. 각각의 분리된 문제는 최적성이 우수한 할당문제(assignment problem)로 변환하여 구성한다. 모의 칩마운터에 대한 비교 시뮬레이션을 수행하여 제안된 방법과 다중피더배치의 유용성을 검증한다.

II. 문제의 정의

그림 2는 단일헤드 겐트리형 칩마운터의 조립 사이클을 보여준다. PCB상에 부품이 장착되는 위치를 장착점(mount point), 피더로부터 부품을 흡착하는 위치를 흡착점(pickup point)이라 한다. 흡착점에서 장착점으로의 이동경로는 전향경로(forward path), 장착점으로부터 흡착점까지의 이동경로는 후향경로(backward path)라 한다. 하나의 조립헤드는 ‘부품 흡착 - 전향경로 이동 - 부품 장착 - 후향경로 이동’의 사이클을 반복적으로 수행하며 PCB상의 모든 장착점에 부품을 실장한다.

장착점 집합  $P$ , 부품타입 집합  $C$  및 피더라인 집합  $L$  을 다음과 같이 정의한다.

$$P = \{1, 2, \dots, n_p\} \tag{1}$$

$$C = \{1, 2, \dots, n_c\} \tag{2}$$

$$L = \{1, 2, \dots, n_l\} \tag{3}$$

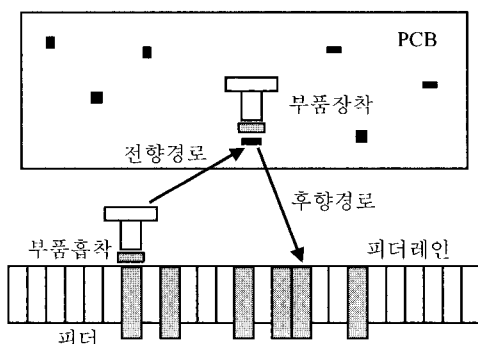


그림 2. 단일헤드 겐트리형 칩마운터의 조립 사이클.  
Fig. 2. Assembly sequence for gantry-type single head chip mounter.

단,  $n_c \leq n_p$  이다. 또한 장착점에 대한 부품타입을 지정하는 함수를  $\tau: P \rightarrow C$ , 피더라인에 배치된 피더의 부품타입을 지정하는 함수를  $\xi: L \rightarrow C$  이라 정의한다. 부품타입  $c \in C$  에 해당하는 장착점의 집합  $P_c$  는 다음과 같다.

$$P_c = \{p \in P | \tau(p) = c\} \tag{4}$$

각 부품타입  $c \in C$  의 피더배치수량을  $\mu_c$  라 정의한다. 모든 부품타입에 대하여  $\mu_c = 1$  인 피더배치를 단일피더배치라 하며,  $\mu_c \geq 1$  인 부품타입이 존재하는 피더배치를 다중피더배치라 한다. 본 논문은 다중피더배치를 고려하며, 이때 각 부품타입에 대한  $\mu_c$  는 미리 주어지는 값이라 가정한다.

한 사이클 당 소요되는 시간은 부품 흡착 시간, 전향경로 이동 시간, 부품 장착 시간 그리고 후향경로 이동 시간의 합으로 표현된다. 이 중 부품의 흡착 및장착에 소요되는 시간은 조립헤드의 Z축 이동 시간 및 대기 시간이므로 피더배치나 조립순서의 변경과 무관하게 일정한 값을 갖는다. 한 사이클 당 조립헤드의 XY 축 이동에 소요되는 시간  $t(i, l, j)$  는 다음과 같다.

$$t(i, l, j) = t(i, l) + t(l, j) \tag{5}$$

단,  $t(i, l)$  는 장착점  $i \in P$  에서 슬롯  $l \in L$  까지의 후향경로 이동시간이며,  $t(l, j)$  는 슬롯  $l \in L$  에서 장착점  $j \in P$  까지의 전향경로 이동시간이다.

피더배치변수  $x_{cl} \in \{0, 1\}$ , 전향경로변수  $y_{lp} \in \{0, 1\}$  및 후향경로변수  $z_{pl} \in \{0, 1\}$  을 다음과 같이 정의한다.

$$x_{cl} = \begin{cases} 1, & \text{피더라인 } l \in L \text{ 에 부품타입 } c \in C \text{ 의} \\ & \text{피더가 배치된 경우} \\ 0, & \text{나머지 경우} \end{cases} \tag{6}$$

$$y_{lp} = \begin{cases} 1, & \text{피더라인 } l \in L \text{ 에서 장착점 } p \in P \text{ 로의} \\ & \text{전향경로가 존재하는 경우} \\ 0, & \text{나머지 경우} \end{cases} \text{ 단 } \xi(l) = \tau(p) \tag{7}$$

$$z_{pl} = \begin{cases} 1, & \text{장착점 } p \in P \text{ 에서 피더라인 } l \in L \text{ 로의} \\ & \text{후향경로가 존재하는 경우} \\ 0, & \text{나머지 경우} \end{cases} \tag{8}$$

부품타입당 한 개의 피더가 배치되는 단일피더배치의 경우, 그 부품타입에 해당하는 장착점까지의 전향경로는 피더배치와 함께 자동적으로 결정된다. 피더에서 흡착한 부품은 그 부품타입에 해당하는 장착점으로 이동하여 장착되어야 하기 때문이다. 즉  $x_{cl} = 1$  인 경우,  $p \in P_c$  인 모든 장착점에 대하여  $y_{lp} = 1$  이다. 그러나 다중피더배치의 경우, 전향경로변수는 피더배치변수와 독립적으로 결정되어야 한다.

칩마운터의 최적화 문제는 피더배치변수, 전향경로변수 후향경로변수를 결정변수로 하는 정수계획문제로서 다음과 같

이 정의할 수 있다.

$$\min \sum_{c \in C} \sum_{l \in L} \sum_{i \in P} \sum_{j \in P} t(i, l, j) x_{cl} y_{lj} z_{il} \quad (9)$$

$$\text{s. t.} \quad \sum_{c \in C} x_{cl} = 1, \quad \forall l \in L \quad (10)$$

$$\sum_{l \in L} x_{cl} = \mu_c, \quad \forall c \in C \quad (11)$$

$$\sum_{l \in L} y_{lj} = 1, \quad \forall j \in P \quad (12)$$

$$\sum_{l \in L} z_{il} = 1, \quad \forall i \in P \quad (13)$$

$$\sum_{i \in P} z_{il} = \sum_{j \in P} y_{lj}, \quad \forall l \in L \quad (14)$$

목적함수 (9)는 조립헤드의 총 이동시간을 의미한다. (10)은 각 피더라인에는 하나의 부품 피더가 배치되어야 함을 의미하며, (11)은 각 부품타입 당  $\mu_c$  개의 피더를 배치한다는 것을 의미한다. (12)와 (13)은 각 장착점은 단 한번 방문되어야 하며 이때 장착점으로 이동하는 전향 경로와 장착점으로부터 피더로 이동하는 후향경로의 수가 모두 하나임을 의미한다. (14)는 각 피더라인으로 돌아오는 후향경로의 수는 그 피더라인에서 공급하는 부품의 수와 같음을 의미한다.

### III. 문제의 구성

복잡한 칩마운터 최적화 문제의 해를 구하기 위하여, 전체 문제를 전향경로 최적화 문제와 후향경로 최적화 문제로 분리하여 구성한다. 전향경로 최적화 문제(FP: forward-path optimization problem)는 총 전향경로 이동시간을 최소화시키는 피더배치 변수 및 전향경로 변수를 결정하는 문제이다. 후향경로 최적화 문제(BP: backward-path optimization problem)은 총 후향경로 이동시간을 최소화시키는 후향경로 변수를 결정하는 문제이다.

전향경로 최적화 문제 수학적 정의는 다음과 같다.

$$\text{(FP)} \quad \min \sum_{c \in C} \sum_{l \in L} \sum_{j \in P_c} t(l, j) x_{cl} y_{lj} \quad (15)$$

$$\text{s. t.} \quad \sum_{c \in C} x_{cl} \leq 1, \quad \forall l \in L \quad (16)$$

$$\sum_{l \in L} x_{cl} = \mu_c, \quad \forall c \in C \quad (17)$$

$$\sum_{l \in L} y_{lj} = 1, \quad \forall j \in P \quad (18)$$

(16)은 각 피더에는 최대 하나의 피더가 배치될 수 있음을 의미하며, (17)은 각 부품타입 당  $\mu_c$  개의 피더가 배치되어야 함을 의미한다. 또한 (18)은 각 장착점에 도달하는 전향경로는 하나임을 의미한다.

단일피더배치의 경우, 피더배치변수가 결정되면 전향경로 변수가 자동적으로 결정된다. 또한 피더배치변수를 결정하는 피더배치 문제는 각 부품타입 당 피더를 할당하는 문제로 변환될 수 있다. 즉, 단일피더배치 경우의 전향경로 최적화 문제 FPs는 다음과 같다.

$$\text{(FPs)} \quad \min \sum_{c \in C} \sum_{l \in L} \sum_{j \in P_c} t(l, j) x_{cl} \quad (19)$$

$$\text{s. t.} \quad \sum_{c \in C} x_{cl} \leq 1, \quad \forall l \in L \quad (20)$$

$$\sum_{l \in L} x_{cl} = 1, \quad \forall c \in C \quad (21)$$

위 문제는 전형적 선형계획문제인 할당문제로서 최적해를 구하는 방법이 잘 알려져 있다.

위 문제의 최적해인 피더배치변수  $x_{cl}^* \in \{0,1\}^{C \times L}$  로부터 전향경로변수의 최적해  $y_{lj}^* \in \{0,1\}^{L \times P}$  를 다음과 같이 구할 수 있다.

$$y_{lj}^* = 1 \Leftrightarrow x_{cl}^* = 1, \quad j \in P_c \quad (22)$$

다중피더배치의 경우는 피더배치변수가 결정되더라도 전향경로변수가 자동적으로 결정되지 않으므로 단일피더배치 경우와 같은 해법의 적용을 위하여 확장된 부품타입의 집합  $\bar{C}$  을 정의한다.

$$\bar{C} = \{1^1, \dots, 1^{\mu_1}, 2^1, \dots, 2^{\mu_2}, \dots, n_C^1, \dots, n_C^{\mu_{n_C}}\} \quad (23)$$

즉, (2)의 부품타입  $c \in C$  을 피더배치수량  $\mu_c$  개로 분할하여, 이를  $c^1, \dots, c^{\mu_c}$  로 표시한다. 이때 분할된 부품타입에 대하여 각 한 개의 피더배치가 허용되므로, 단일피더배치 경우의 해법을 적용시킬 수 있다. 또한 부품타입  $c \in C$  에 해당하는 장착점들을  $\mu_c$  개의 그룹으로 분할하고, 각 그룹을 부품타입  $c^1, \dots, c^{\mu_c}$  에 부여하여야 한다.

이 단계는 전향경로 최적화의 첫 번째 단계로서 클러스터링 방법을 사용하여 해결할 수 있다. 다음 단계는 단일피더배치 경우와 동일하게 할당문제의 해법을 적용하여 피더배치변수를 구하고, 이로부터 전향경로변수를 구하는 단계이다. 다중피더배치를 고려한 전향경로 최적화 문제는 다음의 이 단계로 구성된다.

$$\text{(FPm-1)} \quad \min \sum_{c \in C} \sum_{p \in P_c} \sum_{i=1}^{\mu_c} d(p, k_i) \quad (24)$$

$$\text{s.t.} \quad \bigcup_{i=1}^{\mu_c} P_{c^i} = P_c, \quad \forall c^i \in \bar{C} \quad (25)$$

$$\bigcap_{i=1}^{\mu_c} P_{c^i} = \emptyset, \quad \forall c^i \in \bar{C} \quad (26)$$

$$K = \{k_1, \dots, k_{\mu_c}\} \quad (27)$$

$$\text{(FPm-2)} \quad \min \sum_{\bar{c} \in \bar{C}} \sum_{l \in L} \sum_{j \in P_{\bar{c}}} t(l, j) x_{\bar{c}l} \quad (28)$$

$$\text{s.t.} \quad \sum_{\bar{c} \in \bar{C}} x_{\bar{c}l} \leq 1, \quad \forall l \in L \quad (29)$$

$$\sum_{l \in L} x_{\bar{c}l} = 1, \quad \forall \bar{c} \in \bar{C} \quad (30)$$

문제 (FPm-1)은 전형적 클러스터링 문제에 해당하며, (FPm-2)는 전형적 할당문제에 해당한다.

잘 알려진 해법을 사용하여 해를 구할 수 있으며, 구해진 피더배치변수  $x_{\bar{c}l}^* \in \{0,1\}^{\bar{C} \times L}$ 로부터, 전향경로변수  $y_{ij}^* \in \{0,1\}^{L \times P}$ 를 다음과 같이 구할 수 있다.

$$y_{ij}^* = 1 \Leftrightarrow x_{\bar{c}l}^* = 1, \quad j \in P_{\bar{c}} \quad (31)$$

후향경로 최적화 문제는 총 후향이동시간을 최소화시키는 문제이다. 후향경로는 부품 장착 후 부품 흡착을 위하여 피더로 복귀하는 헤드의 경로로서, 전향경로와 달리 배치된 피더의 부품타입을 고려하지 않고 임의로 결정할 수 있다. 그림 3과 같이, 후향경로 최적화 문제는 수송문제와 유사하게 모델링될 수 있다. 이 때 전향경로 및 후향경로를 포함한 전체 경로가 연결되어야 한다[8][9]. 다음은 후향경로 최적화 문제의 수학적 모델이다.

$$\text{(BP)} \quad \min \sum_{i \in P} \sum_{l \in L} t(i, l) z_{il} \quad (32)$$

$$\text{s.t.} \quad \sum_{l \in L} z_{il} = 1, \quad \forall i \in P \quad (33)$$

$$\sum_{i \in P} z_{il} = \lambda_l, \quad \forall l \in L \quad (34)$$

$$\sum_{i \in P} \sum_{m \in L} \sum_{l \in L} x_{\tau(i)m}^* y_{mi}^* z_{il} \leq |P| - 1 \quad (35)$$

위의 BP문제는 장착점 집합  $P$ 의  $n_p$  개 원소를 공급노드로 하고, 흡착점 집합  $L$ 의  $n_L$  개 원소를 수요노드로 하는 수송문제이다. (33)은 각 공급노드의 공급수량 제한 조건이며 각 장착점에서 피더로 이동하는 후향경로의 수가 하나임을

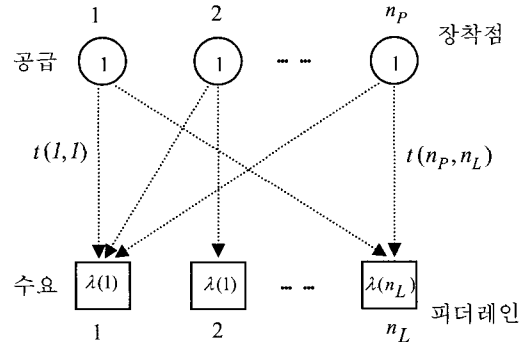


그림 3. 후향경로 문제의 수송문제 모형.  
Fig. 3. Transportation problem model of BP.

의미한다. (34)는 각 수요노드의 수요수량 제한조건이며 각 피더로 이동하는 후향경로의 수는 그 피더의 흡착도수  $\lambda_l$ 와 같음을 의미한다. 위의 문제는 총 공급수량과 총 수요수량이 동일한 균형수송문제(balanced transportation problem)이며, 각 수요노드를 수요수량만큼 확장시키면  $n_p$  개의 수요노드와  $n_p$  개의 공급노드를 갖는 할당문제로 변환된다.

#### IV. 최적화 알고리즘

앞장에서 변환된 문제의 해를 구하기 위한 알고리즘을 구성한다. 그림 4는 본 논문에서 제시하는 단일헤드 겐트리형 칩마운터의 최적화 알고리즘의 전체 흐름도이다. 전체 문제는 전향경로 최적화단계와 후향경로 최적화단계로 나누어진다. 전향경로 최적화단계에서는 피더배치와 전향경로를 결정한다.

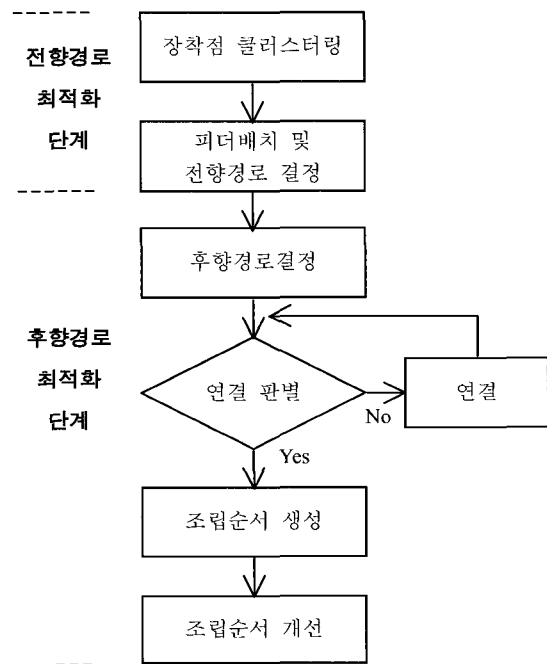


그림 4. 제안된 알고리즘의 전체 흐름도.  
Fig. 4. Flowchart of proposed algorithms.

후향경로 최적화단계에서는 후향경로를 결정하고 앞서 구한 전향경로와 후향경로로부터 전체 조립순서를 생성한다. 각 단계별 알고리즘은 다음과 같다.

1. 전향경로 최적화

Step 1 : FPM-1의 최적화 단계(클러스터링 알고리즘): 각 부품타입 당 장착점 집합  $P_i$ 을 클러스터링 알고리즘을 이용하여  $\mu_i$  개의 그룹으로 분할한다. 클러스터링 결과 확장된 부품  $\bar{c} \in \bar{C}$ 에 일대일로 대응하는 장착점 집합  $P_{\bar{c}}$ 이 생성되며, 다중피더배치 문제를 단일피더배치 경우와 동일하게 할당 알고리즘을 적용하여 해를 구할 수 있다.

Step 2 : FPM-2의 최적화 단계(할당 알고리즘): FPM-2문제의 해를 구한다. 이로부터 피더배치변수  $x_{ci}^*$ 와 전향경로변수  $y_{lp}^*$ 가 결정된다.

Step 1의 클러스터링을 하기 위해서 K-means 알고리즘을 적용한다. 클러스터링 알고리즘 중 비계층적 기법의 하나인 K-means 알고리즘은 구현이 쉽지만 K개의 시작점에 의존적이어서 국지적 최적해에 수렴할 수도 있다[12]. 효과적인 장착점 클러스터링을 위해 K개의 시작점은  $P_i$ 의 장착점들 중 서로 가장 거리가 떨어진 장착점으로 정한다. 또한 시작점은 각 클러스터의 대표값이 되며 알고리즘 수행 동안 변하지 않도록 한다. 다음은 장착점 클러스터링을 위한 알고리즘이며, 그림5는 클러스터링의 수행 단계를 보여준다.

S1. 그림 5 (a)와 같이 임의의 같은 부품타입  $\bar{c} \in \bar{C}$ 에 속한 장착점 집합  $P_i$ 을 선택한다.

S2. 그 부품에 할당된 피더 슬롯 수  $\mu_i$  개의 시작점(클러스터)을 구한다. 시작점은 그림 5(b)와 같이  $P_i$ 의 장착점들 중 서로 거리가 가장 떨어진 장착점을  $\mu_i$  개 만큼 결정한다.

S3. 그림 5(c)-(e)처럼  $P_i$ 에 속하는 각 장착점들을  $\mu_i$  개의 시작점 중 가장 가까운 클러스터로 할당한다.

S4. 클러스터링이 완료되면 클러스터 중심점을 다시 계산한다.

Step 2의 FPM-2할당문제에 헝가리안 알고리즘(Hungarian algorithm)[13]을 적용하여 해를 구한다. 헝가리안 알고리즘은 할당문제의 최적해를 구하는 대표적인 알고리즘으로서, 선형 계획법의 쌍대성(duality) 원리에 의해 최적성이 입증되어 있다. 헝가리안 알고리즘을 수행하면 피더배치변수  $x_{ci}^*$ 와 전향 경로변수  $y_{lp}^*$ 를 결정할 수 있다.

2. 후향경로 최적화 알고리즘

Step 3 : BP의 최적화 단계(할당 알고리즘): 문제의 해를 구하여 후향경로변수  $z_{pi}^*$ 를 결정한다.

Step 4 : 경로 연결 판별 단계(연결 판별 알고리즘): 앞 단계에서 구한  $y_{lp}^*$ 와  $z_{pi}^*$ 로부터 하나의 연결된 시퀀스(sequence)를 만들 수 있는지를 판별한다. 하나의 연결된 시퀀스를 얻을 수 있다면 Step 6로, 여러 개로 분할된 서브시퀀스가 생성된다면 Step 5로 간다.

Step 5 : 경로 연결 단계(연결 알고리즘): 여러 개로 분할된

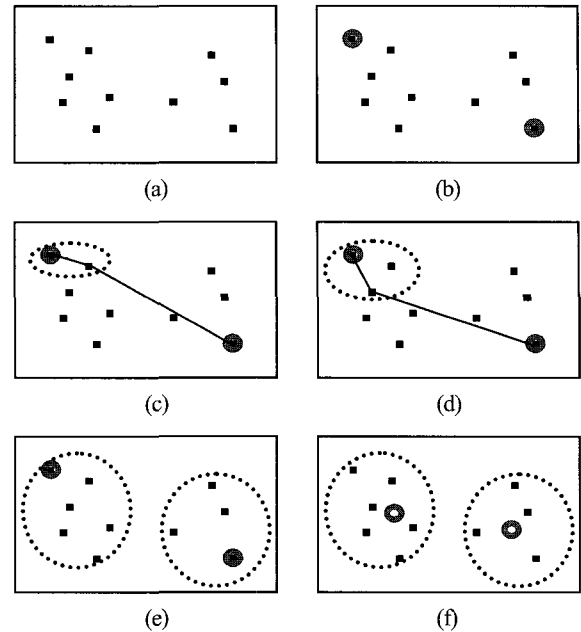


그림 5. (a)장착점 집합  $P_i$ , (b)  $\mu_i = 2$ 인 경우 2개의 시작점, (c)-(e)클러스터링 단계, (f)클러스터의 중심점.

Fig. 5. (a) The placement set  $P_i$ , (b) two start points for  $\mu_i = 2$ , (c)-(e) clustering procedure, (f) center points.

서브 시퀀스 중 임의의 두 서브시퀀스를 선택하여 후향경로를 각각 선택한다. 선택된 두 개의 후향경로를 변경해주어 두 서브시퀀스를 연결해준 후 Step 4로 간다.

Step 6 : 조립순서 생성 단계(오일러 투어 알고리즘): 모든 경로  $y_{lp}^*$ ,  $z_{pi}^*$ 를 한번씩 경유하는 조립헤드의 조립순서를 결정한다.

Step 7 : 조립순서 개선 단계(2-Opt 알고리즘): 2-opt를 사용하여 장착순서를 개선한다.

Step 3의 BP균형수송문제는 각 수요노드를 수요수량  $n_p$  개 만큼 확장시켜 할당문제로 변환한다. 변환된 문제는 결정변수  $z_{pi}$ 의 값이 0또는 1이며 공급수량 및 수요수량이 모두 1인 할당문제이므로 할당 알고리즘을 적용한다. 이 할당문제에 헝가리안 알고리즘을 적용하여 최적해를 구하며, 알고리즘의 해로부터 후향경로변수  $z_{pi}^*$ 을 구한다.

Step 4의 연결 판별 알고리즘[9]은 앞 단계에서 얻어진 전향경로와 후향경로를 합하여 구해진 시퀀스의 연결 여부를 판별하는 단계로서, 서로 인접한 전향경로와 후향경로를 저장시키는 과정을 반복한다. 모든 전향경로와 후향경로가 저장될 수 있으면 연결이라 판별하고 그렇지 않으면 비연결이라 판별한다. 그림 6(a)은 비연결 시퀀스의 예이며, 그림 6(b)은 연결 시퀀스의 예이다.

Step 5는 비연결 시퀀스를 연결하여 하나의 시퀀스로 구성하는 연결 알고리즘이다. 비연결 시퀀스는 몇 개의 연결된 서브시퀀스로 구성된다. 각 서브시퀀스 중 하나의 경로를 다른 서브시퀀스로 연결하여 서브시퀀스들을 연결시킬 수 있

다. 이때 전체 경로의 이동 거리가 증가하는 것을 억제하기 위해 가장 시간 차  $t(p_1, l_1) + t(p_2, l_2)$  가 적어지는 경로를 선택하여 연결해주는 발견적 기법을 적용한다[9].

그림 6(a)의 비연결 시퀀스에서 두 서브시퀀스의 후향경로  $z_{p_1l_1}$  와  $z_{p_2l_2}$  를 그림 6(b)처럼 바꾸어 줌으로써 두 개로 분리된 서브시퀀스는 하나로 연결된다.

Step 6는 비연결 연결조건을 만족하는 시퀀스로부터 조립헤드의 장착순서를 결정해 주는 단계이다. 조립헤드의 대기점에서 가장 가까운 피더로부터 시작하여 모든 경로를 한번씩 경유하는 경로를 찾는다. 이는 오일러 투어 알고리즘[14]을 적용하여 구할 수 있다. 그림 7의 번호는 생성된 오일러 투어의 이동 순서이며, 칩마운터 조립헤드의 조립순서를 의미한다.

Step 7는 오일러 투어 알고리즘으로부터 생성된 조립순서를 개선하기 위하여 국지적 탐색방법인 2-opt를 사용한다.

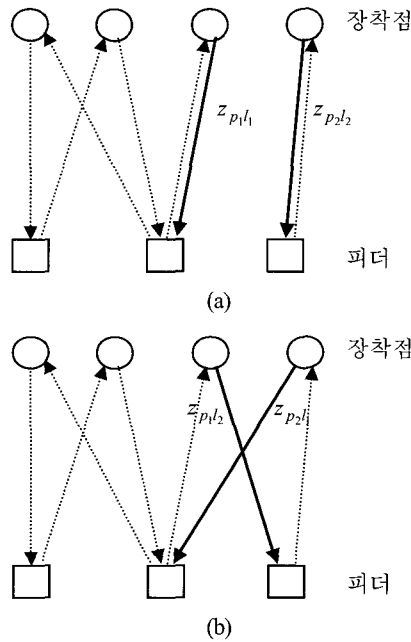


그림 6. (a) 비연결 시퀀스, (b) 연결 알고리즘에 의해 생성된 연결 시퀀스.

Fig. 6. (a) Disconnected sequence, (b) connected sequence by connection algorithm.

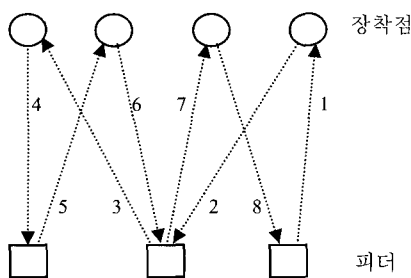


그림 7. 오일러 투어 및 장착순서.  
Fig. 7. Euler's tour and placement sequence.

본 논문에서 제안한 단일헤드 겐트리형 칩마운터 최적화 문제의 전체 계산복잡도는 할당 알고리즘의 계산 복잡도에 의존하며 헝가리안 알고리즘을 적용하여 해를 구한 경우  $O(n^3)$  이다. 그러나 계산적으로 효과적인 할당 알고리즘을 사용하는 것이 가능하므로 계산시간의 단축이 가능하다.

할당 알고리즘은 최적해를 보장하며 할당 알고리즘으로부터 얻어진 경로로 전체 연결된 시퀀스를 얻게 되는 경우 최적의 조립 순서를 얻을 수 있다. 비연결 시퀀스인 경우 이를 연결 시퀀스로 변환하기 위해 발견적 기법이 적용되어야 하므로 근사적 최적해를 구할 수 있다. 할당 알고리즘에 의하여 구성된 시퀀스가 여러 개의 서브시퀀스로 분리되는 경우, 제안된 알고리즘의 성능은 저하될 수 있다[9]. PCB상에 같은 부품타입 당 장착점들이 편중되어 분포되어 있고, 이에 따라 부품타입 당 장착점들의 가장 가까운 위치에 부품 피더가 배치된 경우 많은 연결 서브시퀀스로 이루어진 해를 얻게 된다. 그러나 대부분의 PCB에는 여러 부품타입이 장착되며, 각 부품타입 당 장착점들이 고르게 분포되어 있으므로 많은 연결 서브시퀀스가 발생할 가능성은 드물다. 또한 몇 개의 서브시퀀스가 발생하여 근사적 최적해를 구하게 되더라도 2-opt를 이용하여 해를 개선할 수 있다.

V. 시뮬레이션

새로이 제시한 알고리즘의 성능을 평가하고 그 유용성을 검증하기 위해, 모의 칩마운터를 대상으로 하여 시뮬레이션을 수행하였다. 칩마운터의 X축 및 Y축 겐트리는 모두 동일한 사다리꼴 속도 프로파일을 갖는다고 가정하였으며, Z축 속도는 0으로 가정하였다. 칩마운터의 전후면 피더열에는 각각 32개씩 모두 64개의 피더 슬롯이 있으며, 중복 배치되는 부품의 피더 수는 사용자가 임의로 입력할 수 있도록 설정하였다. 알고리즘은 C++언어로 프로그래밍 되었으며, Microsoft사의 Visual C++로 제작되었다. 제작된 프로그램은 IBM-PC호환기종 Pentium-IV 급 2.66GHz/MS-Windows XP환경에서 실행되었다. 제안된 알고리즘의 성능을 평가하기 위하여, 2가지의 비교대상 알고리즘을 구현하였다. 첫 번째 비교대상 알고리즘(method 1)은 전체 칩마운터의 문제를 피더배치 문제와 장착순서 문제로 나누었다. 피더배치는 탐욕적 알고리즘(greedy algorithm)[15]을 이용하여 구하였다. 장착순서 문제는 TSP문제로 변환하여 최근접탐색법(Nearest Neighbor)[16]으로 초기 장착순서를 생성하고 2-opt기법[16]에 의하여 장착순서를 개선시키는 방법을 사용하였다.

두 번째 비교대상 알고리즘(method 2)은 전체 문제를 피더배치 문제, 장착순서의 전향경로 문제 그리고 후향경로 문제의 3가지 부분문제로 분리하였다. 피더배치 문제는 할당 알고리즘을 적용하였으며, 장착순서의 전향경로는 탐욕적 알고리즘, 후향경로는 할당 알고리즘을 이용하여 구하였다.

표 1은 시뮬레이션에 사용된 보드들의 부품 및 장착점의 수를 보여준다.

표 2는 제안된 알고리즘과 두 가지의 비교대상 알고리즘들의 전체 장착 시간에 대한 결과이다. 이때 개선율은 다음과 같이 계산되었다.

표 1. 테스트 보드.

Table 1. Test PCBs.

테스트보드	부품타입 수	장착점 수
1	7	150
2	7	370
3	20	150
4	20	370
5	35	150
6	35	370

표 2. 총 장착시간 비교.

Table 2. Total cycle time for test PCBs.

테스트 보드	피 더 레 인 수	총 장착시간(sec)			개선율(%)	
		Method 1	Method 2	제안 방법	%1.	%2.
1	7(*)	176.78	176.30	176.30	0.27	0
	20	181.83	159.45	131.94	27.44	17.25
	28	180.12	153.06	128.44	28.69	16.09
2	7(*)	435.78	435.31	435.31	0.11	0
	27	432.41	369.68	311.76	27.90	15.67
	36	454.84	351.85	307.40	32.42	12.63
3	20(*)	164.54	163.64	163.64	0.55	0
	25	161.57	158.29	147.64	8.62	6.73
	34	160.69	152.71	136.36	15.14	10.71
4	20(*)	413.82	410.84	410.84	0.72	0
	28	402.94	381.56	353.61	12.24	7.33
	38	399.95	359.16	338.53	15.36	5.74
5	35(*)	160.50	158.18	158.18	1.45	0
	41	158.46	156.69	144.83	8.60	7.57
	47	157.26	151.87	141.14	10.25	7.07
6	35(*)	399.28	395.71	395.71	0.89	0
	46	392.69	380.67	365.88	6.83	3.89
	52	393.19	380.32	350.76	10.79	7.77

Notes: (\*)는 단일피더배치 경우 실험 결과를 의미한다.

개선율 % 1 (Improvement %1) =

$$\frac{\text{method 1의 장착시간} - \text{제안된 방법의 장착시간}}{\text{method 1의 장착시간}} \times 100(\%)$$

개선율 % 2 (Improvement %2) =

$$\frac{\text{method 2의 장착시간} - \text{제안된 방법의 장착시간}}{\text{method 2의 장착시간}} \times 100(\%)$$

표 2의 피더레인 수에 (\*)표기가 되어 있는 실험 결과는 단일피더배치의 경우 총 장착시간을 보여준다. 이 경우 제안된 방법은 method 1에 비해 최대 1.45%의 시간 단축 효과를 나타내며, method 2와 제안된 방법은 알고리즘의 차이가 없으므로 총 장착시간은 같다. (\*)표기가 되어 있는 경우를 제외한 나머지 실험 결과는 다중피더배치의 경우 총 장착시간을 나타낸다. 장착점이 많은 부품을 임의로 선택하여 다수 개의 피더에 배치를 하였다. 이 경우에 제안된 방법은 method 1에 비해 최대 35.42%의 시간 단축율을 보이며, method 2에 비해서는 최대 17.25%의 시간 단축율을 보인다. 그러므로 다중피더배치 경우, 장착점을 클러스터링하여 할당 알고리즘을 적용한 것이 장착시간 단축에 효과가 있음을 알 수 있다.

그림 8은 부품타입수가 7개이고 장착점이 150개인 테스트 보드에 대한 시뮬레이션 결과이다. 이 결과는 다중피더배치를 고려한 경우의 장착시간을 각 방법 별로 보여준다. 그래프의 가로축은 사용된 피더 수량이며, 세로축은 장착시간을 나타낸다. 제안된 방법과 method 2처럼 할당알고리즘을 이용하여 피더배치를 한 경우 중복피더의 수에 따라 장착시간이 감소함을 알 수 있다. 반면에 탐욕적 알고리즘으로 피더배치 문제를 결정한 method 1의 피더배치 방법은 중복된 피더를 효과적으로 배치하지 못하였기 때문에 중복피더 수가 증가하더라도 장착시간이 단축되지 않았다.

결과적으로 할당알고리즘이 단일헤드 칩마운터의 최적화 문제에 유용하게 적용될 수 있음을 확인할 수 있었으며, 다중피더배치를 고려한 결과 단일피더배치에 비해 전체 장착시간 단축에 효과가 있음을 검증하였다.

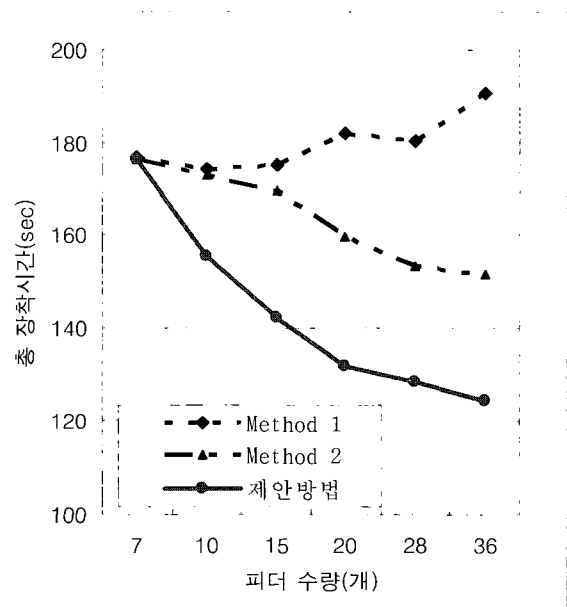


그림 8. 다중피더배치에 따른 총 장착시간 비교.

(대상 PCB: 부품타입수=7, 장착점수=150).

Fig. 8. Total cycle time comparison according to the multiple feeder assignment.

(Test PCB: No. of component types = 7, No. of components = 150).

**VI. 결론**

본 논문은 PCB조립장비인 단일헤드 겐트리형 칩마운터를 대상으로 총 장착시간의 단축을 위한 피더배치와 장착순서의 최적화 알고리즘을 제시하였다. 특히 특정 부품을 다중피더배치하는 방법이 장착시간 단축에 효과가 있음을 검증하였다.

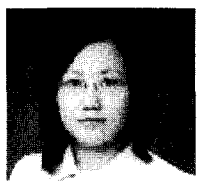
실제 생산 라인의 칩마운터에서는 장착시간을 단축하기 위해 특정 부품타입을 하나 이상의 피더라인에 배치하는 경우가 많이 있다. 그러나 기존에 이루어진 대부분의 연구가 단일피더배치의 경우에 국한되어 왔다. 본 논문에서는 다중피더배치를 고려한 단일헤드 겐트리형 칩마운터의 최적화 문제를 수학적으로 모델링 하였다. 합리적 시간 내에 최적해를 구하기 위해 전향경로 최적화 문제와 후향경로 최적화 문제로 분리하였으며, 각각을 최적성이 우수한 할당문제로 모델링 하였다. 다중피더배치 문제를 고려하기 위하여 클러스터링 알고리즘을 이용하였으며, 할당 알고리즘의 해로부터 완전한 장착 순서를 구하지 못하는 경우를 위해 발전적 기법의 연결 알고리즘을 적용하였다. 모의 시뮬레이션 결과로부터 할당 알고리즘이 칩마운터의 전체 장착시간 단축에 효과가 있음을 보여주었다.

또한 다중피더배치의 경우 장착점 클러스터링은 같은 피더로부터 부품을 공급받는 장착점들간의 거리가 줄어들도록 장착점을 분리하였으며, 결과적으로 조립헤드의 이동 거리를 줄일 수 있었다. 본 논문에서 제안된 알고리즘은 부품타입 별 배치 피더의 수량을 주어진 값으로 가정하였다. 현장에서 사용자가 피더 수량을 작업 조건에 맞게 임의로 지정하는 경우도 있으나, 조립시간의 단축 측면에서는 최적 수량으로 결정되는 것이 바람직하다. 추후 부품타입 별 피더 수량을 최적화시키는 문제에 관한 연구를 진행할 예정이다.

**참고문헌**

[1] 박태형, “전자조립용 CAM 시스템의 기술동향.” 전자공학회지, 제26권 제3호, pp. 48-61, 1999.  
 [2] R. Kumar and H. Li, “Integer programming approach to printed circuit board assembly time optimization”, *IEEE Trans. on Components, Packaging and Manufacturing Technology, Part-B: Advanced Packaging*, vol. 18, no. 4, pp. 720-727, 1995.  
 [3] S. X. Hu and M. W. Carter, “A component reel arrangement problem”, *Proceedings of the 36th Midwest Symposium on Circuits and Systems*, vol. 1, pp. 190-193, August, 1993.

[4] K. Altmkemer, B. Kazaz, M. Koksalan and H. Moskowitz, “Optimization of printed circuit board manufacturing: integrated modeling and algorithms”, *European journal of Operational Research*, vol. 124, pp. 409-421, 2000.  
 [5] I. Or and E. Duman, “Optimization issues in automated production of printed circuit boards: operations sequencing, feeder configurations and line balancing problems”, *Proc. 1996 IEEE Conf. on Emerging Technology and Factory Automation*, pp. 227-232, 1996.  
 [6] R. Kumar and Z. Luo, “Optimizing the operation sequence of a chip placement machine using TSP model”, *IEEE Trans. on Electronics Packaging Manufacturing*, vol. 26, no. 1, January, 2003.  
 [7] W. Wang, P. C. Nelson and T. M. Tirpak, “Optimization of high-speed multistation SMT placement machines using evolutionary algorithms”, *IEEE, Trans. On Electronics Packaging Manufacturing*, vol. 22, no. 2, April, 1999.  
 [8] M. O. Ball and M. J. Magazine, “Sequencing of insertions in printed circuit board assembly”, *Operation Research*, vol. 36, no. 2, pp. 192-201, 1988.  
 [9] 박태형, 김철환, “수송알고리즘에 의한 칩마운터의 조립 순서계획”, 제어 자동화 시스템 공학 논문지, 제6권 제9호, pp. 836-843, 2000.  
 [10] G. W. DePuy, J. C. Ammons, and L. F. McGinnis, “Multiple assignment of component types to feeder slots on automated printed circuit card placement machines”, *IEEE Trans. on Electronics Packaging Manufacturing*, vol. 23, no. 3, pp. 157-164, July, 2000.  
 [11] B. Kazaz and K. Altmkemer, “Optimization of multi-feeder (depot) printed circuit board manufacturing with error guarantees”, *European journal of Operational Research*, vol. 150, Issue 2, pp., 370-394, October, 2003.  
 [12] A. K. Jain, M. N. Murty and P. J. Flynn, “Data clustering: a review”, *ACM computing surveys*, vol. 31, no. 3, September 1999.  
 [13] A. Sultan, *Linear Programming an Introduction with Application*, Academic Press, 1933.  
 [14] H. A. Elsel, M. Gendreau and G. Laporte, “Arc routing problems, Part 1: the chinese postman problem”, *Operations Research*, vol. 43, no.2, pp. 231-242, March-April, 1995.  
 [15] Z. Michalewicz and D. B. Fogel, *How to solve it: Modern heuristics*, Springer, 2000.  
 [16] G. Reinelt, *The Traveling salesman: Computational solutions for TSP Applications*, Springer-Verlag, 1994.



**김 경 민**

1979년 9월 26일생. 2003년 충북대학교 전기전자공학부 학사. 2003년~현재 동대학원 제어계측공학과 석사 과정 재학중. 관심분야는 전자조립 시스템, 최적화 알고리즘.



**박 태 형**

1963년 8월 29일생. 1988년 서울대학교 제어계측공학과 학사. 1990년 동대학원 석사 및 1994년 동대학원 박사. 1994년~1997년 삼성테크윈 정밀기기연구소 선임연구원. 2000년~2001년 Univ. of Toronto 방문교수. 1997년~현재 충북대학교 전기전자컴퓨터공학부 교수. 관심분야는 반도체 및 전자 조립 시스템, 로봇 동작계획 및 최적화 알고리즘.