

수송알고리즘에 의한 칩마운터의 조립순서계획

An Assembly Sequence Planning of a Chip Mounter Using Transportation Algorithm

박태형, 김철한

(Tae-Hyung Park and Cheol-Han Kim)

Abstract : A sequence planning method is proposed to reduce the assembly time of gantry-type chip mounters with single head. The overall path of the chip mounter is divided into forward and backward path, and formulate the optimization problem is formulated as an transportation problem and an Euler's tour problem. The transportation algorithm is applied to find optimal backward path, and Euler's tour algorithm is used to generate an assembly sequence. Simulation results are presented to verify the usefulness of the proposed method.

Keywords : chip mounter, assembly sequence, optimization, transportation algorithm

I. 서론

칩마운터는 표면실장형의 전자부품을 인쇄회로기판(PCB: Printed Circuit Board)에 조립하는 전용 로봇이다. PCB를 조립하는 SMT(Surface Mount Technology) 라인은 스크린프린터, 칩마운터, 리플로우어 등으로 구성되며, 칩마운터의 조립 생산성은 전체 라인의 생산성에 가장 크게 영향을 준다[1]. 칩마운터는 기본적으로 부품을 공급하는 피더에서 부품을 흡착하여 PCB에 부품을 장착하는 메커니즘을 갖고 있으며, 기구적 구조에 따라 겐트리형, 듀얼 겐트리형 및 로터리형으로 구분될 수 있다[2].

본 논문에서 다루고자 하는 칩마운터는 그림 1 과 같이 조립헤드의 수가 하나인 겐트리형 칩마운터이다. 조립헤드는 피더 옆에 부착된 부품공급용 피더에서 부품을 흡착하고, PCB의 장착점에 부품을 장착하는 왕복운동에 의하여 PCB 조립을 완료한다. X축 및 Y축 겐트리의 독립적인 구동에 의하여 조립헤드의 X-Y 운동이 이루어지며, 조립헤드의 Z축 구동에 의하여 부품의 흡착 및 장착이 이루어진다. 또한 조립헤드의 회전축(R축)구동에 의하여 흡착 및 장착 각도를 조절한다. 조립헤드의 Z축 이동시간의 총합은 흡착 및 장착 순서에 관계없이 일정하다고 가정한다. 또한 조립헤드의 회전축 구동시간은 X축 및 Y축 구동시간에 비하여 충분히 빠르고, X-Y축 구동과 동시에 구동되므로 전체 조립시간에 영향을 주지 않는다고 가정한다.

피더배치 및 조립순서에 따라 칩마운터의 PCB 조립시간은 달라지며, 효과적인 피더배치와 조립순서를 결정하기 위한 많은 연구가 진행되고 있다 칩마운터 제조업체들은 피더배치 및 조립순서를 자동으로 생성하는 소프트웨어를 상용화하여 사용자에게 제공하고 있으며[3,4], 이 소프트웨어는 칩마운터의 생산성 및 사용 편의성 향상을 위한 매우 중요한 제품으로 인식되고 있다.

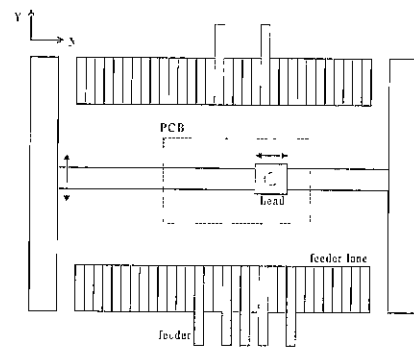


그림 1. 단일헤드 겐트리형 칩마운터 (평면도).

Fig. 1. Gantry-type chip mounter with single head (top-view).

피더배치 및 조립순서를 동시에 통합하여 최적화 하는 것은 NP-hard의 최적화 문제로서, 최적해를 구하는 것이 계산적으로 매우 어렵다고 알려져 있다.[5][6] 따라서 전체 문제를 몇 개의 문제로 분할하고 이를 계층적으로 구성하여 근사적 최적해를 구하는 발견적 기법(heuristic)에 의한 접근이 시도되고 있다. 특히 기구적 구조에 따라 서로 다른 발견적 기법이 적용되며 최적화 알고리즘의 구조 또한 상이하다.[6]-[9] 겐트리형 칩마운터에 대하여 피더배치 알고리즘과 조립순서 알고리즘을 계층적으로 구성하여 전체 조립시간의 단축을 도모하는 연구가 진행되었다.[5]-[9][12][13] 피더배치 최적화 문제는 각 부품별 피더의 최적 배치 열을 결정하는 문제이며, 조립순서 문제는 배치된 피더와 PCB의 장착점을 경유하는 최적의 경로를 찾아내는 문제이다. 조립헤드의 수가 하나인 경우, 조립순서 최적화 문제는 피더배치 최적화 문제와 비교하여 상대적으로 복잡하며 전체 조립시간의 결정에 큰 비중을 차지하고 있다. 본 논문은 단일헤드 겐트리형 칩마운터의 조립순서 최적화 문제를 다루고자 한다.

단일헤드 겐트리형 칩마운터의 조립순서 최적화 문제는, 피더의 흡착점과 PCB의 장착점에 대한 조립헤드의 방문순서를 결정하는 문제로서 네트워크 프로그래밍 기법의

접수일자 : 2000. 1. 5., 수정완료 : 2000 5 6.

박태형 : 충북대학교 전기전자 공학부

김철한 : 충북대학교 제어계측공학과 대학원

※ 본 논문은 한국 과학재단 핵심 전문연구(981-0928-154-1) 지원으로 수행되었습니다.

적용이 가능하다. 네트워크의 구성 및 경로 생성을 에 대한 여러 가지 방법이 제안되었으며, 흡착점과 장착점을 모든 포함한 네트워크를 구성하는 방식과 장착점만에 의하여 네트워크를 구성하는 방식으로 구분될 수 있다. Ball 등 [10]은 조립순서 최적화 문제를 흡착점과 장착점을 모든 포함한 네트워크에 대한 RPP(Rural Postman Problem)로 구성하였다. 즉 흡착점과 장착점으로 구성된 노드를 한번 이상 방문하고 다시 최초의 노드로 돌아오는 문제를 구성하였으며, minimum spanning tree 알고리즘, 오일러 투어 알고리즘 및 발견적 기법을 적용하여 근사적 최적해를 구하는 방법을 제시하였다. 그러나 Kumar 등 [5], Andreou [11], McGinnis 등 [12] 및 Leu 등 [13]은 조립순서 최적화 문제를 장착점만을 포함한 네트워크에 대한 TSP (Traveling Salesman Problem)로 구성하였다. 즉 장착점으로부터 구성된 노드를 한번 씩 방문하고 최초의 노드로 돌아오는 경로 생성의 문제로 구성하였다. 이 경우 장착점에서 다음 장착점까지의 이동시간은, 다음 장착점에 부품을 공급하는 흡착점으로 이동하는 시간과 그 흡착점에서 다음 장착점으로 이동시간의 합으로 계산되었다. TSP로 구성된 문제에 대하여 근사적 최적해를 제공할 수 있는 여러 가지 TSP 알고리즘이 적용되었으며 [5][11][12], 유전자 알고리즘도 적용되었다 [13].

본 논문은 단일헤드 쥘트리형 칩마운터의 조립순서 최적화를 위한 새로운 방법을 제시한다. 흡착점과 장착점을 모두 포함한 네트워크에 대한 최적화 문제를 구성하며, 이를 수송네트워크(transportation network)의 최적화 문제와 오일러 투어 문제로 변환하여 구성한다. 수송알고리즘에 의하여 수송네트워크를 최적화 시키고, 이로부터 장착점을 한번 씩 방문하는 오일러 투어를 생성시킨다. 수송네트워크에 의하여 오일러 투어가 직접 생성되는 경우 최적의 조립순서가 생성된다. 그러나 수송네트워크에서 오일러 투어가 직접 생성되지 않는 경우, 수송네트워크를 변경시켜 근사적 최적해를 구하기 위한 발견적 기법의 알고리즘을 제시한다. 모의 PCB 조립 시스템에 대한 비교 시뮬레이션에 의하여 제안된 방법의 유용성을 검증한다.

II. 문제의 정의

단일헤드 쥘트리형 칩마운터의 조립순서계획을 위하여, 각 부품의 피더는 피더 열에 이미 배치되어 있다고 가정한다. 또한 각 피더는 서로 다른 종류의 부품을 공급한다고 가정한다. 그림 2는 칩마운터 조립헤드의 조립경로를 보여준다. PCB 상에 부품이 장착되어야 하는 위치를 장착점 (mount point)이라 하고, 피더로부터 부품을 흡착하는 위치를 흡착점(pickup point)이라 한다. 또한 조립헤드의 최초 위치를 대기점(wait point)이라 한다. PCB가 컨베이어에 의하여 작업 스테이션으로 이송되면, 대기점에 정지하고 있던 조립헤드는 조립작업을 시작한다. 조립헤드는 흡착점으로 이동하여 부품을 흡착하고, 이를 장착점으로 이송하여 장착한다. 흡착점과 장착점 사이의 왕복운동에 의하여 모든 장착점에 대한 부품 장착이 완료되면, 조립 헤드는 다시 최초의 대기점으로 이동하여 정지한다.

칩마운터의 조립경로는 노드(N)와 아크(A)로 구성된 방향 네트워크 $G=(N, A)$ 에 의하여 표현될 수 있다. 노드는 다음과 같이 대기점 집합 W, 흡착점 집합 P, 장착점 집합 M로 구성된다.

$$N = W \cup P \cup M \tag{1}$$

$$W = \{0\} \tag{2}$$

$$P = \{1, 2, \dots, l\} \tag{3}$$

$$M = \{1, 2, \dots, n\} \tag{4}$$

단, $l < n$ 이다. 각 흡착점은 서로 다른 종류의 부품을 공급하며, 장착점에는 지정된 종류의 부품이 장착되어야 한다. 즉 각 장착점에 대하여 부품을 공급하는 흡착점이 지정되어 있다. 부품함수 $\xi : M \rightarrow P$ 를 각 장착점에 대하여 흡착점을 지정하는 함수로 정의한다. 흡착점 $p \in P$ 에서 부품이 공급되는 장착점의 집합 M_p 는 다음과 같이 정의되며,

$$M_p = \{m \in M \mid \xi(m) = p\} \tag{5}$$

M_p 는 다음의 조건을 만족하여야 한다.

$$M = \bigcup_{p=1}^l M_p \tag{6}$$

$$M_p \cap M_q = \emptyset, \quad \forall p, q \in P, p \neq q \tag{7}$$

흡착점 p에 대한 집합 M_p 의 원소의 수를 흡착도수 (pickup degree) λ_p 라 정의한다. 이때 흡착도수의 총합은 장착점의 수와 동일하다. 즉

$$\sum_{p=1}^l \lambda_p = n \tag{8}$$

네트워크의 아크는 다음과 같이 대기점부터 흡착점까지의 아크 집합 $A_{W \rightarrow P}$, 흡착점부터 장착점까지의 아크 집합 $A_{P \rightarrow M}$, 장착점부터 흡착점까지의 아크 집합 $A_{M \rightarrow P}$, 장착점부터 대기점까지의 아크 집합 $A_{M \rightarrow W}$ 에 의하여 구성된다.

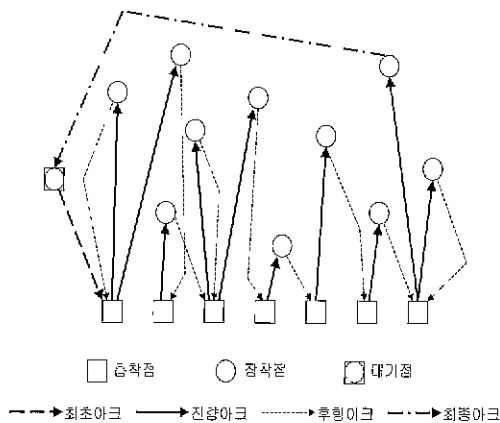


그림 2. 칩마운터의 조립순서 문제
Fig. 2. Assembly sequence problem for a chip mounter.

$$A = A_{W \rightarrow P} \cup A_{P \rightarrow M} \cup A_{M \rightarrow P} \cup A_{M \rightarrow W} \quad (9)$$

$$A_{W \rightarrow P} = \{ \langle w, p \rangle \mid w \in W, p \in P \} \quad (10)$$

$$A_{P \rightarrow M} = \{ \langle p, m \rangle \mid p \in P, m \in M_p \} \quad (11)$$

$$A_{M \rightarrow P} = \{ \langle m, p \rangle \mid m \in M, p \in P \} \quad (12)$$

$$A_{M \rightarrow W} = \{ \langle m, w \rangle \mid m \in M, w \in W \} \quad (13)$$

단, $\langle w, p \rangle \in A_{W \rightarrow P}$ 는 초기아크, $\langle p, m \rangle \in A_{P \rightarrow M}$ 은 전향아크, $\langle m, p \rangle \in A_{M \rightarrow P}$ 는 후향아크, $\langle m, w \rangle \in A_{M \rightarrow W}$ 는 최종아크를 의미한다.

노드와 노드사이를 이동하는데 소요되는 시간은 각 노드의 위치좌표로부터 계산될 수 있다. 노드 $i, j \in N$ 의 위치좌표가 각각 $(x_i, y_i), (x_j, y_j)$ 일 때, 조립헤드가 노드 i 에서 노드 j 까지 이동하는데 소요되는 시간 t_{ij} 는 다음과 같다.

$$t_{ij} = \max \{ \tau_x(|x_i - x_j|), \tau_y(|y_i - y_j|) \} \quad (14)$$

단, τ_x 및 τ_y 는 각각 X축 및 Y축의 이동거리로부터 이동시간을 계산하는 함수로서, 각 축의 속도 프로파일에 의하여 구할 수 있다. 각 축이 독립적으로 구동하여 조립헤드를 이동시키므로, 조립헤드의 이동시간은 각 축 이동시간의 최대 값이다. 칩마운터의 조립시간에는 조립헤드의 Z축 이동시간, 흡착 및 장착 지연시간, 부품 정렬시간 등이 포함되어야 한다. 그러나 이들 시간은 조립순서의 변경과 무관하게 일정한 값을 갖는다고 가정하여 최적화문제에 포함시키지 않는다

칩마운터의 조립순서 최적화문제는 대기점, 흡착점, 장착점으로 구성된 노드에 대하여, 최적의 초기아크, 전향아크, 후향아크 및 최종아크를 생성하는 문제이다. 생성된 네트워크에 모든 장착점을 한번씩 방문하고 대기점으로 돌아오는 방향 경로가 존재하여야 하며, 이 경로의 이동시간이 최소화되어야 한다. 0과 1의 값을 갖는 이진변수 $x_{wp}, x_{pm}, x_{mp}, x_{mw}$ 를 각각 초기아크 $\langle w, p \rangle$, 전향아크 $\langle p, m \rangle$, 후향아크 $\langle m, p \rangle$ 및 최종아크 $\langle m, w \rangle$ 의 생성 여부를 결정하는 변수로 정의하면, 칩마운터의 조립순서 최적화문제는 다음과 같은 최적화 문제로 표현될 수 있다.

$$\min \sum_{p=1}^l t_{wp} x_{wp} + \sum_{p=1}^l \sum_{m=1}^n t_{pm} x_{pm} + \sum_{m=1}^n \sum_{p=1}^l t_{mp} x_{mp} + \sum_{m=1}^n t_{mw} x_{mw} \quad (15)$$

s.t.

$$\sum_{p=1}^l x_{wp} = 1 \quad (16)$$

$$\sum_{m=1}^n x_{mw} = 1 \quad (17)$$

$$\sum_{m=1}^n x_{pm} = \lambda_p, \quad \forall p \in P \quad (18)$$

$$x_{wp} + \sum_{m=1}^n x_{mp} = \lambda_p, \quad \forall p \in P \quad (19)$$

$$x_{mw} + \sum_{p=1}^l x_{mp} = 1, \quad \forall m \in M \quad (20)$$

$$\sum_{p=1}^l x_{pm} = 1, \quad \forall m \in M \quad (21)$$

$$x_{wp}, x_{pm}, x_{mp}, x_{mw} \in \{0, 1\} \quad (22)$$

(15)의 목적함수는 전체 아크의 이동에 소요되는 시간을 의미한다. 첫 번째 항은 초기아크 이동시간, 두 번째 항은 전향아크 총 이동시간, 세 번째 항은 후향아크 총 이동시간, 네 번째 항은 최종아크 이동시간이다. (16),(17)은 대기점에 대한 출력아크 및 입력아크의 수가 모두 1임을 의미한다. (18),(19)는 각 흡착점에 대한 출력아크 및 입력아크의 수가 모두 흡착도수와 같음을 의미한다. (20),(21)은 각 장착점에 대한 출력아크 및 입력아크의 수가 모두 1임을 의미한다.

위의 최적화 문제에 대한 해 $x_{wp}, x_{pm}, x_{mp}, x_{mw}$ 를 구하고, 이 값이 1 인 경우에 대하여 아크 $\langle w, p \rangle, \langle p, m \rangle, \langle m, p \rangle, \langle m, w \rangle$ 를 구성하여 방향 네트워크 (N, \overline{A}) 를 구성할 수 있다. 즉

$$\overline{A} = \{ \langle i, j \rangle \in A \mid x_{ij} = 1 \} \quad (23)$$

이 때 네트워크 (N, \overline{A}) 의 노드의 수 $|N|$ 는 대기점 1 개, 장착점 n 개 및 흡착점 l 개의 합이다. 또한 아크의 수는 최초아크 1 개, 최종아크 1 개, 전향아크 n 개 및 후향아크 $n-1$ 개의 합이다. 그러므로 다음의 식이 성립된다.

$$|N| = n + l + 1 \quad (24)$$

$$|\overline{A}| = 2n + 1 \quad (25)$$

조립경로는 대기점 노드에서 출발하여 모든 장착점 노드를 한번씩 방문하고 또한 흡착점 노드는 흡착도수 만큼 방문하고 다시 대기점 노드로 돌아오는 경로이다. 즉 조립경로는 네트워크 (N, \overline{A}) 에 존재하는 모든 아크를 한번씩 방문하고 돌아오는 오일러 투어(Euler's tour)[14]를 의미한다. 일반적으로 네트워크에 오일러 투어가 존재하기 위해서는 다음의 두 조건을 만족시켜야 한다.[14][15]

조건 1. 모든 노드에 대한 입력아크 수와 출력아크 수의 합은 짝수이다 (균형조건)

조건 2. 임의의 두 노드사이에 경로가 존재한다. (연결조건)

네트워크 (N, \overline{A}) 는 (16)-(22)에 의하여 균형조건을 만족시킨다. 그러나 연결조건을 보장할 수 없다. 따라서 조립순서의 최적화문제는 (16)-(22)의 균형조건을 만족하고 (15)의 목적함수를 최소화하는 연결 네트워크 (N, \overline{A}) 를 생성하는 문제이다. 이 문제는 조합최적화 문제의 하나인 RPP(Rural Postman Problem)의 범주에 속한다. 그러나 RPP는 NP-hard의 문제로서 최적해를 구하는 알고리즘의 구현이 힘들다고 알려져 있다[10].

III. 문제의 변환

II장에서 정의된 칩마운터의 조립순서 최적화 문제를 간단한 형태로 변환시킨다. (15)는 조립경로의 총 이동시

간이며, 두 번째 항은 전향아크 총 이동시간이다. (11)과 같이 전향아크 $\langle p, m \rangle$ 는 흡착점 p 부터 $m \in M_p$ 를 만족하는 장착점 m 까지의 아크이다. 그러나 흡착점 p 로부터 부품이 공급되어야 하는 장착점의 집합 M_p ((5))는 미리 결정되어 있으며, (6),(7)의 조건에 의하여 유일하게 결정된다. 그러므로 전향아크 총 이동시간은 조립순서의 변경과 무관하게 일정한 값을 갖는다.

$$\begin{aligned} \sum_{p=1}^l \sum_{m=1}^n t_{pm} x_{pm} &= \sum_{m \in M_1} t_{1m} x_{1m} + \sum_{m \in M_2} t_{2m} x_{2m} \\ &+ \dots + \sum_{m \in M_l} t_{lm} x_{lm} \quad (26) \\ &= \text{constant} \end{aligned}$$

따라서 (15)-(22)의 최적화 문제에서 결정변수 x_{pm} 을 제거하면, 다음과 같이 단순화된 형태의 최적화 문제로 변환된다.

$$\min \sum_{p=1}^l t_{wp} x_{wp} + \sum_{m=1}^n \sum_{p=1}^l t_{mp} x_{mp} + \sum_{m=1}^n t_{mw} x_{mw} \quad (27)$$

s.t.

$$\sum_{p=1}^l x_{wp} = 1 \quad (28)$$

$$\sum_{m=1}^n x_{mw} = 1 \quad (29)$$

$$x_{wp} + \sum_{m=1}^n x_{mp} = \lambda_p, \quad \forall p \in P \quad (30)$$

$$x_{mw} + \sum_{p=1}^l x_{mp} = 1, \quad \forall m \in M \quad (31)$$

$$x_{wp}, x_{mp}, x_{mw} \in \{0,1\} \quad (32)$$

그림 3은 단순화된 조립순서 최적화문제의 예를 묘사한다. 그림 2와 비교하여 전체 아크 중 전향아크를 제외하고 최적의 최초아크, 후향아크 및 최종아크 만을 결정하는 문제이다.

단순화된 최적화 문제를 수송 문제(transportation problem)[16]로 변환하기 위하여 방향 네트워크 (N, A) 의 노드 집합 N 을 다음과 같이 다시 정의한다.

$$N = P' \cup M' \quad (33)$$

$$P' = \{0, 1, 2, \dots, l\} \quad (34)$$

$$M' = \{0, 1, 2, \dots, n\} \quad (35)$$

단, 확장된 흡착점 집합 P' 는 대기점과 흡착점을 원소로 한다. 또한 확장된 장착점 집합 M' 는 대기점과 장착점을 원소로 한다. 각 집합의 원소 0 는 대기점을 의미한다. 또한 아크 집합 A 를 다음과 같이 다시 정의한다.

$$A = A_{P \rightarrow M} \cup A_{M \rightarrow P'} \quad (36)$$

$$A_{M \rightarrow P'} = \{ \langle m', p' \rangle \mid m' \in M', p' \in P' \} \quad (37)$$

단, $A_{M \rightarrow P'}$ 는 확장된 장착점의 집합에서 확장된 장착점 집합으로의 아크 집합으로서, 최초아크, 후향아크 및 최

종아크를 포함한다. 최초아크, 후향아크 및 최종아크를 결정하는 이진변수 x_{wp}, x_{mp}, x_{mw} 를 다음과 같이 $x_{m'p'}$ 로 대체한다.

$$x_{m'p'} = \begin{cases} x_{wp} & (m'=0, p'=1, \dots, l) \\ x_{mp} & (m'=1, \dots, n, p'=1, \dots, l) \\ x_{mw} & (m'=1, \dots, n, p'=0) \end{cases} \quad (38)$$

위의 (38)에 의하여 (27)-(32)의 최적화 문제는 다음과 같이 변환된다

$$\min \sum_{m'=0}^n \sum_{p'=0}^l t_{m'p'} x_{m'p'} \quad (39)$$

s.t.

$$\sum_{m'=0}^n x_{m'p'} = \lambda_{p'}, \quad \forall p' \in P' \quad (40)$$

$$\sum_{p'=0}^l x_{m'p'} = 1, \quad \forall m' \in M' \quad (41)$$

$$x_{m'p'} \in \begin{cases} \{0\} & (\text{if } m'=p'=0) \\ \{0,1\} & (\text{otherwise}) \end{cases} \quad (42)$$

위의 문제는 확장된 장착점 집합 M' 의 $n+1$ 개 원소를 공급노드로 하고, 확장된 흡착점 집합 P' 의 $l+1$ 개 원소를 수요노드로 하는 수송문제이다. (39)는 총 수송비용으로서 전체 이동시간에서 전향아크 이동시간을 제외시킨 시간이다. (40)은 각 수요노드의 수요수량 제한조건으로서, 각 흡착점의 입력아크 수는 흡착도수와 같음을 의미한다 이때 추가된 대기점($p'=0$)을 고려하기 위하여 수요수량을 다음과 같이 설정한다.

$$\lambda_{p'} = \begin{cases} 1 & (p'=0) \\ \lambda_p & (p'=1, \dots, l) \end{cases} \quad (43)$$

(41)은 각 공급노드의 공급수량 제한조건으로서, 각 장착점 및 대기점의 출력아크 수는 모두 1 임을 의미한다. $m'=p'=0$ 인 경우 $x_{m'p'}$ 는 대기점에서 대기점으로의 아크를 의미하므로, 이의 발생을 금지시키기 위하여 (42)의 조건이 부과된다.

그림 4 는 (39)-(42)의 수송문제에 대한 모형이다. 공급노드 0 및 수요노드 0 는 모두 대기점을 의미한다. 따라서 공급노드 0 에 대한 출력아크는 최초아크를 의미하며, 수요노드 0 에 대한 입력아크는 최종아크를 의미한다. 또한 다른 아크들은 모두 후향아크를 의미한다. 공급노드의 총 공급수량은 (41)에 의하여 $n+1$ 이다. 수요노드의 총 수요수량은 (40)에 의하여 $\sum_{p'=0}^l \lambda_{p'} = \lambda_0 + \sum_{p'=1}^l \lambda_{p'}$ 이며, (8) 및 (43)에 의하여 $n+1$ 이다. 따라서 변환된 문제는 총 공급수량과 총 수요수량이 동일한 균형수송문제(balanced transportation problem)이다.

위의 수송문제의 최적해 $x_{m'p'}$ 를 구하여 확장된 장착점의 집합으로부터 확장된 흡착점 집합까지의 최적 아크 집합을 다음과 같이 설정할 수 있으며,

$$\overline{A}_{M \rightarrow P'} = \{ \langle m', p' \rangle \in A_{M \rightarrow P'} \mid x_{m'p'} = 1 \} \quad (44)$$

전향아크의 집합 $A_{P \rightarrow M}$ 을 추가하여 네트워크 (N, \overline{A}) 를 구성할 수 있다. 즉,

$$\overline{A} = A_{P \rightarrow M} \cup \overline{A}_{M \rightarrow P} \quad (45)$$

이 때 II장에서 논한 바와 같이 네트워크 (N, \overline{A}) 는 연결되어야 한다.

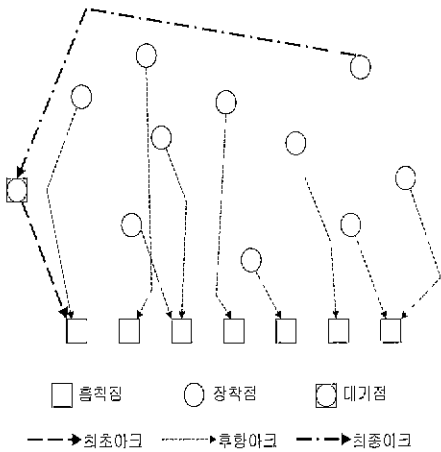


그림 3. 단순화된 칩마운터의 조립순서 문제.
Fig. 3. Simplified assembly sequence problem for a chip mounter.

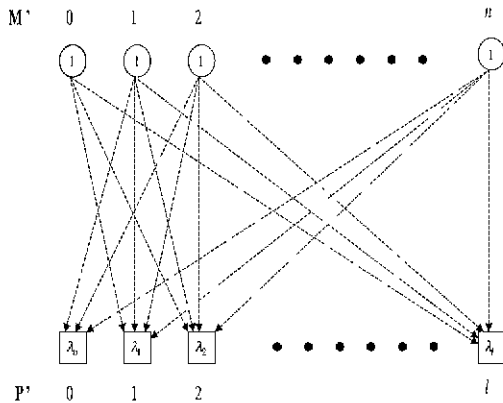


그림 4. 수송문제의 최적화 모형.
Fig. 4. Optimization model of the transportation problem.

IV. 조립순서계획 알고리즘

III장에서 변환된 문제의 해를 구하기 위한 알고리즘을 구성한다. 우선 (39)-(42)의 수송문제에 대한 최적해를 구하여 최적의 수송네트워크를 생성한다. 아크집합 $\overline{A}_{M \rightarrow P}$ ((44)) 에 의하여 구성되는 수송네트워크는 최초아크, 후향아크 및 최종아크를 포함한다. 수송네트워크에 미리 지정된 전향아크 $A_{P \rightarrow M}$ 을 추가하여 전체 네트워크 (N, \overline{A}) 를 구성한다. 전체 네트워크에 대한 연결 여부

를 판별하고, 연결 네트워크가 아닌 경우 아크집합 $\overline{A}_{M \rightarrow P}$ 를 수정하여 연결 네트워크로 만든다. 최종적으로 연결 네트워크에서 모든 아크를 한번씩 경유하는 오일러 투어를 생성한다. 오일러 투어에 의하여 대기점에서 출발하여 모든 장착점과 흡착점을 경유하는 전체 경로를 생성시키며, 이로부터 조립 순서를 얻는다.

다음은 본 논문에서 제안하는 칩마운터 조립순서계획 알고리즘의 전체 흐름도 이다

Step 1 : (수송알고리즘) (39)-(42)의 수송문제의 해를 구하고, 이로부터 아크집합 $\overline{A}_{M \rightarrow P}$ ((44))를 생성한다.

Step 2 : (연결 판별 알고리즘) 전체 네트워크 (N, \overline{A}) ((45)) 의 연결 여부를 판별한다. 연결된 경우 Step 4로 가며. 연결이 안된 경우 Step 3으로 간다.

Step 3 : (연결알고리즘) 전체 네트워크 (N, \overline{A}) 를 연결 서브네트워크로 분할시키고, 임의의 두 연결 서브네트워크 $(N_\alpha, \overline{A}_\alpha)$, $(N_\beta, \overline{A}_\beta)$ 를 선택한다. 선택된 각 서브네트워크의 아크 $\langle m_\alpha, p_\alpha \rangle \in \overline{A}_\alpha$, $\langle m_\beta, p_\beta \rangle \in \overline{A}_\beta$ 중 다음 값을 최소화시키는 아크 쌍을 선택하고,

$$\Delta t = t_{m_\alpha, p_\alpha} + t_{m_\beta, p_\beta} \quad (46)$$

다음과 같이 아크 쌍을 변경하여 $\overline{A}_{M \rightarrow P}$ 를 수정하고 Step 2로 간다.

$$\{\langle m_\alpha, p_\alpha \rangle, \langle m_\beta, p_\beta \rangle\} \Rightarrow \{\langle m_\alpha, p_\beta \rangle, \langle m_\beta, p_\alpha \rangle\} \quad (47)$$

Step 4 : (오일러 투어 알고리즘) 대기점에서 출발하여 모든 아크를 한번 씩 경유하고 다시 대기점으로 돌아오는 경로를 발견한다.

Step 1 의 수송문제는 잘 알려진 수송알고리즘에 의하여 최적해를 구할 수 있다. 그러나 변환된 문제의 경우 결정 변수의 값이 0 또는 1로서, 할당문제(assignment problem)로 변환하여 할당알고리즘을 적용하는 것이 바람직하다[16]. 할당문제는 공급수량 및 수요수량이 모두 1 인 수송문제로서, 균형수송문제는 간단히 할당문제로 변환될 수 있다. (39)-(42)의 균형수송문제에 대하여 각 수요노드를 수요수량 만큼 확장시키면, $n+1$ 개의 수요노드와 $n+1$ 개의 공급노드를 갖는 할당문제로 변환된다. 헝가리안 알고리즘(Hungarian algorithm)[14]-[16] 은 수송문제 및 할당문제의 최적해를 구하는 대표적인 알고리즘으로서, 선형계획법의 쌍대성(duality) 원리에 의하여 그 최적성이 입증되어 있다. 헝가리안 알고리즘은 구현이 용이하고 $O(n^3)$ 의 계산복잡도를 갖는다. 본 논문은 헝가리안 알고리즘을 적용하여 Step 1 의 최적해를 구한다. 한편 계산복잡도가 $O(n^2 \log l + l^2 \log^2 l)$ 인 efficient minimum-cost flow algorithm[17]의 적용 또한 가능하다.

그림 5 및 그림 6(a) 는 Step 1 의 수송문제의 최적해를 구하고 아크 집합 $\overline{A}_{M \rightarrow P}$ 을 생성하여 구성된 네트워크 (N, \overline{A}) 의 예를 보여준다. 각 노드의 입력아크 수와 출력 아크 수는 동일하여, 네트워크의 균형조건이 성립된다. 그림 5 는 연결 네트워크이며 그림 6(a) 는 비연결 네

트트워크이다.

Step 2 는 네트워크의 연결여부를 판별하는 알고리즘으로서, 노드 0 (대기점)에서 시작하여 각 노드에 이웃한 모든 노드들을 저장시키는 과정을 반복한다. 네트워크에 존재하는 모든 노드가 저장될 수 있으면 연결이라 판별하고, 그렇지 않으면 비연결이라 판별한다. 이 알고리즘의 계산 복잡도는 네트워크의 아크 수 (25)에 비례하며 따라서 $O(n)$ 이다.

Step 3 는 아크집합 $\overline{A_{M \rightarrow P}}$ 를 변경하여 비연결 네트워크를 연결네트워크로 구성하는 알고리즘이다. 비연결 네트워크는 몇 개의 연결 서브네트워크로 분할될 수 있다.

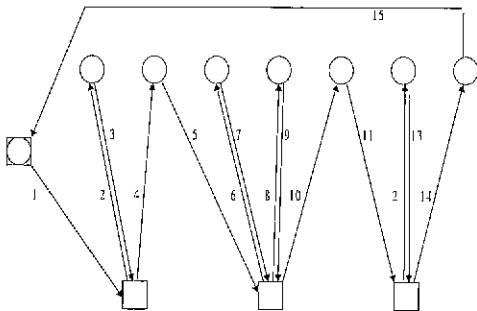
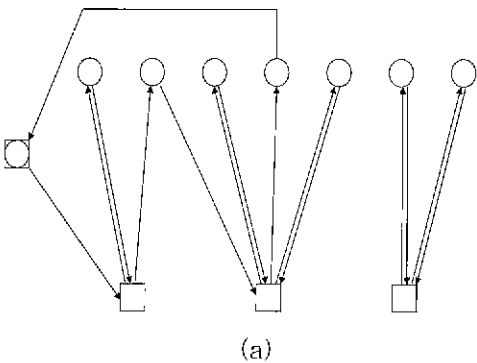
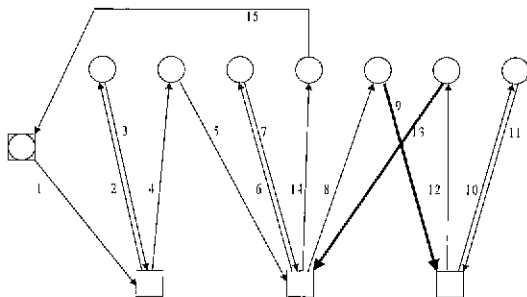


그림 5 연결 네트워크 및 오일러 투어.
Fig. 5. Connected network and Euler's tour.



(a)



(b)

그림 6. (a) 비연결 네트워크, (b) 연결알고리즘에 의하여 생성된 연결 네트워크 및 오일러 투어.
Fig. 6. (a) Disconnected network, (b) Connected network and Euler's tour by connection algorithm.

또한 각 서브네트워크 중 하나의 아크를 다른 서브네트워

크로 연결하여 서브네트워크를 연결시킬 수 있다. 이때 전체 아크 이동시간의 증가를 억제하기 위하여 (46),(47)의 발견적 기법을 적용한다. 그림 6(b) 는 연결알고리즘에 의하여 비연결 네트워크를 연결 네트워크로 변경시키는 예를 보여준다. 서브네트워크의 수를 s 라 하면 연결알고리즘의 계산 복잡도는 $O(\frac{n^2}{s})^2$ 이다.

Step 4 는 균형조건과 연결조건을 모두 만족시키는 네트워크에서 오일러 투어를 찾아내는 과정이다. 대기점에서 시작하여 모든 아크를 1회씩 경유하고 다시 대기점으로 돌아오는 경로를 찾는 것은 칩마운터의 조립순서를 결정하는 것을 의미한다. 잘 알려진 오일러 투어 알고리즘[14]의 적용이 가능하며 계산복잡도는 네트워크의 아크 수에 의존하여 $O(n)$ 이다. 그림 5 및 그림 6(b) 의 번호는 생성된 오일러 투어의 아크 이동 순서이며, 칩마운터 조립순서의 조립순서를 의미한다.

본 논문에서 제안한 조립순서계획 알고리즘의 전체 계산복잡도는 수송알고리즘(Step 1)의 계산복잡도에 의존하며 헝가리안 알고리즘을 적용하는 경우 $O(n^3)$ 이다. 수송알고리즘은 최적해를 보장하며, 따라서 Step 1 의 결과로 구성된 네트워크가 연결 네트워크인 경우 최적의 조립순서계획이 가능하다. 그러나 비연결 네트워크의 경우 Step 3 의 발견적 알고리즘이 적용되어야 하므로 근사적 최적해를 구할 수 있다.

수송알고리즘에 의하여 구성된 네트워크가 비연결 네트워크이고 또한 많은 수의 연결 서브네트워크로 분리되는 경우, 제안된 알고리즘의 성능은 저하될 수 있다. 이러한 네트워크는 어떤 장착점에서 그 장착점에 부품을 공급하는 흡착점으로 향하는 후향아크가 많이 발생하는 경우 주로 발생될 수 있다. PCB 상에 장착점이 부품 종류별로 편중되어 분포되어 있고, 이에 따라 부품 종류별 장착점들의 최 근접 위치에 부품 피더를 배치한 경우가 그 대표적 예이다. 그러나 대부분의 PCB에는 장착점들이 부품 종류별로 고르게 분포되며, 많은 수의 연결 서브네트워크를 갖는 비연결 네트워크의 발생 가능성은 높지 않다. 그러므로 본 논문에서 제시한 방법은 칩마운터의 조립순서계획문제에 효과적으로 적용될 수 있다.

V. 시뮬레이션 결과

모의 PCB 조립 시스템에 제안된 칩마운터의 조립순서 계획 알고리즘을 적용하여 그 유용성을 검증하였다. 칩마운터의 X축 및 Y축 겐트리는 모두 동일한 사다리꼴의 속도 프로파일을 갖는다고 가정하였으며, 최고 이동속도는 100(mm/sec), 가감속 시간은 0.2(sec)로 설정하였다. 칩마운터의 피더열 사이의 간격은 16(mm), 피더 열과 PCB 원점 사이의 거리는 300(mm)으로 설정하였다. PCB의 크기를 300(mm)×300(mm)으로 설정하였으며, 임의로 난수를 발생하여 장착점의 위치를 설정하였다 또한 가장 많은 장착점을 갖는 부품의 피더를 PCB의 최 근접 피더 열에 배치하는 방법으로 부품 종류별 흡착점의 위치를 설정하였다. 시뮬레이션은 IBM PC 호환기종(OS: Windows98)에서 MATLAB[18]을 사용하여 수행되었다.

제안된 방법의 성능을 분석하기 위하여 TSP 기법을 적용한 비교대상 알고리즘을 구현하였다. 즉 장착점의 방문 순서를 TSP 알고리즘에 의하여 우선 결정하고 이로부터 후향아크를 포함한 네트워크를 구성하는 방식이다.[11]-[13] 이때 장착점에서 다음 장착점까지의 이동시간은, 장착점에서 다음 장착점에 부품을 공급하는 흡착점까지의 이동시간과, 그 흡착점에서 다음 장착점까지의 이동시간의 합으로 계산되었다. TSP 알고리즘은 nearest neighbor 방법에 의하여 초기 경로를 생성하고 3-Opt 기법에 의하여 경로를 개선시키는 방법을 사용하였으며, 이 방법은 TSP

의 해법으로서 매우 효과적이라고 알려져 있다[19]. 비교대상 알고리즘의 계산 복잡도는 $O(n^3)$ 으로서 제안된 알고리즘과 유사하다.

표 1 은 흡착점의 수를 20개로 고정하고, 장착점의 수를 40 개에서 400 개까지 변화시키며 총 이동시간을 비교한 결과이다. 이 때 각 흡착점의 흡착도수는 모두 동일하게 설정하였다. 장착점 수의 증가에 따라 총 이동시간은 최소 4.5% 에서 최대 16.5% 까지 단축되었음을 알 수 있다.

표 2 는 장착점의 수를 120개로 고정하고, 흡착점의 수를 10개에서 60개까지 변화시키며 전체 이동시간을 비교한 결과이다. 각 흡착점의 흡착도수는 모두 동일하게 설정되었다. 흡착점 수의 증가에 따라 최소 5.0% 에서 최대 8.0% 까지 시간 단축 효과가 있음을 알 수 있다. 이 경우는 장착점의 수가 고정된 상태에서 흡착점의 수가 증가된 경우이며, (25)에 의하여 총 아크의 수는 모두 동일하다. 따라서 표 1 의 경우와 비교하여 시간 단축율이 상대적으로 작아짐을 알 수 있다.

여러 가지 경우에 대한 실험 결과, 수송알고리즘에 의하여 생성된 네트워크가 여러 개의 연결 서브네트워크를 갖는 비연결 네트워크가 되는 경우는 매우 드물게 발생되었다. 일부 발생된 경우에 대하여도 시간 단축율이 크게 저하되지 않음을 확인하였다. 결과적으로 본 논문에서 제안한 방법이 단일헤드 쥘트리형 칩마운터의 조립순서계획에 유용하게 적용될 수 있음을 확인할 수 있었다.

표 1. 장착점 수의 변화에 따른 총 이동시간 비교 (흡착점 수 = 20개).

Table 1. total moving time vs. number of mount points.

장착점 수 (개)	TSP 방법 (sec)	제안 방법 (sec)	단축율 (%)
40	224.7	214.6	4.5
80	434.2	408.4	5.9
120	640.6	599.0	6.5
160	863.0	811.0	7.0
200	1117.7	1013.8	9.3
240	1342.9	1201.8	10.5
280	1555.4	1389.6	10.7
320	1761.9	1538.6	12.7
360	1968.2	1693.7	14.0
400	2197.3	1835.6	16.5

표 2. 흡착점 수의 변화에 따른 총 이동시간 비교 (장착점 수 = 120개).

Table 2. Total moving time vs. number of pickup points.

흡착점 수 (개)	TSP 방법 (sec)	제안 방법 (sec)	단축율 (%)
10	647.3	618.0	5.0
20	640.6	599.0	6.5
30	653.7	608.5	6.9
40	662.5	613.9	7.3
60	662.8	609.7	8.0

VI. 결론

PCB 조립장비인 단일헤드 쥘트리형 칩마운터를 대상으로 하여, 조립시간의 단축을 위한 새로운 조립순서계획 알고리즘을 제시하였다. RPP 또는 TSP 알고리즘을 적용하는 기존의 접근방법과 달리 본 논문은 수송알고리즘을 적용하여 조립경로를 생성시켰다. 즉 최적해를 얻을 수 있는 수송알고리즘을 적용하여 최적의 네트워크를 구성하고, 이로부터 오일러 투어를 발견하여 조립경로를 생성시키는 방법이다. 수송알고리즘에 의하여 생성된 네트워크에서 직접 오일러 투어를 발견할 수 있는 경우는 최적의 조립경로 생성이 가능하다, 그러나 그렇지 않은 경우 네트워크의 수정이 필요하며, 이를 위한 발전적 기법의 알고리즘이 제시되었다. TSP 알고리즘을 적용하는 경우와의 비교 시뮬레이션을 통하여 제안된 방법이 조립시간 단축에 효과가 있음을 검증하였다. 흡착점과 장착점 사이의 왕복운동을 하는 쥘트리형 칩마운터의 경우, 장착점에서 흡착점에 이르는 후향경로의 최적화가 매우 중요하며, 후향경로의 최적화 문제는 수송문제로의 모델링이 가능하다는 점에 착안하여 새로운 방법을 제시하였다.

제시된 알고리즘의 계산복잡도는 수송알고리즘의 계산 복잡도에 의존하며, 헝가리안 알고리즘을 적용하는 경우 $O(n^3)$ (n : 장착점의 수) 이다. 그러나 계산적으로 보다 효과적인 수송알고리즘들에 의한 대체가 가능하므로 계산 시간의 단축이 가능하다. 본 논문에서 제시한 방법은 칩마운터에 내장되는 조립순서계획 소프트웨어의 개발에 활용될 수 있다. 또한 단일헤드 쥘트리형 칩마운터와 매커니즘이 유사한 Pick & Place 로봇의 조립순서 계획문제에 적용 가능하다. 발전적 기법이 적용된 연결알고리즘의 성능을 높이기 위한 추가 연구가 필요하며, 제안된 방법을 다헤드 쥘트리형 칩마운터 등에 적용하기 위한 연구가 추후 진행될 예정이다

V. 참고문헌

[1] T. L. Landers, W. D. Brown et. al., *Electronics Manufacturing Processes*. Prentice-Hall, 1994.
 [2] 박태형, "전자조립용 CAM 시스템의 기술동향", 전자공학회지, 제 26권 제 3호, pp. 48-61, 1999.
 [3] A. R. Hidde and A. Gierse, "An AI-based manufacturing design rule checker and path optimizer for PCB production preparation and manufacturing".

- IEEE Trans. on Components, Hybrid, and Manufacturing Technology*, vol. 15, no. 3, pp. 299-305, 1992.
- [4] 박태형, *Path Optimizer 활용법*, 삼성항공산업(주), 1998.
- [5] 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.
- [6] I. Or and E. Duman, "Optimization issues in automated production of printed circuit boards: operations sequencing, feeder configurations and load balancing problems", *Proc. 1996 IEEE Conf. on Emerging Technology and Factory Automation*, pp. 227-232, 1996.
- [7] W. Wang, P. C. Nelson, and T. M. Tirpak, "Optimization of high-speed multistation SMT placement machine using evolutionary algorithms", *IEEE Trans. on Electronics Packaging Manufacturing*, vol. 22, no. 2, pp. 137-145, 1999.
- [8] J. Sohn and S. Park, "Efficient operation of a surface mounting machine with a multihead turret", *International J. of Production Research*, vol. 34, no. 4, pp. 1131-1143, 1996.
- [9] S. H. Lee, B. H. Lee, and T. H. Park, "A hierarchical method to improve the productivity of a multi-head surface mounting machine", *Proc. of the 1999 IEEE Int. Conf. on Robotics & Automation*, pp. 2110-2115, 1999.
- [10] M. O. Ball and M. J. Magazine, "Sequence of insertions in printed circuit board assembly", *Operations Research*, vol. 36, no. 2, pp. 192-201, 1988.
- [11] D. A. Andreou, "Optimization of the traveling sequence of a 'pick and place' robot by simulating its operation as a traveling salesman problem", *Proc 21st Annual Simulation Symposium*, pp. 91-105, 1988.
- [12] L. F. McGinnis, J. C. Ammons, M. C. Carlyle et. al., "Automated process planning for printed circuit card assembly". *IIE Transactions*, vol. 24, no. 4, 1992.
- [13] M. C. Leu, H. Wong, and Z. Li, "Planning of component placement / insertion sequence and feeder setup in PCB assembly using genetic algorithm", *Journal of Electronic Packaging in Trans. of the ASME*, vol. 115, 1993.
- [14] K. G. Murty, *Network Programming*, Prentice-Hall, 1992. U.S.A
- [15] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*, Prentice-Hall, 1993. U.S.A
- [16] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, Wiley, 1990., Singapore
- [17] J. B. Orlin, "A faster strongly polynomial minimum cost flow algorithm", *Proc. 20th ACM Symposium on the Theory of Computation*, pp. 377-387, 1988.
- [18] D. M. Euter, *Engineering Problem Solving with Matlab*, Prentice-Hall, 1997.
- [19] B. Golden, B. T. Doyle, and W. Stewart, "Approximate traveling salesman algorithms", *Operations Research*, vol. 28, no. 3, pp. 694-710, 1980.



박태형

1988년 서울대학교 제어계측공학과 학사. 1990년 동 대학원 석사 및 1994년 동 대학원 박사. 1992~1994년 제어계측신기술 연구센터 연구원, 1994년~1997년 삼성항공산업(주) 정밀기기연구소 선임연구원. 1997년~현재 충북대학교

전기전자공학부 조교수 관심분야는 반도체 및 전자 조립 시스템, 최적화 알고리즘 응용.



김철한

1999년 충북대학교 전기전자공학부 학사. 1999년~현재 동 대학 제어계측공학과 석사과정. 관심분야는 반도체 및 전자 조립 시스템, 최적화 알고리즘 응용.