

표면실장기의 최적 운영을 위한 모델링 및 알고리즘 개발

Development of Algorithm for Optimal Operation of Surface Mounters

이영해*, 김 정*

Young-Hae Lee*, Jeong Kim

Abstract

Surface mount technology has been evolving in the electronics industries. Efficient operation of surface mounters is closely related with the productivity of the electronic products. In this study, modeling and optimal algorithm for allocating feeders and sequencing mounting jobs in the rotary type surface mounter, which consider all the constraints, in the hardware and are easy to be used in the field, are developed.

1. Introduction

전자 산업의 발달과 더불어 각종 전자 장비에 사용되는 PCB(Printed Circuit Board) 산업은 오늘날 눈부시게 발달하고 있다. 따라서 특히 PCB 위에 칩을 장착시키는 효율적인 운영은 기계의 생산성 향상을 위해 요구되는 중요한 기술로 대두되고 있다. PCB 산업의 고 기능화, 고 집적화, 경량화 등에 맞추어 SMT(Surface Mount Technology; 표면실장기술)의 기술이 급속히 발전되어 SMD(Surface Mount Device; 표면실장부품) 부품

과 이를 위한 기계들이 속속 개발되고 있으며, 이를 PCB 위에 정확하고 고속으로 자동 탑재하는 기기인 고 기능의 로터리형 표면실장기(Rotary Type Surface Mounter: RTSM)의 개발이 국내외에서 활발히 진행되고 있다.

전자공학 및 기계공학 등의 발달 덕분에 하드웨어의 신뢰도는 상당한 수준에 올라와 있으며 앞으로 더욱 발달될 것으로 기대된다. 하지만 운영 방법에 대해서는 아직까지 낮은 수준에 머물러 있는 바, 현장의 작업자들은 기계 운영을 위해 매우 고되고 저능률의 작업을 수행하고 있다.

* 한양대학교 산업공학과

표면실장기의 운영을 위해 작업자들은 부품의 카세트 위치를 결정하고 장착 순서를 결정한 후, 이에 대한 NC Part Program을 작성한다. 현장에서 오랫동안 근무해 온 작업자라고 하더라도 수작업으로 일일이 적게는 50점, 많게는 350점 이상의 부품 장착에 대해서 효율적인 장착 순서를 결정하고 카세트의 위치를 결정하는 데에는 시간이 많이 요구될 뿐 만 아니라 매우 고된 일이 된다. 또한 그렇게 결정된 장착 순서와 카세트의 위치가 기계 가동율을 향상시키는 지에 대한 확신을 보장해 주지 못하기 때문에 합리적인 운영 전략을 컴퓨터를 이용하여 자동적으로 생성시키는 알고리즘 및 소프트웨어를 개발하는 것이 더욱 요청되고 있다.

PCB 조립작업과 관련된 기존의 연구를 살펴 보면 다음과 같다. Chan[7]등은 칩을 삽입할 때, 발생하는 작업 순서 결정 문제를 풀기 위해 TSP(Traveling Salesman Problem)를 적용하였으나, 좀더 효율적인 알고리즘이 개발되지 않는 한 TSP의 적용이 어렵다는 결론을 내렸다. Leipala and Navalainen[9]은 로터리형 표면실장기의 운영을 위한 카세트의 위치를 임의로 배정하여 초기의 장착 순서를 결정한 후 카세트의 위치를 계속 바꾸어 가면서 초기해를 향상시켜 주었다. Vellanki[10]등은 PCB 조립 공정에서 가장 중요한 문제들 중 하나인 TSP와 유사한 부품의 최적 삽입 순서 결정에 관한 문제를 인공 신경 회로망 기법을 이용하여 외판원이 30개의 도시를 방문하는 TSP에 적용하였다. 그러나 휴리스틱을 이용한 방법도 도시의 수가 증가할수록 컴퓨터 수행 시간이 오래 걸리는 것이 현재까지 문제가 되고 있으며 계속적으로 컴퓨

터 수행 시간 단축 문제가 연구의 중요한 문제점으로 대두되고 있다. Walas and Askin[11]은 로터리형 표면실장기와 유사한 방법으로 작업을 수행하는 NC Turret Punch Press의 Tool의 위치와 Hitting Sequence를 결정하는 경험적 알고리즘을 제시하였다. 그들은 Hitting Sequence를 NNR(Nearest Neighborhood Rule)을 적용하여 노드들 간의 거리를 고려하여 결정해 준 후 QAP(Quadratic Assignment Problem)의 경험적인 해법인 VNZ 방법을 활용하여 Tool의 배치를 계속 바꾸어 가면서 초기해를 향상시켰다. 손진현[12]은 표면실장기를 이용하여 부품 장착 작업을 수행하는 효율적인 운영 알고리즘들을 Leipala and Navalainen[9]의 방법을 보다 발전시켰다. 그는 먼저 PCB 위에 장착할 부품들중에서 단 한개의 부품을 필요로 하는 모든 노드들을 먼저 추려내어 최단거리방법을 이용하여 부품 장착의 순서와 부품 카세트의 위치를 결정하여 고정시킨 후 나머지 부품들에 대해서는 Leipala and Navalainen[9]나 Walas and Askin[11]의 방법과 매우 유사하게 초기해를 계속 향상시켜 주었다.

이외에도 많은 연구자들이 표면실장기의 효율적인 운영을 위한 알고리즘을 개발해 왔으나 매우 복잡한 시스템을 시스템 분석 및 모델링을 정확하게 하지 않거나 생략한 관계로 모든 변수들을 고려할 수 없었으며, 따라서 현장에 잘 적용되지 않았다. 또한 RTSM 문제는 작업 소요 시간이 최소가 되도록 하는 가능해를 구하기 위해서는 실제로 기계에서 요구되는 모든 제약 조건들을 만족해야 한다. 본 연구에서는 로터리형 표면실장기의 운영 시스템을 분석하여 시스템의 운영 효율

에 영향을 주는 인자들과 변수들을 정의하고 정의된 모든 인자와 변수들을 고려한 모델링과 이를 위한 알고리즘을 제시한다.

2. 시스템의 구조

일반적으로 로터리형 표면실장기는 크게 릴 축(Reel Axis), 인덱스 테이블(Index Table)부, 그리고 X-Y 테이블(X-Y Table)부의 3 부분으로 구성되어 있다. 로터리형 표면실장기의 구성은 시스템의 종류마다 다양하지만 본 연구의 대상으로 삼은 로터리형 표면실장기(그림 1 참조)에 대한 개략적인 설명은 아래와 같다[1,2].

지하지만 12mm와 16mm 부품의 카세트는 2단위의 스페이스를 차지한다. 그리고 24mm 부품의 카세트는 3단위의 스페이스를 차지하고 32mm부품의 카세트는 4 단위의 스페이스를 릴 축에서 차지하게 된다. PCB 위에 장착될 부품들은 테이프에 박혀서 부품 카세트에 감겨져 있으며 부품 카세트는 릴 축에 나란히 놓여 있다. 릴 축은 수평 운동을 수행하면서 지정된 부품의 카세트를 피킹 스테이션(Picking Station) (①)에 위치시킨다.

릴 축의 단위 움직임을 피치라고 하며 0 피치 또는 1 피치 이동을 위해 요구되는 시간은 연구대상 로터리형 표면실장기 운영의 최소 시간(예를 들면, 본 연구에서 고려한 시

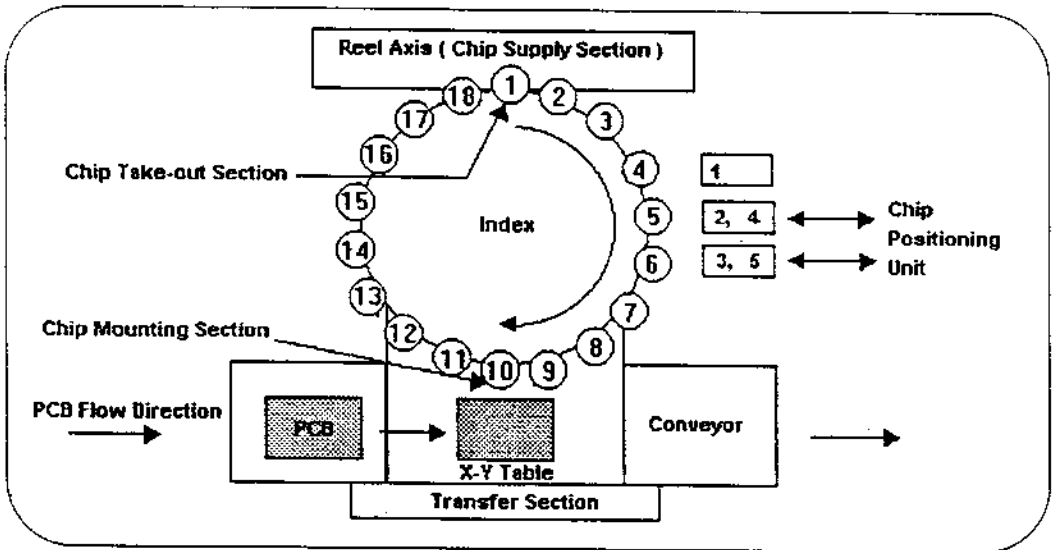


그림 1. 로터리형 표면실장기의 개략도

그림 1의 릴 축에는 다양한 부품 카세트를 장착시킬 수 있게 되어 있다. 릴은 일반적으로 8mm단위로 구분된다. 따라서 8mm부품의 카세트는 릴 축에서 1 단위의 스페이스를 차

스템에서는 0.25초임)이 요구되며, 2 피치 이동에 요구되는 시간은 0.3초 등, 릴 축에서 소요되는 시간은 지정된 부품을 피킹 스테이션까지 위치시키기 위하여 요구되는 이동 피

치수에 따라 달라진다[2]. 따라서 가급적 카세트를 움직이지 않거나 1 피치씩만 릴 축을 이동하여 부품을 피킹 스테이션에 위치시킬 수 있으면 이상적이라고 할 수 있다. 그런데 본 연구에서 대상으로 하는 표면실장기는 하드웨어의 특성상 부품을 헤드에 공급하기 위해 릴 축을 10피치 이상 움직일 수 없다[2]. 릴 축에서 작업 소요 시간에 영향을 주는 변수는 이동 피치수가 되는데 이는 연속적으로 장착되는 부품의 크기 및 위치에 따라 달라진다.

인덱스 테이블은 18개의 헤드를 가지고 있으며 작업 스테이션 위에서 회전할 수 있다. 인덱스 테이블에 부착되어 있는 각각의 헤드들은 피킹 스테이션에 도달한 카세트로부터 부품을 피킹한다. 그리고 이를 부품 검사(②), 부품의 Positioning(④⑤⑥) 등 여러 Operation을 수행하도록 여러 개의 스테이션을 거친 후, Mounting Station(①)으로 운반한다. 운반하는 속도는 부품의 종류에 따라 다르다. 포지셔닝할 때 Shuttling이 발생하면 인덱스 테이블의 속도가 영향을 받는다. Shuttling은 (2)번 유닛과 (4)번 유닛을 이용하여 부품을 포지셔닝하거나 또는 (3)번 유닛과 (5)번 유닛을 이용하여 포지셔닝 할 때 발생한다. 따라서 하드웨어의 (2)번 유닛과 (4)번 유닛, 그리고 (3)번 유닛과 (5)번 유닛을 이어서 사용할 수 없다. 역으로도 마찬가지이다.

그리고 Mounting Angle(장착 각도)이 180도 이상 차이가 있는 부품을 Mounting할 때, Mounting Station에서 인덱스 테이블의 지연이 발생하므로 작업 소요 시간에 영향을 준다. 따라서 인덱스 테이블에서 운영 최적화

에 영향을 주는 인자는 인덱스 테이블의 지정속도, 사용되는 포지셔닝 유닛, 그리고 장착 각도이다.

X-Y 테이블은 각 부품에 따라 지정된 속도로 PCB를 상하좌우로 이동시켜서 Mounting Position을 Mounting Station에 위치시킨다. 현재의 Mounting Position은 다음의 Mounting Position을 결정하는 데 영향을 준다. 거리의 계산은 체비체프 거리의 개념을 따른다. X-Y 테이블의 속도가 High Level로 지정된 경우 0.25초 동안에 X-Y 테이블은 현재 위치에서 32mm까지 이동이 가능하지만 Middle Level로 지정된 경우에는 22mm, 그리고 Low Level로 지정된 경우에는 9mm까지 이동이 가능하다[2].

3. 모델링

로터리형 표면실장기의 운영 최적화를 위한 일반적으로 이용되는 간단하고, 전형적인 수리 모델을 아래에 제시하는데 이를 위한 몇가지 가정들은 다음과 같다.

- 1) 인덱스 테이블의 속도는 모든 부품에 대해 동일하다.
- 2) 장착 각도의 상대적인 차이는 표면실장기의 운영 시간에 영향을 미치지 않는다.
- 3) 부품을 포지셔닝하기 위한 유닛의 사용이 변화되어도 표면실장기의 운영시간에 영향을 미치지 않는다.
- 4) 부품의 크기는 모두 동일하다.

위의 가정을 생각한다면 단지 PCB 상의 노드들 간의 거리와 부품 카세트 위치에 따른 릴 축의 피치 이동수가 주요 변수가 된다. 이

들 두 변수만을 고려한다면 m 개의 카세트를 l 개의 릴 축 위에 배치시키고, n 개의 장착점에 대해 장착 순서를 결정하는 문제가 된다. 위의 가정들만을 고려한 최적화 모델은 다음과 같다[11]

$$Min \sum_{r=1}^l \sum_{i=1}^l \sum_{j=1}^n \sum_{k=1}^n c_{ijrs} x_{ijk} y_{lr} y_{ls} \quad (1)$$

$$s.t \sum_{i=1}^n \sum_{k=1}^n x_{ijk} = 1 \quad 1 \leq j \leq n. \quad (2)$$

$$\sum_{j=1}^n \sum_{k \in K_0^1} x_{ijk} = 1 \quad 1 \leq i \leq n. \quad (3)$$

$$\sum_{v=1}^n \sum_{k=1}^n x_{ivk} \geq \sum_{v=1}^n \sum_{k=1}^{k_0} x_{vjk}, \quad 1 \leq k_0 \leq n. \quad (4)$$

$$\sum_{i,l} x_{ijk} = \sum_{v,v \neq j} x_{jv(k+1)} \quad 1 \leq j \leq n, 1 \leq k \leq n-1 \quad (5)$$

$$x_{ijk} = 1 \text{ or } 0 \quad (6)$$

$$\sum_{u=1}^l y_{ur} = 1 \quad 1 \leq r \leq l \quad (7)$$

$$\sum_{r=1}^l y_{ur} = 1 \quad 1 \leq u \leq l \quad (8)$$

$$y_{ur} = 0 \text{ or } 1 \quad (9)$$

여기서

c_{ijrs} : 만일 장착점 j 가 장착점 i 의 뒤에 오고, 장착점 i 를 포함하는 카세트가 릴 축 위의 위치 r 에 있고, 장착점 j 를 포함하는 카세트가 릴 축 위의 위치 s 에 있을 때, 재배치(피치 이동)에 대한 지연 시간.

$$x_{ijk} = \begin{cases} 1 & \text{장착점 } i \text{에서 장착점 } j \text{까지의} \\ & \text{이동이 } k \text{번째이면} \\ 0 & \text{Otherwise} \end{cases}$$

$$y_{tir} = \begin{cases} 1 & \text{장착점 } i \text{를 포함하는 카세트의} \\ & \text{위치가 릴 축 위의 위치 } r \text{이면.} \\ 0 & \text{Otherwise} \end{cases}$$

t_i : 장착점 i 를 포함하는 카세트

목적식 (1)은 인접하는 장착점의 n 개 쌍을 선택하여 요구된 카세트의 위치에 따라 비용(시간)이 정해진다. 식(2)와 (3)은 각 장착점 위치에 정확히 한 번씩만 거쳐 간다. 식(4)는 장착점에 대하여 선행 제약을 나타낸다. 식(5)는 가능한 장착 순서가 얻어지도록 부분환(subtour)을 갖지 않게 함으로 일련의 순서를 갖게 한다. 식(7)은 각 위치(location)가 정확히 하나의 카세트를 갖는 것을 요구한다. 식(8)은 각 카세트가 정확히 한 위치에 할당되게 한다. 이상의 식에 대한 값은 2개의 고전적인 최적화 문제로 볼 수 있는데, 주어진 식에서 x_{ijk} 는 식(2)에서부터 식(6)을 거쳐 그 해, 즉 장착점 순서의 도출이 가능하며, 만일 y_{ur} 이라는 변수를 이용하여 목적식을 다시 고쳐 쓴다면, QAP로 생각할 수 있다. 비슷하게, 변수 y_{ur} (카세트 위치)의 조합은 제한식(7)에서 식(9)를 거쳐 그 해를 구할 수 있는데, 이는 선행 제약이 있는 TSP로 남게 된다.

위의 수리 모델은 2개의 고전적인 최적화 문제인 TSP와 QAP로 나누어서 생각할 수 있음을 알 수 있다. 그러나 TSP는 NP-Complete의 전형적인 예로서 최적해를 구하는 다항식 알고리즘이 존재하지 않는다고 알려져 있으며, 또한 모든 경우의 카세트 위치를 고려한 상황 하에서 장착 순서를 결정하는 문제는 QAP 문제로써, NP-Hard로 알려져 있다. 따라서 이는 모든 노드들에 대해 모든 카세트 위치의 경우들을 고려하여 장착 순서를 결정해야 하는 문제이기 때문에 합리적인 시간 내에 최적해를 구하는 것이 어렵다. 이와 관련된 연구로는 몬테칼로 시뮬레이션 기법을 이용한 알고리즘[7]이나 Glover[8]의 Tabu

Search와 같은 방법들이 있다. Tabu Search 알고리즘은 비교적 우수한 해를 제시한다고 알려져 있으나 개방형(open-ended)의 형태이기 때문에 컴퓨터 수행 시간이 너무 많이 요구된다.

현재까지 로터리형 표면실장기의 효율적인 운영을 위해 개발된 알고리즘들은 위의 수리모형을 기본으로 하여 X-Y 테이블의 이동거리만을 고려하여 개발되었다. 그런데 이렇게 결정된 운영 방법에 따라 표면실장기를 운영할 경우, 다른 변수의 영향을 고려하지 못한 관계로 기계의 효율적 운영이 이루어지지 않을 뿐만 아니라, 하드웨어의 제약으로 인해 대부분 기계를 돌리지 못하는 경우들이 빈번하게 발생된다. 따라서 이러한 문제의 해결을 위하여 본 연구에서는 표면실장기의 운영에 영향을 주는 모든 인자와 변수들을 고려한 카세트의 위치와 장착 순서를 결정할 수 있는 알고리즘을 제시한다.

4. 시스템 분석 및 인자의 선정

릴 축과 인덱스 테이블 그리고 X-Y 테이블은 각각 다른 모터에 의해 동작함으로써 부품의 공급과 이동 및 장착 작업은 순차적으로 진행되지 않고 각각 다른 작업 스테이션에서 동시에 이루어진다. 그리고 장착해야 할 PCB 상의 모든 점들을 네트워크 상의 노드로 생각할 수 있다. 부품은 자신들에 지정된 정보에 따라 PCB 위에 장착된다. 그러나 PCB 위의 노드들에 한 부품이 어떤 부품 다음에 장착되느냐에 따라 장착 시간은 달라질 수 있다. 장착 순서에 따라 장착 시간을 변화시키는 모든 인자들을 RTSM을 구성하는 주요

부분인 릴 축과 인덱스 테이블 그리고 X-Y 테이블 각각에서 선정한다.

4.1 인덱스 테이블

인덱스 테이블에는 테이블의 속도, 포지셔닝 유닛, 그리고 장착 각도 등의 3가지 인자가 있어서 부품 장착 작업 시간에 영향을 준다. 인덱스 테이블의 속도는 부품의 종류마다 다양해 질 수 있다. 인덱스 테이블의 속도가 다양해질 때 인덱스 테이블은 가장 느린 속도에 지배를 받아 회전하게 된다. 각각의 지정된 인덱스 테이블의 속도를 PCB에 장착되는 장착점의 기준에 따라 인덱스 테이블 속도에 대한 매트릭스를 시간으로 환산하여 표현하면 아래의 식(10)과 같이 나타낼 수 있다.

$$IN_{ij} = Time(s_i, s_j) \quad (10)$$

여기서 IN_{ij} 는 한 헤드가 인덱스 테이블의 회전 지정 속도 s_i 인, 노드 i 에 장착할 부품을 피킹 한 후, 바로 다음의 다른 헤드가 인덱스 테이블의 회전 지정 속도 s_j 인, 노드 j 에 장착할 부품을 피킹 했을 때 인덱스 테이블이 회전하게 되는 지정 속도를 시간으로 환산한 값이다.

또한 부품이 PCB의 한 노드에 장착되기 전, 인덱스 테이블의 헤드에 물려 있는 부품의 자세를 바로 잡아 주는 역할을 하는 것을 포지셔닝 유닛이라고 한다. 부품의 모양에 따라 사용되는 포지셔닝 유닛은 달라질 수 있다. 포지셔닝 작업은 그림 1의 ④, ⑤, 그리고 ⑥번 스테이션에서 진행되며 유닛의 종류는 (1)번, (2)번, (3)번, (4)번, 그리고 (5)번 유닛 등 5가지가 있다. 부품마다 사용되는 유

닛은 지정되어 있으므로 작업이 진행되는 도중 임의로 변화되지 않으나, 사용되는 유닛의 순서에 따라 유닛의 교체가 있어야 됨으로 이에 따라 부품을 포지셔닝을 하는 시간은 달라질 수 있다. 예를 들어서 (2)번 유닛을 사용한 후, (4)번 유닛을 바로 사용하게 되면 하드웨어의 특성상 포지셔닝을 완료할 때까지 기계는 정지하게 되는 결과를 초래하게 된다. 한편 (2)번 유닛을 사용한 후 (4)번 유닛을 사용하기 전, 다른 유닛을 몇 번 사용하느냐에 따라 포지셔닝 시간은 달라진다. 이는 (4)번 유닛을 사용한 후 (2)번 유닛을 사용하는 경우, (3)번 유닛을 사용한 후 (5)번 유닛을 사용하는 경우, 그리고 (5)번 유닛을 사용한 후 (3)번 유닛을 사용한 후에도 마찬가지이다. 이에 대하여 사용 포지셔닝 유닛에 대한 매트릭스를 시간으로 환산하여 부품을 PCB에 장착시키는 노드에 따라 아래의 식(11)과 같이 나타낼 수 있다.

$$PO_{ij} = Time(u_i, u_j) \quad (11)$$

여기서 PO_{ij} 는, 노드 i 에 장착할 부품을 포지셔닝하기 위하여 사용되는 유닛가 u_i 이고 노드 j 에 장착할 부품을 포지셔닝하기 위하여 사용되는 유닛이 u_j 일 경우, 노드 i 다음에 노드 j 에 부품을 장착할 때 포지셔닝을 위해 요구되는 시간을 의미한다.

그리고 부품을 장착시킬때 장착할 각도는 미리 지정된다. 그런데 장착 각도는 이전의 장착점의 각도에 의해 영향을 받게 되는데, 이전 장착 각도와 상대적 차가 180도 이상 이 되는 경우에는 장착 시간의 지연을 초래하게 된다. 이에 대하여 PCB에 장착되는 상대적 각도 차이에 대한 매트릭스를 시간으로

환산하여 식 (12)에 나타낸다.

$$AN_{ij} = Time(|g_i - g_j|) \quad (12)$$

여기서 AN_{ij} 는 인덱스 테이블의 한 헤드가 피킹한 부품을 노드 i 에 각도 g_i 로 장착한 후 바로 다음의 헤드가 피킹한 부품을 노드 j 에 각도 g_j 로 장착할 때, 장착 각도의 조정을 위해 요구되는 시간을 나타낸다.

한편 각 부품은 지정된 속도로 헤드에 붙잡혀서 테이블 위를 움직인다. 부품이 ④번, ⑤번, 그리고 ⑥번 스테이션에 이르게 되면 포지셔닝을 위해 잠시 정지한 후 그 일이 완료되면 다시 움직임을 시작하여 장착 작업을 위해 ①번 스테이션에 이르게 된다. 그리고 ①번 스테이션에서 대기중인 PCB 위에 장착이 되는데 만일 장착 각도를 조정할 필요가 있으면 각도 조정을 위해 잠시 멈춘 후 조정이 완료되면 준비된 PCB 위의 지정된 점에 장착된다. 이러한 일련의 과정은 그림 2와 같다.

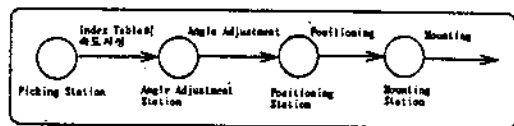


그림 2. 인덱스 테이블 구동 원리

따라서 인덱스 테이블 만을 고려할 때 이의 작업 소요 시간은 식 (13)과 같다.

$$INT_{ij} = \max(IN_{ij}, PO_{ij}, AN_{ij}) \quad (13)$$

여기서 INT_{ij} 는 한 부품이 노드 i 에 장착되고 이어서 다른 부품(이전의 부품과 같은 종류 혹은 다른 종류의 부품)이 노드 j 에 장착되기 위해, 인덱스 테이블에서 요구하는 작업 소요 시간을 나타낸다.

4.2 X-Y 테이블

X-Y 테이블은 PCB의 지정된 노드를 장착 스테이션에 대기시킨다. 테이블의 지정 속도에 따라 단위 시간당 이동거리가 달라진다. 시스템의 특성상 이동거리는 체비체프 개념을 따르며 식(14)로 표현될 수 있다.

$$XY_{ij} = Time\{\max(|x_i - x_j|, |y_i - y_j|)\} \quad (14)$$

여기서 XY_{ij} 는 X-Y 테이블이 노드 i의 위치인 (x_i, y_i) 부터 노드 j의 위치인 (x_j, y_j) 까지 PCB를 운반하기 위하여 필요한 시간을 나타낸다.

인덱스 테이블과 X-Y 테이블의 관계를 살펴본다. 부품을 피킹한 헤드가 장착 스테이션에 도착하여 부품을 장착하려고 할 때 PCB가 준비되어 있지 않으면, 준비될 때까지 그 헤드는 기다려야 한다. 이와 반대로, X-Y 테이블이 PCB를 장착 스테이션으로 운반하여 장착 준비를 완료했을 지라도, 부품을 피킹한 헤드가 도착하지 않았으면 그 헤드가 도착할 때까지 X-Y 테이블은 움직일 수 없다. 즉 인덱스 테이블과 X-Y 테이블 중에서 늦게 작업을 완료하는 부분이 두 부분의 사이클 타임을 결정한다. 이러한 관계를 도시하면 그림 3과 같다.

4.3 릴 축

릴 축은 수평 운동을 수행하면서 부품 카세트를 피킹 스테이션으로 운반하며, 피치 이동수는 부품 카세트의 크기와 위치에 따라 달라진다. 카세트의 크기는 부품의 크기에 의해 결정된다. 부품 카세트의 위치는 결정 변수가 된다. 가장 이상적인 카세트의 위치는 부품을 장착하는 순서대로 카세트의 위치를

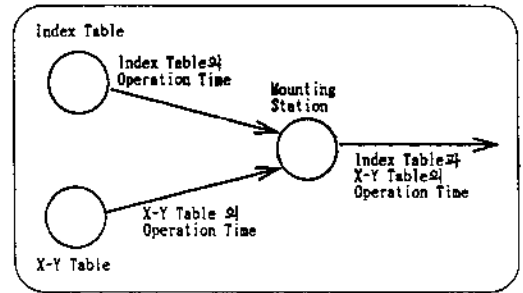


그림 3. 인덱스 테이블과 X-Y 테이블의 관계

정하는 것이라고 생각할 수 있다. 이 때 릴 축의 이동에 의해 추가로 발생하는 지연은 없다. 임의의 한 헤드가 부품을 카세트로부터 피킹한 후, 바로 다음의 헤드가 다른 종류의 부품을 피킹하려고 할 때, 그 부품의 카세트가 이전 카세트의 바로 옆에 배치되어 있으면 릴 축의 이동 피치수를 최소로 하게 하는 위치라고 할 수 있다. 카세트 크기만 고려한 피치 이동수에 대한 매트릭스를 식 (15)와 같이 구성할 수 있다.

$$PT_{ij} = Time(p_i, p_j) \quad (15)$$

여기서 PT_{ij} 는 노드 i에 장착할 부품 카세트의 크기가 p_i 이고 노드 j에 장착할 부품 카세트의 크기가 p_j 일때, 노드 i에 부품을 장착한 후 바로 이어서 노드 j에 부품을 장착할 때 릴 축의 이동 피치수를 시간으로 나타낸 것이다.

카세트의 위치를 추가로 고려하면 식 (15)에 나타난 매트릭스의 각 요소는 같거나 더 커지게 된다. 인덱스 테이블과 X-Y 테이블, 그리고 릴 축의 관계를 그림 4와 같이 나타낼 수 있으며, 작업 소요 시간은 식(16)과 같이 나타낼 수 있다.

$$= \max\{INT_{ij}, XY_{ij}, PT_{ij}\} \quad (16)$$

여기서 T_{ij} 는 노드 i 에 부품을 장착시킨 후 바로 노드 j 에 부품을 장착시키기 위하여 요구되는 최소한의 작업 시간을 의미한다. 식 (16)에는 카세트의 위치를 고려하지 않았기 때문에 각각의 요소는 현재의 값 이상이 된

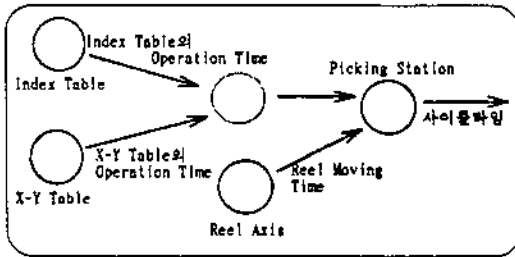


그림 4. 인덱스 테이블, X-Y 테이블, 릴 축 간의 관계

다.

이상에서 살펴 본 내용을 바탕으로 RTSM의 최적 운영에 영향을 주는 인자들을 정리하면 표 1과 같다.

표 1. RTSM의 운영에 영향을 주는 인자들

구 분	인덱스 테이블	X-Y 테이블	Reel Axis
인자의 종류	인덱스 테이블 속도	X-Y 테이블 속도	카세트 크기
	포지셔닝 유닛	테이블 이동거리	카세트 위치
	장착각도		

5. 알고리즘의 개발

3장에 보여진 수리 모델은 많은 가정을 하여 모델을 단순화시켰음에도 불구하고 QAP 문제가 되어 이를 해결할 수 있는 바람직한 해법은 아직까지 개발되지 않았다. 본 장에

서는 부품을 PCB 위에 장착할 노드 수가 작은 경우, 기존 해법을 이용하여 최적해를 산출할 수 있는 알고리즘을 제시한다.

5.1 부품의 크기가 모두 기본 단위이고 릴 축에 카세트를 장착점 수 만큼 배치시킬 수 있는 경우

부품의 크기가 기본 단위인 8mm 부품을 담게 되는 카세트는 릴 축에서 1 단위 만큼의 공간을 차지하게 된다. 따라서 릴 축이 1 피치 이동함으로써 적용되는 최소 단위시간 내에 부품 카세트를 피킹 스테이션으로 옮길 수 있다. 그리고 부품 카세트를 릴 축에 원하는 만큼 배치할 수 있을 때(일부 모델은 100개까지 가능) 부품 카세트는 장착 순서대로 배치가 가능하게 된다. 이러한 경우, 최적의 부품 카세트 위치 및 장착 순서는 다음과 같은 절차로 결정된다.

Step 1. 최적 장착 순서 결정.

시간에 대한 매트릭스 T 를 이용하여 아래와 같은 수리 모델을 풀어서 최적 순서를 결정한다.

$$\min Z = \sum_{i=1}^n \sum_{j=1}^n T_{ij} x_{ij} \quad (17)$$

$$\text{s.t. } \sum_{i=1}^n x_{ij} = 1 \quad \text{for all } j \quad (18)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{for all } i \quad (19)$$

$$x_{ij} = 1 \text{ or } 0 \quad (20)$$

x_{ij} should make Hamiltonian cycle.

Step 2. 부품 카세트 위치 결정.

Step 1에서 결정된 장착 순서대로 부품 카

세트를 릴 축에 배치한다.

<예제 1.>

표 2에 가정한 T 매트릭스를 사용하여 위의 알고리즘을 설명하면 다음과 같다.

표 2. 가정된 매트릭스 T

	A01 (1)	A01 (2)	B02 (3)	B02 (4)	C20 (5)	D22 (6)	D22 (7)
A01 (1)	-	0.6	0.25	0.45	0.5	0.6	0.7
A01 (2)	0.8	-	0.9	0.7	0.5	0.25	0.45
B02 (3)	0.5	0.25	-	0.7	0.8	0.9	0.45
B02 (4)	0.30	0.40	0.45	-	0.25	0.7	0.8
C20 (5)	0.9	0.35	0.6	0.35	-	0.3	0.25
D22 (6)	0.45	0.7	0.8	0.35	0.45	-	0.5
D22 (7)	0.3	0.45	0.5	0.6	0.7	0.8	-

Step 1. 장착 순서를 결정한다.

5.1절의 Step 1.을 풀면 아래와 같은 최적의 장착 순서가 얻어진다.

A01 → B02 → A01 → D22 → B02 → C20 → D22

Step 2. 카세트의 위치를 결정한다.

T 를 이용하여 최적의 장착 순서를 결정하고 그 순서대로 카세트를 릴 축에 배치하면 된다. 즉 PCB의 모든 노드 수 만큼 릴 축에 카세트를 장착할 수 있다면 앞에서 구한 장착 순서와 카세트의 위치가 최적어 되는 것이다. 왜냐하면 T 는 카세트의 위치를 제외한 시스템의 모든 변수를 대표하며 그 T 를 이용

하여 최적의 장착 순서를 결정하고 동시에 그 장착 순서대로 카세트의 위치를 배치하였기 때문에 카세트의 위치에 의해 최적 장착 순서가 전혀 영향을 받지 않게 되기 때문이다. 이리므로 카세트의 최적 위치는 표 3과 같다.

표 3. 최적 카세트의 위치

카세트위치	릴 1	릴 2	릴 3	릴 4	릴 5	릴 6	릴 7
카세트의 종류	A01	B02	A01	D22	B02	C20	D22
배치된 릴 번호	1	2	3	4	5	6	7

5.2 장착되는 부품 종류의 수 만큼 릴 축에 카세트를 배치하는 경우

부품의 크기가 다양하고 릴 축에 부품 카세트를 장착할 수 있는 공간이 부품 종류의 수 만큼 허용할 때 최적의 부품 카세트의 위치와 장착 순서는 다음 Principle 1과 Principle 2를 기본으로 하여 아래 절차에 따라 결정된다.

Principle 1. 최적의 장착 순서대로 배치된 부품 카세트의 위치는 같은 부품 종류의 카세트들끼리 모아질 수 있다.

Principle 2. 피치 이동 수의 합이 최소가 되도록 같은 종류의 카세트들끼리 모아질 때, 결정된 최적 장착 순서는 카세트의 합친 결과에 의해 변동된 피치 이동 시간의 영향을 가장 적게 받는다.

Step 1. 5.1절의 Step 1과 동일한 방법으로 장착 순서를 결정한다.

Step 2. 같은 부품들끼리 모아서 릴 축에 순서대로 배치시킬 때 각각에 대한 피치이동 수의 합을 계산하여 매트릭스를 만든다.

Step 3. 아래의 수리 모델을 풀어 카세트의 위치를 결정한다.

$$\min F_m = \sum_{k=1}^r \sum_{l=1}^r p_{kl} y_{kl} \quad (21)$$

$$\text{s.t. } \sum_{k=1}^r y_{kl} = 1 \quad \text{for all } l \quad (22)$$

$$\sum_{l=1}^r y_{kl} = 1 \quad \text{for all } k \quad (23)$$

$$y_{kl} = 1 \text{ or } 0 \quad (24)$$

여기에서 p_{kl} 은 임의의 부품 카세트 k 를 릴의 위치 l 에 배치시켰을 때 피치 이동 수의 함을 나타내고 y_{kl} 은 부품 카세트의 위치결정 변수가 된다.

Step 4. 결정된 카세트의 위치를 이용하여 카세트 위치에 의한 피치 이동 매트릭스를 식(25)와 같이 만든다.

$$FM_{ij} = \text{Time}(m_i, m_j) \quad (25)$$

여기서 노드 i 에 장착될 부품의 카세트가 릴 축의 m_i 에 있고 노드 j 에 장착될 부품의 카세트가 릴 축의 m_j 에 있을때, FM_{ij} 는 노드 i 에 부품을 장착한 후 바로 다음에 노드 j 에 부품을 장착할 때 릴 축이 이동하는 데 요구하는 시간을 의미한다.

Step 5. 카세트의 위치가 결정되었으므로 이를 포함하여 RTSM의 모든 변수들을 포함하는 새로운 매트릭스를 식(26)과 같이 만든다.

$$NTOT_{ij} = \max(T_{ij}, FM_{ij}) \quad (26)$$

Step 6. 다음의 수리모델을 풀어서 장착 순서를 결정한다.

$$\min Z = \sum_{i=1}^n \sum_{j=1}^n NTOT_{ij} x_{ij} \quad (27)$$

$$\text{s.t. } \sum_{i=1}^n x_{ij} = 1 \quad \text{for all } j \quad (28)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{for all } i \quad (29)$$

$$x_{ij} = 1 \text{ or } 0 \quad (30)$$

x_{ij} should make Hamiltonian cycle.

<예제 2.>

표 2에 제시된 매트릭스 T 를 이용한다.

Step 1. 초기 장착 순서를 결정한다.

T 에서 구한 초기의 장착 순서는 아래와 같다.

$$A01 \rightarrow B02 \rightarrow A01 \rightarrow D22 \rightarrow B02 \rightarrow C20 \rightarrow D22$$

$$(1) \quad (3) \quad (2) \quad (6) \quad (4) \quad (5) \quad (7)$$

Step 2. 피치 이동수에 대한 매트릭스를 만든다.

먼저 릴 축의 제한이 없을 때 최적 카세트의 위치는 표 3과 같다. 한편 릴 축의 제한 때문에 같은 부품을 릴 축의 임의의 한 위치에 모아야 한다. 그런데 만일 부품 A01을 릴 번호 1에 모으려면 릴 번호 3번에 있는 A01만 릴 번호 1로 이동 시키면 된다. 이때 이동되는 피치 이동수는 2단위가 된다. 만일 릴 번호 4에 A01을 배치시키려면 릴 번호 1과 3에 있는 A01을 이동시켜야 한다. 이때 피치 이동수는 3단위가 된다. 이와 같이 같은 종류의 부품을 모아 릴 축에 배치시키면 부품 종류의 수만큼 카세트를 사용할 수 있게 된다. 위 예제에서 부품의 종류는 4가지이다. 따라서 같은 릴 축을 1번부터 4번까지 사용한다. 이동피치 수에 대한 매트릭스를 구하면 표 4와 같다.

Step 3. 카세트의 위치를 결정한다.

초기 장착 순서대로 작업을 수행할 경우 피치 이동을 최소로 하게 하는 부품 카세트

표 4. 피치 이동 수에 대한 매트릭스

카세트종류 \ 릴 번호	릴 번호			
	릴 1	릴 2	릴 3	릴 4
A01	2	2	2	4
B02	5	3	3	3
C20	5	4	3	2
D22	9	7	5	3

의 위치를 결정한다. 표 4의 각 요소들을 식 (21)의 p_{ki} 에 대입하여 풀면 아래 표 5와 같은 카세트의 위치를 구할 수 있다.

표 5. 릴 측에 배치된 카세트의 위치

릴 번호	릴 1	릴 2	릴 3	릴 4
카세트의 종류	A02	B02	C20	D22

Step 4. 피치 이동 매트릭스를 구성한다.

표 5에 나타난 카세트의 위치 정보를 토대로 식 (25)를 이용하여 피치이동 매트릭스를 만든다. 그 결과를 나타내면 표 6과 같다.

Step 5. 모든 변수들을 통합하여 새로운 매트릭스를 만든다.

장착 순서는 카세트의 크기와 피치 이동을 모두 고려하여 결정되어야 한다. 따라서 위에서 구한 피치 이동 매트릭스를 식 (26)에 대입하여 매트릭스 $NTOT$ 를 구한다. 이를 표 7에 나타낸다.

Step 6. 최적의 장착 순서를 결정한다.

$NTOT$ 를 식(27)에 대입하여 5.2절의 Step 6.을 풀면 최종 장착 순서는 아래와 같다.

A01 → B02 → A01 → D22 → B02 → C20 → D22

(1) (3) (2) (6) (4) (5) (7)

이상에서 살펴 본 바와 같이 현재까지 QAP

표 6. 카세트의 위치에 따른 릴 측의 피치 이동 매트릭스

	A01	A01	B02	B02	C20	D22	D22
A01	-	0.25	0.25	0.25	0.35	0.4	0.4
A01	0.25	-	0.25	0.25	0.35	0.4	0.4
B02	0.25	0.25	-	0.25	0.25	0.35	0.35
B02	0.25	0.25	0.25	-	0.25	0.35	0.35
C20	0.35	0.35	0.25	0.25	-	0.25	0.25
D22	0.4	0.4	0.35	0.35	0.25	-	0.25
D22	0.4	0.4	0.35	0.35	0.25	0.25	-

표 7. NTOT 매트릭스

	A01	A01	B02	B02	C20	D22	D22
A01	-	0.6	0.25	0.45	0.5	0.6	0.7
A01	0.8	-	0.9	0.7	0.5	0.4	0.45
B02	0.5	0.25	-	0.7	0.8	0.9	0.45
B02	0.30	0.40	0.45	-	0.25	0.7	0.8
C20	0.9	0.35	0.6	0.35	-	0.3	0.35
D22	0.45	0.7	0.8	0.35	0.45	-	0.5
D22	0.4	0.4	0.5	0.6	0.7	0.8	-

문제로 알려져서 기존의 해법으로는 풀어질 수 없었던 수리 모델은 장착 노드의 수가 작을 경우 기존의 해법을 이용하여 최적해를 산출할 수 있는 절차를 하드웨어의 특성을 고려하여 제시하였다. 장착 노드의 수가 적고 $NTOT$ 의 모든 값이 상수인 경우에는 현재 알고리즘을 이용하여 최적해를 구할 수 있다. 포지셔닝 유닛의 사용은 $NTOT$ 의 값을 변수로 만들어 주기 때문에 최적해를 구하기 위하여 모든 경우의 수들을 고려한 원초적인 TSP문제를 풀어야 한다.

8. 결론

본 연구에서는 로타리형 표면실장기를 운영을 위한 최적 운영 알고리즘을 다루었다. 현재까지 개발된 대부분의 알고리즘들의 특징은 X-Y 테이블의 이동거리와 릴 축의 이동만을 고려하여 부품 카세트의 위치와 장착 순서를 결정해 주었다. 본 연구에서 개발된 알고리즘은 시스템의 운영 효율에 영향을 주는 모든 인자들을 선정하여 알고리즘의 개발에 반영하였다. 그리고 QAP의 해결을 위해 기존방법과는 달리 장착을 위해 고려할 모든 변수들을 반영하여 카세트의 위치를 결정한 후, 결정된 카세트의 위치를 고려하여 부품의 장착 순서를 결정하는 알고리즘을 제시하였다. 제시된 알고리즘을 장착 부품수가 많은 경우에 적용할 때 최적해를 구하는 데에는 컴퓨터 수행 시간이 많이 걸릴 수도 있다. 따라서 이에 대한 적절한 경험적 알고리즘의 개발이 요구된다고 할 수 있다. 또한 앞으로 PCB 작업의 분배 및 생산계획 등을 위한 PCB 조립라인 운영을 위한 소프트웨어의 개발이 더욱 요청될 것으로 사료되는 바, 본 연구를 통해서 개발된 여러 기법들, 예를 들면 시스템 분석 방법 및 절차, 변수들의 매트릭스 표현 기법, 변수의 통합을 통한 모델의 단순화 등 제시된 여러가지의 기법들이 요긴하게 사용될 수 있을 것으로 생각된다. 또한 제시된 QAP에 대한 알고리즘은 제조, 생산관리 등을 포함한 다양한 분야에서 사용될 수 있을 것으로 사료된다.

참고 문헌

- [1] 상태규, "SANYO의 고정도 표면실장기 술," 전자부품, 5월, 218-221, 1990.
- [2] "Chip형 전자부품 자동장착장치 DCM - 70L 매뉴얼", 대우전자(주) 생산기술연구소, 1993.
- [3] 유태연, "성공적인 SMT공정설치를 위한 전략," 전자부품, 5월, 148-159, 1990.
- [4] 이상복, "PCB 최적삽입경로 및 부품함 위치 배정 연구," 서울대학교 산업공학과 박사학위논문, 1993.
- [5] 이영해, 김원구, 김정, "Path Optimization for Chip Mounting Devices", Proceedings of 2nd International Meeting, 495-498, DSI, Seoul, 1993.
- [6] Cerny, V., "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", J. of Optim. Theory and Appl., 45, 41-51, 1985.
- [7] Chan, D. and D. Mercier, "IC Insertion: An Application of the Traveling Salesman Problem", International Journal of Production Research, Vol. 27, No. 10, 1837-1841, 1989.
- [8] Glover, F., "Heuristic for Integer Programming Using Surrogate Constraint", Dec. Sci., 8, 156-166, 1977.
- [9] Leipala, T. and O. Nevalainen, "Optimization of the Movements of a Component Placement Machine", European Journal of Operational Research, 38, 167-177, 1989.
- [10] Vellanki, M. and C. H. Dagli, "Artificial

- Neural Network Approach in Printed Circuit Board Assembly”, *Journal of Intelligent Manufacturing*, Vol. 4, No. 1, 109-119, 1993.
- [11] Walas, Robert A. and Ronald G. Askin “An Algorithm for NC Turret Punch Press Tool Location and Hit Sequencing”, *IIE Transactions*, Vol. 16, No. 3, 280-287, 1984.
- [12] Sohn, Jin-Hyeon and Sung-Soo Park, “Efficient Operation of a Surface Mounting Machine with a Multihead Turret”, *Technical Report*, Dept. of I.E., KAIST, 1994.
- [13] 이영해, 김정, 김덕한, 경규형, “칩마운터의 고생산성을 위한 최적화에 관한 연구”, 보고서, 한양대 생산공학연구소, 1994.