

# 인쇄회로기판 조립공정에 있어서 여러 개의 헤드를 가진 부품자동삽입기의 효율적 운용\*

## Efficient Operation of a Component Placement Machine with a Multi-head in a PCB Assembly Process\*

박성수\*\*, 손진현\*\*

Sungsoo Park\*\*, Jinhyeon Sohn\*\*

### Abstract

Efficient operation of a component placement machine with a multi-head in a PCB assembly process is considered. The problem consists of two subproblems, which are the optimal allocation of the component tape reels in the rack and the optimal mounting sequence of the components on a printed circuit board. We analyze the problem and propose a heuristic algorithm to solve it. The heuristic can be used for the one head case too. Computational experiences on some real world problems show the effectiveness of the heuristic in terms of speed and solution quality.

### 1. 서론

전기, 전자 회로 부품의 소형화 및 생산자동화로 인하여 인쇄회로기판(PCB) 조립공정에서 여러 가지 형태의 부품자동삽입기의 사용이 일반화되어 있다. 이러한 자동삽입기의 사용에 있어서 부품테이프 릴의 장착위치 및

PCB상의 부품 삽입순서에 따라 기판의 조립 생산 시간이 큰 영향을 받게 되므로, 릴의 장착위치 및 부품 삽입순서의 결정이 중요한 연구과제로 대두되고 있다([1],[9],[12]). 특히, 최근에는 릴을 장착하는 랙(rack)이 크며 여러 개의 헤드를 가지고 있는 자삽기의 사용이 증가하고, 제품생산에 있어서도 다품종

\* 이 논문은 1993년도 학술진흥재단의 공모과제 연구비에 의하여 연구되었음

\*\* 한국과학기술원 산업공학과

생산의 시대에 접어들게 됨에 따라 생산되는 기판의 형태가 자주 바뀌게 됨으로써, 짧은 시간내에 릴의 장착위치 및 부품 삽입순서에 대한 합리적인 의사결정을 내릴 수 있는 효과적인 알고리즘의 개발이 더욱 필요하게 되었다.

본 연구에서는 여러 개의 헤드를 가진 부품자동삽입기를 사용한 표면탑재방식(surface mounting method)의 인쇄회로기판 조립공정을 대상으로 하나의 기판 조립에 필요한 작업시간을 최소화하는 문제를 고려하고 있다. 일반적으로, 표면탑재방식의 기판 조립공정은 두 단계를 거치게 되는데, 첫단계는 저항과 축전기 같은 수백 개의 작은 부품들을 삽입하는 것이고 두번째 단계는 IC 칩과 같은 큰 부품들을 삽입하는 것이다. 일반적으로 첫단계가 조립공정의 bottle neck이므로 본 연구는 첫단계에서의 작업시간 단축을 고려한다. 본 연구의 대상이 되는 첫단계에서 사용되는 자삽기의 개략적인 구조는 그림 1과 같다. 이러한 자삽기의 효율적인 운용을 위해서는 두가지의 면을 동시에 고려해야 하는데, 하나는 부품테이프 릴의 최적 배치이고 또 하나는 부품 삽입순서의 최적화이다.

하나의 헤드를 가진 문제는 Leipälä and Nevalainen[9]에 의해서 연구되었는데, 랙에 장착된 릴의 위치가 고정되어 있는 경우 부품 삽입순서를 결정하는 문제는 외관원문제(traveling salesman problem)가 되고, 부품 삽입순서가 고정되어 있는 경우 릴의 배치문제는 이차배정문제(quadratic assignment problem)가 되므로, 그들은 이 문제를 외관원문제와 이차배정문제가 결합되어 있는 것으로 보았다. 여기에서 두 하위 문제는 NP-hard

문제이므로([2],[8]), 그들은 현실적인 실용성을 고려하여 휴리스틱 알고리즘을 개발하였다.

여러 개의 헤드를 가진 자삽기의 경우에 있어서, 부품 삽입순서가 고정되어 있는 경우 릴의 배치문제는 앞과 동일한 것으로 볼 수 있지만, 릴의 위치가 고정되었다 하여도 전체적인 삽입 순서가 정해지지 않은 상황에서는 두 부품 사이의 거리가 고정되어 있지 않게 된다. 따라서, 본 연구에서는 두 부품간의 거리를 나타내는 새로운 함수를 개발하기보다는 테이블과 랙의 움직임을 평활화시키는 방법을 이용하였다. 이것은 앞의 논문[9]에서 사용된 거리함수의 원용으로 볼 수 있는데, 그 타당성은 최종해의 우수성에 의해 입증될 것이다. 이 경우, 본 연구의 대상이 되는 문제도 외관원문제와 이차배정문제가 결합된 것으로 볼 수 있는데, 연구 대상이 되는 문제의 크기(부품수 300이상, 릴의 수 50이상)를 고려하여 본 연구의 목적을 효과적이면서도 실제 현장에서 현실적으로 사용가능한 알고리즘을 개발하는 데 두었다. 그것을 위하여 [9]에서 수행된 하나의 헤드를 가진 자삽기에 대한 연구를 여러 개의 헤드를 가진 경우로 확장하고, 더욱 많은 부품을 필요로 하는 기판에도 합리적인 시간 내에 최종해를 얻을 수 있는 더욱 효과적이고 효율적인 알고리즘을 개발하였다.

앞서의 연구[9]에서는 먼저 임의의 릴 배치하에서 최장지점 삽입방법(farthest insertion method)에 의해 초기해를 구하고, 다음에 릴의 위치를 두개씩 맞바꾸어 가면서 매번 완전히 새로운 삽입순서를 구하여 앞서의 해와 비교하여 점차 나은 해를 구해나간다.

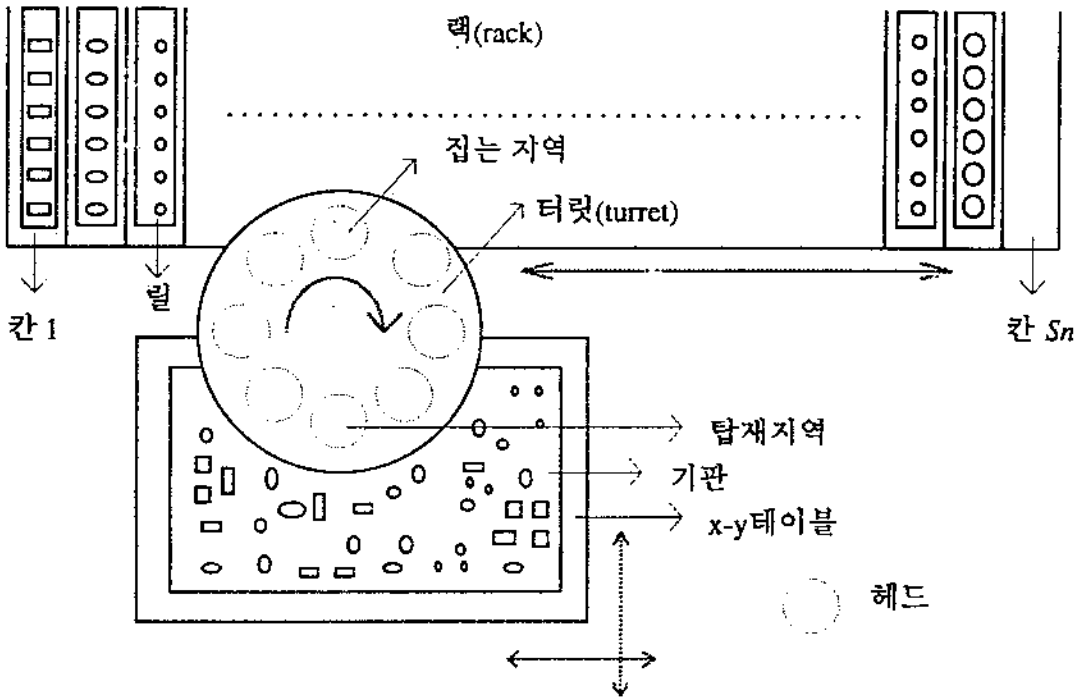


그림 1. 대상자삽입기의 개요

이 경우, 그 실행 방법은 간단하나 릴의 수와 부품의 수가 클 경우 최종해를 찾는 데 너무 많은 시간이 소요되고 매번 새로운 릴의 위치마다 새롭게 구해진 삽입순서가 휴리스틱에 의해 구해지므로 삽입순서의 개선의 여지가 있게 된다. 본 연구에서 제시하는 해결방안은 이와는 달리 초기의 릴 배치하에서 최초의 삽입순서를 정하고, 릴의 위치 변화에 따라 이미 구한 삽입순서의 일부분을 변화시켜 점차 개선된 해를 얻게 된다. 이 방법에 의해 최종해의 값이 향상되었을 뿐만 아니라, 최종해를 얻는 데 소요되는 시간이 상당히 단축되어 이전보다 더 큰 현실적인 문제에 대하여 합리적인 시간내에 최종해를 얻을 수 있었다.

## 2. 문제 설명

본 연구의 대상이 되는 자삽기는 앞의 그림 1과같이 여러 개의 헤드를 가진 터릿(turret), 부품테이프 릴이 장착되는 랙(rack), 기관이 고정되는 작업테이블 그리고 제어장치로 이루어져있다. 기계는 제어장치에 입력되는 프로그램에 따라 동작된다. 기계의 동작과 문제의 모형화를 위한 기호들과 가정은 다음과 같다. 다음에서 한 단위 공정은 부품삽입동작의 한 사이클(cycle)을 의미한다.

1. 터릿에서 헤드의 수는  $H$ (대상 모델에서는 8이고 일반적으로 여러 개의 헤드인 경우 짝수)이다. 헤드는 집는 지역(picking position)에서 하나의 부품을 잡고 그것을 탑재지역(mounting position)에서 기관위에 놓게 된다.

터릿은 한 공정마다 한 구획(segment, 이웃한 헤드사이의 각도)만큼 회전한다. 따라서 집는 지역에서 잡힌 부품은  $H/2$  공정후에 기판에 삽입된다.

2. 랙에서 칸(section)의 수는  $S_n$ (대상 모델에서는 100)이다. 부품테이프 릴은 랙의 첫 번째 칸부터 연속적으로 배치되고, 릴의 번호는 배치된 칸의 번호에 따라 부여된다. 각 릴들은 서로 다른 종류의 부품들을 가지고 있다. 랙은 헤드가 원하는 부품이 집는 지역에 위치하도록 좌우로 이동한다.

3. 작업테이블은  $x, y$  방향으로 동시에 독립적으로 움직인다. 기판상의 부품이 놓일 위치는 테이블의 이동에 의하여 탑재지역에 놓이게 된다.

4. 한 공정은 다음의 동작들로 이루어져 있다.

(a) 터릿의 회전, 랙의 이동, 테이블의 이동. 이들은 서로 독립적으로 움직인다.  $\mu$ 를 터릿의 한 구획 이동시간이라고 하고  $Tr$ 을 랙의 이동시간,  $Txy$ 를 테이블의 이동시간이라고 할 경우 위의 동작이 완료되기 위한 시간은  $\tau = \max\{\mu, Tr, Txy\}$ 로 구할 수 있다. 일반적으로  $\mu$ 는 고정되어 있다.

(b) 위의 동작 후에 헤드가 부품을 집는 것과 탑재하는 것이 동시에 이루어진다. 처음의  $H/2$  공정은 집는 동작만을 마지막  $H/2$ 의 공정은 탑재동작만을 하게 된다.

하나의 기판이 조립된 후 조립된 기판을 다음 단계로 이동하고 새로운 기판을 테이블 위로 올려놓는 동안 테이블과 랙은 원위치로 돌아갈 수 있는 것으로 본다. 하나의 기판에서 부품이 삽입될 지점의 총수(또는 삽입될 부품의 총수)를  $N$ , 서로 다른 종류의 부품수

를  $R$ 이라하고,  $(x_i, y_i, r_i)$ 를 기판상에 지점  $i$ (부품  $i$ 가 삽입될 지점)( $1 \leq i \leq N$ )의 좌표( $x_i, y_i$ )와 지점  $i$ 에 탑재될 부품(부품  $i$ )이 감겨 있는 릴의 번호  $r_i$ ( $1 \leq r_i \leq R$ )를 나타내는 것으로 하자.

기판의 교체 시간을 무시할 경우 우리의 작업은 주어진  $N$ 개의 부품을 삽입하는 데 필요한 시간을 최소화하는 것이 된다. 부품을 집고 삽입하는 동작에 필요한 시간은 고정되어 있으므로 우리는 릴의 배치와 삽입 순서에 따라 변동되는 4-(a)의 동작시간만을 고려한다.

문제의 분석을 위하여 터릿의 한 구획 회전시간( $\mu$ )을 단위시간으로 보고, 그 단위시간 동안 움직일 수 있는 테이블의  $x$  또는  $y$  방향의 거리를  $w$ , 랙이 이동할 수 있는 칸수를  $v$ 라 놓는다. 일반적으로  $v$ 는 1보다 작지 않다. 대상 모델에서  $w$ 는 25mm 이고  $v$ 는 1이다.

### 3. 문제 분석

#### 3.1 목적함수

릴의 위치가 고정되고  $N$ 개의 부품의 삽입 순서가 결정되었다면, 위 4-(a) 동작의  $k$  번째 공정시간  $\tau(k)$ 는 다음과 같이 계산될 수 있다.

$$\tau(k) = \begin{cases} \max\{1, Tr(i_k)\} & \text{for } 1 \leq k \leq H/2 & (1) \\ \max\{1, Tr(i_k), Txy(i_k-H/2)\} & \text{for } H/2 < k < N & (2) \\ \max\{1, Txy(i_k-H/2)\} & \text{for } N \leq k < N+H/2 & (3) \end{cases}$$

여기에서

$$Tr(i_k) = |r_{i_k} - r_{i_{k+1}}| / v & (4)$$

$$T_{xy}(i_k) = \max\{|x_{i_k} - x_{i_{k-1}}|, |y_{i_k} - y_{i_{k+1}}|\} / w \quad (5)$$

$i_k$  : 주어진 삽입순서에서  $k$ 번째로 삽입되는 부품의 번호(혹은 지점의 번호)

위에서, 1은 터릿이 한 구획 회전하는데 필요한 단위시간을 의미하고,  $Tr(i_k)$ 는 주어진 삽입순서에서  $k$ 번째 부품이 집힌 뒤에  $k+1$ 번째 부품이 감겨 있는 릴이 집는 지역에 위치하기 위한 랙의 이동시간을,  $Txy(i_k)$ 는  $k$ 번째 부품이 삽입된 뒤에  $k+1$ 번째 부품이 삽입될 지점이 탑재지역으로 오기위한 테이블의 이동시간을 각각 나타낸다. 처음의  $H/2$  공정은 터릿의 회전과 랙의 이동만을 고려하며, 마지막  $H/2$  공정은 터릿의 회전과 테이블의 이동만을 고려한다. 여러 개의 헤드를 가진 자삽기의 효율적인 운용을 위한 문제에서 본래의 목적함수는 다음의 값을 최소화하는 것이다.

$$T = \sum_{k=1}^{(n-1)+H/2} \tau(k) \quad (6)$$

릴의 위치가 고정되고 완전한 삽입순서가 주어졌다면,  $T$ 의 값은 위의 식들에 의해서 계산될 수 있다. 그러나, 여러 개의 헤드를 가진 자삽기의 경우 식 (2)에서 볼 수 있는 바와 같이 랙과 테이블이 서로 다른 부품을 위하여 움직이게 되므로, 비록 릴의 위치가 고정되어도 부품의 삽입순서가 완전히 주어지지 않는다면, 임의의 두 부품간의 거리를 정의한다는 것이 불가능하게 된다. 이러한 특성이 기존의 외판원문제를 위한 효율적인 알고리즘들을 본 문제에 그대로 적용하기 어렵게 만든다. 따라서, 본 문제를 그대로 풀기보다는 랙과 테이블의 이동시간을 평활화시키는데 초점을 맞춰 하나의 헤드를 가진 경우를

원용하여 문제의 해결을 시도한다. 이 경우, 기판 상의 두 지점  $i, j$ 의 거리(시간)함수로 다음의 식을 사용한다.

$$Dr(i,j) = \max\{1, |r_i - r_j| / w\} \quad (7)$$

$$Dxy(i,j) = \max\{1, |x_i - x_j| / w, |y_i - y_j| / w\} \quad (8)$$

$$d(i,j) = \max\{Dr(i,j), Dxy(i,j)\} \quad (9)$$

(7)은 터릿의 회전시간과  $i$ 지점에 탑재되는 부품이 감겨있는 릴에서  $j$ 지점에 탑재되는 부품이 감겨있는 릴로 이동하기 위한 랙의 이동시간을 나타내고, (8)은 터릿의 회전시간과  $i$ 지점에서  $j$ 지점으로 이동하는 데 필요한 테이블의 이동시간을 나타낸다.

이제부터 식 (9)의  $d(i,j)$ 를 릴의 위치가 결정되었을 경우 두 지점  $i, j$ 의 거리함수로 사용할 것이다. 이러한 식 (9)는 본질적으로 Leipälä and Nevalainen[9]에 의해 사용된 거리함수와 동일하다. 그들은 두 지점  $i, j$ 의 거리함수로  $d(i,j) = \max\{1, |x_i - x_j| / w, |y_i - y_j| / w, |r_i - r_j| / w\}$ 를 사용하였다. 본 연구에서는 랙과 테이블의 움직임을 독립적으로 나타내기 위하여, 앞의 식들을 사용할 것이다. 또한, 실제 프로그램 상에 있어서도 랙의 움직임을 독립시키는 이점이 있다. 마지막  $H/2$  공정과 기판이 바뀌는 사이에 랙은 처음 삽입될 부품의 릴 위치로 돌아가고, 처음  $H/2$  공정 동안 테이블은 첫부품이 삽입될 지점으로 돌아간다고 가정한다. 이러한 추가적인 가정들은 릴이 고정된 경우에 남은 문제를 대칭적인 (symmetric) 외판원문제로 만든다. 이제 새로운 목적함수는 다음의 값을 최소화하는 것이다.

$$D = \sum_{k=1}^{N-1} d(i_k, i_{k+1}) \quad (10)$$

여기에서,  $i_k$ 는 주어진 삽입순서에서  $k$ 번째로 삽입되는 지점의 번호이다. 본 논문에서 사용하고자 하는 방법은  $D$  값을 최소화시킴으로써  $T$  값을 감소시키고자 하는 것이다. 추가적으로 식 (7)~(9)를 가지고 다음과 같은 성질들을 얻을 수 있다.

<성질 1>

주어진 삽입 순서에서 모든  $D_{xy}(j_k, i_{k+1})$  가 어떤  $D_r(i_k, i_{k+1})$  보다 작지 않다면, 처음의  $H/2$  의 값을 제외한  $T$  의 값은  $D$  의 값과 동일하다.

<성질 2>

하나의 기관을 조립하는 데 필요한 부품이 모두 다른 것이고  $v \geq 1$  이면,  $D$  를 최소화 하는 문제는 외판원문제로 볼 수 있다.

만약,  $v \geq 1$  이고 첫번째 밑부터 필요한 부품을 모두 삽입한 다음 옆의 밑로 이동한다면, 성질 1의 조건을 만족시킬 수 있다. 사용빈도가 한번인 부품들의 밑 배치와 삽입순서를 정하는 데 성질 2를 사용하였다. 또한, 성질 1과 2로 부터 성질 2의 가정이 성립한다면,  $T$ 를 최소화하는 문제가 외판원문제가 됨을 알 수 있다.

3.2 문제의 정식화

$p_i$ 를 입력데이터에서 주어지는 기관상의  $i$  지점에 삽입될 부품의 종류라고 하자( $1 \leq p_i \leq R$ ). 기관상에 가상의(dummy) 지점  $0=(0,0)$ 을 도입한다. 가상의 지점에는 아무런 부품도 삽

입되지 않는다. 이전변수  $X_{ij}(0 \leq i, j \leq N)$ 와  $Y_{pr}(1 \leq p, r \leq R)$ 을 도입하여 식(9)의  $D$  값을 최소화시키는 문제를 다음과 같이 수식적으로 정식화시킬 수 있다.

$$\text{Minimize } \sum_{s=1}^R \sum_{i=1}^N \sum_{j=1}^N C_{ijst} X_{ij} Y_{p,s} Y_{p,t} \quad (11)$$

subject to

$$\sum_{i=0, i \neq j}^N X_{ij} = 1 \quad j = 1, \dots, N. \quad (12)$$

$$\sum_{j=0, j \neq i}^N X_{ij} = 1 \quad i = 1, \dots, N. \quad (13)$$

$$u_i - u_j + N \cdot X_{ij} + (N-2)X_{ji} \leq N-1 \quad (14)$$

$$i, j = 1, \dots, N, i \neq j$$

$$0 \leq u_i \leq N-1 \quad i = 1, \dots, N. \quad (15)$$

$$X_{ij} = 0 \text{ or } 1 \quad i, j = 1, \dots, N. \quad (16)$$

$$\sum_{p=1}^R Y_{pr} = 1 \quad r = 1, \dots, R. \quad (17)$$

$$\sum_{r=1}^R Y_{pr} = 1 \quad p = 1, \dots, R. \quad (18)$$

$$Y_{pr} = 0 \text{ or } 1 \quad p, r = 1, \dots, R. \quad (19)$$

여기에서

$C_{ijst} = \max\{1, |x_i - x_j|/w, |y_i - y_j|/w, |s - t|/v\}$ 는 부품  $p_i$ 가 칸(section)  $s$ 에 있고 부품  $p_j$ 가 칸  $t$ 에 있을 경우 기관상의 두 지점  $i, j$ 의 거리를 나타냄.

$X_{ij} = 1$  ; 지점  $i$ 로부터 지점  $j$ 로의 이동이 있음을 나타냄.

$Y_{pr} = Y_{pr} = 1$ ; 지점  $i$ 에서 필요한  $p$  타입의 부품이 칸  $r$ 에 있음을 나타냄.

만약,  $r_i = s$  이고  $r_j = t$ 일 경우  $C_{ijst}$ 의 값은

$d(i,j)$ 와 동일하므로 목적함수 식은  $D$ 를 최소화시키는 것이다. 제약식 (12)~(16)은 외판원문제의 제약식들이다([8]). 제약식 (12)와 (13)은 모든 지점들을 정확히 한번만 거치게 한다. 제약식 (14)와 (15)는 Desrochers and Laporte[5]에 의해 주어진 subtour를 제거하기 위한 강화된 식이다. 제약식 (17), (18), (19)는 이차배정문제의 제약식들이다([2]). 그것들은 각각의 부품타입들이 정확히 랙의 한 칸에 할당되어야 함을 의미한다. 위의 식들은 [9]에 의해 주어진 식과 유사하다. 여기에서는 모든 지점  $i$ 에 대하여  $C_{iost} = C_{iost} = 0$ 를 가정함으로써, 외판원문제가 완전히 대칭적이 된다. 하나의 해가 얻어지면, 첫번째 삽입위치는  $X_{0k} = 1$ 이 되는  $k$ 가 되고 가상의 지점 0를 제거한다. 위에서, 릴의 위치가 고정되었을 경우, 남은 문제는 대칭적인 외판원문제가 되고, 부품의 삽입순서가 고정되어 있을 경우, 최적의 릴 배치를 찾는 문제는 이차배정문제가 된다. 이 두 하위 문제가 NP-hard임이([2],[8]) 앞의 문제가 NP-hard임을 보여준다. 본 연구의 대상이 되는 실제문제의 크기(부품수 300이상, 릴의 수 50이상)가 크므로, 현실적인 사용가능성을 고려하여 최적해를 구하는 것보다 휴리스틱 알고리즘을 개발하였다.

#### 4. 알고리즘

알고리즘은 두 단계로 구성되어있다. 첫단계에서는 초기 릴의 위치와 부품의 삽입순서를 결정하게 된다. 두번째 단계에서는 릴의 위치를 두개씩 맞바꾸어 가면서 삽입순서를 변경시켜 더 나은 해를 찾아 나가는 것이다.

전반적인 전략은 Leipälä and Nevalainen[9]와 동일하지만, 본 연구에서는 초기 릴의 위치를 정하는 다른 방법을 제시하고 개선된 해를 구성해 나가는 데 있어서 새로운 방법을 사용한다. [9]에서는 릴의 위치를 교환할 때마다 매번 완전히 새로운 삽입순서를 구성하지만, 여기에서는, 교환 대상이 되는 릴과 관련이 없는 부품들의 삽입순서를 그대로 유지한 채, 위치변경이 고려되는 릴의 부품을 요하는 지점들의 삽입순서의 변경만을 고려한다. 이 방법은 다음 절에서 보이는 바와 같이 계산시간을 상당히 단축시킬 뿐만 아니라 최종해의 값을 향상시키고 있다.

먼저, 알고리즘을 실제 프로그램함에 있어서, 입력데이터를 부품의 종류별로 그룹화시키고 동일한 부품을 요하는 지점들을 한 묶음으로 처리하였고, 릴을 나타내기 위하여 배열을 사용하였다. 각각의 원소 릴 $[i]$ 는 부품종류와 그 부품의 사용빈도 그리고 그 부품이 탑재될 지점들에 관한 정보(지점번호,  $x$ ,  $y$  좌표, 삽입순서)를 가지고 있다. 또한, 삽입순서는 위치명과  $x$ ,  $y$  좌표 그리고 릴 번호를 가지고 있는 노드들의 리스트로 나타내었다.

첫단계에서, 각 릴의 부품 사용빈도(하나의 기관에서)에 따라 증가하는 순서로 릴들을 분류(sorting)한다.  $I$ 를 사용빈도가 1인 릴들의 집합이라 하고,  $J$ 를 사용빈도가 1보다 큰 릴들의 집합이라고 하자.  $J$ 의 원소들은 사용빈도가 증가되는 순서대로 나열되어 있다고 가정한다. 먼저,  $I$ 에 속하는 릴에 들어 있는 지점들과 가상지점 0만을 가지고 식 (8)을 시간함수로 사용하여 여행경로(tour)를 구성한다.  $I$ 에 속하는 릴에 들어 있는 모든 지

점  $i$ 에 대하여  $D_{xy}(i,0) = 0$ 으로 놓는다. 이 단계에서  $D_{xy}(0,i)$ 는 원점에서 첫번째 삽입지점을 찾기 위해 실제거리로 계산한다. 선행된 테스트에서 이러한 조정이 조금 더 나은 초기해를 제공해 주었다. [6]에서 외판원문제의 여행경로 구성에 있어서 임의삽입방법(arbitrary insertion heuristic)이 상당히 좋은 결과를 보여주고 있으므로, 본 알고리즘에 그 방법을 부분적으로 수용하였다. 만약 지점  $k$ 가 두 지점  $i$ 와  $j$  사이에 삽입된다면, 시간 증가분은  $D_{xy}(i,k) + D_{xy}(k,j) - D_{xy}(i,j)$ 로 계산된다.  $I$ 에서 여행경로에 포함되어 있지 않은 지점을 임의로 선택하고 그것을 시간함수  $D_{xy}$ 를 가지고 시간 증가분이 가장 작은 위치에 삽입한다. 하나의 완전한 여행경로를 구성한 뒤에  $I$ 에 속하는 릴들을 각 릴에 속하는 지점들에 의해서 구성된 여행경로에 의해 주어진 순서대로 첫번째 칸부터 차례로 랙에 할당한다.

$I$ 에 속한 릴의 배치를 위의 방법에 의해 마친 뒤  $J$ 에 속한 릴들을 랙에 할당한다. 먼저,  $J$ 에서 마지막 원소를 골라 그 릴을 랙의 비어 있는 칸 중에서 맨 앞에 할당하고  $J$ 에서 제거한다. 그 다음,  $J$ 에서 앞의 두 원소를 골라 그 릴들을 그 다음 칸들에 할당하고  $J$ 에서 제거한다. 이 과정을  $J$ 가 빌 때까지 반복한다. 이렇게  $J$ 의 릴들을 할당하는 방법은 현재 모 회사에서 실제로 릴의 배치에 사용되고 있는 방법이다. 선행 테스트에서 이러한 릴배치하에서 좋은 초기해가 제공되었다. 그러나, 현재 사용되고 있는 방법은 삽입순서의 결정이 제시된 방법과 다르고 해를 개선하는 과정이 들어 있지 않다. 그 결과의 차이는 다음의 적용사례에서 설명하겠다.

위와 같이 초기 릴의 배치가 결정된 후에 모든 지점(혹은 부품)의 삽입순서를 결정한다. 삽입순서는 가상의 지점 0에서 시작하여 0에서 끝나는 것으로 한다. 앞에서  $I$ 에 속한 릴의 지점들과 가상지점 0을 가지고 얻은 부분적인 삽입순서에  $J$ 에 속한 릴의 지점들을 가지고 점차 완전한 삽입순서를 구성하게 된다. 완전한 삽입순서를 결정하기 위하여 그에 속하는 지점들이 현재의 삽입순서에 들어 있지 않은 릴 중 가장 작은 번호를 가진 것을 선택한다. 그리고 그 릴의 부품이 탑재될 지점들 중에서 아무 곳이나 선택하여, 그 지점을 현재의 부분적인 삽입순서에 넣는다. 그 삽입위치는 시간 증가분이 가장 작은 위치를 찾는다. 여기에서, 지점  $k$ 가 두 지점  $i$ 와  $j$  사이에 삽입된다면, 시간 증가분은  $d(i,k) + d(k,j) - d(i,j)$ 의 값이 된다. 이 방법을 선택된 릴의 모든 지점들이 삽입될 때까지 계속해 나간다. 선택된 릴의 모든 지점들이 삽입되었다면 다음 번호의 릴을 선택하여 앞의 방법대로 계속하여 완전한 삽입순서를 구성한다.

두번째 단계에서는 릴의 위치를 바꾸는 것과 삽입순서의 변화가 같이 고려된다. 릴의 위치변경 방법으로는 이차배정문제의 해결을 위하여 Heider[7]에 의해 제시된 한 쌍씩 교환하는 방법(simplified pair-exchange method)을 사용한다. [9]에서도 Heider의 방법을 사용했는데 거기에서는 매 릴 교환 시마다 완전히 새로운 삽입순서가 결정되게 되나, 여기에서는 현재 주어진 삽입순서를 재사용하게 된다. 릴  $i$ 와 릴  $j$ 의 교환이 고려될 때, 그 릴의 부품이 탑재되는 지점들이 현재의 삽입순서에서 제거되고, 그 지점들의 릴 번호를 바꾼 후에 앞서와 같은 최소시간증가 방법에



의해 그 지점들을 현재의 삽입순서에 재삽입하게 된다. 그 결과 총 시간이 감소된다면 실제로 릴  $i$ 와 릴  $j$ 의 위치를 바꾸게 된다. 그러면, 우리는 새로운 개선된 해를 갖게 된 것이고 다음의 릴 교환으로 나아가게 된다. 해를 개선하는 과정을 좀 더 체계적으로 나타내면 다음과 같다.

#### 〈개선 과정〉

현재의 삽입순서  $S$ 는 위의 첫단계에서 얻어진다.

(Cycle) Do

초기화 :  $k=1, gain = 0$  으로 놓는다.

Step 0.  $k = k+1$

Step 1. (제거)

$S$ 에서 릴  $k$ 와 릴  $k$ 의 부품을 필요로 하는 지점들을 제거한다.

이 과정에서 총시간감소분을 계산한다.

Step 2. (재삽입)

릴  $k$ 와 릴  $k$ 의 릴번호를 바꾸고, 그에 속하는 지점들을 시간함수  $d(i,j)$ 를 가지고 최소시간증가 방법에 의해 재삽입한다. 이 과정에서 총시간증가분을 계산한다.

이렇게 얻어진 새로운 삽입순서를  $S'$ 이라 하자.

Step 3. (평가)

만약, 릴을 바꾼 결과가 이전보다 좋다면 (시간감소분 > 시간증가분),  $S'$ 을  $S$ 로 놓고,  $gain \leftarrow (gain + \text{순 감소시간})$  으로 놓는다. 그렇지 않으면, 앞의  $S$ 를 재구성한다.

$k$ 를 1만큼 증가시키고  $k \leq R$  인 경우 Step 1 으로 돌아간다.

Step 4.  $k$ 를 1만큼 증가시키고  $k < R$  인 경

우 Step 0으로 돌아간다.

While ( $gain > 0$ )

위의 알고리즘에서  $gain$ 의 값이 미리 지정된 값보다 작아질 때 멈추거나 반복하는 총 횟수를 미리 정해 놓을 수도 있다.

## 5. 적용사례

알고리즘을 평가하기 위하여, 실제 현장에서의 문제와 Leipälä and Nevalainen[9]에 의해 주어진 문제에 대하여 C언어로 작성된 프로그램을 486PC(DXII-66)에서 수행하여 보았다. 예제 1과 2는 현장의 문제고 예제 3은 [9]에서 인용한 문제이다. 표 1은 각 예제의 데이터에 대한 개괄적인 자료이다.

표 1. 데이터의 명세

	예제 1	예제 2	예제 3
기판의 크기(0.01mm)	(21000,18000)	(20850,16850)	
삽입될 부품의 수(N)	327	583	90
사용되는 릴의 수(R)	46	99	37
사용빈도가 인 릴의 수	10	27	18
동일 부품의 최대 사용빈도	30	33	13

표 2는  $D$ 값을 나타낸다. A행과 B행에서 괄호 속의 값은 주어진 해에서  $H=8$ 일 때 계산된  $T$ 의 값을 나타낸다. 표 2에서  $O$ 는 제시된 알고리즘을 나타내고  $L\&N$ 은 [9]의 알고리즘을 약간 변경시킨 것이다. [9]에서는 모든  $i \in N$ 에 대하여  $d(0,i)$ 를 실제 거리로 계산하였고  $d(i,0)=0$ 으로 놓았다. 그리고, 마지막에 삽입되는 지점을 임의로 고정시키고 최장삽입방법(farthest insertion heuristic [6])으로 완전한 삽입순서를 결정하였다. [9]에서는 여행경로의 마지막 지점을 변경시켜 가

면서 얻은 여러 해 중에서 가장 좋은 것을 선택한다. [9]의 알고리즘을 실행하는 데 있어서, 임의의 마지막 지점마다 해를 얻는 것이 너무 시간이 많이 걸리므로 약간의 조정을 통하여 마지막 지점을 고정시키지 않았다. 최장삽입방법에 있어서 현재 여행경로에 들어 있지 않은 각 지점  $i$ 에서 여행경로를 구성하고 있는 지점들에 대한 거리를 계산하고, 그 값중 가장 작은 값이 지점  $i$ 의 현 여행경로에 대한 거리로 취급된다. 그 거리가 가장 큰 값을 갖는 지점이 현 여행경로에 삽입할 지점으로 선택된다. 지점  $i$ 에서 현 여행경로까지의 거리를 계산할 때 그 값이 0인  $d(i,0)$ 를 고려하지 않았다. 그렇지 않으면 모든 지점에서 현 여행경로에 대한 거리는 0이 될 것이므로 거리를 계산하는 의미를 상실한다. 이러한 조정이 마지막 지점을 고정해야만 되는 필요를 완화시킨 것이다. 최종해가 얻어졌을 때,  $D$ 값이 우리의 알고리즘에서 얻어진 값과 정확히 비교되게 하기 위하여, 총시간에서  $d(0,i)$ 의 값을 제거하였다. 또한, 초기의 릴 배치는 제시된 알고리즘에서 얻어진 초기 릴배치와 동일한 것을 사용하였다.

제시된 알고리즘을 실행할 때, 삽입되는 지점들의 삽입가능 위치를 제한하였다. 임의삽입방법에서 지점  $k$ 가 두 지점  $i$ 와  $j$  사이에 삽입될 때, 만약 지점  $i$ 와  $j$ 에서 필요한 부품의 릴이 지점  $k$ 에서 필요한 부품의 릴에서 멀리 떨어져 있다면 락의 이동시간 때문에 시간증가분이 커질 것이다. 따라서 지점  $k$ 의 릴번호와 지점  $i$ 와  $j$ 의 릴번호의 차이가  $5 \cdot v$  이내인 경우만 삽입가능 위치로 제한하였다. 두 알고리즘( $O$ 와  $L\&N$ ) 모두에서,  $gain$ 의 값이 1보다 작을 경우 프로그램을 중단시

켰다. 표 2에서 반복 횟수란 개선과정에서 릴의 교환이 완전히 이루어진 총횟수를 의미한다.

기대했던 바와 같이  $D$ 값을 감소시킴으로써  $T$ 값을 감소시킬 수 있었다. 본래 여러 개의 헤드를 가진 경우 처음에 집는 공정만  $H/2$ 만큼이 있으므로  $T$ 값은  $D$ 값보다 최소한 4만큼 크게 된다.

표 2에서, A행은  $w=25\text{mm}$ 이고  $v=1$ (대상 자삼기의 실 상황)에서 예제 1과 2의 실행결과를 보여준다. 회사에서 사용해왔던 알고리즘에 의하면  $T$ 값은 각각 512.23과 879.99였다. 제시된 알고리즘에 의하여 약 25% 정도의 시간이 감소되었다. 제안한 알고리즘은 최종해를 얻는데 수 분이 소요되었으나,  $L\&N$ 은 예제 1에 대하여 약 1 시간, 예제 2에 대하여 약 41 시간이 소요되었다.

B행은  $w=50\text{mm}$ 이고  $v=2$ (즉 터릿의 회전 시간이 2배인 경우)에서 예제 1과 2의 실행결과를 보여준다. C행은 [9]에서의 조건과 동일한 하나의 헤드인 경우  $w=60\text{mm}$ 이고  $v=4$ 에서의 예제 3의 실행결과를 보여준다. 예제 1과 2에서의  $D$ 의 이상적인 값이 326과 582이고, 예제 3의 하한이 90.21인 것([9])을 고려할 때, 제시된 알고리즘이 상당히 좋은 결과를 가져옴을 알 수 있다.

순수한 외판원문제에서, 만약 임의삽입방법에 의해 초기의 경로를 구성하고 반복적으로 몇 개의 지점을 제거한 뒤 다시 최소시간 증가 방법에 의해 그 지점들을 재삽입하는 경로 구성 알고리즘을 고려한다면 총시간이 어느 정도 감소해 나갈 것이다. 제시된 알고리즘은 이러한 과정을 개선과정에서 실행시킨 것이다. [9]에서는 릴의 교환에 따른 개

표 2. 예제 1, 2, 3의 알고리즘 적용결과

		알고리즘	초기해의 값	반복 횟수	최종해의 값	CPU time (sec)
(A)	예제 1	O	409.78	5	369.86	86
			(429.96)		(379.33)	
			411.09	2	398.59	3,996
	예제 2	L&N	(432.84)		(416.50)	
			756.55	6	655.07	397
			(785.22)		(676.78)	
(B)	예제 1	O	761.78	5	718.30	149,244
			(793.19)		(735.54)	
			334.16	2	326.33	55
	예제 2	L&N	(339.57)		(330.33)	
			331.96	2	327.53	3,945
			(337.24)		(331.53)	
(C)	예제 3	O	595.76	2	582.64	218
			(600.90)		(586.64)	
			597.37	2	590.88	58,998
	예제 3	L&N	(603.51)		(596.83)	
			103.21	2	95.44	6
			102.59	2	96.00	195

선은 고려했지만, 한번 구성된 삽입순서의 개선은 고려하지 않은 것으로 볼 수 있다.

## 6. 결론

본 연구에서는 여러 개의 헤드를 가진 부품자동삽입기의 효율적인 운용을 위한 발견적 해법을 제시하였다. 여러 개의 헤드인 경우, 연속된 부품을 위한 랙과 테이블의 이동이 동시에 일어나지 않게 되는데, 본 연구에서는 하나의 헤드를 가진 경우를 원용하여 그 난점의 극복을 시도하였다. 초기 릴의 배치는 부품의 사용빈도를 고려하여 할당시킨다. 해의 개선과정에서, 릴의 위치교환과 삽입순서의 개선을 동시에 고려하려고 노력하

였다. 실행결과들이 이러한 접근방식이 잘 동작함을 보여준다.

실행결과에서 볼 수 있듯이, 헤드의 움직임이 빠를수록 개선과정의 효과가 증대한다. 하나의 헤드를 가진 경우와 같이 헤드의 움직임이 늦다면, 운용시간을 증가시키지 않고 랙과 테이블이 더 많이 움직일 수 있게 된다. 이것은 해의 값이 삽입순서와 릴의 배치에 덜 영향을 받게 됨을 의미한다. 이러한 상황은 여러 개의 헤드를 가진 자삽기의 경우와 같이 헤드의 움직임이 빨라지게 됨으로써 바뀌게 되었다. 즉, 운용시간을 증가시키지 않는 랙과 테이블의 이동범위가 짧아지게 됨으로써 부품삽입순서와 릴의 위치를 찾는 것이 더욱 중요하게 되었다.

실제 자삽기의 운용에 있어서, 공정의 정확도를 높이기 위하여 작은(혹은 높이가 낮은) 부품부터 먼저 삽입하는 제약을 가하기도 한다. 이 경우에도, 제시된 알고리즘을 약간 조정하여 사용할 수 있다. 부품이 크기별로 그룹화될 경우, 먼저 동일 그룹내의 릴들이 함께 위치하도록 랙에 할당한다. 물론, 그룹의 할당은 크기가 증가되는 순서로 한다. 그리고 삽입순서에 있어서도, 하나의 그룹내의 부품 삽입이 다 끝난 뒤에 다음 그룹의 부품으로 이행한다. 이 경우, 해의 개선과정에 있어서 릴의 교환을 동일 그룹내로 제한한다면, 얻어지는 해는 추가적인 삽입순서의 제한을 위배하지 않게 될 것이다.

#### 참 고 문 헌

- [1] Ball, M.O., and Magazine, M.J., "Sequencing of Insertions In Printed Circuit Board Assembly", *Operations Research*, Vol. 36, No. 2, pp192-201, March-April 1988.
- [2] Burkard, R.E., "Quadratic assignment problems", *European Journal of Operational Research*, Vol. 15, pp283-289, 1984.
- [3] Carmon, T.F., Maimon, O.Z., and Dar-el, E.M., "Group set-up for printed circuit board assembly", *International Journal of Production Research*, Vol 27, pp1795-1810, 1989.
- [4] Cunningham, P., and Brown, J., "A LISP-based heuristic scheduler for automatic insertion in electrics assembly", *International Journal of Production Research*, Vol. 24, pp1395-1408, 1986.
- [5] Desrochers, M., and Laporte, G., "Improvements and extentions to the Miller-Tucker-Zemlin subtour elimination constraints", *Operation Research Letters*, Vol. 10, pp27-36, 1991.
- [6] Golden, B., Bordin, L., Doyle, T., and Stewart, W. Jr., "Approximate traveling salesman algorithms", *Operations Research*, Vol 28, pp694-711, 1980.
- [7] Heider, C.H., "A computational simplified pair-exchange algorithm for the quadratic assignment prblem", Paper No.101, Center for Naval Analyses, Arlinton, Virginia, USA, 1972.
- [8] Laporte, G., "The traveling salesman problem: An overview of exact and approximate algorithms", *European Journal of Operational Research*, Vol. 59, pp231-247, 1992.
- [9] Leipälä, T., and Nevalainen, O., "Optimization of the movements of a component placement machine", *European Journal of Operational Research*, Vol. 38, pp167-177, 1989.
- [10] Lin, S., and Kernighan, B.W., "An effective heuristic algorithm for the traveling-salesman problem", *Operations Research*, Vol. 21, pp498-516, 1973.
- [11] Maimon, O., and Shtub, A., "Grouping methods for printed circuit board assembly", *International Journal of Production Research*, Vol. 38, pp1397-1390, 1991.

- 
- [12] Mettala, E.G., and Egbelu, P.J., "Alternative approaches to sequencing robot moves for PCB assembly", *International*

*Journal of Computer Integrated Manufacturing*, Vol. 2, pp243-256, 1989.