

로터리형 표면실장기 효율적 운용을 위한 알고리즘 및 소프트웨어 개발

이 영 해,* 김 정,** 경 규 형,** 김 덕 한***

Developing Algorithm and Software for Efficient Operation of Rotary Type Surface Mounter

Young-Hae Lee,* Jung Kim,** Kyu-Hyoung Kyoung,** Duck-Han Kim***

ABSTRACT

Surface mounters are used to mount the parts of electronic products on PCB(Printed Circuit Board). Those machines rely on high precision, high productivity, and software for accomodation of use. In this paper, two algorithms and a software which can give the users the accomodation and high productivity for rotary type surface mounters are presented. The algorithms for development of the software consider all factors which affect the effeciency of the operation. And the interface between user and machine for the accomodation of use can be achieved through the developed software.

Key Words : Rotary Type Surface Mounter, Optimal Operation, Mounting Sequence, Cassette Location.

1. 서 론

전자 산업의 발전과 더불어 1980년대 중반 이후부터 NC 및 CNC 기술이 급속하게 발전하였다. 이와 더불어 표면실장기술의 진보로 여러 전자 장비에 사용되는 부품들의 소형화 및 경량화가 이루어지고 있으며 이러한 발전추이는 앞으로 더욱 가속화될 전망이다. 한편 표면실장기술은 다양한 전자 산업의 생산성에 매우 큰

영향을 주는 고급기술이므로 선진국에서는 기술이전을 매우 꺼리고 있으며 국내에서는 개발 여건이 매우 열악하여 제조업체의 대부분은 필요한 하드웨어 및 소프트웨어를 수입에 의존하고 있는 실정이다. 이는 표면 실장기에 사용되는 부품의 국내 수요가 다품종 소량이고 국내 대기업의 외국기기 수입 선호 등으로 중소기업에서는 장비에 사용되는 부품 개발을 기피하고 있기 때문이다.^(3,4) 따라서 이에 대한 대책이 국가적으로 수립되어

* 한양대학교 산업공학과(정회원)

** 한양대학교 대학원

*** 대우자동차

야 할 필요가 있다. 표면실장기술은 하드웨어의 측면과 소프트웨어의 측면으로 크게 나누어 생각할 수 있다. 하드웨어 측면에서 볼 때 표면실장기의 생산성 향상을 위해 고품질의 부품 생산 및 장비의 신뢰성에 관심을 두고 이를 발전시킬 수 있으며 이는 전자공학 및 기계, 제어 공학 등의 진보로 이룩될 수 있다. 표면 실장기 (surface mounter)의 생산성은 장비의 신뢰성 이외에도 운용 방법에 크게 의존한다. 고속, 대량 생산용인 로터리형 표면 실장기 (rotary type surface mounter)를 운용하기 위하여 먼저 부품 카세트의 위치와 장착 순서가 결정되어야 한다. 작업자들은 이제까지 자신의 경험에 의존하여 이러한 운용 방법을 수립한 후 NC 파트 프로그램을 작성한다. 그런데 많은 장착점과 카세트의 위치에 대해서 효율적인 운용 방법을 수립하는 일이 작업자들에게는 매우 고된 일이 될 뿐만 아니라 작업자들의 능력에 따라 표면 실장기의 생산성이 달라질 수 있기 때문에 이의 효율적 운용을 위한 소프트웨어의 개발이 시급히 요청되었다.

Leipala Nevalainen은 단일 헤드를 갖는 메니플레이터의 운영을 위한 부품 피더의 위치를 임의로 배치하고 초기의 장착순서를 결정한 후, 노드 교환 장법을 적용하여 최종 부품 피더의 위치와 장착순서를 결정하는 알고리즘을 제시하였다.⁽⁵⁾ Walas and Askin은 다수의 헤드를 가진 메니플레이터의 운영을 위한 알고리즘을 제시하였는데 표면 실장기의 운영상에서 고려해야 할 인자들이 알고리즘에 충분히 반영되지 않아서 실제 적용에 어려움이 있다.⁽⁶⁾ 이. 와 김.⁽⁷⁾은 시스템 분석을 통해 표면 실장기 운용에 영향을 주는 모든 인자들과 변수들을 결정한 후 이를 매트릭스 형태로 표현하였다. 그리고 시스템의 동작 원리를 이용하여 변수를 통합시킴으로써 한 변수를 이용하여 시스템의 최적 운용을 위한 알고리즘을 개발하였다. 본연구에서는 로터리형 표면 실장기의 사용자들에게 편의성을 제공해 줄 뿐만 아니라 생산성의 향상에 크게 기여 할 수 있는 운용 알고리즘 및 소프트웨어를 개발하였다.

2. 표면 실장기의 구성 및 운영을 위한 고려사항

로터리형 표면 실장기의 종류는 다양하지만 본 연구의 대상인 시스템의 개략도는 Fig. 1과 같다.

로터리형 표면 실장기의 운용을 위한 소프트웨어를 개발하기 위하여 고려해야 할 인자들과 하드웨어에 대

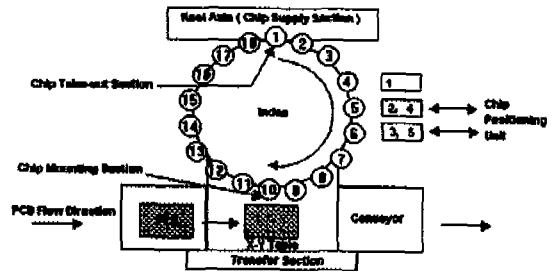


Fig. 1 Schematic diagram of rotary type surface mounter

한 제약 조건들을 설명하면 아래와 같다.

2.1 표면 실장기의 운영에 영향을 주는 인자의 선정

로터리형 표면 실장기는 크게 릴 축 (Reel Axis), 인덱스 테이블 (Index Table) 그리고 XY테이블 (X-Y Table)의 세부분으로 구성되어 있다. 릴 축은 PCB에 장착될 부품을 담고 있는 부품 카세트 배치 장치로써 장착 작업이 수행될 때 수평운동하며 부품을 인덱스 테이블의 헤드에 공급하는 역할을 수행한다. 부품의 크기가 다양하기 때문에 릴 축을 점유하게 될 부품 카세트의 크기도 다양하다. 한편 릴 축의 이동에는 시간을 요하기 때문에 부품 카세트의 크기와 위치는 부품 조립 시간에 영향을 주는 인자가 된다.

인덱스 테이블에는 18개의 헤드가 있다. 테이블이 회전하면서 헤드는 피킹 스테이션 (Picking Station: ①)에서 부품을 피킹한 후, 요구되는 여러 스테이션을 방문한다. 헤드가 모든 스테이션을 방문하면서 각 부품에 요구되는 여러 작업들이 이루어지게 되고 Chip Mounting Station에서 부품은 PCB 위에 장착된다. 인덱스 테이블이 회전할 때의 회전 속도는 각 부품에 따라 다르게 지정될 수 있다. 그리고 각 부품의 포지셔닝 (Positioning: ④, ⑤, ⑥)을 위해 사용되는 유니트 (Unit)도 다를 수 있다. 또한 PCB위에 장착되는 부품의 장착 각도는 각 부품마다 다를 수 있다. 따라서 표면 실장기의 운용을 위해 인덱스 테이블에서 고려해야 할 인자는 인덱스 테이블의 지정 회전 속도, 부품의 장착 각도, 그리고 포지셔닝 유니트이다.

PCB는 XY 테이블 위에 올려져서 이동한다. XY 테이블은 X-축, Y-축으로 이동하면서 장착 작업을 돕는다. 즉 PCB의 지정된 장착점이 장착 스테이션 (Mounting Station: ⑩)에 오도록 XY테이블은 수평

Table 1 Factors of the efficient operation of rotary type surface mounter

구분	인덱스 테이블	X-Y테이블	릴 축
인자의 종류	인덱스 테이블 속도	X-Y 테이블 속도	부품 카세트의 크기
	Positioning Unit 장착 각도	X-Y 테이블 이동거리	부품 카세트의 위치

및 수직 운동을 수행한다. XY테이블의 이동 거리는 부품 조립 시간에 영향을 주기 때문에 PCB위의 장착점 간의 거리는 표면실장기의 운용전략을 수립할 시 고려해야 할 인자가 된다. 이상에서 살펴 본 인자들을 정리하여 Table 1에 나타낸다.

2.2 하드웨어에 대한 고려사항

로터리형 표면 실장기의 운영을 위해 고려해야 할 사항으로써 먼저 릴 축의 이동을 들 수 있다. 릴 축의 단위이동을 피치라고 한다. 릴 축이 이동의 기본 단위인 8mm를 이동할 때에는 단위시간(예를 들면 0.25초⁽⁸⁾)이 소요된다. 본 연구의 대상 시스템은 한 번에 10피치 이상의 릴축 이동을 허용하지 않는다고 가정한다.

본 시스템을 이용하여 부품을 PCB위에 장착시키는 작업을 수행하는 순서를 나타내면 Fig. 2와 같다.

부품의 포지셔닝을 위하여 사용되는 유니트는 (2)번 다음에(4)번을, (3)번 다음에 (5)번을 바로 이어서 사용할 수 없으며 그 역으로도 마찬가지이다. (2)번 유니트를 사용한 후 (4)번 유니트를 사용하면 유니트의 변경을 위한 시간이 너무 많이 소요됨으로(약 1.5초 이상⁽⁸⁾) (2)번을 유니트를 사용한 후에는 반드시 (4)번 이외의 다른 유니트를 사용해야 하는데 다른 유니트를 1회 사용한 수 (4)번 유니트를 사용하게 되면 포지셔닝을 위하여 0.8초가 소요되며 2회 사용 후(4)번 유니트를 사용하면 0.25초가 소요된다. 이와 같은 조건은(3)번과 (5)번의 경우에도 동일하게 적용되며 그 역으로도 마찬가지이다. 많은 수의 부품을 PCB위에 조립해야 할 경우 최적 운영

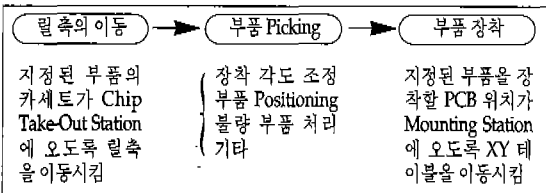


Fig. 2 Mounting procedure of the chip

알고리즘을 사용하여 부품 카세트의 위치와 장착순서를 결정하는 데에는 많은 시간이 요구되어 현장 적용을 위한 경험적 알고리즘의 개발이 요구된다. 제3장에서는 소프트웨어의 개발에 사용된 경험적 알고리즘을 소개한다.

3. 알고리즘 개발

알고리즘의 개발을 위한 수리모델을 먼저 제시하고 개발된 알고리즘을 소개한다.

3.1 수리모델

로터리형 표면 실장기의 운영 최적화에 대한 수리모델을 아래 식(3.1)부터 식(3.9)까지에 제시한다. 수리모델에 대한 가정은 아래와 같다.⁽²⁾

- 1) 인덱스 테이블의 속도는 모든 부품에 대해 동일하다.
- 2) 장착각도의 상대적인 차이는 부품 조립 시간에 영향을 주지 않는다.

- 3) 부품을 포지셔닝하기 위한 유니트의 사용이 변화되어도 부품 조립시간에 영향을 미치지 않는다.

- 4) 부품 카세트의 크기는 모두 동일하다고 본다.

위의 가정을 생각한다면 표면 실장기의 운영전략을 수립할 경우 PCB상의 부품 장착 위치 간의 거리와 릴 축의 이동시간을 고려하여 부품 카세트의 위치와 부품 장착순서를 결정할 수 있다. 이것은 m개의 부품 카세트를 릴의 l개의 위치에 배치하고 PCB위의 n개 장착 위치에 대한 장착순서를 결정하는 문제가 된다.⁽⁶⁾

$$\min \sum_{r=1}^l \sum_{s=1}^l \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n C_{ijrs} X_{ijk} Y_{c_i} Y_{c_j} \quad (3.1)$$

$$S.t. \sum_{i=1}^n \sum_{k=1}^n X_{ijk} = 1 \quad 1 \leq j \leq n \quad (3.2)$$

$$\sum_{j=1}^n \sum_{k=1}^n X_{ijk} = 1 \quad 1 \leq i \leq n \quad (3.3)$$

$$\sum_{v=1}^n \sum_{k=1}^{k_0} X_{ivk} = 1 \geq \sum_{v=1}^n \sum_{k=1}^{k_0} X_{vj k} \quad 1 \leq k_0 \leq n \quad (3.4)$$

$$\sum_{c_i, v=j}^n X_{ijk} = \sum_{v, v=j}^n X_{jv(k+1)}$$

$$1 \leq j \leq n, 1 \leq k \leq n-1 \quad (3.5)$$

$$X_{ijk} = 1 \text{ or } 0 \quad (3.6)$$

$$\sum_{w=1}^l Y_{wr} = 1 \quad 1 \leq r \leq l \quad (3.7)$$

$$\sum_{w=1}^l Y_{uw} = 1 \quad 1 \leq u \leq l \quad (3.8)$$

$$Y_{wr} = 1 \text{ or } 0 \quad (3.9)$$

여기서 X_{ijk} 는 장착점 순서를 나타내며 Y_{wr} 은 릴축의 카세트 위치를 나타내는 이진변수이다. C_{irs} 은 장착점 i 에 장착되는 부품을 릴의 위치 r 에 있는 카세트가 포함하고 있으며 장착점 j 에 장착하는 부품을 릴의 위치 s 에 있는 카세트가 포함하고 있을 때, 장착점 i 에 있어서 장착점 j 에 부품을 장착하는데 요구되는 장착소요시간이다. 식(1)은 인접하는 장착점의 n 개 쌍을 선택하여 장착순서의 모든 경우들 중에서 장착시간이 최소가 되는 경우를 선택하는 목적함수이다. 식(2)와 식(3)은 장착점을 오직 한 번씩만 방문하게 한다. 식(4)와 식(5)는 장착점에 대한 선행 제약과 부분환이 생기는 것을 방지한다. 식(7)은 모든 부품 카세트는 각각 릴 축 위의 한 위치에 배치되게 한다.

한편 현장에서 요구하는 알고리즘은 수리모델을 세우는 과정에서 가정된 모든 것들이 포함되어야 하고 빠른 시간내에 해를 제시해 주어야 함으로 이를 만족시킬 수 있는 새로운 알고리즘이 개발될 필요가 있었다. 이.와 김.은 시스템 운영에 고려해야 할 인자들과 제약들을 고려하여 현장 적용을 위한 최적 알고리즘을 개발하였다.⁽¹⁾ 이.와 김.에 의해 제시된 알고리즘은 계산량이 많아 문제의 크기가 큰 경우에는 최적해를 주기 위해 매우 많은 컴퓨터 수행시간을 요구하여 현장 적용에 적절하지 않다. 따라서 현장에서 실시간에 적용 가능한 경험적 알고리즘을 본 연구에서 개발하였다. 그것들은 Algorithm I과 Algorithm II의 두가지이며 이를 소프트웨어의 개발에 이용하였다. Algorithm I은 해를 구하는 컴퓨터 수행 시간이 많이 요구되지만 Algorithm II에 의한 것보다 항상 좋은 해를 제시해 준다. 반면에 Algorithm II는 Algorithm I에 비해 다소 떨어지는 해를 제시해 주지만 매우 짧은 시간 내에 해를 제시해 준다. 소프트웨어 개발에 이용된 알고리즘들

에 대한 설명은 아래와 같다.

3.2 Algorithm I

연속적인 PCB 조립작업을 수행할 때 부품 한 개를 PCB 위에 장착하는데 요구되는 시간 T_{ij} 는 Table 1에 제시된 인자들과 Figure 2.에 나타난 장착 순서를 참조하여 식(3.10)과 같이 나타낼 수 있다.^(1,2)

$$T_{ij} = \max(I_{ij}, X_{ij}, P_{ij}) \quad (3.10)$$

여기서, PCB위의 노드 i 다음에 노드 j 에 부품을 장착할 때, I_{ij} 는 그 부품이 인덱스 테이블에서 보내는 시간이고 X_{ij} 는 XY테이블이 PCB를 이동시키는데 요구되는 시간이며, P_{ij} 는 릴 축이 이동하는데 요구되는 시간이다. 위 식(3.10)은 Table 1에 나타난 인자들 중에서 부품 카세트의 위치를 제외한 모든 인자를 포함한다. Algorithm I을 이용하여 부품 카세트의 위치와 부품 장착순서를 동시에 결정하는 절차는 아래와 같다.

Step 0. S_1 는 장착순서를 의미한다. 시작노드 $i=1$ 로 둔다.

$S_2 \rightarrow$

Step 1. 한 부품을 노드 i 에 장착한 후 다음에 장착할 노드를 결정한다. 이는 노드 i 에 장착하는 부품과 같은 종류의 부품을 장착할 다른 노드들 중에서 가장 작은 T_{ij} 값을 갖는 노드를 선택한다. 만일 노드 i 에 장착되는 부품과 같은 부품을 장착할 다른 노드가 없으면 Step 2.로 간다. (선택되지 않은 행렬 T 의 모든 원소에 대해서)

$$S_{22} = \min(T_{ij})$$

Step 3.으로 간다.

Step 2. 모든 노드가 선택되었는지 확인한다. 모두 선택되었으면 Step 4.로 간다. 그렇지 않으면 현재 선택된 노드 다음에 가장 빨리 작업을 수행할 수 있는 노드를 선택한다. (선택되지 않은 모든 T 의 원소에 대해서)

$$S_{22} = \min(T_{ij})$$

Step 3. 초기 장착 순서를 결정한다.

$$S_{21} \rightarrow S_{22} \rightarrow \dots$$

Step 1.로 간다.

Step 4. 시작 노드 i 가 마지막 노드이면 Step 5.로 간다. 그렇지 않으면 시작 노드를 $i=i+1$ 로 두고 Step 0.으로 간다.

- Step 5. 카세트의 위치를 결정한다.
Step 3에서 구한 초기 장착순서의 그룹들 중에서 조립시간의 합이 가장 작은 그룹을 선택한다. 그 그룹은 같은 종류의 부품들을 먼저 장착시키게 되도록 순서가 정해져 있으므로 그 그룹의 순서대로 카세트를 배치시키고 Step 6으로 간다.
- Step 6. 종료.

3.3 Algorithm II

경험적 알고리즘의 효율은 시작 노드에 의해 크게 좌우된다. Algorithm II는 시작노드를 결정한 후 이를 이용하여 카세트의 위치와 부품의 장착순서를 결정한다. 먼저 시작노드를 선정하는 절차를 설명한 후, 부품 카세트의 위치와 장착순서를 결정하는 절차를 설명한다.

본 연구의 대상 시스템을 운영함에 있어서 시작노드는 기본시간인 0.25초 안에 장착 될 수 있고 그것의 부품카세트가 릴 축의 맨 앞쪽에 배치되어 있으며, 장착노드가 원점에서 가까이 있을 때 좋은 노드라고 생각되어 이를 시작노드로 정한다.

Stage 1. 시작노드의 결정

- Step 1. 릴 축의 단위 피치를 차지하게 되는 부품을 추려낸다.
- Step 2. 인덱스 테이블의 속도가 0으로 지정된 부품을 그 중에서 추려낸다.
- Step 3. 장착각도가 180도 미만으로 지정된 부품을 추려낸다.
- Step 4. XY 테이블을 가장 적게 움직여서 부품을 장착할 수 있는 노드를 선택한다. 즉 Step 3에서 추려진 노드들 중에서 PCB의 원점으로부터 가장 가까이 있는 노드를 선택하고 이를 시작노드로 고정한다.

Stage 2. 카세트의 배치 및 부품 장착 순서의 결정

- Step 1. 선택된 노드에 장착할 부품의 카세트를 현재 배치된 카세트 다음에 배치한 후 Step 2로 간다.
- Step 2. 현재노드에 부품을 장착한 후 그와 같은 종류의 부품들 중에서 가장 빨리 장착할 수 있는 다음의 노드를 선택한다. 즉 T에 대해서 최근 거리 인접점방법(Nearest Neighborhood Rule)을 적용한다. 같은 종류의 부품이 모두 선택될 때까지 이를 반복한다.

- Step 3. PCB 위에 장착할 모든 부품에 대한 카세트의 배치가 완료되었으면 Step 4로 가고, 그렇지 않으면 현재 노드에서 아직 tour가 형성되지 않았고 가장 빨리 장착할 수 있는 다음의 노드를 선택한 후 Step 1.로 간다.
- Step 4. 구성된 tour의 순서대로 부품 장착순서를 정하고 카세트의 배치를 종료한다.

4. 실험 및 결과

본 연구에서 제시한 알고리즘들은 C언어를 이용하여 프로그램되어 PC-486 DX-II 50에서 수행되었다. 실험 대상의 데이터들은 난수를 발생시켜 만들어졌다. 각각의 경우들에 대해 10회를 모의실험하여 평균을 내었으며 그 결과를 Table 2.에 나타낸다. 비교 대상으로 삼은 기존 방법은 현재 로터리형 표면 실장기를 사용하고 있는 공장에서 실제 운용하고 있는 방법으로서 인덱스 테이블의 회전 지정 속도의 빠른 순의 배열에 따라서 카세트의 위치를 결정한 후, 이를 고려하여 1번 카세트가 포함하는 부품 종류의 장착점을 찾아 최단거리 인접점 방법으로 장착순서를 결정한다. 1번 카세트의 부품이 장착되는 모든 노드들에 대한 장착 순서가 결정되었으면 2번 카세트가 포함하는 장착 부품들에 대해 동일한 방법으로 장착순서를 결정한다. 현장의 작업자들은 장착할 PCB위의 노드가 표시되어 있는 맵(Map)을 이용하여 이러한 작업을 수행한다. 장착순서가 모두 결정되었으면 NC 파트 프로그램을 작성하여 표면 실장기에 로딩 시킨 후 시험 가동을 실시한다. 시험가동을 하면서 NC 파트 프로그램을 수정하여 최종적인 카세트의 위치와 장착 순서를 결정한다. 이러한 기존방법을 프로그래밍하여 개발된 알고리즘의 비교 대상으로 삼았으며 이를 Table 2에 나타내었다.

Table 2에서 수행시간은 알고리즘을 이용하여 해를 구할 때 소요되는 컴퓨터 수행시간을 의미한다. Table 2에서 보여준 바와 같이 난수 발생하여 얻은 데이터를 가지고 부품 카세트의 위치와 장착순서를 결정할 때에는 개발된 Algorithm I과 Algorithm II가 기존방법에 비해 훨씬 좋은 해를 제시해 준다. 이는 개발된 알고리즘들은 모든 변수들을 고려하여 카세트의 위치와 장착순서를 결정해 준 데서 기인한다. Table 2에서 제시한 실험결과에 대한 그래프를 Fig. 3에 나타낸다.

Table 2 Results of experiments for algorithms.

장착점 개수	Algorithm II (1)		Algorithm I (2)		기 존 방 법		생산성향상률(%)	
	조립시간	수행시간	조립시간	수행시간	조립시간	수행시간	(2) 대 비	(1) 대 비
50	27.44	3.41	26.90	9.02	39.8	3.15	31.06	32.41
100	57.65	5.12	56.91	44.77	81.98	4.83	27.04	27.86
150	91.19	7.35	89.63	169.34	124.27	5.13	26.62	27.87
200	124.80	9.80	123.10	458.75	167.43	6.98	25.46	26.47
300	184.62	16.62	182.45	2172.65	247.60	8.92	24.22	26.31
400	253.32	26.62	247.85	*****	334.00	11.18	24.15	*****
500	307.45	414.38	301.85	*****	413.51	13.52	25.65	*****

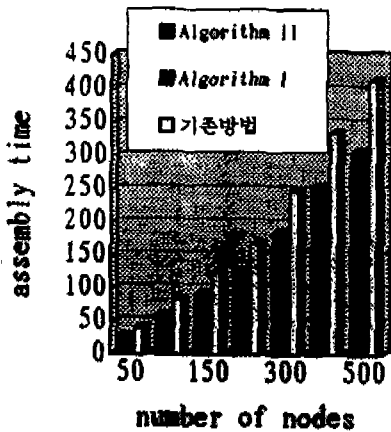


Fig. 3 Graph for experimental results

5. 소프트웨어 개발

소프트웨어는 IBM PC호환기종(386급 이상이 바람직함)상에서 구현할 수 있으며 운영체제는 MS-DOS이다. 본 연구를 통해서 개발된 소프트웨어의 구성도를 Fig. 4에 나타낸다. 소프트웨어의 사용을 위해 요구되는 하드디스크의 메모리는 실행 시 중간 결과의 저장을 위해 필요한 메모리를 포함하여 약 3M이다. 소프트웨어는 크게 OPTIMIZATION과 SIMULATION의 두 부분으로 구성된다. 이에 대한 설명은 아래와 같다.

5.1 OPTIMIZATION부

표면 실장기의 최적 운영을 위한 부품 카세트의 위치와 장착 순서를 결정하는 기능을 수행하는 부분으로서 원시 데이터의 입력을 위한 INPUT메뉴, 최적화를 수

행한 후 결과를 파일에 출력시키는 OUTPUT메뉴, 그리고 파일의 수정과 삭제, 복사, 프린트 등을 위한 FILE MANAGER의 세부 메뉴로 구성하였다. OPTIMIZATION의 초기 화면을 Figure 5에 나타내고 입력 데이터를 지정하기 위하여 Input 명령을 실행시킨 후의 화면상태를 Figure 6에, 그리고 파일 관리를 위해 FILE MANAGER를 열었을 때의 화면을 Figure 7에 나타낸다.

5.2 SIMULATION부

SIMULATION부에서는 정해진 부품 카세트의 위치와 장착 순서에 따라 현장에서 작업을 수행하기 전, 작업의 진행 사항을 컴퓨터 상에서 확인할 수 있도록 파일을 선택하고 실행을 위한 Mounting메뉴, 장착 작업이 수행됨에 따라 변화되는 사이클 타임의 확인을 위한 Time Check메뉴, 그리고 시뮬레이션 결과의 출력을 위한 Print메뉴 등의 세부 메뉴로 구성되어 있다. 장착 작업의 경로를 모니터 상에서 볼 수 있게 하였으며 의심이 되는 장착 순서를 확인할 수 있도록 이전의 경로를 볼 수 있는 기능을 첨가하였다. 또한 부품 장착이 파일 전체에서 얼마나 진행되었는지 확인할 수 있도록 작업의 진행 사항을 그래프와 시간으로 볼 수 있게 하였다. 시뮬레이션이 수행되는 경우, Mounting실행 시의 화면과 경로를 확인할 수 있는 화면을 Figure 8과 Figure 9에 각각 나타낸다.

6. 개발된 소프트웨어의 특성

개발된 소프트웨어는 여러가지 특성을 지니고 있다. 첫째, 현장 사정을 충분히 반영한 알고리즘을 이용하여

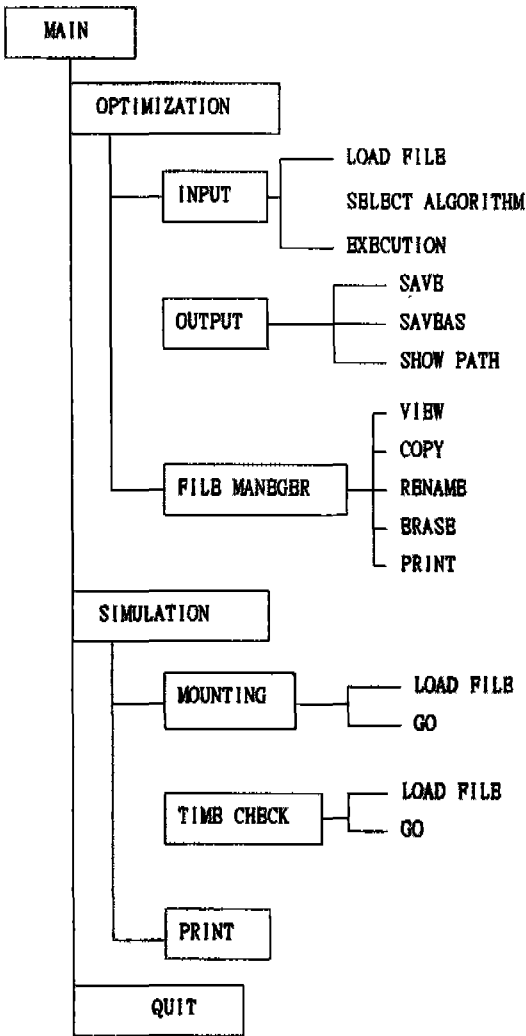


Fig. 4 Structure of the developed software

소프트웨어가 개발되었기 때문에 이를 이용하여 해를 구한 후 현장 사정을 고려하여 다시 해를 조정해야 하는 수고를 요구하지 않는다. 기존 연구에서는 대부분 시스템의 복잡성과 이로 인한 메커니즘의 이해의 부족 때문에 모든 인자들을 고려하지 않고 모델을 단순화시킨 알고리즘을 개발하여 이론적인 수준에 머무를 수밖에 없었다. 이론적인 알고리즘에 의해 해를 산출하면, 기계를 직접 돌려 가며 하드웨어의 여러가지 제약 조건들을 만족시키면서 효율적인 운영을 위한 최종 해를 구해야 할 필요가 있었다. 그러나 본 연구에 의한 소프트웨어는 현장의 이러한 수고를 요구하지 않는다. 둘째,

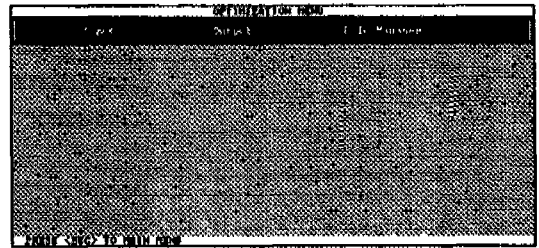


Fig. 5 Initial screen of the OPTIMIZATION menu

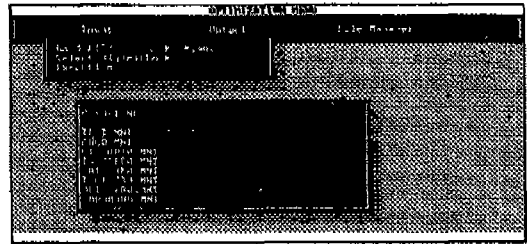


Fig. 6 An example screen for the file search

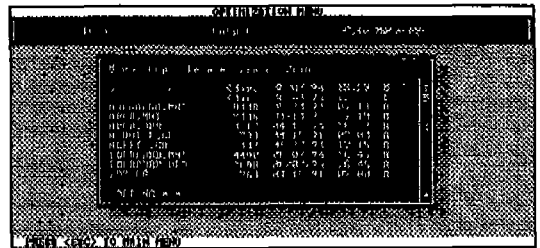


Fig. 7 Initial screen of FILE MANAGER menu

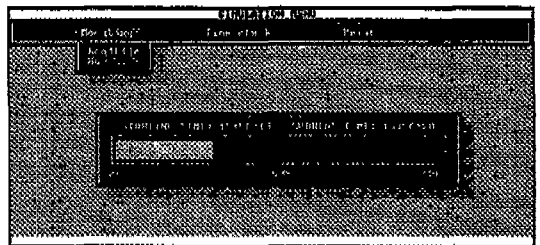


Fig. 8 An example Screen of Mounting

소프트웨어는 POP-UP메뉴 방식으로 되어 있다. 따라서 컴퓨터를 사용하는 초보자라고 하더라도 컴퓨터의 모니터 상에 나타나는 내용을 보고 원하는 결과를 얻을 수 있다. 셋째, 시뮬레이션이 수행될 때 부품의 장착 경로를 확인할 수 있게 하였다. 컴퓨터 키보드 상의 F1 키를 누르면 전체 시뮬레이션 과정의 작업 경로를 모니

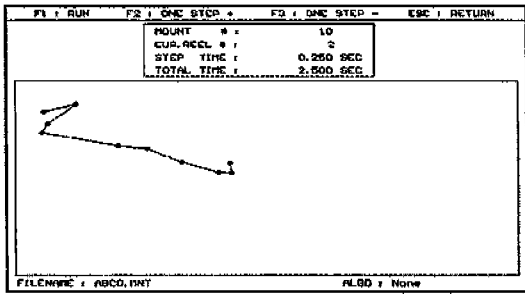


Fig 9 Screen for mounting path simulation

터 상에서 보여준다. 한편 시뮬레이션이 진행되는 도중 F2키를 누르면 현재의 장착 경로의 다음 상태를 보여주고 멈춘다. F2키를 다시 누르면 시뮬레이션이 현재 위치에서 다시 다음의 한 스텝을 보여준다. 한편 F3키를 누르면 현재의 이전 경로를 보여준다. 이로써 소프트웨어를 통해 산출한 해를 기계에 바로 로딩 시키기 전에 작업의 경로를 확인할 수 있게 하였다. 넷째, 현장에서 사용하는 원시데이터를 그대로 활용한다. 최적화를 위해 필요로 하는 정보만을 추출하는 수고를 요구하지 않으며 또한 최적화 한 후에도 원시데이터와 동일한 포맷으로 해를 산출해 주기 때문에 기계의 운영을 위한 별도의 NC Part Program의 작성을 요구하지 않는다.

7. 결 론

본 연구는 하드웨어의 모든 제약 조건들을 고려하여 표면 실장기의 최적 운영을 위한 해를 산출할 수 있는 알고리즘 및 소프트웨어를 개발하여 제시하였다. 이로써 외국산 소프트웨어에 의존하던 국내 기업의 욕구를 충족시킬 수 있는 기반을 마련하였다. 또한 소프트웨어의 활용을 위해 대형 컴퓨터의 도입을 검토할 때, 가질 수 있는 중소기업들의 부담을 줄일 수 있게 되었다. 그런데 본 연구에서 제시한 소프트웨어는 표면 실장기 기

계 한 대의 효율을 극대화시키려는 관점에서 개발된 것이나, 앞으로는 여러 대의 기계로 구성된 PCB조립라인의 효율을 극대화 할 수 있는 소프트웨어의 개발을 기대한다.

참 고 문 헌

1. 이영해, 김정, "표면실장기의 최적운영을 위한 모델링 및 알고리즘 개발", 대한산업공학회지, 20, 3, pp.79-92, 1994. 9.
2. 이영해, 김정, 김덕한, 경규형, "칩 마운터의 고생산성을 위한 최적화에 관한 연구", 보고서, 한양대 생산공학 연구소, 1994.
3. 김정, 이영해, "로터리형 칩 마운터의 최적 운영", 로보틱스 및 자동화 연구회 Workshop 논문집, pp.71-74, 1994. 3
4. 김두근, "CNC 컨트롤러의 개발 사양 결정 방법에 관한 연구", 산업공학, 7권, 1호, pp.P5-12, 1994. 3
5. 신경훈, "국내 CNC의 취약기술", 산업공학, 7권, 1호, pp.13-18, 1994. 3
6. Leipala, T. and O. Nevalainen, "Optimization of the Movements of a Component Placement Machine", European Journal of Operational Research, 38, pp.167-177, 1989.
7. Walas, Robert A. and Ronald G. Askin, "Algorithm for NC Turret Punch Press Tool Location and Hit Sequencing", IIE Transactions, Vol.16, No.3, pp.280-287, 1984.
8. "Chip형 전자부품 자동장착장치 DCM-70L 매뉴얼", 대우전자(주) 생산기술연구소, 1993.