

# SMT 검사기의 경로 계획을 위한 통합적 접근 방법

## A Unified Approach to Path Planning of SMT Inspection Machines

김 화 중, 정 진 회, 박 태 형\*  
(Hwa-Jung Kim, Jin-He Jung, and Tae-Hyoung Park)

**Abstract :** We propose a path planning method to improve the productivity of SMT (surface mount technology) inspection machines with an area camera. A unified method is newly proposed to determine the FOV clusters and camera sequence simultaneously. The proposed method is implemented by a hybrid genetic algorithm to increase the convergence speed. Comparative simulation results are then presented to verify the usefulness of the proposed algorithm.

**Keywords :** path planning, optimization, genetic algorithm, SMT inspection machines.

### 1. 서론

최근 컴퓨터, 휴대폰, PDA 등의 보급으로 전자제품의 수요가 급증함에 따라, 표면실장기술(SMT; surface mount technology)을 이용한 인쇄회로기판(PCB; printed circuit board) 전자 조립시스템의 중요성이 증대되고 있다. SMT 인라인 시스템은 표면실장부품을 인쇄회로기판에 조립하는 전자조립시스템으로서, 전자부품의 소형화, 고밀도화 등 부품제작기술의 발전과 조립라인 자동화 추세에 의해 현재 대부분의 전자제품 제조라인에서 채택되고 있다[1].

SMT 인라인 시스템을 구성하는 장비 중 검사기는 부품 장착 이상, 납땀 불량, 회로 단락 등 인쇄회로기판 조립 시 발생하는 이상을 자동으로 검출하는 전용 로봇 시스템이다. SMT 검사기는 영상처리기술의 발달에 힘입어 최근 많이 보급되고 있으며, 조립라인의 생산성 향상 및 품질 신뢰성 향상에 크게 기여하고 있다. SMT 검사기는 영역 카메라(area camera)를 사용하는 형식과 선형 스캔 카메라(line-scan camera)를 사용하는 형식으로 구분되며, 영역 카메라를 사용하는 경우가 보다 많이 사용되고 있다[2-4]. 본 논문은 영역 카메라형 SMT 검사기의 경로계획을 위한 방법을 제안한다.

그림 1은 영역카메라 형 검사기의 평면을 보여준다. 평면 운동을 하는 XY 로봇에 영역 카메라가 부착되어 이동하며 PCB 내의 모든 검사 윈도우에 대한 검사를 수행한다. 이 때 카메라가 한 번에 획득할 수 있는 영상영역을 FOV (field-of-view)라 한다. 검사기의 작업 프로그래밍을 위해서는 전체 기판영역을 몇 개의 FOV로 어떻게 분할하여 검사할 것인지를 결정하여야 하며, 또한 각 FOV에 대한 카메라의 방문 순서를 결정하여야 한다. 검사기의 경로계획 문제는 전체 검사시간을 최소화 시키는 FOV 생성과 FOV

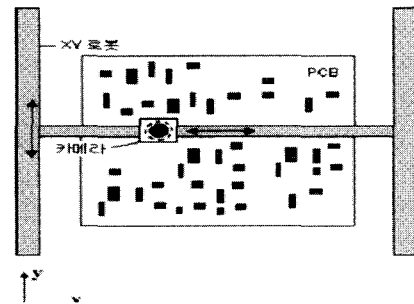


그림 1. 영역 카메라 형 SMT 검사기.

Fig. 1. SMT inspection machine with area-camera.

순서를 결정하는 문제이다.

검사기의 경로계획문제는 복잡한 조합최적화 문제의 범주에 속하며, 전역적 최적해를 구하는 것은 매우 어렵다. 그러므로 근사 최적해를 구하는 방법으로서, 전체 문제를 분할하여 계층적으로 해결하는 방법이 시도되어 왔다. 즉 FOV를 우선 생성시키고, 그 결과로부터 FOV의 순서를 결정하는 방법이다. 이 때 FOV 생성 문제는 일종의 클러스터링 문제의 범주에 속하며, FOV 순서결정 문제는 전형적인 순회 외판원 문제(TSP; traveling salesman problem)이다. 따라서 검사기의 경로계획을 위한 연구는 주로 FOV 생성 방법에 집중되었다. 순차적인 클러스터링 알고리즘[5], 인공신경망을 사용한 클러스터링 알고리즘 [6]을 검사기 FOV 생성문제에 적용한 연구가 발표되었다. 또한 ISODATA 알고리즘 [7]과 유전자 알고리즘 [8]을 검사기 FOV 생성문제에 적용한 연구가 발표되었다.

그러나 검사기 경로계획 문제에 있어서, FOV 생성과 FOV 순서는 서로 밀접하게 연결되어 있다. 따라서 보다 효과적인 검사시간 단축을 위해서는 FOV 생성과 FOV 순서가 동시에 결정되는 것이 바람직하다. 본 논문은 처음으로 FOV 생성과 FOV 순서를 동시에 결정하는 경로계획 방법을 제안한다. 즉 기존의 계층적 접근 방법에 대하여 통합적 접근

\* 책임저자(Corresponding Author)

논문접수 : 2004. 1. 12., 채택확정 : 2004. 7. 6.

김화중 : KEC(주) 메카트로닉스연구소(jjackji@korea.com)

정진회, 박태형 : 충북대학교 제어측정공학과 및 컴퓨터정보통신연구소(jjh1104@dclick.net/taehpark@chungbuk.ac.kr)

※ 본 논문은 2004년 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었음.

근을 시도하며, 이를 위한 하이브리드 유전자 알고리즘을 제시한다. 상용 검사기를 대상으로 비교평가를 수행하며, 본 논문에서 제시한 방법의 유용성을 검증한다.

**II. 검사기 경로계획 문제**

그림 2는 SMT 검사기의 윈도우, FOV 및 검사경로를 보여준다. 검사기가 기판에서 검사하고자 하는 대상을 윈도우라 정의한다. 윈도우는 모두 사각형의 영역이며, 부품 윈도우, 납땜 윈도우 및 브리지 윈도우로 구분될 수 있다. 부품 윈도우는 부품 장착의 이상, 납땜 윈도우는 납땜 불량, 브리지 윈도우는 리드 간 단락을 검출하는 영역이다. 하나의 부품에 대하여 여러 개의 윈도우가 설정될 수 있으며, 하나의 기판에는 통상 수백 내지 수천 개의 윈도우가 존재한다. 검사기는 카메라를 이동시켜 모든 윈도우에 대한 영상을 획득하고 처리하여 검사한다.

카메라가 한 번에 획득하여 처리할 수 있는 영역의 크기를 FOV(field of view)라 정의한다. FOV의 크기는 검사기의 고유 값으로서 통상 가로 세로 각 수십 mm 이내의 크기를 가지며, 하나의 FOV에 여러 개의 윈도우가 포함될 수 있다. 따라서 기판에 존재하는 모든 윈도우에 대한 검사를 수행하기 위해서는, 카메라를 이동시켜 여러 번의 영상을 획득하여야 한다. 이 때 각 FOV에 포함되는 윈도우들의 묶음을 클러스터라 정의한다. 검사기는 각 FOV의 중심위치에 카메라를 일정시간 정지하여 영상을 획득하고, 다른 FOV로 이동하며 검사작업을 수행한다. 이 때 각 FOV를 방문하는 카메라의 이동경로를 검사경로라 정의한다.

기판에 존재하는 검사 윈도우의 집합  $W$  및 FOV의 집합  $F$  를 다음과 같이 정의한다.

$$W = \{1, \dots, m\} \tag{1}$$

$$F = \{1, \dots, n\} \tag{2}$$

이 때 FOV의 총 수  $n$ 은 주어진 값이 아니며, 결정되어야 하는 변수이다. 이진변수  $z_{ij} \in \{0, 1\}^{m \times n}$ 을 윈도우  $i \in W$ 가 FOV  $j \in F$ 에 속하였을 경우에만 1이 되는 값으로 정의한다. 또한 이진변수  $x_{ij} \in \{0, 1\}^{n \times n}$ 을 카메라가 FOV  $i \in F$  방문 직후 FOV  $j \in F$ 로 이동하였을 경우에만 1이 되는 값으로 정의한다. 변수  $z_{ij}$ 는 FOV 생성을 위한 값이며, 변수  $x_{ij}$ 는 FOV 순서결정을 위한 값이다.

검사기 경로계획 문제는 다음의 최적화 문제로 표현될 수 있다.

(3)의  $t_{acq}$ 는 각 FOV에 대한 카메라의 영상획득시간이며,  $t_{ij}$ 는 FOV  $i$ 에서 FOV  $j$ 로의 이동시간 이다. 그러므로 (3)은 모든 윈도우에 대한 총 검사시간의 최소화를 의미한다. (4)는 모든 윈도우는 한 개의 FOV에 포함되어야 함을 의미하며, (5)는 모든 FOV는 한 개 이상의 윈도우를 포함하여야 함을 의미한다. (6)에서  $X_{max}^j$  및  $X_{min}^j$ 는 FOV  $j$ 에 포

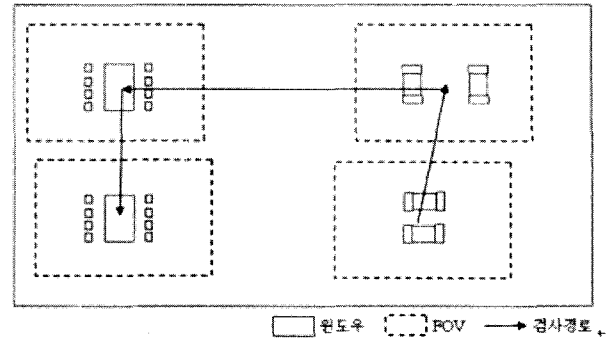


그림 2. 윈도우, FOV 및 검사경로.

Fig 2. Windows, FOV and path.

$$\min t_{acq} + \sum_{i \in F} \sum_{j \in F} t_{ij} x_{ij} \tag{3}$$

s.t.

$$\sum_{j \in F} z_{ij} = 1, \quad \forall i \in W \tag{4}$$

$$\sum_{i \in W} z_{ij} \geq 1, \quad \forall j \in F \tag{5}$$

$$X_{max}^j - X_{min}^j \leq X_{FOV}, \quad \forall j \in F \tag{6}$$

$$Y_{max}^j - Y_{min}^j \leq Y_{FOV}, \quad \forall j \in F \tag{7}$$

$$\sum_{j \in F} x_{ij} = 1, \quad \forall i \in F \tag{8}$$

$$\sum_{i \in F} x_{ij} = 1, \quad \forall j \in F \tag{9}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subset F \tag{10}$$

함된 윈도우 집합  $\{i \in W | z_{ij} = 1\}$ 에 대한 X 좌표의 최대값 및 최소값을 의미한다. 즉 (6)은 각 FOV에 포함된 윈도우들의 X 좌표는 FOV의 고정된 크기  $X_{FOV}$  범위 내에 위치하여야 함을 의미한다. 마찬가지로 (7)은 윈도우들의 Y 좌표에 대한 제약을 의미한다. (8) 및 (9)는 각 FOV를 카메라가 단 1 회 씩 모두 방문하여야 함을 의미한다. (10)에서  $|S|$ 는 FOV의 부분집합  $S$ 의 원소의 개수이며, 일부 FOV만을 포함하는 사이클이 구성될 수 없음을 의미한다.

(3)의 첫 번째 항은 FOV의 총 수를 최소화시키는 경우 최소화되며, 두 번째 항은 총 이동 경로를 최소화시키는 경우 최소화된다. 변수  $z_{ij}$ 의 결정은 FOV의 총 수는 물론 총 이동경로에도 영향을 준다. 또한 변수  $x_{ij}$ 의 결정은 총 이동경로에 영향을 준다. 즉, 전체 목적함수의 최소화를 위해서는 두 변수를 동시에 결정하는 것이 바람직함을 알 수 있다.

**III. 경로계획 알고리즘**

앞 장에서 정의된 바와 같이 SMT 검사기의 경로계획 문

제는 FOV 생성 및 FOV 순서를 결정하는 문제로서, 상호 영향을 주는 두 가지 변수를 결정하는 문제이다. 기존의 연구는 FOV 생성 단계와 FOV 순서결정 단계를 분리하여 접근하는 계층적 방법을 사용하였다[5-8]. 즉 FOV 생성 변수를 우선 구하고, 이로부터 FOV 순서 변수를 결정하는 방법이다. FOV 생성 문제는 클러스터링 문제의 범주에 속하며, FOV 순서 결정 문제는 순회 외판원 문제의 범주에 속한다. 그러므로 FOV 생성 단계에서는 잘 알려진 클러스터링 알고리즘 [9][10]을 변형하여 적용하며, FOV 순서결정 단계에서는 전형적인 TSP 알고리즘 [11][12]을 직접 적용하는 방법을 사용한다. 이 방법은 복잡한 최적화 문제를 두 개의 잘 알려진 최적화 문제로 분할시킴으로서, 비교적 수월하게 해를 구할 수 있는 장점이 있다. 그러나 변수들 상호 간의 영향을 무시하고 해를 결정하여야 하므로, 해의 최적화 성능은 떨어질 수 있다.

FOV 생성 변수와 FOV 순서 변수의 상호 영향을 고려하기 위해서는, 두 변수를 동시에 결정하는 통합적 접근이 바람직하다. 본 논문은 경로계획의 성능을 높이기 위한 통합적 방법을 제시한다. 복잡한 최적화문제를 분리시키지 않고 직접 해결하기 위하여, 유전자 알고리즘을 적용한다. 유전자 알고리즘은 복잡한 최적화 문제의 근사 최적해(near-optimal solution)를 구하는 데 광범위하게 사용되고 있으며, 국지적 최적해에 머무는 현상을 제거시키는 데 효과가 있다고 알려져 있다[13]. 특히 검사기 경로계획과 관련이 있는 클러스터링 문제 [14][15] 및 순회 외판원 문제 [16][17]에도 적용한 사례가 많이 있다.

그림 3은 본 논문에서 제시하는 경로계획 알고리즘의 흐름도이다. 기본적으로 유전자 알고리즘의 흐름도와 동일하다. 초기 집단 (population)을 확률적으로 생성시키고, 적합도를 기준으로 우수한 개체들을 복제한다. 확률적 교배 및 돌연변이 연산을 통하여 복제된 개체들을 변이시키며, 또한 순서결정 연산을 추가로 수행시킨다. 복제 및 변이 동작을 반복하여, 우수한 개체를 선택하고 이로부터 새로운 개체를 생성시키는 진화 법칙을 적용한다. 결과적으로 가장 우수한 개체를 선택하여 최적해로 한다. 유전자 알고리즘을 적용하기 위해서는 새로이 염색체 및 적합도를 정의하여야 하며, 교배 및 돌연변이 연산을 정의하여야 한다. 순서결정 연산은 검사기 경로계획문제를 위하여 추가로 고안된 연산이며, 순서 최적화를 위한 국지탐색방법을 모든 개체에 대하여 적용한다. 이 연산은 해의 수렴속도를 빠르게 하기 위한 미소 조절(fine tuning) 기능을 수행하며, 결과적으로 전체 알고리즘은 하이브리드 유전자 알고리즘의 범주에 속한다.

유전자 알고리즘 수행의 기본 단위인 염색체 (chromosome)를 그림 4와 같이 정의한다. 염색체를 구성하는 유전자 (gene)는 윈도우 번호  $w_{ij} \in W$  또는 0의 값을 갖는다. 이 때  $w_{ij}$ 는  $i$  번째 방문 FOV 에 속한  $j$  번째 윈도우이며, 0은 FOV 사이의 구분을 위하여 사용하는 기호이다. 정의된 염색체는 FOV 생성 및 FOV 순서를 동시에 표현할 수 있다. 이 때 각 염색체의 윈도우 및 FOV는 검사 동작 수행을 위한 제약조건을 모두 만족시켜야 한다. 한 세대에 대하여

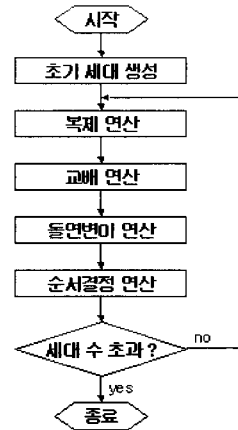


그림 3. 하이브리드 유전자 알고리즘 흐름도.

Fig. 3. Flowchart of hybrid genetic algorithm.

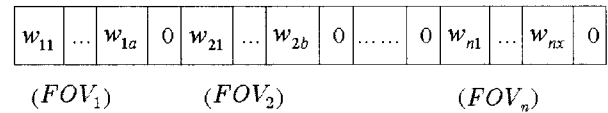


그림 4. 염색체의 정의.

Fig. 4. Definition of chromosome.

$p$ 개의 염색체를 생성시키며, 초기 세대는 확률적으로 랜덤하게 생성시킨다.

각 염색체에 대하여 FOV 생성과 FOV 이동순서가 표현되므로, 총 검사시간을 계산할 수 있다. 어떤 세대의  $k$  번째 염색체  $V_k$  ( $k = 1, \dots, p$ )의 검사시간이  $t_k$  이고, 그 세대에 대한 검사시간의 최대값을  $t_{max}$ 라 하면, 염색체 적합도 (fitness)  $f_k$ 를 다음과 같이 정의한다.

$$f_k = (t_{max} - t_k) \times \frac{p}{\sum_{i=1}^p (t_{max} - t_i)} \quad (11)$$

최대 검사시간을 갖는 염색체의 적합도는 0이며, 검사시간이 감소할수록 적합도는 증가한다. 평균값을 갖는 경우의 적합도는 1이다.

적합도는 염색체의 복제 단계에서 사용된다. 적합도가 높은 염색체는 다음 세대로 복제되며, 낮은 염색체는 도태시킨다. 이 때 확률적 샘플링 오류를 방지하기 위하여 잔여확률샘플링(remainder stochastic sampling with replacement) [13] 방법을 사용한다. 유전자 알고리즘을 구성하는 각 연산자는 다음과 같다.

1. 교배 연산자

교배(crossover)는 염색체에 변화를 주어 새로운 염색체로 진화시키는 과정이다. 교배확률을 사용하여 선택된 염색체  $V_1, V_2$ 에 대하여, 다음의 교배 연산을 수행하여 새로운 염색체  $V'_1, V'_2$ 를 생성한다. 이 때 새로운 염색체가 검사 제한조건을 만족하도록 다음과 같이 정의한다.

- S1. 검색체  $V_1$ 에서 하나의 FOV를 임의로 선택한다.
- S2. 검색체  $V_1$ 에서 선택한 FOV의 윈도우들을 검색체  $V_2$ 에서 모두 찾는다. 이들 윈도우를 포함하는 FOV를 모두 선택한다.
- S3.  $V_2$ 의 선택된 FOV가 포함하는 윈도우들을 검색체  $V_1$ 에서 모두 찾는다. 이들 윈도우를 포함하는 FOV를 모두 선택한다.
- S4.  $V_1$ 과  $V_2$ 의 선택된 FOV가 포함하는 윈도우가 동일할 때까지 S2-S3를 반복한다.
- S5.  $V_2$ 의 선택된 FOV들을  $V_1$ 으로 이동시켜  $V_1'$ 을 얻고,  $V_1$ 의 선택된 FOV들을  $V_2$ 로 이동시켜  $V_2'$ 를 얻는다.

예를 들어, 다음 두 개의 검색체가 선택된 경우,

|       |   |   |   |   |    |    |    |    |   |    |   |    |    |   |   |    |    |   |   |    |    |    |    |   |
|-------|---|---|---|---|----|----|----|----|---|----|---|----|----|---|---|----|----|---|---|----|----|----|----|---|
| $V_1$ | 1 | 3 | 0 | 4 | 5  | 10 | 11 | 15 | 0 | 13 | 0 | 16 | 18 | 7 | 8 | 2  | 14 | 0 | 6 | 9  | 12 | 17 | 0  |   |
| $V_2$ | 1 | 3 | 4 | 6 | 10 | 0  | 18 | 7  | 0 | 8  | 2 | 14 | 0  | 9 | 0 | 13 | 16 | 0 | 5 | 11 | 12 | 15 | 17 | 0 |

$V_1$ 에서 하나의 FOV를 임의로 선택한다.

|       |   |   |   |   |   |    |    |    |   |    |   |    |    |   |   |   |    |   |   |   |    |    |   |
|-------|---|---|---|---|---|----|----|----|---|----|---|----|----|---|---|---|----|---|---|---|----|----|---|
| $V_1$ | 1 | 3 | 0 | 4 | 5 | 10 | 11 | 15 | 0 | 13 | 0 | 16 | 18 | 7 | 8 | 2 | 14 | 0 | 6 | 9 | 12 | 17 | 0 |
|-------|---|---|---|---|---|----|----|----|---|----|---|----|----|---|---|---|----|---|---|---|----|----|---|

선택된 FOV의 모든 윈도우를  $V_2$ 에서 찾고, 이를 포함하는 FOV를 선택하면 다음과 같다.

|       |   |   |   |   |    |   |    |   |   |   |   |    |   |   |   |    |    |   |   |    |    |    |    |   |
|-------|---|---|---|---|----|---|----|---|---|---|---|----|---|---|---|----|----|---|---|----|----|----|----|---|
| $V_2$ | 1 | 3 | 4 | 6 | 10 | 0 | 18 | 7 | 0 | 8 | 2 | 14 | 0 | 9 | 0 | 13 | 16 | 0 | 5 | 11 | 12 | 15 | 17 | 0 |
|-------|---|---|---|---|----|---|----|---|---|---|---|----|---|---|---|----|----|---|---|----|----|----|----|---|

$V_2$ 의 윈도우 13이  $V_1$ 에는 누락되어 있으므로,  $V_1$ 에서 윈도우 13을 포함하는 FOV를 선택한다.

|       |   |   |   |   |   |    |    |    |   |    |   |    |    |   |   |   |    |   |   |   |    |    |   |
|-------|---|---|---|---|---|----|----|----|---|----|---|----|----|---|---|---|----|---|---|---|----|----|---|
| $V_1$ | 1 | 3 | 0 | 4 | 5 | 10 | 11 | 15 | 0 | 13 | 0 | 16 | 18 | 7 | 8 | 2 | 14 | 0 | 6 | 9 | 12 | 17 | 0 |
|-------|---|---|---|---|---|----|----|----|---|----|---|----|----|---|---|---|----|---|---|---|----|----|---|

선택된 윈도우가 모두 동일하므로, 선택된 FOV들을 상대편 검색체로 이동시켜 새로운 검색체를 만든다. 즉

|        |   |   |   |   |   |    |    |    |   |    |   |   |   |   |    |   |   |   |    |    |   |   |    |    |    |    |   |
|--------|---|---|---|---|---|----|----|----|---|----|---|---|---|---|----|---|---|---|----|----|---|---|----|----|----|----|---|
| $V_1'$ | 1 | 3 | 0 | 4 | 5 | 10 | 11 | 15 | 0 | 18 | 7 | 0 | 8 | 2 | 14 | 0 | 9 | 0 | 13 | 16 | 0 | 5 | 11 | 12 | 15 | 17 | 0 |
|--------|---|---|---|---|---|----|----|----|---|----|---|---|---|---|----|---|---|---|----|----|---|---|----|----|----|----|---|

|        |   |   |   |   |   |    |    |    |   |    |   |    |    |   |   |   |    |   |   |   |    |    |   |
|--------|---|---|---|---|---|----|----|----|---|----|---|----|----|---|---|---|----|---|---|---|----|----|---|
| $V_2'$ | 1 | 3 | 0 | 4 | 5 | 10 | 11 | 15 | 0 | 13 | 0 | 16 | 18 | 7 | 8 | 2 | 14 | 0 | 6 | 9 | 12 | 17 | 0 |
|--------|---|---|---|---|---|----|----|----|---|----|---|----|----|---|---|---|----|---|---|---|----|----|---|

이와 같은 교배를 수행할 경우, 검사 제한조건을 만족하는 검색체를 새로이 생성시킬 수 있으며, FOV 총 수와 FOV 이동경로에 변화가 발생된다.

2. 돌연변이 연산자

돌연변이는 교배와 달리 하나의 검색체를 임의로 선택하고 이를 변이시키는 것이다. 교배 연산과 마찬가지로 새로운 검색체가 검사 제한조건을 만족하여야 한다. 돌연변이 확률을 사용하여 검색체  $V_1$ 을 선택하고 다음의 돌연변이 연산을 수행하여 새로운 검색체  $V_1'$ 를 얻는다.

- S1. 검색체  $V_1$ 에서 하나의 FOV를 임의로 선택한다.
- S2. 선택된 FOV의 모든 윈도우에 대하여, 검색체 내의 다른 FOV로 이동 가능한지의 여부를 확인한다. 가능 윈도우를 모두 이동시켜 새로운 검색체  $V_1'$ 을 얻는다.

예를 들어 다음의 검색체가 선택된 경우,

|       |   |   |   |    |    |    |   |    |   |   |   |   |    |   |   |   |   |    |   |    |    |   |
|-------|---|---|---|----|----|----|---|----|---|---|---|---|----|---|---|---|---|----|---|----|----|---|
| $V_1$ | 3 | 0 | 4 | 17 | 11 | 15 | 0 | 13 | 9 | 6 | 2 | 0 | 16 | 8 | 7 | 1 | 5 | 14 | 0 | 10 | 12 | 0 |
|-------|---|---|---|----|----|----|---|----|---|---|---|---|----|---|---|---|---|----|---|----|----|---|

임의의 FOV를 선택한다.

|       |   |   |   |   |   |    |    |    |   |    |   |    |    |   |   |   |    |   |   |   |    |    |   |
|-------|---|---|---|---|---|----|----|----|---|----|---|----|----|---|---|---|----|---|---|---|----|----|---|
| $V_1$ | 1 | 3 | 0 | 4 | 5 | 10 | 11 | 15 | 0 | 13 | 0 | 16 | 18 | 7 | 8 | 2 | 14 | 0 | 6 | 9 | 12 | 17 | 0 |
|-------|---|---|---|---|---|----|----|----|---|----|---|----|----|---|---|---|----|---|---|---|----|----|---|

선택된 FOV에 포함된 각 윈도우 13, 9, 6, 2가, 다른 FOV 범위 내에 포함될 수 있는지를 확인한다. 이동이 가능한 윈도우를 이동시켜, 새로운 검색체  $V_1'$ 를 얻는다.

|       |   |   |   |   |    |   |    |   |   |   |   |    |   |   |   |    |    |   |   |    |    |    |    |   |
|-------|---|---|---|---|----|---|----|---|---|---|---|----|---|---|---|----|----|---|---|----|----|----|----|---|
| $V_2$ | 1 | 3 | 4 | 6 | 10 | 0 | 18 | 7 | 0 | 8 | 2 | 14 | 0 | 9 | 0 | 13 | 16 | 0 | 5 | 11 | 12 | 15 | 17 | 0 |
|-------|---|---|---|---|----|---|----|---|---|---|---|----|---|---|---|----|----|---|---|----|----|----|----|---|

돌연변이 연산은 교배와 마찬가지로 검색체에 변이를 일으켜서, FOV 총 수 및 이동경로를 변화시킨다.

3. 순서결정 연산자

교배 및 돌연변이 연산은 한 세대 중 확률적으로 선택된 일부 검색체에 대하여 수행되지만, 순서결정 연산은 한 세대의 모든 검색체에 대하여 수행된다. 임의의 검색체  $V_1$ 에 대한 순서결정 연산은 다음과 같다.

- S1. 검색체  $V_1$ 의 FOV 중, 첫 번째 FOV를 현재 FOV로 설정한다.
- S2. 현재의 FOV 중심위치를 기준으로, 아직 방문하지 않은 가장 가까운 FOV를 다음 FOV로 설정한다.
- S3. 모든 FOV를 모두 방문할 때까지 S2를 반복하여, 새로운 검색체  $V_1'$ 를 얻는다.
- S4. 검색체  $V_1'$ 의 각 FOV에 대하여, 순서를 교환하여 가장 많은 이동시간의 단축이 이루어지는 FOV를 찾아 순서를 교환한다.
- S5. 모든 FOV에 대하여 더 이상 개선되지 않을 때까지 S4를 반복하여, 새로운 검색체  $V_1''$ 를 얻는다.

순서결정 연산은 순회 외판원 문제의 해를 구하는 전형적 알고리즘과 동일하다. S1-S3는 최근접탐색법 (nearest neighbor search)에 의하여 초기 경로를 구성하는 과정이며, S4-S5는 2-opt 방법에 의하여 초기 경로를 개선시키는 과정이다. 즉, 검색체  $V_1$ 이 다음과 같은 FOV 순서를 갖는 경우,

$$V_1 = (FOV_1) (FOV_2) \dots (FOV_n)$$

S1-S3에 의하여 새로운 FOV 순서를 갖는 검색체  $V_1'$ 를 얻으며,

$$V_1' = (FOV_1') (FOV_2') \dots (FOV_n')$$

S4-S5에 의하여, 검색체  $V_1'$ 을 개선한 검색체  $V_1''$ 를 얻는다.

$$V_1'' = (FOV_1'') (FOV_2'') \dots (FOV_n'')$$

교배 및 돌연변이 연산이 FOV의 총 수와 이동경로를 모두 변이시키는 반면, 순서결정 연산은 이동경로만을 변이시킨다. 국지탐색 방법을 적용하여 FOV 순서를 결정하며, 결과적으로 해를 빠르게 수렴시키는 역할을 한다. 일반적 유전자 알고리즘에 국지탐색 방법을 혼용하는 방법을 하이브리드 유전자 알고리즘 [13]이라 한다. 하이브리드 유전자 알고리즘은 유전자 알고리즘의 장점인 탐색의 다양성을 유지하면서, 수렴속도 문제를 개선시킬 수 있는 실용적 방법이다.

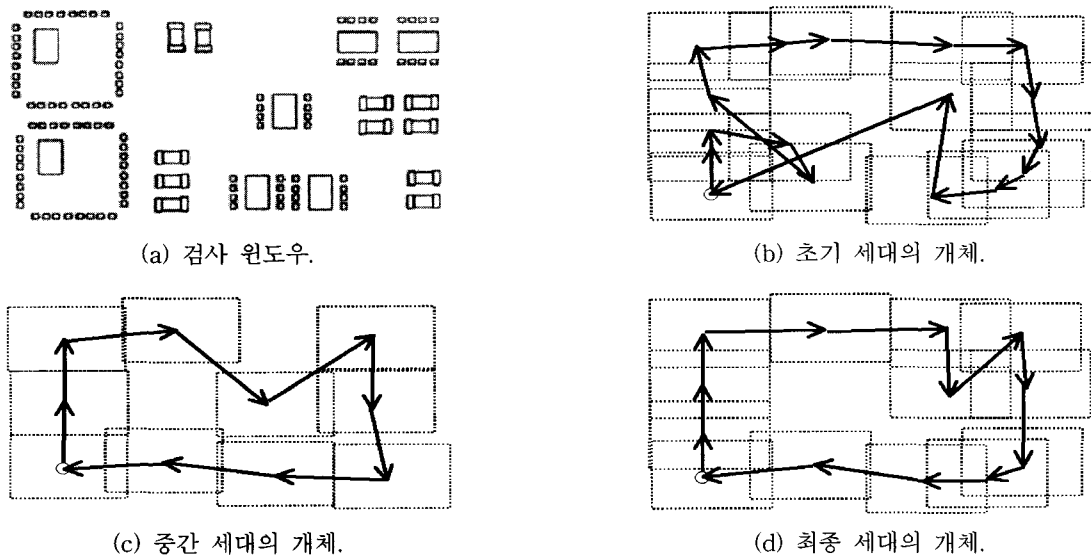


그림 5. 경로계획의 예.

Fig. 5. An example of path planning.

표 1. 방법 별 FOV 수, 이동거리 및 검사시간.

Table 1. Total FOVs, distance and inspection time for methods.

| 기판<br>번호 | 윈도우수 | 계층적 방법 1     |               |               | 계층적 방법 2     |               |                | 통합적 방법       |               |                |
|----------|------|--------------|---------------|---------------|--------------|---------------|----------------|--------------|---------------|----------------|
|          |      | FOV 수<br>(개) | 이동 거리<br>(mm) | 검사시간<br>(sec) | FOV 수<br>(개) | 이동 거리<br>(mm) | 검사 시간<br>(sec) | FOV 수<br>(개) | 이동 거리<br>(mm) | 검사 시간<br>(sec) |
| 1        | 1136 | 79           | 1441.6        | 29.8          | 75           | 1379.9        | 28.3           | 75           | 1344.4        | 27.6           |
| 2        | 1350 | 81           | 2040.9        | 31.6          | 77           | 2012.4        | 30.3           | 77           | 1902.3        | 29.8           |
| 3        | 1578 | 94           | 1627.2        | 35.2          | 91           | 1621.6        | 34.2           | 91           | 1480.4        | 33.5           |
| 4        | 1623 | 99           | 1672.3        | 37.1          | 95           | 1596.9        | 35.5           | 96           | 1561.8        | 34.7           |
| 5        | 1881 | 110          | 1888.3        | 41.1          | 107          | 1853.9        | 40.0           | 104          | 1727.1        | 38.5           |
| 6        | 2049 | 116          | 2426.8        | 44.2          | 110          | 2336.6        | 42.0           | 111          | 2205.2        | 41.1           |
| 7        | 2441 | 145          | 2768.5        | 54.5          | 139          | 2684.2        | 52.0           | 140          | 2545.5        | 50.9           |
| 8        | 3029 | 165          | 2842.5        | 61.7          | 164          | 2762.2        | 60.5           | 163          | 2626.0        | 58.6           |

IV. 시뮬레이션 결과

삼성테크윈(주)의 SMT 검사기 AI-400 [18] 을 대상으로 시뮬레이션을 수행하였다. 카메라의 FOV 크기는 16(mm)×12(mm) 이고, 영상획득시간은 0.25(sec/FOV) 로 설정하였다. 또한 XY 축은 사다리꼴 가감속을 갖는다고 가정하였으며, 최대속도는 700(mm/sec), 가속시간은 0.2(sec) 로 설정하였다. 기판의 최대 크기는 256(mm)×245 (mm) 로 하였으며, 검사 윈도우의 크기는 최소 0.9(mm)×0.25(mm)에서 최대 5.7(mm)×3.9(mm) 로 하였다.

본 논문에서 제시한 통합적 방법의 성능 평가를 위하여 기존에 제시된 계층적 방법 [8] 과 비교하였다. 계층적 방법 1 은 S 자 알고리즘으로 FOV를 생성하고, 최근접탐색법과 2-opt 방법으로 FOV 순서를 결정하는 방법이다. 계층적 방법 2 는 유전자 알고리즘으로 FOV를 생성하고, 최근접탐색법과 2-opt 방법으로 FOV 순서를 결정하는 방법이다. 계층적 방법에 적용된 S 자 알고리즘 및 유전자 알고리즘은 클러스터링 알고리즘의 범주에 속하며, 최근접탐색법과 2-opt

방법은 전형적인 순회외판원 알고리즘이다.

통합적 방법을 하이브리드 유전자 알고리즘으로 구현하였으며, 한 세대에서의 염색체 수는 100개, 교배확률은 30%, 돌연변이확률은 20%로 하였으며, 최대 세대 수는 100 으로 제한하였다. 이 파라미터들은 다양한 실험을 통하여 얻어진 값이다. 모든 알고리즘은 Visual C++ 을 사용하여 프로그래밍 되었으며, MS-Windows XP / IBM-PC Pentium IV 1.5 GHz 상에서 실행되었다.

그림5(a)와 같은 검사 윈도우를 갖는 기판에 하이브리드 유전자 알고리즘을 적용하였다. 그림 5(b)는 초기 세대 중 선택된 개체에 대한 경로계획 결과를 보여준다. 17 개의 FOV를 갖는 검사경로가 생성된다. 그림 5(c)는 10 세대 경과 후, 선택된 개체의 경로계획 결과이다. 13 개의 FOV를 갖는 검사경로가 생성된다. 그림 5(d)는 전 세대 중 가장 좋은 개체의 경로계획 결과를 보여 준다. 10 개의 FOV를 갖는 효과적 검사경로가 생성되었다.

표 1은 방법 별 FOV 수 및 이동거리, 검사 시간을 비교

표 2. 통합적 방법의 성능 개선율.

Table 2. Improvement ratio of the unified method.

| 기관<br>번호 | 윈도우<br>수 | 계층적 방법 1 대비 |             | 계층적 방법 2 대비 |             |
|----------|----------|-------------|-------------|-------------|-------------|
|          |          | 이동거리<br>(%) | 검사시간<br>(%) | 이동거리<br>(%) | 검사시간<br>(%) |
| 1        | 1136     | 6.75        | 6.13        | 2.56        | 2.47        |
| 2        | 1350     | 6.79        | 5.70        | 5.47        | 1.65        |
| 3        | 1578     | 9.02        | 4.77        | 8.71        | 1.82        |
| 4        | 1623     | 6.61        | 4.80        | 2.20        | 2.25        |
| 5        | 1881     | 8.54        | 6.35        | 6.84        | 3.78        |
| 6        | 2049     | 9.10        | 5.70        | 5.59        | 2.14        |
| 7        | 2441     | 8.05        | 4.81        | 5.17        | 2.15        |
| 8        | 3029     | 8.24        | 5.29        | 5.19        | 3.24        |
| 평균 개선율   |          | 7.89        | 5.44        | 5.22        | 2.44        |

하여 보여준다. 다양한 윈도우 수 및 분포를 갖는 8 가지 기관에 대하여 3 가지 방법을 각 각 적용하였으며, 검사시간은 영상획득시간과 이동시간의 총합으로 계산하였다.

계층적 방법 2 는 계층적 방법 1 에 비하여 FOV 수를 감소시키며, 결과적으로 전체 검사시간도 단축시키고 있다. 계층적 방법 1 과 계층적 방법 2 는 FOV 생성단계에서는 FOV 수를 최소화시키고, FOV 순서결정 단계에서는 이동거리를 최소화 시킨다. 이 때 FOV 순서결정 단계에서 동일한 알고리즘을 사용하므로, FOV 수가 전체 검사시간을 좌우하게 된다. 그러나 통합적 방법의 경우, 계층적 방법 2 에 비하여 FOV 수는 증가하나, 전체 검사시간이 단축되는 경우를 확인할 수 있다.(기관 4, 6, 7 번) 그 이유는 통합적 방법의 경우, FOV 수 및 FOV 이동경로를 동시에 결정하기 때문에, FOV 수가 증가하여도 전체 검사시간이 단축되는 경우, 이를 선택할 수 있기 때문이다.

표 2는 다른 방법에 대한 통합적 방법의 개선율을 보여준다. 통합적 방법은 계층적 방법 1에 비해 평균 5.4%의 검사시간 단축을 가져오며, 계층적 방법 2 에 비해 평균 2.4%의 검사시간 단축효과가 있음을 보여준다.

## V. 결론

인쇄회로기판 조립라인에서 납땜 검사 등에 사용되는 SMT 검사기의 새로운 경로계획 방법을 제시하였다. FOV 생성 단계와 FOV 순서결정 단계의 계층구조를 갖는 기존의 계층적 방법에 대하여, 본 논문은 FOV 생성과 FOV 순서결정을 동시에 수행하는 통합적 방법을 제시하였다. 계층적 방법은 각 단계별로 목적함수가 다르나, 통합적 방법은 검사시간 최소화의 단일 목적함수를 갖는다. 특히 검사기 경로계획 문제에서 결정하여야 하는 변수들이 상호 독립적이지 못하므로, 최적화 성능을 높이는 데에 통합적 방법이 보다 유리하다. 제안된 방법의 실용성을 높이기 위하여 국지탐색 방법을 혼용한 하이브리드 유전자 알고리즘으로 제안된 방법을 구현하였다. 실제 장비 및 실제 기관을 대상으로 한 시뮬레이션을 통하여 성능개선 효과를 확인하였다.

휴대폰, 컴퓨터 등 정보통신기기의 수요가 급증함에 따라, 인쇄회로기판 조립라인의 생산성이 매우 중요시되고 있으며, 특히 제품의 불량을 자동 검출하는 SMT 검사기의 사용이 크게 증가하고 있다. 본 논문에서 제시한 SMT 검사기의 경로계획 방법은, 궁극적으로 검사기의 검사시간 단축을 통한 제품 생산성 향상에 기여할 수 있으리라 기대된다.

## 참고문헌

- [1] 박태형, "전자조립용 CAM 시스템의 기술동향", 전자공학회지, 제26권 제3호, pp. 48-61, 1999.
- [2] 최재완, 박경택, "비전 시스템을 이용한 검사 공정 자동화 기술 개발에 관한 연구", 한국자동제어학술회의 논문집, pp. 128-130, 1994.
- [3] T. S. Newman and A. K. Jain, "A survey of automated visual inspection", *Computer Vision and Image Understanding*, vol. 61, no. 2, pp.231-261, 1995.
- [4] Y. Hara, N. Akiyama, K. Karasaki, "Automatic inspection system for printed circuit boards", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 5, no. 6, pp. 623-630, 1983.
- [5] 유창목, 차영엽, 김철우, 권대갑, 윤한중, "비전 시스템을 이용한 J-리드 납땜검사", 한국자동제어학술회의 논문집, pp. 1169-1172, 1997.
- [6] 오제휘, 차영엽, "비전 시스템에서 신경 회로망을 이용한 검사영역에 관한 연구", 제어 · 자동화 · 시스템 공학회 논문지, 제4권, 제3호, pp. 378-383, 1998.
- [7] 이수길, 김화중, 박태형, "인쇄회로기판 검사기의 경로계획 알고리즘", 2002년도 대한전기학회 하계학술대회 논문집, pp. 2439-2441, 2002.
- [8] 김화중, 박태형, "SMT 검사기의 경로계획을 위한 클러스터링 알고리즘", 한국 퍼지 및 지능시스템 학회 논문지, 제13권 제4호, pp. 480-485, 2003.
- [9] A. K. Jain, M. N. Murty, P. J. Flynn, "Data clustering: a review", *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.
- [10] B. S. Everitt, S. Landau, M. Leese, *Cluster Analysis*, Arnold, 2001.
- [11] G. Reinelt, *The Traveling Salesman: Computational Solutions for TSP Applications*, Springer-Verlag, 1994.
- [12] M. Bellmore, G. Nemhauser, "The traveling-salesman problem: a survey", *Operation Research*, vol. 16, pp. 538-558, 1968.
- [13] Z. Michalewicz, *Genetic Algorithms+Data Structures =Evolution Programs*, Springer-Verlag, 1996.
- [14] H. S. Kim and S. B. Cho, "An efficient genetic algorithm with less fitness evaluation by clustering", *IEEE Trans. on. Evolutionary Computation*, vol. 2, pp. 887-894, 2001.
- [15] L. Y. Tseng, S. B. Yang, "A genetic approach to

the automatic clustering problem" *Pattern Recognition*, vol. 34, pp. 415-424, 2001.

[16] J. Gu, "Efficient local search with search space smoothing: a case study of the traveling salesman problem", *IEEE Trans. on Systems, Man and Cybernetics*, vol. 24, no.5, pp. 728-735, 1994.

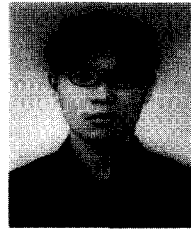
[17] R. Baraglia, J. I. Hidalgo, and R. Perego, "A hybrid heuristic for the traveling salesman problem", *IEEE Trans. on Evolutionary Computation*, vol. 5, no.6, pp. 613-622, 2001.

[18] 삼성테크윈(주), "SMT 검사장비 AI400", <http://www.samsungtechwin.com>.



**김 화 중**

2001년 충북대 제어계측공학과 졸업.  
2004년 동 대학원 제어계측공학과(석사). 2004년~현재 KEC(주) 메카트로닉스 연구소 근무. 관심분야는 최적화 알고리즘, 머신비전.



**정 진 회**

2003년 충북대 제어계측공학과 졸업.  
2003년~현재 동 대학원 제어계측공학과. 관심분야는 최적화 알고리즘, 패턴 인식, 머신비전.



**박 태 형**

1988년 서울대 제어계측공학과 졸업.  
1992년 동 대학원 제어계측공학과 (석사). 1994년 동 대학원 제어계측공학과 (박사). 1992년~1994년 제어계측센터 연구원. 1994년~1997년 삼성 테크윈(주) 정밀기계연구소 선임 연구원.

1997년~현재 충북대학교 제어계측공학과 교수. 2000년~2001년 University of Toronto 방문교수. 관심분야는 로봇틱스 및 자동화, 전자조립 및 검사시스템 등.