# PCB 홀 천공순서의 평가 및 NC코드의 생성*

최후곤** · 이호찬*** · 서준성**

# Evaluation of Tool Paths and NC Codes Generation for PCB Drilling Operations

Hoo-Gon Choi · Ho-Chan Lee · Jun-Sung Seo

〈Abstract〉

The process of determining the optimal tool path in PCB(printed circuit board) drilling operations is identical with that of solving a TSP(traveling salesman problem). However, the optimal solution will be ruined when a drill bit needs tracking back in its tool paths. The back tracking occurrences shorten a life of the main spindle and result in inaccurate mechanical movements. In this study, the performances of four heuristics(Nearest Neighbor, Convex Hull, Greatest Angle and Most Eccentric Ellipse) are evaluated to obtain feasible tool paths along with less number of back trackings for a large number of holes(more than 2000holes/bit) and to generate corresponding NC codes for a given CNC drill. Also, the operations of these algorithms are visualized to show a user the graphic image of tool visitation with PCB holes on a computer screen.

## 1. 서론

진정한 의미의 자동화는 고성능 하드웨어의 구입뿐만 아니라 이러한 하드웨어의 효율을 극대화할 수 있는 운용 소프트웨어의 개발과 하드웨어와 하드웨어간의 인터페이스에 필요한 소프트웨어의 개발을 통해 설비의 효율향상을 도모하여 생산라인을 포함한 기업의 생산성을 향상시키는 것이다. 또한 하드웨어와 소프트웨어가 잘 조화되어야만 설비투자에 대한 기대효과를 충족시킬 수 있다. 이러한 생산 시스템의 개발은 국내 기업들이 겪는 노동력의 변동, 인건비의 상승 및 가격경쟁, 제품의 품질향상 그리고 생산성 향상이라는 당면 문제를 해결할 수 있는 CIM 체제로 나아가기 위한 선결과제이기도 하다. 따라서 자동화 기기의 효율적인 운용 소프트웨어의 개발은 그 중요성이 강조될 수밖에 없고, 이러한 운용소프트웨어의 개발은 국내 제조업체의 체질에 맞게 개발되어야 하고 현장적용이 쉬워야 한다.

본 연구는 홀 천공(drilling)작업에서 다량의 홀을 보다 효율적으로 처리하는 해법에 관한 연구이다. 즉, 천공을 위한 공구가 원점에서 출발하여 모든 홀을 가공하고 다시 원점으로 되돌아오는 경로를 최적화하는

* 본 연구는 1994년도 학술진흥재단의 공모과제 연구비지원에 의하여 수행되었음.

** 성균관대학교 산업공학과

*** LG-EDS 제조 · 통신사업부 EMS사업팀

문제를 다룬다. 이 문제는 기존의 traveling salesman 문제(TSP)와 동일하다. 그러나 TSP에서 최적해를 구하는 문제가 어려운 것은 방문도시의 수가 클 때이고 홀 천공에서는 홀의 수가 대단히 많은 경우이다. 가능한 해의 숫자가 기하급수적으로 커져, 그 중에서 최적해를 결정한다는 것은 현재의 컴퓨터 기술상(기억용량 및 속도제약)으로 거의 불가능하다. 따라서 많은 발견적 해법이 소개되어 사용되고 있다. 이러한 발견적 해법들의 해가 과연 최적해에 얼마나 근접해 있는가 하는 해의 질(質)에 대한 평가도 자주 연구되고 있으나, 이때 연구대상인 홀의 수는 생산현장의 현실(실제로 인쇄기판 천공, 또는 반도체 조립 작업)과는 거리가 있는 약 500개 이내의 숫자이다.

현재 국내의 유수 전자업체에서는 홀 천공을 위한 고가의 소프트웨어를 국외에서 별도로 수입하여 사용하고 있으며, 중소기업체에서는 작업자 또는 생산계획수립자들의 경험에 의하여 홀 천공을 하거나, 제품생산을 의뢰한 대기업으로부터 소프트웨어 또는 NC 테이프를 받아 사용하고 있는 실정이다. NC 코드의 자동생성이라는 분야는 NC의 고정밀화에 따라 편리성, 유연성, 저가성이란 측면에서 중요한 과제로 대두되었으며, 각종 시뮬레이터의 개발에 도입되고 있다. 국내에 개발되어 있는 후처리기(postprocessor)에 걸맞는 프로그램의 개발은 NC기계의 활용을 배가하는데 반드시 필요한 작업이다.

## 2. 선행연구의 검토

Magirou[8]는 TSP의 발견적 해법을 이용하여 드릴링작업의 효율을 증가시켰다. 그의 연구에서는 작업자가 수작업으로 드릴작업을 수행하던 것을 TSP의 발견적 해법들 중에서 Convex Hull해법을 이용하여 드릴작업의 시간을 거의 30%로 감소시켰다. Magirou는 기존의 수많은 해법 중에 아래의 세 가지의 기준으로 Convex Hull해법을 선정하였다.

선정기준 1 : 발견적 해법의 평균 수행도와 최저 수행도(worst case performance)

선정기준 2 : 컴퓨터의 부담 즉, 메모리와 계산시간

선정기준 3 : 컴퓨터 프로그래밍의 수월성

Magirou는 아울러 BASIC 프로그램을 통하여 얻은 경로의 좌표값을 NC코드로 변환하였다. 그러나 Magirou의 연구에서는 몇 가지의 중요한 고려사항이 빠져있다. 첫째, 드릴링작업의 수행도 평가에 있어서 back tracking에 대한 언급이 없다. Back tracking(3절에서 설명)은 드릴의 이동이 모터에 의해 행해질 때 순방향으로 회전하던 모터가 반대방향으로 회전할 때 발생한다. 이런 현상은 정확한 홀의 위치에 드릴의 헤더를 정확하게 위치시킬 수 없게 만든다. 따라서 발견적 해법의 총 이동거리에 대한 결과가 양호하더라도 back tracking의 발생수가 많으면 실제로 인쇄회로기판내의 홀의 위치에 대한 정확도가 떨어져 품질을 급격히 저하시킨다. 둘째, Magirou의 연구에서 홀 가공순서의 사용 가능한 홀의 총수는 500개정도이다. 따라서 홀의 수가 수천, 수만개가 되는 제품의 경우에는 수십개의 부분집합으로 나누어서 처리해야 하는 번거러움이 있다. 그러나 부분집합으로 나누는 수가 증가할수록 구해진 경로는 더욱 국소적인 최적해로 변한다. 셋째, 발견적 해법을 통하여 생성된 경로가 어떤 결함이 있는지의 여부와 실제로 해법을 적용하여 생성된 경로를 NC기계에 로딩(loading)하기 전에 생성된 경로가 실제 드릴링작업에서 합리적으로 실행되는지의 여부를 알아보기 위한 검증이 없다.

한국과학기술연구원 부설 시스템공학센터[12]에서는 드릴링작업을 위해 총 여섯개의 해법들을 제시하였다.

1) Digitizing순서에 의한 경로 결정
2) 수직 수평이동방법에 의한 경로 결정
3) 홀 간 최단거리 이동방법에 의한 경로 결정
4) 정수계획법에 의한 경로 결정
5) 발견적 해법에 의한 경로 결정
6) 동적 계획법에 의한 경로결정

Digitizing에 의한 방법과, 수직 · 수평이동방법, 최단경로에 의한 방법들은 현업에서 주로 사용하고 있는 방법인데 이들을 이 연구에서 각 해법들은 컴퓨터

프로그래밍하여 사용을 편리하게 하였다. 그러나 이들 방법은 최적 공구경로 생성방법은 아니다. 정수계획법, 발견적 해법 및 동적계획법을 이용하면 홀가공 순서 결정에 사용할 수 있다는 방법론만 제시하였을 뿐 실제로 프로그램화가 이루어지지 않아서 실용화가 될 수 없고, 여섯개의 해법에 대한 수행도 평가가 이루어지지 않았다. 실용화 단계까지 가더라도 200~300개 이내의 홀가공순서 결정에만 사용할 수 있으며, 수천개의 홀 가공순서를 결정할 때는 원래의 데이터를 합리적으로 분할하여 이것을 수십번 수행한 후 하나로 합쳐야 된다. 이렇게 경로를 생성한다 할지라도 이것은 수백개로 분할된 데이터에서만 최적일 뿐이고, 전체적인 경로에서의 최적해에 대한 보장이 없다. 그러나 시스템공학센터의 연구는 도면인식(auto-digitizing) 또는 CAD 시스템을 통해 홀의 좌표를 인식하여 저장하고, NC 공구의 최단경로 선택 시스템을 거쳐 직접 NC기계에 전송하는 DNC화를 시도하였다는 점에서 의의가 있다.

Lin과 Kerninghan[7]은 두번에 걸쳐 TSP의 발견적 해법을 연구하였다. 처음에 발표한 해법은 초기해의 경로에 있는 점과 점사이의 링크를 3개씩 교환하여 더 짧은 경로를 찾는 방법에 대한 연구인데, 일반적으로 3-opt[5]이라고 부르고 있다. 이 해법은 초기해를 구한 후 최적해에 근사한 해를 얻고자 할 때 경로 개선절차로서 많이 사용된다. 기존의 연구[1][2][10]에서는 이 해법을 최적해에 가장 근사한 해를 얻을 수 있는 해법이라고 평가했다. 두번째 발표한 해법은 링크의 교환갯수를 3개에서 임의의 k개로 확장시킨 해법이다. 이 해법은 k-opt이라고 부르고 있으며, 한번에 교환되는 링크의 수가 고정되어있지 않기 때문에 한번 교환되는 링크에 의해서 총 이동거리가 감소되는 폭이 3-opt보다 훨씬 크다.

Lin과 Kerninghan[7]은 위의 해법을 이용하여 홀의 수가 100개인 예제들을 수행하여 근사 최적해를 얻었다. 특히 Lin과 Kerninghan은 실제로 드릴의 경로결정을 위하여 105개와 318개의 홀에 대한 경로를 결정하였다. 그러나 105개의 홀을 가진 예제는 근사 최적해를 얻었으나 318개의 홀을 가진 예제는 문제의 크기가 너무 커져 근사 최적해를 얻지 못하였다.

본 연구에서는 2000개 이상의 다량의 홀을 최적해에 가깝도록(문헌조사, 각종 연구보고등에서 참조) 가공할 수 있는 발견적 해법을 조사하고, 이들의 수행도를 가공경로와 back tracking수라는 기준에서 평가한다. 이를 통해 선정된 최적해에 가까운 해법들을 프로그램화하여 평가기준들의 값을 결정한다. 또 공구경로 결정에 있어서 최적성과 컴퓨터 화면상에 결정된 공구경로의 제시를 통한 드릴링작업에서의 여러가지 시각적 모의실험을 해법별로 수행하며, 가장 효율적인 공구경로가 해로 찾아지면 드릴링용 CNC기계를 위한 NC코드를 자동으로 생성한다.

## 3. 해법의 선정기준 및 전제조건

기 개발된 TSP해를 얻기 위한 해법들 중에는 0-1 정수계획법, 동적계획법 등과 같은 최적화이론을 적용한 연구들과 수많은 발견적 해법들이 있다. 최적화 이론의 수리적 모형들은 최적해를 보장해줄 수 있지만, 도시의 수(홀)가 많으면 계산량과 컴퓨터 메모리의 부족 등으로 실용화가 어렵다. 더구나 본 연구에서는 기존의 최적화 이론에서 고려한 홀 수의 10배 이상의 홀을 고려하기 때문에 최적화 이론은 본 연구에서 제외한다. 기존의 연구된 발견적 해법과 그의 변형 해법은 수백개에 이르는데, 본 연구에서는 다음과 같은 기준을 사용하여 4개의 발견적 해법을 선정하였다.

1) 컴퓨터 프로그램이 쉬운 해법 : 아무리 좋은 해법이라도 복잡하고 계산시간이 긴 해법은 PC의 계산시간 및 계산속도와 용량문제 등으로 수행에 어려움이 따른다[2].
2) 기존의 해법 비교 및 평가연구에서 좋은 수행결과를 보인 해법 : 기존의 연구에서는 도시의 수가 100~500이내에서 수행도를 평가하였으나, 그들의 연구결과는 모든 해법을 고려할 수 없는 상황에서 선정기준으로서 의미가 있다[1][2].
3) 계산과정 및 해법실행 과정에서 상대적으로 많은 정보량을 기억해야 하는 해법의 제외 : 계산과정에서 중간과정 계산값의 저장을 많이 요구하

는 해법은 메모리의 제한 용량때문에 많은 도시를 고려할 수 없다.

위와 같은 선정기준에 의거하여 본 연구에서는 Nearest Neighbor, Convex Hull, Greatest Angle Method, Most Eccentric Ellipse Method를 선정하였다.

선정된 해법을 프로그램화하여 수행도를 평가하기 위하여 다음과 같은 전제 조건을 선정한다.

1) 각 홀의 좌표값들을 1차로 수직 · 수평이동방법을 적용하여 좌표값들을 X축, Y축 기준으로 재정리한다. 수직 · 수평이동방법이란 원점을 기준으로 X축 및 Y축을 수직 · 수평으로 이동하며 전체 홀들을 가공하는 방법을 말한다. 즉 Y축홀 → X축홀 → Y축홀 → X축홀과 같이 이동한다.
2) 현업에서 생산되는 인쇄회로기판내의 홀 수는 수천개에서 수만개 이상이다. 따라서 PC의 제한된 용량과 속도내에서 프로그램을 실행하기 위해서는 기판내의 홀들을 그룹으로 분할해야 한다. 즉 전제조건 1)에서 언급된 바와 같이 재정리된 좌표값을 순차적으로 2000개씩 분할하여 프로그램에 적용한다.
3) 하나의 인쇄회로기판내에 있는 각기 다른 홀 지름에 대하여 각각의 홀 지름별로 프로그램이 수행된다.

## 4. 수행도의 평가기준

TSP인 인쇄회로기판의 홀 가공순서 결정을 위하여 해법들을 적용시킨 기존의 연구에서는 생성된 홀간의 거리와 컴퓨터의 계산시간만으로 수행도를 평가하였다. 그러나 이들 연구의 홀 가공순서를 따라가면 모터의 회전축의 회전방향이 어느 홀에서 갑자기 역으로 전환되는 경우(back tracking)가 있다. 이런 경우는 모터의 수명을 단축시키거나, 홀간의 간격이 극도로 짧아 정밀한 위치공차를 요구하는 홀 가공에서는 바람직하지 못하여 실제로 생산현장에서 되도록 제거시키고자 하는 현상이다. 즉 공구가 순차적으로 홀을 가공할 때 공구의 이동은 양(+)의 X축과 양(+)의 Y축으로 이동하여 한개의 홀을 가공한 후 back tracking을 발생시키지 않고, 다음 홀이 있는 지점으로 이동하기 위하여 양(+)의 X축과 양(+)의 Y축을 선택하는 것이 바람직하다. 음(−)의 방향도 마찬가지이다. 만약 X축이나 Y축 중에서 한축이라도 양(+)의 방향으로 이동하다가 음(−)의 방향으로 이동을 하거나 음(−)의 방향에서 양(+)의 방향으로 이동을 하면, 주어진 위치공차를 만족할 수 없는 경우가 발생한다. 따라서 X축과 Y축이 양(+)의 방향에서 음(−)의 방향으로 이동하거나 음(−)의 방향에서 양(+)의 방향으로 이동하는 횟수가 해법들이 산출한 공구경로내에서 적게 발생할수록 바람직하므로 back tracking은 인쇄회로기판 홀 가공순서 결정시 중요하게 고려되어야 할 요소이다.

본 연구에서는 다음과 같은 기준으로 인쇄회로기판의 홀 가공순서 수행도를 평가한다.

1) X축과 Y축에 대한 back tracking의 발생 횟수
2) 전체 이동거리

## 5. 선정된 해법들의 기본개념

### 5.1 Nearest Neighbor[1][2]

Nearest Neighbor해법에 의한 공구경로의 생성은 공구의 원점에서 가장 가까운 거리에 있는 홀을 경로의 시작점으로 한다. 경로내의 두번째 홀부터는 가장 최근에 선정된 홀에서부터 가장 가까운 거리에 있는 홀을 선정하여 경로에 첨가시킨다. 모든 홀이 경로내에 포함될 때까지 이 과정을 되풀이하고 경로내의 시작홀과 끝홀을 연결하여 경로를 완성한다. 이 해법의 수행절차는 다음과 같다.

〈단계 1〉 임의의 홀을 선택한다. 본 연구에서는 원점에서 가장 가까운 홀을 선택한다. 이 홀이 공구경로의 시작점이다.
〈단계 2〉 경로에 첨가된 홀에서 가장 가까운 홀을 찾아서 경로에 첨가시킨다.
〈단계 3〉 단계 2를 모든 홀이 경로에 포함될 때까

지 반복하고 처음의 홀과 마지막 홀을 연결한다.

이 해법의 공구경로 결정을 위한 계산횟수는 홀의 수가 $n$일때, $n^2$회이다[11].

### 5.2 Convex Hull을 구하는 절차

본 연구에서 사용한 해법 중에 Nearest Neighbor를 제외한 Convex Hull해법, Greatest Angle해법, Most Eccentric Ellipse해법의 초기 작업은 convex hull을 구하는 것이다. Convex hull이란 XY 평면에 임의의 좌표값을 가지는 홀들의 가장 외부에 있는 홀들을 찾아서 이 홀들을 연결해 놓은 것으로 정의된다. 나머지 홀들은 convex hull의 내부에 놓이게 되고, 이때 모든 홀들을 싸고 있는 홀들의 집합을 convex hull이라고 한다. 본 연구에서 convex hull은 다음과 같이 구성된다.

〈단계 1〉 홀들의 집합에서 X축의 좌표값이 가장 작은 홀을 선택한다. 이홀은 convex hull상에 있으며 최초의 hull점이 되고 $h_1$으로 표시한다.

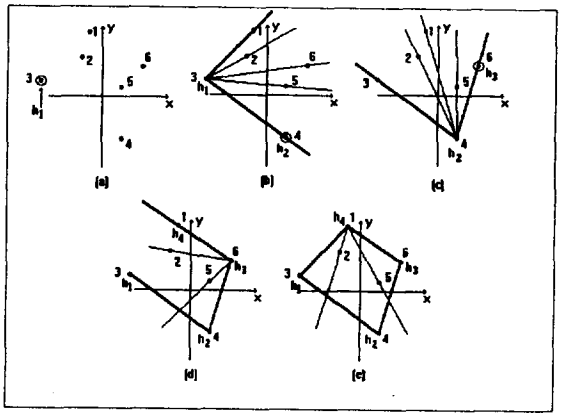
〈단계 2〉 $h_1$을 꼭지점으로 사용하여 이 홀과 다른 홀들에 선을 그어 두 선분사이의 각을 구하고, 구한 각들 중에 가장 큰 각을 가지는 두 선분의 끝에 있는 홀 중에서 임의의 홀을 선택한다. 결정된 홀을 다른 홀들과 구분하여 $h_2$로 표시하고, 이것이 두 번째 hull점이 된다.

〈단계 3〉 $h_1$과 $h_2$를 포함하는 선분을 긋고 이 선분과 $h_2$를 포함하고 다른 한 홀(아직 hull에 포함되지 않은홀)을 연결하는 선분을 그어서 이들 각 중에서 가장 큰 각을 이루는 홀을 선정하고 $h_3$으로 표시한다.

〈단계 4〉 가장 최근에 결정된 hull점을 $h_i$로 하고 이것과 $h_{i-1}$를 포함하는 선분과 $h_i$와 다른 한 홀을 선분으로 연결하여 각을 구하고, 구한 각 중에서 가장 큰 각을 이루는 홀을 선정하는 과정을 필요한 만큼 반복한다.

〈단계 5〉 남아있는 홀들에 대하여 $h_1$이 다음 hull점의 후보가 될 때 홀들의 집합에 대한 convex hull은 결정된다.

위의 과정을 〈그림 1〉로 나타내었다.



〈그림 1〉 Convex hull을 찾는 과정

### 5.3 Convex Hull해법[2]

이 해법은 Golden et. al[2]의 연구에서 속도와 정확도면에서 좋은 평가를 받았다. 이 해법에서는 처음에 convex hull을 구성하고, 남아있는 홀들과 convex hull내의 연속적인 두 홀이 이루는 삼각형의 각 변에 대한 길이를 이용하여 경로를 구한다. 이 해법의 수행절차는 다음과 같다.

〈단계 1〉 홀들의 convex hull을 구성하고 그 hull을 초기의 부분경로(subtour)로 사용한다.

〈단계 2〉 아직 부분경로에 포함되지 않은 홀 $k$를 부분경로내의 홀 $i$와 홀 $j$ 사이에 삽입할 것인가를 결정한다. 즉, 각 홀 $k$에 대하여 $C_{ik}+C_{kj}-C_{ij}$가 최소인 홀$(i,j)$를 찾는다. 여기서 $C_{ij}$는 홀 i에서 홀 j까지의 거리를 나타낸다.

〈단계 3〉 단계 2에서 찾은 모든 홀$(i,j,k)$에서 $(C_{ik}+C_{kj})/C_{ij}$가 최소인$(i^*, j^*, k^*)$를 결정한다.

〈단계 4〉 홀 $i^*$와 홀 $j^*$ 사이에 홀 $k^*$를 삽입한다.

〈단계 5〉 경로 내에 포함되지 않은 홀이 없을 때까지 단계 2에서 단계 4를 반복한다.

### 5.4 Greatest Angle Method[9]

Greatest Angle Method는 Norback과 Love[10]에 의해 연구된 해법으로서 이 해법의 수행도는 Lawler et. al[1]에서 다른 해법들과 비교되어 있다. 이 방법은 먼저 홀들의 집합에 대하여 convex hull을 구한 후 hull내의 연결된 두 홀과 남아있는 홀의 각을 구하여 구한 각 중에서 최대인 각을 hull내의 연결된 두 홀사이에 삽입한다. 이 해법의 수행절차는 다음과 같다.

〈단계 1〉 Convex hull을 구성하여 초기의 부분 경로로 둔다.

〈단계 2〉 남아있는 홀들과 이미 형성된 convex hull내에 존재하는 연결된 두 홀이 이루는 각들을 구한다.

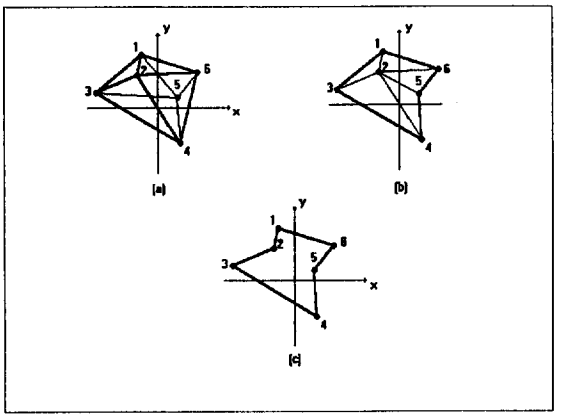
〈단계 3〉 단계 2에서 구한 각들 중에서 가장 큰 각을 이루는 홀점을 선정하여 convex hull내의 연결된 두 홀사이에 삽입한다.

〈단계 4〉 삽입된 홀은 남아있는 홀들의 각을 구하기 위한 홀로 사용된다.

〈단계 5〉 단계 2에서부터 단계 4까지 남아있는 홀이 없어질 때까지 반복한다.

〈그림 2〉는 위의 수행절차를 한 예에 대하여 보이고 있다.

〈그림 2〉에서 초기의 convex hull은 1-3-4-6-1로 구성되어 있다. 〈그림 2〉의 (a)에서는 홀 4, 5, 6번이 이루는 각이 가장 크기 때문에 경로내의 4번홀과 6번 홀사이에 5번홀이 삽입되어 새로운 부분경로는 1-3-4-5-6-1이 된다. 〈그림 2〉의 (b)에서 남아있는 hull내부의 홀들과 각 꼭지점에 해당하는 홀들이 이루는 각들 중에서 가장 큰각을 찾으면 홀 1, 2, 3번이 이루는 각이 선택되고, 새로운 홀 2번은 홀 1번과 홀 3번 사이에 삽입되어 새로운 경로는 1-2-3-4-5-6-1이 된다. 〈그림 2〉의 (c)는 경로가 완성된 후 홀들을 연결한 것이다.




〈그림 2〉 Greatest Angle Method로 경로를 구하는 과정

### 5.5 Most Eccentric Ellipse Method[9]

이 해법은 구성된 convex hull사이에 아직 hull에 포함되지 않은 홀들을 삽입하기 위하여 convex hull내의 연결된 두 홀과 남아있는 홀들이 이루는 타원의 편심율(아래의 수행절차 참조)들을 구한다. 구해진 편심율들 중에서 가장 큰 편심율을 가지는 홀(convex hull에 포함되지 않은 홀 중에서)을 convex hull내의 연결된 두 홀 사이에 삽입하는 과정을 반복하여 경로를 완성한다. 이 해법의 수행절차는 다음과 같다.

〈단계 1〉 Convex hull을 구하여 초기의 경로로 둔다.

〈단계 2〉 Convex hull내의 임의의 연결된 두 홀과 hull내부에 남아있는 홀들이 이루는 타원의 편심율을 계산한다. 편심율 = $d_3/(d_1+d_2)$이고, 여기서 $d_1$, $d_2$, $d_3$는 타원을 이루는 삼각형의 각 변의 길이를 나타낸다.

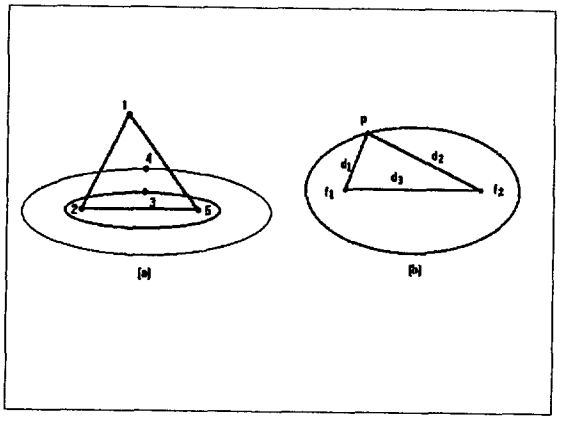
〈단계 3〉 단계 2에서 구한 편심율 중에서 최대 편심율을 가지는 홀(hull내부에 남아있는 홀)을 선택하여 convex hull내의 연결된 두 홀사이에 삽입한다.

〈단계 4〉 삽입된 홀은 남아있는 홀들의 편심율을 구하기 위하여 convex hull을 이루는 홀로 사용된다.

〈단계 5〉 단계 2에서 단계 4까지 남아있는 홀이 없

어질 때까지 반복한다.

위의 절차를 〈그림 3〉으로 나타내었다.

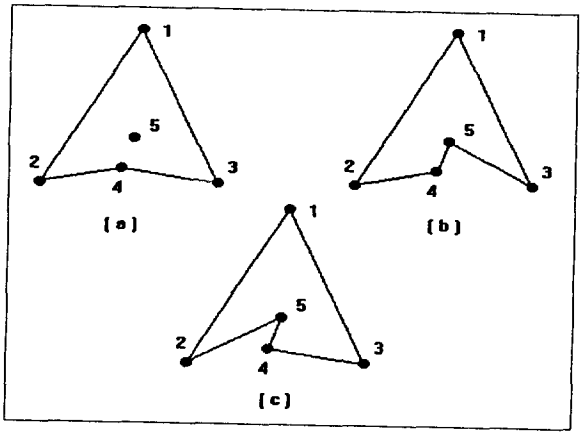


〈그림 3〉 Most Eccentric Ellipse Method와 편심률을 구하는 과정

위의 〈그림 3〉의 (a)에서는 홀 2, 3, 5번이 가장 큰 편심율을 가지므로 최초로 경로에 삽입되는 홀은 3번 홀이다. 그 다음으로 홀 1, 4, 5번이 이루는 타원의 편심율이 크기 때문에 4번 홀이 경로에 삽입되어 경로는 1-2-3-5-4-1이 된다. 〈그림 3〉의 (b)는 기존의 경로내에 두 홀과 추가될 홀 하나가 주어졌을 때 이들 세 홀에 대하여 편심율을 구하는 과정을 나타낸다.

Greatest Angle해법과 Most Eccentric Ellipse해법에 의해 구한 경로는 항상 최적해가 아니다. 특히 〈그림 4〉와 같은 경우가 발생 할 수 있다. 〈그림 4〉의 (a)는 Greatest Angle해법과 Most Eccentric Ellipse해법을 적용하기 전의 모양을 나타내고, 그림 4의 (b)는 Greatest Angle 해법과 Most Eccentric Ellipse해법을 적용한 후의 모습을 나타낸다. (b)에서 생성된 경로 1-2-4-5-3-1의 길이는 (c)에서 생성된 경로 1-2-5-4-3-1의 길이보다 더 긴 것이다.

이런 경우를 처리하기 위한 방안은 두 가지가 있다. 첫째방법은 부분경로를 이웃한 부분경로와 거리를 비교하여 더 짧은 거리의 경로를 기존의 부분경로와 대체하여 전체경로를 형성하는 방법이다. 만약에 더 짧은 거리의 경로를 얻지 못하면 현재의 경로가 최적 경로이다. 두번째 방법은 경로 개선절차를 적용하는 방법이다. 경로 개선절차는 미리 결정된 경로내의 링크들을 더 짧은 경로를 얻을 수 있는 링크로 바꾸어 주는 것으로서 대표적인 경로 개선절차로는 Lin[14][15]의 해법이 있다. 본 연구에서는 부분경로들의 거리를 비교하여 더 짧은 거리의 경로를 전체경로에 포함하는 방법을 사용하는데, 최종적으로 얻을 수 있는 경로는 〈그림 4〉의 (c)에서 보인것과 같이 4번홀과 5번홀의 위치가 바뀐 1-2-5-4-3-1이다.



〈그림 4〉 불합리한 경로의 생성

## 6. 평가기준의 적용

### 6.1 사용 예제에 대한 설명

위에서 선정된 해법들의 수행도 평가를 위해 현업에서 생산하는 인쇄회로기판의 좌표값을 사용하였다. 현업에서는 최적화기법이 아닌 수직 · 수평이동방법에 의해 홀 가공순서를 결정하고 있으며, 인쇄회로기판내의 전체 홀 수가 1000개에서 10000개정도의 홀을 포함하는 기판 중에서 10개를 선정하였다. 홀의 총 수가 100개 이내의 홀을 가지는 홀지름은 모두 제외하였는데, 홀의 수가 100개 정도이면 실용성이 낮아진다는 결점도 있지만, 최적화기법을 사용하여 최적해를 구할 수 있기 때문이다. 본 연구에서 사용한 예제는 가공방법에 따라 D/S, MASS, SLOT으로 구분된다. D/S는 양면 인쇄회로기판을 말하며, MASS와 SLOT은

다층 인쇄회로기판의 가공방법을 말하는데, 가공방법에 따라 공구의 원점이 달라지므로 구분을 하였다. D/S는 공구의 원점이 인쇄회로기판의 X축 정중앙과 Y축 5mm 지점이고, MASS는 X축 정중앙과 Y축 6mm 지점이고, SLOT은 인쇄회로 기판의 X축 정중앙과 Y축 정중앙에 위치한다. 인쇄회로기판의 크기(가로 * 세로, 단위 : mm)는 340*406, 305*406, 510*510, 510*610로 구분된다. 그러나 본 연구에서 발견적 해법의 수행도 평가에 고려되는 요인은 총 홀수이다. 가공방법이나 크기는 단지 사용자의 편의를 위한 공구의 경로를 보여주는 컴퓨터 그래픽 프로그램의 입력데이터로만 사용된다.

### 6.2 해법의 적용

선정된 4개의 해법들은 C언어로 프로그램화되었으며, 선택된 18개의 예제에 적용되어 인쇄회로기판의 홀 가공순서가 결정되었다. 기존의 연구 [1][2][3][4][5][6][7][8][10]에서는 500개 이내의 홀에 대한 가공순서를 결정하였기 때문에 발견적 해법의 프로그램을 쉽게 수행할 수 있었다. 그러나 하나의 주어진 드릴 bit의 지름에 대하여 2000개의 홀에 대한 가공순서를 결정하기 위해서는 PC의 메모리와 속도를 고려해야 하므로 해법들을 프로그램화할 때, 배열(dimension)을 가능하면 사용하지 않고, 대신에 데이터베이스를 이용하여 중간 과정의 정보를 보관하고, 필요할 때마다 정보를 활용할 수 있도록 하였다. 선정된 해법들은 다음과 같은 절차를 따라서 PC에서 실행되었다.

1) 현업에서 얻은 좌표값 데이터를 홀 지름별로 각각 나눈뒤 화일에 저장한다.
2) 프로그램화된 해법을 수행할 때 사용자가 입력해야하는 정보는 홀의 수와 입력 화일명과 출력 화일명이다.
3) 컴퓨터의 실행이 끝나면 결정된 홀 가공순서에 따른 좌표값들이 출력 화일명에 기록된다. 생성된 홀 가공순서에 대한 총 이동거리와 back tracking의 수를 구하는 프로그램을 별도로 실행한다.
4) 출력 화일명과 홀의 수를 컴퓨터 그래픽 프로그램에 입력, 실행하면 드릴이 홀을 가공하는 순서(공구경로)가 화면에 나타난다.

### 6.3. 평가기준의 적용

본 연구에서는 TSP의 발견적 해법인 Nearest Neighbor, Convex Hull, Greatest Angle Method, Most Eccentric Ellipse Method의 해법을 인쇄회로기판에 적용하고 산출된 공구경로를 평가하고자 한다. 평가 기준은 4절에서 결정한 back tracking의 발생 횟수, 공구의 총 이동거리, 각 예제에 대한 총 이동거리 및 홀당 평균 이동거리, 컴퓨터 실행시간이다. 이 중에서 홀당 평균 이동거리 및 컴퓨터 실행시간은 7절의 수행도 평가결과에서 설명하였다.

## 7. 컴퓨터 모델 및 분석결과

본 연구에서 개발한 프로그램은 초기의 입력 데이터를 변환시키는 프로그램, 네개의 해법, 컴퓨터 그래픽 프로그램, 해법 실행후 총 이동거리와 back tracking을 구하는 프로그램으로 모두 일곱개의 모듈로 구성되어 있다. 컴퓨터 그래픽 프로그램이 실행되는 모듈은 기존의 TSP heuristics가 응용된 연구에서는 볼 수 없는 것으로서 다량의 홀을 가공할 때 사용자가 공구의 경로를 천공작업전에 파악하여 공구의 이동상황을 미리 예측할 수 있도록 하며, 공구수명에 따른 공구교환시점을 결정할 수 있도록 공구의 이동거리를 표시해 준다. 이동경로를 화면에 나타내는 시간도 공구이동속도를 변화시킴으로써 조정된다. 아울러 이 모듈의 실행에 의하여 저장되는 공구의 좌표값은 NC코드의 자동생성을 위한 모듈로 전이된다. 이러한 모듈은 TSP알고리즘 또는 heuristics를 현장에 적용하기 위한 가교역할을 담당하는 것이다. 아울러 개발된 프로그램은 사용자위주로 개발되어, 필요한 기업에 공급할 수 있다.

각 해법을 적용할 때는 다음의 4단계를 거친다.

〈단계 1〉 수직 · 수평이동방법의 데이터를 초기 입

력데이터로 변환

〈단계 2〉 각 해법의 실행 및 출력화일의 생성

〈단계 3〉 출력화일을 통한 총 이동거리와 back tracking수 결정

〈단계 4〉 출력화일을 통한 공구경로와 홀의 위치 확인

### 7.1 프로그램의 실행결과

#### 7.1.1 Nearest Neighbor해법

Nearest Neighbor해법은 Golden et . al[2]의 연구에서 예상된 바와 같이 간단한 절차에도 불구하고 좋은 수행도를 보인다. 이러한 결과는 이 해법의 컴퓨터 실행시간은 2000개의 홀에 대해 가공순서를 결정하는데 약 2시간에서 3시간이 소요되었다. Nearest Neighbor 해법을 적용하여 얻은 공구의 총 이동거리가 가장 짧게 나타났다.

#### 7.1.2 Convex Hull해법

Convex hull은 컴퓨터 실행시간이 상대적으로 가장 긴 해법이다. 이것은 해법이 가지고 있는 계산적인 복잡성에 연유하며 계산횟수는 홀의 갯수가 n개 일때, $n^2\log(n)$이다[2]. 본 연구에서는 2000개의 홀 가공순서를 결정하는데 평균 5시간이 소요 되었다. Convex Hull 해법에 의해 생성된 공구의 총 이동거리는 다른 해법에 비해 훨씬 길며, back tracking의 수도 다른 해법에 비해 많음으로 해법의 수행도가 타 해법보다는 좋지 않다.

#### 7.1.3 Greatest Angle해법

Greatest Angle해법은 기존의 부분경로에 남아있는 홀을 경로내에 삽입할 때 convex hull상의 두 홀과 남아있는 홀이 이루는 각들 중에서 가장 큰 각을 택하여 삽입하는 것이 convex hull 해법과 다르다. Greatest Angle해법을 사용하여 구한 공구의 총 이동거리는 수직 · 수평이동방법에 비하여 감소하였다. 특히 Greatest Angle해법으로 구한 경로의 back tracking 횟수는 다른 해법에 비해 월등히 감소하였다. Greatest Angle해법의 실행시간은 2000개의 홀 가공순서를 결정해주는데 평균 40분이 소요되었다.

#### 7.1.4 Most Eccentric Ellipse해법

Most Eccentric Ellipse해법은 convex hull상의 두 홀과 남아있는 홀이 이루는 타원의 편심율을 구하여 가장 큰 편심율을 가지는 것부터 경로에 삽입하는 것이 convex Hull 해법과 다른 점이다. Most Eccentric Ellipse해법을 18개의 예제에 적용한 결과는 총 이동거리와 back tracking의 발생횟수에 있어서 기존의 수직 · 수평이동방법에 비해 좋은 결과를 나타내었다. 이 해법의 실행결과는 Greatest Angle해법과 유사하다. Most Eccentric Ellipse 해법에 의해 2000개의 홀 가공순서를 결정하는데 소요되는 시간은 약 40분에서 1시간이 걸린다.

#### 7.1.5 총 이동거리 및 Back Tracking 수 결정

총 이동거리와 back tracking의 수를 구하는 프로그램은 각 해법에서 얻은 출력 데이터 화일을 입력화일로 입력하고, 홀의 갯수를 입력하면, 즉시 공구의 총 이동거리와 X축과 Y축에 대한 각각의 back tracking의 수를 구할 수 있다.

### 7.2 수행도 평가결과

각 홀당 평균 이동거리는 각 해법을 적용하여 구한 18개 예제의 총 이동거리의 합을 각 예제의 홀 수를 합한 값으로 나누어서 구하였다. 〈표 1〉은 각 해법이 18개 예제에 대하여 X축과 Y축에서 발생하는 back tracking 수의 총합, 홀당 평균 이동거리, 컴퓨터 실행시간을 각 해법별로 보이고 있다.

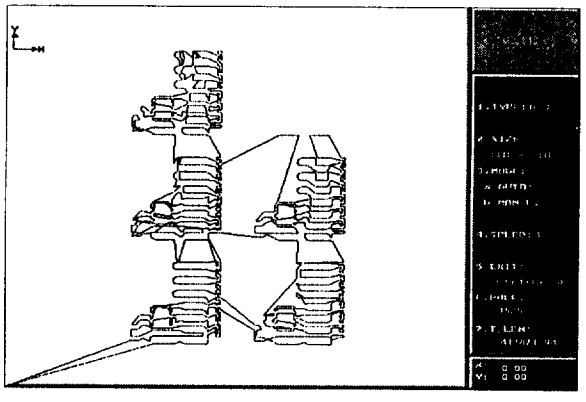
〈표 1〉에서 Convex Hull해법을 제외한 Nearest Neighbor해법, Greatest Angle해법 및 Most Eccentric Ellipse해법은 기존의 수직 · 수평이동방법에 비해서 홀당 평균 이동거리와 총 back tracking의 수에서 감소를 보였으며, Convex Hull해법은 오히려 증가하였다. 따라서 홀당 평균 이동거리의 감소는 Nearest Neighbor해법, Most Eccentric Ellipse해법, Greatest Angle해법 순으로 감소하였으며 Convex Hull해법은 증가하였다. 총 back tracking의 수는 Greatest Angle해

〈표 1〉 수행도 요약(사용된 예제의 총 홀수: 38520개)

| 해법의 종류 | 총 이동 거리(m) | 홀당 평균 이동거리(mm) | Back Tracking 수의 총합 | 컴퓨터 실행시간 (2000개 기준) |
|---|---|---|---|---|
| 수직 · 수평이동방법 | 247.8391 | 6.4341 | 2405 | 0.33 시간 |
| Nearest Neighbor | 175.2254 | 4.5489 | 2028 | 2 시간 |
| Greatest Angle | 219.6592 | 5.7025 | 1721 | 0.75 시간 |
| Convex Hull | 301.4027 | 7.8246 | 2958 | 5 시간 |
| Eccentric Ellipse | 205.0081 | 5.3221 | 1959 | 0.83 시간 |

법, Most Eccentric Ellipse해법, Nearest Neighbor해법 순으로 감소하였으며, Convex Hull해법은 증가하였다. 컴퓨터 평균 실행시간은 수직 · 수평이동방법이 가장 적게 소요되며, Greatest Angle해법, Most Eccentric Ellipse 해법, Nearest Neighbor해법순으로 실행시간이 증가하였다.

각 해법에 의해 결정된 공구경로를 실제 제품을 가공하기 전에 컴퓨터 화면을 통하여 볼 수 있게 하였는데 이것은 사용자에게 기존의 수직 · 수평이동방법에 의해서 생성된 공구의 경로와 각 해법에서 생성된 공구경로의 비교를 쉽게 할 수 있도록 할 뿐만 아니라, 이미 결정된 공구의 경로에 대해서 불합리한 경로의 수정을 용이하게 해준다. 공구경로를 보여주는 컴퓨터의 화면은 크게 두 부분으로 나누어진다. 컴퓨터 화면의 왼쪽부분에서는 공구가 진행하면서 가공하는 홀의 위치와 공구의 이동경로를 보여준다. 오른쪽의 화면은 현재 가공되는 홀의 좌표값과 이미 진행된 공구의 이동거리가 합산되면서 표시된다. 이 화면에 나타나는 또 다른 정보는 가공되는 인쇄회로기판에 대한 주요 사양이 표시된다. 특히 사용자의 편의를 위하여 공구의 이동속도를 조절할 수 있게 하였으며, 키를 칠 때마다 한 홀씩 관측할 수도 있고, 모든 홀을 한꺼번에 연속적으로 관측할 수도 있게 하였다. Greatest angle 해법을 예제(2000개의 홀)에 적용하였을 때 화면에 나타난 공구이동경로를 〈그림 5〉에 나타내었다.



〈그림 5〉 화면에 나타난 홀과 공구의 이동경로 (2000개 홀의 경우)

아울러 각 해법에 의해 결정된 공구경로의 해당 좌표값들은 NC코드자동생성을 위하여 개발된 모듈로 전이된다. 이 모듈의 기본적인 역할은 다량의 홀을 천공하기 위해 많은 양의 NC코드를 지루하게 생성하지 않고 좌표값 입력(컴퓨터 그래픽프로그램모듈로부터)에 의해 자동으로 생성시키는 것이다.

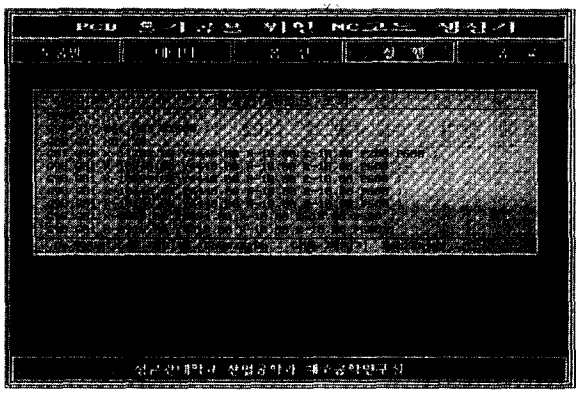
## 8. NC code의 자동생성

PCB(인쇄회로기판) 홀 천공작업을 위한 NC코드 자동생성모듈에서는 결정된 홀 가공 순서 및 좌표데이타를 바탕으로 실제 홀 가공을 위한 NC코드를 생성시켜 준다. NC코드를 수작업으로 작성하거나 APT 등을 이용해 NC코드를 생성시켰던 기존의 방법은 각 코드에 대해 다소 전문적인 지식을 필요로 하며 작성된 코드상의 오류를 발견하는데 어려움이 많았다. 또한 PCB홀 가공을 위한 NC코드 등을 작성하는 경우에는 수천 내지 수만 개의 좌표데이타를 반복적으로 입력하여야 하며, 이러한 작업은 매우 지루한 작업일 수 밖에 없다. 따라서 홀 가공 순서에 따른 좌표데이타

를 기초로 몇몇 기본 데이타의 입력만으로 원하는 홀 가공을 위한 NC코드 자동생성 기능의 개발은 필수적인 것이라 할 수 있다.

본 연구에서 개발된 여덟 번째 모듈인 NC코드 생성기는 머시닝 센타용 CNC인 SNC Series 32[13]를 기준으로 개발되었으며, 화일읽기, 데이타입력, 실행 등의 기능들로 구성되어 있다. 개발된 코드생성모듈은 선정된 알고리즘을 통해 얻어진 홀 가공 순서를 데이타 화일로 입력받은 후 생성될 NC코드의 화일명, 절삭시의 공구이송속도, 주축회전수(rpm), 휴지시간(dwell time) 그리고 실제 PCB 홀 드릴링가공을 위한 드릴의 복귀높이, PCB기판의 두께, 바닥면 여유 등을 차례로 입력한 후 코드생성기를 실행시켜 후처리(postprocessing)과정을 거치도록 하는 작업만으로 원하는 코드를 정확하게 얻을 수 있도록 해준다.

〈그림 6〉은 코드생성모듈 실행 후 생성된 NC코드를 확인할 수 있도록 화면상에 출력된 결과를 보여준다.



〈그림 6〉 PCB홀 가공을 위한 NC코드 생성 결과 화면

생성된 NC코드는 대략 다음과 같은 NC워드로 구성된다.

O : 프로그램 번호로써 프로그램의 선두에 지령한다.
N : 시퀀스 번호, 블럭의 선두에 지령한다.
G : 준비기능으로 동작모드를 제어한다.
F : 공구의 이송속도를 지령한다.
X, Y, Z : 각 축 방향의 이동거리
P : 휴지(dwell) 시간을 지정한다.
M : 보조기능으로 주축 및 프로그램 등을 제어한다.
S : 주축기능을 제어한다.

## 9. 결론 및 추후과제

본 연구에서는 인쇄회로기판 드릴링 작업을 위한 공구의 경로를 결정하기 위하여 TSP의 발견적 해법을 사용하였다. 선정된 해법은 Nearest Neighbor, Convex Hull, Greatest Angle, Eccentric Ellipse이다. 선정된 해법은 C언어를 이용하여 공구경로 생성을 목적으로 프로그래밍되었으며 또한 자동 NC 프로그램 생성을 위한 프로그램이 개발되었으며, 1회 실행에 홀 지름(또는 근사적으로 드릴 bit의 지름)당 최고 2000개의 홀의 가공순서를 결정할 수 있다. 이것은 기존의 발견적 해법을 프로그램화하여 500개의 홀 가공순서를 결정하는 것에 비해 4배 이상의 홀 가공순서를 결정할 수 있으며, 이로 인하여 홀의 수가 많은 인쇄회로기판의 작업을 위한 드릴의 총 이동거리와 back tracking의 발생수를 크게 줄일 수 있는 해법을 결정하였다. 특히, 결정된 공구의 경로는 컴퓨터의 화면을 통해 직접 확인할 수 있으며, 결정된 공구경로를 위한 NC 코드를 자동으로 생성할 수 있다. 또한 선정된 각 해법에 대하여 총 이동거리와 back tracking의 발생수를 기준으로 수행도 평가를 하여 사용자의 필요에 알맞은 해법을 선택하여 실제 생산현장에서 쉽게 사용할 수 있도록 하였다.

18개의 예제를 통하여 얻은 결과를 각 해법별 장단점을 파악하기 위하여, 사용된 전체 예제의 총 홀 수로 18개의 예제에서 얻어진 공구의 총 이동거리를 나눈 각 홀당 평균 이동거리와 18개의 예제에서 발생한 back tracking의 수를 합한 값으로 〈표 1〉과 같은 결과를 얻었다.

결론적으로 사용자에게는 Convel Hull 해법을 제외한 Nearest Neighbor해법, Most Eccentric Ellipse해법, Greatest Angle해법을 추천할 수 있다. 특히 고정밀도를 요구하는 제품에 대해서는 back tracking의 수가 최소인 Greatest Angle해법으로 공구의 경로를 결정해 주면 기존의 방법에 의한 것보다 뛰어난 정밀도를 얻을 수 있을 것이며, 정밀도는 떨어지더라도 생산시간을

감소시키고자 한다면 홀당 평균 이동거리가 최소인 Nearest Neighbor해법을 사용하면 된다.

본 연구에서는 발견적 해법 중에서 최적해에 가장 근사하게 접근한다고 알려진 Lin 등[5][6]의 발견적 해법인 3-opt를 제외하였다. 그 이유는 해법의 복잡도가 너무 높아 PC에서 실행하기에는 엄청나게 많은 계산 시간을 요구하고, 많은 홀을 동시에 고려하기에는 메모리의 부족이 문제되었기 때문이다. 실제로 해법을 코딩하여 홀의 수가 20개인 경우에 적용시킨 결과, 총 이동거리는 수직 · 수평이동방법과 비교하여 약 50% 감소되는 좋은 수행도를 보였다. 그러나 홀의 수가 증가된 경우에는 실행이 불가능하였다. 그러므로 3-opt에 대한 보다 진전된 연구가 필요한 것으로 판단된다.

## 【참고문헌】


[1] Lawler, E. L., Lenstra, J.K, Rinnooy Kan, A. H. G. and Shmoys, D. B., The Traveling Salesman Problem, John Wiley & Sons, 1986.

[2] Golden, B., Bodin, L., Doyle, T. and Stewart, W., "Approximate Traveling Salesman Algorithms", Operations Research, Vol. 28, No. 3, pp. 694-711, 1980.

[3] Held, M. and Karp, R. M., "The Traveling Salesman Problem and Minimum Spanning Trees", Operations Research, Vol. 18, No. 6, pp. 1138-1162, 1970.

[4] Karg, R. L. and Thompson, G. L., "A Heuristic Approach to Solving Traveling Salesman Problems", Management Science, Vol. 10, No. 2, pp. 225-248, 1964.

[5] Kerninghan, B. W. and Lin, S., "An Efficient Heuristic Procedure for Partitioning Graphs", BSTJ 49, pp. 291-308, 1970.

[6] Lin, S., "Computer Solutions of the Traveling-Salesman Problem", BSTJ 44, pp. 2245-2269, 1965.

[7] Lin, S. and Kerninghan, B. W., "An Effective Heuristic for the TSP", Operations Research Vol. 21, No. 3, pp. 498-516, 1973.

[8] Magirou, V. F., "The Efficient Drilling of Printed Circuit Boards", Interfaces, Vol. 16, No. 4, pp. 13-23, 1986.

[9] Morin, T. L. and Marsten, R. E., "Branch and Bound Strategies for Dynamic Programming", Operations Research, Vol. 24, pp. 611-627, 1976.

[10] Norback, J. P. and Love, R. F., "Geometric Approaches to Solving the Traveling Salesman Problem", Management Science, Vol. 23, No. 11, pp. 1208-1223, 1977.

[11] Rosenkrantz, D., Stearns, R. and Lewis, P., "Approximate Algorithms for the Traveling Salesperson Problem", Proceedings of the 15th Annual IEEE Symposium of Switching and Automata Theory, pp. 33-42, 1974.

[12] 한국과학 기술연구원 부설 시스템공학 센터, "최적 TOOL PATH 결정 시스템에 관한 연구", 1991.

[13] SAMSUNG CNC SNC Series 32 조작 및 프로그램 설명서, 삼성전자, 1994.



최후곤

1975년 서울대학교 산업공학 학사

1979년 서울대학교 대학원 산업공학 석사

1981년 Iowa State University 산업공학 석사

1985년 Iowa State University 산업공학 박사

85~89년 Montana State University 산업공학과 조교수

현 재 성균관 대학교 산업공학과 교수

관심분야 CIM, CAPP, Robotics, Manufacturing(machining process)