

레이저 절단공정에서의 토치경로생성과 소프트웨어 개발에 관한 연구

A Study on Torch Path Generation and Software Development for Laser Cutting Process

한국찬*, 나석주**

*, ** 한국과학기술원 기계공학과

1. 서론

레이저 절단시스템을 이용하는 작업자의 생산성은 마이크로 컴퓨터기반의 CAD/CAM 시스템의 도입에 의해 크게 향상될 수 있다. 더욱이 최적 토치경로 생성은 생산시간에 큰 영향을 미치는 요소이다. 최근 많은 상업용 CAD/CAM 패키지는 토치경로의 생성을 제공하여 화염절단, 플라즈마절단, 레이저 절단 등에 다양하게 적용되고 있다. 그러나 대부분의 시스템에서는 작업자의 경험에 의존하는 대화식을 사용하거나 최적경로와는 무관한 순차적인 토치경로를 생성하는 것이 일반적이다^{1,2)}.

본 연구에서는 불규칙적인 부재들이 이미 배치되어 있는 2차원 레이저 절단공정에서 최소의 거리를 가지는 토치경로를 생성하고자 하였으며, 이를 위해 대규모 최적화 문제에 효과적으로 적용되고 있는 시뮬레이티드 어닐링(simulated annealing) 알고리즘을 기반으로 하는 토치경로 알고리즘을 제안하였다. 또한 이를 MS-windows용 응용프로그램으로 개발하였으며, 그래픽 사용자 인터페이스 방식의 부재배치 및 수정, 경로의 생성과 검증을 위한 시뮬레이션까지 제공하도록 하였다.

2. 접근방법

2.1. 형상표현과 데이터 구조

본 연구에서 다루는 부재는 내부 홀을 가지는 다각형으로 각 선분들의 끝점이 반시계방향으로 연결되어있다. 부재를 정의하기 위해 필요한 정보로는 여러 개의 edge와 각 정점들의 좌표이다. Fig.1은 2D 레이저 절단에서의 토치경로 알고리즘의 구현을 위해 개발된 데이터 구조이다. ELEMENT는 부재의 내부 및 외부 윤곽에서 필요로 하는 모든 정보를 포함하는 기본 데이터 구조이며, EDGE는 각 ELEMENT에서 edge-vertex 리스트로서 사용된다. 메모리의 효율적인 사용을 위해 포인터를 사용한 다중 링크 구조를 적용하였다. Fig.2는 Fig.3의 부재에 대한 계층적인 데이터 구조를 나타낸다.

2.2. 토치 경로문제의 수식화

본 연구에서 사용된 가격함수는 토치의 경로길이이다. 부재들의 두 출발점 사이의 경로의 합이 가능한 한 짧을수록 최적의 해에 접근하게 되는데, 이를 수학적으로 표현하면 다음과 같다.

$$\text{minimize } \sum_i \sum_j \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

여기서 x와 y는 부재의 내부 혹은 외부 윤곽의 출발점이다.

실제의 경우 네스팅된 절단 판재는 작업테이블의 평면베드에 놓여있기 때문에 부재를 절단하더라도 절단된 부재가 낙하하지는 않아 부재의 내부 홀을 나중에 절단하더라도 이론적으로는 문제가 되지 않는다. 그러나 열변형과 이동의 문제로 인해 일반적으로 내부 홀을 먼저 절단하는 것이 유리하며, 본 연구에서도 이를 구속조건으로 취급하였다. 이밖에 lead-in 절단에 대해서는 다루지 않았으며 부재의 절단방향은 반시계 방향으로 고정하였다.

2.3. Simulated Annealing 알고리즘의 구현

확률적 상태전이는 각 제어변수(θ)에서 평형상태에 도달할 때까지 계속된다. 또한 제어변수는 냉각규칙에 따라 감소되어 종결조건이 만족될 때까지 반복하게되어 최종적으로 최소의 가격

함수에 따른 최적경로를 얻게되어 있다. 여기에서 SA 알고리즘³⁴⁾은 이웃함수(neighborhood structure), 가격함수, 평가함수 그리고 냉각스케줄 등으로 구성되어 있다. 최적 토치경로 생성을 위해 구현된 SA 알고리즘은 Fig.4와 같다. 여기에서 *iTEMP*는 어닐링이 수행되는 회수이며, *nOVER*는 각 제어변수에서의 최대 반복회수이며, *nLIMIT*는 각 제어변수에서 허용되어야 할 새로운 해의 최대회수이다.

특히 어닐링 스케줄에 있어서 SA의 최종해의 질은 이웃함수와 그 생성 메카니즘에 의해 많은 영향을 받는다. 토치경로 문제에서 이웃함수는 토치경로의 한부분을 재배열하여 얻게된다⁵⁾. 본 연구에서의 이웃함수는 (1)경로의 한 부분을 임의로 선택하여 그부분에 대해서 방향을 반대로 하는 *reversed pathing*과(Fig.5(a)) (2)선택된 경로의 한 부분을 임의의 다른 부분으로 삽입하는 *shifted pathing*과(Fig.5(b)) (3)부재의 내부 혹은 외부의 출발점을 임의로 변경하는 방법(Fig.6)을 적용하였다.

3. 실험결과

Fig.7은 토치경로 문제에 SA을 적용한 최적화 과정을 보여주고 있다. 초기 35개의 부재에 대해 매우 임의적인 경로로부터 시작하여 차츰 최적화 과정이 진행함에 따라 경로의 길이가 감소해감을 알 수 있다. 이때 제어변수는 초기 1.5에서 출발하였으며 각 제어변수에 대해 100N을 시도하였다. 또한 냉각규칙은 $\epsilon_{i,j} = 0.9 \times \epsilon_i$ 이다. 또한 *iTEMP*와 ϵ_s 를 각각 300과 0.001로 하여 하나라도 먼저 도달하면 종결하도록 하였다. Fig.8은 내부 홀을 부재들의 가진 또다른 적용 예를 보여주고 있다.

4. 시스템 개발 및 적용

토치 경로생성 및 검정 시스템은 절단부재의 배치 및 수정과 더불어 소프트웨어로 구현하였다. 이를 위해 IBM PC에서 Borland C++와 Object Windows Library를 이용하여 MS-windows용 응용 프로그램으로 개발하였다. 다중화면분할(multi-windows)이 가능하고 그래픽 사용자 운용체제이기 때문에 초보자라 할 지라도 쉽게 사용할 수 있도록 하였다. Fig.9은 다중분할화면을 통해 동시에 여러 개의 작업을 할 수 있는 것을 보여주며, Fig.10은 절단경로의 시뮬레이션을 2D혹은 3D view로 보여주고 있는 장면이다.

5. 결론

토치경로 문제를 풀기 위해 본 연구에서는 SA알고리즘을 적용하여 국부해가 아닌 전역해를 구할 수 있는 가능성을 향상시켰으며, 효과적인 이웃함수의 생성방법을 적용하여 비구속 최적화 문제를 성공적으로 풀어보았다. 특히 내부 홀이 있는 부재에 대해 적용하여 좀더 실제적인 문제를 해결하고자 하였다. 이와 같은 알고리즘을 현장에서 적용이 가능하도록 개인용 컴퓨터 상에서 그래픽 사용자 운영방식의 소프트웨어로 구현하였다.

6. 참고문헌

- 1 Manber, U. and S. Israni. Pierce point minimization and optimal torch path determination in flame cutting. J. of Manufacturing systems, 3(1), 1984, pp.81-89.
- 2 Jackson, S. D. and R. O. Mittal. Automatic generation of 2-axis laser cutter NC machine program and path planning from CAD. Computer in Industry, 21, 1993, pp.223-231.
- 3 Rutenbar, R. A. Simulated annealing algorithms: an overview. IEEE Circuits and Devices Magazine, 1989, pp.19-26.
- 4 Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. Science, 220(4598), 1983, pp.671-680.
- 5 Lin, S. Computer solutions of the traveling salesman problem. The Bell Syst. Tech. J., 44, 1965, pp.2245-2269.

```

typedef struct edge EDGE
typedef struct elmt ELEMENT
struct edge // Structure of edges
{
    float stx, sty; // Start vertex of edge
    float edx, edy; // End vertex of edge
    EDGE *NextEdge; // Next edge
};

struce elmt // Structure of contours
{
    ELEMENT *NextElmt; // Next element of contour
    ELEMENT *FirstHole; // First hole list of part
    EDGE *FirstEdge; // First edge list of contour
    EDGE *LastEdge; // Last edge list of contour
};

```

Fig. 1 Key data structure declarations.

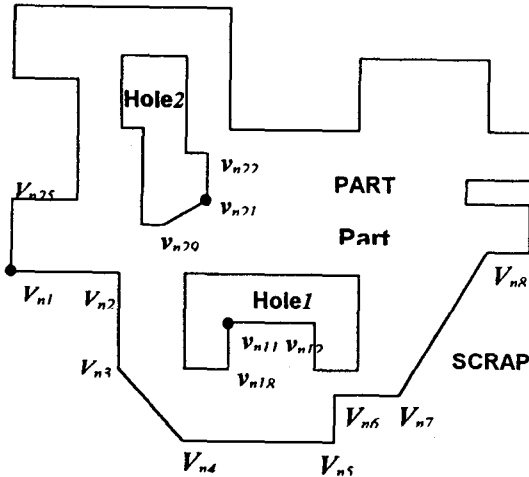


Fig. 3 Definition of irregular part contours.

```

generate random path, X;
while(stopping_criterion(loop_count)<iTEMP and l > e) {
    while(inner_loop_criterion<nOVER and nsucc<nLIMIT) {
        do a shifted pathing, Xi;
        ΔC=cost(X)-cost(Xi);
        if (ΔC<0 or exp(ΔC/l) > R) accept(X=Xi);
        else {
            do a reversed pathing, Xr;
            ΔC=cost(X)-cost(Xr);
            if (ΔC<0 or exp(ΔC/l) > R) accept(X=Xr);
        }
        choose a new piercing point in a outer contour, Xp;
        ΔC=cost(X)-cost(Xp);
        if (ΔC<0 or exp(ΔC/l) > R) accept(X=Xp);
        else {
            choose a new piercing point in a inner contour, Xh;
            ΔC=cost(X)-cost(Xh);
            if (ΔC<0 or exp(ΔC/l) > R) accept(X=Xh);
        }
    }
}
update l;

```

Fig. 4 SA algorithm

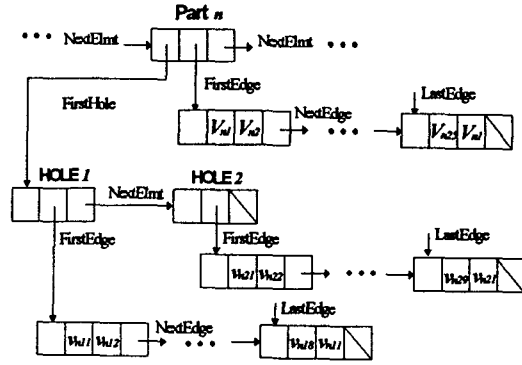


Fig. 2 Hierarchical data

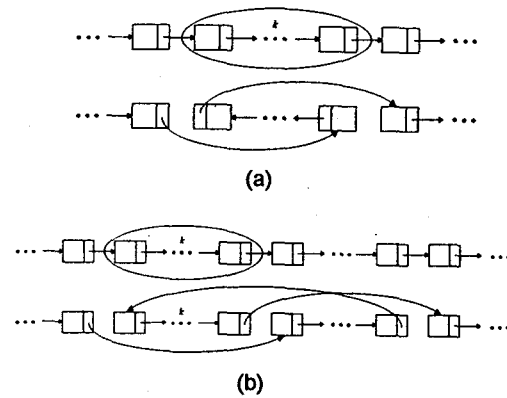


Fig. 5 Cutting path generation mechanism; (a) Shifted pathing (b) Reversed pathing.

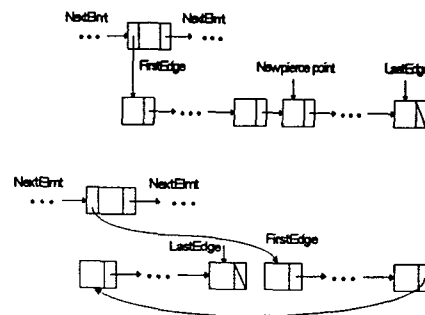


Fig. 6 Rearranging of piercing point.

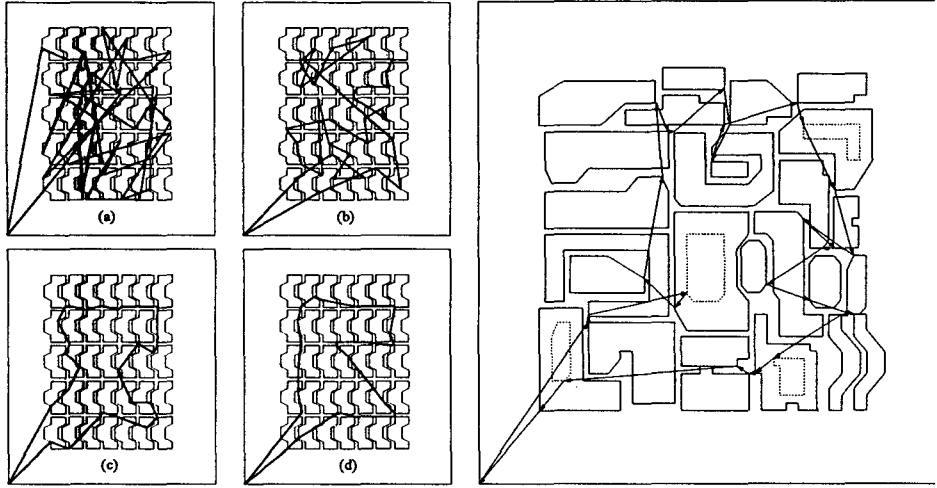


Fig. 7 Intermediate results of SA optimization process with increasing time from (a) to (d).

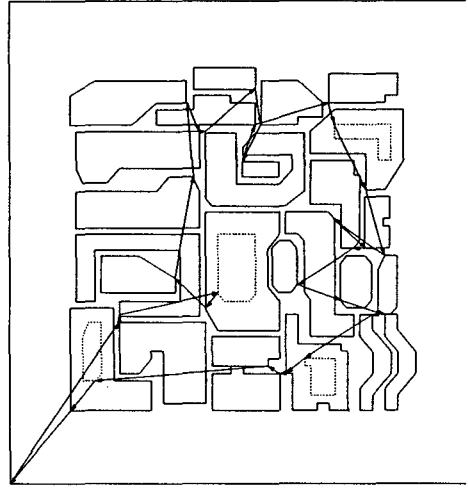


Fig. 8 Example of torch path optimization for irregular parts with holes.

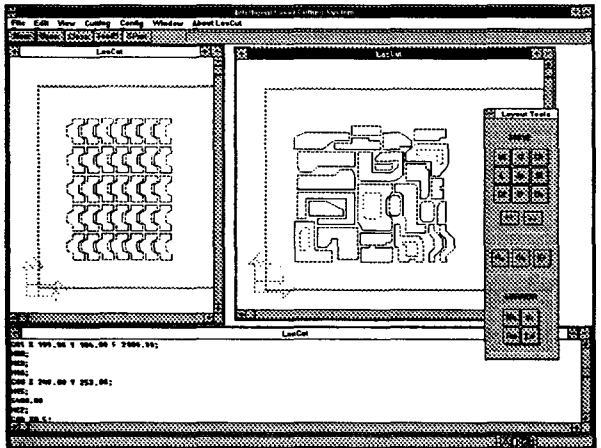


Fig. 9 Multi-windows for different operations

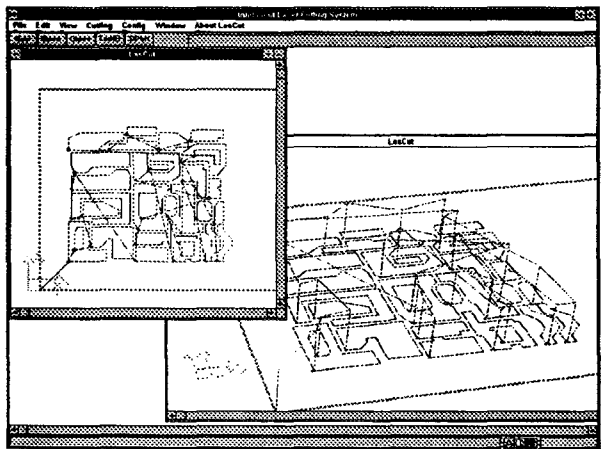


Fig. 10 Co-display of cutting simulation results with 2D and 3D view ports