향상된 그레이팅 배치를 위한 A* 알고리즘의 적용

이해영*, 조대호*

An Application of A* Algorithm for Improved Grating Allocation

Hae-Young Lee, Tae-Ho Cho

Abstract

CAD (Computer-Aided Design) 시스템은 보다 정확한 공학적 분석과 보다 많은 설계 대안을 고려하므로 설계의 질을 증가시킨다. 이러한 과정 중에 수행되는 최적화에는 다양한 탐색 기법들이 사용되고 있다. 그레이팅 설계를 자동화하는 AutoCAD 기반 시스템인 GDS(Grating automatic Drawing System)에서도 다양한 탐색 기법들이 최적화를 위해 사용되고 있다. 실제 그레이팅의 설계 공정에 있어서, 그레이팅은 특정 제약 조건과 우선순위에 맞추어 배치되어야한다. 그러나 현 GDS는 이러한 조건을 만족하는 최적 해를 찾는 것을 보장하지 않는다. 그러므로 본 논문에서는 그레이팅 배치에서 준수해야할 제약 조건과 우선순위를 만족하는 최적 해를 찾기 위한 하용 가능한 A* 알고리즘을 GDS에 적용하며, 시뮬레이션을 통하여 균등 비용 탐색과 상대적인 효율성을 평가한다.

Key Words: A* Algorithm, Heuristic Search, CAD/CAM

* 성균관대학교 컴퓨터공학과

1. 서론

제품의 설계 공정(design process)에 CAD (Computer-Aided Design) 시스템을 사용하는 근본적인 이유 중 하나는, CAD 시스템이 보다 완전한 공학적 분석(engineering analysis)과 더 많은 설계 대안의 고려를 가능하게 하므로, 결과적으로 설계의 질을 증가시키기 때문이다[1]. 보다 높은 질의 설계를 얻기 위한 분석 과정(analysis process)은 반복적으로 수행되는 최적화 속에서 구현되며, 다양한 탐색 기법들이 최적화에 사용되고 있다[2].

그레이팅(그림 1)을 생산하는 S사에서 설계 공정에 사용되고 있는 GDS(GDS; Grating automatic Drawing System)는, 설계자가 지정한 철골(frame)에 맞추어 그레이팅을 자동 배치하고, 배치된 그레이팅의 정보를 바탕으로 세부 도면을 자동으로 생성한다. GDS에서도 설계의 질을 향상시키기 위한 최적화에 다양한 탐색 기법이 사용되고 있다. 그레이팅은 여러 제약 조건을 고려하며 배치되어야 하지만, 현 시스템은 언덕 오르기 탐색(hill-climbing search)[3]과 유사한 기법을 사용하여 제약 조건을 만족하는 최적 해를 찾는 것을 보장할 수 없다.

본 논문에서는, GDS의 그레이팅 배치 문제에서 제약 조건을 만족하는 최적 해를 찾기 위한 허용 가능한(admissible) A^* 알고리즘[4]을 적용하고, 시뮬레이션을 통하여 적용 알고리즘의 효율성을 평가한다.

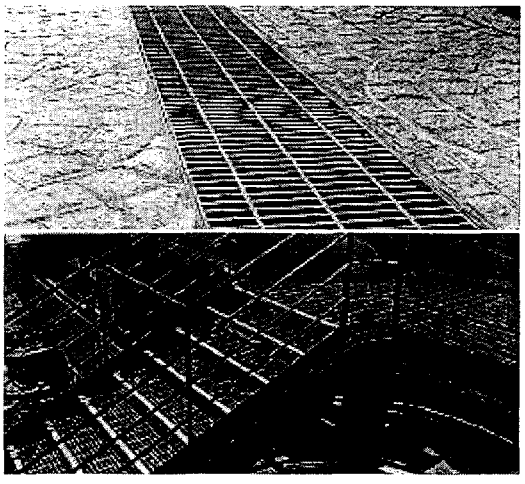

그림 1. 그레이팅

2. GDS에서의 그레이팅 배치

2.1 그레이팅 배치의 제약 조건

GDS는 설계자가 지정한 배치 구획(allocation boundary) 내에 자동으로 그레이팅을 배치한다(그림 2). 배치 구획이 항상 배치 가능(배치 구획 내에 그레이팅 배치 가능)한 것은 아니며, 배치 가능 여부는 배치 구획의 폭 $w(\in \aleph)$에 따라 결정된다. 이와 같은 제약 조건(constraint)은 제조 공정에서의 효율성, 설치시의 용이성 등을 고려하며 그레이팅을 배치해야하기 때문이다(이에 대한 설명은 생략한다).

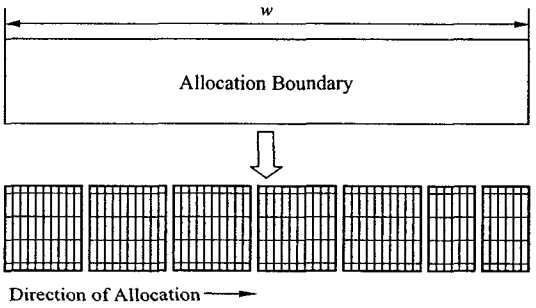

그림 2. 그레이팅의 배치

배치 구획 내에 오픈(open)이 존재하면(그림 3의 ①), 오픈의 중심(centroid)을 기준으로 배치 구획을 분할하고(그림 3의 ②), 분할된 배치 구획 내에 오픈이 없는 경우와 동일한 방법으로 그레이팅을 배치해야한다(그림 3의 ③).

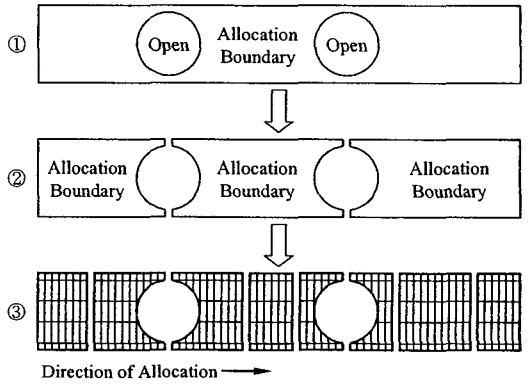

그림 3. 배치 구획의 분할

배치 구획이 하나의 오픈에 의하여 두개의 배치 구획으로 분할되기 위해서는, 오픈의 중심을 지나고 그레이팅 배치 방향 d와 수직인 선을 l, l과 분할된 배치 구획 사이의 거리를 a와 b, 분할된 배치 구획간의 간격(gap)을 $g \in G(G=\{g|g_{min} \le g \le g_{max},\ g_{min},\ g,\ g_{max} \in \aleph\}$으로 허용되는 그레이팅간의 간격 집합)라 할 때(그림 4), a, b, g가 표 1의 분할 조건 중 하나를 만족(분할 가능)해야 하며, 분할된 배치 구획은 배치 가능해야한다. 이러한 제약 조건 역시 제조 공정에서의 효율성과 설치시의 용이함으로부터 기인한다.

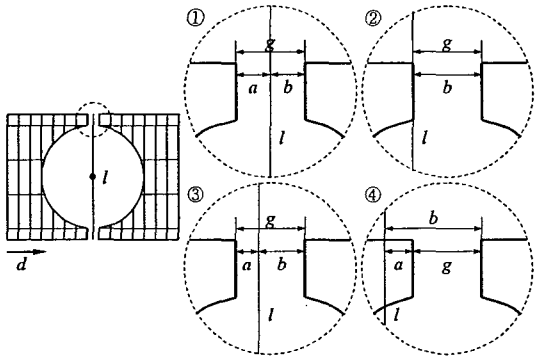

그림 4. 배치 구획의 분할 조건

우선순위	조 건
1	a=b
2	$a=0 \vee b=0$
3	$g=a+b \wedge a \neq 0 \wedge b \neq 0 \wedge a \neq b$
4	$(b=a+g \wedge 1 \le a \le 13) \vee (a=b+g \wedge 1 \le b \le 13)$

표 1. 배치 구획의 분할 조건 및 우선순위

2.2 제약 조건 내에서의 최적 그레이팅 배치

오픈으로 인하여 배치 구획의 분할이 발생할 때, 2.1절의 제약 조건을 만족하는 최적 해는, 오픈에서 높은 우선순위의 분할 조건을 만족하며 분할되는 수가 많은 경우이다(단, 분할된 모든 배치 구획은 배치 가능해야한다). 예를 들어, 배치 구획 내에 분할을 발생시키는 3개의 오픈이 존재할 때, 각 오픈에서의 분할이 분할 조건 1, 4, 4를 만족하는 경우와 2, 2, 2를 만족하는 경우만 배치 가능하다면, 분할 조건 1, 4, 4를 만족하며 분할되는 경우가 최적 해이다.

3. A* 알고리즘을 사용한 최적 해 탐색

3.1 상태 공간 그래프

배치 구획 B 내에 배치 방향 d의 순으로 m개의 오픈 $O_1, \cdots, O_m$이 존재하고, 이로 인해 B가 $m+1$개의 배치 구획 $B_1, \cdots, B_{m+1}$(d 방향순)로 분할된다하면, 시작 노드를 n_0(깊이 0), 목표 노드를 n_{goal}(깊이 $m+1$)로 하는 그림 5와 같은 상태 공간 그래프로 표현할 수 있다. 그래프에서 노드 n은 n의 깊이가 i라 하면, O_i에서의 분할 상태를 나타내는 트리플(triple) $n=(a, b, g)$로 표현할 수 있다(a, b, g는 그림 4 참조). 문제를 단순화하기 위하여, 본 논문에서는 a, $b \in \aleph$이라 가정한다.

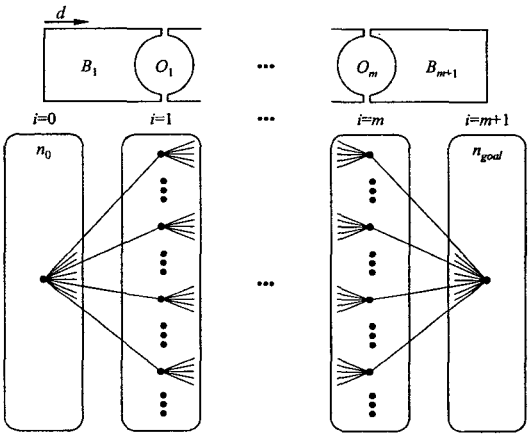

그림 5. 상태 공간 그래프

3.2 휴리스틱 평가 함수

A* 알고리즘에서 휴리스틱 평가 함수(heuristic evaluation function) f^*는 $f^*(n)=g^*(n)+h^*(n)$으로 정의되며, $g^*(n)$은 시작 노드 n_0에서 노드 n까지의 비용이며, $h^*(n)$은 노드 n에서 목표 노드 n_{goal}까지의 휴리스틱 추정 비용이다[4].

노드 n의 부모 노드 중, 함수 g^*의 값이 최소인 부모 노드를 n_{parent}, n_{parent}에서 n까지의 비용(즉, 아크의 값)을 $cost(n)$, n의 깊이를 i, n이 만족하는 분할 조건의 우선순위(priority)를 $p(n)$, 노드 n의 자식 노드들로부터 기대할

수 있는 가장 높은 분할 조건 우선순위를 $p^*(n)$이라 하면, 함수 g^*, $cost(n)$, h^*는 다음과 같다.

$g^*(n)$
- $= 0$ if $n=n_0$
- $= g^*(n_{parent})+cost(n)$ otherwise

$cost(n)$
- $= \infty$ if B_i가 배치 불능
- $= m^4$ if $i \le m$, $p(n)=1$
- $= m^4+m^2$ if $i \le m$, $p(n)=2$
- $= m^4+m^2+m$ if $i \le m$, $p(n)=3$
- $= m^4+m^2+m+1$ if $i \le m$, $p(n)=4$
- $= 0$ otherwise

$h^*(n)$
- $= (m-i)\cdot m^4$ if $0 \le i < m$, $p^*(n)=1$
- $= (m-i)\cdot m^4+m^2$ if $0 \le i < m$, $p^*(n)=2$
- $= (m-i)\cdot m^4+m^2+m$ if $0 \le i < m$, $p^*(n)=3$
- $= (m-i)\cdot m^4+m^2+m+1$ if $0 \le i < m$, $p^*(n)=4$
- $= \infty$ if $i=m$, B_i가 배치 불능
- $= 0$ otherwise

노드 n의 깊이를 i라 하면, 휴리스틱 함수 $h^*(n)$은 오픈 O_{i+1}에서의 기대 비용과 O_{i+2}~O_m에서 분할 조건 1을 만족하며 분할될 것으로 가정한 비용의 합을 반환한다.

노드 n의 깊이가 i-1, 오픈 O_i의 중심을 지나고 배치 방향 d와 수직인 선 l과 배치 구획 B_i와의 최대 거리(즉, n이 표현하는 B_i의 시작 위치와 l과의 거리)를 w_i, 함수 $D(w_1, w_2)$가 w_1+1, …, w_2 ($w_2 \le w_2$) 중 하나 이상이 배치 가능하면 참을, 그렇지 않다면 거짓을 반환한다면, 함수 $p^*(n)$은 다음과 같다.

$p^*(n)$
- $= 1$ if $D(\lfloor w_i-0.5\cdot g_{max} \rfloor, \lceil w_i-0.5\cdot g_{min} \rceil)$
- $= 2$ if $D(w_i, w_i) \vee D(w_i-g_{max}, \lceil w_i-0.5\cdot g_{max} \rceil)$
- $= 3$ if $D(\lfloor w_i-0.5\cdot g_{min} \rfloor, w_i-1)$
- $= 4$ otherwise

3.3 시뮬레이션을 통한 효율성 평가

적용 알고리즘의 효율성을 평가하기 위해 시뮬레이션 기법을 사용하였으며, 기존 GDS의 그레이팅 배치 알고리즘(hill-climbing search)은 적용 알고리즘(A* algorithm)과 비교하기 적절하지 않으므로, 균일 비용 탐색(uniform-cost search)[4]과 비교하였다. 비교 척도로는 닫힌 노드의 수(closed node; 탐색 종료 후 리스트 *CLOSED* 내의 노드)를 사용하였으며, 오픈의 수 m을 1부터 10까지 증가시키면서 시뮬레이션 하였다. 표 1은 m=5, 10(관찰된 최대 값은 10)에서의 시뮬레이션 결과이며, 그림 6, 7은 각각에서 닫힌 노드 수의 히스토그램(histogram)이다.

	m=5		m=10	
	Uniform	A*	Uniform	A*
Min.	357	7	1392	12
Mean	726.73	178.53	1755.24	810.50
Max.	978	729	2022	1795
σ	97.14	176.68	96.95	444.48
Median	742	125.5	1769	825
Mode	807	7	1812	12

표 2. 닫힌 노드 수의 통계

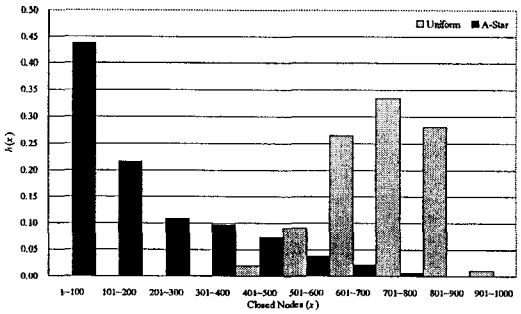

그림 6. 닫힌 노드 수의 히스토그램(m=5)

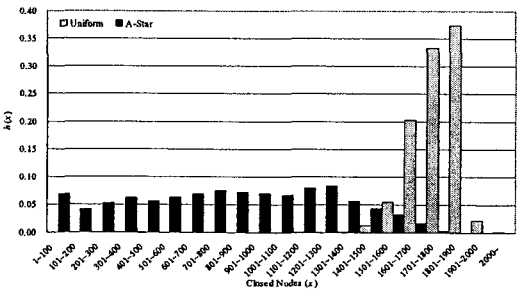

그림 7. 닫힌 노드 수의 히스토그램(m=10)

시뮬레이션 결과, 휴리스틱 함수가 노드 자식들의 비용만을 추정함에도 불구하고, A* 알고리즘이 균일 비용 탐색에 비하여 월등한

효율을 보였다. 탐색 그래프의 깊이가 증가함(즉, 오픈의 수가 늘어남)에 따라 상대적인 효율은 감소하였으나(표준 편차가 증가하여), 관찰된 최대 값 이하에서는 충분히 효율적이었다.

4. 결론 및 향후 과제

적용한 A* 알고리즘이 찾은 해가 제조 공정, 미관 등의 다양한 가능성을 고려하는 최적 해라고는 할 수 없지만, 적어도 2.1절의 제약 조건 내에서는 최적 해이다(그래서 '최적' 그레이팅 배치가 아닌 '향상된' 그레이팅 배치라는 용어를 사용했다). 즉, 본 연구를 통해 제약 조건 내에서 GDS의 설계 품질을 향상시켰다고 할 수 있다.

현 상황에서는 그레이팅 배치에 균일 비용 탐색을 사용하더라도 큰 무리가 없다. 그러나 현재 GDS는 그레이팅 배치시의 모든 요구 사항(즉, 고려 사항과 제약 조건)을 만족하지 못하고 있으며, 차후 이를 만족해야 하는 상황에서 균일 비용 탐색은 문제가 될 가능성이 있다. A* 알고리즘은 확장성이 매우 좋으므로 [5], 휴리스틱 평가 함수의 조정만으로도 현재보다 더 많은 요구 사항(모든 요구 사항은 아니더라도)의 만족이 가능하다. 결국, 본 연구는 GDS의 그레이팅 배치에서 향후의 복잡한 문제를 풀기위한 기초를 다졌다는데 가장 큰 의의가 있다.

참고문헌

[1] Groover, M., and E. Zimmers, *CAD/CAM: Computer-Aided Design and Manufacturing*, Prentice Hall, 1984.

[2] Lee, K., *Principles of CAD/CAM/CAE*, Prentice Hall, 1999.

[3] Pearl, J., *Heuristics: Intelligent Strategies for Computer Problem Solving*, Addsion-Wesley, 1984.

[4] Nilsson, N., *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann, 1998.

[5] Matthews, J., "Basic A* Pathfinding Made Simple," *AI Game Programming Wisdom*, Charles River Media, 2002.