

A* 알고리즘을 적용한 금속 그레이팅 생산 공정에서의 절단문제 해결

김진명*, 조대호*

Application of A* Algorithm to solve a Cutting Problem in Metal Manufacturing Process

Jin Myoung Kim, Tae Ho Cho

Abstract

In a metal grating manufacturing process, the cutting operation allocates the gratings and cut them out from given panels or a plate sheets. Before the cutting operation an operator generates a cutting plan. The cutting plan should decide how pieces of metal rectangles i.e., gratings, are allocated and cut from the panel. This plan generation is a deal of weight on the production cost. the generation of cutting plan is similar to the general two-dimensional cutting problem. In this paper, we first define cutting problem and A* algorithm of Artificial Intelligence to solve the problem. Also, through a simulation, we compare the proposed cutting algorithm to an existing method in terms of material loss

Key Words: Cutting Problem, Combinatorial optimization, Intelligent search

1. 서론

절단 문제는 NP-hard 문제로 잘 알려져 있다. 이것은 순수한 최적의 조합 문제들 중 하나로, 컴퓨터 과학, 산업 공학, 업무 계획, 제조업, 경영 그리고 생산 공정 등 많은 실세계에서 통용된다[1]. 절단 문제는 요구되는 크기의 금속 각형을 절단하거나, 가구를 만들기 위하여 나무 시트를 절단할 때 발생한다. 절단에 의하여 버려지는 자재의 최소량과 관련된 문제는 절단하고자 하는 모든 조각을 가장 작은 면적의 크기로 배치하는 문제와 동치이다.

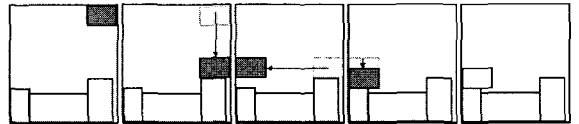
절단 문제는 Kantorovich[2]가 가장 처음으로 다루었고, 이 후에 Gilmore와 Gomory에 의하여 연구 되었다[3, 4]. Christofides와 Whitlock는 트리 탐색 방법을 사용하여 2차원 절단 문제를 최적으로 해결하였으며[5], Beasley 또한 같은 방법을 사용하였다[6]. Beasley는 dynamic programming을 이용하여 최적 알고리즘과 휴리스틱 알고리즘을 비교하였다[7]. Hifi와 Zissimopolous는 Christofides와 Whitlock에 의하여 사용된 해결방법을 개선하여 정확한 알고리즘으로 표현하였다[8]. 최근, Cung는 Hifi와 Zissimopolous가 최상우선 branch and bound 탐색 알고리즘을 이용하여 절단 문제를 해결한 방법을 개선한 새로운 버전의 알고리즘을 개발하였다[1]. Viswanathan과 Bagchi는 최상우선 탐색방법을 2차원 절단 문제에 적용하여 해결하였고[9], Victor Parada Daza는 Wang[10]의 알고리즘을 휴리스틱 방법을 적용하여 향상시켰다[11].

이 논문에서 연구 대상이 되는 그레이팅 (grating) 생산 공정에서의 절단 계획은 패널 (panel)에 원하는 크기의 그레이팅을 절단하기 위하여 배치하는 방법을 계획한다. 이러한 그레이팅의 배치 계획에서 절단 문제와 유사한 문제가 발생한다. 따라서 이 논문에서는 그레이팅의 생산비용을 절감하기 위한 절단 계획의 생성에 대한 방안을 제시한다. 우리는 연구 대상이 되는 문제를 정의하고, 인공지능 탐색

방법 중 하나인 A* 알고리즘을 적용하여 해결한다.

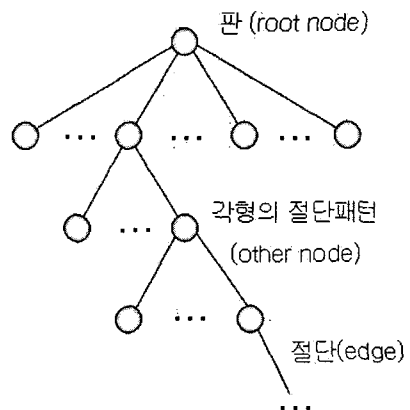
2. 배경연구

Jakobs[12]는 절단해야할 각형의 조각들을 판에 위치시키기 위하여 BLH(Bottom Left Heuristic)을 사용하였다. 배치 전략은 <그림 1>과 같이 오른쪽 위에 위치한 판을 아래로 이동시키고 난 후, 왼쪽으로 이동시킨다.



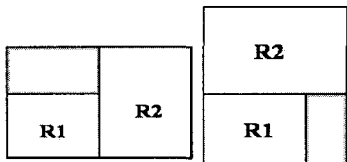
<그림 1> Bottom Left Heuristic

Christofides와 Whitlock은 가능한 모든 절단 탐색(depth first branch and bound) 알고리즘을 제안하였다[7]. 트리의 최상위 노드 (root)는 절단 대상이 되는 하나의 각형 조각이다. <그림2>에서, 트리에서 노드는 원하는 크기의 각형을 표현하며 절단 대상이 되는 판에서의 배치형태를 포함하고 있다. 트리에서의 간선(edge)은 절단과 같다.

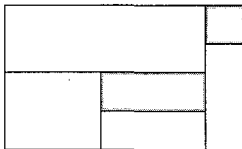


<그림 2> 탐색공간

Wang은 절단하려 하는 조각의 넓이에 비례하는 가중치를 주어 절단 문제를 제한하여 해결하는 알고리즘을 제안 하였다[10]. 이러한 방법은 최적의 해를 보장하지 않는다. 이것은 모든 절단 패턴이 각 각형들의 수직이나 수평적으로 조합되는 것을 기본으로 한다. 두 각형의 수직 조합과 수평 조합은 그림<그림 3>과 같이 이루어진다. 어두운 부분은 손실되는 영역을 나타낸다. 또한 <그림 4>의 어두운 부분은 각형들의 조합에 의하여 생성된 내부 손실(internal waste)이라고 한다.



<그림 3> 수평, 수직 조합



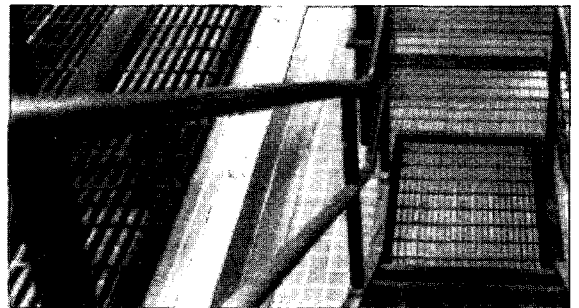
<그림 4> 내부 손실

3. 금속 그레이팅 생산 공정

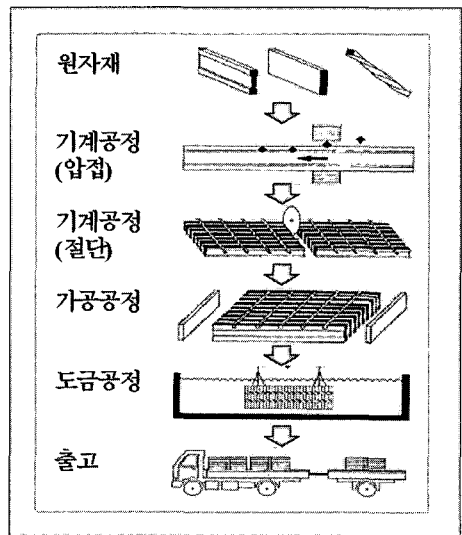
그레이팅은 <그림 5>와 같이 그물 형태의 모양을 가지고 있으며 길가의 배수로에 사용된다. 또한 이것은 선박이나 공사 현장에서 여러 가지 형태의 바닥을 구성하는데 사용된다.

그레이팅의 생산 공정은 <그림 6>과 같은 순서로 이루어진다. <그림 6>의 절단 공정에서는 패널을 원하는 형태의 그레이팅으로 절단하기 전에 절단 계획을 생성한다. 절단 계획은 자재의 손실을 줄이기 위하여 배치 가능한 그레이팅의 조합을 생성하여 선택한다. 따라서 절단 계획은 금속 그레이팅의 생산 공정에서 제품의 생산 비용에 대하여 중요한 역할을 한다.

따라서 우리는 절단 계획에서 발생하는 절단 문제와 그레이팅의 절단과 배치 조건에 대하여 알아보고 A* 알고리즘을 적용하여 효과적인 절단 계획을 생성하는 방법을 제시한다.



<그림 5> 그레이팅의 사용



<그림 6> 금속 그레이팅 생산공정

4. A* 알고리즘의 적용

4.1 A* 알고리즘

그래프의 탐색에서, A* 알고리즘은 경험적 평가 함수를 사용하여 탐색한다[13,14]. 경험적 평가 함수는 현재 노드에서 가장 좋은 다음

노드를 선택하고 확장하는데 도움을 준다. 이 함수는 문제가 되는 대상의 특징 정보를 기반으로 한다.

$h(n)$ 을 노드 n 과 목표 노드 사이의 최단 경로(n 에서 모든 목표 노드까지의 모든 가능한 경로 중에서)의 실제 값이라고 하자.

$g(n)$ 을 시작 노드 n_0 에서 노드 n 까지의 최단 경로의 값이라고 하자. 그럼 수식 (1)은 n_0 에서 목표 노드까지 노드 n 을 통하여 갈 수 있는 모든 가능한 경로 중에서 최단 경로의 값이 된다.

$$f(n) = g(n) + h(n) \quad (1)$$

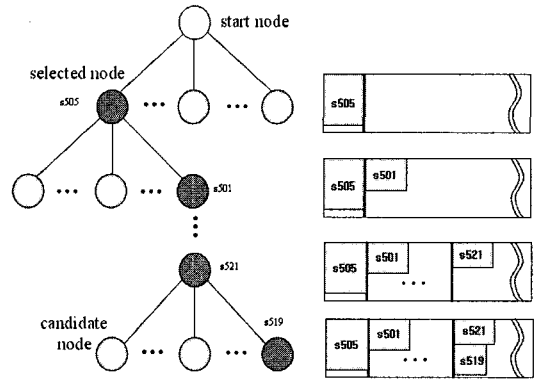
각 노드 n 에 대하여 $\hat{h}(n)$ (휴리스틱 요소)을 $h(n)$ 의 추정값이라고 하고, $\hat{g}(n)$ (깊이 요소)을 $g(n)$ 이 A* 알고리즘에 의하여 지금까지 발견된 노드 n 까지의 경로 중에서 최단 경로의 값이라고 하면, A* 알고리즘에서는 수식 (2)를 평가함수로 사용한다.

$$\hat{f}(n) = \hat{g}(n) + \hat{h}(n) \quad (2)$$

<그림 7>은 A* 알고리즘이 적용된 가상적인 탐색공간을 보여준다. 트리에서 선택 대상이 되는 노드들은 평가함수에 의하여 확장된다.

<그림 7>에서, 회색의 노드는 선택될 가능성이 있는 노드이고, 가장 어두운 노드는 휴리스틱 평가 함수에 의해서 선택된 노드를 나타낸다. <그림 7>의 오른쪽에서 각각의 패널은 <그림 7>의 탐색 트리의 각 레벨에 대한 그레이팅의 배치 상태를 나타낸다. 노드는 그레이

팅을 의미하고 노드와 노드를 연결하는 간선은 절단 계획에서의 자재의 손실을 의미한다.



<그림 7> 탐색 공간과 그레이팅의 배치

4.2 그레이팅의 배치

생산 공정에서 절단대상이 되는 그레이팅의 정보는 GDS로부터 생성되어 입력된다. <그림 8>은 GDS로부터 입력된 정보가 사용자로부터 편집되는 것을 보여준다. 절단 계획의 대상이 되는 작업의 그레이팅 정보는 Item-ID, Drawing number, Type 등으로 구성된다.

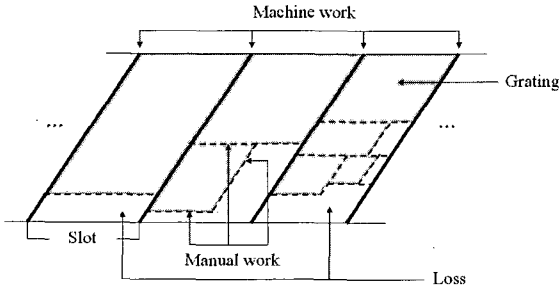
금속 그레이팅 생산 공정에서 절단 계획은 원하는 형태의 크기로 패널을 절단 하여 그레이팅을 생산하기 이전에 계획을 생성한다. 절단 계획은 그레이팅의 배치 조합으로 이루어진다. 따라서 그레이팅 생산 공정에서 발생하는 자재의 손실은 절단 계획 단계에서 결정된다.

<그림 9>는 패널에서 그레이팅의 배치, 절단, 슬롯의 생성 그리고 자재의 손실을 보여준다. 절단 작업은 Machine work와 Manual work로 나누어진다. <그림 9>에서 굵은선(수직 절

	Item ID	Dwng No.	Item No.	Type	Item Name	Spec. ID	End-Bar	Width (mm)	Lgth. (mm)	CB Qty.	LEP (mm)	REP (mm)	Item Qty.	Weight (Kg)
1	200107-001K-0001	Dwg-A1	01	R	Steel Grating	MS-F0190045-002	F0190045-6100	995	840	8	70.0	70.0	8	184.4
2		Dwg-A1	02	R	Steel Grating	MS-F0190045-002	F0190045-6100	903	480	4	90.0	90.0	1	12.0
3		Dwg-A1	03	R	Steel Grating	MS-F0190045-002	F0190045-6100	813	480	4	80.0	100.0	2	21.7
4		Dwg-A1	04	R	Click!	Click!	Click!							
5														

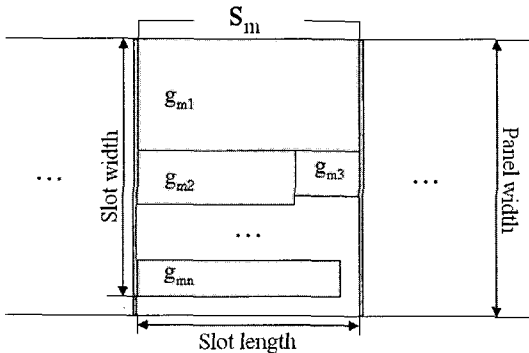
<그림 8> 절단 데이터 입력

단)이 Machine work를 의미하고 점선이 Manual work를 의미한다. 슬롯은 Machine work에 의하여 생성된다.



<그림 9> Machine and manual works

<그림 10>은 슬롯에서의 그레이팅 배치를 보여준다. 임의의 실수 m 에 대하여, 패널에 배치된 그레이팅으로 인하여 생성된 슬롯의 집합을 S_m 이라고 하고, M 을 배치하고 절단해야 할 그레이팅의 전체 집합이라고 하자. 그러면 M 은 다음을 만족한다.



<그림 10> 슬롯에서의 그레이팅 배치

$$M = \{g_k | 1 \leq k \leq p\} \tag{3}$$

$$S \subset M, \cup S_m = M, \cap S_m = \emptyset$$

수식 (3)에서, 집합 M 의 원소의 수는 p 이고, S_m 은 M 의 분할이다.

그레이팅의 배치에 의한 슬롯의 길이는 식 (4)의 조건을 만족해야 한다.

$$g_{mn} = (L_{mn}, W_{mn})$$

$$slotlength \leq panelwidth \tag{4}$$

$$slotlength = Max(L_{m1}, L_{m2}, \dots, L_{mn})$$

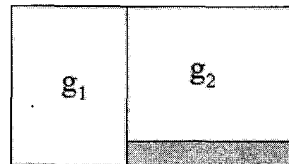
수식 (4)에서, $\{L_{m1}, L_{m2}, \dots, L_{mn}\}$ 의 최대 값은 슬롯에서 가장 먼저 배치되는 그레이팅의 길이이며 <그림 10>에서 g_{m1} 의 길이 값이다. 슬롯 S_m 의 너비(Slot width)는 S_m 에 포함된 모든 그레이팅의 너비의 합에서 서브 그레이팅의 너비를 뺀 것이다. <그림 10>에서 슬롯 S_m 에 포함된 그레이팅 중 서브 그레이팅이 아닌 그레이팅을 W 이라 하고 개수가 i 이면, 슬롯의 너비는 식 (5)와 같다.

$$slotwidth = \sum_{k=1}^i W_{mk} \tag{5}$$

$$slotwidth \leq panelwidth$$

서브 그레이팅은 2개 이상의 그레이팅이 가로 방향으로 조합되어 패널의 슬롯에 배치 될 때, 가장 처음 배치된(가장 큰 너비를 갖는) 그레이팅을 제외한 것을 말한다. <그림 11>에서 g_2 는 g_1 의 서브 그레이팅이며 어두운 부분은 조합으로 인하여 발생한 자재 손실이다.

절단 계획에서 그레이팅의 배치는 수식 (3), (4) 그리고 (5)를 만족해야 한다.



<그림 11> 서브 그레이팅

5. 휴리스틱 평가함수

우리는 절단계획에서 발생하는 절단 문제를 A* 알고리즘을 적용하여 그래프로 표현하고, 절단 계획에서 그레이팅의 절단과 배치의 조건에 대하여 알아보았다. 이 장에서 우리는 절단계획에서 그레이팅의 배치 조건을 기반으로 하여 휴리스틱 평가 함수를 만들겠다.

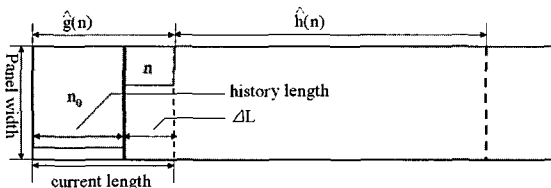
5.1 깊이 요소의 추정

A* 알고리즘에서, 휴리스틱 평가 함수는 수식 (2)에서 정의 되었다. 탐색 대상이 되는 트리에서, p번째 노드 n_p 에 대하여 $g(n_p)$ 는 다음과 같이 정의된다.

$$\begin{aligned} \hat{g}(n_p) &= 0 && \text{if } n_p = n_0 \\ &= (n_{parent}) + cost(n_p) && \text{otherwise} \end{aligned} \quad (6)$$

$$\begin{aligned} cost(n_p) &= \text{current length} - \text{history length} \\ &= \Delta L \end{aligned}$$

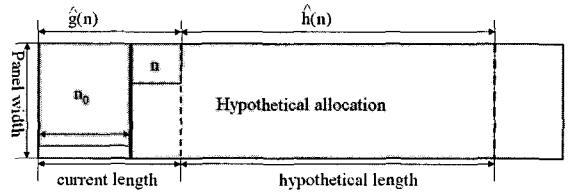
n_0 은 초기노드으로써 어떠한 그레이팅도 배치 되지 않은 상태이고, n_{parent} 는 n_p 노드 이전의 모든 노드에 대한 탐색 비용의 누적을 의미한다. <그림 12>에서 current length는 노드가 배치되었을 때의 길이를 의미하며, history length는 현재 배치하는 노드 이전의 모든 노드들의 배치로 인하여 생성된 길이를 의미한다. 따라서 ΔL 는 cost가 된다.



<그림 12> 그레이팅의 배치

5.2 휴리스틱 요소의 추정

배치하지 않은 그레이팅의 집합을 M 이라 하고, M 의 그레이팅 들이 <그림 12>과 같이 자재의 손실 없이 배치된다고 가정하자.



<그림 13> 휴리스틱 요소의 추정

자재의 손실이 전혀 발생하지 않고 배치되는 경우 모든 슬롯의 길이의 합은 다음과 같다.

$$\text{current length} + \text{hypothetical length} \quad (7)$$

따라서 그레이팅의 배치로 인하여 생성되는 자재의 손실은 $0 \leq loss$ 이므로, 가상적인 배치로 인한 길이 수식 (7)의 값은 항상 실제 배치에 의하여 발생하는 길이보다 작다. <그림 13>에서 패널의 너비(Panel width)는 고정되어 있으므로 가상적인 길이(hypothetical length)를 수식 (8)과 같이 구할 수 있다.

$$\hat{h}(n) = \left\{ \sum_{k=1}^p g_{k.width} \times g_{k.length} \right\} / \text{panelwidth} \quad (8)$$

where

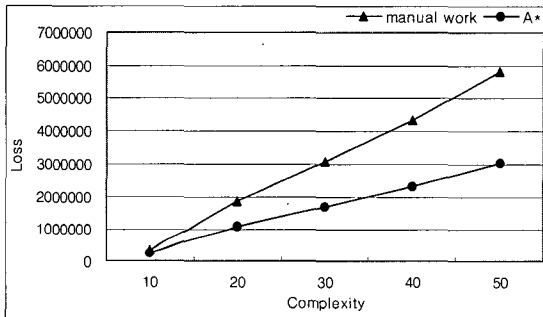
$$M = \{g_n | 1 \leq n \leq p\}$$

자재의 손실이 없이 가상적인 배치가 이루어지기 때문에 수식 (8)에서 해야 하는 그레이팅의 집합의 모든 원소의 넓이의 합을 패널의 길이로 나누게 되면 가상적인 길이(hypothetical length)를 구할 수 있게 된다.

6. 시뮬레이션 결과

절단 계획에서 우리는 시뮬레이션을 통하여

기존의 방법과 A* 알고리즘이 적용된 방법을 비교 평가 하겠다. <그림 14>는 절단 계획에서 기존의 방법과 A* 알고리즘을 적용한 것을 자재의 손실을 기준으로 하여 시뮬레이션을 수행한 결과 이다. <그림 14>를 통하여 제안된 방법이 기존의 것보다 자재의 손실이 작은 것을 알 수 있다. 이것은 A* 알고리즘이 적용된 절단 계획이 보다 효율적으로 그레이팅을 배치하여 계획이 생성된 것을 알 수 있다.



<그림 14> 자재의 손실

<그림 14>에서 복잡도가 증가 할수록 각각의 방법에 의하여 생성된 절단 계획에서 발생하는 자재 손실의 차이가 증가한다.

7. 결론

절단 계획은 그레이팅 생산 공정에서 중요한 부분을 차지한다. 이전의 절단 계획은 정해진 알고리즘이 없이 수작업에 의하여 생성되어, 절단 계획에 대한 효율성과 신속성이 결여되어 있다.

이 논문에서 우리는 금속 그레이팅 생산 공정의 절단 계획에서 계획을 생성할 때 발생하는 절단 문제와 이에 대한 적절한 해결 방법에 대하여 연구하였다. 우리는 절단 계획을 그래프로 표현하고, A* 알고리즘을 적용하여 가장 좋은 절단 계획을 생성하는 방법에 대하여 알아보았다. 또한 시뮬레이션을 통하여 A* 알고리즘이 적용된 방법이 기존의 방법보다 생

산 공정에서 자재의 손실이 감소하는 것을 보였다. A* 알고리즘의 확장성이 대단히 좋다. 따라서 휴리스틱 평가 함수의 개선만으로 더 효율적인 결과를 얻을 수 있다[15]. 제품의 생산 비용에 영향을 주는 요소는 자재의 손실 이외에도 존재하기 때문에 이에 대한 연구가 필요하다.

참고문헌

- [1] Van-Dat Cung, Mhand Hifi, Bertrand Le Cun, "Constrained two dimensional cutting stock problems best first branch and bound algorithm", *Op Res.* 7, pp. 185-210, 2000.
- [2] L. V. Kantorovich, "Mathematical method of organizing and planning production", *Mgmt Sci.*, pp363-442, 1960.
- [3] P. C. Gilmore, R.E. Gomory, "Multistage cutting problems of two and more dimensions", *Op Res.*, pp94-119, 1965.
- [4] P. C. Gilmore, "Cutting stock, linear programming, knapsacking, dynamic programming and integer programming, some interconnections", *Ann. Discrete Math.*, pp217-235, 1979.
- [5] N. Christofides, C. Whitlock, "An algorithm for two dimensional cutting problems", *Operations Research*, pp30-44, 1977.
- [6] J. E. Beasley, "An exact two dimensional non-guillotine cutting tree search procedure", *Oper. Res.*, pp49 - 64, 1985.
- [7] J. E. Beasley, "Algorithms for unconstrained two-dimensional guillotine cutting", *J. Oper. Res. Soc.* 36(4) 297 - 306 1985.
- [8] M. Hifi, V. Zissimopoulos, "Constrained two dimensional cutting: An improvement of Christofides and Whitlock's exact algorithm", *J. Oper. Res. Soc.* pp324 - 331, 1997.

- [9] L. V. Viswanathan, "Best-first search methods for constrained two dimensional cutting stock problems", *Operations Research* Vol. 41, 1993.
- [10] P. Y. Wang, "Two algorithms for constrained two dimensional cutting stock algorithms", *Operations Research*, pp. 573-586, 1983.
- [11] Victor Parada Daza, et al., "Exact solutions for constrained two dimensional cutting problems" *European Journal of Oper. Res.*, pp633-644, 1995.
- [12] S. Jakobs, "On genetic algorithms for the packing of polygons", *Eur. J. Oper. Res.*, pp165-181, 1996.
- [13] Nils J. Nilsson (1998), *Artificial Intelligence: A new synthesis*, Morgan Kaufmann Publishers.
- [14] Kevin Knight (1991), *Artificial Intelligence*, McGraw-Hill
- [15] J. Matthew, "Basic A* Path finding Made Simple," *AI game Programming Wisdom*, Charles River Media, 2002

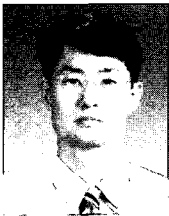
주 작 성 자 : 김 진 명
 논문 투고일 : 2005. 09. 07
 논문 심사일 : 2005. 09. 21(1차), 2005. 11. 01(2차),
 2005. 11. 04(3차)
 심사판정일 : 2005. 11. 04

● 저자소개 ●



김진명

2004 공주대학교 자연과학대학 응용수학과 학사
 2005 성균관 대학교 공과대학 컴퓨터공학과 석사과정
 관심분야: 모델링 및 시뮬레이션, 지능제어, 키 관리



조대호

1983 성균관대학교 전자공학과 학사
 1987 University of Alabama 전자공학과 석사
 1993 University of Arizona 전자 및 컴퓨터공학 박사
 1993 ~ 1995 경남대학교 전자계산학과 전임강사
 1995 ~ 1999 성균관대학교 전기전자 및 컴퓨터공학부 조교수
 1999 ~ 2004 성균관대학교 전기전자 및 컴퓨터공학부 부교수
 2004 ~ 현재 성균관대학교 전기전자 및 컴퓨터공학부 교수
 관심분야: 모델링 및 시뮬레이션, 네트워크 보안, 지능 제어, ERP