

판재 최적절단 시스템에 관한 연구  
- Investigation of Optimization Nesting  
Systems on a Board -

이 장 규\* · 이 선 곤\*\* · 조 대 희\*\*\* · 김 봉 각\*\*\*\*

Rhee Zhang-Kyu\* · Lee Sun-Kon\*\* · Jo Dae-Hee\*\*\* · Kim Bong-Gak\*\*\*\*

**Abstract**

This paper investigates the optimal nesting system for a board. A hybrid method is used to search the optimal solution for rectangular nesting problem. This method is composed of heuristic approach algorithm. An engineer's experience of board nesting in which a loss occurred to sheet because of various individual error and diffidence. So, item layout at resource sheet were evaluated in engineering algorithm logic in which specially designed was installed. The nesting system consists of Lisp and Visual Basic. The system was controlled by AutoCAD so as to best item batch path test.

**Keywords : Rectangular Nesting, Scrap, Optimization, Heuristic Algorithm**

---

\* 인천대학교 기계공학과

\*\* 유한대학 기계과

\*\*\* 인천대학교 교육대학원 기계교육전공

\*\*\*\* 충청대학 항공자동차기계학부

## 1. 서 론

판재를 소재로 다루는 조선, 중공업, 철판, 섬유, 피혁, 유리, 가구 및 합판 등의 원자재(resource sheet, 원판)의 비용이 생산비의 절반이상을 차지하고 있는 제조 분야에서 생산공정의 고도화도 중요하지만, 원자재의 버림률(scrap or loss ratio, 스크랩)을 최소화 하는 네스팅(nesting)은 재료의 경제적인 활용과 이에 따른 생산시간 단축을 통하여 생산성을 높이는 의미에서 경제적인 활용도가 생산공정을 단축하는 것만큼 중요하다.

최근에는 생산단가의 절감, 자재구매 및 관리의 신뢰성 확립 차원에서 부품의 효율적인 배치 알고리즘에 관한 연구가 기계·섬유 등의 분야뿐만 아니라 전자부품의 고밀도화, 고집적화 설계 등에서도 이루어지고 있다. 이러한 이유로 부품의 자동 배치와 절단을 할 수 있는 시스템에 관한 연구가 필요하다.

국내 중소기업체에서는 네스팅시스템에 대한 필요성은 인정하면서도 경제적인 이유와 사용상의 어려움 등 때문에 대부분 수련공의 경험에 의존하고 있는 실정으로 이에 따른 생산성 저하 및 기술 인력 수급에 관한 문제성이 제기되고 있다.

본 논문에서 연구한 판재 네스팅이란 부품(items)의 최적배치를 문제로 원판을 최대한 활용하여 버림률을 최소화 하면서 많은 수의 부품을 절단할 수 있는 최적 절단 배치를 목표로 하는 것을 의미한다. 일반적으로 사각 부품의 패턴을 네스팅하는 배치 알고리즘은 한 원판 상에 단일 부품만을 네스팅하는 방법으로 부품의 크기나 형태에 따라 많은 스크랩(waste board)이 발생하는 문제가 있다면, 본 논문에서 제시하는 새로운 배치 알고리즘은 사용 가능한 다수의 원판 과 다수의 부품에서 최적 배치 형태를 찾아내는 방법이다. 이는 네스팅 후 남은 스크랩 부분도 원판으로 재등록하여 다시 최적 배치를 찾아주는 알고리즘의 연속적인 루핑(looping)으로 버리는 원자재 즉, 스크랩의 양을 최소화하여 원자재의 효율성을 극대화 시키는 배치 알고리즘이다.

평판 형태의 원자재에서 스크랩을 최소화하고 효율적인 원자재 관리를 위한 절단계획을 수립하기 위하여 Kirkpatrick 등에 의하여 개발된 조합최적화문제(combinatorial optimization problem)의 해를 구하는 풀림모사기법(SSA, simulated annealing algorithm)의 응용과 이에 몇 가지의 효율적인 경험법칙(heuristics)들을 고안하여 절단배치 기술을 구현하였고, 이를 위해 Lisp 및 Visual Basic을 이용한 Nesting Logic 프로그램을 디자인하여 AutoCAD로 자동배치를 구현하였다.

## 2. 네스팅 구현방법의 분류

네스팅은 설계가 끝난 각 부품들의 절단작업 이전 단계에 이루어지는 절단계획으로 제조업체에서 부품을 제작하기에 앞서 원판인 사각형의 판에서 부품을 제작할 수 있는 크기로 자르는 계획을 세우는 것이다.

## 2.1 다단계 접근법

본 연구에서는 빠른 시간 내에 전체적인 배치윤곽을 결정하는 1단계 초기배치와 얻어진 초기배치의 구조를 왜곡시키지 않으면서 최적배치에 가까운 해를 얻어나가는 2단계 상세배치의 반복 Logic으로 구성된 다단계 접근법으로 프로그램을 구성하였다. 1단계 배치는 Kohonen 모델과 Fuzzy C-Means(FCM) 알고리즘의 함수를 결합한 SOAL(self organization assisted layout) 알고리즘의 응용에 기반을 두었으며, 2단계에서는 폴립모사에 의한 배치방법에 기반을 둔 경험법칙을 이용하여 SOAL 응용으로부터 얻은 초기배치의 한계를 극복하여 최적해를 구하였다.

## 2.2 절단방법

일반적인 제조공정에서 많이 사용되는 작두형(guillotine cut) 절단방법을 채택하여 알고리즘을 구성하였다. 여기서 작두형 절단방법은 주로 직사각형 형상을 이루는 철판 또는 합판, 유리, 섬유 및 피혁 제품 등의 절단에서 많이 사용되고 있다. 그 밖의 절단 방법에는 일반적인 절단방법(general cut)이 있으며, 이는 판재 가공에서의 편칭, 레이저나 화염을 이용한 절단 방법과 옷감의 재단과 같은 가공방법이 이에 속한다.

## 2.3 형상의 종류

형상의 종류는 일정 형상(regular shapes)과 임의 형상(irregular shapes)으로 구분되는데, 여기에서는 판재 배치에 적합한 직사각형 일정 형상 배치 알고리즘을 선택하였다. 이를 위하여 ① 절곡된 부품의 경우 블랭크 크기(blank size)를 계산하여 절곡 전의 철판 사이즈로 부품의 사이즈를 대치하고, ② 임의 곡면으로 이루어진 경계가 있는 부품은 근사화 된 직선으로 이루어진 사각형 형태로 단순화 하였다.

## 3. 최적해 탐색방법

### 3.1 배치 적용순서

다수의 원판에서 다수의 부품을 효율적으로 등분하여 스크랩을 최소화하기 위한 효과적인 정렬방법을 <표 1>에 나타내었다. <표 1>의 8단계 알고리즘은 시행착오를 거친 경험적 방법에 근거를 두고 작성하였다.

<표 1> 경험법칙에 근거한 배치법

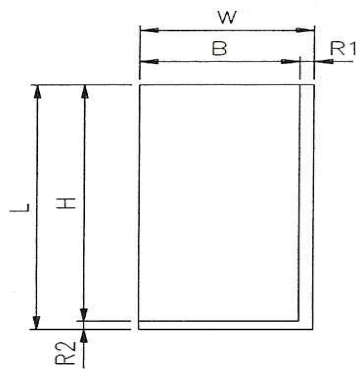
1	부품 DATA 입력
	부품이름, 재질, 치수(폭, 높이, 두께)
2	부품 DATA 순서결정
	면적계산(가로×세로) → 두께 순(두꺼운 것 우선) → 재질 순 → 크기 순(큰 것 우선)
3	원판 Group 설정(두께별, 재질별)
	자동배치 시 고려한 사항 : 부품과 같은 두께의 원판 Group → 부품과 같은 재질의 원판 Group → 배치 방안의 선택 기준(최소 Scrap) → 배치 알고리즘
4	부품 Size로 자동적으로 원판설정 및 스크랩을 최소화 하는 방안으로 원판의 물량계산
5	원판에 의한 1차 절단 후 발생하는 스크랩을 원판으로 등록 후 재배치
	부품과 같은 두께 원판 Group → 부품과 같은 재질 원판 Group → 최소 스크랩 발생 배치방안 선택 → 배치 알고리즘
6	부품 배치 완료
7	배치된 배열 상태 Display
	원판 : 두께, 폭, 높이, 재질 부품 : 두께, 폭, 높이, 재질
8	원판별 사용 물량 집계
	원판 재질, 원판 소요량, 소요된 물량의 비중

### 3.2 원판을 부품의 가로(B), 세로(H)로 등분하는 방법

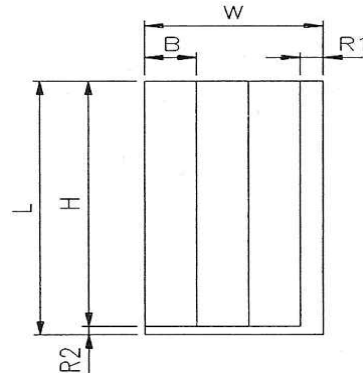
<표 1>의 4번째 방법을 서술하는 것이며, 구성 알고리즘은 원판을 부품의 가로와 세로로 등분하는 방법으로 다음과 같은 두 가지의 경우로 프로그래밍 하였다. 이의 구성을 위하여 가로방향의 등분수를 X, 세로 방향의 등분수를 Y, 원판폭을 W, 원판의 높이를 L, 그리고 가로 방향의 스크랩을 R1, 세로방향의 스크랩을 R2로 표기하였다.

#### 3.2.1 원판과 부품이 같은 방향으로 등분하는 경우

두 가지 경우로 표현할 수 있다. 첫 번째는 <그림 1>에 표현된 것과 같은 하나의 원판에 하나의 부품이 네스팅되는 경우로 등분한 결과, 값 표기는 가로방향  $B \times X$ , 세로방향  $H \times Y$ 로 표현되며, 가로방향으로 등분한 나머지 값 표기 R1은  $W - B \times X$ , 그리고 세로방향으로 등분한 나머지 값 표기 R2는  $L - H \times Y$ 로 계산된다. 두 번째는 <그림 2>와 같이 하나의 원판에 다수의 부품이 네스팅되는 것으로 이 경우  $X = W/B$ ,  $R1 = W - (B \times X)$ ,  $R2 = L - (H \times Y)$ 로 정의된다.



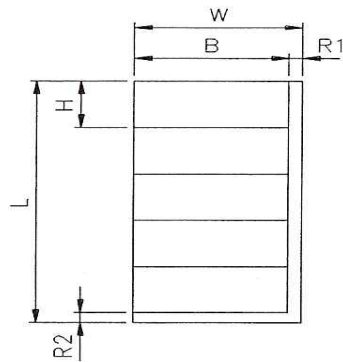
<그림 1> 경우 1



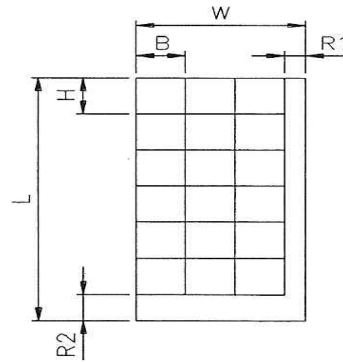
<그림 2> 경우 2

### 3.2.2 원판은 고정되고 부품의 방향을 90°회전하여 등분하는 경우

<그림 3>의 경우 등분한 결과, 값 표기는 가로방향  $H \times X$ , 세로 방향  $B \times Y$ 이며,  $R1 = W - H \times X$ ,  $R2 = L - B \times Y$ 이고, <그림 4>의 경우  $X = (W/B)$ ,  $Y = (L/H)$ ,  $R1 = W(B \times X)$ ,  $R2 = L - (H \times Y)$ 이 된다.



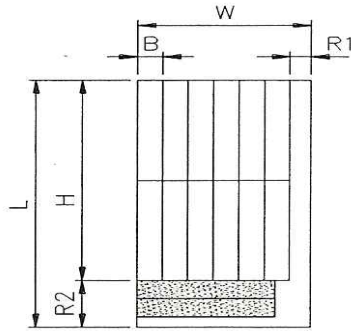
<그림 3> 경우 1



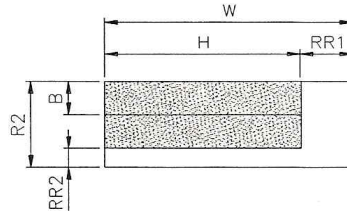
<그림 4> 경우 2

### 3.3 스크랩 처리방안

$R1/H$ 가 0 아닌 경우  $R2/B$ 가 0이 아니면 등분 처리를 하며, 또한  $R1/B$ 가 0이 아닌 경우  $R2/H$ 가 0이 아니면 원판으로 재등록하여 등분 처리를 수행한다. <그림 5>에서 등분한 결과 발생하는 스크랩을 원판으로 재등록하여 다른 부품을 다시 네스팅한 결과를 <그림 6>에 나타내었다. 이 경우 적용된 조건은  $R2 > R1$ 이고  $R1/H \neq 1$ ,  $R2/B \geq$ 이며, 스크랩의 가로방향 등분수를  $XX$ , 세로방향 등분수를  $YY$ 로, 표현할 경우  $XX = (W/B)$ ,  $YY = (R2/B)$ ,  $RR1 = W - (H \times XX)$ ,  $RR2 = R2 - (B \times YY)$ 로 나타낼 수 있다.



<그림 5> 경우 1



<그림 6> 경우 2

#### 4. 변수 정의

원판에 들어가는 부품의 개수를 결정하는 방법으로 크게 두 가지 경우로 나눌 수 있으며, 각각 경우의 수를 프로그램 언어인 Lisp로 정리하였다. 이를 위하여 원판폭을 기준으로 부품폭으로 나눈 등분수를 폭폭수량으로, 원판길이를 기준으로 부품 길이로 나눈 등분수를 길길수량, 원판폭을 부품의 길이 방향으로 나눈 등분수를 폭길수량, 그리고 원판길이를 부품의 폭방향으로 나눈 등분수를 길폭수량으로 정의하였다.

1) CASE 1.(→C1)

(SETQ X1 (FIX (/W B))

→ 원판폭 / 부품폭; 폭폭수량(X1)

(SETQ Y1 (FIX (/ L H))

→ 원판길이 / 부품길이; 길길수량(Y1)

2) CASE 2.(→C2)

(SETQ X2 (FIX (/ W H))

→ 원판폭 / 부품길이; 폭길수량(X2)

(SETQ Y2 (FIX (/ L B))

→ 원판길이 / 부품폭; 길폭수량

(SETQ C1 (\*X1 Y1))

→ 1의 경우 폭수량과 길이 수량을 곱한 값; 폭수량

(SETQ C2 (\* X2 Y2))

→ 2의 경우 폭수량과 길이 수량을 곱한 값; 길이 수량

(SETQ X1A (RTOS X1 2 0))

(SETQ C1A (RTOS C1 2 0))

#### 4.1 원판에 대한 부품이 들어가는 추가 개수

(SETQ P1 (FIX (/ (-W (\*B X1)) H)))

→ [원판폭 - (폭폭 수량 × 부품폭)] / 부품폭 = 추가폭 수량(P1)

(SETQ P2 (FIX (/ (- L (\*H Y1)) B)))

→ [원판길이 - (길길 수량 × 부품길이)] / 부품폭 = 추가폭 수량(P2)

(SETQ P3 (FIX (/ (- W (\* H X2)) B)))

→ [원판폭 - (폭길수량 × 부품길이)] / 부품폭 = 추가폭 수량(P3)

(SETQ P4 (FIX (/ (- L (\* B Y2)) H)))

→ [원판길이 - (길폭수량 × 부품폭)] / 부품길이 = 추가 길이 수량(P4)

#### 4.2 모든 경우에 들어갈 수 있는 부품의 총 개수

(SETQ V1 (+ C1 (+ (\* P1 Y2) (\* P2 X2))))

→ (폭 수량 × 길이 수량) + (추가 폭 수량 × 추가 길이 수량) = 총 폭 수량(V1)

(SETQ V2 (+ C2 (+ (\* P3 Y1) (\*P4 X1))))

→ (길이 폭 × 폭 길이 수량) + (추가 폭 수량 × 추가 길이 수량) = 총 길이 수량(V1)

폭길 수량 = 0 이면

(IF (=X2 0)

(MCP1)

(MCP2)

(IF (>= (-W (\* B X1)) H) [원판폭 - (폭폭수량 × 부품폭)] ≥ 부품길이

(MCP3)

(IF (>= (-W (\* B X1)) H) [원판폭 - (폭폭수량 × 부품폭)] ≥ 부품길이

(MCP3)

(IF ( $\geq$  (- L (\* H Y1)) B) [원판길이 - (길길수량 × 부품길이)]  $\geq$  부품폭  
(MCP4)

(IF ( $\geq$  (- W (\* H X2)) B) [원판폭 - (폭길수량 × 부품길이)]  $\geq$  부품폭  
(MCP5)

(IF ( $\geq$  (- L (\* B Y2)) H) [원판길이 - (길폭 수량 × 부품폭)]  $\geq$  부품길이  
(MCP6)

즉, 상기 경우를 자세히 분석하면 4가지 경우로 나눌 수 있다. 첫 번째는 부품의 길이가 원판의 폭보다 큰 경우에는 원판의 길이 방향으로만 배열할 수 있다. 두 번째는 부품의 길이가 원판의 폭보다는 작지만 세 번째 경우처럼 부품이 더 이상 들어가지 않는 경우는 세로 방향으로 배열하여 많이 들어가는 경우를 우선으로 한다. 세 번째 경우는 부품의 길이가 원판 폭보다는 작고 부품이 더 들어 갈수 있는 경우에는 세로 방향과 비교해서 많이 들어가는 경우를 우선으로 한다. 그리고 네 번째의 경우는 두 번째의 경우와 세 번째 경우의 중간 경우로 원판의 폭 방향으로 배열 했을 경우 남은 부분의 길이가 부품의 폭보다 작아서 더 이상 들어 갈수 없지만 원판의 길이 방향으로 배열하면 남은 부분에 부품을 더 배열할 수 있는 경우가 생긴다. 물론 이 경우에도 개수를 비교하여 많이 배열할 수 있는 것을 우선으로 한다. 이를 순서대로 표현하면 <그림 7>과 같다.

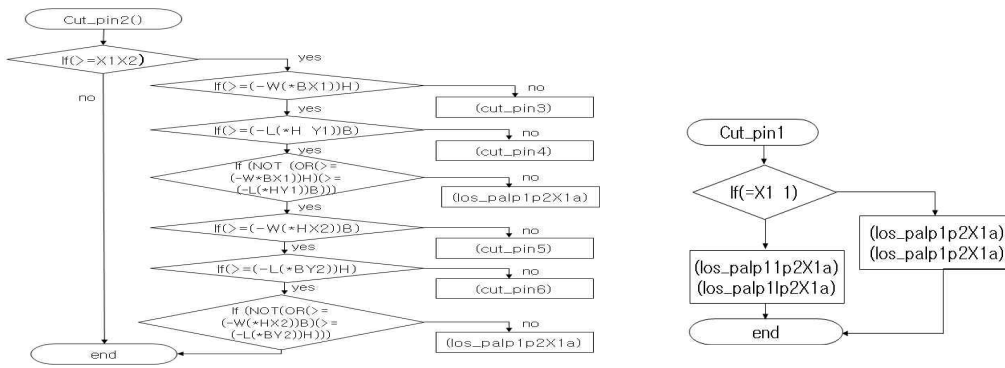
### 5. 프로그램 실행 예

본 연구에서 이루어진 알고리즘을 기반으로 작성된 프로그램을 구동시킨 결과의 일부를 <그림 8>에 나타내었다. 배치도를 보면 그림에 각기 다른 부품의 명칭이 나타나 있는 것을 볼 수 있으며, 이때 글자의 방향은 부품이 배치되는 방향을 표시한 것이다.

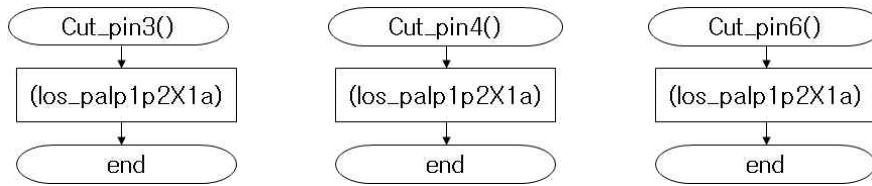
또한 자동 배치된 부품 데이터의 제원은 <표 2>와 <표 3>에 나타내었다.

<그림 8>을 보면 6개의 부품이 필요한 “KK 0”의 경우 원판 ③에 4개가 배치되었으며, 이때 “KK 0” 4개를 배치하고 남게 되는 원판 ③의 스크랩 부분을 다시 원판으로 등록하여 다른 부품의 네스팅에 사용되었다. “KK 0”의 나머지 2개는 원판 ④에 배치된 모습을 볼 수 있다. 또한 6개의 부품이 필요한 “KK 0”는 원판 ④에 2개 그리고 원판 ⑤에 4개가 배치되었다. “OP 10”은 원판 ③에 1개 그리고 원판 ④에 9개를 배치하는 것이 서로 다른 다수의 원판과 부품들 사이의 배치에서 얻을 수 있는 최적의 결과임을 의미한다.



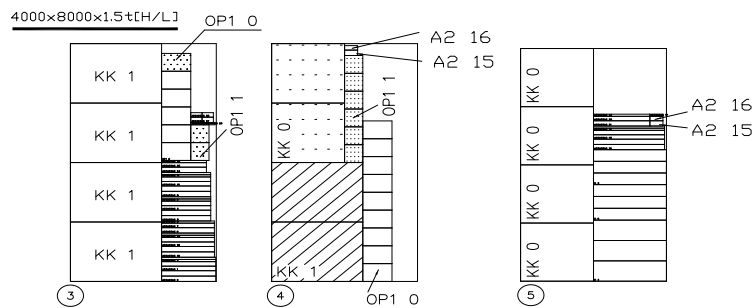


(a) 메인 순서도



(b) 세부 순서도

<그림 7> 순서도



<그림 8> 실행 예

<표 2> 부품 리스트

Name	Mat'l	T	W	L	Q'ty
KK 1	H/L	1.5	2500	2000	6
KK 0	H/L	1.5	2000	2000	6
OP1 0	H/L	1.5	800	600	10
OP1 1	H/L	1.5	500	600	8
A2 15	H/L	1.5	358	181	2
A2 16	H/L	1.5	358	152	2

&lt;표 3&gt; 원판 ④에 배치된 단품 리스트

Sheet No.	Item Name	Mat'l	Sheet Size
4	A2 15	H/L	1.5x4000x8000
4	A2 16	H/L	1.5x4000x8000
4	KK 0	H/L	1.5x4000x8000
4	KK 1	H/L	1.5x4000x8000
4	OP 10	H/L	1.5x4000x8000
4	OP 11	H/L	1.5x4000x8000

## 6. 결 론

본 연구에서는 다수의 원판에서 버림률을 최소화하는 원판과 부품을 검색하여 초기 배치 후 일반적으로 버려지는 스크랩을 다시 원판으로 등록하여 같은 알고리즘을 단계로 반복하여 최적의 절단 배치도를 구성하도록 하는 판재 절단 배치시스템의 알고리즘을 구현하였다.

프로그램을 구동해 본 결과 모든 경우의 수를 다 고려하여 엔지니어가 직접 계산한 절단계획과의 차이는 없었으며, 오히려 엔지니어의 개인오차에 의한 실수를 미연에 방지함은 물론이고, 설계시간의 단축과 자재 수급계획에서 발생하는 비능률을 줄여 궁극적으로 신뢰성에 기반을 둔 원가절감 및 생산성 향상에 기여하는데 활용 가능성이 있음을 확인하였다.

또한, 알고리즘 구현으로 계산되어진 데이터를 이용한 자재의 소요량과 AutoCAD에 의해 출력된 배치도는 대량생산시스템에서 다품종 소량생산 체제구축을 위한 자재 및 생산 계획의 주요자료로 이용할 수 있다. 따라서 현재 대부분의 제조업체에 구축되어져 있는 ERP시스템과 CAD시스템에 본 배치 데이터를 연동시키면 생산계획-자재확립-실행-결과를 통합할 수 있는 효율적인 생산관리시스템을 구축할 수 있을 것으로 기대된다.

## 7. 참 고 문 헌

- [1] 조경호 , 이진우, “임의 형상의 여러 원자재 위에서의 효과적인 배치방안”, 대한기계학회논문집, 19, 8 (1995) : pp. 1854-1868.
- [2] 한국찬 , 나석주, “신경회로망을 이용한 직사각형의 최적배치에 관한 연구”, 대한기계학회 1993 춘추학술대회 논문집, 1, 1 (1993) : pp. 954-957.
- [3] 한국찬, 나석주, “레이저 절단공정에서 절단부재의 최적배치를 위한 네스팅 알고리즘”, 대한용접·접합학회지, 12, 2 (1994) : pp. 11-19.
- [4] 임성국, 양성모, 고석호, 김현정, 한관희, “2차원 공간에서의 휴리스틱 배치 알고리즘 및 구현에 관한 연구”, 한국CAD/CAM학회논문집, 4, 3 (1999) : pp. 53-59.
- [5] 이강길, 노상도, 강형석, 강성재, 허준, “디지털 가상공장을 이용한 자재 배치 효율화에 관한 연구”, 한국CAD/CAM학회 2006 학술발표회논문집, (2006) : 486-491.