Journal of the Society of Korea Industrial and Systems Engineering
Vol. 34, No. 4, pp.129-137, December 2011.

# Flow Shop 배치설계를 위한 휴리스틱 알고리즘

남기호† · 옥창훈 · 서윤호

고려대학교 산업경영공학과

# A Heuristic Algorithm for Flow Shop Layout Design

Kee-ho Nam† · Chang-Hun Ok · Yoonho Seo

Department of Industrial Management Engineering, Korea University

To date, facility layout problems has been solved and applied for job shop situations. Since flow shop has more restrictions, the solution space is much smaller than job shop. An efficient heuristic algorithm for facility layout problems for flow shop layouts is needed to be developed. In this thesis, a heuristic algorithm for rectangular bay layouts in a flow shop situation is presented. The proposed algorithm is developed by using slicing tree representation and applied to various flow shop layout problems. The effectiveness of the proposed algorithm in terms of exploration rate and objective function value are shown by comparing our results to simulated annealing.

Keywords : Flow Shop, Layout, Slicing Tree

## 1. 서 론

제조 시스템의 총 운영비용 중 15~70%가 자재취급에 의해 결정된다고 보며, 효율적인 설비배치가 이 비용을 10~30% 감소할 수 있다고 했다[15]. 제조시스템에서 설비는 공정을 수행하는 기계 또는 기계가 배치될 수 있는 지역을 뜻한다. 시장에서 기업 경쟁력을 높이기 위해 제조시스템의 운영비용을 줄여야 하며, 잘못된 설비배치는 재공재고의 증가와 설비 설치비용의 증가로 이어질 수 있다[2, 5]. 따라서 설비배치는 기업의 경쟁력을 결정할 수 있는 중요한 의사결정 중 하나라 할 수 있다.

설비배치문제는 작업장 내에 설비의 위치를 결정하는 문제로써, 설비들 간의 총 물류이동거리를 최소화하는 것이 목적이다. 총 물류이동거리는 설비들 간의 거리와 물류량의 곱의 합으로 표현된다[4]. 이 거리는 일반적으로 직각거리나 직선거리가 사용된다[15].

설비배치 문제는 Quadratic Assignment Problem(이하 QAP) 형태로 주로 연구되었다. QAP는 전체 설비들 간의 총 물류이동거리를 최소화하도록 격자 형태로 나눠진 지역(또는 site)에 설비를 할당하는 문제이다. Skorin-Kapov[13]은 타부 서치를 이용하여 지역이 15개 이하일 경우 최적값을 찾았다. 하지만 현실적으로 설비는 다양한 크기와 모양을 가졌고, 설비가 놓일 지역이 정해져 있지 않기 때문에 QAP 모형을 적용하는 한계가 있다. 이러한 한계를 극복하기 위하여 크기와 모양이 다양한 설비의 배치문제(Unequal Area Facility Layout Problem, 이하 UAFLP)가 연구되었다[5].

UAFLP는 QAP보다 알고리즘이 더욱 복잡하며, QAP와 마찬가지로 설비의 수가 클 경우, 유효한 시간 내에 최적해 탐색이 어렵다. 이를 해결하기 위해 Bazaraa[1]가 분기한정법(Branch and Bound), van Camp et al.[16]이 비선형 프로그래밍 모델과 휴리스틱 모델을 제안했고, Meller and Gau[8]는 UAFLP를 위한 첫 Mixed Integer Program-

논문접수일 : 2011년 10월 28일 논문수정일 : 2011년 11월 31일 게재확정일 : 2011년 12월 10일
† 교신저자 shadow1005@korea.ac.kr

ming(이하 MIP)을 소개했다. 또한 Gen and Cheng[3], Rajasekharan and Peters[10], Li and Love[6]는 유전자 알고리즘을, Mir and Iman[9], McKendall et al.[7]과 Wang et al.[17]은 Simulated Annealing(이하 SA)을, Scholz et al.[11]은 슬라이싱 트리 기법을 이용한 타부 서치를, Komarudin and Wong[18]은 개미군집최적화을 이용한 알고리즘을 제안하였다.

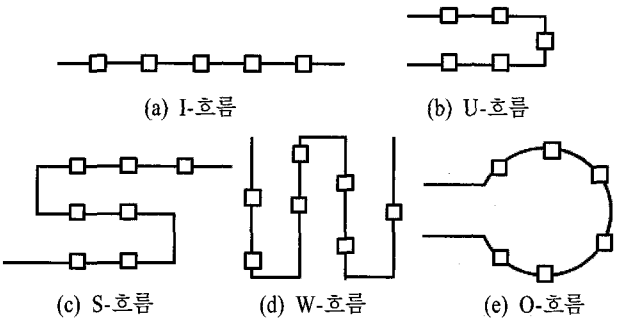
설비배치 문제에 대한 연구는 일반적으로 job shop 상황을 가정하고 있다. Job shop은 flow shop의 일반적인 형태로 모든 작업들이 각각 다른 작업순서를, 반면에 flow shop은 동일한 작업순서를 갖는다[12]. Flow shop의 경우 물류의 흐름이 모두 동일하기 때문에 물류의 흐름을 이어가는 것이 효율적이다. 본 연구에서 다루는 flow shop 배치설계 문제에서는 서로 연결된 공정을 담당하는 설비는 인접해야 하는 제약조건이 있으므로, job shop에 맞춰 고안된 기존의 해법을 flow shop에 적용하는 것은 비효율적이다. 본 연구에서는 이러한 비효율을 감축하기 위해 flow shop 전용 배치 알고리즘을 개발하였다.

본 연구의 목적은 반도체 공정과 같이 모든 제품이 같은 순서의 설비를 지나야 한다는 제약을 가진 flow shop 공정에 대하여, 직사각형 모양의 설비가 놓일 공간(이하 베이)을 배치해 이동거리를 최소화하는 배치를 찾아내는 것이다. 이를 위해 본 연구에서는 슬라이싱 트리 기법을 사용하였고, 효율적인 배치를 제공하는 휴리스틱 알고리즘을 개발한다.

제 2장에서는 문제 정의 및 목적함수를 계산하기 위한 모델을 제시하였고, 제 3장에서는 논문에서 쓰인 슬라이싱 트리의 구조 및 유효하고 효율적인 배치 대안을 찾기 위한 휴리스틱 알고리즘을 보인다. 제 4장에서는 제안하는 알고리즘을 통해 나온 결과를 보여 효율성을 입증하고, 제 5장은 결론 및 추후 연구 방향에 대하여 기술한다.

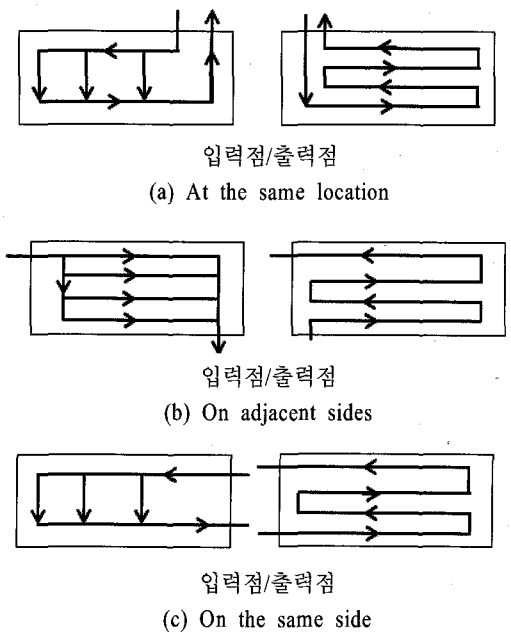
## 2. 문제 기술

Flow shop에서 피가공품들은 일반적으로 모두 동일한 순서에 따라 가공된다[12]. 따라서 flow shop은 <그림 1>과 같이 주로 'Line flow pattern'을 띄며, 이는 I-흐름, U-흐름, S-흐름, W-흐름, 그리고 O-흐름 등의 모양으로 나타난다. 이러한 모양은 주로 생산 라인의 길이와 주어진 공간의 제약에 따라 정해진다. 그 예로, 긴 직선 라인(I-흐름)은 공간이 비교적 얇아 공간 사용이 비효율적 일 때 쓰인다. Line flow 구조는 공정의 역방향으로 흐르는 경우가 없는 조립 공정과 같은 생산 라인에서 가장 효율적이다.




<그림 1> Line Flow Pattern의 예

또한, 입력점과 출력점의 위치 또한 흐름을 정할 때의 중요한 요소 중 하나이다[12]. 입력점과 출력점의 위치는 각각 하나의 위치로 고정되어 있다고 가정한다. 입력점과 출력점의 위치에 따른 다양한 흐름의 모양의 예를 다음의 <그림 2>에서 볼 수 있다.




<그림 2> 입력점과 출력점의 위치에 따른 흐름

본 연구에는 flow shop에서의 설비배치문제를 다룬다. Flow shop에서는 생산능력에 따라 각 공정을 수행하는 설비대수가 결정될 수 있으며, 따라서 베이의 대체적인 면적이 계산 가능하다. 설비가 배치될 빈 공간(floor)과 입출력점 그리고 공정순서와 각 공정을 담당할 설비대수가 주어지면, 입력점부터 출력점까지 각 bay의 중심점을 연결하는 직각거리의 합이 최소화 되도록 bay의 크기와 모양을 결정하여 주어진 공간 내에 배치하는 문제이다. 본 연구의 제약조건은 아래와 같다.

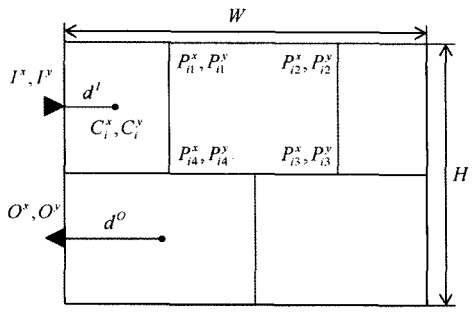
① 모든 베이의 넓이 합은 작업장 넓이와 같다.

② 작업장과 베이는 직사각형 형태를 갖는다.
③ 연속적 공정인 두 베이는 맞닿아야 한다.
④ 첫 번째 베이는 작업장 입력점을 포함해야 한다.

## 2.1 변수 정의

본 논문에 사용되는 변수는 다음과 같이 정의하였으며, <그림 3>에 나타난다.

$N$ : 베이 수
$i$ : 베이 인덱스($i=1,\cdots,N$)
$W,H$ : 작업장 가로, 세로
$A_i$ : 베이 $i$ aspect ratio 하한값($0 \le A_i \le 1$)
$\alpha_i$ : 베이 $i$의 aspect ratio($A_i \le a_i \le 1$)
$S_i$ : 베이 $i$의 넓이 비
$I^x, I^y$ : 입력점의 x, y 좌표
$O^x, O^y$ : 출력점의 x, y 좌표
$P^x_{i1}, P^y_{i1}$ : 베이 $i$의 좌측 상단 꼭지점 x, y 좌표
$P^x_{i2}, P^y_{i2}$ : 베이 $i$의 우측 상단 꼭지점 x, y 좌표
$P^x_{i3}, P^y_{i3}$ : 베이 $i$의 우측 하단 꼭지점 x, y 좌표
$P^x_{i4}, P^y_{i4}$ : 베이 $i$의 좌측 하단 꼭지점 x, y 좌표
$C^x_i, C^y_i$ : 베이 $i$ 중심점의 x, y 좌표
$d^I$ : 입력점과 첫번째 베이 사이 직각거리
$d^O$ : 출력점과 마지막 베이 사이 직각거리




<그림 3> 5개 베이 배치 예

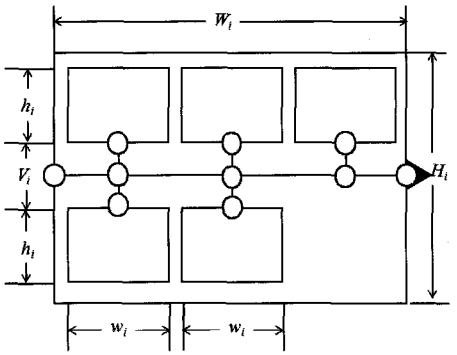
## 2.2 목적함수

Flow shop 설비배치 문제의 목적함수는 입력점부터 출력점까지 각 bay의 중심점을 연결하는 직각거리 합을 최소화하는 것으로써, 식 (1)과 같다.

$$Min \sum_{i=1}^{N-1} \left( |C^x_i - C^x_{i+1}| + |C^y_i - C^y_{i+1}| \right) + d^I + d^O \qquad (1)$$

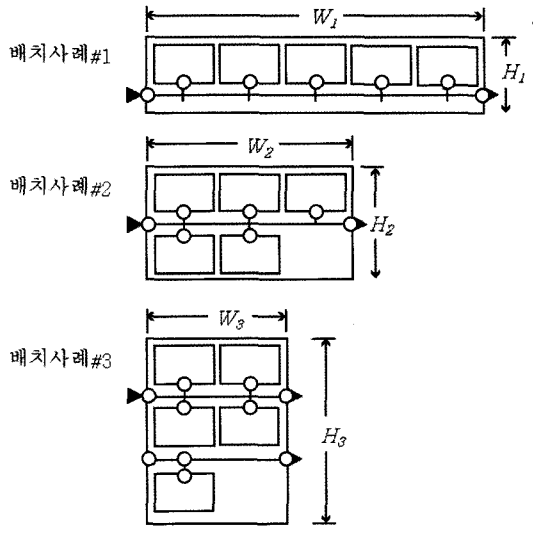
목적함수는 첫 번째 베이와 입력점간의 거리, 베이 중심점 간의 직각거리의 합, 그리고 마지막 베이와 출력점간의 거리의 합으로 평가하며, 이 값이 최소화 되는 배치를 찾는 것이다.

베이에 배치되는 설비 종류와 대수, 그리고 사용되는 물류 장비에 의해 대체적인 베이 형태가 결정된다. 기본적으로 설비 후면, 측면, 전면에는 설비의 이동 및 유지 및 보수를 위하여 어느 정도 간격을 두어야 한다. 특히 전면부에는 물류장비가 이동할 수 있는 간격이 필요하다. 예를 들어 5대의 설비가 필요한 베이의 최소 면적은 아래 <그림 4>에 설명된 것과 같이 $(2h_i + V_i + 2\gamma_i) \times (3w_i + 4\gamma_i)$로 표현될 수 있다. 여기서 V는 물류장비가 이동할 수 있는 간격을 의미하고, $\gamma$는 설비와 벽간의 좁은 간격을 의미한다.




<그림 4> 작업장 및 설비 기호

또한 설비를 배치하는 방식에 따라 여러 가지 베이 형태가 가능하다. 예를 들어 다섯 대의 설비를 각각 다른 방식으로 배치한 사례를 <그림 5>에 도식하였다. 각각의 경우는 서로 다른 베이 면적과 형태를 나타내고 있다.

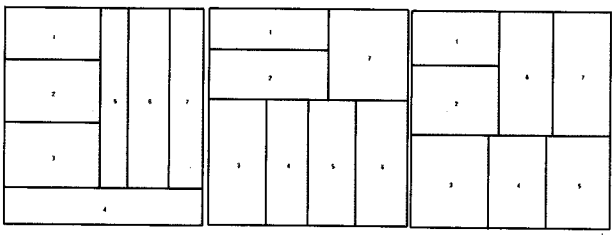



<그림 5> 설비 다섯 대일 때의 배치 형태

따라서 설계자는 베이의 aspect ratio에 제약을 두어 다양한 베이 배치 결과를 유도할 수 있다. 베이의 aspect ratio $\alpha_i$은 베이의 가로와 세로의 비율을 의미하며, 식 (2)로 표현된다.

$$\alpha_i = \frac{Min(|P_{i2}^x - P_{i1}^x|, |P_{i1}^y - P_{i4}^y|)}{Max(|P_{i2}^x - P_{i1}^x|, |P_{i1}^y - P_{i4}^y|)} \quad (2)$$

식 (2)에 따르면 $\alpha_i$는 0과 1사이 값($0 < \alpha_i \leq 1$)이다. 베이의 가로세로비에 제약을 두기 위해 각 베이에 대해 가로세로비 하한값 $A_i$를 정의한다. 즉 $A_i < \alpha_i \leq 1$을 의미한다. 만약 $\alpha_i < A_i$이면, 그 베이는 유효하진 않는 것으로 한다. 모든 베이 $i$의 aspect ratio 하한값을 각각 $A_i$ = 0.14, 0.25, 0.4로 설정하였을 때 <그림 6>과 같은 결과를 생성할 수 있다.



(a) $A_i$ = 0.14 (b) $A_i$ = 0.25 (c) $A_i$ = 0.4

<그림 6> Aspect Ratio에 따른 레이아웃 변화

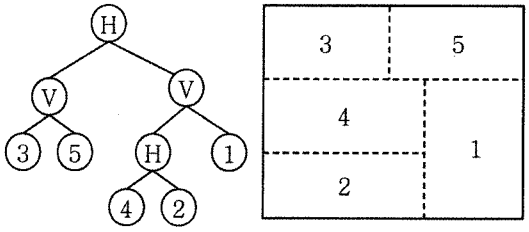
## 3. 휴리스틱 알고리즘

### 3.1 Slicing Tree 구조 소개

슬라이싱 트리는 Tam[14]에 의해 최초로 소개되었다. 슬라이싱 트리는 크기가 다를 때의 배치 문제를 표현하기에 유용한 방법으로 설비배치문제에 적용되어 좋은 성과를 보였다.

슬라이싱 트리 기법은 초기 사각형을 가로 혹은 세로 방향으로 재귀적으로 잘라 하나의 슬라이싱 구조를 만들어내는 기법이다. 슬라이싱 트리는 각 베이를 나타내는 N개의 잎 노드와, 잘리는(이하 컷) 방향(H : 가로/V : 세로)을 나타내는 N-1개의 내부 노드로 구성되어 있다.

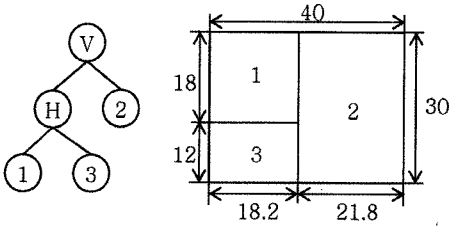
<그림 7>은 슬라이싱 트리와 그에 따라 잘려진 구조를 나타낸다. 뿌리 노드가 가로(H) 방향이므로, 첫 번째 컷은 작업장을 가로 방향으로 완전히 자른다. 뿌리 노드의 왼편에 있는 베이(3, 5)와 오른편에 있는 베이(1, 2, 4)로 나뉘어 각각 위쪽과 아래쪽에 배치된다. 뿌리 노드 이하의 내부 노드들의 방향에 따라 베이가 순서대로 배치되어 최종적으로 그림 오른쪽과 같은 구조가 완성된다.




<그림 7> 슬라이싱 트리와 구조

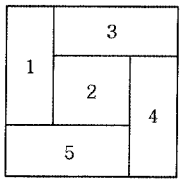
<그림 8>의 그림은 하나의 예제를 그림으로 표현한 것이다. 첫 번째 표에서 각 베이의 크기를 알 수 있고, 슬라이싱 트리의 각 내부 노드의 아래에 위치한 베이들의 크기의 합은 내부 노드의 방향대로 잘리는 위치를 결정하게 된다. 뿌리 노드의 자식노드 중 왼편의 베이 1과 3의 크기는 25이고, 오른편의 베이 2의 크기는 30이 되므로, 세로 방향 컷의 위치가 결정된다.

| 베이 | 크기 |
|---|---|
| 1 | 15 |
| 2 | 30 |
| 3 | 10 |




<그림 8> 슬라이싱 구조의 예제

슬라이싱 트리 기법으로 모든 배치 구조를 표현할 수 있는 것은 아니다. 아래 <그림 9>의 wheel 구조는 slicing tree로 표현 할 수 없다[11].




<그림 9> Wheel 구조

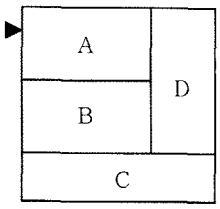
MIP 기법으로 정해진 설비배치문제의 최적해는 슬라이싱 트리가 표현할 수 없는 위 <그림 9>와 같은 배치를 포함할 수 있다. 그러나 현실적으로 슬라이싱 트리 기법의 컷은 그 방향에 따라 통로 배치가 용이하므로 실제 공장설계에 적용하기에 용이한 측면이 있다[11].

### 3.2 Flow Shop-Slicing Tree

슬라이싱 트리는 베이, 슬라이싱 순서, 슬라이싱 방향을 나타내는 3개의 숫자열로 구성되어 있다. 베이의 개수가 $N$개일 때, 슬라이싱 트리로 나타낼 수 있는 모든 해집단의 수는 $N!(N\text{-}1)!2^{N-1}$이다.

컷의 방향과 순서가 슬라이싱 구조의 형태를 결정하므로, $(N\text{-}1)!2^{N-1}$개의 슬라이싱 구조가 있다. Job shop을 가정하였을 때, 하나의 완성된 구조 내에 베이를 할당할 수 있는 총 경우의 수는 $N!$이다. 하지만 제 2장의 제약조건 ③인 '연속적 공정인 두 베이는 맞닿아야 한다.'는 가정에 의해 고려해야 할 배치의 숫자는 현저히 줄어든다. 추가적으로 제약조건 ④와 같이 입력점을 고려한다면, 해집단은 더욱 작아지게 된다.

예를 들어, <그림 10>에서와 같이 슬라이싱 구조 내에는 총 4!, 즉 24개의 배치가 가능하다. 그러나 A, B, C, D에 각각 (1, 2, 3, 4), (1, 2, 4, 3), (1, 4, 3, 2), (1, 4, 2, 3), 총 네 가지 경우로 할당되었을 때만 위의 제약조건 ③과 ④를 만족한다. 전체 해집단의 크기보다 매우 작은 경우의 수만을 고려하면 된다는 것을 알 수 있다.



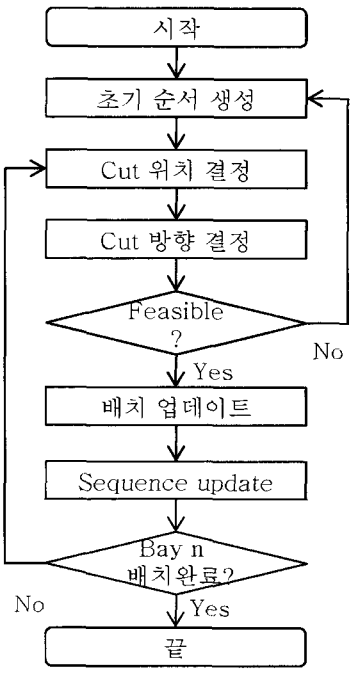

<그림 10> 슬라이싱 구조의 예

본 논문에서는 flow가 유지되지 않는 해는 배제하기 때문에, 배치가 진행되는 동안 유효한 해집단을 효율적으로 찾아낼 수 있다. 유효한 해의 탐색 비율은 실험결과에서 보인다.

### 3.3 Flow Shop 배치 알고리즘

앞선 절에서 슬라이싱 트리의 구조가 베이의 순서, 슬라이싱의 순서, 그리고 슬라이싱의 방향 세 개로 나뉜다고 했다. 기존 슬라이싱 트리를 이용하는 논문에서는 세 가지 구성요소 모두를 랜덤으로 생성하여, 그 중 가장 목적함수 값이 작은 해를 선택한 후, 휴리스틱으로 이웃해를 찾아 최적값을 도출하는 방법이 사용되었다.

본 논문의 알고리즘에서는 슬라이싱 트리를 나타내기 위한 구조 중 베이의 순서와 슬라이싱의 순서는 랜덤으로 결정하되, 컷의 방향은 흐름을 유지할 수 있는 방향으로 결정한다. 알고리즘 순서는 <그림 11>과 같다.



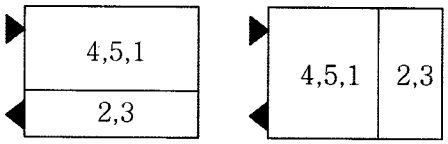

<그림 11> 알고리즘 순서도

초기 베이의 순서를 결정할 때, flow shop의 특성을 이용한다. Flow shop에서는 $i$번째 베이와 $(i+1)$번째 베이는 맞닿아야 유효하다는 제약조건이 있다. 컷을 실행할 때 인접해야 하는 베이들이 같은 묶음에 할당되면 맞닿을 수 있다는 특성이 있으므로, 1, 2, 3, …, $n$의 순서를 유지한다. 그러나 순서를 이대로 고정하기만 한다면 다양한 모양의 레이아웃이 나올 수 없다는 단점이 있으므로, $n\text{-}2$, $n\text{-}1$, $n$, 1, 2, …, $n\text{-}3$과 같은 순서도 허용한다. 예를 들어, 3개의 베이가 있다고 할 때, 초기 순서는 (1, 2, 3), (2, 3, 1), 그리고 (3, 1, 2) 세 가지의 경우가 있을 수 있으며, 이 중 랜덤으로 채택된다.

다음으로 컷의 위치를 랜덤으로 결정하게 된다. 하나의 묶음을 두 개의 묶음으로 자르는데, 어디서 자를 것인가를 결정하게 된다. 컷의 위치가 결정되면, 우선적으로 첫 번째 공정을 담당하는 베이 1을 포함한 묶음을 입력점이 있는 쪽에 위치하도록 배치한다. 또한, 베이 1을 입력점이 있는 쪽에 배치하는 것과 마찬가지로, 마지막 공정을 담당하는 베이 $n$은 출력점이 있는 쪽에 배치한다. 만일 베이 1이 입력점 쪽에 할당함으로써 베이 n이 출력점 쪽에 할당될 수 없다면 비유효한 것으로 판단하여, 슬라이싱의 방향을 바꾼다.

아래 <그림 12>에서 볼 수 있듯이, 5개의 공정으로 이루어진 레이아웃일 경우, 마지막 공정은 베이 5가 된다. 랜덤으로 (4, 5, 1, 2, 3)의 초기 순서가 (4, 5, 1)과 (2, 3)으로 나뉘었고, 첫 번째 선택된 컷의 모양은 가

로 모양이다. 이 경우, 베이 1은 입력점 쪽에 할당되었으나, 베이 5가 출력점 쪽에 할당되지 못하는 경우가 발생하여 비유효하게 된다. 이와 같은 경우에 컷의 방향을 세로 모양으로 바꿔줄 경우, 입력점과 출력점 모두를 만족시켜, 유효한 배치가 된다.



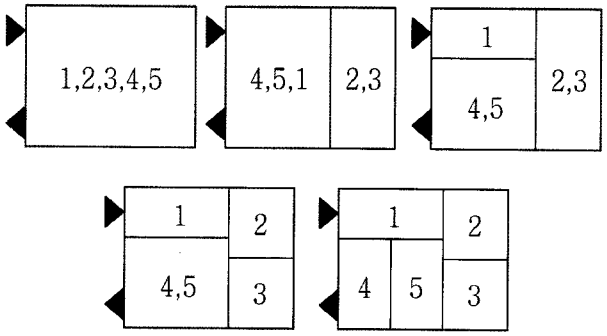

<그림 12> 컷 방향 전환

Flow shop은 공정의 흐름에 따라 물량이 이동하므로, 공정의 순서에 따라 베이의 위치를 결정하는 순서를 결정한다. 즉, 첫 번째로 배치되는 베이는 베이 1이기 때문에, 베이 1을 포함한 묶음을 우선적으로 배치하여 베이 1이 입력점에 맞닿도록 한다. 그 후에 베이 2를 배치하게 되는데, 베이 1로부터 베이 2로 물량이 이동하므로, 베이 2의 입력점은 베이 1이 된다. 이와 같은 과정으로 베이 $i$가 배치 된 후, 베이 $i+1$의 입력점은 베이 $i$의 모든 면이 되어, 베이 $i+1$은 베이 I에 인접하여 배치된다. $n$번째 공정까지 배치하여 모든 베이의 배치가 완료됐을 경우, 하나의 레이아웃이 완성된다. <표 1>과 <그림 13>은 이와 같은 과정을 순차적으로 나타낸 것이다.

배치 도중에 베이 $i$와 베이 $i+1$이 맞닿을 수 없는 상황이 있으면 이 또한 비유효한 배치로 간주한다. 그

<표 1> 슬라이싱 트리 과정

| 해당묶음 | 컷 위치 | 컷 방향 | 나뉜 후 |
|---|---|---|---|
| 45123 | 3/2 | V | 451/23 |
| 451 | 2/1 | H | 45/1 |
| 23 | 1/1 | H | 2/3 |
| 45 | 1/1 | V | 4/5 |




<그림 13> 컷 순서

즉시 초기 베이 순서를 다시 생성하여 알고리즘을 처음부터 실행하여 비유효한 배치안의 배치에 쓰이는 더 이상의 시간 소모를 단축하도록 한다. 유효한 레이아웃의 모든 베이가 완료된 후, 목적함수를 계산한다. 이웃해가 현재해보다 이동거리가 짧고, aspect ratio 등 모든 제약조건을 만족할 경우 이웃해를 현재해로 수용한다.

알고리즘 종료 조건으로서 여러 가지 방법이 있으나, 본 논문에서는 유효 레이아웃의 개수가 일정량 이상일 경우 종료한다. 종료 조건을 $j$개의 유효 레이아웃이라 한다면, 알고리즘에 의해 레이아웃을 계속 생성하며, 유효한 레이아웃이 $j$개에 달했을 경우 알고리즘을 종료한다.

## 4. 실험결과

알고리즘은 C++로 구현되어 있으며, 인텔코어 i3 2.94 GHz, 3GB 램 환경에서 실험 하였다. 제안하는 알고리즘의 평가를 위해 베이의 개수가 10개, 15개, 20개일 때 aspect ratio를 변화시킴에 따라 레이아웃의 모양이 어떻게 변하는지 살펴보기로 한다. 실험 변수는 <표 2>에 정리되어 있으며, 베이 면적의 비는 범위 내에서 임의로 생성되었다. 실험을 통해 각 aspect ratio 범위 내에서의 최소 이동 거리는 <표 3>에 정리되었다.

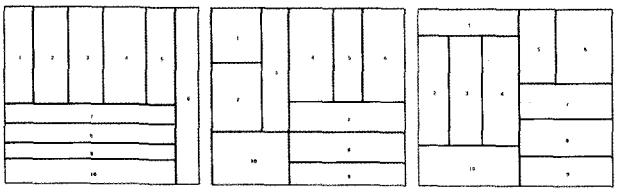
<표 2> 실험 변수

| 실험 변수 | 범위 |
|---|---|
| $W$ | 100 |
| $H$ | 100 |
| $(I^x, I^y)$ | (0,90) |
| $(O^x, O^y)$ | (0,10) |
| $S_i$ | 4~6 |
| $A$ | 0.05~0.25 |

<표 3> 실험 결과

| N | A | 총이동거리 |
|---|---|---|
| 10 | 0.1 | 274.36 |
| | 0.2 | 321.94 |
| | 0.25 | 348.62 |
| 15 | 0.07 | 295.57 |
| | 0.11 | 390.75 |
| | 0.14 | 397.93 |
| 20 | 0.05 | 393.04 |
| | 0.07 | 530.60 |
| | 0.08 | 600.03 |

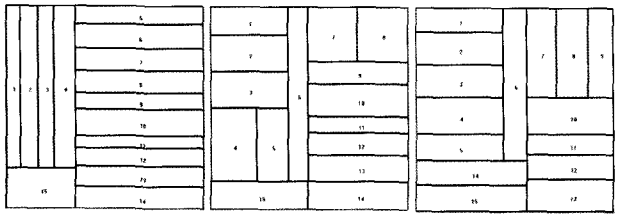
<그림 14>는 베이 개수가 10일 경우 모든 베이 $i$에 대해 $A_i$ = 0.1, 0.2, 0.25로 변화 시키며 실험한 결과다. 그림에서 알 수 있듯 aspect ratio가 작을수록 가로와 세로의 차이가 큰 직사각형 형태의 베이가 배치되는 반면 이동 거리는 줄어들었고, aspect ratio가 커짐에 따라 가로와 세로의 차이가 줄어든 반면 이동거리는 늘어나는 것을 볼 수 있다. 따라서 Flow Shop Layout Problem (이하 FSLP)에서 베이 내 설비의 모양에 따라 적절한 기준 aspect ratio를 설정하는 것이 중요하다.



(a) $A_i$ = 0.1 (b) $A_i$ = 0.2 (c) $A_i$ = 0.25

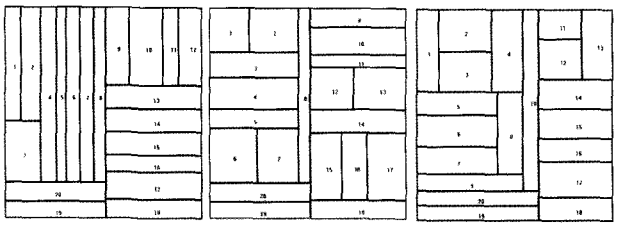
<그림 14> $A_i$값에 따른 배치결과(N = 10)

<그림 15>는 베이 개수가 15개일 경우 모든 베이 $i$에 대해 $A_i$ = 0.07, 0.11, 0.14로 변화 시키며 실험한 결과이며, <그림 16>은 베이 개수가 20개일 경우 모든 베이 $i$에 대해 $A_i$ = 0.05, 0.07, 0.08로 변화 시키며 실험한 결과이다.



(a) $A_i$ = 0.07 (b) $A_i$ = 0.11 (c) $A_i$ = 0.14

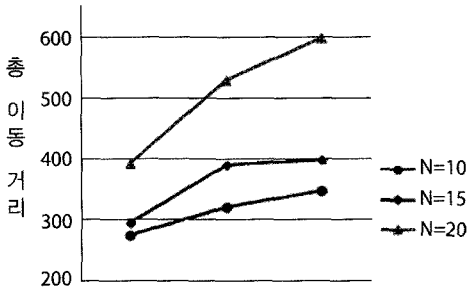
<그림 15> $A_i$값에 따른 배치결과(N = 15)



(a) $A_i$ = 0.05 (b) $A_i$ = 0.17 (c) $A_i$ = 0.08

<그림 16> $A_i$값에 따른 배치결과(N = 20)

실험은 베이의 개수를 늘려감에 따라 적절한 aspect ratio를 적용하였으며, 그 결과 베이 개수와 aspect ratio가 총이동거리에 영향을 미치는 것을 알 수 있었다. Aspect ratio의 변화에 따른 총이동거리의 변화는 <그림 17>에 정리되었다. 그림과 같이 총이동거리는 베이의 개수와 aspect ratio가 커짐에 따라 늘어난다는 것을 확인할 수 있었다. 그 이유는 aspect ratio가 작을 경우, 베이의 폭이 좁아지기 때문에 베이 중심점 간의 총이동거리가 짧아지고, 반대일 경우 베이의 폭이 커지기 때문에 aspect ratio가 작을 때보다 상대적으로 총이동거리가 커졌기 때문이다.

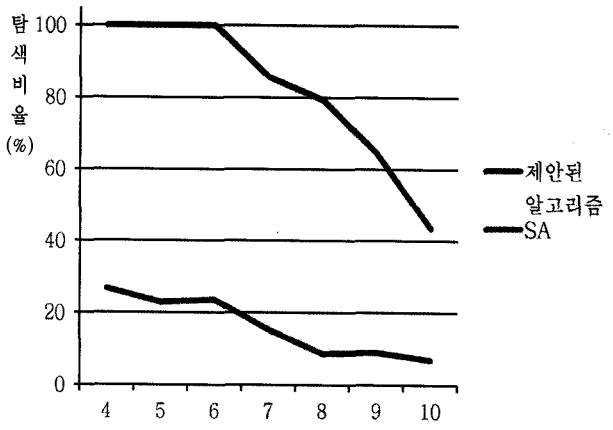



<그림 17> Aspect Ratio 변화에 따른 총 이동거리 변화

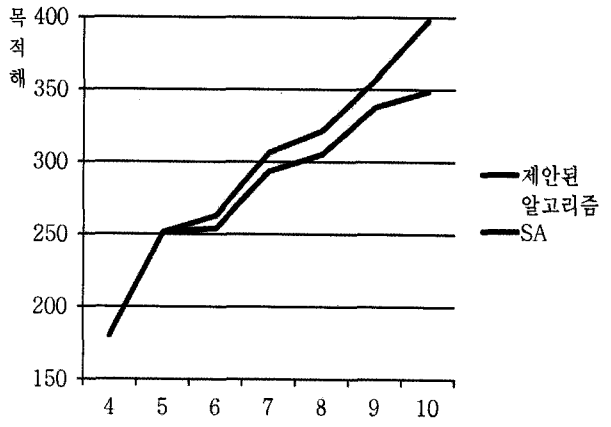
제안된 알고리즘의 효용성을 알아보기 위해 SA 기법과의 비교 실험을 하였다. Aspect ratio, 입력점과 출력점의 위치, 베이의 크기, 반복 수 등의 변수를 동일하게 지정하고, 제안된 알고리즘과 SA가 생성한 유효한 베이 배치를 찾는 비율을 실험한 결과가 <표 4>과 <그림 18>, <그림 19>에 정리되었다. <표 4>과 같이 6개 이하 베이의 실험에서 제안된 알고리즘은 100% 유효한 베이 배치를 가지는 해만 찾았으나, SA의 경우 랜덤성에 의해 비교적 낮은 비율을 보였다. 7개 이상의 베이의 실험에서도 SA에 비해 높은 비율을 보였다. 또한, 유효한 해를 탐색하는 비율이 높기 때문에, 상대적으로 더 좋은 목적해를 가지는 배치를 생성하였다. 베이 수가 많아짐에 따라서 제안된 알고리즘과 SA의 목적해의 차이가 커지는 것을 확인할 수 있었다. 따라서 제안된 알고리즘은 기존의 휴리스틱 기법 중 하나인 SA보다 흐름 공정 설계에 더 적합하다.

<표 4> 유효 레이아웃 탐색 비율 및 목적해 비교

| 베이 수 | 제안된 알고리즘 | | Simulated Annealing | |
|---|---|---|---|---|
| | 탐색비율 | 목적해 | 탐색비율 | 목적해 |
| 4 | 100 % | 180 | 26.73 % | 180 |
| 5 | 100 % | 251.43 | 22.92 % | 251.43 |
| 6 | 100 % | 253.94 | 23.57 % | 262.76 |
| 7 | 85.56 % | 293.28 | 15.32 % | 306.22 |
| 8 | 79.32 % | 305.03 | 8.76 % | 321.35 |
| 9 | 64.52 % | 337.69 | 9.06 % | 356.75 |
| 10 | 43.56 % | 348.62 | 6.98 % | 397.67 |




<그림 18> 유효 레이아웃 탐색 비율 그래프




<그림 19> 목적해 비교 그래프

## 5. 결론 및 추후연구

본 연구는 FSLP의 효율적인 베이 배치를 제공할 수 있는 휴리스틱 알고리즘을 제안하였고, 실험을 통해 flow shop의 특징을 유지하는 효율적인 베이 배치가 가능하다는 것을 보였다. 또한 aspect ratio의 변화를 통한 배치 형태의 변화 분석 및 기존 휴리스틱 기법과의 효용성 비교를 하였다.

본 연구에서 제안하는 알고리즘을 더욱 발전시킨다면 flow shop 형태를 가지는 반도체 공정 등 다양한 분야에 활용할 수 있을 것이다. 특히 베이 내 동종 또는 이종 설비의 배치와 복층 건물 구조, 흐름이 일부 상이한 제품을 고려할 수 있는 연계 방법론이 연구된다면 현장 활용성을 더욱 높일 수 있다. 또한, 기존의 설비배치 연구가 job shop 문제에 초점이 맞춰져 이뤄졌던 만큼 FSLP의 효율성에 대한 검증은 객관적으로 이뤄지지 않은 상태이다. 따라서 기존 설비배치문제와의 객관적인 효율성 비교를 위한 방법론의 연구가 필요하다.

## 참고 문헌

[1] Bazaraa, M. S.; "Computerized Layout Design : A Branch and Bound Approach," *AIIE Transactions*, 7(4) : 432-438, 1975.

[2] Chae, J. and Peters, B. A.; "A Simulated Annealing Algorithm Based on a Closed Loop Layout for Facility Layout Design in Flexible Manufacturing Systems," *International Journal of Production Research*, 44(13) : 2561-2572, 2006.

[3] Gen, M. and Chen, R.; "Genetic Algorithms and Engineering Design," John Wiley and Sons, New York, 1997.

[4] Heragu, S. S. and Kusiak, A.; "A Machine Layout Problem in Flexible Manufacturing Systems," *Operations Research*, 36(2) : 258-268, 1988.

[5] Kim J. G. and Kim Y. D.; "Layout Planning for Facilities with Fixed Shapes and Input and Output Points," *International Journal of Production Research*, 38(18) : 4635-4653, 2000.

[6] Li, H. and Love, P.; "Genetic Search for Solving Construction Site-level Unequal-Area Facility Layout Problems," *Automation in Construction*, 9(2) : 217-226, 2000.

[7] McKendall, Jr, Shang. J., and Kuppusamy, S.; "Simulated Annealing Heuristics for the Dynamic Facility Layout Problem," *Computers and Operations Research*, 33(8) : 2431-2444, 2006.

[8] Meller, R. D. and Gau, K. Y.; "The Facility Layout Problem : Recent and Emerging Trends and Perspectives," *Journal of Manufacturing System*, 15 : 351-366, 1996.

[9] Mir, M. and Iman M. H.; "A Hybrid Optimization Approach for Layout Design of Unequal-Area Facilities," *Computers and Industrial Engineering*, 39(1-2) : 49-63, 2001.

[10] Rajasekharan, M., Peters, B. A., and Yang T.; "A Genetic Algorithm for Facility Layout Design in Flexible Manufacturing Systems," *International Journal of Production Research*, 36(1) : 95-110, 1998.

[11] Scholz, D., Petrick A., and Domschke W.; "STaTS : A Slicing Tree and Tabu Search Based Heuristic for the Unequal Area Facility Layout Problem," *European Journal of Operational Research*, 197(1) : 166-178, 2009.

[12] Sipper, B. R. L. and Bulfin, Jr. R.; "Production: planning, control, and integration," McGraw-Hill, Inc., New York, 1997.

[13] Skorin-Kapov, J.; "Tabu Search Applied to the Qua-

dratic Assignment Problem," *ORSA Journal on Computing*, 2(1) : 33-40, 2000.

[14] Tam, K. Y.; "Genetic Algorithms, Function Optimization, and Facility Layout Design," *European Journal of Operational Research*, 63(2) : 322-346, 1992.

[15] Tompkins J. A., White J. A., Bozer Y. A., and Tanchoco J. M. A.; "Facilities planning, 4th Edition," John Wiley and Sons, New York, 137-285, 1996.

[16] van Camp, D. J., Carter, M. W., and Vannelli A.; "A Nonlinear Optimization Approach for Solving Facility Layout Problems," *European Journal of Operational Research*, 57(2) : 174-189, 1991.

[17] Wang, T. Y., Wu K. B., and Liu, Y. H.; "A Simulated Annealing Algorithm for Facility Layout Problems Under Variable Demand in Cellular Manufacturing Systems," *Computers in Industry*, 46(2) : 181-188, 2001.

[18] Komarudin, Kuan Yew Wong; "Solving Facility Layout Problems Using Flexible Bay Structure Representation and Ant System Algorithm," *Expert Systems with Applications*, 37(7) : 5523-5527, 2010.