Journal of the Korean Institute
of Industrial Engineers
Vol.23, No.2, June, 1997

그래프 이론을 이용한 설비배치 계획에 관한 연구*

A Study on Facility Layout Planning Using Graph Theory*

김재곤** · 이근철** · 김영대***

Jae-Gon Kim** · Geun-Cheol Lee** · Yeong-Dae Kim***

Abstract

We consider a facility layout problem with the objective of minimizing total transportation distance, which is the sum of rectilinear distances between facilities weighted by the frequency of trips between the facilities. It is assumed that facilities are required to have rectangular shapes and there is no empty space between the facilities in the layout. In this study, a graph theoretic heuristic is developed for the problem. In the heuristic, planar graphs are constructed to represent adjacencies between the facilities and then the graphs are converted to block layouts on a continual plane using a layout construction module. (Therefore, each graph corresponds to a layout.) An initial layout is obtained by constructing a maximal weighted planar graph and then the layout is improved by changing the planar graph. A simulated annealing algorithm is used to find a planar graph which gives the best layout. To show the performance of the proposed heuristic, computational experiments are done on randomly generated test problems and results are reported.

1. 서 론

설비배치란 생산 시스템의 초기 설계 단계에서 생산에 필요한 다양한 설비들의 배치를 설계하는 것을 말한다. 설비배치는 시스템의 운용 비용과 효율에 상당한 영향을 미치고, 한 번 결정된 설비배치를 바꾸기 위해서는 많은 비용과 노력이 요구되므로 설비배치의

* 이 논문은 1995년도 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음

** 한국과학기술원 산업공학과 박사과정

*** 한국과학기술원 산업공학과 부교수

최적화가 매우 중요하다. 설비배치에 대한 연구에서는 시스템 내에서의 총 물류이동거리를 최소화하는 설비배치안(block 배치)을 생성하는 문제가 흔히 다루어진다. 본 연구에서도 이러한 문제를 다루고자 한다.

설비배치 문제와 관련하여 많은 알고리듬들이 제시되어 왔다. 기존의 다양한 알고리듬들에 대한 소개와 설명은 Kusiak과 Heragu [6]에 나와 있다. 가장 잘 알려진 CRAFT [1]를 비롯하여, 기존의 대부분의 알고리듬들이 지니고 있는 주요한 단점은 최종 설비배치에서 설비들이 불규칙적이거나 이상한 모양을 갖는다는 점이다. 그 이유는 설비들이 놓여질 평면을 단위 면적을 가지는 같은 모양의 사각형들로 나누고, 각 설비들에 면적과 같은 개수의 사각형을 할당하여 설비를 배치하는 방법을 택하고 있기 때문이다.

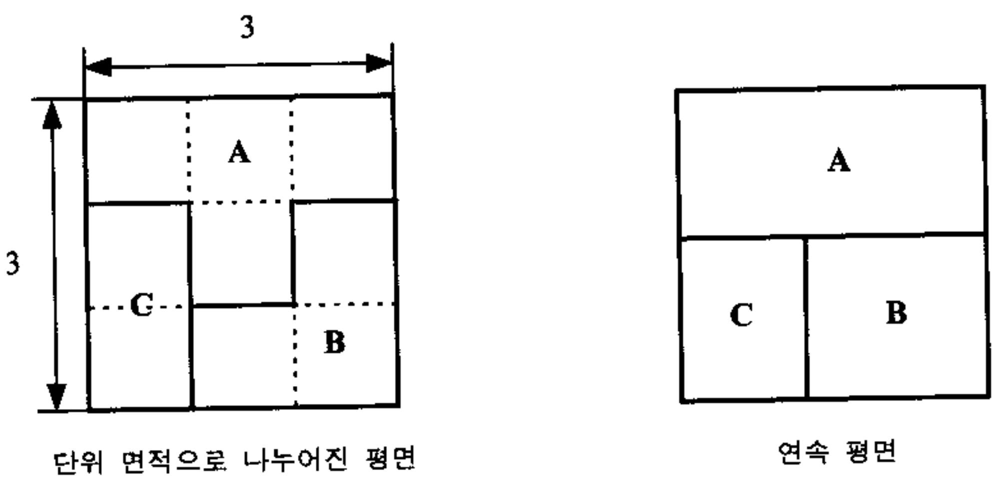

그림 1. 두 평면의 비교

이 단점을 극복하기 위해 최근의 연구에서는 설비들이 배치될 평면으로써 단위 면적들로 나누어진 평면이 아니라 연속평면(continual plane)을 사용하고 있다. 즉, 후자의 경우에는 설비들이 실수 값의 너비 및 길이를 가질 수 있는 반면 전자의 경우에는 어떤 값의 정수 배의 너비나 길이만 가질 수 있다. 예를 든다면, 설비들이 배치될 평면의 너비와 길이가 각각 3이라고 할 때, 넓이가 2, 3, 4인 세 개의 설비들을 단위 면적으로 나누어진 평면과 연속평면에 배치한 설비배치안이 그림 1과 같이 구해질 수 있다. Tam [8]은 설비를 연속평면에 배치하기 위해 슬라이싱 트리(slicing tree)를 사용하였다. 슬라이싱 트리는 길로틴 컷(guillotine cut)을 사용해 연속평면을 수직 또는 수평으로 나누어 설비들과 같은 개수의 사각형을 만드는데 필요한 정보를 나타내는 이진 트리(binary tree)이다. 가장 좋은 배치안을 주는 슬라이싱 트리를 찾기 위해, 시뮬레이티드 어닐링(simulated annealing; SA) 알고리듬을 사용되었다. 한편 Tate와 Smith [9]는 베이 구조(bay structure)를 사용하여 설비들을 연속평면에 배치하였다. 여

기서 베이란 같은 너비(혹은 길이)를 가지며 일렬로 배치된 설비들의 집합을 나타낸다. 베이의 개수와 각각의 베이에 포함될 설비들과 그들의 배치순서가 정해졌을 때, 연속평면을 수직으로 나누어 베이와 같은 개수의 사각형을 만들고, 각각을 다시 수평으로 나누어 설비들을 할당해 설비배치안을 만든다. 가장 좋은 설비배치를 주는 베이의 개수와 각각의 베이에 속하는 설비들 및 배치순서를 정하기 위해, 유전자 알고리듬(genetic algorithm)이 사용되었다.

본 연구에서는 설비들간의 총 물류이동거리를 최소화해 주는 설비배치를 연속평면에서 생성해 주는 알고리듬을 제시하고자 한다. 연속평면에서 설비배치가 이루어지므로 설비들의 길이와 너비의 값은 실수가 된다. 설비들이 놓여질 정사각형 혹은 직사각형인 연속평면의 모양 및 크기는 미리 주어져 있으며, 각 설비는 정사각형 혹은 직사각형의 모양을 가지고, 설비와 설비 사이에는 빈 공간이 존재하지 않는다고 가정한다.

본 연구에서 제시하고 있는 알고리듬은 그래프이론에 근거한 기법을 사용하고 있다. 이 기법에서는 그래프를 이용하여 설비배치안에서 어떤 설비들이 서로 인접해야 하는지를 나타낸다. 그래프에서 마디(vertex)는 설비를, 마디를 연결하는 호(edge)는 설비들의 인접 여부를, 호의 값은 인접 설비들 사이의 물류량을 나타낸다. 이를 위해, 2차원 평면 상에서 마디를 연결하는 어떠한 호도 서로 겹치지 않게 그려질 수 있는 그래프인 평면 그래프(planar graph)를 사용한다. 이는 일반적으로, 그래프가 평면 그래프이어야만 이에 의해 표현되는 설비 간의 인접여부를 만족시킬 수 있는 설비배치안이 존재하기 때문이다. 평면 그래프 중에서 특히, 하나의 호라도 더 첨가하면 더 이상 평면이 되지 않는 그래프를 최대 평면 그래프(maximal planar graph; MPG)라고 하는데, 본 연구에서는 설비 간의 인접 여부를 나타내기 위해 MPG를 사용한다. 또한, MPG가 구성되었을 때, 주어진 MPG를 이용해 연속평면에 설비들을 배치하는 설비배치 생성모듈을 개발하여 사용한다. 본 연구에서 제시한 알고리듬은 초기 MPG를 구해 초기 설비배치안을 구하고 이를 개선시켜 나가는 방법이며, 설비배치의 개선을 위해 SA 알고리듬을 사용한다.

2. 초기 MPG의 생성

초기 배치안을 얻기 위해서는 초기 MPG가 먼저 구성되어야 한다. 본 연구에서는 Leung [7]이 제시한 알고리듬을 변형한 Kim과 Kim [4]의 알고리듬을 사용하도록 한다. 이 알고리듬에서는 우선, 마디들 사이에 물류량이 가장 큰 세 개의 마디를 선택하여 삼각형 모양의 평면 그래프를 구성한다. 그리고, 이 세 마디를 제외한 나머지 마디들 중에서 다시 세 개의 마디로 삼각형을 만들어서 앞에서 이미 만들어진 그래프의 한 면(세 개의 호로 둘러싸인 부분)에 삽입시켜 새로운 평면 그래프를 만든다. 이때, 새롭게 만들어질 평면 그래프에서 모든 호의 값의 합이 최대가 되도록 평면 그래프를 만든다. 이러한 절차를 모든 마디들이 평면 그래프에 포함될 때까지 반복한다. 만일 총 마디의 개수가 3의 배수가 아닐 경우 이 절차를 반복하면 마지막으로 하나 또는 두 개의 마디가 남

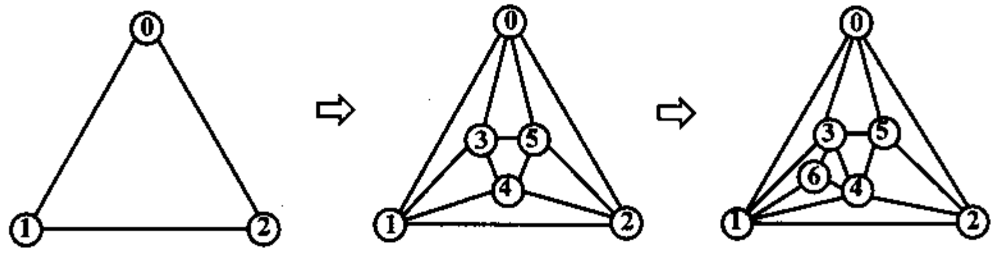

그림 2. 예제에서 초기 MPG 생성 과정

게 되는데 이 경우에는 각 마디들을 그래프의 면에 개별적으로 삽입한다. 이 알고리듬은 물류량이 많은 마디들끼리 인접하게 해주고 또, 한 마디가 동시에 여러 마디에 연결되는 우산효과(umbrella effect)를 피할 수 있게 해 준다는 점에서 효과적이다. (참고로, 우산효과가 있을 때는 평면 그래프에서 설비배치안이 쉽게 구해지지 않는다고 알려져 있다 [2].) 이 알고리듬에 대한 자세한 설명은 Kim과 Kim [4]에 나와 있다.

위의 알고리듬을 설비의 개수가 6인 예제에 적용한 결과가 그림 2에 나타나 있다. 이 예제에서 사용된 설비들과 관련한 데이타는 표 1에 주어져 있다. 여기서 설비 0은 외부(exterior)를 나타내는 가상 설비이다. 초기 MPG를 구성하기 위해 설비 0, 1, 2가 처음에 선택되었고, 그 다음 설비 3, 4, 5 그리고, 마지막으로 설비 6이 선택되었다.

3. 설비배치 생성 모듈

이 모듈은 주어진 MPG로부터 설비 간의 인접함과 설비들의 모양을 동시에 고려하면서 연속평면에 설비배치안을 생성한다. 여기서는 1절에서 언급한 바 있는 Tate와 Smith [9]에서 사용된 베이 구조(bay structure)를 사용하여 설비를 연속 평면에 배치한다. 베이 구조의 설비배치를 만들기 위해서는 베이의 개수와 각 베이에 속하게 될 설비들 및 각 베이에서의 설비들의 배치순서가 정해져야 한다. Tate와 Smith [9]에서는 그러한 정보들을 미리 모두 만들어 놓고 한꺼번에 모든 설비들을 배치하였지만, 본 모듈에서는 주어진 정보를 이용해 하나의 설비를 배치하고 지금까지 배치된 설비들에 대한 정보를 이용하여 다시 다음 설비를 배치하는 방식으로 설비들을 하나씩 배치한다.

표 1. 예제에 관한 데이타

설비	면적	설비간의 물류량					
		0	1	2	3	4	5
1	1	20					
2	2	25	0				
3	1.5	0	8	5			
4	2	0	12	7	0		
5	1.5	0	8	10	14	7	
6	1	0	9	0	6	4	0

표 1에서 주어진 예제를 사용하여 얻은 초

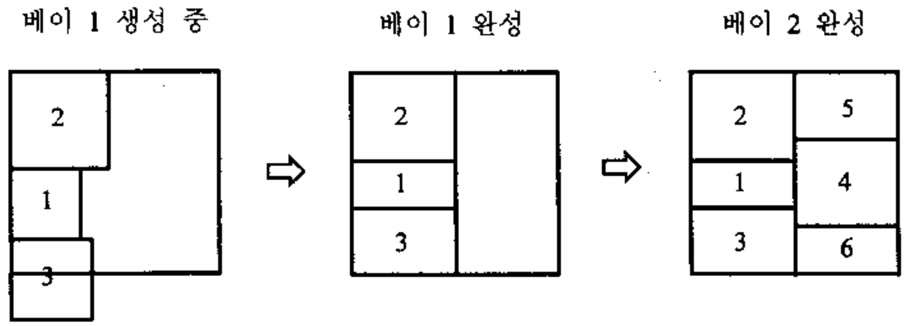

그림 3. 예제에서 초기 설비배치안의 생성

기 MPG(그림 2)로부터 설비배치안을 만들어 내는 과정을 예로 들어 설명하도록 하겠다. 이 예제에서 설비들이 놓여질 평면의 너비와 길이는 모두 3으로 주어져 있다고 가정한다. 주어진 예를 사용했을 때 설비배치가 생성되는 과정은 그림 3에 나와 있다. 우선, 외부는 가상 설비로서 설비 0으로 표시되며, 이미 배치가 완료되어 가상 베이인 베이 0에 포함되어 있다고 가정한다. (여기서 가상 베이란 알고리듬의 전개를 위해 도입한 것으로 실제 설비배치안에서는 나타나지 않는다.) 그런 다음, 다음에 놓여질 설비(이하 배치설비라 칭함)를 결정하여 연속평면 아래쪽 방향으로 배치한다. 여기서, 배치설비를 결정할 때, 아직 배치되지 않은 설비들 중에서 배치설비가 놓여질 위치 주변에 이미 놓여져 있는 (가상 설비를 포함한) 설비들과 MPG에서 인접관계가 가장 많은 설비를 선택하도록 한다. 이와 같이 함으로써, MPG에서 주어진 설비들의 인접관계를 실제 설비배치안에서 가능한 한 많이 만족시킬 수 있다. 첫 번째 배치설비는 주어진 연속평면의 맨 왼편 위쪽 가장자리에 놓여진다. 따라서 첫 번째 배치설비는 반드시 외부(설비 0)와 접하게 되므로 MPG에서 설비 0과 호를 가지는 설비들(설비 1, 2, 3, 5)을 찾는다. 이중에서 설비 0과 가장 많은 물류량을 가지는 설비, 즉 설비 2를 첫 번째 배치설비로 선택한다. 배치설비는 배치될 때 정사각형 모양을 가진다. 두 번째 놓여질 설비는 앞에서 선택된 설비(설비 2)의 바로 아래에 놓여지는데 이 경우 이 설비는 반드시 설비 0과 설비 2와 접하게 된다. 따라서 주어진 MPG에서 설비 0, 설비 2와 호로 연결된 설비들(설비 1과 5) 중에서 설비 0, 설비 2와 가장 물류량이 많은 설비, 즉 설비 5를 배치설비로 선택한다. 이와 같은 방법을 반복하여 새로운 배치설비를 선택하여 바로 이전에 놓여진 설비의 아래쪽에 배치한다. 만약 배치설비를 배치했을 때, 배치설비가 주어진 연속평면의 아래쪽 경계를 넘어가게 되면 지금까지 선택된 설비들로 베이를 구성하여 설비들을 배치한다. 이 과정에서 설비들의 모양이 바뀌게 되어, 같은 베이에 속하는 설비들은 동일한 너비를 가지게 된다. 주어진 예에서는 설비 3을 배치하였을 때, 연속평면의 아래쪽 경계가 넘어가게 되어, 설비 2, 설비 1, 설비 3들로 베이 1을 구성하였다. 베이 1의 너비는 설비 2, 1, 3의 면적의 합을 주어진 연속평면의 길이로 나눈 값이 되며 설비 2, 1, 3들은 이 너비를 가진다. 앞의 과

정을 반복하여 연속평면의 왼쪽부터 오른쪽으로 새로운 베이를 하나씩 순서대로 만들어 나간다.

위에서 설명한, 주어진 MPG로부터 베이 구조의 설비배치안를 만드는 절차를 각 단계별로 정리하면 다음과 같다. 우선 이를 위해 사용하는 기호를 정의한다.

N 설비의 개수

e 배치설비

W 설비가 놓여질 연속평면의 너비

w 설비가 놓여질 평면에서 이미 만들어진 베이들을 제외한 나머지 사각형의 너비

Ψ 이미 배치된 설비들의 집합

k 현재 고려되고 있는 (새롭게 만들어질) 베이

Ω_k 베이 k에 속하는 설비들의 집합

u_k 베이 k의 너비

f_1 이미 배치된 설비들 중에서, 배치설비를 배치할 때, 배치설비의 위쪽 경계선과 접하게 되는 설비

f_2 이미 배치된 설비들 중에서, 배치설비를 배치할 때, 배치설비의 왼쪽 경계선과 접하게 될 가능성이 있는 설비들 중에서 가장 먼저 배치된 설비

f_3 이미 배치된 설비들 중에서, f_2의 아래쪽 경계선과 접하고 있는 설비

$p(i)$ 설비 i 바로 이전에 배치된 설비

$n(i)$ 설비 i 바로 이후에 배치된 설비

$a(i, j)$ MPG에 호 (i, j)가 존재하면 1, 그렇지 않으면 0

Q 현재 고려되고 있는 베이가 최종 베이(연속평면의 가장 우측에 존재하는 베이)이면 1, 그렇지 않으면 0의 값을 가지는 변수

C 현재 고려되고 있는 베이에 속하게 될 설비들이 완전히 결정되었으면 1, 그렇지 않으면 0의 값을 가지는 변수

설비배치안 생성을 위한 알고리듬은 다음과 같다.

[알고리듬]

Step 0. $w = W$, $f_1 = 0$, $f_2 = 0$, $f_3 = 0$, $k = 1$, $\Psi = \{0\}$, $\Omega_0 = \{0\}$.

Step 1. 아직 배치되지 않은 설비들의 평균 면적 m을 구한다. 만약 $w < 1.5\sqrt{m}$이면, $Q = 1$로, 그렇지 않으면 $Q = 0$로 둔다. (이 조건은 제일 마지막 베이에서 길이가 긴 모양의 설비가 생기지 않게 하기 위해 도입된 것으로 경험적으로 결정되었다) $\Omega_k = \Phi$, $C = 0$으로 둔다.

Step 2. 배치설비 e를 다음과 같이 선택한다. $e = \underset{1 \le i \le N, i \notin \Psi}{\operatorname{argmax}} [2 \cdot a(i, f_1) + a(i, f_2) + a(i, f_3)]$. 만약 e가 여러 개 존재하면 그 중에서 설비 f_1, f_2, f_3들과 물류량이 가장 큰 설비를 선택한다.

Step 3. 1) $p(e) \in \Omega_{k-1}$이면, 설비 e를 아직 베이에 의해 할당되지 않은 연속평면의 사각형 부분의 맨 왼편 위쪽 가장자리에 배치한다. 그렇지 않으면, 설비 e를 설비 $p(e)$의 바로 아래쪽에 배치한다.

2) 만약 $Q = 0$이면 설비 e는 임시로 정사각형 모양을 가지고, $Q = 1$이면 설비 e는 너비가 w인 직사각형 모양을 가진다.

3) $\Omega_k \leftarrow \Omega_k \cup \{e\}$, $\Psi \leftarrow \Psi \cup \{e\}$.

4) 설비 e를 배치했을 때, 설비 e가

주어진 연속평면의 아래쪽 경계선을 벗어나거나 아직 배치되지 않고 남아 있는 설비가 더 이상 없을 때는 C = 1로 두고 step 4로 간다. 그렇지 않으면 step 5로 간다.

Step 4. 집합 Ω_k에 속하는 설비들로 구성된 새로운 베이(베이 k)를 만든다. 베이 k의 너비 u_k는 Ω_k에 속하는 설비들의 면적의 합을 연속평면의 길이로 나눈 값이 된다. w의 값을 $w = w - u_k$로 수정한다. 만약 $w > 0$이면 $k = k + 1$로 두고 step 5로 간다. 그렇지 않으면($w = 0$) 알고리듬을 종료한다.

Step 5. 1) 만약 $C = 0$이면, $f_1 = e$, $f_2 = u$로 둔다. (여기서 u는 Ω_{k-1}에 속하면서 현재 설비배치안에서 설비 f_1에 접하는 설비 중에서 가장 나중에 배치된 설비임.) 만약 $C = 1$이면, $f_1 = 0$, $f_2 = v$로 둔다. (여기서 v는 Ω_{k-1}에 속하는 설비 중에서 가장 먼저 배치된 설비임.)

2) 만약 $n(f_2) \in \Omega_{k-1}$이면, $f_3 = n(f_2)$로 두고, 그렇지 않으면 $f_3 = 0$로 둔다.

3) 만약 $C = 0$이면 step 2로 간다. 그렇지 않으면, step 1로 간다.

위에서 제시한 알고리듬을 앞서 사용한 예제에 적용시켜 본 결과가 표 2에 주어져 있다.

제시된 알고리듬에서 Step 1은 새로운 베이를 만들기 위해 배치설비들을 선택하기에 앞서 이 베이가 최종 베이가 되어야 하는지 아닌지를 결정한다. 만약 아직 배치되지 않고 남아 있는 설비들로 하나의 베이를 구성해도 설비들의 모양이 이상해지지 않을 것이라 판단되면, 즉 설비의 너비가 길이에 비해 많이 크지 않을 것이라 판단되면 새로운 베이는 최종 베이(Q = 1)가 된다. 제시된 알고리듬에서는 최종 베이의 여부를 판단하기 위한 조건으로 "$w < 1.5\sqrt{m}$"을 (경험적으로 찾아내어) 사용하였다.

배치설비를 선택하는 데 있어 Step 2의 방법을 사용하는 이유는 다음과 같다. 설비 f_1, f_2, f_3들은 배치설비가 놓여지게 될 위치 주변에 이미 놓여져 있는 설비들이므로 설비배치안에서 배치설비와 인접할 가능성이 높다. 특히, 설비 f_1은 최종 설비배치안에서 배치설비

표 2. 예제에서 초기 MPG로부터 설비배치안을 만드는 과정

k	f_1	f_2	f_3	Q	e	C	비 고
1	0	0	0	0	2	0	
1	0	2	0	0	1	0	
1	1	0	0	0	3	1	C = 1이므로 설비 2, 1, 3 으로 베이 1을 만든다.
2	0	2	1	1	5	0	Q = 1이므로 지금 부터 만들어질 베이 2가 최종 베이가 된다.
2	5	2	1	1	4	0	
2	4	3	0	1	6	1	설비배치안 생성 완료

와 반드시 인접하게 되고, 설비 f_2와 f_3는 최종 설비배치안에서 배치설비와 반드시 접하지는 않지만 접할 가능성이 크다. 그러므로 step 2에서 배치설비를 선택할 때 MPG에서 f_1과 호를 가지고 있는 설비에 대해서는 가장 높은 가중치 2을 주고, f_2, f_3과 호를 가지고 있는 설비에 대해서는 각각 가중치 1을 줘서 가중치의 총 합이 가장 큰 설비를 배치설비로 선택한다. 이와 같은 방법으로 배치설비를 선택함으로써, MPG에서 나타난 설비 간의 인접관계를 실제 설비배치 안에서 가능한 한 많이 만족시킬 수 있다.

배치설비가 결정되면 배치설비를 배치하기 전에 배치설비의 모양을 결정하여야 하는데, 배치설비의 모양은 이 설비가 속하게 될 베이에 들어갈 설비들이 모두 결정되기 전까지는 정해질 수 없다. 따라서 배치설비의 모양도 그 전에는 결정될 수 없다. 그러므로 Step 3에서 배치설비의 모양을 임시로 정사각형으로 둔다. 배치된 설비들의 최종 모양은 Step 3의 조건에 의해 베이에 들어갈 설비들이 모두 결정되었을 때, Step 4에서 베이가 만들어지면서 결정되게 된다. 이때 베이 내에서 설비들은 같은 너비를 가지며 설비 간의 배치순서, 즉 상대적 위치는 바뀌지 않는다.

제시된 알고리듬을 사용하여 만들어지는 설비배치안은 MPG에 의해 나타내어지는 설비 간의 인접조건을 항상 모두 만족시키지는 않는다. 비록 Step 2에 의해 가능한 많은 인접조건들을 만족시킬 수는 있지만 설비들의 모양과 면적의 제약 때문에 모든 인접조건을 정확히 만족시키기는 불가능하기 때문이다.

4. 설비배치의 개선

2절에서 제시한 방법으로 초기 MPG를 만들고, 이를 이용해 앞절에서 제시한 설비배치 생성모듈을 사용하여 초기 설비배치안을 만들어 낸다. 초기 설비배치안에서는 물류량이 많은 설비들끼리 서로 인접하도록 설비들을 배치하여 목적값인 총 물류이동거리를 줄이도록 노력하였다. 하지만, 대부분의 경우에 있어 초기 설비배치안이 최소 물류이동거리를 보장해 주지는 못한다. 특히 설비들의 개수가 늘어날 수록 가능한 설비배치안의 수는 기하급수적으로 늘어나므로 이 경우에는 초기 배치안보다 더 우수한 목적값을 가지는 설비배치안들이 더 많이 존재할 수 있을 것이다. 이 절에서는 주어진 설비배치안을 개선시키는 방법을 제시하도록 하겠다.

3절에서 제시한 설비배치 생성모듈을 사용하여 설비들을 배치하면, MPG가 바뀔 때마다 새로운 설비배치안이 구해진다. 따라서 MPG를 바꿔 나가면서 설비배치안을 개선할 수 있다. 본 연구에서는 이러한 개선 방법을 사용하여 해를 향상시키는 기법을 적용하기로 한다.

이 연구에서는 MPG를 바꾸기 위해 마디 교환 방법과 마디 이동 방법을 사용한다. 전자는 그래프에서 임의의 두 마디를 서로 교환하는 것이고, 두 번째 방법은 그래프의 어떤 면에 속해, 3개의 호와 연결되어 있는 마디들 중 하나를 다른 면으로 이동 시키는 것이다. 이 두 가지 방법을 사용하여 Kim과 Kim [4]은 네 가지 설비배치 개선 방법을 제시하였다. 이 네 가지 방법은 "숨은 호(hidden edge; 설비 간에 물류량이 있음에도 불구하

고 MPG에는 빠져 있는 호)에 연결된 마디 교환"법(vertex interchanges with hidden edge sets; INHE), "마디 이동 후 마디 교환"법 (vertex movements followed by vertex interchanges; MOIN), "마디 교환 후 마디 이동" 법(vertex interchanges followed by vertex movements; INMO), 마지막으로 "교환 및 이동"법(simultaneous interchanges and movements; SIM)이다. 숨은 호는 마디 교환을 통해 호로 바뀌어질 수 있는데, INHE에서는 이러한 마디 교환들 가운데서 목적값을 가장 많이 줄일 수 있는 대안을 선택하여 MPG 및 설비배치안을 개선시킨다. 이와 같은 개선은 더 이상 목적값을 줄일 수 없을 때까지 반복된다. MOIN은 목적값을 가장 많이 감소하게 하는 마디 이동을 더 이상의 개선이 되지 않을 때까지 계속 수행한 뒤, 다시 마디 교환을 같은 방법으로 수행하여 설비배치안을 개선시키는 방법이다. INMO는 MOIN과 비교해서 마디 교환이 마디 이동보다 먼저 수행된다는 점에서 다르다. SIM은 마디 교환과 마디 이동의 모든 가능한 대안들을 다 고려해서 그 중에서 목적값을 가장 많이 개선시키는 대안을 선택하여 더 이상 목적값을 개선할 수 없을 때까지 계속 수행하는 방법이다.

이러한 개선 방법들은 나쁜 해(설비배치안)로의 이동을 허용하지 않으므로 부분 최적해(local optimum)에 빠질 수 있는 단점이 있다. 부분 최적해에서 벗어나기 위해 본 연구에서는 SA(simulated annealing) 알고리듬을 사용하였다. SA 알고리듬은 Kirkpatrick *et al.* [5]에 의해 제시된 알고리듬으로서 흔히 조합형 최적화 문제를 풀기 위해 사용된다. Greedy 한 구역탐색(local search) 방법들과는 달리 SA 알고리듬에서는 좋은 해로의 이동은 물론이고 나쁜 해로의 이동도 부분적으로 허용함으로써 부분 최적해에서 벗어나지 못하는 것을 방지할 수 있다. 나쁜 해로의 이동은 확률적으로 허용되는데, 그 확률은 $\exp(-\Delta/T)$로 주어진다. 여기서 T는 조절 인자로써 온도라고 하며, Δ는 현재 해의 목적값과 인접해의 목적값의 차이이다. 알고리듬의 수행 초기에는 온도를 높게 함으로써 해의 이동이 일어날 확률을 높게 하고 알고리듬이 진행될수록 온도를 서서히 떨어뜨려 최적해에서 해의 이동이 멈추게 한다.

본 연구에서 제시된 SA 알고리듬에서는 해의 형태를 MPG로 표현하여, 이를 변경시켜가며 가장 좋은 설비배치안을 주는 MPG를 찾는다. 인접해는 마디 교환이나 마디 이동에 의해 생성하며 초기 온도를 정하는 방법과 온도를 떨어뜨리는 방법은 다음과 같이 정하였다. 초기온도를 T_0라 하면 그 온도에서 나쁜 해로의 이동이 허용될 확률이 대략적으로 정해진 값 F_0가 되도록 정한다. 이를 위해, 시험적으로 $2N$개의 이동을 연속적으로 행하여 목적값의 평균 증가량 $\overline{\Delta}$을 나쁜 해로의 이동에 대해서만 계산하여, 식 $\exp(-\overline{\Delta}/T_0) = F_0$로부터 T_0를 정한다. 온도는 정해진 횟수의 해의 이동이 수행 될 때까지는 그대로 유지되는데, 온도가 동일하게 유지되는 구간을 에폭(epoch)이라 한다. 제시된 알고리듬에서 에폭의 길이는 예비실험을 통해 $3N$으로 정하였다. 온도는 한 에폭이 끝날 때마다 일정한 비율로 줄어들게 한다. 즉, k 번째 에폭에서의 온도(T_k)는 $T_k = \gamma \cdot T_{k-1}$으로 정해진다. 여기서 γ는 냉각비라 불리는 인자로

서 1보다 작은 값을 가진다. 제시된 알고리듬에서는 일련의 실험을 통해, 계산시간 및 해의 질적인 면에서 좋은 결과를 준 $F_0 = 0.68$ 그리고 $\gamma = 0.98$의 값을 선택하여 사용하였다. 알고리듬의 종료조건으로는 Johnson *et al.* [3]이 제시한 방법을 사용한다. 이 방법에서는 카운터를 사용하는데, 해의 개선이 없이 한 에폭이 끝나면 카운터 값을 하나 증가시키고, 이제까지 가장 좋은 새로운 해가 발견되면 다시 0으로 돌려놓는다. 카운터의 값이 2(일련의 실험을 통해 선택된 값임)가 되면 SA 알고리듬을 종료시킨다.

5. 실험 결과

제시된 알고리듬(이후 GSA라고 칭한다)을 CRAFT [1]와 Kim과 Kim [4]의 다른 4개의 알고리듬(INHE, MOIN, INMO, SIM)과 비교하여 보았다. Kim과Kim [4]은 설비배치안을 만들기 위해 CRAFT와 마찬가지로 설비가 놓여질 평면을 단위 면적을 가지는 동일한 모양의 사각형으로 나누어 설비들의 면적에 해당하는 개수의 사각형을 할당하여 설비를 배치하는 방법을 사용하였지만, 본 연구에서는 3절에서 제시된 연속 평면에 설비를 배치하는 방법을 사용하여 INHE, MOIN, INMO, SIM을 적용하였다. CRAFT는 FORTRAN으로, 나머지 알고리듬들은 C 언어로 작성되었고, 실험은 80486 프로세서가 장착된 PC에서 수행되었다.

비교를 위해 총 80개의 문제를 무작위로 생성하였는데, 설비의 개수 (12, 18, 24, 30)에 따라 각각 20개씩의 문제를 만들었다. 우선, 표 3과 같이 설비의 개수에 따라 설비들의 면적에 관한 데이타와 연속평면의 규격에 대한 데이타를 각각 2가지씩 생성하였다. 그리고 설비의 개수와 면적의 8가지 모든 조합에 대해 설비 간의 물류량을 10개씩 생성하였다. 물류량은 [0, 10] 사이의 정수값을 가지는 일양분포를 사용해 생성하였다. 외부는 그래프 이론을 이용한 알고리듬들에서만 사용되고 CRAFT에서는 사용되지 않기 때문에, 외부에 의한 영향을 없애기 위해 모든 문제에서 외부와 설비들 간의 물류량은 0으로 두

표 3. 테스트 문제에 관한 데이터

설비의 개수	바닥의 규격 (가로, 세로)	설비들의 면적
12	(5, 5)	2, 2, 3, 2, 1, 1, 2, 3, 4, 1, 1, 3
		5, 1, 2, 4, 1, 1, 2, 2, 1, 1, 3, 2
18	(6, 6)	1, 2, 1, 2, 3, 2, 4, 1, 2, 1, 2, 1, 3, 2, 2, 1, 4, 2
		1, 2, 2, 1, 1, 3, 1, 6, 2, 2, 1, 1, 2, 2, 1, 2, 5, 1
24	(7, 7)	1, 2, 3, 2, 3, 2, 3, 2, 2, 2, 1, 4, 1, 2, 2, 1, 3, 2, 1, 3, 1, 2, 1, 3
		1, 1, 1, 4, 2, 1, 2, 3, 4, 3, 2, 1, 2, 1, 4, 5, 1, 2, 1, 2, 3, 1, 1, 1
30	(8, 8)	2, 2, 3, 1, 4, 3, 2, 1, 2, 3, 2, 3, 2, 2, 1, 2, 3, 4, 3, 2, 1, 2, 2, 2, 4, 1, 1, 2, 1, 1
		6, 5, 1, 1, 1, 2, 3, 2, 1, 1, 2, 4, 2, 1, 3, 3, 1, 1, 2, 2, 1, 1, 2, 4, 2, 1, 1, 2, 2, 4

었다. CRAFT 알고리듬은 초기 배치안을 필요로 하기 때문에, 이 실험에서는 임의로 생성된 3개의 초기 배치안으로 각 문제들을 3번씩 풀었다.

표 4. 알고리듬들의 성능 비교

알고리듬	평균 RPR					*NBS	평균 순위
	N=12	N=18	N=24	N=30	전체		
GSA	1.1	0.9	0.6	0.3	0.59	44	1.83
INHE	6.2	5.2	2.7	3.0	4.47	5	5.18
MOIN	4.5	3.7	3.3	1.7	2.66	9	3.91
INMO	3.8	4.4	2.7	2.4	2.65	8	3.79
SIM	3.7	4.0	3.1	7.9	2.68	7	3.98
CRAFT-a	8.5	6.1	5.2	6.8	5.3	0	5.73
CRAFT-b	5.2	3.1	3.4	3.7	3.1	10	3.79
CRAFT-w	12.2	9.5	7.4	10.4	7.9	0	7.58

*NBS : 가장 좋은 해를 찾는 횟수

표 5. 알고리듬들의 평균 수행 시간 (초)

알고리듬	설비들의 개수			
	12	18	24	30
GSA	10.7	41.5	98.5	216.2
INHE	0.3	2.7	13.1	19.8
MOIN	0.3	2.9	15.2	45.2
INMO	0.4	3.3	17.2	53.7
SIM	0.5	3.5	17.1	54.9
CRAFT	0.2	0.8	2.1	4.0

표 4와 5는 알고리듬들의 수행 결과를 보여 주고 있다. 각 알고리듬의 수행결과로 나오는 설비배치안을 알고리듬의 해라고 하고 설비배치안의 목적값, 즉 총 물류이동거리를 해의 값이라 부르기로 한다. 표 4에서, CRAFT-a, CRAFT-b, CRAFT-w 는 각 문제에 대해 3번씩 수행된 CRAFT의 해들 중에서 평균 값과 가장 좋은 값 그리고 가장 나쁜 값들을 나타낸다. 이 연구에서 알고리듬들의 수행도 비교를 위해 사용되는 평가 척도들은 3가지로서, 상대적 수행 비(relative performance ratio; RPR)와 가장 좋은 해를 주는 횟수 그리고 해의 평균 등급(rank)이다. 여기서, V_a를 알고리듬 a가 주는 해의 값이라 하고 V_b를 모든 알고리듬 가운데 가장 좋은 해의 값이라 하면, 알고리듬의 a의 RPR은 $(V_a - V_b) / V_b \times 100$으로 표현되어진다. 표 4에서 볼 수 있듯이, 제시된 알고리듬 GSA가 모든 평가 척도에서 기존의 알고리듬보다 우수하다. 특히, GSA 알고리듬의 성능은 설비의 개수가 많아질수록 더욱 좋아짐을 보이고 있다. 비록 GSA가 가장 긴 수행시간을 요하지만, 큰 문제에 대해서도 수행시간이 그리 길지 않고, 또 수행시간이 길다 하더라도 좋은 해를 얻을 수 있다면 이는 충분히 보상될 수 있으므로 GSA가 설비배치 문제를 푸는데 우수한 알고리듬이라고 할 수 있겠다. 실제로, 설비배치 문제는 자주 풀려져야 하는 문제가 아니므로 빠른 알고리듬 보다는 좋은 해를 주는 알고리듬이 필요하다고 하겠다. 또한, GSA를 비롯한 INHE, MOIN, INMO, SIM은 결과로 얻어지는 최종 설비배치안에서 불규칙적이거나 이상한 형태를 가지는 설비가 없다는 점에서 기존의 잘 알려진 CRAFT와 비교해 훨씬 우수하다고 할 수 있다.

6. 결 론

본 연구에서는 총 물류 이동 거리를 최소화하는 설비배치 문제를 풀기 위해 그래프이론을 이용한 알고리듬을 제시하였다. 설비들

간의 인접함의 여부를 나타내기 위해 MPG를 사용하였으며, 주어진 MPG를 연속평면에서 설비배치안로 바꾸기 위한 방법을 제시하였다. 이 방법을 사용하면 설비배치안에서 모양이 이상하거나 불규칙적인 설비는 생기지 않게 되는 장점이 있다. 초기 배치안을 구하기 위해 먼저 Kim과 Kim [4]이 제시한 방법을 사용하여 MPG를 구하였으며, 설비배치를 개선시키기 위해 가장 좋은 설비배치를 주는 MPG를 SA 알고리듬을 사용해 찾았다. 실험을 수행한 결과, 제시된 알고리듬이 CRAFT와 Kim과 Kim [4]이 제시한 4개의 알고리듬들보다 좋은 결과를 주었다.

본 연구에서 MPG를 설비배치안으로 바꾸는 방법을 제시하였지만, 설비의 개수가 많아지거나 설비들 간의 면적의 차이가 많이 날 경우, MPG에서의 설비들 간의 인접 관계를 설비배치안에 반영하기가 힘들어진다. 따라서 MPG에서의 설비들 간의 인접 관계를 설비배치안에서 더욱 정확히 반영해줄 수 있는 방법에 대한 연구가 필요하다고 하겠다.

참 고 문 헌

[1] Armour, G. C., and Buffa, E. S., "A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities", *Management Science*, Vol. 9, pp. 294-309, 1963.

[2] Giffin, J. W., Foulds, L. R., and Cameron, D. C., "Drawing a block plan from a relationship char with graph theory and microcomputer", *Computers and Industrial Engineering*, Vol. 10, pp. 109-116, 1986.

[3] Johnson, D., Aragon, C., McGoech, L., and Schevon, C., "Optimization by Simulated Annealing: an Experimental Evaluation; Part I, Graph Partitioning", *Operations Research*, Vol. 37, pp. 862-892, 1989.

[4] Kim, J-Y., and Kim, Y-D., "Graph Theoretic Heuristics for Unequal-sized Facility Layout Problems", *Omega*, Vol. 23, pp. 391-401, 1995.

[5] Kirkpatrick, S., Gelatti, C. D., and Vecchi, M. P., "Optimization by Simulated Annealing", *Science*, Vol. 220, pp. 671-680, 1983.

[6] Kusiak, A., and Heragu, S. S., "The Facility Layout Problem", *European Journal of Operational Research*, Vol. 29, pp. 229-251, 1987.

[7] Leung, J., "A New Graph Theoretic Heuristic for Facility Layout", *Management Science*, Vol. 38, pp. 594-605, 1992.

[8] Tam, K. Y., "A Simulated Annealing Algorithm for Allocating Space to Manufacturing Cells", *International Journal of Production Research*, Vol. 30, pp. 63-87, 1992.

[9] Tate, D. M., and Smith, E. A., "Unequal-Area Facility by Genetic Search", *IIE Transactions*, Vol. 27, pp. 465-472, 1995.

96년 9월 최초 접수, 97년 6월 최종 수정