

# 입력점과 출력점이 고정된 설비배치설계를 위한 휴리스틱 알고리즘

정동화 · 서윤호<sup>†</sup>

고려대학교 정보경영공학과

## Heuristic Algorithm for Facility Layout Design with Fixed Input and Output Points

Donghwa Jeong · Yoonho Seo

Department of Information Management Engineering, Korea University, Seoul 136-713

The facility layout problem (FLP) involves the positioning of facilities within a given workplace in order to minimize the material flow distance between facilities. In this paper, we focus on the FLP in which the each facility has a rectangular shape and an input and output points. We develop a heuristic algorithm in which the facilities are arranged according to the layout algorithm using center of gravity and then simulated annealing algorithm is applied to improve the solution. A comparison of the computational results with exiting algorithms shows that the proposed algorithm provides better solutions in reasonable range of computational time.

**Keyword:** facility layout problem, center of gravity, candidate bonding point, simulated annealing algorithm

### 1. 서론

제품수명주기의 단축과 소비자 수요의 다양성으로 인하여 유연성과 효율성을 동시에 달성해야 하는 제조시스템이 기업에게 요구되고 있다. 이러한 흐름에 맞추어 유연 제조시스템(Flexible Manufacturing System, FMS)과 셀 제조시스템(Cellular Manufacturing System, CMS)이 연구되어 왔으며, 최근에는 재구성 가능 제조시스템(Reconfigurable Manufacturing System, RMS)이 연구되고 있다(Kim and Kim, 2000; Koren *et al.*, 1999). 이러한 시스템에서 설비배치, AGV 경로설계와 같은 문제들이 제조라인의 신속하고 효율적인 구성을 위하여 연구되어왔다. 또한 제조시스템에서 총운영비용의 20~50%는 자재조달비용이 차지하고 있으며, 효율적인 설비배치는 이 비용의 10~30%를 줄일

수 있다고 보고되어 있다(Tompkins, 1996).

설비배치문제는 설비를 작업장 내에서 총 물류비용을 최소화 할 수 있도록 배치하는 문제로 정의된다(Heragu and Kusiak, 1988). 작업장 내에서 설비는 생산에 필요한 기계 및 기타 장비로 구성되어있다. 설비배치문제에서 배치의 평가는 설비간의 거리와 물동량의 곱으로 계산되며, 설비간의 거리는 설비의 중심점간의 직선거리와 직각거리 등을 사용하여 계산하여왔다(Tompkins and White, 2003). 그러나 입력점과 출력점의 위치가 정해진 설비를 배치하는 경우에 중심점간의 거리는 실제 거리를 반영하지 못하는 단점이 있었다. 따라서 입력점과 출력점의 위치가 정해진 설비배치를 위한 알고리즘이 연구되어왔다.

본 연구의 목적은 유효한 시간 내에 입출력 점이 고정되고 4

본 연구는 한국과학재단 특정기초연구(R01-2006-000-10941-0)지원에 의해 연구되었음.

<sup>†</sup>연락처 : 서윤호 교수, 136-713 서울 성북구 안암동 5가 1번지 고려대학교 정보경영공학과, Fax : 02-929-5888,

E-mail : yoonhoseo@korea.ac.kr

2007년 01월 접수, 2회 수정 후 2007년 05월 게재확정.

개의 방위(기본방위, 90°, 180°, 270°)를 가지는 직사각형 모양의 설비들 간의 최단직각거리와 물류량의 곱의 합을 최소화하는 설비배치를 찾는 것이다. 이를 위하여 본 연구에서는 유효한 시간 내에 효율적인 설비배치를 제공할 수 있는 휴리스틱 알고리즘을 개발하였다. 휴리스틱 알고리즘은 설비를 배치하기 위한 배치알고리즘과 최적해 탐색을 위한 시뮬레이티드 어닐링 알고리즘으로 구성된다.

2장에서는 기존연구를 고찰하고, 3장에서는 본 연구에서 다루고자 하는 설비배치문제에 대한 정의를 설명한다. 4장에서는 무게중심점을 이용하여 설비를 배치하는 배치 알고리즘과 해법 개발을 위한 시뮬레이티드 어닐링 알고리즘으로 구성된 휴리스틱 알고리즘에 대해 기술하고 5장에서는 기존의 해법들과의 실험비교를 통하여 제안하는 알고리즘의 효율성을 입증하고, 마지막으로 6장에는 결론 및 추후 연구방향에 대해서 기술한다.

## 2. 기존연구고찰

설비배치문제는 QAP(Quadratic Assignment Problem)로써 연구되었다. QAP에서 모든 설비들은 같은 모양 및 크기를 가지고 있고, 정해진 위치에 설비를 할당하는 문제로 정의된다. 그러나 실제 작업장에서 설비들은 보통 다른 모양 및 크기를 지니고 있으므로, QAP는 실제 작업장을 반영하지 못한다는 단점을 지니고 있다. 그러므로 다양한 크기의 설비를 배치하기 위한 연구가 진행되어왔다. 다양한 크기의 설비를 위한 배치문제는 입력점과 출력점의 위치가 주어진 문제와 주어지지 않는 문제로 분류된다.

입력점과 출력점의 위치가 정해지지 않는 설비배치문제에서 설비들 간의 물류는 설비의 중심점을 통해 흐른다고 가정하였다. 그러므로 이러한 문제에서 설비들 간의 거리는 설비의 중심점들 간의 거리로 측정된다. Heragu and Kusiak(1998)은 설비배치문제로 적용될 수 있는 기계배치문제(Machine Layout Problem, MLP)의 4가지 종류에 대해 기술하였고, 또한 이 문제를 해결하기 위한 2단계 알고리즘을 제안하였다. Chiang(2001)은 타부탐색(Tabu Search)을 이용한 휴리스틱 알고리즘을 제안하였다. 또한 Lee and Lee(2002)는 총 물류비용을 최소화하고 공간 활용도를 높이기 위한 혼성유전자 알고리즘(Hybrid Genetic Algorithm)이 적용된 접근법을 제안하였다. 또한 Solimanpur *et al.*(2005)는 개미알고리즘(Ant algorithm)을 제안하였다. 개미알고리즘에서, 각 개미는 한 개의 에이전트로 고려되며 문제를 해결하기 위하여 서로 통신한다.

입력점과 출력점의 위치가 정해진 문제에서 물류는 설비의 출력점으로부터 다른 설비의 입력점으로 흐른다고 가정하였으며, 그러므로 설비들 간의 거리는 설비의 입력점과 출력점들 간의 거리로 측정된다. Das(1993)는 혼합정수계획모형(Mixed Integer Programming Model)과 4단계로 구성된 휴리스틱 알고리

즘을 제안하였고 Welgama and Gibson(1993)은 두단계로 구성된 알고리즘을 제안하였다. Rajasekhran *et al.*(1998)은 Das(1993)의 혼합정수계획모형이 적용된 혼합정수계획모형과 유전자 알고리즘을 제안하였고, Das(1993)의 알고리즘과 성능을 비교하였으며, 제안하는 알고리즘의 효율성을 입증하였다. Kim and Kim(2000)은 혼합정수계획모형이 적용된 2단계로 구성된 휴리스틱 알고리즘을 제안하였다. 먼저 초기해를 생성하기 위한 단계와 부분최적화를 반복하는 단계로 구성되어 있으며, 기존 문헌과의 성능비교를 통해 효율성을 입증하였다.

## 3. 문제정의

본 연구는 입력점과 출력점의 위치가 고정된 직사각형 모양의 설비, 물류량, 작업장의 크기가 주어지면, 설비들 간의 최단직각거리와 물류량의 곱의 합을 최소화하는 설비배치를 찾는 문제를 다룬다. 설비의 입력점과 출력점은 설비의 면이나 내부에 위치하며 같은 위치에 있을 수 있다. 설비들 간의 물류는 출력점에서 입력점으로 흐른다고 가정하였으며, 따라서 거리는 각 설비의 입출력 점간의 최단직각거리로 계산된다. 또한 본 연구에서 설비는 직사각형 모양의 작업장에서 수평, 수직으로 배치된다고 가정하였다. 따라서 본 연구에서는 <그림 1>과 같이 설비는 주어진 설비모형(0°) 및 이를 중심으로 시계방향으로 90°, 180°, 270°로 회전한 형태의 설비배치를 고려하였다.

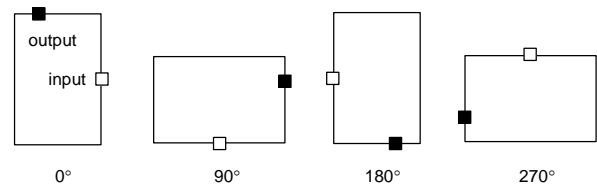


그림 1. 설비의 회전

다음과 같은 기호와 변수를 정의하였다.

- $N$  : 배치되는 설비의 수
- $W_{Max}^x$  : 2차원 공간에서  $x$ 축의 최대좌표
- $W_{Min}^x$  : 2차원 공간에서  $x$ 축의 최소좌표
- $W_{Max}^y$  : 2차원 공간에서  $y$ 축의 최대좌표
- $W_{Min}^y$  : 2차원 공간에서  $y$ 축의 최소좌표
- $m, n$  : 설비의 인덱스,  $m, n = 1, \dots, N$
- $f_m$  : 설비  $m$
- $s$  : 방위를 나타내는 인덱스,  $s = 1, 2, 3, 4$
- $q_{mn}$  : 설비  $m$ 에서  $n$ 으로 흐르는 물류량
- $d_{Hm}$  : 기본방위(0°)에서 설비  $m$ 의 중심점으로부터 좌, 우 변 까지 상대적 가로거리
- $d_{Vm}$  : 기본방위(0°)에서 설비  $m$ 의 중심점으로부터 위, 아래 변까지 상대적 세로거리

- $x_m, y_m$  : 설비  $m$ 의 중심점  $x, y$ 좌표
- $x_m^I, y_m^I$  : 설비  $m$ 의 입력점의  $x, y$  좌표
- $x_m^O, y_m^O$  : 설비  $m$ 의 출력점의  $x, y$ 좌표
- $h_m^I, v_m^I$  : 기본방위( $0^\circ$ )에서 설비  $m$ 의 중심점으로부터 입력점까지 상대적인 가로, 세로길이
- $h_m^O, v_m^O$  : 기본방위( $0^\circ$ )에서 설비  $m$ 의 중심점으로부터 출력점까지 상대적인 가로, 세로길이
- $x_{mn}^+ = \max\{0, x_m^O - x_n^I\}$
- $x_{mn}^- = \max\{0, x_n^I - x_m^I\}$
- $y_{mn}^+ = \max\{0, y_m^O - y_n^I\}$
- $y_{mn}^- = \max\{0, y_n^I - y_m^I\}$
- $z_{ms} = \begin{cases} 1, & \text{설비 } m \text{이 방위 } s \text{를 가질때,} \\ 0, & \text{그 이외의 경우} \end{cases}$

$W_{Max}^x, W_{Min}^x, W_{Max}^y, W_{Min}^y$ 는 2차원 공간에서  $x, y$ 좌표의 범위를 나타내기 위해 주어지며, 본문제에서  $W_{Max}^x, W_{Max}^y$ 는 충분히 큰 수로 설정하며,  $W_{Min}^x, W_{Min}^y$ 는 충분히 작은 수로 설정한다. 또한  $|W_{Max}^x - W_{Min}^x|, |W_{Max}^y - W_{Min}^y|$ 는 각각 작업장의 가로, 세로 길이로 표현될 수 있다. 직사각형의 설비형태 및 입력점과 출력점은 직사각형의 가로, 세로 및 설비의 중심점으로부터 상대 위치로 정의될 수 있다. 예를 들어, <그림 2>의 설비  $m$ 은 중심점으로부터 입력점까지의 상대좌표값( $h_m^I, v_m^I$ ), 출력점까지의 상대좌표값( $h_m^O, v_m^O$ )과 좌우변까지의 상대거리( $d_{Hm}$ ), 위아래변까지의 상대거리( $d_{Vm}$ )로 정의될 수 있으며, 따라서 설비  $m$ 은 식 (1)과 같이 가로, 세로 길이, 입력점과 출력점의 상대적인  $x, y$  좌표 값으로 표현된다.

$$f_m = \{(d_{Hm}, d_{Vm}), (h_m^I, v_m^I), (h_m^O, v_m^O)\} \quad (1)$$

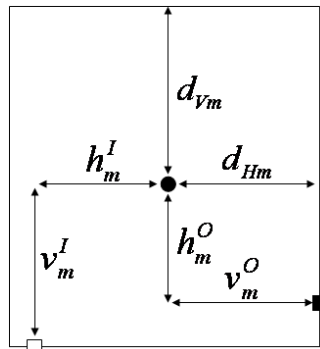


그림 2. 설비  $m$ 의 표현

설비 방위를 나타내는  $s$ 가 1인 경우는 기본형태( $0^\circ$ )를 나타내며, 2인 경우는 시계방향으로  $90^\circ$ 회전, 3인 경우는  $180^\circ$ 회전, 4인 경우는  $270^\circ$ 회전된 형태를 의미한다. 또한 <그림 1>에서

알 수 있듯이,  $s=1$ 일 때와  $s=3$ 일 때는 가로 및 세로길이가 같고,  $s=2$ 일 때와  $s=4$ 일 때가 가로 및 세로길이가 같다는 것을 알 수 있다. 설비의 방위  $s$ 에 따라 설비를 표현하는 상대 좌표 값은 다음과 같다.

$$s = 1 : \{(d_{Hm}, d_{Vm}), (h_m^I, v_m^I), (h_m^O, v_m^O)\} \quad (2)$$

$$s = 2 : \{(d_{Vm}, d_{Hm}), (h_m^I, -v_m^I), (h_m^O, -v_m^O)\} \quad (3)$$

$$s = 3 : \{(d_{Hm}, d_{Vm}), (-h_m^I, -v_m^I), (-h_m^O, -v_m^O)\} \quad (4)$$

$$s = 4 : \{(d_{Vm}, d_{Hm}), (-h_m^I, v_m^I), (-h_m^O, v_m^O)\} \quad (5)$$

따라서 설비의 방위  $s$ 에 따라 설비의 입력점과 출력점의 위치는 식 (6) ~ 식 (9)에 따라 결정된다.

$$x_m^I = x_m + h_m^I(z_{m1} - z_{m3}) + v_m^I(z_{m2} - z_{m4}), \forall m \quad (6)$$

$$y_m^I = x_m + h_m^I(z_{m4} - z_{m2}) + v_m^I(z_{m1} - z_{m3}), \forall m \quad (7)$$

$$x_m^O = x_m + h_m^O(z_{m1} - z_{m3}) + v_m^O(z_{m2} - z_{m4}), \forall m \quad (8)$$

$$y_m^O = x_m + h_m^O(z_{m4} - z_{m2}) + v_m^O(z_{m1} - z_{m3}), \forall m \quad (9)$$

본 연구의 목적함수는 4개의 방위를 가지는 설비들 간의 물류량과 입력점과 출력점들간의 직각거리의 곱의 합을 최소화하는 것으로써 식 (10)으로 표현된다. 또한 모든 설비는 한 개의 방위만을 가지며, <그림 3>과 같이  $x, y$  좌표로 표현되는 2차원 공간 안에서, 설비들 간의 겹침을 피해 배치된다.

$$\text{Minimize } \sum_{\substack{m, n = 1 \\ m \neq n}}^N q_{mn} (x_{mn}^+ + x_{mn}^- + y_{mn}^+ + y_{mn}^-) \quad (10)$$

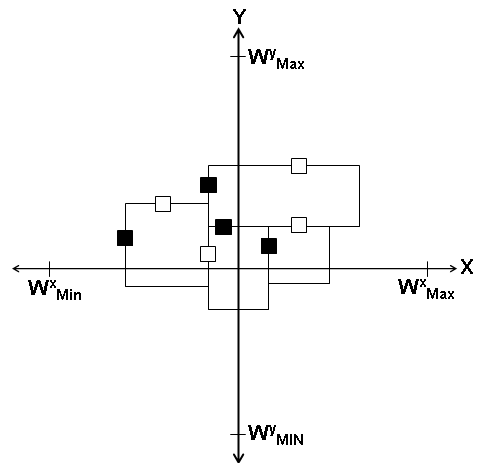


그림 3. 2차원 공간 내에서 설비배치

설비배치문제는 NP-hard 문제로써, 수리모델을 사용했을 경우, 작은 사이즈의 문제를 해결하는데 많은 시간이 소요된다. 4장에서는 유효한 시간 내에서 효율적으로 설비를 배치할 수 있는 알고리즘을 제안한다.

### 4. 휴리스틱 알고리즘

본 연구에서 제안하는 휴리스틱 알고리즘은 무게중심점을 이용한 배치 알고리즘과 시뮬레이티드 어닐링 알고리즘(Simulated Annealing(SA) Algorithm)으로 구성된다. 배치 알고리즘은 주어진 순서에 따라 설비를 순차적으로 배치하는 알고리즘으로써 유효한 설비배치 및 목적함수를 구하기 위해 사용된다. 그러므로 배치 알고리즘을 사용하여 해를 탐색하는 문제는 조합 최적문제로 정의될 수 있다. 따라서 본 연구에서는 보다 짧은 시간 안에 좋은 목적해를 탐색할 수 있는 시뮬레이티드 어닐링 알고리즘을 개발한다.

#### 4.1 배치 알고리즘

본 연구에서 제안하는 배치 알고리즘은 설비간의 중첩이 일어나지 않는 작업장 내 직사각형 모양의 지역(zone)에 설비를 배치하는 방법이다. 또한 효율적인 배치를 위해 무게중심점(Center of Gravity)과 후보결합점(Candidate Bonding Point)을 사용하였으며, 무게중심점은 4.1.2장, 후보결합점은 4.1.3장에서 설명한다.

설비배치 알고리즘을 위해 다음과 같은 기호와 변수가 사용된다.

- $i$  : 지역번호 인덱스
- $z_i$  : 지역(zone)  $i$
- $cx_n^I$  :  $f_n$ 이 배치될 때, 입력점에 대한 무게중심점  $x$ 좌표
- $cy_n^I$  :  $f_n$ 이 배치될 때, 입력점에 대한 무게중심점  $y$ 좌표
- $cx_n^O$  :  $f_n$ 이 배치될 때, 출력점에 대한 무게중심점  $x$ 좌표
- $cy_n^O$  :  $f_n$ 이 배치될 때, 출력점에 대한 무게중심점  $y$ 좌표
- $bx_i^I$  :  $z_i$ 에서 후보결합점의 입력점  $x$ 좌표
- $by_i^I$  :  $z_i$ 에서 후보결합점의 입력점  $y$ 좌표
- $bx_i^O$  :  $z_i$ 에서 후보결합점의 출력점  $x$ 좌표
- $by_i^O$  :  $z_i$ 에서 후보결합점의 출력점  $y$ 좌표
- $F_i^n$  :  $f_n$ 이  $z_i$ 에서 배치되었을 때, 목적함수값
- $F_{\min}^n$  :  $f_n$ 이 배치되었을 때, 최소 목적함수값
- $x_{\max}^i$  :  $z_i$ 의 최대  $x$ 좌표
- $x_{\min}^i$  :  $z_i$ 의 최소  $x$ 좌표
- $y_{\max}^i$  :  $z_i$ 의 최대  $y$ 좌표
- $y_{\min}^i$  :  $z_i$ 의 최소  $y$ 좌표
- $NZ$  :  $z_i$ 의 수
- $\alpha$  : 현재 배치된 설비의 수
- $s_i = \begin{cases} 1, & z_i \text{가 유효한 범위를 가질때,} \\ 0, & \text{그 이외의 경우.} \end{cases}$

본 연구에서 지역(zone)은 1개 이상의 이미 배치가 완료된 설

비의 변과 맞닿고 있으며, 다른 지역에 완전히 포함되지 않는 직사각형 모양의 지역으로 정의하며 지역  $i(z_i)$ 는 식 (11)과 같이 가로와 세로의 최소, 최대좌표로 표현된다.

$$z_i = \{x_{\min}^i, x_{\max}^i, y_{\min}^i, y_{\max}^i\} \tag{11}$$

위에서 정의한 지역에 따라, <그림 4>의  $x, y$  공간좌표에서 3개의 지역( $z_1 \sim z_3$ )이 도출될 수 있다.  $z_1$ 는  $f_2$ 의 윗변과  $f_1$ 의 좌변,  $z_2$ 는  $f_3$ 의 윗변,  $f_1$ 의 우변과  $f_4$ 의 좌변,  $z_3$ 은  $f_1$ 의 윗변과  $f_4$ 의 좌변과 맞닿아 있는 직사각형 모양의 지역으로 표현되어진다. 그러므로 <그림 4>에서 도출되는  $z_1 \sim z_3$ 는 각각  $z_1 = \{W_{Min}^x, X_1, Y_1, W_{Max}^y\}$ ,  $z_2 = \{X_2, X_3, Y_1, W_{Max}^y\}$ ,  $z_3 = \{W_{Min}^x, X_3, Y_2, W_{Max}^y\}$ 로써 그 범위가 표현된다.

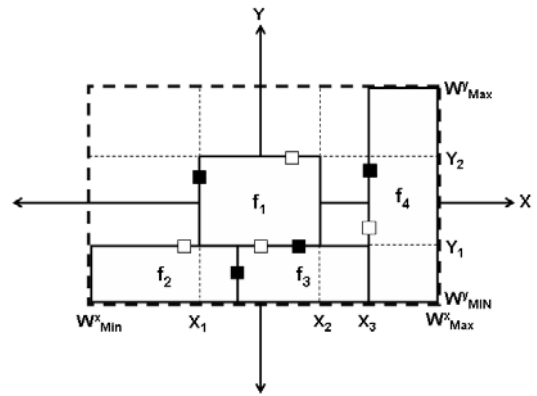


그림 4. 2차원 공간에서 지역의 표현

#### 4.1.1 초기배치

<그림 5>와 같이, 2차원 공간의 중심좌표(0, 0)에 초기에 배치되는 설비의 중심점을 고정시켜 초기 설비를 배치한다. 배치가 완료된 후 다음에 배치되는 설비를 위하여  $z_1 = \{W_{Min}^x, -d_{Hm}, W_{Min}^y, W_{Max}^y\}$ ,  $z_2 = \{W_{Min}^x, W_{Max}^x, d_{Vm}, W_{Max}^y\}$ ,  $z_3 = \{d_{Hm}, W_{Max}^x, W_{Min}^y, W_{Max}^y\}$ ,  $z_4 = \{W_{Min}^x, W_{Max}^x, W_{Min}^y, -d_{Vm}\}$ ,  $s_1 \sim s_4 = 1, NZ = 4$ 로 설정한다.

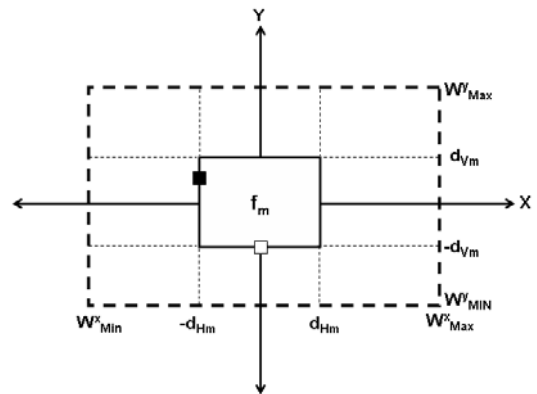


그림 5. 2차원 공간에서 초기배치

4.1.2 무게중심점 계산

본 연구에서 정의하는 무게중심점은 배치되는 설비와 이미 배치가 완료된 설비들 간의 물류량 및  $x, y$ 좌표를 고려한 중심점을 뜻하며 입력점과 출력점에 대한 중심점으로 구성되어 있다. 본 연구에서는 무게중심점을 기준으로 가까운 곳에 설비가 배치되었을 때, 좋은 목적해를 가지는 배치결과가 나올 것이라고 가정하였다. 무게중심점은 지역에서 설비를 배치하기 위한 후보결합점을 설정하기위해 사용되며, 아래의 식 (12)~식 (15)에 의해 계산된다(단,  $n = \alpha + 1$ ).

$$cx_n^I = \begin{cases} \frac{\sum_{m=1}^{\alpha} q_{nm}x_m^I}{\sum_{m=1}^{\alpha} q_{nm}}, & \sum_{m=1}^{\alpha} q_{nm} > 0 \\ \frac{\sum_{m=1}^{\alpha} x_m^I}{\alpha}, & \text{그 이외의 경우} \end{cases} \quad (12)$$

$$cy_n^I = \begin{cases} \frac{\sum_{m=1}^{\alpha} q_{nm}y_m^I}{\sum_{m=1}^{\alpha} q_{nm}}, & \sum_{m=1}^{\alpha} q_{nm} > 0 \\ \frac{\sum_{m=1}^{\alpha} y_m^I}{\alpha}, & \text{그 이외의 경우} \end{cases} \quad (13)$$

$$cx_n^O = \begin{cases} \frac{\sum_{m=1}^{\alpha} q_{nm}x_m^O}{\sum_{m=1}^{\alpha} q_{nm}}, & \sum_{m=1}^{\alpha} q_{nm} > 0 \\ \frac{\sum_{m=1}^{\alpha} x_m^O}{\alpha}, & \text{그 이외의 경우} \end{cases} \quad (14)$$

$$cy_n^O = \begin{cases} \frac{\sum_{m=1}^{\alpha} q_{nm}y_m^O}{\sum_{m=1}^{\alpha} q_{nm}}, & \sum_{m=1}^{\alpha} q_{nm} > 0 \\ \frac{\sum_{m=1}^{\alpha} y_m^O}{\alpha}, & \text{그 이외의 경우} \end{cases} \quad (15)$$

4.1.3 후보결합점 설정 및 설비배치

후보결합점은 지역에서 설비를 배치하기 위해 사용되며, 입력점과 출력점으로 구성된다. 입력점과 출력점에 대한 무게중심점으로부터  $z_i$ 까지의 가장 가까운 지점이 후보결합점의 입력점과 출력점으로 설정되며, 식 (16)~식 (19)를 사용하여 후보결합점의 입력점과 출력점의 위치를 계산할 수 있다. 또한 본 연구에서  $z_i$ 에서 배치되는 설비와 이미 배치가 완료된 설비들 간의 물류는  $z_i$ 에서의 후보결합점의 입력점과 출력점을 통해 흐른다고 가정하였다.

$$bx_i^I = \begin{cases} x_{\min}^i, cx_n^I < x_{\min}^i \\ cx_n^I, x_{\min}^i \leq cx_n^I \leq x_{\max}^i \\ x_{\max}^i, \text{그 이외의 경우} \end{cases} \quad (16)$$

$$by_i^I = \begin{cases} y_{\min}^i, cy_n^I < y_{\min}^i \\ cy_n^I, y_{\min}^i \leq cy_n^I \leq y_{\max}^i \\ y_{\max}^i, \text{그 이외의 경우} \end{cases} \quad (17)$$

$$bx_i^O = \begin{cases} x_{\min}^i, cx_n^O < x_{\min}^i \\ cx_n^O, x_{\min}^i \leq cx_n^O \leq x_{\max}^i \\ x_{\max}^i, \text{그 이외의 경우} \end{cases} \quad (18)$$

$$by_i^O = \begin{cases} y_{\min}^i, cy_n^O < y_{\min}^i \\ cy_n^O, y_{\min}^i \leq cy_n^O \leq y_{\max}^i \\ y_{\max}^i, \text{그 이외의 경우} \end{cases} \quad (19)$$

<그림 6>과 같이  $z_i$ 에서  $f_n$ 이 배치된다고 가정하였을 때, 목적함수  $f(x)$ 는 다음과 같은 식 (20)~식 (22)을 사용하여 표현할 수 있으며, 이때 목적함수  $f(x)$ 는  $F_n^i$ 으로 표현될 수 있다(단,  $n = \alpha + 1$ ).  $t(x)$ 는  $f_n$ 의 출력점과 후보결합점의 입력점간의 거리와 물류량을 고려한 함수이고,  $q(x)$ 는  $f_n$ 의 입력점과 후보결합점의 출력점간의 거리와 물류량을 고려한 함수이다.

$$f(x) = t(x) + q(x) \quad (20)$$

$$t(x) = \sum_{m=1}^{\alpha} q_{nm} (|x_n^O - bx_i^I| + |y_n^O - by_i^I|) \quad (21)$$

$$q(x) = \sum_{m=1}^{\alpha} q_{nm} (|x_n^I - bx_i^O| + |y_n^I - by_i^O|) \quad (22)$$

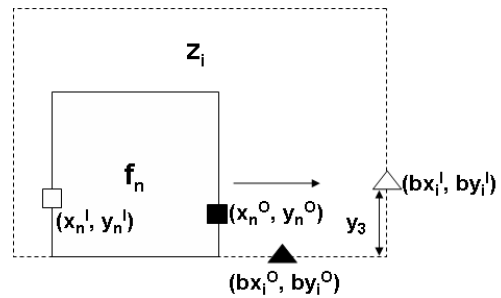


그림 6.  $z_i$ 에서  $f_n$ 의 수평이동

<그림 6>에서  $f_n$ 이  $z_i$ 에서 수평으로 움직인다고 가정하였을 때,  $t(x)$ 와  $q(x)$ 는 각각  $x_n^O = bx_i^I, x_n^I = bx_i^O$ 일 때, 최소값을 가진다. 또한  $f_n$ 으로부터 나가는 물류량의 합이  $f_n$ 으로 들어오는 물류량의 합보다 많다는 가정에서,  $x_n^O = bx_i^I$ 일 때,  $F_n^i$ 는 최소값을 가진다는 것을 알 수 있다. 또한  $f_n$ 이 세로로 움직인다고 가정하였을 때,  $t(x)$ 와  $q(x)$ 는 각각  $y_n^O = by_i^I, y_n^I = by_i^O$

일 때, 최소값을 가진다는 결과를 도출할 수 있다. 결론적으로, 주어진 가정에서  $f_n$ 으로 들어오는 물류량의 합이 나가는 물류량의 합보다 더 많을 때는 <그림 7>의 Case 1과 같이 후보결합점의 출력점과  $f_n$ 의 입력점을 붙이고, 그 반대일 경우는 <그림 7>의 Case 2와 같이 후보결합점의 입력점과  $f_n$ 의 출력점을 붙이며, 물류량의 합이 같을 때는 랜덤하게 Case 1과 Case 2 중 하나를 선택하여  $f_n$ 을 배치한다.  $f_n$ 이 배치될 때, 모든  $z_i$ 에서  $f_n$ 의 4방위를 고려하여 각각 배치하고, 가장 최소값을 가지는 위치를  $f_n$ 의 현재위치로 고정시킨다.

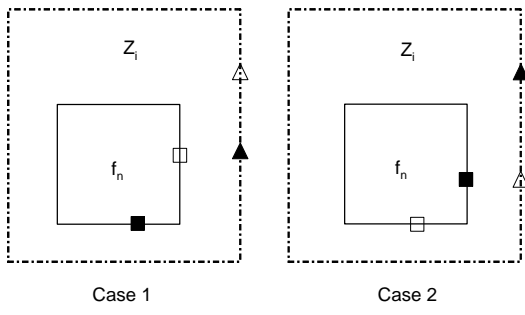


그림 7. 물류량에 따른  $z_i$ 에서  $f_n$ 의 배치

4.1.4 설비의 재배치

3.1.3에서 제안하는 방법을 사용하여 설비를 배치하였을 때, <그림 8>과 같이 설비가  $z_i$ 의 범위를 벗어나는 경우가 발생할 수 있다. 이러한 경우, 배치가 완료된 설비와의 겹침이 일어날 수 있으므로  $z_i$ 로부터 벗어난 거리  $x, y$ 만큼 설비를 재이동하여 설비가  $z_i$ 내에서 배치되도록 한다.

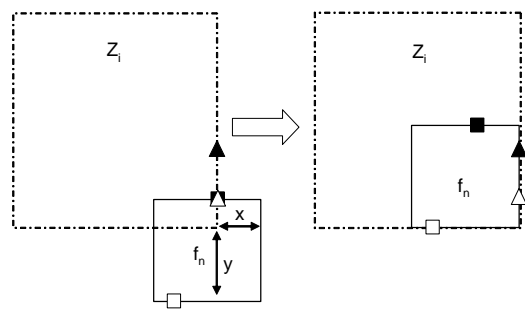


그림 8. 설비의 재배치

4.1.5 지역 업데이트 및 정렬

한 개의 설비가 배치된 후, 새로운 지역이나 소멸되는 지역이 도출될 수 있다. 따라서 다음 설비를 위한 지역의 업데이트가 필요하다. 설비가 지역에서 배치가 되었을 때, 발생할 수 있는 배치상태는 <그림 9>와 같다. Case 1은 설비가 지역을 완전히 덮은 상태이므로 유효한 지역이 없고, Case 3, 4, 5, 7은 유효한 지역이 1개, Case 2, 6, 9, 11, 13, 15은 유효한 지역이 2개, Case 8, 10, 12, 14은 유효한 지역이 3개가 도출되어질 수 있다.

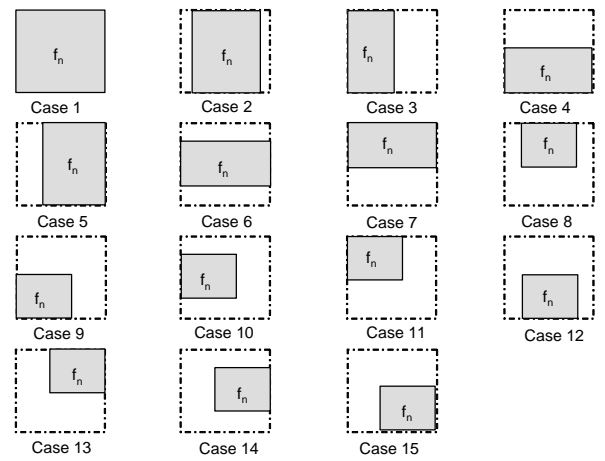


그림 9. 설비의 배치결과

본 연구에서는 설비가 배치된 지역에서 유효한 지역이 없을 때를 Case 1, 1개가 있을 때를 Case 2, 2개가 있을 때를 Case 3, 3개가 있을 때를 Case 4로 정의하였다. 지역은 다음의 지역 업데이트 규칙을 따라 업데이트 된다.

<지역 업데이트 규칙>

- Case 1. 설비가 배치된  $z_i$ 를 지운다( $s_i=0$ ).
- Case 2. 설비가 배치된  $z_i$ 의 범위정보( $x_{min}^i, x_{max}^i, y_{min}^i, y_{max}^i$ )를 수정한다.
- Case 3. 하나의 지역을  $z_i$ 로써 범위정보를 수정하고, 다른 하나의 지역은 새로운 지역으로 추가한다( $NZ=NZ+1$ ).
- Case 4. 하나의 지역을  $z_i$ 로써 범위정보를 수정하고, 다른 두 개의 지역은 새로운 지역으로 추가한다( $NZ=NZ+2$ ).

<그림 10>에서와 같이  $z_1$ 에서 배치된  $f_n$ 이  $z_2=\{X_2, X_4, Y_1, Y_4\}$ 로 표현되는  $z_2$ 의 범위에 영향을 주는 경우도 발생한다. 그러므로 배치가 완료된 설비에 의해 영향을 받는 모든 지역에 대한 업데이트가 필요하며, 위에서 기술한 지역 업데이트 규칙을 사용하여 지역을 업데이트 한다.

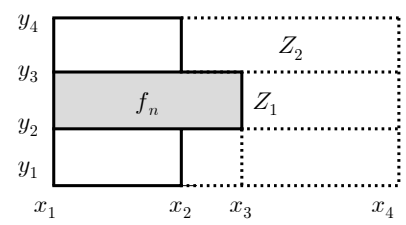


그림 10. 배치가 완료된 설비에 영향을 받은 지역

<그림 11>에서와 같이 배치되는  $f_n$ 에 의하여  $z_1=\{X_1, X_3, Y_2, Y_3\}$ 에서  $z_1=\{X_2, X_3, Y_2, Y_3\}$ 으로 수정된  $z_1$ 는  $z_2=\{X_2, X_3, Y_1, Y_4\}$ 로 표현되는  $z_2$ 의 범위에 완전히 포함된다. 본 연

구에서 지역은 다른 지역에 완전히 포함되지 않는 지역으로 정의하였으므로,  $z_1$ 을 유효하지 않는 상태로 수정한다.

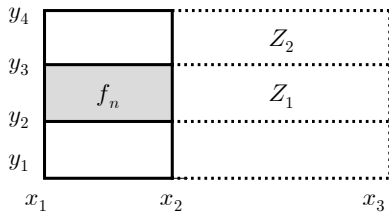


그림 11. 배치가 완료된 설비에 의해 범위가 수정된 지역

다음과 같은 규칙 사용하여 업데이트 된 지역을 정렬한다.

<정렬절차>

- Step 1.  $m, i$ 을 1로 설정한다.
- Step 2.  $z_m$ 을 선택한다.
- Step 3. 모든  $z_n$ 에 대해서  $z_m$ 이 완전히 포함되지 않는다면,  $z_i = \{x_{min}^m, x_{max}^m, y_{min}^m, y_{max}^m\}$ ,  $s_i = 1, i = i + 1$ 로 설정한다. 그렇지 않으면,  $m = m + 1$ 로 설정하고 Step 2으로 이동한다(단  $m < n \leq N, S_m, S_n = 1$ ).
- Step 4. 만약  $m = NZ$ 이면,  $NZ = i$ 로 설정하고 정렬을 마친다, 아니면  $m = m + 1$ 로 설정하고 Step 2로 이동한다.

4.1.6 전체절차

본 연구에서 제안하는 배치 알고리즘의 전체 절차는 다음과 같다.

<배치 알고리즘 전체절차>

- Step 1. 초기배치 및  $n=2$ 로 설정한다.
- Step 2. 지역을 업데이트, 정렬하고 무게 중심점을 구한다.  $i = 1, s = 1, F_{Min}^n = \infty$ 로 설정한다.
- Step 3.  $z_i$ 를 선택하고 후보결합점을 설정한다.
- Step 4. 만약  $s = 1$ 이면, 기본방위를 유지하고,  $s = 2$ 이면,  $f_n$ 를 시계방향으로  $90^\circ$  회전,  $s = 3$ 이면,  $180^\circ$  회전,  $s = 4$ 이면,  $270^\circ$  회전한다.
- Step 5.  $f_n$ 를  $z_i$ 에서 배치하고, 배치결과가 유효하면 Step 8로 간다. 유효하지 않으면 다음 단계로 간다.
- Step 6.  $f_n$ 를  $z_i$ 에서 재배치한다.
- Step 7. 배치결과가 유효하면 다음단계로 간다, 유효하지 않으면 Step 9로 간다.
- Step 8.  $F_i^n$ 을 구한다. 만약  $F_i^n < F_{Min}^n$ 이면,  $F_{Min}^n = F_i^n$ 로 설정한다.
- Step 9. 만약  $s = 4$ 이면, 다음단계로 이동한다. 아니면  $s = s + 1$ 로 설정하고 Step 4로 간다.
- Step 10. 만약  $i = NZ$ 이면,  $F_{Min}^n$  값을 가지는  $f_n$ 의 위치를 고정,

$n = n + 1$ 로 설정하고 다음단계로 간다. 아니면,  $i = i + 1$ 로 설정하고 Step 3으로 간다.

Step 11. 만약  $n = N + 1$ 이면, 마치고  $F_{Min}^n$ 를 최종목적 함수 값으로 도출한다. 아니면 Step 2로 간다.

4.2 시뮬레이티드 어닐링 알고리즘

시뮬레이티드 어닐링 알고리즘은 Kirkpatrick *et al.*(1983)에 의해 제안되었으며, 조합최적문제를 해결하는 하나의 해법이다. 시뮬레이티드 어닐링 알고리즘에서 이웃해가 현재해보다 좋으면 이웃해를 현재해로 수용하고, 이웃해가 현재해보다 좋지 않으면 수용확률과 랜덤변수(0~1)와 비교하여 수용확률이 랜덤변수보다 좋으면 현재해로 수용하고 아니면 다른 해를 탐색한다. 기본적인 시뮬레이티드 어닐링 알고리즘에서 이웃해 탐색이 랜덤으로 수행되며 확률적으로 해를 수용하기 때문에 항상 좋은 해를 찾는다고 보장할 수 없다.

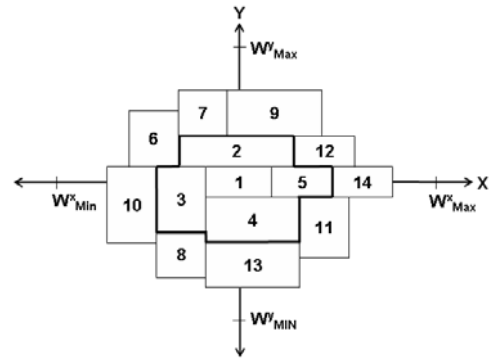


그림 12. 그룹으로 표현되는 설비

본 연구에서 제안하는 배치알고리즘을 사용하였을 때, 주어진 시퀀스의 앞쪽에 위치한 설비들은 중심쪽에, 바깥쪽에 위치한 설비들은 바깥쪽에 위치할 확률이 높다. 본 연구에서는 이러한 특성을 고려하고 해 탐색을 조절할 수 있는 시뮬레이티드 어닐링 알고리즘을 개발한다. 먼저 물류량이 크거나 설비들 간의 물류 연관성이 높은 설비들이 중앙에 배치되고, 물류량이 작거나 설비들 간의 물류 연관성이 낮은 설비들이 바깥쪽에 배치되었을 때 좋은 해를 가지는 배치결과가 도출되어질 수 있다고 가정하였다. 예를 들어, 1~14까지 순차적으로 있는 설비와 배치알고리즘을 사용하여 <그림 12>와 같은 결과를 도출하고 배치결과가 좋은 해를 가진다고 가정하였을 때, 설비 1, 2, 3, 4, 5은 물류량이 크거나 설비들 간의 물류 연관성이 높으며, 설비 6, 7, 8, 9, 10, 11, 12, 13, 14은 물류량이 작거나 설비 간의 물류 연관성이 낮다고 표현할 수 있다. 본 연구에서는 물류량이 많거나 물류연관성이 높은 설비부터 물류량이 작거나 물류연관성이 낮은 설비까지 그룹(G)형태로 표현하였다 ( $G_1 = \{1, 2, 3, 4, 5\}$ ,  $G_2 = \{6, 7, 8, 9, 10, 11, 12, 13, 14\}$ ).  $G_1$ 부터  $G_2$ 까지 갈수록 설비의 수는  $(i+1)^2$  형태와 유사하게 증가하는 것

을 알 수 있다. 그러므로 그룹은 주어진 시퀀스에서 앞쪽에서 뒤쪽으로  $(i+1)^2$  형태로 나누어 구분 지을 수 있으며, 본 연구에서는 주어진 시퀀스의 뒤쪽에서 앞쪽으로 그룹간의 설비 이동을 통한 점진적인 최소화를 시킬 수 있는 시뮬레이티드 어닐링 알고리즘을 개발하였다.

다음과 같은 기호와 변수를 정의하였다.

- $F_i, F_c$  : 초기해, 현재해
- $F_n, F_m$  : 이웃해, 최소해
- $T$  : 현재온도
- $e$  :  $e^2$ 은  $N$ 을 넘지 않는 최대수
- $k$  : 구간분할탐색과 구간 최소화의 반복횟수,  $k = 1, \dots, e-1$
- $T_k$  :  $k$ 번째 구간분할탐색을 시작할 때, 현재온도로 설정되는 온도
- $\beta$  : 쿨링 상수
- $\Delta = F_n - F_c$
- $x$  : 랜덤변수,  $0 \leq x \leq 1$
- $P$  : 수용확률
- $r$  : 해 탐색 반복횟수
- $i_k$  :  $k$ 번째 반복에서 구간분할탐색 반복횟수
- $o_k$  :  $k$ 번째 반복에서 구간최소화 반복횟수
- $l_1, l_2$  : 전체최소화 반복횟수
- $I, J$  : 각각 이웃해 탐색을 위해 교환되며, 선택구간과 확률에 의하여 랜덤하게 발생하는 변수
- $w$  :  $I$ 의 발생구간과 확률을 계산하고 구간을 표현하기 위한 변수,  $w=0, \dots, k-1$
- $b_{k,w}$  :  $k$ 번째 반복에서  $I$ 가 발생하는  $w$ 번째 구간
- $b_{j,k}$  :  $k$ 번째 반복에서  $J$ 가 발생하는 구간
- $P(b_{k,w})$  : 구간  $w$ 에서 발생하는  $I$ 의 선택확률
- $Max_r$  : 이웃해 탐색 총 반복횟수

모든  $k$ 에 대한  $i_k, o_k$ 와  $l_1, l_2$ 의 합은  $Max_r$ 과 같으며 다음의 절차를 따라  $i_k, o_k, l_1, l_2$ 를 재설정한다.

4.2.1 이웃해 탐색

본 연구에서 제안하는 이웃해 탐색방법은 구간분할탐색, 구간최소화, 전체 최소화로 구성된다. 구간분할탐색과 구간최소화의 반복을 통해 시퀀스의 바깥쪽에서부터 안쪽으로 최소화를 진행하고, 최종적으로 전체 최소화를 수행하는 순서로 구성된다. 먼저 구간분할탐색을 통해 시퀀스를 그룹형태로 구간을 분할하고 이웃해를 탐색하며, 그 다음 단계인 구간최소화에서는 구간분할탐색을 통해 분할된 구간만을 고려한 이웃해 탐색을 수행하여 분할된 구간을 최소화하는 것을 절차를 반복 수행 한다. 구간분할탐색과 구간최소화의 반복수행 후, 전체 최소화단계에서는 시퀀스를 반으로 나누어 이웃해를 탐색하

며, 현재해보다 더 좋은 이웃해만을 현재해로 수용한다.

구간분할탐색 단계에서  $b_{k,w}, b_{j,k}, P(b_{k,w})$ 는 식 (23)~식 (25)에 의해 계산될 수 있으며, 절차 또한 아래에 기술하였다. 예를 들어  $N=20$ 이면  $e=4$ 가 되며( $e^2 < N$ ), 식 (23)~식 (25)와 구간분할탐색 절차를 이용하여 <그림 13>과 같은 결과를 도출할 수 있다. 구간분할탐색에서  $I$ 의 선택확률을 사용한 이유는  $k$ 번째 구간탐색을 수행한 후, 다음 단계에서 분할된 구간  $b_{k,w}$ 만을 고려하여 이웃해를 탐색하는 구간최소화를 수행하게 되므로, 새롭게 생성된 구간에서 발생하는  $I$ 의 이동을 높여주기 위해 사용하였다. 또한 구간최소화가 진행된 구간에서 발생하는  $I$ 의 이동을 고려하지 않았을 때, 지역해에 머무를 수 있으므로 이동확률을 주었다.

$$\begin{cases} e^2 \leq b_{k,w} \leq N, & w = 0 \\ (e-w)^2 \leq b_{k,w} < (e-w+1)^2, & w \neq 0 \end{cases} \quad (23)$$

$$1 \leq b_{j,k} < (e-k)^2 \quad (24)$$

$$P(b_{k,w}) = \frac{w+1}{\sum_{w=0}^{k-1} (w+1)} \quad (25)$$

<구간분할탐색 절차>

- Step 1.  $k=1, w=0$ 으로 설정한다.
- Step 2.  $b_{k,w}, b_{j,k}, P(b_{k,w})$ 를 계산한다.
- Step 3. 만약  $w=k-1$ 이면  $r=i_k$ 가 될 때까지  $r$ 을 1씩 증가시키면서 이웃해를 탐색하고,  $r=i_k$ 이면 구간 최소화 단계로 간다.  $w=k-1$ 이 아니면  $w=w+1$ 로 설정하고 Step 2로 간다.

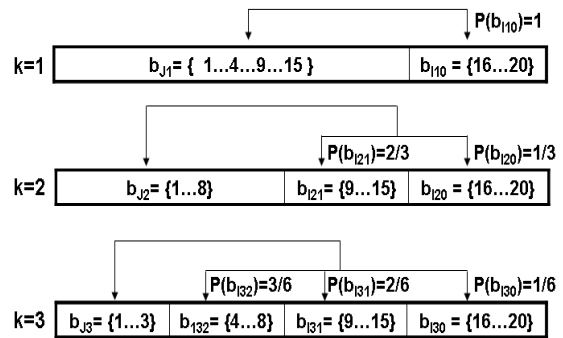


그림 13.  $N=20$  일 때, 구간분할탐색 결과

구간최소화는  $k$ 번째 구간분할탐색이 완료된 후 시작한다. 구간최소화는 구간분할탐색에서 식 (26)에 의해 분할된 전체 구간의 최소화를 수행하고 낮은 온도에서 수행되므로 현재해가 좋은 이웃해나 현재해와 근사한 이웃해로만 이동한다. 즉 구간최소화에서, 이웃해 탐색을 위한  $I, J$ 는 같은 구간에서 발생하며  $I$ 의 선택확률은 사용되지 않고, 분리된 구간을 나타내



는 변수  $w$  또한 사용하지 않는다(단,  $I \neq J$ ). 그러므로  $I, J$ 가 발생하는 구간은  $b_{I_k}, b_{J_k}$ 로 표현할 수 있으며 식 (26)에 의해 계산되며 절차 또한 아래에 기술하였다. <그림 14>는  $N=20$ 일 때, 식 (26)를 이용하여 도출되는 결과를 도식한 것이다.

$$\begin{cases} (e-k+1)^2 \leq b_{I_k} \leq N \\ (e-k+1)^2 \leq b_{J_k} \leq N \end{cases} \quad (26)$$

<구간최소화 절차>

- Step 1. 구간분할탐색에서 사용된 변수  $k$ 를 사용하여  $b_{I_k}, b_{J_k}$ 를 구한다.
- Step 2.  $r = o_k$ 가 될 때까지 이웃해를 탐색하며  $r$ 을 1씩 증가시킨다.  $r = o_k$ 이 되었을 때, 만약  $k = n-1$ 이면 전체 최소화 단계로 가고, 아니면  $k = k+1$ 로 설정하고, 구간분할탐색으로 간다.

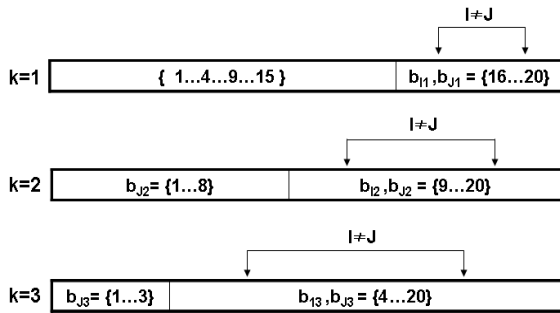


그림 14.  $N=20$  일 때, 구간최소화 결과

전체 최소화는 구간분할탐색과 구간최소화의 반복수행이 끝난 후 수행하며 전체 구간을 절반으로 나누어 두 부분을 최소화하는 두 단계로 구성된다.  $I, J$ 가 발생하는 구간은  $b_I, b_J$ 로 표현할 수 있으며 식 (27), 식 (28)에 의해 계산되고, 낮은 온도에서 수행된다(단,  $I \neq J$ ). 따라서 현재해가 좋은 이웃해나 현재해와 근사한 이웃해로만 이동하게 된다. 또한  $r = l_1$ 이 될 때까지 식 (27)을 통해 계산된 구간만을 고려한 이웃해 탐색을 반복 수행하고,  $l_1 < r < l_2$ 일 때는 식 (28)을 통해 계산된 구간만을 고려한 이웃해 탐색을 반복 수행한다.

$$\begin{cases} (N/2) \leq b_I \leq N \\ (N/2) \leq b_J \leq N \end{cases} \quad (27)$$

$$\begin{cases} 1 \leq b_I < N/2 \\ 1 \leq b_J < N/2 \end{cases} \quad (28)$$

4.2.2 초기화

초기화에서는 변수  $e$ 를 계산한다( $e^2 < N$ ). 또한  $r = 0, k = 1, T = T_k$ 로 설정하고, 초기시퀀스(1, 2, ..., N)를 이용하여  $F_i$ 를 구하고,  $F_c, F_n, F_m = F_i$ 로 설정한다.

4.2.3 해의 수용

시뮬레이티드 어닐링 알고리즘에서 이웃해가 현재해보다 좋거나 같을 때는 이웃해를 현재해로 설정한다. 또한 이웃해가 현재해보다 좋지 않을 때, 확률적인 방법으로 해의 수용을 결정한다. 즉 수용확률  $P$ 가 랜덤으로 발생하는 변수  $x$ 보다 클 때는 이웃해를 현재해로 받고, 아닐 때는 다른 이웃해를 탐색한다. 본 연구에서는  $P$ 의 균형( $0 \leq P \leq 1$ )을 유지하기 위해 로그함수를 사용하였으며 식 (29)와 같다.

$$P = e^{(-\log(\Delta)/T)} \quad (29)$$

4.2.4 온도업데이트

시뮬레이티드 어닐링 알고리즘에서 온도  $T$ 는 해의 이동을 조절하는 중요한 실험변수로 작용한다. 온도가 높을 때는, 해의 이동이 자유스럽지만, 온도가 낮아짐에 따라, 어느 일정한 값으로 수렴하게 된다. 온도 업데이트는 설정된 온도를 반복 수행을 진행하면서 온도를 낮추는 것을 의미한다. 냉각속도가 느리면 전체 알고리즘의 효율성을 저하시키고, 냉각속도가 빠르면 해의 이동이 지역해에 국한될 수 있다는 단점이 있다. 본 연구에서는 실험을 통하여 적절한 파라미터를 도출하였으며,  $k$ 번째 구간분할탐색을 시작할 때,  $T_k$ 를  $T$ 로 설정하여 해의 이동확률을 높여주었고,  $k$ 가 증가함에 따라  $T_k$ 를 점차 낮게 설정하였다.  $T$ 는 아래의 식 (30)에 의해 계산된다.

$$\begin{cases} T = T_k, & r = o_k \\ T = TB^{r-1}, & r \neq o_k \end{cases} \quad (30)$$

4.2.5 시뮬레이티드 어닐링 알고리즘 절차

본 연구에서 제안하는 시뮬레이티드 어닐링 알고리즘의 전체 절차는 다음과 같다.

<시뮬레이티드 어닐링 알고리즘 절차>

- Step 1. 초기화:
- Step 2. 이웃해 탐색:
  - 만약  $0 \leq r < i_k$ 이거나  $o_{k-1} \leq r < i_k$ 이면, 2.1로 간다; 만약  $i_k \leq r < o_k$ 이면, 2.2로 간다; 만약  $o_{k-1} \leq r < l_1$ 이면, 2.3로 간다; 만약  $l_1 \leq r < l_2$ 이면, 2.4로 간다; 만약  $r = l_2$ 이면, 알고리즘을 마친다.
  - 2.1 식 (23)~식 (25)를 사용하여  $F_n$ 를 구한다.
  - 2.2 식 (26)를 사용하여  $F_n$ 를 구한다.
  - 2.3 식 (27)을 사용하여  $F_n$ 를 구한다.
  - 2.4 식 (28)를 사용하여  $F_n$ 를 구한다.
- Step 3. 현재해와 이웃해의 비교:
  - 만약  $\Delta \leq 0$ 이면,  $F_c = F_n$ 으로 설정 하고 4.1로 간다; 만약  $\Delta > 0$ 이고,  $P > x$ 이면,  $F_c = F_n$ 으로 설정하고 4.1로 간다; 그 외에는 4.2로 간다.

Step 4. 업데이트:

- 4.1  $F_n < F_m$ 이면,  $F_m = F_n$ 으로 설정하고  $T = T * \beta$ ,  $r = r + 1$ 로 설정한다. 그리고 만약  $r = o_k$ 이면,  $T = T_k$ 로 설정 하고 Step 2로 간다.
- 4.2  $r = r + 1$ 로 설정한다. 만약  $r = o_k$ 이면,  $T = T_k$ 로 설정하고 Step 2로 간다.

5. 수치실험 및 분석

제안하는 휴리스틱 알고리즘은 C++로 구현되었으며, AMD 2GHz, 1G Ram를 가진 PC에서 실험하였고, 실험변수는 <표 1>과 같다.

표 1. 실험변수

Size	Parameter
10,12	$Max_r = 5000, i_k = 1000 \sim 1200, o_k = 500 \sim 700, l_1 = 750, l_2 = 750$
14,16	$Max_r = 6000, i_k = 1200 \sim 1400, o_k = 500 \sim 700, l_1 = 1000, l_2 = 1000$
18	$Max_r = 7000, i_k = 1200 \sim 1500, o_k = 700 \sim 900, l_1 = 1200, l_2 = 1200$
20	$Max_r = 8000, i_k = 1200 \sim 1500, o_k = 500 \sim 900, l_1 = 1500, l_2 = 1500$
30	$Max_r = 12000, i_k = 1200 \sim 1800, o_k = 500 \sim 1200, l_1 = 2000, l_2 = 2000$

본 연구에서는 제안하는 알고리즘의 성능평가를 위하여 비교 알고리즘에서 사용하였던 데이터를 사용하였으며 입력점과 출력점의 위치가 주어진 문제와 주어지지 않는 문제(설비의 중심점들 간의 직각거리를 고려한 문제)를 사용하여 성능평가를 하였다. <표 2>에서 실험예제 1, 2에 대한 실험결과는 Das(1993)의 실험데이터를 사용한 결과이며, 실험예제 3, 4, 5은

Rajasekharan *et al.*(1998)의 실험예제를 사용한 결과이다. 또한 실험예제 6, 7에 대한 실험결과는 Deb and Bhat(2005)의 실험예제를 사용한 결과이며, 실험예제 8, 9에 대한 실험결과는 Heragu and Kusiak(1988)의 설비 데이터와 Nugent *et al.*(1968)의 물류량 데이터를 사용한 결과이다. 제안하는 알고리즘은 총 5번의 반복실험을 하였으며, 평균값, 최대값, 최소값을 도출하여 제안하는 알고리즘의 성능평가를 하였다. 또한 실험예제 1, 2, 3, 4, 5의 결과는 Kim and Kim(2000)의 결과이고, 실험예제 6, 7의 결과 또한 출처와 같고, 실험예제 8, 9의 결과는 Solimanpur *et al.*(2005)의 결과이다. 실험예제 1~5에서는 설비의 입출력 점의 위치가 정의되어 있으며, 실험예제 6~9에서는 설비 간의 거리를 중심점간의 직각거리를 사용하였다.

제안하는 알고리즘의 평균값과 가장 좋은값은 기존연구보다 더 좋은 해를 지녔으며, 5번의 실험결과 중 가장 안 좋은 결과도 실험예제 4, 5번을 제외한 나머지 실험예제에서도 기존의 연구보다 더 좋은 해를 도출하였다. <그림 15>, <그림 16>은 설비 수에 따라 제안하는 알고리즘의 평균값 및 가장 좋은값과 기존 연구에 의해 도출된 실험결과값을 비교하였을 때 향상된 비율을 도식한 것이다. <그림 15>에서 알 수 있듯이, 입력점과 출력점이 주어진 문제에서 제안하는 알고리즘은 설비 10개의 문제서부터 설비 14개까지는 향상율이 2.48%에서 0.36%로 감소하지만 설비 18개의 문제에서는 향상율이 7.29%로 크게 증가하는 것을 알 수 있다. 또한 <그림 16>에서 알 수 있듯이, 입력점과 출력점이 설비의 중앙에 배치된 문제에서, 제안하는 알고리즘은 문제의 크기가 증가함에 따라 향상율 또한 증가하는 것을 알 수 있다.

또한 제안하는 시뮬레이티드 어닐링 알고리즘의 성능을 평가하기 위하여 기본 시뮬레이티드 어닐링 알고리즘과 실험비교를 하였다. 기본 시뮬레이티드 어닐링 알고리즘에 관한 내용은 시뮬레이티드 어닐링 알고리즘에 관한 문헌을 참조하길 바란다. 최대반복수와 쿨링상수는 같은 조건으로 두었으며, 초기온도  $T=10 \sim 45$ 로 설정하였으며 제안하는 알고리즘의 실험과 마찬가지로 5번의 반복실험을 하였다.

표 2. 기존연구와의 실험결과 비교

Compared algorithm	Exam.	Size	Compared objective value	Proposed algorithm			
				Average value	Best value	Worst value	Computing times(Sec)
Kim and Kim(2000)	1	10	15629.3	15240.62	15175.5	15354.2	2.39
	2	12	36676.5	36171.64	35656.1	36674.3	3.78
	3	14	41691.3	41540.2	41080.5	41834	6.66
	4	16	55064.1	54843.86	54411.4	55159.5	10.5
	5	18	66489.2	61638.54	60938.9	62317.2	12.43
Deb and Bhat(2005)	6	12	21318	20482.8	20435.5	20518.5	3.48
	7	18	62382	57207.58	57024.4	57467	11.42
Solimanpur <i>et al.</i> (2005)	8	20	119.71	103.8892	103.093	104.514	17.39
	9	30	334.8698	232.5714	228.913	235.888	86.18

표 3. 기본 SA알고리즘과 실험결과 비교

Compared algorithm	Exam.	Basic SA algorithm			Proposed algorithm		
		Average value	Best value	Worst value	Average value	Best value	Worst value
Kim and Kim(2000)	1	15759.1	15361.8	16135.6	15240.62	15175.5	15354.2
	2	36910.6	36420.2	37202.2	36171.64	35656.1	36674.3
	3	42067.48	41375.7	42532.7	41540.2	41080.5	41834
	4	55615.14	55258.7	55064.1	54843.86	54411.4	55159.5
	5	62225.48	61236.4	62992.1	61638.54	60938.9	62317.2
Deb and Bhat.(2005)	6	20763.42	20518.5	21126.7	20482.8	20435.5	20518.5
	7	58023.32	57622	58539.7	57207.58	57024.4	57467
Solimanpur <i>et al.</i> (2005)	8	105.5874	104.089	106.854	103.8892	103.093	104.514
	9	236.2666	233.562	238.166	232.5714	228.913	235.888

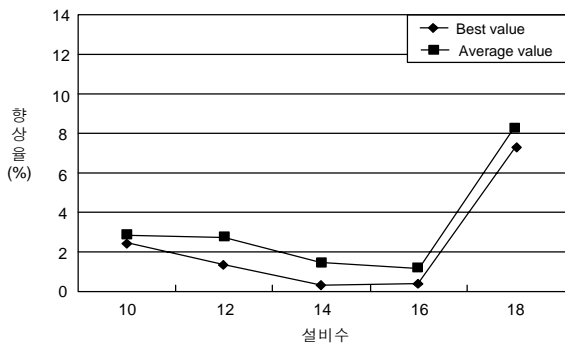


그림 15. Kim and Kim(2000) 알고리즘과 비교하였을 때 제안하는 알고리즘의 향상율

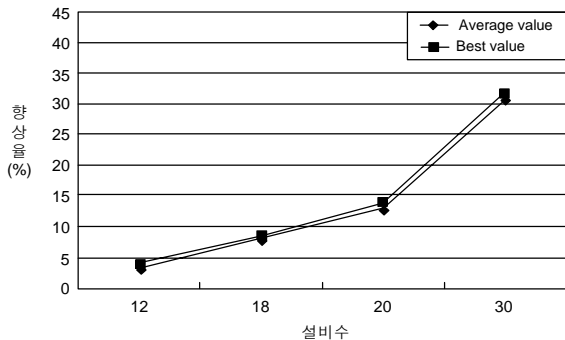


그림 16. Deb and Bhat(2005)과 Solimanpur *et al.*(2005)와 비교하였을 때 제안하는 알고리즘의 향상율

<표 3>에서 알 수 있듯이, 제안하는 알고리즘의 평균값과 가장 좋은해는 이웃해를 랜덤하게 찾는 기본 시뮬레이티드 어닐링 알고리즘에 비하여 모든 실험예제에서 더 좋은 해를 도출하였으며, 제안하는 알고리즘의 가장 안 좋은값도 실험예제 4번을 제외한 모든 실험예제에서 더 좋은값을 도출하였다.

그러므로 제안하는 알고리즘은 입력점과 출력점이 주어지지 않는 문제에서 모두 조사된 기존 해법에 비해 더 좋은 해를 도출하는 것으로 분석되었다. <그림 17>은 제안하는

알고리즘의 수행시간이다. 대부분의 실험에서 제안하는 알고리즘의 수행시간은 조사된 문헌에 기술된 알고리즘의 수행시간에 비하여 빠른 시간 내에 해를 도출하였으나, 실험환경이 다르므로 수행시간 비교는 생략하였다.

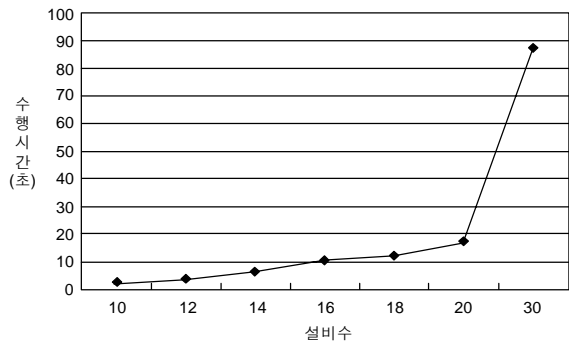


그림 17. 제안하는 알고리즘의 수행시간

## 6. 결론 및 추후연구

본 연구는 입출력점이 고정된 설비배치를 위한 수리모델과 유효한 시간 안에 효율적인 설비배치를 제공할 수 있는 휴리스틱 알고리즘을 제안하였다. 제안하는 알고리즘의 실험결과와 설비 30개 이하의 문제에서 90초 이내에 기존 해법들에 비해 0.36~30.54%의 개선효과가 있는 설비배치결과를 도출하는 것으로 분석되었다.

본 연구에서 제안하는 휴리스틱 알고리즘은 설비의 수가 많고, 빠른 시간 내에 생산라인을 구성해야 하는 유연 제조 시스템이나 재구성 가능 제조시스템에서 사용될 수 있을 것이다.

실제 공장에서의 물류는 AGV와 같은 운송수단에 의해 운송되며, 공장 내 트랙을 따라 움직이므로 직각거리를 사용한 평가는 한계가 있다. 추후 설비를 배치할 때, 흐름거리를 고려한 개발이 연구과제가 될 수 있다.

## 참고문헌

- Tompkins, J. A., Bozer, Y. A., Frazelle, E. H., Tanchoco, J. M. A., and Trevino, J. (1996), *Facilities planning*, New York: Wiley, 137-285.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983), Optimization by simulated annealing, *Science*, 220, 671-680.
- Kim, J. G. and Kim, Y. D. (2000), Layout planning for facilities with fixed shapes and input and output points, *International Journal of Production Research*, 38(18), 4635-4653.
- Das, S. K. (1993), A facility layout method for flexible manufacturing systems, *International Journal of Production Research*, 31, 279-297.
- Rajasekharan, M., Peters, B. A. and Yang, T. (1998), A genetic algorithm for facility layout design in flexible manufacturing systems, *International Journal of Production Research*, 36(1), 95-110.
- Deb, S. K. and Bhattacharyya, B. (2005), Solution of facility layout problems with pickup/drop-off locations using random search techniques, *International Journal of Production Research*, 43(22), 4787-4812.
- Welgama, P. S. and Gibson, P. R. (1993), A construction algorithm for the machine layout problem with fixed pick-up and drop-off points, *International Journal of Production Research*, 31(11), 2575-2590.
- Solimanpur, M., Vrat Prem, and Shankar Ravi (2005), An ant algorithm for the single row layout problem in flexible manufacturing systems, *Computers & Operations Research*, 32, 583-598.
- Montreuil, B. (1990), A modeling framework for integrating layout design and flow network design, *In proceeding of Material Handling Research Colloquium*, Hebron, Kentucky, USA, 43-58.
- Heragu, S. S. and Kusiak, A. (1988), A Machine layout problem in flexible manufacturing systems, *Operations Research*, 36(2), 258-268.
- Heragu, S. S. and Kusiak, A. (1990), Machine layout: an optimization and knowledge-based approach, *International Journal of Production Research*, 28, 615-63.
- Chiang, W. C. (2001), Visual facility layout design system, *International Journal of Production Research*, 39(9), 1811-1836.
- Nugent, C. E., Vollman, T. E., and Ruml, J. (1968), An experimental comparison of techniques for the assignment of facilities to locations, *Operations Research*, 16, 150-173.
- Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritchow, G., Ulsoym G., and Brussel, H. V. (1999), Reconfigurable Manufacturing Systems, *Annals of the CIRP*, 48(2), 527-540.
- Kim, Y. G., Yoon, B. S., and Lee, S. B. (1997), *Meta Heuristics*, Yeongji Moonhwasa, Seoul, Korea.



## 정동화

한국외국어대학교 산업시스템공학과 학사  
 현재: 고려대학교 정보경영공학과 석박사통  
 합과정  
 관심분야: 제조 및 물류시스템설계,  
 Meta-heuristics



## 서윤호

고려대학교 산업공학과 학사  
 Pennsylvania State University 산업공학 석사,  
 박사  
 현재: 고려대학교 정보경영공학과 교수  
 관심분야: 제조, 조립 및 물류시스템의 VR  
 적용지능설계