

# 블록단위 설비배치를 위한 유전자 알고리즘의 적용

우성식 · 박양병\*

Applying a Genetic Algorithm to a Block Facility Layout

Sung-Sik Woo · Yang-Byung Park\*

## ABSTRACT

The most research on facility layout problems ignored the actual shapes of activity spaces and the aisles between activities. In many cases, the research also ignored the actual shape of building where the activities are to be arranged.

In this paper, we present a block based layout technique that applies a genetic algorithm to search for a very good facility layout with horizontal aisles. From the extensive experiments for two different cases with respect to the shape of activity space, it was found that the proposed method generated better layouts than the ones obtained by applying Tam's algorithm in all test problems. The proposed algorithm showed about 10% improvement of performance on the average. We determined the best combination of the reproduction rule and the genetic operators with their probabilities for each test problem through the experiments.

## 1. 서 론

설비배치 문제는 배치대상이 되는 활동(부서)들에 대한 여러 정성적요인을 고려하면서 활동간 물자흐름량과 거리의 곱의 총합을 최소화하는 활동들의 배열을 결정하는 문제이다. 최적의 설비배치를 결정하는 문제는 활동의 수가 증가할수록 계산량이 지수적으로 증가하는 NP-complete 문제에 속한다. 설비배치 문제는 제조설비배치, 사무실배치, VLSI 설계 등의 다양한 형태[4]로 나

타나고 있다.

그동안 설비배치 문제를 해결하기 위해 많은 연구가 진행되어 왔다. Muther[6]는 정량적 및 정성적요인을 함께 고려하면서 단계별 과정을 거쳐 배치대안을 생성해 주는 SLP기법을 개발하였다. 그러나 이 기법은 실용성이 매우 높은 것으로 인식되고 있지만, 주관적인 판단에 근거하여 수작업으로 관련도를 작성하는 단계에서 활동의 수가 증가하면 실행이 거의 불가능한 치명적인 단점을 가지고 있다. 그후, 컴퓨터를 이용한 배치알고리

\* 경희대학교 산업공학과

들[11, 12]이 많이 소개되었는데, 대표적으로 CRAFT, COFAD, ALDEP 등이 있다. 이들 알고리즘의 적용에 의해 생성된 활동공간들은 때때로 실제 사용할 수 없는 요철모양의 불규칙한 형상으로 나타나기 때문에 최종배치안을 얻더라도 수작업에 의한 상당한 재조정이 필요하다. 설비배치 문제의 또 다른 접근방법으로서 총이동거리와 같은 정량적요인만을 고려하면서 탐색기법을 이용하는 배치알고리즘의 연구가 있다. 탐색기법으로는 주로 genetic algorithm, tabu search, simulated annealing기법을 사용하였다. 대표적인 연구결과로 Tam[9, 10], Glover[2], Chan[1] 등의 논문이 있다. 이들 탐색기법을 이용하여 개발된 알고리즘도 생성된 배치안의 형상이 불규칙하여 재조정이 필요한 문제점을 내포하고 있다.

특히, 유전자 알고리즘(genetic algorithm)을 이용한 배치결과가 다른 탐색기법에 의한 것보다 더 뛰어난 것으로 입증된[10] 이후, 유전자 알고리즘을 이용한 배치기법의 연구가 활발히 진행되고 있다. Tam[10]은 각 활동공간들의 배치방향을 문자로써 표현한 배치해(string)를 생성해 내는 유전자 알고리즘을 개발하였고 Suresh[8]는 이차배정문제(QAP)를 해결하기 위한 처음으로 유전자 알고리즘을 소개하였다. 그리고 Chan[1]은 이차배정문제를 해결하기 위해 교차변이 연산자들의 성능을 비교실험하였다.

본 논문에서는 유전자 알고리즘을 이용하여 활동공간 총이동거리를 최소화하는 최선의 해를 탐색하는 블럭단위의 설비배치 기법을 제안한다. 제안된 방법에서는 블럭단위의 배치로 인해 직사각형 전체영역의 형상을 원하는 대로 유지할 수 있게 되며, 동시에 가로(수평)방향의 통로를 자동적으로 생성할 수 있게 된다. 배치할 직사각형 활동공간의 형태를 가로와 세로의 크기에 따라 두가지 경우로 나누어 다양한 실험을 수행한 결과, 제안

된 배치기법은 모든 실험문제에서 뛰어난 성능을 보였다. 유전자 알고리즘의 적용을 위한 유전연산자의 종류와 발생확률 그리고 재생규칙은 각 실험문제에서 모든 가능한 변수조합에 대한 계산실험을 통하여 결정하였다.

## 2. 블럭단위의 배치

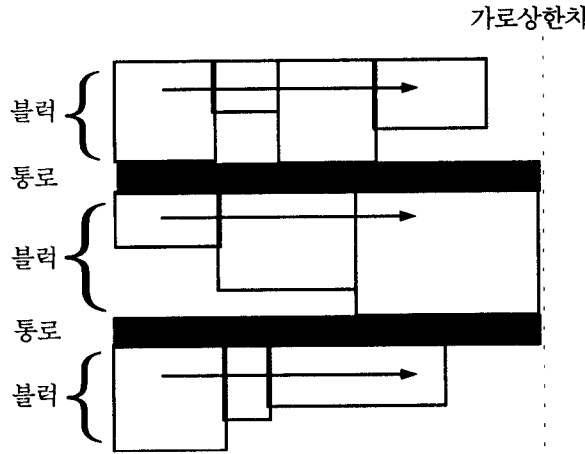
각 활동공간은 서로 다른 면적의 직사각형으로 가정하며, 배치할 전체영역의 가로길이의 상한치는 주어진다. 활동간의 거리는 각 공간 중심점 사이의 직각거리로 가정하며, 각 활동간의 흐름량은 사전에 주어진다.

배치는 전체영역의 맨 좌측에서부터 우측방향으로 한개 활동씩 순서대로 이루어진다. 이때 생성된 활동배치열을 블럭이라 정의한다. 만일 블럭의 길이가 가로상한치를 초과하게 되면, 해당블럭에 속한 활동공간들의 가장 긴 세로길이만큼 아래로 이동하여 전체영역의 맨 좌측에서부터 배치가 계속되면서 새로운 블럭이 구축된다. 그림 1에서와 같이 블럭과 블럭사이에는 가로(수평)방향의 통로가 생성된다.

배치대상이 되는 활동공간의 형태는 가로와 세로의 길이가 주어진 경우와 면적과 세로/가로 비율의 상한치, 그리고 블럭의 폭이 주어진 경우 2가지로 나누었다. 2가지 경우에 대한 블럭단위의 배치과정은 다음과 같다.

- (i) 활동공간의 가로와 세로의 길이가 주어진 경우

블럭의 폭은 블럭내 활동공간들의 가장 긴 세로의 길이로써 결정되기 때문에 일정하지 않다. 따라서 각 블럭을 형성할 때 통로와 활동공간사이의 빈공간(slack)을 가능한 한 줄이는 방향으로



〈그림 1〉 블록단위의 배치와 통로의 생성

배치가 이루어진다. 빈공간의 존재를 인정하므로 배치결과의 재조정이 필요없다.

(ii) 활동공간의 면적과 세로/가로 비율의 상한치 그리고 블록의 폭이 주어진 경우

주어진 블록폭을 활동공간의 세로길이를 하여 가로길이를 계산한다. 만일 가로/세로 비율의 상한치를 초과하면, 세로/가로 비율의 상한치에 맞게 가로길이를 계산하고 면적에 따라 세로길이를 새로이 계산한다. 이때 세로길이는 블록폭보다 작아지지만 블록의 폭은 주어진 크기로 일정하게 유지한다. 따라서 각 블록에는 빈공간이 생길 수 있다. 만약 세로/가로 비율의 상한치가 주어지지 않는다면 블록의 폭이 모든 활동공간의 세로길이가 되므로 빈공간은 생기지 않는다. 하지만 구해진 활동공간의 형태는 실제 사용할 수 없는 비실용적인 모양일 수 있다.

위에서 설명한 두 가지 경우에 대해 배치를 수행하기 위해서는 반드시 배치설계자가 배치할 전체영역의 가로길이를 사전에 알아야 한다. 만일 전체영역의 폭을 조절할 수 있다면, 반복적인 실험을 통해 최적의 전체영역 형태를 결정할 수 있다.

### 3. 유전자 알고리즘의 적용

유전자 알고리즘은 적자생존의 원리를 탐색기법에 적용한 것으로 보다 더 나은 해를 찾는 데 매우 뛰어난 성능을 보이고 있다[3]. 유전자 알고리즘은 해의 표현과 각 연산자의 적용방법에 따라 그 성능이 크게 다르게 나타날 수 있다.

#### (1) 배치해의 표현

배치해는 활동들의 블록단위 배치순서를 나타내는 활동번호 string으로 표현된다. 따라서 실제 배치는 구해진 string을 토대로 블록단위로 이루어진다. 배치해를 십진수의 string으로 표현하기 때문에 배치순서의 이해와 탐색능력이 뛰어난 변형된 교차변이 연산자의 적용이 용이하게 된다.

#### (2) 교차변이(crossover) 연산자

교차변이 연산자는 유전자 연산자들 중에서 해의 탐색범위를 가장 크게 넓히는 연산자이다. 두 개의 string을 교배하면 새로운 string내에서 활동번호가 중복될 수 있다. 따라서 새로운 배치해에

중복된 활동번호가 나타나지 않도록 하는 변형된 교차변이 연산자가 필요하다. 교차변이 연산자로는 traveling salesman problem에서 성능이 뛰어난 것으로 알려진 PMX(Partially Matched Crossover), OX(Order Crossover), CX(Cycle Crossover) 방법들[3]을 선택하였다. 그리고 string내 임의의 두 지점사이의 유전자(십진수)들을 서로 교환하는 two-cut crossover 방법을 적용하였다.

### (3) 돌연변이(mutation) 연산자

현재의 해가 비록 높은 적합도를 보이더라도 탐색의 범위를 가능한 한 더 넓히는 것이 필요하다. 유전자 알고리즘을 적용할 때 빠른 수렴(premature)을 막기 위한 보험정책으로 돌연변이 연산자를 적용한다. 돌연변이 연산자로는 Inversion, Insertion, Swap기법[5]들을 선택하였다.

### (4) 재생규칙(reproduction rule)

유전자 알고리즘을 적용할 때 다음 세대에 더 좋은 유전자가 복사되도록 하기 위해서 재생이 필요하다. 재생규칙으로는 해의 빠른 수렴을 막으면서 탐색의 범위를 넓힐 수 있는 Selection, Ranking, Tournament기법들[5]을 적용하였다.

Selection기법에서 목적함수식으로는  $p(x) = \overline{A}/F(x)$ 를 사용하였다. 여기서  $p(x)$ 는 복제될 확률,  $F(x)$ 는 string의 적합도(fitness), 그리고  $\overline{A}$ 는 string의 적합도 평균치를 나타낸다. Ranking기법에서 스케일링 파라미터로  $q=0.15$ ,  $r=0.01$ 을 설정하였다. Tournament기법에서 tournament의 크기는 2개로 하였다.

### (5) 적합도 평가(fitness evaluation)

적합도는 string을 토대로 블럭단위 배치를 수행한 결과 구해진 활동간 총이동거리로서 아래 식으로 표현된다.

$$\text{Min } F(x) = \sum_{i=1}^n \sum_{j \neq i} w_{ij} \cdot d_{ij}$$

여기서

$F(x)$  : 총이동거리

$n$  : 활동개수

$w_{ij}$  : 활동  $i$  와  $j$  사이의 흐름량

$d_{ij}$  : 활동  $i$  와  $j$  사이의 직각거리

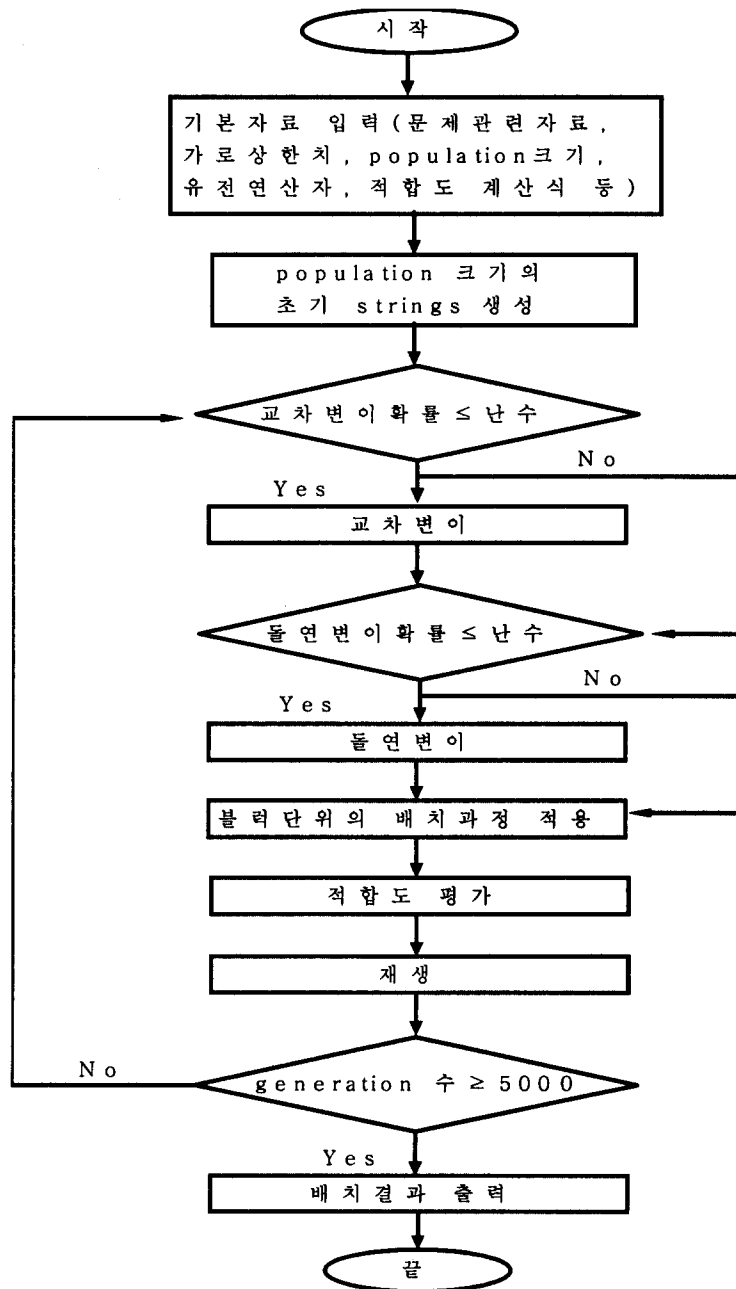
### (6) 유전연산자 발생확률

유전자 알고리즘의 적용에서 사용할 유전연산자의 종류와 발생확률 그리고 재생규칙은 각 문제마다 계산실험을 통하여 결정하였다. 교차변이와 돌연변이의 확률은 각각 0.1~1.0 사이에서 0.1씩 증가하면서 실험하였다.

제안된 블럭단위 배치기법의 과정을 순서대로 요약 정리하면 그림 2와 같다.

## 4. 계산실험

제안된 유전자 알고리즘은 C++ 언어로 프로그래밍 되었으며, 펜티엄급 586 PC를 이용 실험을 수행하였다. 실험문제로는 기존의 여러 논문들에 소개되어 있는 문제중 활동공간의 수가 5~30개인 문제[7, 10]들을 선정하였다. 실험문제와 함께 실험에서 사용한 모든 자료는 저자들로부터 입수할 수 있다. 유전자 알고리즘의 적용시 population 크기는 20개 string으로 하였으며, 모든 실험에서 탐색은 5000 generations에서 종료하였다.



〈그림 2〉 제안된 배치기법의 순서도

먼저, PC를 이용하여 최적의 배치해를 구할 수 있는 5~10개 활동공간 문제에 대해 제안된 배치기법의 최적성을 분석하였다. 최적성은 제안된 배치기법에 의한 해의 에러율로써 표현되는데, 에러

율은 최적해에 대한 제안된 배치기법 해의 오차를 백분율로써 나타낸 것이다. n개 활동공간 문제의 최적해는 n!개 string의 모든 경우에 대해 블록단위 배치를 적용하여 이들중 최소의 총이동

〈표 1〉 5~10개 활동공간 문제에서 제안된 배치기법의 최적성 분석

활동수(개)	활동공간의 형태	제안된 배치기법에 의한 해	최적해	제안된 배치기법의 에러율
5	(i)	512	512	0.00%
	(ii)	533	533	0.00%
6	(i)	896	896	0.00%
	(ii)	884	884	0.00%
7	(i)	1319	1319	0.00%
	(ii)	1230	1230	0.00%
8	(i)	1965	1940	0.02%
	(ii)	1806	1806	0.00%
9	(i)	2673	2673	0.00%
	(ii)	2644	2664	0.00%
10	(i)	3709	3709	0.00%
	(ii)	3591	3519	0.02%

거리를 갖는 배치해이다.

유전자 알고리즘의 적용에서 교체변이 연산자로는 발생확률 0.25로써 PMX방법을, 돌연변이 연산자로는 발생확률 0.1로써 Insertion방법을, 그리고 재생규칙으로는 Tournament방법을 사용하였다. 이들은 실험을 통해 결정된 것이다. 표 1에 제안된 기법의 최적성에 대한 분석결과가 정리되어 있다. 제안된 배치기법은 표에 나타난 바와 같이, 활동수가 8개와 10개인 두가지 경우만을 제외하고 모든 문제에서 최적해를 구할 수 있었다.

PC를 이용하여 최적해를 구하기 어려운 15~30개 활동공간 문제들에 대한 제안된 배치기법의 수행도는 표 2에서와 같이 Tam의 알고리즘에 의한 배치결과[10]와 비교하였다. 표에서 제안된 배치기법의 효율은 Tam의 알고리즘 적용에 의해 얻어진 해에 대한 제안된 배치기법에 의한 해의 개선정도를 백분율로써 나타낸 것이다. 비교결과, 제안된 배치기법에 의한 배치는 대략 10~35%의 총이동거리 개선효과를 보였다. 유전자 알고리즘의 적용을 위한 연산자와 재생규칙의 결정은 실

〈표 2〉 15~30개 활동공간 문제에서 제안된 배치기법의 수행도 평가

활동수(개)	활동공간의 형태	제안된 배치기법에 의한 해	Tam의 알고리즘에 의한 해	제안된 배치기법의 효율
15	(i)	9120	13762	33.7%
	(ii)	9855	12240	19.4%
20	(i)	21885	26921	18.7%
	(ii)	22656	28646	20.9%
30	(i)	50492	55668	9.3%
	(ii)	52885	58824	10%

〈표 3〉 30개 활동공간 문제에서 연산자와 재생규칙 조합에 따른 제안된 배치기법의 수행도 비교

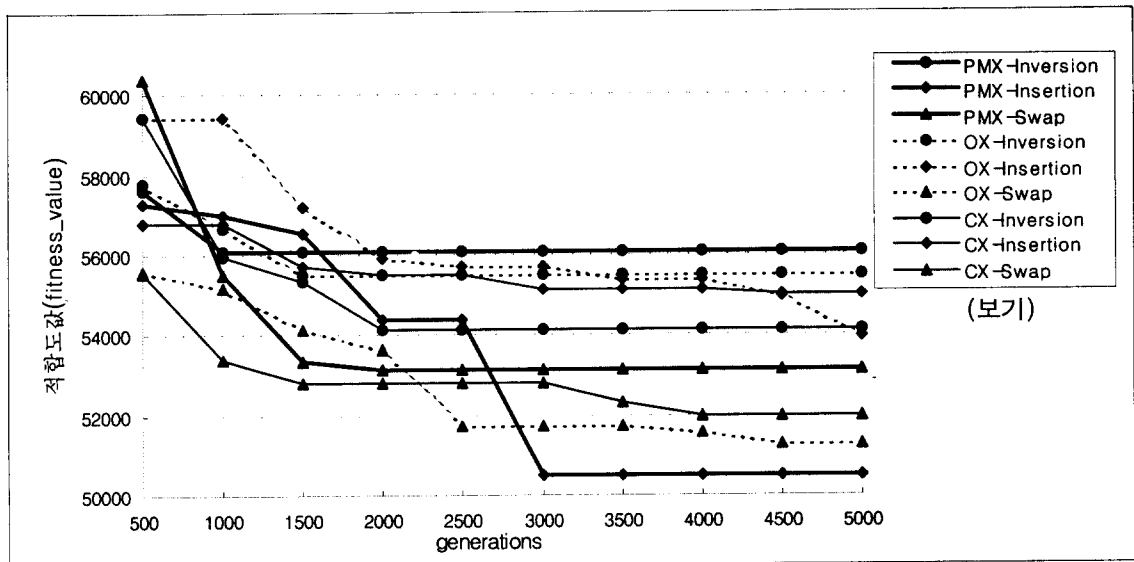
활동공간의 형태	재생규칙	Selection			Ranking			Tournament		
		Inversion	Insertion	Swap	Inversion	Insertion	Swap	Inversion	Insertion	Swap
(i)	PMX	59471	58857	56663	65418	57418	58231	56067	50492	51242
	OX	59854	57073	58724	56149	58232	55753	55494	53963	51294
	CX	56573	58211	59579	59406	58963	58267	54129	55017	51967
(ii)	PMX	58649	57949	59903	58186	57481	59429	55861	54989	53125
	OX	59418	59108	59144	58667	58338	56207	56119	53096	54187
	CX	58843	58123	58091	58991	57507	58337	56130	53194	52869

험을 통해서 이루어졌다.

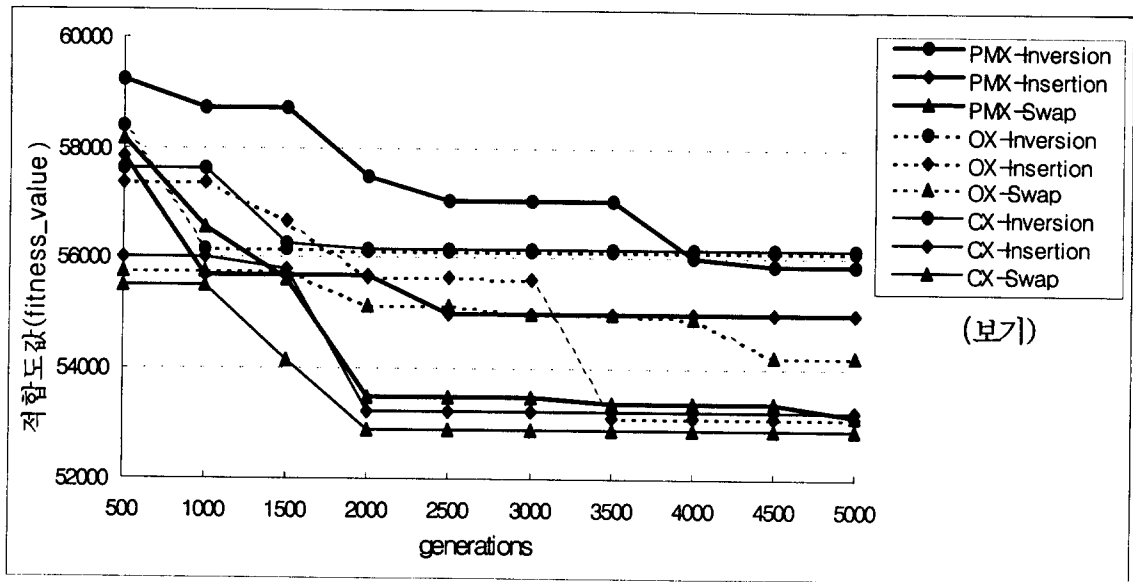
참고로, 30개 활동공간 문제에서 유전연산자와 재생규칙 조합에 따른 제안된 배치기법의 수행도를 비교실험한 내용이 표 3에 정리되어 있다. 활동공간의 형태가 (i)인 경우, 교차변이 연산자로는 발생확률 0.25로써 PMX방법이, 돌연변이 연산자로는 발생확률 0.1로써 Insertion방법이, 그리고 재생규칙으로는 Tournament방법이 가장 좋은 수행도를 보였다. 블럭의 폭을 10으로 고정한 (ii)

의 경우, 교차변이 연산자로는 확률 0.1로써 CX방법이, 돌연변이 연산자로는 확률 0.1로써 Swap방법이, 그리고 재생규칙으로는 Tournament방법이 가장 좋은 수행도를 나타내었다.

그림 3은 30개 활동공간 문제에서 Tournament 재생규칙을 사용할 때 유전연산자 조합에 따른 제안된 배치기법 해의 변화를 그래프로 보여준다. (a)와 (b) 경우에서, 제안된 배치기법의 수행도는 연산자조합과 generation 수의 증가에 따라 전



(a) 활동공간의 형태가 (i)인 경우



(b) 활동공간의 형태가 (ii)인 경우

〈그림 3〉 30개 활동공간 문제에서 연산자의 조합과 generation수의 증가에 따른 제안된 배치기법 해의 변화

혀 다르게 변화하였다. 따라서 최선의 유전연산자와 재생규칙은 모든 실험문제 각각에 대해서 수행한 계산실험을 통해 결정하였다.

그림 4는 30개 활동공간 문제에서 제안된 배치기법을 적용한 결과 얻어진 배치결과를 보여준다. (a)의 경우, 활동공간의 가로와 세로를 고정된 크기로 가정하였기 때문에 블록의 폭은 한 블록에 포함된 활동들의 세로길이 최대치에 의해 결정된다. 따라서 통로간 간격이 일정하지 않게 된다. (b)의 경우, 블록의 폭이 주어지기 때문에 통로간 간격이 일정하다.

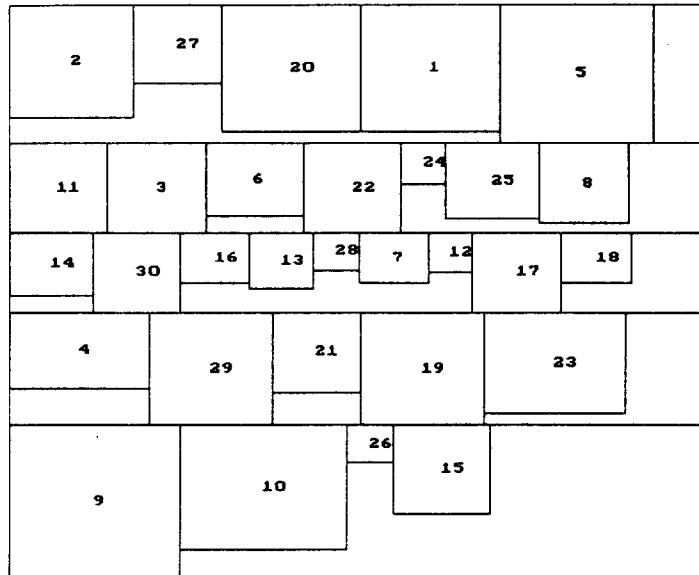
## 5. 결 론

본 논문에서는 유전자 알고리즘을 이용하여 활동간 총이동거리를 최소화하는 최적의 해를 탐색하기 위한 블록단위의 설비배치기법을 제안하였다. 이 기법은 직사각형 전체영역의 가로크기를

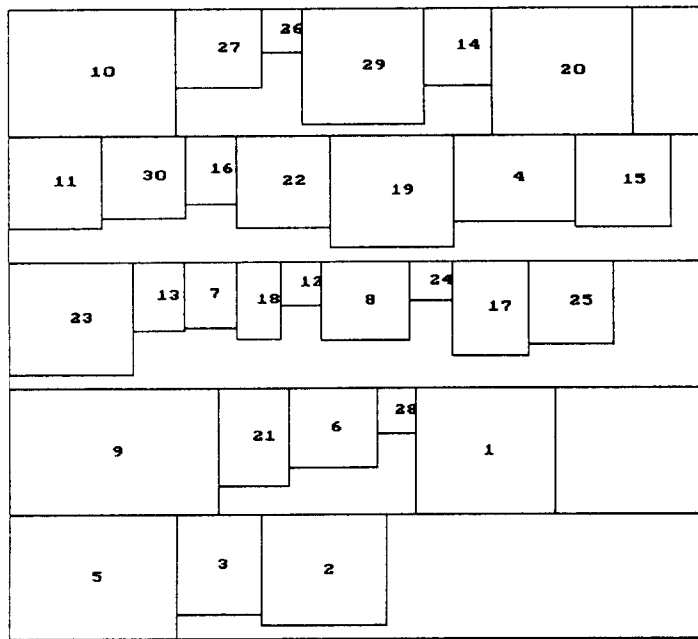
원하는대로 유지할 수 있을 뿐더러 활동간의 가로방향 통로를 생성해 주기 때문에, 기존의 배치기법들과 비교하여 보다 실용적인 배치해를 구할 수 있게 해준다.

배치할 직사각형 활동공간의 형태를 가로와 세로의 크기에 따라 두가지 경우로 나누어 다양한 실험을 수행한 결과, 제안된 배치기법은 모든 문제에서 뛰어난 성능을 보였다. 특히, 5~10개 활동공간 문제에서는 거의 모든 경우에서 최적해를 구할 수 있었으며, 15~30개 활동공간 문제에서는 Tam의 알고리즘과 비교하여 대략 10~35% 해의 개선효과를 보였다. 또한 실험결과, 제안된 배치기법의 수행도는 유전연산자와 발생확률 그리고 재생규칙의 조합과 generation 수의 증가에 따라 매우 유의적으로 변화하는 것으로 밝혀졌다. 따라서 최선의 유전연산자와 발생확률 그리고 재생규칙은 모든 실험문제 각각에 대한 계산실험을 통하여 결정하였다.





(a) 활동공간의 형태가 (i)인 경우



(b) 활동공간의 형태가 (ii)인 경우

<그림 4> 30개 활동공간 문제의 배치결과

추후, 직사각형외의 다양한 형태의 전체영역과 활동공간을 다루면서 가로(수평)와 세로(수직)방향의 통로를 동시에 구축해 주는 배치기법에 대한 연구가 지속적으로 필요하다. 나아가서 유전자 알고리즘의 적용에서 정성적 요인을 반영한 새로운 적합도 평가방법의 개발은 좀더 실용적인 해의 생성에 크게 도움을 줄 것이다.

## 참 고 문 헌

- [1] K. C. Chan, and H. Tansri, "A Study of Genetic Crossover Operations on the Facilities Layout Problem", *Computers and Industrial Engineering*, Vol. 26 557-550, 1994
- [2] F. Glover, J. P. Kelly and M. Laguna, "Genetic Algorithm and TabuSearch : Hybrids for Optimization", *Computers and Operations Research*, Vol. 22, 11-134, 1995.
- [3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.
- [4] A. Kusiak and S. S. Heragu, "The Facility Layout Problem", *European Journal of Operations Research*, Vol. 29, 229-251, 1987.
- [5] Z. Michalewicz, *Genetic Algorithms + Data structure = Evolution Programs*, 2nd ed., Springer-Verlag, 1994.
- [6] R. Muther, *Systematic Layout Planning*, Industrial Education Institute, Boston, 1961.
- [7] C. E. Nugent, T. E. Vollman and J. Ruml, "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations", *Operations Research*, Vol. 16, 150-173, 1968.
- [8] G. Suresh, V. V. Vinod and S. Sahu, "A Genetic for Facility Layout", *International Journal of Production Research*, Vol. 33, 3411-3423, 1995
- [9] K. Y. Tam, "A Simulated Annealing Algorithm for Allocating Space to Manufacturing Cells", *International Journal of Production Research*, Vol. 30, 63-87, 1991.
- [10] K. Y. Tam, "Genetic Algorithms, Function Optimization, and Facility Layout Design", *European Journal of Operations Research*, Vol. 63, 322-346, 1992.
- [11] J. A. Tompkins and J. A. White, *Facilities Planning*, 2nd ed., Wiley, 1996.
- [12] J. A. White, *Facilities Layout and Location : An Analytical approach.*, 2nd ed., Prentice Hall, NJ, 1992.