工學碩士 學位論文

Hybrid GA를 이용한 블록단위의 최적설비배치에 관한 연구

A Study on Optimal Facility Layout of Block Facility using Hybrid GA

指導敎授 李 哲 榮

2001年 2月

韓國海洋大學校 大學院

物流시스템工學科

李 容 煜

목 차

List of Tables

List of Figures

A Study on Optimal Facility Layout of Block Facility using Hybrid GA

Yong-Wook, Lee

Department of Logistics Engineering

Graduate School of Korea Maritime University

Abstract

Facility layout is the early stage of system design that requires a mid-term or long-term plan. Since improper facility layout might incur substantial logistics cost including material handling and re-installment costs, due consideration must be given to decisions on facility layout.

Facility layout is concerned with how to arrange equipment necessary for production in a given space. Its objective is to minimize the sum of all the products of each equipment's amount of flow multiplied by distance. Facility layout also is related to the issue of NP-complete, i.e., calculated amounts exponentially increase with the increase of the number of equipment.

This study discusses Hybrid GA developed, as an algorithm for facility layout, to solve the above-mentioned problems. The Algorithm, which is designed to efficiently place equipment, automatically produces a horizontal passageway by the block, if a designer provides the width and length of the space to be handled. In

addition, this study demonstrates the validity of the Algorithm by comparing with existing algorithms that have been developed.

We present a hybrid GA approach to the facility layout problem that improves on existing work in terms of solution quality and method.

Experimental results show that the proposed algorithm is able to produce better solution quality and more practical layouts than the ones obtained by applying existing algorithms.

제 1 장 서 론

1.1 연구의 배경 및 목적

오늘날, 제조원가에 큰 비중을 차지하는 물류비용을 절감하려는 시도가 다양하게 전개되고 있다. 특히, 일단 설치 후에는 변경에 많은 시간과 비용이 따르게 되는 제조 시스템의 경우, 초기 설계시에 관련비용을 최소화하려는 노력이 요구된다. 따라서 좋은 설비배치안을 얻는 것은 이러한 물류비용 절감에 크게 기여할 것임에 틀림없다.

1955년 이래로, 미국 GNP의 약 8% 정도가 매년 새로운 설비를 신축하는 비용으로 투자되고 있고 제조업체 총 운영비용의 20 50%가 자재취급 때문에 발생된다고 추정되고 있다. 한편으로는 효과적으로 설비계획을 할 경우 자재취급비용을 적어도 10 30% 절감할 수 있다는 것을 의미한다. 따라서 설비계획을 효과적으로 수행하였다면, 미국의 연간 제조생산성은 지난 15년간의 어떤 해에 증가된 것보다 약 3배 이상 증가되었을 것이다. 우리 나라에서의 정확한 통계치는 없지만, 고도의 성장을 거듭해온 상황을 고려할 때, 매년 매우 많은 설비들이 계획되어지고 신축되어지고 있다고 생각되어진다. 이러한 상황 아래에서의 설비계획과 설비배치에 관련한 중요성은 매우 높다고 할 수 있다.

시설배치문제는 경영 및 관리적인 면에서 기업의 많은 관심의 대상이 되어 왔다. 작업자의 설비 및 자재 등의 생산요소를 효율적으로 배치한다는 것은 기업의 능률 및 잠재이익과 나아가서는 기업의 사활에도 긴밀한 관련이 있다. 특히, 생산물류의 문제는 제조과정에서 각 설비간에 빈번한 흐름이 존재하므로 각 설비나 활동공간을 어떻게 배치하느냐에 따라 물류비의 절감 효과를 기대할 수 있다. 생산물류의 한 형태인 설비배치문제는 각 설비간의 흐름량과 거리에 따라 총이동거리가 결정된다. 총이동거리는 총비용으로 산정될 수 있고 설비배치문제는 각 설비간의 총이동거리를 최소화하는 것이 궁극적인 목적이다.

본 연구에서는 설계자가 배치할 전체공간의 가로와 세로의 크기를 정해주면 가로와 세로의 크기가 정해진 설비를 수평방향의 블록단위로 통로가 자동적으로 생성되면서 각 설비를 효율적으로 배치하는 방법을 제시한다. 제시한 설비배치문제를 보다 효율적으로 수행하기 위해 Hybrid GA(Genetic Algorithm)를 이용한 설비배치알고리즘을 개발·적용하여 최적의 배치를 탐색하는 새로운 방법을 소개한다. 또한, 기존의 관련 연구와 비교·분석하여 제시한 설비배치알고리즘의 유효성과 효율성을 검증하고자 한다.

1.2 기존 연구의 고찰

설비배치를 체계적으로 수립하는 방법에 대해 많은 연구들이 수행되었다. 실제 설비배치에 가장 많이 활용되고 있는 것은 Muther의 체계적 배치계획(System Layout Planning)으로서 현장에서 많이 이용되고 있다. 또한, 다량의 계산을 빠른 시간 내에 가능하게 하는 컴퓨터의 출현으로 설비배치문제에 적용되는 컴퓨터 프로그램은 현재 많은 종류가 발표되었다.

배치문제는 이차할당문제(Quadratic Assignment Problem : QAP)로 모형화할 수 있고 QAP를 해결하기 위한 알고리즘은 크게 최적화 알고리즘과 휴리스틱 알고리즘으로 분류할 수 있다.

최적화 알고리즘의 경우, 예를 들어 생산의 흐름이 다양한 Job shop의 경우를 보면 단지 20개의 부서나 기계를 배치하는데 있어서도 608조 개 이상의 가능한 경우들의 조합이 있을 수 있는데, 이런 문제에 있어서 수없이 많은 대안 중 과연 어느 것이 최적의 배치안인지를 구하는 것은 불가능하다. 따라서 배치문제를 해결하기 위한 여러 가지 휴리스틱 알고리즘이 개발되었다.

이러한 휴리스틱 알고리즘은 무에서 배치안을 생성하는 배치안 생성 알고리즘(construction algorithm)과 초기 배치안을 이용하여 설비들의 위치를 바꾸어 가며 최적 배치안을 찾는 개선 알고리즘(improvement algorithm), 이들을 조합하여 사용하는

혼합형 알고리즘(hybrid algorithm), **그리고 그래프이론을 이용한 알고리즘**(graph theoretic algorithm)**으로 분류되어 발전되었다**[29].

배치형태	배치 알고리즘	제안자	내 용
배치안 생성 알고리즘	ALDEP	Seehof and Evans 1967	임의로 한 설비를 선택하여 좌측상단에 배치하고 이 설비와의 관계를 고려해 다음 설비를 배치
	CORELAP	Lee and Moore 1967	총인접도순위를 고려하여 설비를 배치하는 것
	MAT	Edward et al. 1970	설비들간의 물동량에 따라 설비쌍을 선정하여 배치하는 것
	FATE	Block 1978	MAT를 확장한 FATE
	FLAT	Heragu and Kusiak 1986	설비들간의 물동량과 거리의 곱을 기준으로 3개의 설비를 한 쌍으로 하여 설비들을 배치하는 것
개선 알고리즘	CRAFT	Armour and Buffar 1963	흐름 입력자료를 사용하여 거리에 바탕을 한 목적함수에 의해서 배치
	FLAT	Khalil 1973	자재운반설비의 이동비용을 고려한 설비배치
	MULTIPLE	Bozer et al. 1994	부지를 작은 단위의 정사각형으로 나누어 Spacefilling Curve를 이용하여 설비들을 배치하는 것
혼합형 알고리즘	FLAC	Scriabin and Vergin 1995	설비들간의 거리를 기준으로 설비들의 상대위치를 결정한 후 설비들의 크기 제약을 반영하고, 이를 FLAT에서 사용한 알고리즘을 이용하여 개선하는 것
	BLOCKPLAN	Donaghey and Pire	임의로 초기배치안을 생성한 후 2-opt 알고리즘을 이용한 개선알고리즘을 통해 설비배치

Table 1.1 **설비배치에 관한 기존 연구들**

최근에 들어서는 Tate와 Smith가 1995년 설비들을 설비들의 면적과 aspect ratio를 고려하여 연속평면상에 배치하는 알고리즘을 개발하였다[28].

1995년 Suresh et al은 유전자 알고리즘을 이용하여 각 후보지의 크기가 동일한 경우 총 비용을 최소화하는 설비 배치에 관한 연구를 하였다[22]. 그들은 이 연구에서 QAP을 위한 새로운 Crossover 연산자를 제시하였다.

1996년 이명수 등은 GA에서 X교차 연산자를 개발하여 GA의 기본연산자와 CRAFT의 결과보다도 효과적임을 보였고[12], 1997년 김재곤 등은 그래프이론을 응용하여 Bay를 생성하며 설비를 배치하는 알고리즘을 제시하였다[2]. 그러나 이 알고리즘들도 설비들의 고유한 형태를 유지하지 못하는 약점이 있다.

우성식 등은 1997년 설비들의 고유한 형태를 유지하는 설비배치 알고리즘을 제시하였다[1].

1.3 연구 내용 및 구성

본 논문의 제1장에서는 연구의 배경 및 목적, 기존의 설비배치에 관한 연구들에 대해서 설명하였다.

제2장에서는 본 연구에서 대상으로 하는 설비배치문제의 정의와 설비배치문제의 형태에 관하여 설명한다.

제3장에서는 본 연구에서 대상으로 하는 문제를 해결하기 위하여 제안된 유전자 알고리즘과 시뮬레이티드 어닐링에 관하여 설명하고, 이것을 설비배치문제에 적용하는 단계에 관하여 설명한다.

제4장에서는 기존 연구의 자료를 이용하여 실험을 수행하고 평가한다.

마지막으로, 제5장에서는 본 연구의 결과를 설명하고 추후 연구 방향을 제시한다.

제 2 장 설비배치문제

2.1 설비배치문제의 정의

설비배치(facilities layout)란 배치대상이 되는 설비들의 위치를 평면부지에 할당하는 것으로서 건축 부지 계획, 생산시스템 배치, 사무공간 배치 또는 초고밀도 집적회로 설계 등 매우 다양한 응용분야를 가진다[13].

설비배치문제(facilities layout problem)는 배치의 대상되는 설비들의 상호 인접관계를 고려하면서 설비들간의 물자의 이동거리 또는 정보의 이동시간을 최소화함으로써 원활하고 효율적인 시스템을 구축하기 위한 설비들의 위치배열을 설계하는 것으로 시스템의 운용비용과 효율에 상당한 영향을 끼친다. 또한, 한번 결정된 설비배치를 바꾸기 위해서는 많은 비용과 노력이 요구되므로 초기의 배치설계단계에서 배치계획의 최적화가 필수적이다.

설비배치에 관한 문제는 설비계획문제의 일부분으로 이것을 분류하면 Fig. 2.1과 같은 분류를 가진다. 설비위치(facilities location)문제는 거시적으로 설비의 입지조건이나 소비자와의 관계 등을 고려해 설비의 위치를 선정하는 것이며, 설비설계(facilities design)문제는 설계요소들이 설비목적을 성취하도록 어떻게 지원할 것인가를 결정하는 것으로서 구성요소들을 계획하고 선정하는 설비지원시스템설계(facilities system design)와 자재, 정보 또는 구성요소들의 이동들을 고려해 설비목적을 지원하는 취급시스템설계(handling system design), 그리고 배치설계(layout design)로 이루어진다.

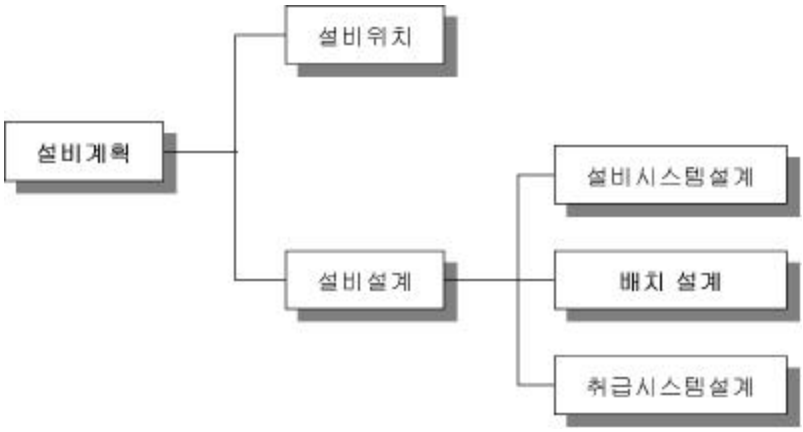


Fig. 2.1 **설비계획의 분류** (Tomkins et al. 1996)

설비배치는 설비들의 형태들이 각 내용에 따라 다르지만 문제의 형성은 같다. 각 설비의 면적과 흐름량은 생산기술자의 측정 데이터로부터 얻어진다. 각 설비간의 면적과 흐름량을 이용하여 총이동거리를 최소로 하고 실제적인 형태의 배치를 형성시키는 것이 설비배치의 목적이라 할 수 있다. 설비배치의 문제는 두 활동간의 가중치가 클수록 기본적으로 두 설비간은 근접시켜야 한다. 설비배치문제는 설비의 수가 증가할수록 계산량이 지수적으로 증가하는 NP-complete문제에 속한다. 물자의 흐름량을 가중치라 정의하고 정량적인 면만을 고려한 일반적인 배치문제의 목적함수는 아래의 식 (2.1)과 같다.

$$Min \quad F(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} C_t \cdot a_{ij} \cdot d_{ij}, \qquad i, j = 1, 2, 3, \cdots, n \qquad (2.1)$$

여기서, a_{ij} = **설비 i 와 j 사이의 흐름량**

C_t = **단위 물동량을 단위거리 이동하는데 드는 비용**

d_{ij} = **설비 i 와 j 사이의 거리**

본 논문에서는 C_t = 1로 놓고, a_{ij}와 d_{ij}의 곱인 설비의 총이동거리에 중점을 두기로 한다. 즉, 물동량과 각 설비(부서)간 거리 곱의 총합을 최소화하는 설비들의 배열을 결정하는 문제로 한다. 각 설비간 물자의 흐름량은 고정되어 있고 설비간 거리는 배치의 순서에 따라 다르므로 물동량이 큰 설비들은 가깝게 위치시켜야 한다.

2.2 설비배치의 형태

2.2.1 블록단위 설비배치의 정의

설비를 배치함에 있어서 구축과정에서 가로방향의 통로가 생성되도록 하였다. 각 설비는 서로 다른 면적의 직사각형으로 가정하였으며 각 설비 사이의 거리는 중심점 사이의 직각거리로 가정하였다. 각 설비 사이의 흐름량은 사전에 주어진다. 설계자가 배치할 전체영역의 우측 상한치를 정해주면 순서대로 설비를 배치하다가 우측 상한치를 초과할 경우 통로가 생성되면서 새로운 열의 좌측부터 다시 배치해간다. 여기서 하나의 통로가 생성되면서 배치되는 공간을 블록이라고 정의하였다. Fig. 2.2에서 보듯이 각 설비들은 순차적으로 좌측에서 우측으로 배치하며 우측 상한치를 초과할 때 블록이 생성되고 다음 블록을 배치할 때 통로가 생성된다.

2.2.2 블록단위 배치의 형태

각 설비의 크기는 가로와 세로가 고정되어 있는 경우와 면적이 주어지고 가로/세로 비율의 상한치가 주어져 있는 경우로 나누었다. 이러한 두 가지 경우에 대해 본 논문에서 제안하고 있는 Hybrid GA를 이용한 설비배치알고리즘을 적용하여 배치안을 형성하였다. 두 경우에 배치를 형성하는 과정을 요약하여 설명하면 다음과 같다.

(ⅰ) 배치방법 (Type 1)

: 각 설비의 가로와 세로의 크기가 고정되어 있는 경우

각 설비의 가로와 세로가 고정되어 있는 경우의 배치는 통로의 생성이 용이하지 않는데 본 논문에서는 수평방향의 통로를 생성하기 위해 서로 다른 면적의 설비를 수평의 형태로 나란히 배치해 나간다.

각 설비를 순차적으로 배치할 때 우측 상한치를 초과할 경우 블록이 생성되면서 다음 블록에 배치해 나간다. 블록의 폭(세로)은 인접 설비 중 세로의 길이가 가장 큰 것을 기준으로 다음 통로가 생성되도록 하였다. 그러므로 블록의 폭은 일정하지 않다. 따라서 이러한 배치 형태는 총이동거리를 최소화하기 위해 각 블록을 형성할 때 총이동거리와 잉여공간을 줄이는 방향으로 배치된다.

이러한 방법은 고정된 가로와 세로를 가정하였기 때문에 전체영역의 모양은 수평방향으로는 일정하지만 수직의 방향은 일정하지 않게 나타난다. 그러나 잉여공간의 존재를 인정하고 수행도의 최소화를 목적으로 한다. 따라서 배치결과를 재조정할 필요가 없게 하였다. 그러므로 불규칙한 모양의 배치형태가 나타날 때 실제 최종배치 때의 수작업을 통한 수행도의 변화를 없게 하였다.

(ⅱ) 배치방법 (Type 2)

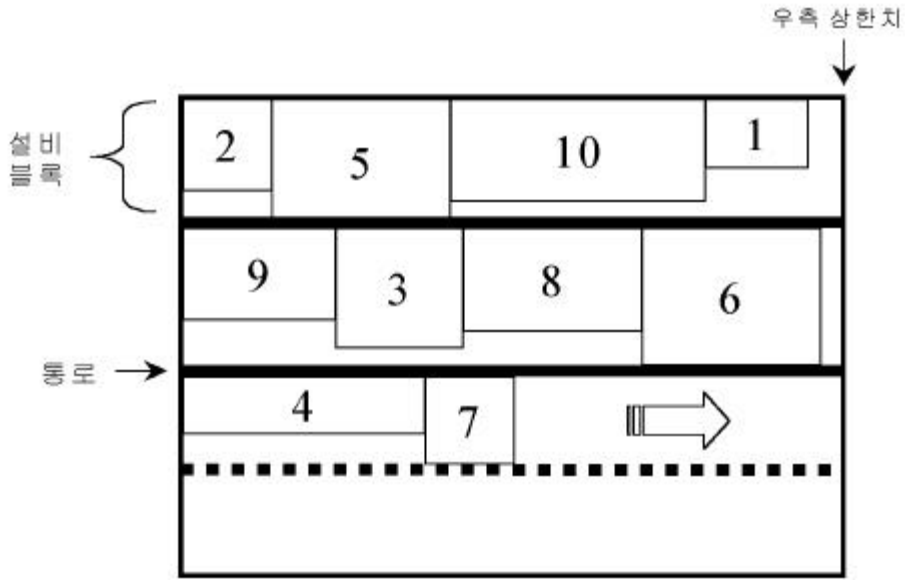
: 각 설비의 가로/세로 비율의 상한치와 블록의 폭이 주어지는 경우

각 설비 사이의 가로/세로 비율의 상한치가 주어지는 경우에는 설계자가 미리 블록의 폭을 정해주면 각 설비의 세로는 블록의 폭이 되고 가로는 주어진 면적에 의해 계산된다. 계산된 가로의 길이가 가로/세로 비율의 상한치를 초과하면 가로/세로 비율의 상한치에 맞게 가로와 세로가 계산되어 배치된다. 따라서 이 경우에는 블록의 폭은 일정하게 유지된다. 여기서 가로/세로 비율의 상한치를 초과하게 되면 상한치에

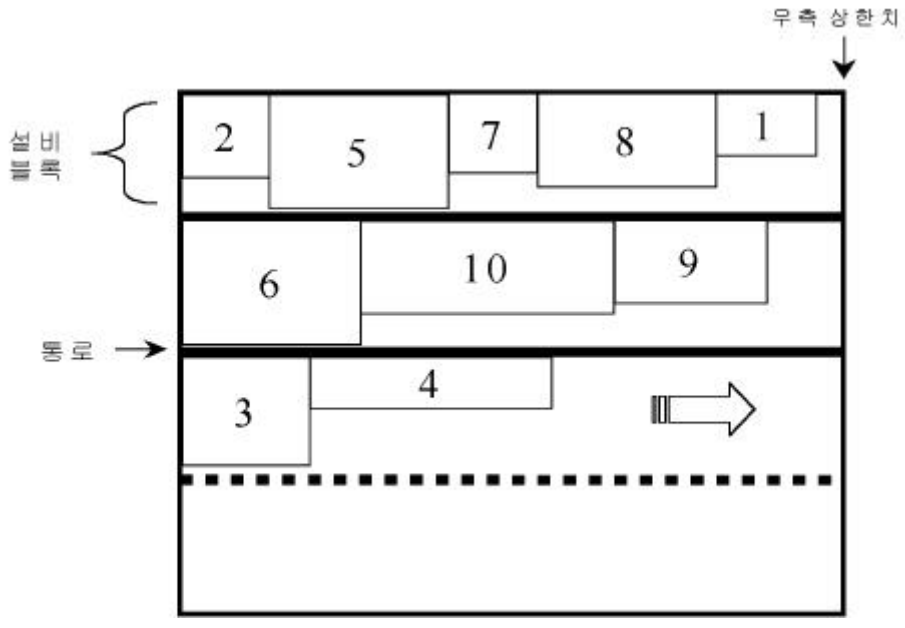
맞게 가로의 값이 계산되고 면적에 의해 세로의 값이 다시 계산된다. 그러므로 가로/세로 비율의 상한치를 초과한 경우의 설비의 세로크기는 블록의 폭보다 작게 된다. 이렇게 배치를 하면 잉여공간이 생기는데 이것은 각 설비의 실제 형태를 유지하기 위한 것이다. 만약 가로/세로 비율의 상한치가 주어지지 않는다면 블록의 폭이 모든 설비 사이의 세로의 크기가 되므로 잉여공간은 생기지 않는다. 하지만 실제적인 활동공간의 형태를 유지하기 위해 가로/세로 비율로 상한치에 맞게 조정하여 배치해야한다.

이러한 배치의 형태는 전체영역이 수평방향으로 격자형인 경우에 사용할 수 있는데 일반적인 부서 배치나 사무실배치, 공장라인의 배치에 적합한 형태라고 할 수 있다. 실험에서 각 통로의 폭은 굵은 선으로 나타나게 하였고 실제 배치에서는 통로의 폭을 정해주면 전체배치의 형태에서는 통로의 폭이 나타나게 할 수 있다.

위에서 가정한 두 가지 형태의 배치는 우측 상한치를 결정해서 순차적으로 배치하는 것이므로 배치 설계자가 전체영역의 형태를 알아야하며 수시로 전체영역의 범위의 조정이 가능하다. 따라서, 최종배치 전에 모의 실험을 통해 총이동거리가 최소인 전체영역의 형태를 파악할 수 있다[1].



배치 방법 : Type 1 (**블록 폭이 일정하지 않음**)



배치 방법 : Type 2 (**블럭 폭이 일정**)

Fig. 2.2 **블록단위 설비배치의 2가지 형태**

제 3 장 설비배치 알고리즘 개발

3.1 조합최적화문제를 위한 해법

여러 분야에서 최적화에 대한 중요성은 꾸준히 증가하고 있다. 선형과 비선형 최적화 즉, 연속변수 함수의 최적해를 위한 탐색 등은 1950년대와 1960년대에 큰 진전이 있었다. 이중 1947년 Dantzig에 의해서 소개된 선형 계획법 문제를 해결하는 심플렉스 알고리즘이 그 중 대표적인 예이다.

이산 변수 함수의 최적해를 위한 탐색인 조합최적화 분야에 있어서의 중요한 결과들은 1970년대 주로 이루어졌다. 예를 들면, 조합최적화 분야에서 가장 잘 알려진 TSP(Traveling Salesman Problem)문제는 1970년대에 근사화와 최적화 알고리즘의 두 분야 모두에서 높은 수준의 질을 가지고 등장하게 되었다.

하지만, 오늘날에도 많은 큰 규모의 조합최적화문제들은 현대의 컴퓨터를 사용하여도 근사적으로 풀 수밖에 없다. 이것은 많은 문제들이 NP-complete로 증명되었기 때문이다. 그러한 문제들은 잘 풀리지 않는다. 왜냐하면 문제 규모의 다항 함수로 제한되는 계산량에 의해서만 풀릴 수 있기 때문이다. 즉, 대부분의 NP-complete 문제들은 지수적으로 증가하는 계산량을 필요로 하기 때문이다. 따라서, 우리는 현실 세계에 존재하는 많은 큰 규모의 조합최적화문제에서 최적해를 구하기가 어려운 상황이다. 이것은 최적해를 구하기 위한 탐색에 엄청난 계산량이 요구되기 때문이다. 이러한 이유로 휴리스틱 또는 근사알고리즘인 Genetic Algorithm, Simulated Annealing, 타부 서치 등을 쓸 수밖에 없게 된다.

설비배치문제는 조합적인 성격 때문에 배치하여야 할 설비의 수가 15개 이상인 경우에는 경험적으로 보아 최적해를 구하는 것이 불가능한 NP-Complete문제에 속하는 조합최적화문제이다. 이러한 대규모의 계산량을 요구하는 조합최적화문제를 해결하기 위한 방법들에는 여러 가지가 있지만 다음과 같은 세 가지 유형으로 나눌 수 있다.

휴리스틱 방법으로 조합최적화문제를 풀 경우, 적용 가능한 문제가 특정한 문제에 한정되고 적용가능한 문제들에서도 보통 여러 개의 초기 상태에서 탐색을 반복적으로 수행해서 계속적으로 더 좋은 해를 선택함에 따라 구한 해들 중 제일 좋은 것을 선택하여 최종해로 한다.

이러한 방법을 사용하면 문제의 크기가 점차 커짐에 따라 많은 초기 상태에서의 반복수행이 필요하게 되어 계산시간이 많이 소요되며, 국소 최소점에 빠질 수가 있는데, 휴리스틱 방법의 경우 이를 최종해로 선택하게 됨으로 항상 최적해를 찾는다는 보장을 할 수 없는 단점이 있다.

신경회로망(Neural network)은 인간의 신경구조를 모방한 것으로 간단한 연산을 수행하는 뉴런(neuron)들이 상호 연결되어 병렬로 동작하는 모델로써 병렬 수행으로 인해 빠른 시간에 문제를 풀 수 있다. 그러나 이 방법으로 조합최적화 문제를 해결할 경우 문제를 각 뉴런의 동작으로 모델링 해야 하기 때문에 한계가 따르며, 휴리스틱 방법과 마찬가지로 국소 최소점에 빠질 가능성이 있어 최적해를 찾지 못하는 단점이 있다.

유전자 알고리즘(Genetic algorithm)은 모든 최적화 문제에 적용이 가능한 범용성이 뛰어난 알고리즘으로써 유전 형질이 우수한 개체를 다음 세대로 전이시키는 과정을 여러 세대 반복함으로 해서 국소 최소점을 찾아가고, 또한 이러한 계산과정과 함께 여러 가지 유전 연산자들을 이용해서 전역적 최적해를 찾아간다. 이러한 전역적 최적해를 발견할 수 있는 높은 가능성과 목적함수에 대한 수학적 제약이 거의 없기 때문에 최적화 문제, 적응적 탐색과 학습, 분류시스템, 신경회로망의 학습 문제, 퍼지 규칙 베이스의 자동 생성 등의 여러 응용분야에 적용되고 있다.

시뮬레이티드 어닐링(Simulated annealing)은 반복적 개선법(Iterative improvement)과 유사한 방법으로 해를 탐색하지만 반복적 개선법과는 달리 좋지 않은 해로의 이동도 확률적으로 받아들임으로써 국소 최소점에서 벗어나 전역적 최적해를 찾아가는 알고리즘이다.

본 논문에서는 조합최적화문제에서 전역적 최적해를 찾아가는데 우수성을 입증

받고 있는 GA와 SA의 단점을 보완하고 그 장점을 결합하는 Hybrid GA 방법을 제안하고 개발한다.

GA의 경우에는 한 세대에 많은 개체를 유지함으로 인해서 근사해로의 수렴속도가 빠른 반면에 근사해 주변에서의 해 탐색은 적용확률이 매우 낮은 돌연변이 연산자를 가지고 수행함으로 해 탐색이 매우 느려지는데 이는 GA가 Hill-Climbing 능력이 부족하기 때문이다. 하지만 SA의 경우에는 근사해로의 수렴과정은 매우 느리지만 근사해 주변에서의 해 탐색이 매우 뛰어나기 때문에 우수한 근사해를 찾아낼 수가 있다. 이러한 두 방법의 단점을 보완하고 장점을 더욱 강화하기 위해서 GA와 SA를 결합하는 Hybrid GA 기법을 제안한다.

Hybrid GA 기법은 초기에 근사해까지는 수렴속도가 빠른 GA를 이용해서 수행하고 근사해 이후의 해 탐색은 SA를 이용해서 수행하는 방법이다. 여기서 주의할 점은 어느 시점에서 GA과정을 종결하고 SA기법을 적용할 것인가와 SA의 온도 파라미터 T를 어느 정도로 설정할 것인가 하는 문제가 있다. 본 연구에서는 GA의 종결과정을 미리 정한 반복횟수동안 해의 개선이 없을 때, 또는, 정해진 일정한 반복횟수까지 반복 수행한 이후로 설정하였다. 이렇게 함으로써 불필요하게 낮은 돌연변이 확률 때문에 생기는 소비적인 계산을 피할 수 있다. 그리고 SA단계에서의 온도 파라미터 T를 어느 정도로 설정할 것인가 하는 문제는 만약 초기에 T를 높게 설정을 하게 되면 좋지 않은 해를 받아들일 확률이 높기 때문에 SA만을 이용하는 기법과 별 차이가 없어지게 된다. 따라서 T는 초기에 낮게 설정해 주어야 한다. 하지만 T를 너무 낮게 설정해 주게 되면 반복적 개선법과 동일하게 됨으로써 지역 최소점에서 벗어날 수 없게 된다. 따라서 T를 주의깊게 설정해 주어야 한다.

3.2 설비배치 알고리즘

앞에서 제시한 설비배치문제를 효율적으로 풀기 위한 설비배치 알고리즘을 제시하면 다음과 같다.

(알고리즘 3.2)

단계 1 : 각 설비간의 흐름량, 거리, 면적, 가로 및 세로의 크기 등을 입력한다.

단계 2 : 임의의 초기배치순서를 결정한다.

단계 3 : 단계 2에서 결정된 배치순서를 이용하여 Genetic Algorithm을 수행하여 새로운 배치 순서를 생성하고, 평가함수값을 구한다.

단계 4 : 종료조건을 검사하고 만족하면 단계 5로 가고, 아니면 단계 3으로 가서 Genetic Algorithm을 다시 수행한다.

단계 5 : Simulated Annealing을 수행하고 새로운 배치순서를 생성한다.

단계 6 : 종료조건을 만족하면 최종배치안을 생성하고, 아니면 단계 5로 가서 Simulated Annealing을 다시 수행한다.

이와 같은 일련의 알고리즘 수행과정을 정리하면 Fig. 3.1과 같다.

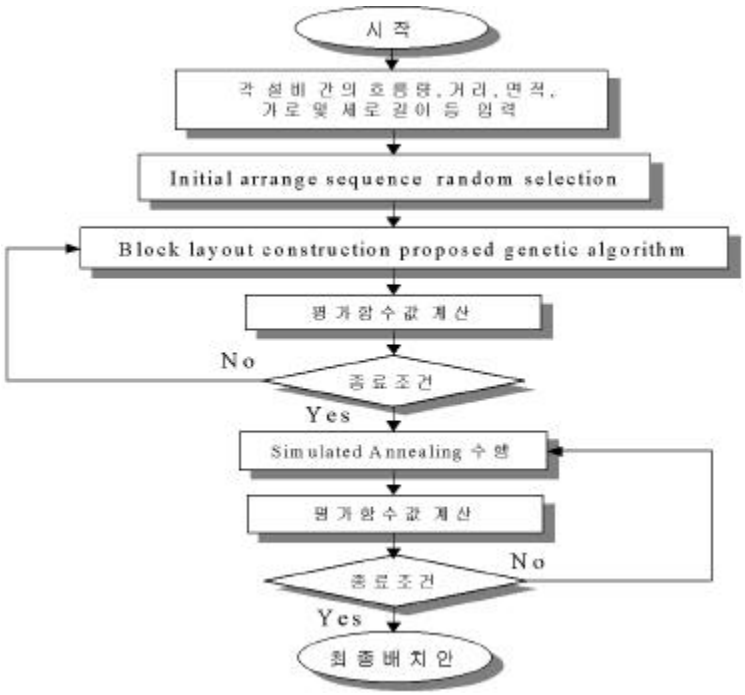


Fig. 3.1 Hybrid GA를 이용한 설비배치알고리즘의 수행 절차

3.3 설비배치문제의 유전자 알고리즘 (Genetic Algorithm)

인공 유전 시스템인 유전자 알고리즘(Genetic Algorithm, GA)은 자연세계의 진화 현상에 기초한 계산 모델로서 전역적 병렬 탐색의 최적화 알고리즘이다. GA는 자연선택(natural selection)의 원리와 자연계의 생물유전학에 기본 이론을 두고 있는 병렬적이며 전역적인 탐색알고리즘으로서, 모든 생물은 주어진 환경 속에 적응함으로써 살아남는다는 다윈의 적자생존(survival of the fittest)의 이론을 기본개념으로 하고 있다. 이를 구현하기 위해 GA는 풀고자하는 문제에 대한 가능 해들을 정해진 형태의 자료구조로 표현한 다음, 이들을 점차적으로 변형함으로써 점점 더 좋은 해들을 만들어 내게 된다.

자연계의 모든 생물들은 유전인자인 염색체에 의해서 생식과 유전 등 진화과정을 반복하여 발전해 간다. GA에서는 생물학적인 유전인자인 염색체(chromosome)에 해당하는 문자열을 가지고 생물과 같은 재생산(reproduction), 교배(crossover), 돌연변이(mutation)를 거쳐서 다음 세대의 새로운 자손을 인공적으로 만들게 되며, 이들 세 가지 연산자들을 반복적으로 적용하여 적합도가 높은 해를 탐색할 수 있게 된다.

GA가 기존의 탐색 또는 최적화 방법과 다른 것은 첫째, GA는 매개변수 그 자체를 사용하는 것이 아니라 매개변수집합을 부호화(coding)해서 사용하며, 둘째, 탐색공간에서 점(point)이 아닌 점들의 집단에 기반한 탐색방법으로써 경사값이나 기타 보조정보가 아닌 단지 적합도 함수(fitness function) 정보만을 사용한다. 셋째, GA는 확정론적인 전이규칙이 아닌 확률론적인 규칙을 사용한다. 따라서, GA는 미분가능성이나 연속성 그리고 볼록성 등과 같은 최적화 함수 정보를 필요로 하지 않을뿐더러 병렬 탐색으로 인해 다수의 봉우리(multimodal)로 이루어진 탐색공간에서 전역 최적해를 찾을 가능성이 매우 높은 것으로 연구·보고 되고 있다.

GA의 이와 같은 특징은 여러 형태의 공학 최적설계문제들에 대해 매우 큰 범용성을 지니게 하는 주요인으로 작용하는 것이다. 일반적인 확정론적 계산-기반 방법들은 특정 설계영역에 한정하여 매우 큰 효율성을 보일 수는 있지만 그 특정 영역밖에 있

는 문제들에 대해선 매우 낮은 효율성을 보이는 단점이 있으며, 무작위 탐색 방법들 역시 설계의 폭넓은 영역에 걸쳐 고른 효율성을 보이기는 하지만 효율성 자체가 매우 낮은 취약점이 있다. 하지만 GA는 설계영역 전반에 걸쳐 고르고 높은 효율성을 보여주고 있어서 범용성과 유용성이 매우 큼을 알 수 있다.

유전자 알고리즘의 기본 내용을 그림으로 표현하면 다음의 Fig. 3.2와 같다.

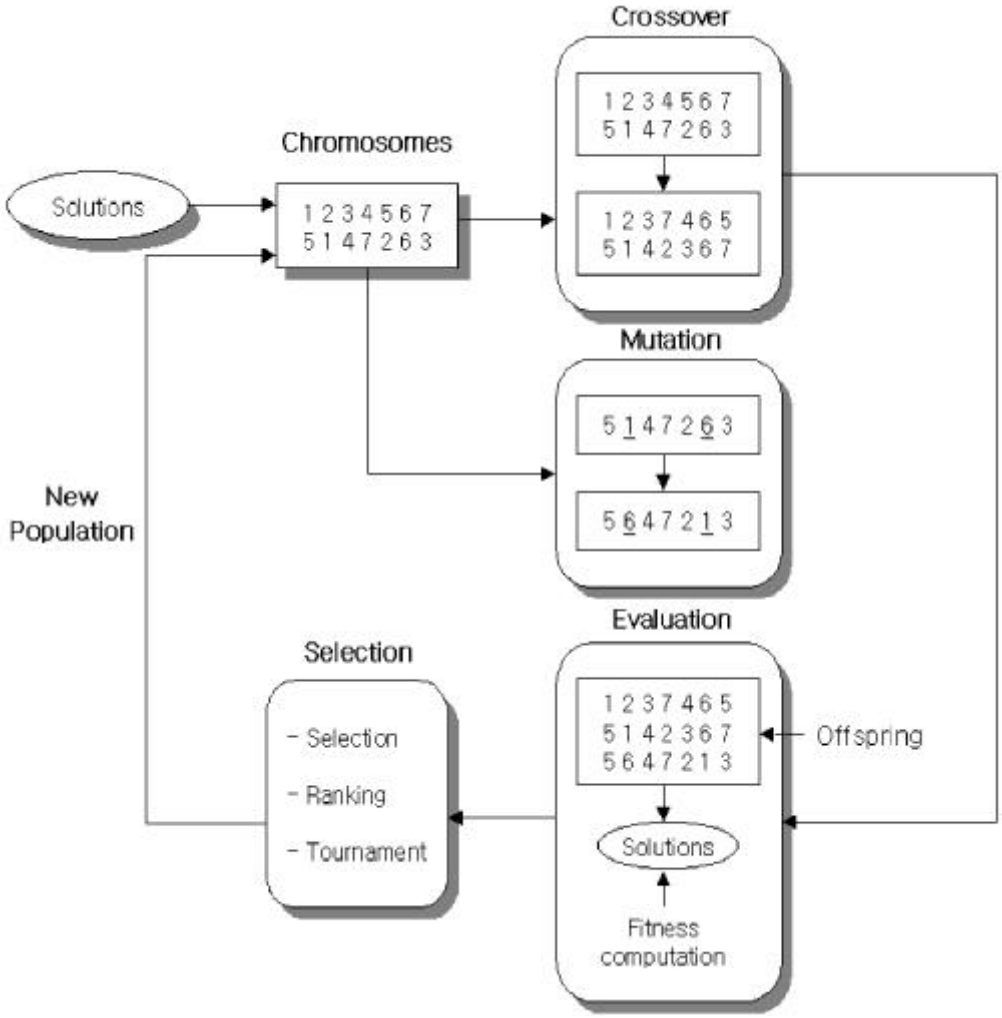


Fig. 3.2 유전자 알고리즘의 기본 내용

3.3.1 초기해 생성

모든 진화연산알고리즘에서는 문제의 가능한 해를 직접적으로 탐색하지 않고 염색체의 형태로 코딩을 하여 이 코딩된 개체에 대하여 유전 연산자를 적용시켜서 탐색을 하게 된다. 주어진 문제를 어떤 식으로 코딩하느냐에 따라서 해 공간의 형태가 바뀌고 적용시킬 수 있는 유전 연산자의 종류가 달라지므로 효율적인 탐색을 위해서는 개체의 표현 방법이 중요하다.

일반적인 유전자 알고리즘에서 초기 해집단의 각각의 멤버는 문제의 변수들을 부호화한 길이가 L인 이진 문자열(bit string)이다. 대부분의 경우에는 초기 해집단은 임의로 생성된다. 초기 해집단이 생성된 이후에, 각각의 문자열은 평가되어지고 적합도(fitness value)를 할당받게 된다.

본 연구에서 배치의 해는 string으로 표현하며, string 내의 번호는 각 설비의 번호이며 설비들의 배치순서에 따라 표기된다. 따라서, 이를 구하기 위해서 이진 문자열구조를 가질 경우에는 2진수 값을 10진수로 변환하는 계산과정이 추가되어야 하고 또한, 문제의 규모가 커질 경우 문자열의 길이가 길어져 많은 기억공간을 차지하게 된다. 따라서, 본 논문에서는 십진수로 string을 표현하였다. 십진수의 string을 표현할 때 주의해야할 점은 string을 임의로 생성시킬 때 각 설비의 번호가 중복되지 않게 생성시키는 것이다.

Figure 3.3 해의 정수 문자열 구조

3.3.2 적합도 평가함수

최선의 해를 탐색하는 기준이 되는 평가함수는 구해진 배치안에서 각 설비들 간의 거리와 물동량 행렬로부터 얻은 물동량을 곱하고 이에 단위거리당 이동비용을 곱한 값으로 표현한다.

$$Min \quad F(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} C_t \cdot a_{ij} \cdot d_{ij}, \qquad i, j = 1, 2, 3, \cdots, n$$

a_{ij} = 설비 i 와 j 사이의 흐름량
C_t = 단위 물동량을 단위거리 이동하는데 드는 비용
d_{ij} = 설비 i 와 j 사이의 거리

본 논문에서는 $C_t = 1$로 놓고, a_{ij}와 d_{ij}의 곱인 설비의 총이동거리에 중점을 두기로 한다. 즉, 물동량과 각 설비(부서)간 거리 곱의 총합을 최소화하는 설비들의 배열을 결정하는 문제로 한다. 각 설비간 물자의 흐름량은 고정되어 있고 설비간 거리는 배치의 순서에 따라 다르므로 물동량이 큰 설비들은 가깝게 위치시켜야 한다.

3.3.3 유전 연산자

유전자 알고리즘에 사용되는 파라미터로는 모집단의 크기, 교차율, 돌연변이율, 종료조건 등이 있다. 유전자 알고리즘의 파라미터의 값에 관한 연구들은 상당히 많이 이루어졌으나 아직까지 실험을 통하여 문제에 적합한 값들을 결정해야하는 기술적인 문제로 남아 있다.

일반적으로 유전자 알고리즘에 적용되는 유전 연산자로는 선택(selection), 교차(crossover), 돌연변이(mutation)가 있다.

(1) **교차**(crossover) **연산자**

교차(crossover)**는 두 부모가 갖는 유전자를 조합하여 자손을 생산하는 과정으로 좋은 해를 이용하는 역할과 해의 탐색 범위를 가장 크게 넓히는 연산자 역할을 한다.**

교차는 2개의 개체 간에 염색체를 부분적으로 서로 바꿈으로써 새로운 개체를 생성하는 것이다. 이 때 부모의 형질이 자손에게 적절히 계승되어야 하며 이것을 형질 유전성(character preservingness) **또는 교배 근접**(crossover neighborhood)**이라 한다.**

① **부분일치교배**(PMX : Partially matched crossover)

보통 TSP **문제에서** A**와** B**의 두 개의 염색체를 교배할 경우** 2**개의 교배점**(cut-point)**을 잡고 그 사이의 중간부분을 일치시켜 교환한 후 나머지 부분의 그 결과에 따라 중복되는 부분을 피하여 조정한다.**

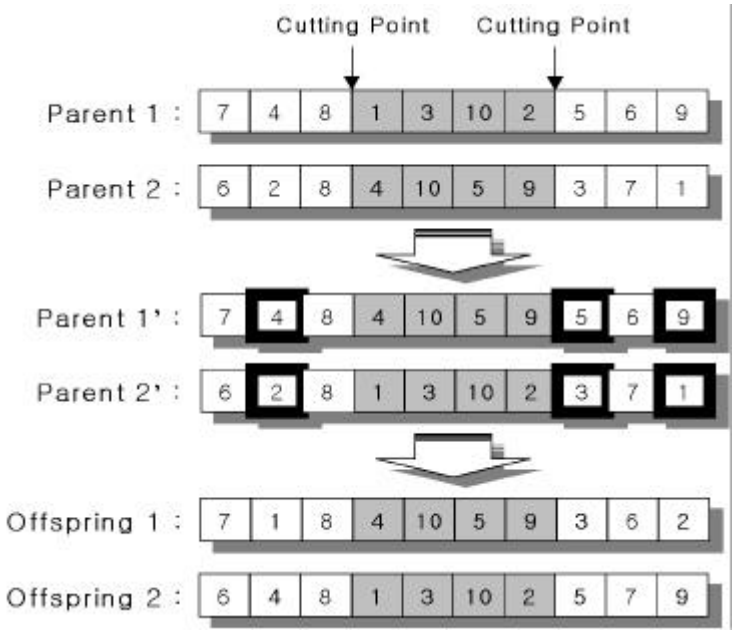


Fig. 3.4 **부분일치교배**(PMX)

② **순서교배**(OX : ordered crossover)

임의의 두 절단점을 지정하여 한 부모의 중간부분은 그대로 자손에 상속하고 다른 한 부모에서는 부모의 중간부분에 있는 인자들을 제외하고 상대적 순서를 보존하여 한 자손을 생산하는 교차이다.

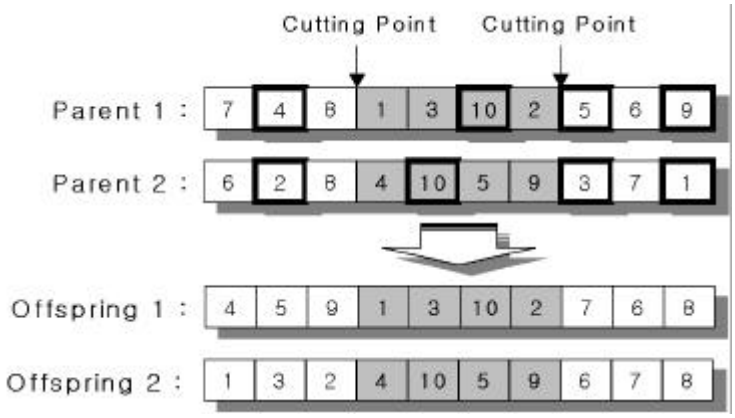


Fig. 3.5 **순서교배**(OX)

③ **주기교배**(CX : cycle crossover)

두 부모의 같은 위치에 있는 두 인자 중 하나를 선택하여 자손의 그 위치에 상속하는 방법이다.

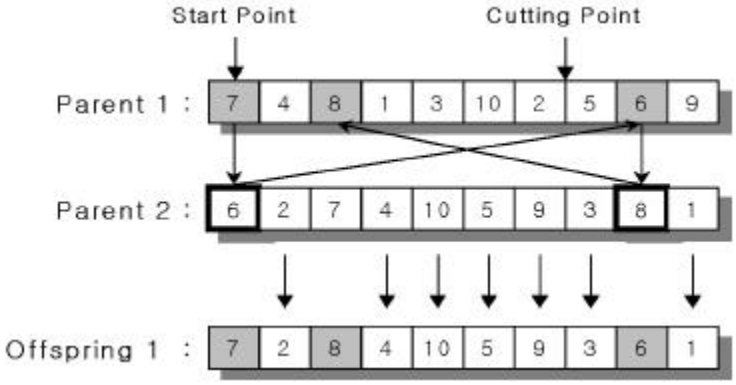


Fig. 3.6 **주기교배**(CX)

(2) **돌연변이**(mutation) **연산자**

교배 과정은 집단의 개체들이 현재 가지고 있는 정보들을 이용해서 결합하고 탐색하는 것에 대해 돌연변이 과정은 현재 집단에 존재하지 않는 새로운 정보를 제공해 줄 수 있다. 이 과정은 과거에 잃어버렸던 중요한 정보를 되찾을 수도 있으며 탐색의 방향이 국부해로 향할 경우 여기서 벗어날 수 있는 가능성을 부여할 수도 있다.

자연계와 마찬가지로 돌연변이는 유전자 알고리즘에서 아주 드물게 일어나는 부가적인 역할을 하며 재생산 및 교배 과정과 함께 사용될 때 전역적 탐색 성능을 더욱 향상시킨다.

돌연변이 연산자는 기존의 연구에서 뛰어난 성능이 입증된 Inversion, Insertion, Swap **기법들을 적용하였다.**

① Inversion

두 절단점(cutting point) **내의 순서를 역전시키는 방법으로 배열의 번호가 서로 역전되므로 많은 해의 탐색이 가능하다.**

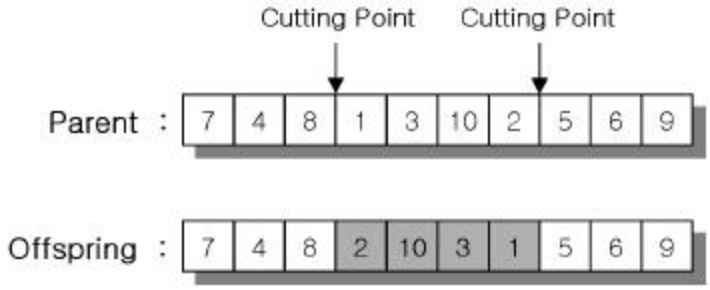


Fig. 3.7 Inversion

② Insertion

임의로 선택된 하나의 번호를 string**에 삽입하는 방법으로 선택된 번호가 삽입되**

면 차례대로 인접한 번호의 배열이동이 이루어지게 된다.

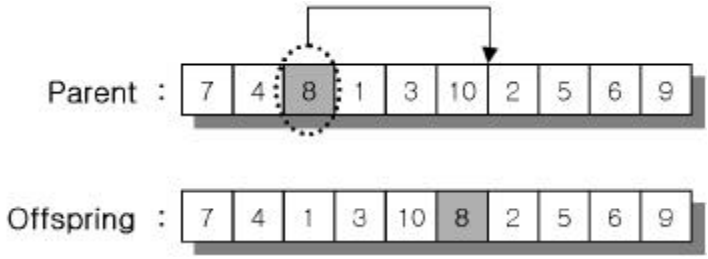


Fig. 3.8 Insertion

③ Swap

임의로 선택된 두 개의 번호를 서로 교환하는 방법이다. 이 방법은 하나의 설비의 배치순서를 교체함으로써 지역해를 벗어날 수 있는 효과적인 방법이다.

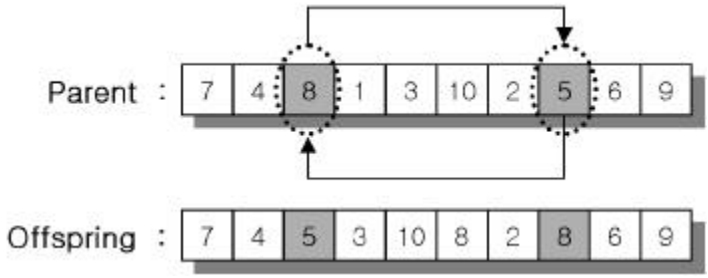


Fig. 3.9 Swap

(3) 선택(selection) 연산자

선택 연산자는 잘 적응한 해들은 살아남고 잘 적응하지 못한 해들은 도태되도록 유도함으로서 자연 선택(natural selection)현상을 모델링한다. 본 연구에서는 비교적 단순하고 많은 유전자 알고리즘에서 사용되는 순위 선택법(ranking selection)과 토너먼트(tournament)선택법을 이용하고 있다.

① 순위 선택법(ranking selection)

순위 선택법은 적합도의 크기 순서에 따라 순위를 매긴 후 순위에 따라 다음 세대에 자손을 남길 확률을 결정하는 방법이다.

순위선택법은 순위에 따라 복사될 개수가 결정되므로 선형식의 기울기가 크면 빠르게 수렴하고 기울기가 작으면 완만한 수렴이 된다. 따라서 순위선택법은 기울기를 어떻게 산정하느냐가 해의 탐색능력에 영향을 미치게 된다. 이러한 기울기에 영향을 미치는 것은 population size에 따라 결정되며 scaling 파라미터는 $q = 0.15$, $r = 0.01$로 설정하였다.

$$p(rank) = q - (rank - 1) \cdot r$$

여기서, $p(rank)$: 복사될 확률, $rank$: 순위, n : population size,

q : $\frac{1}{n}$ $\frac{2}{n}$ 사이의 값, r : 0 $\frac{2}{n(n-1)}$ 사이의 값

② 토너먼트법(tournament)

개체군 중에서 일정한 개수의 개체를 임의로 선택하여 그 중에 최고의 적합도를 가지는 개체를 다음 세대에 남기는 방법이다. 본 논문에서는 각 string의 적합도값을 2개씩 비교하여 이중에서 적합도값이 작은 것을 다음 세대에 2개를 복사하였다.

3.3.4 유전자 알고리즘의 종료조건

알고리즘의 수행을 종료하기 위한 조건은 새로운 세대를 생성해내는 전체 횟수를 제안하는 방법과 일정 횟수동안 해의 개선이 이루어지지 않으면 종료하는 방법, 계산 소요시간을 제한하는 방법 등이 널리 이용되고 있다[16].

본 연구에서는 Hybrid GA기법의 GA 단계의 종료조건으로 정해진 세대 교체횟수의 상한에 도달한 때와 일정기간동안 해의 개선이 관찰되지 않을 때까지로 종료 조건을 설정하였다.

3.4 설비배치문제의 시뮬레이티드 어닐링(Simulated Annealing)

시뮬레이티드 어닐링(Simulated Annealing, SA)은 어닐링이나 결정화 같은 자연계의 물리 현상을 모방한 알고리즘이다. 금속의 어닐링 과정이나 용액의 결정화 과정에서, 분자들은 충분한 에너지를 가지고 자유로운 운동을 하다가 충분히 느린 냉각 과정에서 에너지를 잃고 안정적인 상태가 된다. 이때 분자들의 이동공간을 해 공간으로, 에너지를 온도의 함수로 표현한 것이 시뮬레이티드 어닐링 알고리즘이다.

자연계에서 물질은 일반적으로 높은 에너지 상태에서 낮은 에너지 상태로 내려간다. 그러나 낮은 상태에서 높은 에너지 상태로 이동할 확률 역시 가지고 있다. 이 확률에 의하여 국부적 최저 에너지점에서 나와 최적점에 이를 수 있다. 이 확률은 온도 T에서의 볼츠만 확률분포 형태를 가진다.

높은 온도에서는 승인 확률이 높고 온도가 낮아질수록 승인 확률이 낮아진다. 그러나 아주 낮은 온도에서도 비록 작은 확률이지만 더 높은 에너지 상태로 이동할 확률을 가지므로 국부해에서 빠져나올 확률적 보장을 가진다.

자연계에서의 어닐링 과정이나 결정화 과정이 항상 최저 에너지상태에 도달하는 것은 아니다. 이 과정은 냉각 시간에 민감하게 영향을 받는다. 물질을 냉각하는 속도가 지나치게 빠르면 냉각 과정은 최저 에너지 상태를 찾지 못하고 국부해에 빠진채로 동결된다. 물리 현상만을 모방한 SA 역시 냉각 속도가 최적해를 찾는데 큰 영향을 미친다. 충분히 느린 냉각 속도의 조정이 중요하다. 이런 냉각 속도의 조정을 어닐링 일정(Annealing Schedule)이라고 한다.

SA는 기본적으로 Hill-Climbing이나 Steepest Descent의 변형된 형태이다. 즉, SA는 해 공간 내에서 새로운 해의 선택의 반복 과정으로 이루어진다. 새로운 해는 현재의 해에 일정거리 내에서 선택된다. 선택된 해는 Metropolis 시험을 통과하면 현재의 해로 바뀌고 테스트를 통과하지 못하면 현재의 해에 이웃한 다른 해를 선택하여 테스트를 반복한다.

Metropolis 시험은 낮은 에너지 상태에서 높은 에너지 상태로 이동하는 자연계의

물리 현상을 모방하기 위한 것이다. Hill Climbing이나 Steepest Descent 방법은 새로 선택된 해가 비용을 줄이는 방향으로의 선택일 때만 이동이 승인되지만 Metropolis 시험은 비용을 늘이는 방향으로의 이동도 승인한다.

1. 초기온도 T를 설정하고, 초기해 S를 생성한다.
2. 종료조건(stop criterion)을 만족할 때까지 다음 과정을 반복한다.
   (i) 내부루프기준(inner loop criterion)을 만족할 때까지 다음 과정을 반복한다.
   i. S로부터 이웃해 S
   ii. Δ = cost (S ') - cost (S)을 계산한다
   iii. 만약 $\Delta \leq 0$ (downhill move)이면, S = S ' 로 놓는다.
   iv. 만약 $\Delta \geq 0$ (uphill move)이면, 확률 $e^{(-\Delta E/kT)}$로 S = S ' 로 놓는다.
   (ii) 온도 T를 감소시킨다.
3. 최적해 S를 찾는다.

Fig. 3.10 시뮬레이티드 어닐링 과정

SA의 일반적인 문제 해결 과정은 문제에 적합한 목적함수와 변수를 설정하는 것으로 시작한다. 변수와 목적함수가 문제 설정에 맞지 않으면 계산이 비효율적이 됨은 물론 정확한 해를 구하지 못하게 되는 경우가 많다.

초기치를 입력받으면 초기치에 이웃한 해를 선택하여 주어진 제약조건에 적합한지의 여부를 시험한다. 제약조건에 맞는 새로운 해와 현재의 해를 이용하여 비용을 구하여 Metropolis 시험을 거친다. Metropolis 시험을 통과한 해를 현재의 해로 바꾸고 평형을 이룰 때까지 반복하여 해를 구한다.

SA법에서는 냉각 일정의 조정이 잘못 이루어지면 국부해에 빠질 위험이 있다. 최적해를 찾기 위해서는 충분히 느린 일정을 가져야 하지만, 지나치게 느린 냉각 일정은 해를 찾는 시간이 지나치게 요구됨으로 적당한 냉각 일정을 조정해야 한다.

또한, 초기 온도를 너무 낮게 설정하면 국부해에서 빠져 나오지 못하고 냉각되기 때문에, 충분한 초기 온도의 설정도 중요하다. 따라서 본 시스템에서 범용성을 높이기 위하여 적당한 초기 온도를 설정하도록 하였으며, 초기 냉각은 비교적 빠르게 그리고 냉각될수록 느린 냉각 속도를 가지게 하는 방식을 채택하였다.

본 연구에서는 Hybrid GA기법의 GA단계에서 찾아낸 가장 우수한 하나의 string을 가지고 SA단계를 수행한다.

3.4.1 초기 온도(T)의 설정

Hybrid GA기법에서 초기온도 T는 아주 중요한 역할을 한다. 초기온도의 값은 생성된 모든 전환을 채택할 수 있을 만큼 충분히 커야 한다. 초기온도를 높게 설정하면, 거의 모든 전환에 대해서 $\exp(-\Delta E/kT) \cong 1$ 의 확률을 가지기 때문에 지역 최소값에 빠지지는 않지만, 초기온도를 낮게 설정하는 것보다는 알고리즘 실행시간이 많이 소요된다. 하지만, 초기온도값을 낮게 설정하게 되면, 알고리즘 실행시간은 단축시킬 수 있지만, 지역 최소값에 빠지는 경우가 발생한다. 이런 문제점을 피하기 위해서 초기온도값을 높게 설정한다.

본 연구에서는 실험을 통하여 경험적으로 얻어진 초기온도(T)를 5로 설정하여 시뮬레이티드 어닐링 알고리즘을 실행시키고자 한다.

3.4.2 이웃해 생성(Perturbation)

Fig. 3.10의 단계 2(ⅰ) ⅰ에서 S'는 대개 모든 이웃들에 대해 똑같은 확률을 가지고 현재의 해에 대한 모든 이웃들 사이에서 무작위로 선택된다. 이웃해를 생성하는

방법은 가능한 한 간단한 방법을 이용하는 것이 좋은데 본 연구에서는 유전자 알고리즘의 돌연변이(mutation) 연산자를 이용한 이웃해 생성방법을 이용하였다. 이는 우수한 string 주변의 탐색을 정밀하게 하기 위해서이다.

3.4.3 내부루프의 종료조건

일반적인 내부루프 기준은 초기에 미리 정해지는 상수 L 횟수 동안 루프를 반복 수행하는 것이다. L의 값은 충분히 크게 설정되어야 하고, 그래서 온도가 감소되어지기 전에 균형상태가 이루어져야 한다. 일반적으로 이용되는 내부루프의 종료조건은 L이 이웃의 크기와 같거나 비례해서 취해지거나 초기에 미리 정해진 일정한 횟수동안 반복을 수행하는 것이다. 본 연구에서는 두 번째 방법을 이용하였다.

Fig. 3.10의 단계 2 (ii)에서 새로운 온도는 식 (3.1)에 따라서 대개 계산되며 본 연구에서도 이 방법을 이용하였다.

$$T' = r \cdot T \qquad (3.1)$$

여기서 T'는 새로운 온도이고, r 은 0.95 0.99 사이의 상수이다.

3.4.4 시뮬레이티드 어닐링의 종료조건

Fig. 3.10의 단계 2에 있는 반복의 종료조건은 다음과 같은 방법들을 생각해 볼 수 있다.

첫째, 최종온도를 고정시켜 놓고 어떤 한 온도가 최종온도에 도달했을 경우에 알고리즘 수행을 끝맺는 방법이다.

둘째, 각 온도에서 동일한 에너지(비용)의 값이 연속적으로 몇 번 발생하게 되면, 더 이상의 개선이 없는 것으로 보고 알고리즘 수행을 끝맺는 방법이다.

이와 같은 방법들에는 제각기 장·단점이 있다. 첫 번째 방법은 시간이 많이 소요

되는 단점이 있지만 전역 최적해에 근접할 수 있는 장점이 있고, 두 번째 방법은 각 온도의 어느 시점에서 발생할 수 있으므로 전역 최적해에 도달하기 전에 종료할 경우가 발생할 수 있다.

본 연구에서는 초기온도 T의 일정한 하한을 정해놓고 그 이하로 초기온도가 감소할 때를 종료조건으로 설정하였다.

3.5 설비배치문제의 Hybrid GA 적용 단계

3.5.1 유전자 알고리즘(Genetic Algorithm)의 과정

[단계 0] 초기값을 입력한다.
설비의 개수, 각 설비 간의 흐름량, 설비의 면적, 가로/세로의 비율의 상한치, 고정된 세로 및 가로의 길이, population의 크기, 교차가능률, 돌연변이값 등을 입력한다.

[단계 1] Population 크기의 초기 string을 생성한다.

[단계 2] 교차변이 확률에 따라 string에 대해 교차변이(crossover)를 수행한다.

[단계 3] 돌연변이 확률에 따라 돌연변이(mutation)을 수행한다.

[단계 4] 배치과정에 적용하여 적합도 평가를 한다.

[단계 5] Reproduction Rule을 적용한다.

[단계 6] 종료조건을 만족하면 [단계 7]로,
그렇지 않으면, [단계 3]으로 돌아가서 계산과정을 반복 수행한다.

[단계 7] 종료

3.5.2 시뮬레이티드 어닐링(Simulated Annealing)의 과정

[단계 1] GA 단계에서 가장 우수한 해 하나를 가져온다. 그리고, 초기온도 T와 내부루프의 반복횟수 L을 설정한다.

[단계 2] 종료조건을 만족할 때까지 다음 과정을 반복한다.

[단계 3] 내부루프기준(inner loop criterion)을 만족할 때까지 다음과정을 반복한다.

[단계 3-1] GA 과정에서 얻어진 가장 우수한 해의 이웃해를 생성한다.

[단계 3-2] 이웃해의 목적함수값과 GA 과정에서 얻어진 가장 우수한 해의 목적함수의 차를 계산한다.

[단계 3-3] 만약 목적함수의 차가 0 보다 작거나 같으면, 이웃해를 새로운 해로 받아들인다. 그렇지 않으면, 확률 $e^{(-\Delta/kT)}$ 로 이웃해를 새로운 해로 받아들인다.

[단계 4] 온도 T를 감소시킨다.

[단계 5] 종료조건을 만족하면 [단계 6]으로,
그렇지 않으면, [단계 3]으로 가서 계산 과정을 반복 수행한다.

[단계 6] 종료

Hybrid GA를 **이용한 설비배치알고리즘의** Flowchart**는 아래의** Fig. 3.11**과 같다**.

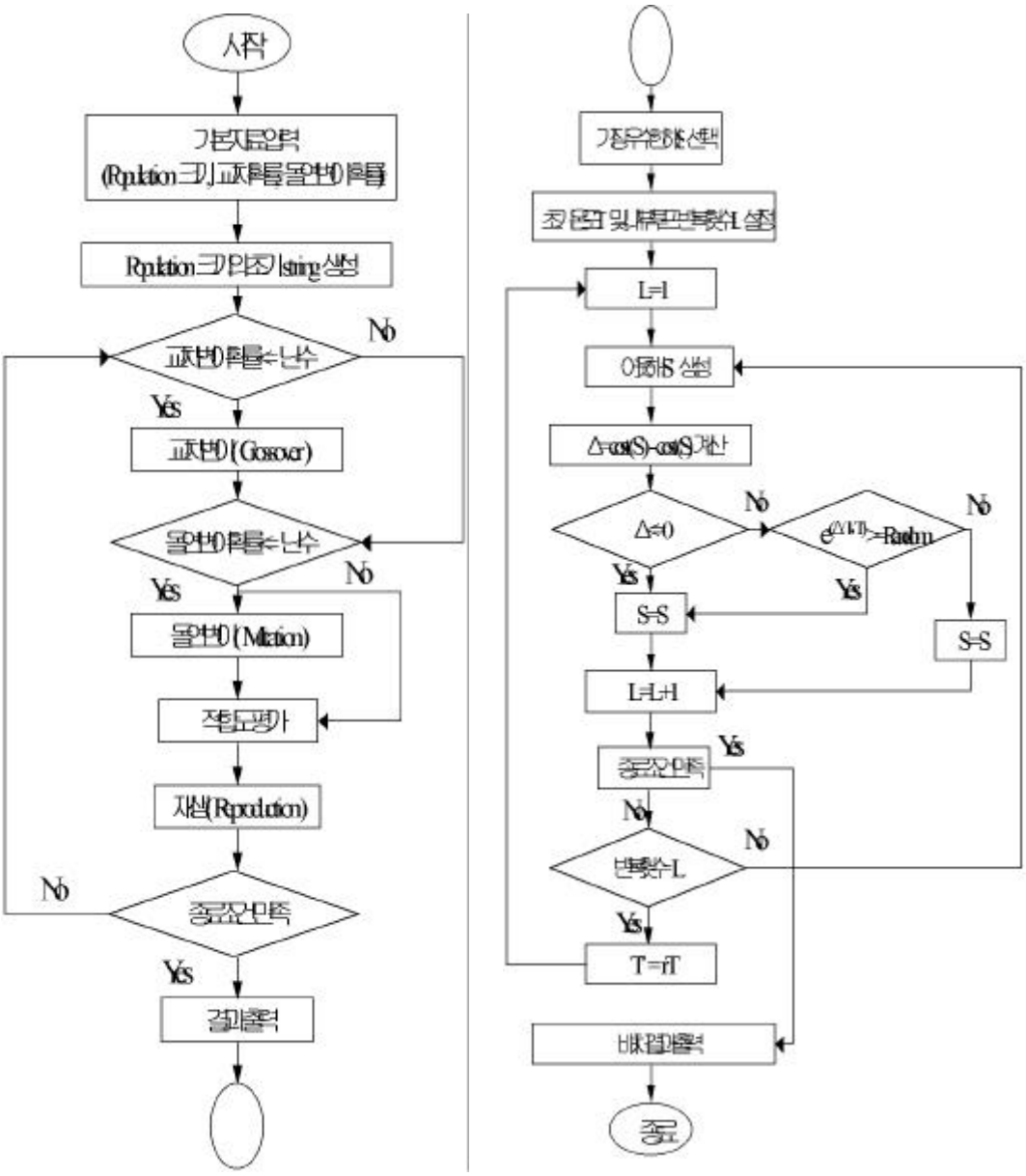


Fig. 3.11 Hybrid GA를 **이용한 설비배치알고리즘의** Flowchart

제 4 장 실험 및 평가

본 연구를 통하여 제시한 Hybrid GA를 이용한 설비배치 알고리즘의 유효성과 효율성을 검증하고자 설비의 수가 5, 6, 7, 8, 9, 10, 15, 20, 30인 경우에 대하여 실험을 하였고 아울러 기존의 Tam의 알고리즘과 우성식 등의 GA 알고리즘의 결과값과 비교·분석을 통하여 제안된 알고리즘의 우수성을 입증한다(실험에 관한 데이터는 부록과 같다[1][15][21]).

제안된 Hybrid GA 기법은 Visual C++로 개발되었고, 펜티엄Ⅱ 450Mhz, RAM 128 Mbyte, Windows 98 운영체제를 가진 PC를 이용하여 실험하였다.

4.1 Parameter 설정

4.1.1 실험을 통한 Genetic Algorithm의 Parameter 설정

모든 경우에 사용된 유전자 알고리즘 parameter 설정 방법은 다음과 같다. Population size는 10 100까지 10씩 증가시키면서 실험을 해 보았을 때 50까지는 좋은 해를 도출하였으나 50이상이 되었을 때에는 50일 때에 비해서 월등히 우수한 해를 도출하지 못하였고 50이후로는 거의 같은 해를 도출하여 Population size를 50으로 정하였다. Crossover rate와 Mutation rate는 0.0 1.0까지 0.05단위씩 증가시키면서 실험을 하였을 때 Crossover rate는 0.2 0.3 사이에서, Mutation rate는 0.1에서 우수한 해를 도출하였다. 가장 우수한 해를 도출하는 세 가지 parameter를 동시에 찾기 위해서 Population size를 50으로 고정시키고 문제 크기가 큰 $n = 20$일 때 Crossover Rate = 0.1 0.3, Mutation Rate = 0.1에서 Crossover Rate를 0.01씩 변화시키면서 실험을 해 본 결과, Population size = 50, Crossover Rate = 0.25, Mutation Rate = 0.1 일 때 가장 좋은 해를 도출하였다. 각 실험의 generation 수는 모두 동일하게 5000번

이었으며, 문제의 크기가 작은 경우는 2000번 미만에서도 근사해가 도출되었다.

4.1.2 실험을 통한 Simulated Annealing의 Parameter 설정

시뮬레이티드 어닐링 알고리즘을 실행하고자 할 때, 냉각 스케줄 등의 입력모수와 초기배열의 설정은 전역 최적해의 존재여부 및 해 탐색 수행시간에 영향을 미치는 중요한 요소가 된다.

각 20회씩의 실험을 통한 경험적 수치에 근거하여 초기온도(T)는 5로 설정하였고 내부루프의 종료조건은 500으로 하였다. 온도감소율(r)은 0.9, 최종온도는 0.5로 설정하였다.

또한, 시뮬레이티드 어닐링 알고리즘은 감소 알고리즘처럼 반복하는 것은 유사하지만 비용이 감소하는 것뿐만 아니라, 비용이 증가하는 이웃해로의 이동도 때때로 허용함으로써 지역 최소값에서 벗어나 전역 최적해를 찾아가는 기법이다. 이때, 비용이 증가하는 경우, 새로운 이웃해를 받아들일 확률은 볼쯔만 인자(Boltzmann factor)인 $\exp(-\frac{\Delta E}{K_B \cdot T})$의 확률로 허용하고 온도에 관계있는 목적함수의 변화에 크게 달려있다. 본 연구에 있어서는 목적함수의 차(ΔE)가 크기 때문에, 실험을 통해 볼쯔만 상수 K_B의 적당한 값을 정할 필요가 있다.

설비개수 (n)	초기온도	온도감소율	볼쯔만 상수(K_B)	내부루프 반복횟수
15	5	0.9	80 90	500
20	5	0.9	150 160	500
30	5	0.9	300	500

Table 4.1 시뮬레이티드 어닐링의 Parameters

4.2 실험 결과 분석

설비배치의 모든 조합에 대한 전수검사가 가능한 5 10개 설비배치문제의 최적값과 개발된 알고리즘의 해를 비교하였다.

그 결과 Table 4.2에서 볼 수 있듯이 제안된 기법에 의한 해는 설비의 수가 5 10개인 경우 모두 최적해를 구할 수 있었다. 기존의 연구에서 최적해를 찾지 못한 경우에도 모두 최적해를 찾을 수 있었다. 설비의 수가 적은 경우에 최적해를 찾을 수 있으므로 설비의 수가 많은 경우에도 최적해이거나 최적해에 근사한 해를 찾을 수 있음을 유추할 수 있다.

설비개수 (n)	배치 방법	최적해	이전의 GA 방법 (Woo and Park)	Hybrid GA	최적해의 String	최적성
5	Type 1	512	512	512	1-2-3-4-5	100%
	Type 2	533	533	533	2-1-3-5-4	100%
6	Type 1	896	896	896	1-2-3-4-5-6	100%
	Type 2	884	884	884	3-2-5-1-4-6	100%
7	Type 1	1319	1319	1319	1-4-5-2-3-7-6	100%
	Type 2	1230	1230	1230	1-4-5-2-3-7-6	100%
8	Type 1	**1940**	**1965**	**1940**	1-2-5-8-7-4-6-3	100%
	Type 2	1806	1806	1806	5-1-2-6-4-8-7-3	100%
9	Type 1	2673	2673	2673	1-2-5-8-7-3-6-4-9	100%
	Type 2	2664	2664	2664	1-2-5-8-7-3-6-9-4	100%
10	Type 1	3709	3709	3709	1-2-5-8-7-3-6-10-9-4	100%
	Type 2	**3519**	**3591**	**3519**	10-9-1-8-7-3-4-6-2-5	100%

Table 4.2 설비개수가 5 10인 경우, Hybrid GA와 기존 알고리즘 간의 결과 비교

PC를 이용해서 최적해를 구하기 어려운 15 30개의 설비배치문제들에 대해 제안된 설비배치알고리즘의 실험결과는 Table 4.3과 같다. 실험 결과를 기존 연구와 비교하면, Tam의 알고리즘에 비해 17 35%의 수행도 개선효과를 가져왔고, Woo and Park의 GA 알고리즘에 비해서는 2 9%의 수행도 개선효과를 가져왔다.

또한, 각각 20회에 걸친 실험 결과값의 평균조차도 이전 연구에 비해 아주 우수한 것으로 나타났으며, 설비의 규모가 큰 경우일수록 더 우수한 근사해를 도출하는 것을 알 수 있었다. 이는 유전자 알고리즘은 많은 개체를 유지함으로서 근사해로 빨리 수렴하는데 세대가 거듭할수록 유전자 알고리즘은 Hill-Climbing 능력의 부족으로 인해 더욱 개선된 해를 찾지 못하지만 SA는 GA의 우수한 해를 받아 많은 실험을 통해 정해진 파라메타값을 가지고 수행함으로써 우수한 결과를 도출할 수 있었다.

설비 개수 (n)	배치 방법	Tam의 알고리즘에 의한 해	Woo & Park의 GA에 의한 해	Hybird GA에 의한 방법		Efficiency of Hybrid GA	
				최소값	평균값	Tam의 알고리즘	Woo & Park의 알고리즘
15	Type 1	13762	9120	8847	8911	35.71%	2.99%
	Type 2	12240	9855	9521	9582	22.21%	3.39%
20	Type 1	26921	21885	20029	20696	25.60%	8.48%
	Type 2	28646	22656	21698	21926	24.25%	4.23%
30	Type 1	55668	50492	46286	46545	18.06%	9.66%
	Type 2	58824	52884	48814	49324	17.02%	7.70%

Table 4.3 설배개수가 15, 20, 30인 경우, Hybrid GA와 기존 알고리즘 간의 결과비교

다음은 본 연구에서 제시한 설비배치알고리즘인 Hybrid GA를 통해서 생성된 설비배치 형태인 Type 1, Type 2의 각 설비의 수 $n = 15, 20, 30$ 일 때, GA단계와 SA단계의 수렴과정 및 결과와 그 최종설비배치안을 보여준다.

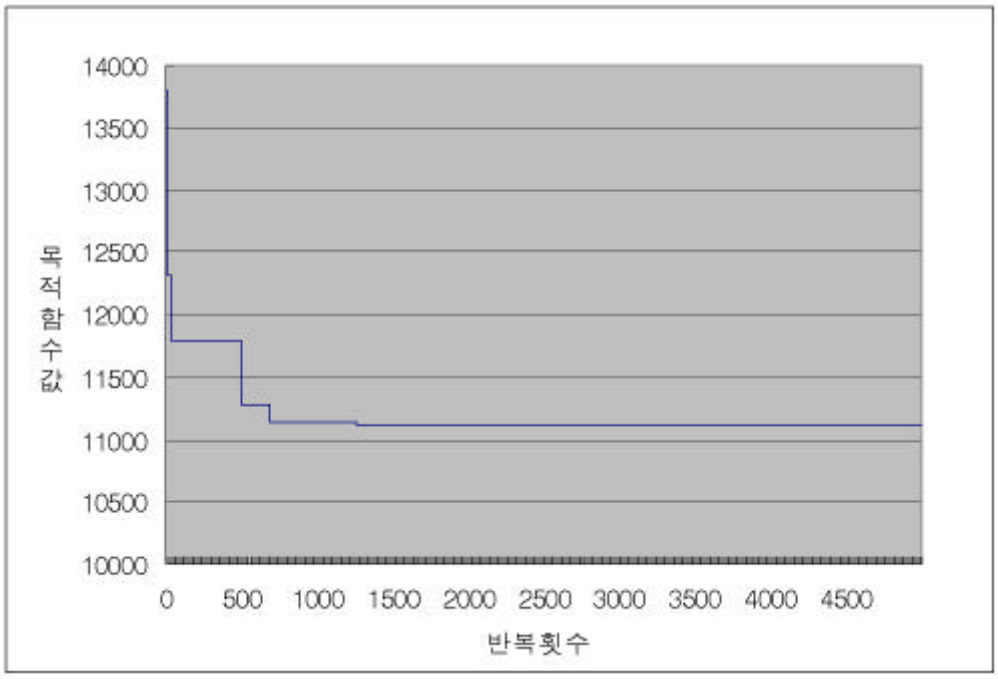


Fig. 4.1 **설비의 수** n=15**일 때**, Type 1**에 대한** Hybrid GA**의** GA**단계 수렴결과**

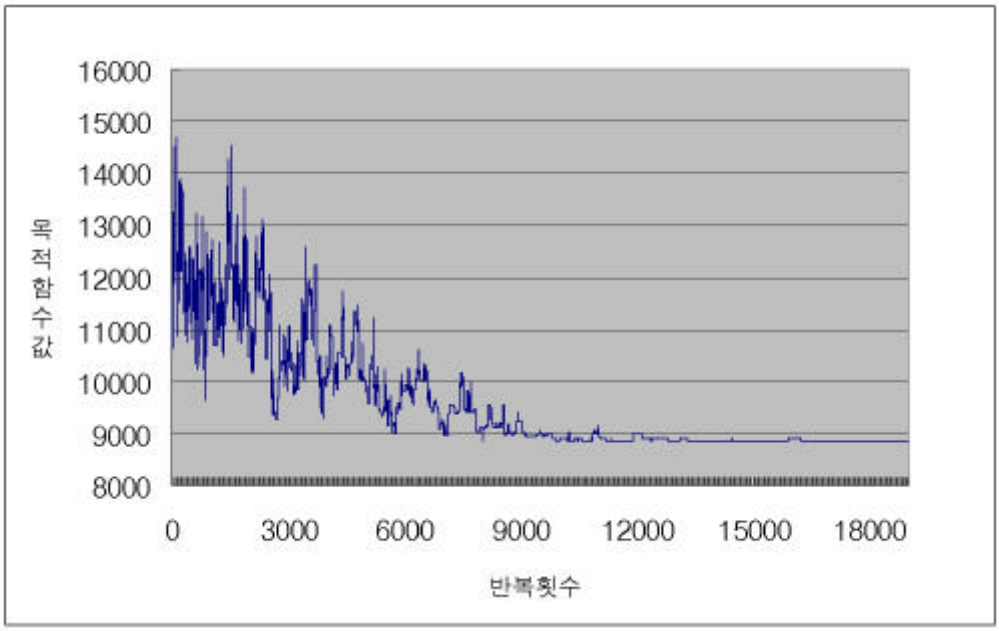


Fig. 4.2 **설비의 수** n=15**일 때**, Type 1**에 대한** Hybrid GA**의** SA**단계 수렴결과**

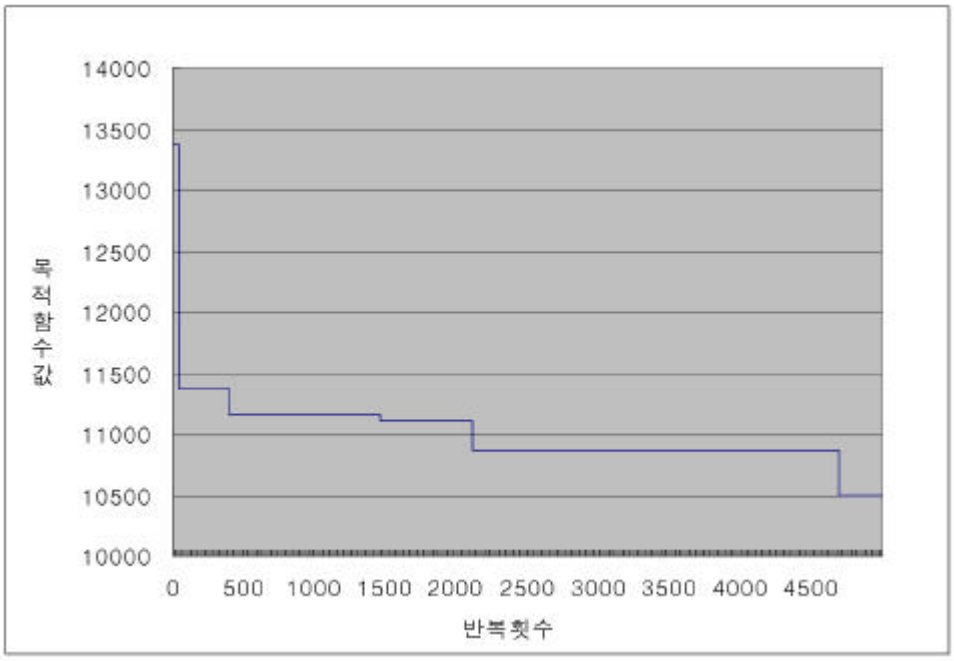


Fig. 4.3 **설비의 수** n=15**일 때**, Type 2**에 대한** Hybrid GA**의** GA**단계 수렴결과**

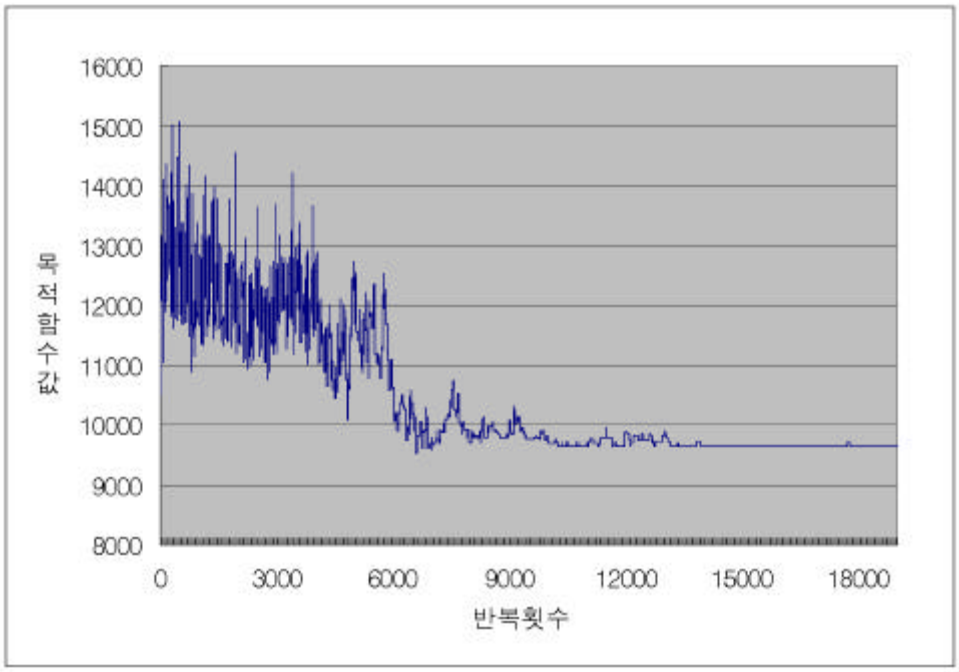


Fig. 4.4 **설비의 수** n=15**일 때**, Type 2**에 대한** Hybrid GA**의** GA**단계 수렴결과**

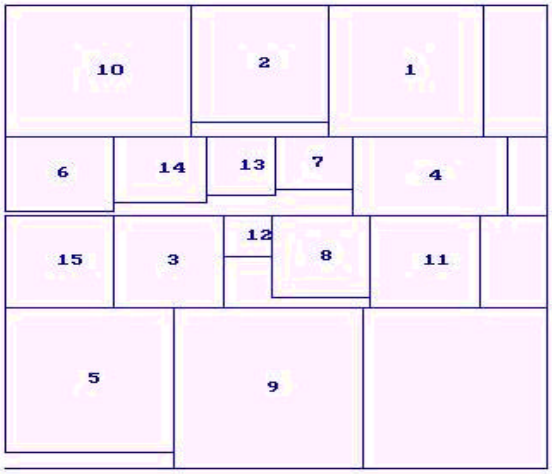


Fig. 4.5 **설비의 수** n=15 **에 대한** Type 1**의 배치 결과**

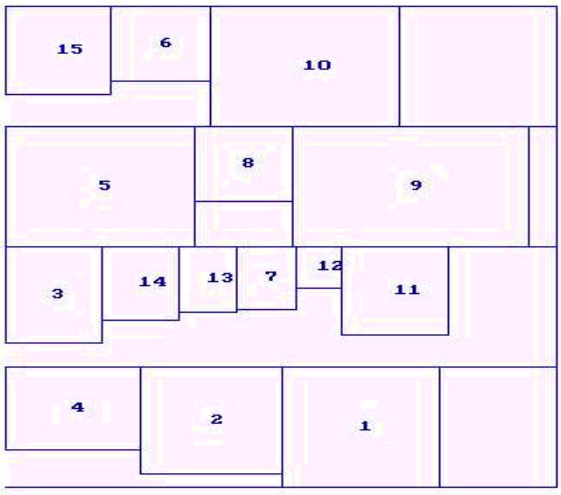


Fig. 4.6 **설비의 수** n=15 **에 대한** Type 2**의 배치 결과**

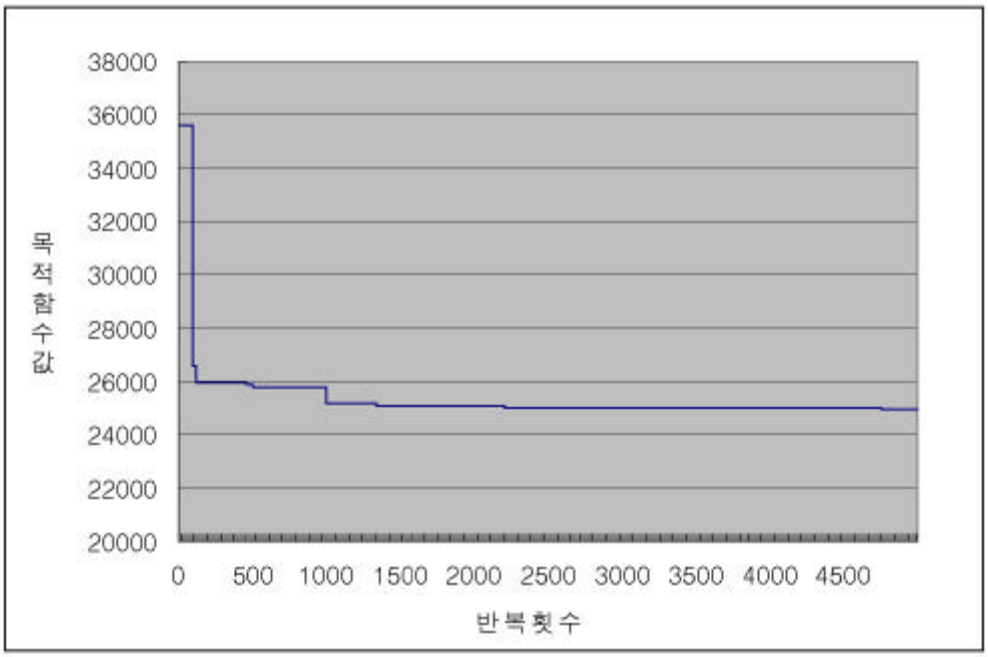


Fig. 4.7 **설비의 수** n=20**일 때**, Type 1**에 대한** Hybrid GA**의** GA**단계 수렴결과**

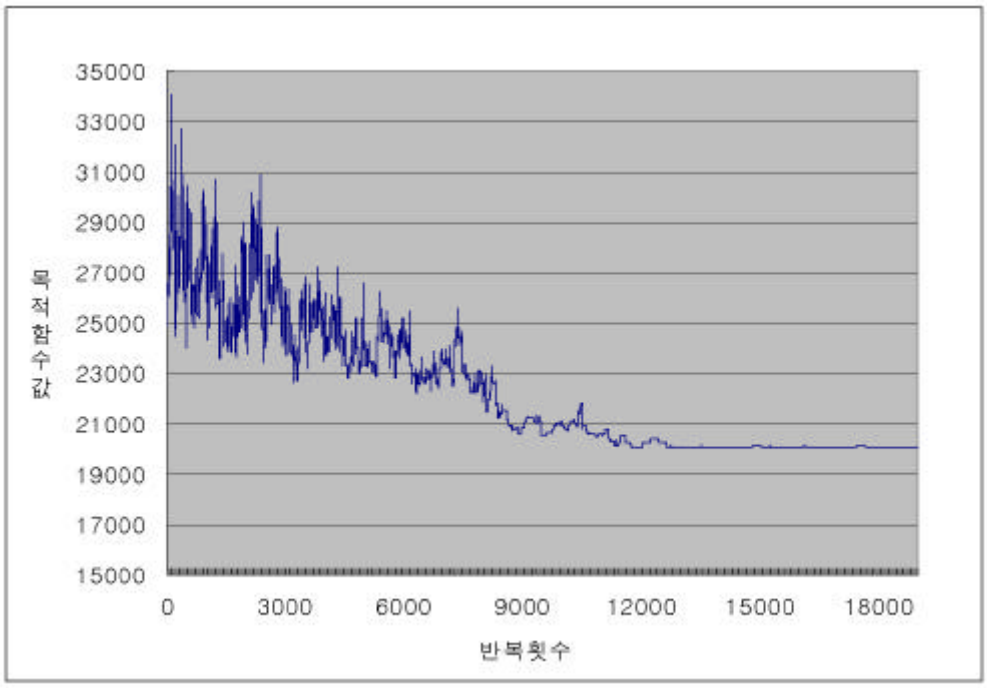


Fig. 4.8 **설비의 수** n=20**일 때**, Type 1**에 대한** Hybrid GA**의** SA**단계 수렴결과**

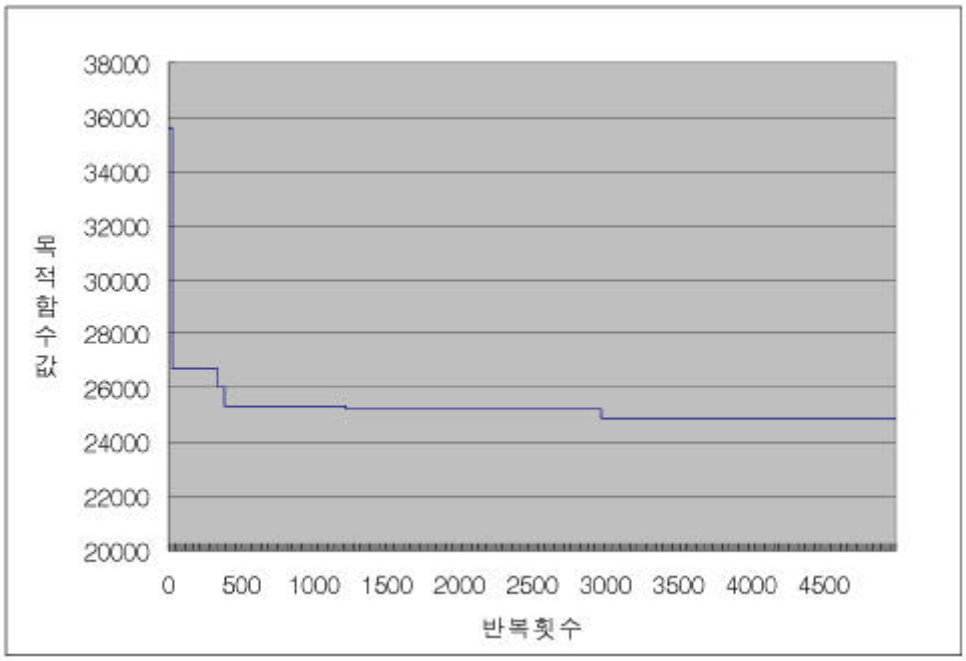


Fig. 4.9 **설비의 수** n=20**일 때**, Type 2**에 대한** Hybrid GA**의** GA**단계 수렴결과**

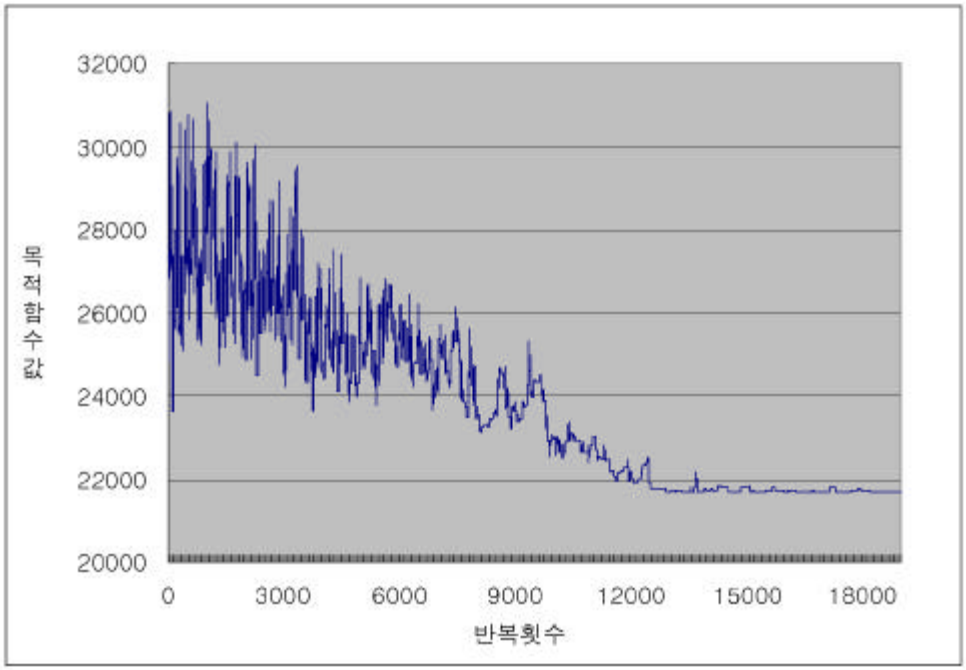


Fig. 4.10 **설비의 수** n=20**일 때**, Type 2**에 대한** Hybrid GA**의** SA**단계 수렴결과**

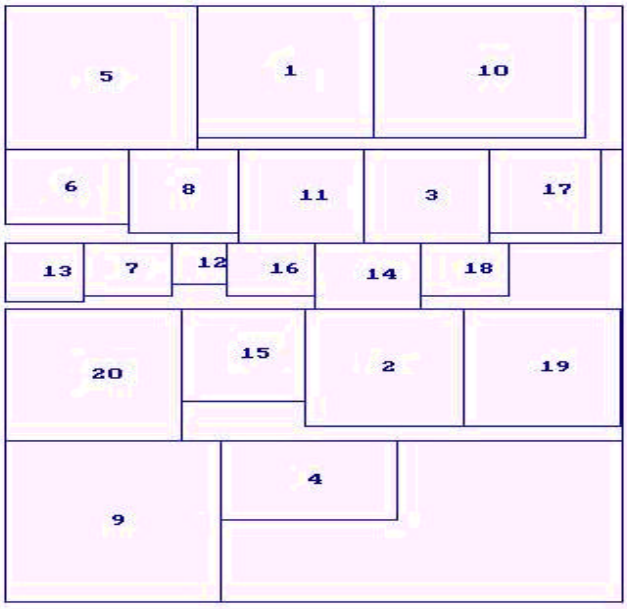


Fig. 4.11 **설비의 수** n=20 **에 대한** Type 1**의 배치 결과**

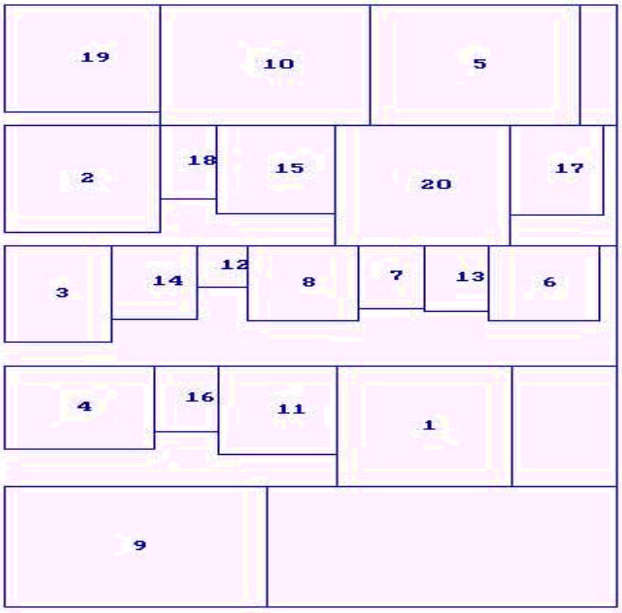


Fig. 4.12 **설비의 수** n=20 **에 대한** Type 2**의 배치 결과**

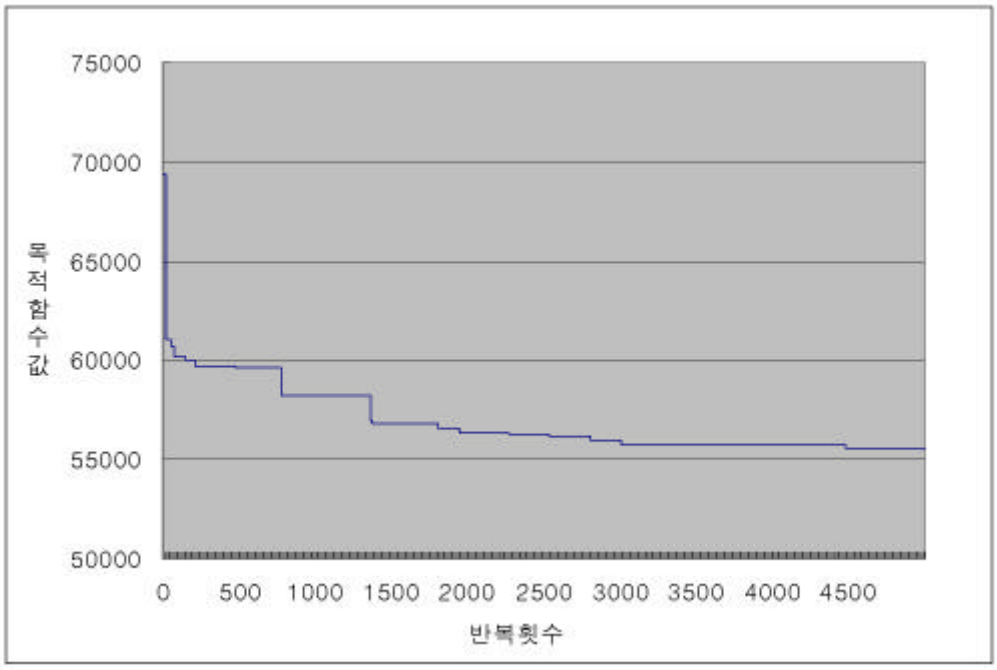


Fig. 4.13 **설비의 수** n=30**일 때**, Type 1**에 대한** Hybrid GA**의** GA**단계 수렴결과**

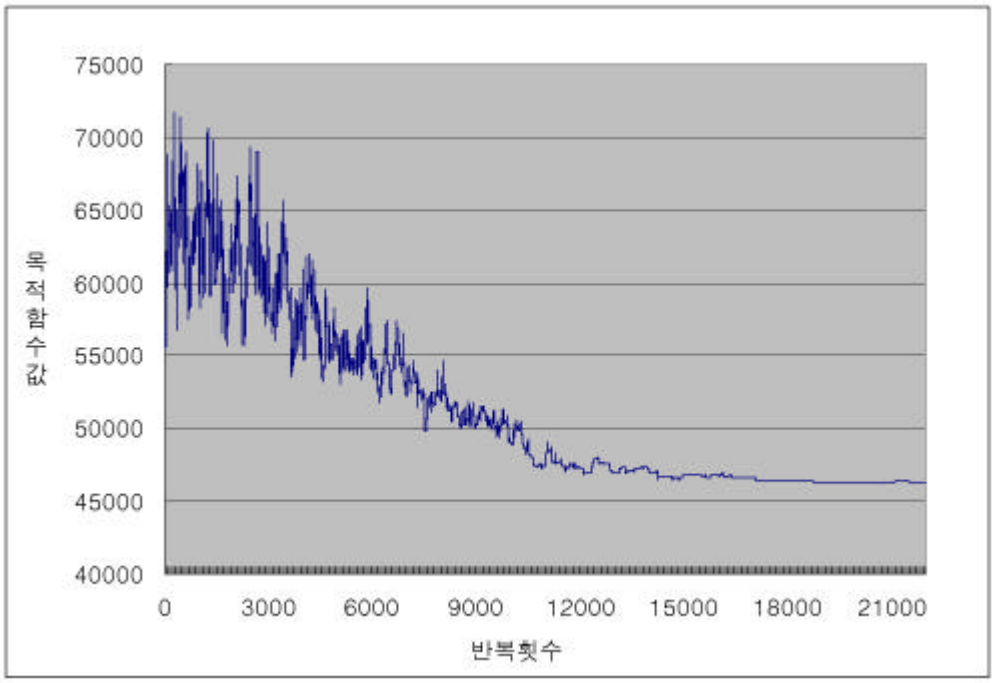


Fig. 4.14 **설비의 수** n=30**일 때**, Type 1**에 대한** Hybrid GA**의** SA**단계 수렴결과**

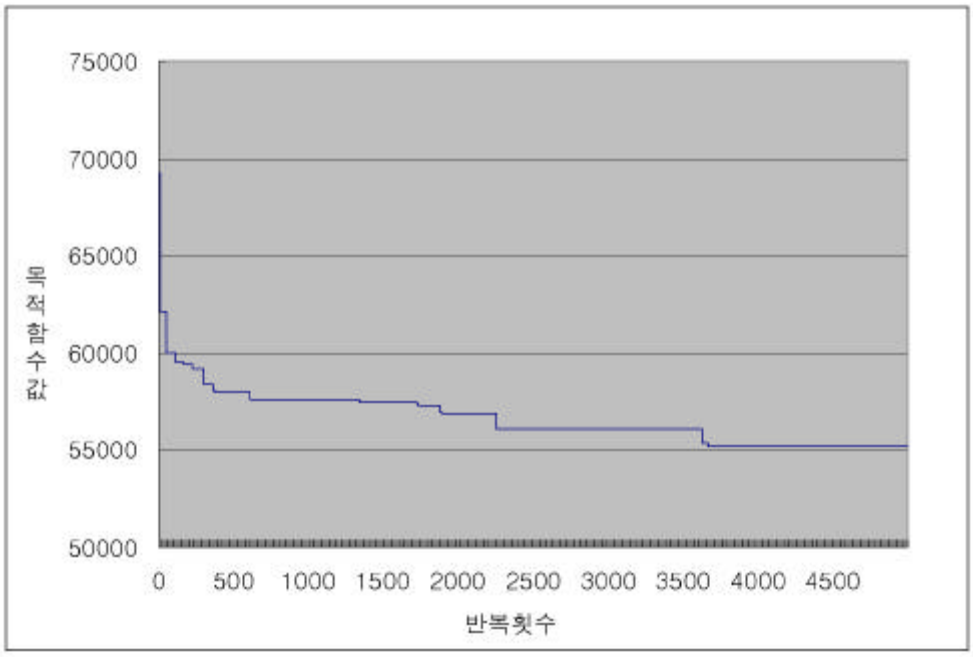


Fig. 4.15 **설비의 수** n=30**일 때**, Type 2**에 대한** Hybrid GA**의** GA**단계 수렴결과**

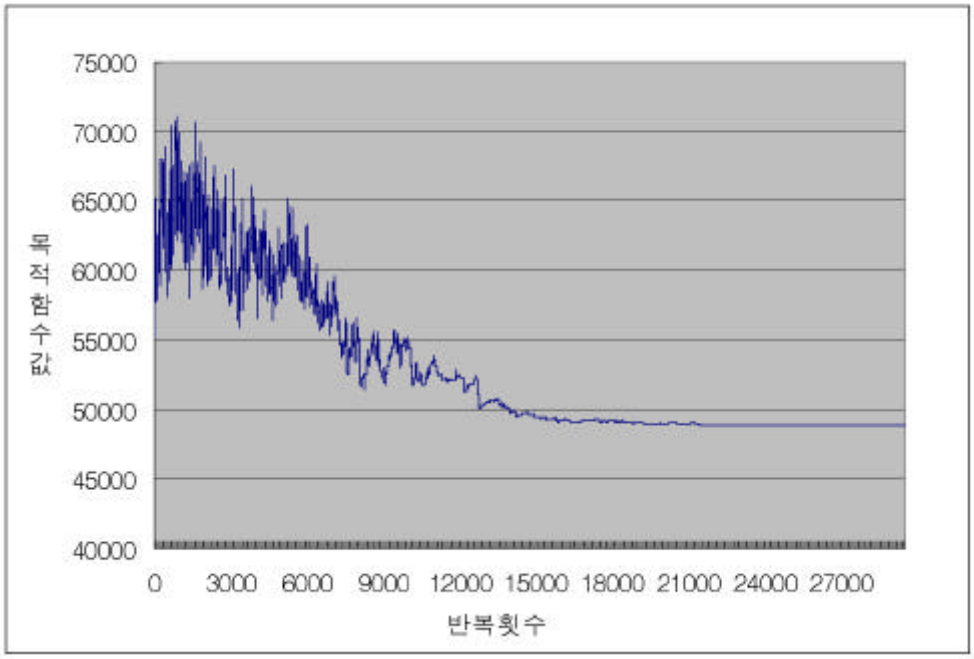


Fig. 4.16 **설비의 수** n=30**일 때**, Type 2**에 대한** Hybrid GA**의** SA**단계 수렴결과**

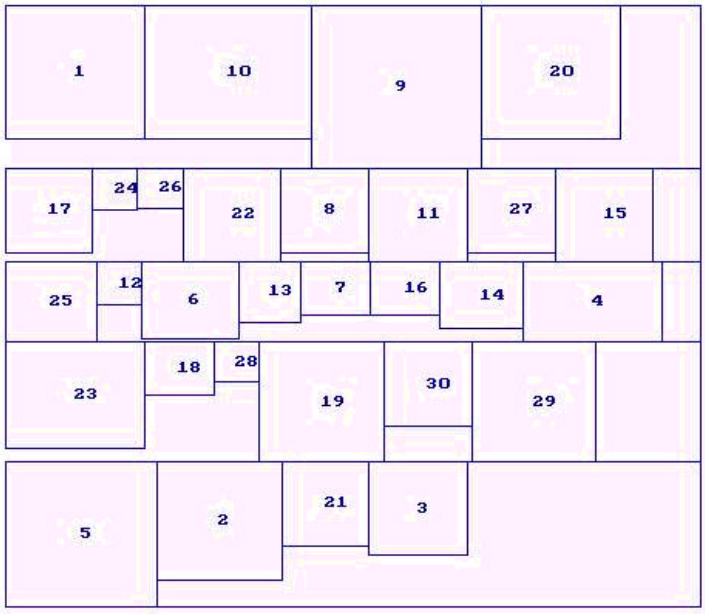


Fig. 4.17 **설비의 수** n=30 **에 대한** Type 1**의 배치 결과**

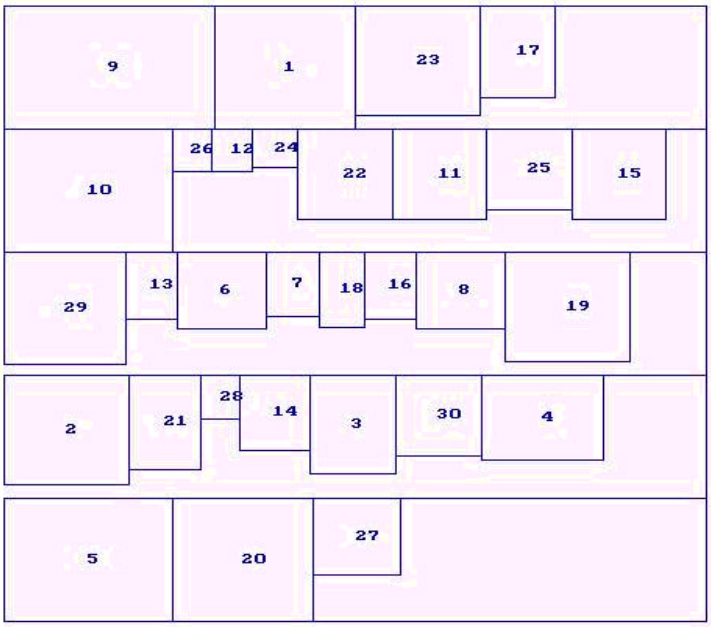


Fig. 4.18 **설비의 수** n=30 **에 대한** Type 2**의 배치 결과**

제 5 장 결론 및 추후 연구과제

시스템 설계의 초기 단계인 설비배치는 중장기 계획으로서 잘못된 설비배치는 막대한 자재 처리 비용과 재설치 비용이 소요되기 때문에 신중한 결정이 요구되는 분야이다.

본 논문에서는 설비배치문제를 해결하기 위해서 설계자가 배치할 전체 공간의 가로와 세로의 크기를 정해주면 블록단위의 수평방향으로 상한치까지 통로가 자동적으로 생성되도록 각 설비들을 효율적으로 배치하는 방법으로 Hybrid GA를 이용한 설비배치알고리즘을 개발하고 적용하였다.

규모가 큰 문제의 경우 GA를 이용하면 근사해로의 수렴속도는 빠른 반면에 근사해 주변에서의 해 탐색은 적용확률이 매우 낮은 돌연변이 연산자를 가지고 수행함으로써 해 탐색이 매우 느려지는데 이는 GA가 Hill-climbing 능력이 부족하기 때문이다. 그러나 SA의 경우에는 근사해로의 수렴과정은 매우 느리지만 초기배열상태가 전역 최적해에 가깝게 형성되면 근사해 주변에서의 해 탐색이 매우 뛰어나기 때문에 우수한 근사해를 찾아 낼 수 있다는 점에 착안하여 두 방법의 단점을 보완하고 장점을 더욱 강화하기 위해서 GA와 SA를 결합한 Hybrid GA 방법을 개발하였다.

본 연구에서의 제안된 기법인 Hybrid GA 기법과 기존 기법과의 결과를 비교·분석한 결과, Tam의 알고리즘에 비해 17 35%의 수행도 개선효과를 가져왔고, Woo and Park의 GA 알고리즘에 비해서는 2 9%의 수행도 개선효과를 가져왔다.

각각 20회에 걸친 실험 결과의 평균값조차도 이전 연구에 비해 우수한 것으로 나타났으며 설비의 규모가 큰 경우일수록 더 우수한 근사해를 도출하는 것을 알 수 있었다.

또한, 본 논문에서는 배치형태를 그래픽으로 구현하여 실제적인 배치 형태로 표현하여 각 설비들 간의 재조정과 보완 작업을 최소화하였다. 따라서 실질적인 배치형태

를 확인할 수 있어 문제점의 파악이 쉽고 적용하기가 편리하다.

추후 연구과제로는 수평 방향의 통로뿐만 아니라 수직 방향의 통로까지 고려하고 나아가서 특정 위치에 특정 설비를 배치하는 경우까지 고려한 알고리즘의 연구가 필요할 것이다.

또한, Hybrid GA기법의 일반성을 입증하기 위해서 보다 많은 조합최적화문제에 적용해보아야 할 것이고 이때 각 파라메타(parameter)의 설정에 대한 세밀한 연구가 필요할 것이다.

참고문헌

[1] 우성식, 박양병, 블록단위 설비배치를 위한 유전자 알고리즘의 적용, 경영과학, 제14권 1호, pp.67-76, 1997

[2] 김재곤, 이근철, 김영대, 그래프 이론을 이용한 설비배치 계획에 관한 연구, Journal of the Korean Institute of Industrial Engineers, Vol.23, No.2, June, pp.359-370, 1997

[3] 홍관수, 다수요인을 가진 설비배치문제를 위한 모형과 Simulated Annealing 알고리즘, 경영과학, 제20권 1호, pp.63-83, 1995

[4] 문기주, 현실제약조건을 이용한 설비배치안 개발 방법 연구, Journal of the Korean Institute of Industrial Engineers, Vol.19, No.1, Mar, pp.31-38, 1993

[5] 이문규, FLASOM-자기조직화 지도를 이용한 시설배치, Journal of the Korean Institute of Industrial Engineers, Vol.20, No.2, June, pp.65-76, 1994

[6] 고창성, 김홍배, 퍼지 집합 접근법에 의한 시설배치계획에 관한 연구, 산업공학, 제6권 1호, pp.67-86, 1993

[7] 박성준, 정의승, 일반 배치문제 해결을 위한 제약만족기법의 적용, Journal of the Korean Institute of Industrial Engineers, Vol.24, No.4, pp.539-550, 1998

[8] 장병탁, 인공진화에 의한 학습과 최적화, 뉴로 퍼지 제어 기술특집

[9] 김환성, 이상용, 인공지능 개념을 이용한 공장 설비배치 알고리즘 개발, 한국품질관리학회지, 제19권 1호, pp.151-162, 1991

[10] 홍성화, 노인규, 유연생산시스템에서 유전자 알고리즘을 이용한 단열 설비 배치에 관한 연구, 한양대학교 석사학위 논문

[11] 조규갑, 이병욱, 셀 제조시스템 설계를 위한 부품-기계 셀의 형성기법, 한국정밀공학회지, 제13권 4호, pp.143-150, 1996

[12] 이명수, 유전 알고리즘을 이용한 설비배치 설계에 관한 연구, 부산대학교 석사학위 논문

[13] **문기주 외 공역, 산업물류를 위한 설비계획, 사이텍미디어**, 1998

[14] **김여근, 윤복식, 이상복, 메타휴리스틱, 영지문화사**, 1997

[15] Tam. K. Y., "Genetic algorithms, function optimization, and facility layout, design", European Journal of Operations Research, Vol.63, pp.322-346, 1992

[16] Goldgerg, D. E., Genetic Algorithms in search, optimization, and machine learning, Addison Wesley, 1989

[17] Jasmit Singh Kochhar, Bryce T. Foster, Sunderesh S. Heragu, "A Genetic algorithm for the unqual area facility layout problem", Computers Operations Research, Vol.25, No.7/8, pp.583-594, 1998

[18] Dijin Gong, Genji Yamazaki, Mitsuo Gen, Weixuan Xu, "A genetic algorithm method for one-dimensional machine location problems", International Journal of Production Economics, 60-61, pp.337-342, 1999

[19] David M. Tate and Alice E. Smith, "Unequal-area facility layout by genetic search", IIE Transactions, Vol. 27, pp.465-472, 1995

[20] K.C.Chan, and H.Tansri, "A study of Genetic Crossoover operations on the facilities layout problem", Computers and Indurstrial Engineering, Vol. 26, pp.557-550, 1994

[21] Nugent C. E., Vollman, T.E, and Ruml. J., "An experimental comparison of techniques for the assignment of facilities to locations", Operations Research, Vol.16, pp.150-173, 1968

[22] Suresh G., Vinod V.V., and Sahu S., "A genetic algorithm for facility layout", International Journal of Production Research, Vol.33, No.12, pp.3411-3423, 1995

[23] Tomkins, James A., John A White, Yavuz A. Bozer, Edward H.Frazelle, J.M.A. Tanchoco, and Jaime Trevino, Facilities Planning 2nd Ed., John Wiley & Sons, Inc., 1996

[24] Heng Li, Peter E.D.Love, Genetic search for solving construction site-level unequal-area facility layout problems, Automation in Construction, Vol.9, pp.217-226, 2000

[25] Kenichiro Matsuzaki, Takashi Irohara, Kazuho Yoshimoto, Heuristic algorithm to solve the multi-floor layout problem with the consideration of elevator utilization, Computer & Industrial Engineering, Vol.36, pp.487-502, 1999

[26] Russell D.Meller, Venkat Narayanan, Pamela H.Vance, Optimal facility layout design, Operation Research Letters 23, pp.117-127, 1999

[27] R. Tavakkoli-Moghaddain and E.Shayan, Facilities layout design by genetic algorithms, Computers industry Engineering, Vol.35, Nos 3-4, pp.527-530, 1998

[28] Tate, D. M., and E. A. Smith, "Unequal Area Facility by Genetic Search", IIE Transactions, Vol. 27, pp.467-472, 1995

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

부 록

1. 각 설비들의 면적과 가로/세로의 비율

설비 번호	설비 면적	가로/세로 비율의 상한치	고정된 가로, 세로의 크기	
1	100	1.0	10.00	10.00
2	80	1.0	8.94	8.94
3	50	1.3	7.07	7.07
4	60	0.8	6.00	10.00
5	120	1.0	10.95	10.95
6	40	1.0	5.71	7.00
7	20	1.4	4.00	5.00
8	40	1.0	6.32	6.32
9	150	1.1	12.24	12.24
10	120	1.5	10.00	12.00
11	50	1.1	7.07	7.07
12	10	1.2	3.16	3.16
13	20	1.5	4.47	4.47
14	30	1.25	5.00	6.00
15	50	1.1	7.07	7.07
16	20	1.5	4.00	5.00
17	40	1.4	6.32	6.32
18	20	1.9	4.00	5.00
19	80	1.0	8.94	8.94
20	100	1.15	10.00	10.00
21	40	1.5	6.32	6.32
22	50	1.1	7.07	7.07
23	80	1.0	8.00	10.00
24	10	1.0	3.16	3.16
25	40	1.1	6.00	6.66
26	10	1.2	3.00	3.33
27	40	1.0	6.32	6.32
28	10	1.3	3.00	3.33
29	80	1.05	8.94	8.94
30	40	1.1	6.32	6.32

2. **각 설비들 간의 흐름량 자료**

(1) **설비의 수가 5개인 경우**

n	1	2	3	4	5
1	0	5	2	4	1
2	5	0	3	0	2
3	2	3	0	0	0
4	4	0	0	0	5
5	1	2	0	5	0

(2) **설비의 수가 6개인 경우**

n	1	2	3	4	5	6
1	0	5	2	4	1	0
2	5	0	3	0	2	2
3	2	3	0	0	0	0
4	4	0	0	0	5	2
5	1	2	0	5	0	10
6	0	2	0	2	10	0

(3) **설비의 수가 7개인 경우**

n	1	2	3	4	5	6	7
1	0	5	2	4	1	0	0
2	5	0	3	0	2	2	2
3	2	3	0	1	0	2	5
4	4	0	1	0	5	2	2
5	1	2	0	5	0	10	0
6	0	2	2	2	10	0	5
7	0	2	5	2	0	5	0

(4) **설비의 수가 8개인 경우**

n	1	2	3	4	5	6	7	8
1	0	5	2	4	1	0	0	6
2	5	0	3	0	2	2	2	0
3	2	3	0	0	0	0	0	5
4	4	0	0	0	5	2	2	10
5	1	2	0	5	0	10	0	0
6	0	2	0	2	10	0	5	1
7	0	2	0	2	0	5	0	10
8	6	0	5	10	0	1	10	0

(5) **설비의 수가 9개인 경우**

n	1	2	3	4	5	6	7	8	9
1	0	3	2	0	0	2	10	5	0
2	3	0	4	0	10	4	0	0	2
3	2	4	0	3	4	0	5	5	5
4	0	0	3	0	0	0	0	2	2
5	0	10	4	0	0	5	2	0	0
6	2	4	0	0	5	0	1	2	2
7	10	0	5	0	2	1	0	10	10
8	5	0	5	2	0	2	10	0	1
9	0	2	5	2	0	2	10	1	0

(6) **설비의 수가 10개인 경우**

n	1	2	3	4	5	6	7	8	9	10
1	0	3	2	0	0	2	10	5	0	5
2	3	0	4	0	10	4	0	0	2	2
3	2	4	0	3	4	0	5	5	5	1
4	0	0	3	0	0	0	0	2	2	0
5	0	10	4	0	0	5	2	0	0	0
6	2	4	0	0	5	0	1	2	2	1
7	10	0	5	0	2	1	0	10	10	5
8	5	0	5	2	0	2	10	0	1	3
9	0	2	5	2	0	2	10	1	0	10
10	5	2	1	0	0	1	5	3	10	0

(7) **설비의 수가 15개인 경우**

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	10	0	5	1	0	1	2	2	2	2	0	4	0	0
2	10	0	1	3	2	2	2	3	2	0	2	0	10	5	0
3	0	1	0	10	2	0	2	5	4	5	2	2	5	5	5
4	5	3	10	0	1	1	5	0	0	2	1	0	2	5	0
5	1	2	2	1	0	3	5	5	5	1	0	3	0	5	5
6	0	2	0	1	3	0	2	2	1	5	0	0	2	5	10
7	1	2	2	5	5	2	0	6	0	1	5	5	5	1	0
8	2	3	5	0	5	2	6	0	5	2	10	0	5	0	0
9	2	2	4	0	5	1	0	5	0	0	10	5	10	0	2
10	2	0	5	2	1	5	1	2	0	0	0	4	0	0	5
11	2	2	2	1	0	0	5	10	10	0	0	5	0	5	0
12	0	0	2	0	3	0	5	0	5	4	5	0	3	3	0
13	4	10	5	2	0	2	5	5	10	0	0	3	0	10	2
14	0	5	5	5	5	5	1	0	0	0	5	3	10	0	4
15	0	0	5	0	5	0	0	0	2	5	0	0	2	4	0

(8) **설비의 수가 20개인 경우**

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	5	0	5	2	10	3	1	5	5	5	0	0	5	4	4	0	0	1
2	0	0	3	10	5	1	5	1	2	4	2	5	0	10	10	3	0	5	10	5
3	5	3	0	2	0	5	2	4	4	5	0	0	0	5	1	0	0	5	0	0
4	0	10	2	0	1	0	5	2	1	0	10	2	2	0	2	1	5	2	5	5
5	5	5	0	1	0	5	6	5	2	5	2	0	5	1	1	1	5	2	5	1
6	2	1	5	0	5	0	5	2	1	6	0	0	10	0	2	0	1	0	1	5
7	10	5	2	5	6	5	0	0	0	0	5	10	2	2	5	1	2	1	0	10
8	3	1	4	2	5	2	0	0	1	1	10	10	2	0	10	2	5	2	2	10
9	1	2	4	1	2	1	0	1	0	2	0	3	5	5	0	5	0	0	0	2
10	5	4	5	0	5	6	0	1	2	0	5	5	0	5	1	0	0	5	5	2
11	5	2	0	10	2	0	5	10	0	5	0	5	2	5	1	10	0	2	2	5
12	5	5	0	2	0	0	10	10	3	5	5	0	2	10	5	0	1	1	2	5
13	0	0	0	2	5	10	2	2	5	0	2	2	0	2	2	1	0	0	0	5
14	0	10	5	0	1	0	2	0	5	5	5	10	2	0	5	5	1	5	5	0
15	5	10	1	2	1	2	5	10	0	1	1	5	2	5	0	3	0	5	10	10
16	4	3	0	1	1	0	1	2	5	0	10	0	1	5	3	0	0	0	2	0
17	4	0	0	5	5	1	2	5	0	0	0	1	0	1	0	0	0	5	2	0
18	0	5	5	2	2	0	1	2	0	5	2	1	0	5	5	0	5	0	1	1
19	0	10	0	5	5	1	0	2	0	5	2	2	0	5	10	2	2	1	0	6
20	1	5	0	5	1	5	10	10	2	2	5	5	5	0	10	0	0	1	6	0

(9) **설비의 수가 30개인 경우**

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	3	2	0	0	2	10	5	0	5	2	5	0	0	2	0	5	6	3	0	1	10	0	10	2	1	1	1	0	1
2	3	0	4	0	10	4	0	0	2	2	1	0	5	0	0	0	0	2	0	1	6	1	0	1	2	2	5	1	10	5
3	2	4	0	3	4	0	5	5	5	1	4	1	0	4	0	4	0	6	3	2	5	5	2	1	0	0	3	1	0	2
4	0	0	3	0	0	0	0	2	2	0	6	0	2	5	2	5	1	1	1	1	2	2	4	0	2	0	2	2	5	5
5	0	10	4	0	0	5	2	0	0	0	0	2	0	0	0	0	2	1	0	0	2	0	5	1	0	2	1	0	2	1
6	2	4	0	0	5	0	1	2	2	1	4	10	10	2	5	5	0	5	0	0	0	10	0	0	0	4	0	10	1	1
7	10	0	5	0	2	1	0	10	10	5	10	10	6	0	0	10	2	1	10	1	5	5	2	3	5	0	2	0	1	3
8	5	0	5	2	0	2	10	0	1	3	5	0	0	0	2	4	5	2	10	6	0	5	5	2	5	0	5	5	0	2
9	0	2	5	2	0	2	10	1	0	10	2	1	5	2	0	3	0	2	0	0	4	0	5	2	0	5	2	2	5	2
10	5	2	1	0	0	1	5	3	10	0	5	5	6	0	1	5	5	0	5	2	3	5	0	5	2	10	10	1	5	2
11	2	1	4	6	0	4	10	5	2	5	0	0	0	1	2	1	0	2	0	0	0	6	6	0	4	5	3	2	2	10
12	5	0	1	0	2	10	10	0	1	5	0	0	5	5	2	0	0	0	0	2	0	4	5	10	1	0	0	0	0	1
13	0	5	0	2	0	10	6	0	5	6	0	5	0	2	0	4	2	2	1	0	6	2	1	5	5	0	0	1	5	5
14	0	0	4	5	0	2	0	0	2	0	1	5	2	0	2	1	0	5	3	10	0	0	4	2	0	0	4	2	5	5
15	2	0	0	2	0	5	0	2	0	1	2	2	0	2	0	4	5	1	0	1	0	5	0	2	0	0	5	1	1	0
16	0	0	4	5	0	5	10	4	3	5	1	0	4	1	4	0	0	3	0	2	2	0	2	0	5	0	5	2	5	10
17	5	0	0	1	2	0	2	5	0	5	0	0	2	0	5	0	0	2	2	0	0	0	6	5	3	5	0	0	5	1
18	6	2	6	1	1	5	1	2	2	0	2	0	2	5	1	3	2	0	5	1	2	10	10	4	0	0	5	0	0	0
19	3	0	3	1	0	0	10	10	0	5	0	0	1	3	0	0	2	5	0	0	5	5	1	0	5	2	1	2	10	10
20	0	1	2	1	0	0	1	6	0	2	0	2	0	10	1	2	0	1	0	0	5	2	1	3	1	5	6	5	5	3
21	1	6	5	2	2	0	5	0	4	3	0	0	6	0	0	2	0	2	5	5	0	4	0	1	0	0	0	5	0	0
22	10	1	5	2	0	10	5	5	0	5	6	4	2	0	5	0	0	10	5	2	4	0	5	0	4	4	5	0	2	5
23	0	0	2	4	5	0	2	5	5	0	6	5	1	4	0	2	6	10	1	1	0	5	0	0	4	4	1	0	2	2
24	10	1	1	0	1	0	3	2	2	5	0	10	5	2	2	0	5	4	0	3	1	0	0	0	5	5	0	1	0	0
25	2	2	0	2	0	0	5	5	0	2	4	1	5	0	0	5	3	0	5	1	0	4	4	5	0	1	0	10	1	0
26	1	2	0	0	2	4	0	0	5	10	5	0	0	0	0	0	5	0	2	5	0	4	4	5	1	0	0	0	0	0
27	1	5	3	2	1	0	2	5	2	10	3	0	0	4	5	5	0	5	1	6	0	5	1	0	0	0	0	0	0	10
28	1	1	1	2	0	10	0	5	2	1	2	0	1	2	1	2	0	0	2	5	5	0	0	1	10	0	0	0	2	2
29	0	10	0	5	2	1	1	0	5	5	2	0	5	5	1	5	5	0	10	5	0	2	2	0	1	0	0	2	0	2
30	1	5	2	5	1	1	3	2	2	2	10	1	5	5	0	10	1	0	10	3	0	5	2	0	0	0	10	2	2	0

감사의 글

대학원 진학 후 여러 모로 부족한 저에게 많은 학문적 가르침을 주시고 돌보아주신 이철영 지도교수님께 진심으로 감사드립니다. 또한, 논문 심사 과정에 지도와 격려의 말씀을 아끼지 않으셨던 남기찬, 김환성 교수님께 감사드립니다. 그리고 학부와 대학원 과정 중에 많은 관심과 가르침을 주신 물류시스템공학과 교수님들께 깊은 감사의 인사를 드립니다.

연구실에서 생활하면서 항상 애정어린 눈으로 지켜봐 주셨던 여기태, 임봉택 선배님께 감사드립니다. 박사과정 중에 계신 박용욱, 정희균, 이재원 선배님, 함께 석사과정을 보내면서 기쁨과 어려움을 함께한 석상문, 박성영, 류형근, 오세웅과 연구실 후배인 이선미, 정주영 등 연구실원들에게 이 자리를 빌어 감사의 인사를 드립니다. 동문 선배이고 누구보다도 많은 도움을 주신 이홍걸, 성경빈 선배님께 감사드립니다. 힘들고 어려울 때를 함께 하면서 우정을 나눈 일문형, 현철, 상준, 동혁, 경일형 등의 동기들에게 고맙다고 말하고 싶습니다. 또한, 고등학교 이후부터 변함없는 우정을 지켜오고 있고 논문 완성에 도움을 준 저의 소중한 친구들인 흥우와 명현에게 너무도 고맙고 자랑스럽게 생각합니다.

언제나 맑고 사랑어린 미소와 눈빛으로 저를 묵묵히 지켜보아 주고 어려울 때 변함없이 따뜻한 마음으로 용기를 북돋아 준 사랑하는 경화에게 고마움을 전합니다. 그리고 군 전역후 방황할 때 조언을 해 주고 함께 이공대 전산실에서 생활하게 해 주며 정신적·물질적으로 많은 도움을 주신 이광인 선배님과 좋은 말씀을 많이 해주신 권인영 선배님께 이 자리를 빌어 감사의 인사를 전합니다.

부족한 아들을 하염없이 믿고 끝없는 사랑으로 지금까지 지켜봐 주신 아버님, 어머님, 듬직한 동생 용훈에게 감사의 마음을 전합니다. 그리고 세상에 그 어느 누구보다도 소중한 우리 가족에게 아들의 조그마한 결실을 드립니다.

끝으로, 저에게 도움을 주신 모든 분들에게 고개숙여 감사드립니다.