유전 알고리즘을 이용한 블록조립공장의 생산일정계획

고시근*

A Production Schedule with Genetic Algorithm in Block assembly shop

Shie Gheun Koh*

ABSTRACT

This paper deals with production scheduling of block assembly shop in shipbuilding company. The Objectives of the schedule are leveling of daily total workload and fair assignment of workload to each department of the shop. Optimizing the objectives, we consider some restrictions which consist of block due dates, workability of each department, and space restriction. We formulate the problem in mathematical form. But the model is too complex to find the optimal solution (the problem is NP-complete) and we use the genetic algorithm to find a good solution.

1. 서 론

생산시스템의 일정계획에 있어 계획기간에 걸친 작업부하의 평준화(기간 평준화)와 작업장간의 부하평준화(작업장 평준화)는 생산현장의 관리자에게 중요한 논제로 다루어져 왔다. 특히 조선공업과 같은 노동집약적산업에서는 그 중요성이 더욱 크게 인식되고 있다[1].

대형 조선사업에서는 선박의 효율적인 건조를 위하여 선체의 일부분(블록)을 먼저 만들어 이들을 선박건조용 도크(Dock)에서 탑재(Erection)하는 방식으로 작업하고 있다. 조선소에 따라 조금씩의 차이는 있지만 이러한 블록은 자체의 중량이 수십톤에서 크게는 백여톤에 이르는 하나의 독립된 제품으로 생각할 수 있다. 철판의 절단에서 블록의 조립에 이르는 과정은 주로 공장의 내부에서 작업이 되므로 내업이라 하고 블록의 조립이후에는 공장의 외부에서 작업이 이루어지므

* 부산공업대학교 산업공학과 조교수

로 외업이라고 한다.

조선공업의 생산일정계획은 선박의 계약에서 출발한다. 계약시 선박의 인도일자가 정해지면 그 일자를 기준으로 후진계획(Backward Scheduling)기법을 사용하여 공장의 부하를 고려한 그 선박의 외업일정 및 내업일정을 계획한다.

본 연구에서는 조선일정계획의 일부인 내업일정계획을 다룬다. 특히 내업중에서 마지막 공정인 블록의 조립일정계획을 대상으로 한다. 조선소의 선체생산(Hull Production)일정계획의 일부분인 조립일정계획은 도크의 부하와 블록조립공장의 대략적인 부하를 고려한 블록탑재일정계획(Erection Schedule)에서 블록의 조립완료요구일(Due Date)을 생성하면 그 일자를 준수하는 조립작업장의 생산일정을 생성하는 것이다. 상위계획인 탑재일정계획에서는 블록의 조립착수일만을 기준으로 하여 조립작업장 전체의 일별조립착수물량을 개략적으로 평준화시킨 블록의 조립완료요구일을 생성한다. 그러나 각 블록은 그 조립공기가 서로 다르므로 실제로 각 일자에 걸리는 작업량은 평준화되어 있지않다. 따라서 조립일정계획에서는 블록의 조립일정에 걸쳐 작업장에 걸리는 부하를 분석하여 조립작업장이 계획기간동안 평준화된 작업량을 갖도록 블록의 조립착수일을 결정한다. 또한 조립작업장은 하나의 부서가 아니라 여러개의 부서로 이루어져 있으므로 각 작업부서의 능력에 맞게 골고루 작업량을 배분해 주어야 한다. 즉 본 연구에서는 조립작업장 전체의 일별부하를 평준화하는 블록별 조립착수일자와 계획기간에 걸친 작업부서간의 공정한 작업분배방안을 찾고자 한다.

이러한 문제는 이재동·홍유신[1]에 의해 연구된 바 있다. 이 연구에서는 4-5개의 Job으로 구성된 블록조립작업에 대해 기간평준화와 작업장평준화문제를 비선형 혼합정수계획법 문제로 모형화하여 선형계획법을 이용한 발견적 기법을 사용해 해를 구하였다.

본 연구는 몇가지 제약조건을 제외하면 이재동·홍유신[1]의 연구와 비슷한 상황을 다른 형태의 비선형정수계획법으로 모형화하여 유전알고리즘(Genetic Algorithm: 이하 GA로 약칭함)을 이용한 해를 구한다. 그런데 위에서 언급한 기간평준화와 작업장 평준화를 위한 일정계획을 작성함에 있어 현실적으로 이러한 두가지 목적을 동시에 달성하는 것은 계획대상기간이 길어지고 그에 따라 블록의 개수가 많아질 경우 시간적인 효율이 상당히 저하되리라는 점과 두가지 문제의 상호의존성이 그렇게 크지 않으리라는 점이 예상된다[1]. 따라서 본 연구에서는 이 두가지 문제를 독립적으로 분석하고자 한다. 우선 1단계로 주어진 조립완료요구일을 준수하며 계획기간에 걸친 조립작업장 전체의 일별 부하를 평준화하는 블록별 조립착수일과 완료일을 결정(기간 평준화)하고 2단계에서는 고정된 착수일과 완료일을 이용해 블록별로 투입가능한 작업부서를 고려하여 작업부서간의 작업부하를 공정하게 배분하는 방법을 결정(작업장 평준화)한다.

본 논문의 구성은 2장에서 기간 평준화와 작업장 평준화문제 각각에 대한 비선형 정수계획법 모형을 구축하고 3장에서는 이 최적화모형들의 해를 얻기 위한 GA의 적용방법을 설명한다. 또한 4장에서는 실제상황에 대한 적용결과를 제시하였다.

2. 모형화

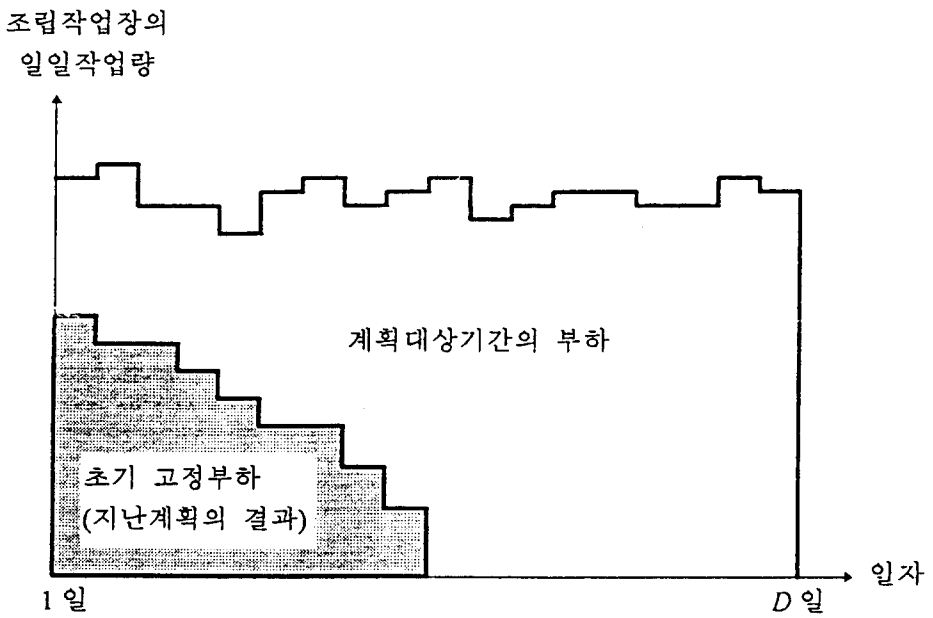
2.1 문제의 가정

본 연구에서는 다음과 같은 가정을 전제로 조립일정계획을 작성한다.

1) 블록별 조립완료요구일은 주어져 있으며 어길 수 없다. 단, 조립완료 요구일 이전에 작업이 완료되는 것은 허용된다. 그러나 완료블록의 저장공간이 제한되어 있으므로 실제 조립완료일은 조립완료요구일보다 정해진 기간이상 빨라질 수 없다.
2) 조립작업은 조립착수일 아침에 시작되어 조립완료일 저녁에 끝나며 각 블록의 조립작업부하는 착수일부터 완료일에 걸쳐 공휴일을 제외한 조립작업기간동안 일정하게 발생한다.
3) 각 조립부서의 작업인원 및 작업영역등 작업능력(Capacity)은 주어져 있으며 공휴일을 제외한 계획기간에 걸쳐 일정한 값으로 고정되어 있다.
4) 블록은 그 블록이 조립될 수 있는 작업부서에 따라 몇 개의 종류로 분류될 수 있다. 또한 작업가능한 부서들 사이에는 우선순위가 있어 블록들은 이 우선순위에 따라 작업장에 배정된다.

2.2 기간 평준화

본 절에서는 계획기간동안 전체 조립작업장의 일별 작업량을 평준화하는 기간 평준화모형을 다룬다. 이 문제를 그림으로 표현하면 [그림 1]과 같다. 조립착수일을 기준으로 계획대상블록을 선정하므로 지난 계획기간에 조립일정이 계획된 블록들 중에서 계획기간의 후반기에 조립착수된 블록들은 이번계획기간의 초기에 고정부하로 존재한다. 이 작업량을 포함하여 이번 계획기간에 조립착수되는 블록들의 작업량을 일별로 평준화하는 것이다.



[그림1] 기간 평준화 문제

[용어 설명]

j : 블록을 나타내는 첨자(j=1, 2, ···, J)

d : 계획기간중 공휴일을 제외한 일자를 표시하는 첨자(d=1, 2, ···, D)

i : 특수블록의 종류를 나타내는 첨자(i=1, 2, ···, I)

w : 작업부서를 표시하는 첨자(w=1, 2, ···, W)

DUR_j : 블록 j의 조립공기

DUE_j : 블록 j의 조립완료요구일

PLN_j : 블록 j의 예정된 조립착수일($=DUE_j-DUR_j+1$),

X_j : 블록 j의 조립완료요구일−블록 j의 실제 조립완료일, 결정변수(정수)

STA_j : 블록 j의 실제 조립착수일($=PLN_j-X_j$)

FIN_j : 블록 j의 실제 조립완료일 ($=STA_j+DUR_j-1$)

$BOUND$: 재고허용기간, 즉 블록이 그 조립완료요구일보다 선행완료될 수 있는 최대일수

L_j : 블록 j가 발생시키는 일일 부하

STD : 작업장 전체의 일일 표준 작업량

$LOAD_d$: 작업장 전체에 대한 d일의 작업량

$LOAD_d=\sum_{j\in\{k|STA_k\le d\le FIN_k\}} L_j$.

UNB_d : 작업장 전체에 대한 d일의 부하 비균형화율(백분율)

$UNB_d=100(LOAD_d-STD)/\ STD$

$\overline{UNB}$ = 계획기간에 걸친 부하 비균형화율의 평균 ($\overline{UNB}=\sum_{d=1}^{D} UNB_d/\ D$)

$LIMIT_i$: 하나의 작업부서에서 동시에 작업할 수 있는 i종류블록의 최대수

NW_i : i종류블록이 조립될 수 있는 작업부서의 수

SPE_{id} : d일자에 작업중인 i종류블록의 총 갯수

$AREA_j$: 블록j의 투영면적

$SPACE_w$: 작업부서 w의 작업가능면적

[목적식 (Objective Function)]

$$Minimize\ F1=\sum_{d=1}^{D}[UNB_d-\overline{UNB}]^2/D \quad (1)$$

[제약조건 (Constraints)]

① 결정변수의 범위

$$0\le X_j\le \min\{BOUND,\ PLN_j-1\},\ j=1,\ 2,\ \cdots,\ J \quad (2)$$

이 식은 다음과 같은 3개의 식을 하나로 표현한 것이다. 즉,

$0\le X_j$: 블록의 조립완료요구일을 어길 수 없다.

$X_j\le BOUND$: 블록의 선행완료일수는 재고허용기간보다 길 수 없다.

$X_j\le PLN_j-1$: 블록의 조립착수일은 계획기간이내여야 한다.

② 블록의 종류에 따른 동시작업 블록갯수 제한

$$SPE_{id}\le NW_i\cdot LIMIT_i,\ i=1,\ 2,\ \cdots,\ I,\ d=1,\ 2,\ \cdots,\ D \quad (3)$$

③ 작업장의 면적 제한 : 조립(용접)작업을 수행하기 위해서는 블록들을 고정시킬 수 있는 공간이 필요한데 이 작업공간의 면적은 작업장 전체의 가용공간에서 작업자의 이동 및 블록의 형태에 의해 발생하는 유휴공간등이 제외되어야 한다.

$$\sum_{STA_j\le d\le FIN_j} AREA_j\ \le\ \alpha\cdot\sum_{w=1}^{W} SPACE_w,\quad 0<\alpha<1,\ d=1,\ 2,\ \cdots,\ D \quad (4)$$

2.3 작업장 평준화

본 절에서는 위의 기간 평준화모형에서 블록의 조립착수 및 완료일자가 결정되면 블록의 실제 작업부서를 결정하는 작업장 평준화 모형을 다룬다. 이 문제에서 고려되어야 할 점은 각 블록이

그 특성에 따라 작업가능한 부서가 정해져 있으며 또한 작업가능한 부서에도 우선순위가 있을 수 있다는 것이다. 예를 들어 어떤 종류의 블록은 특정 작업장에 우선적으로 배정되어야 하며 같은 종류의 블록이 이미 정해진 수만큼 동시에 그 작업장에 배정되어졌다면 나머지 블록들은 2순위, 3순위 작업장에 차례대로 배정된다.

[용어 설명]

Y_j : 블록 j가 조립될 작업부서, 결정변수(정수)

STD_w : 작업장 w의 일일 표준 작업량

$LOAD_{wd}$: 작업장 w의 d일의 작업량

$$LOAD_{wd}=\sum_{Y_j=w,\ STA_j\le d\le FIN_j} L_j$$

UNB_{wd} : 작업장 w의 d일의 부하 비균형화율(백분율)

$$UNB_{wd}=100(LOAD_{wd}-STD_w)/STD_w$$

$\overline{UNB_d}$ = d일의 각 작업장 부하 비균형화율의 평균

$$\overline{UNB_d}=\sum_{w=1}^{W} UNB_{wd}/W$$

VAR_d = d일의 각 작업장 부하 비균형화율의 분산

$$VAR_d=\sum_{w=1}^{W}(UNB_{wd}-\overline{UNB_d})^2/W$$

$CODE_j$: 블록 j의 종류($CODE_j$= 1, 2, ···, I)

WS_i : i종류블록이 조립될 수 있는 작업부서들의 집합

WS_i = {w| 작업장 w에서 i종류 블록의 조립가능}

SPE_{wd} : 작업부서 w에서 d일자에 작업중인 i종류블록의 총 갯수

$SPE_{id}(k)$: i종류블록의 k순위 작업부서에서 d일자에 작업중인 i종류블록의 총 갯수

[목적식 (Objective Function)]

$$Minimize\ F2=\sum_{d=1}^{D} VAR_d/D \qquad (5)$$

[제약조건 (Constraints)]

① 결정변수의 범위

$$Y_j \in WS_{CODE_j} \qquad (6)$$

② 블록의 종류에 따른 갯수 제한

$$SPE_{wid} \le LIMIT_i,\ w=1, 2, \cdots, W,\ i=1, 2, \cdots, I,\ d=1, 2, \cdots, D \qquad (7)$$

③ 블록의 우선순위 제한

$$If\ SPE_{id}(k)<LIMIT_i,\ then\ SPE_{id}(k+1)=0,\ i=1, 2, \cdots, I,\ d=1, 2, \cdots, D \qquad (8)$$

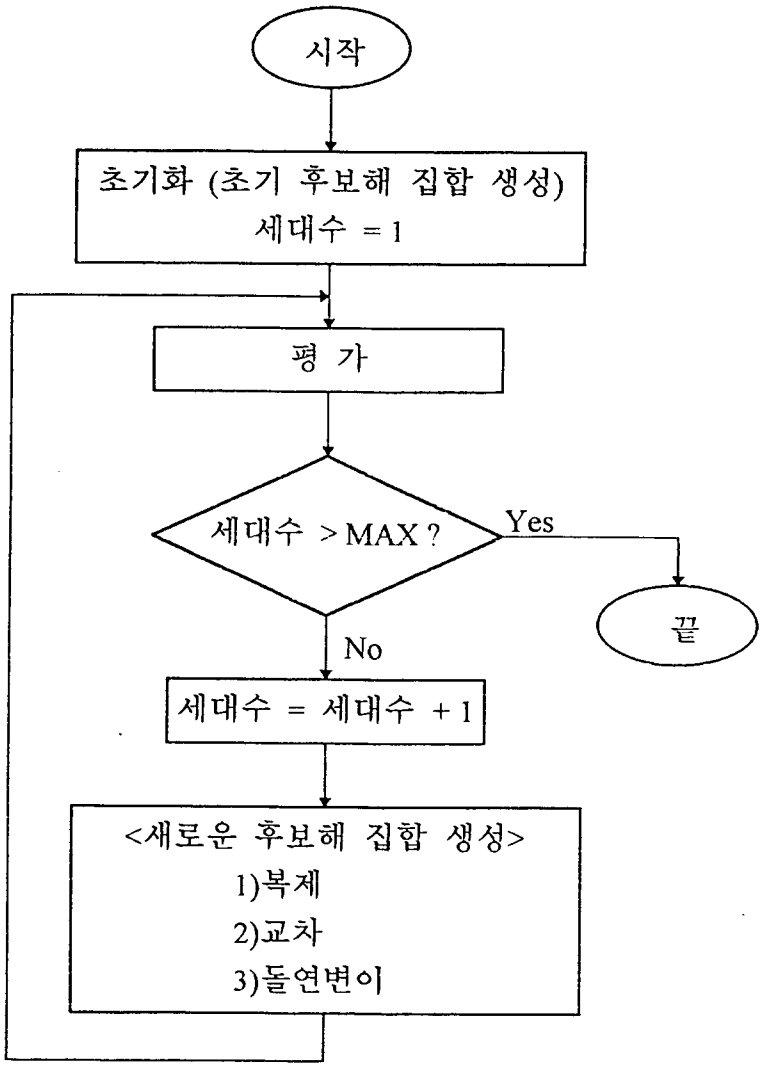
④ 작업장의 면적 제한

$$\sum_{Y_j=w,\ STA_j\le d\le FIN_j} AREA_j \le \alpha \cdot SPACE_w,\ 0<\alpha<1,\ w=1, 2, \cdots, W,\ d=1, 2, \cdots, D \qquad (9)$$

3. 유전알고리즘을 이용한 해법

본 연구의 대상시스템은 문제의 크기와 복잡성으로 인해 수리적인 방법으로 최적해를 구하는 것은 거의 불가능하리라 예상되므로 휴리스틱 기법에 의한 해를 구할 수 밖에 없다. 그런데 인공지능의 한 기법인 GA는 생태계의 적자생존 및 유전법칙에 그 원리를 둔, 최적해에 대한 개선탐색(Adaptive Search) 알고리즘으로서 일반적인 휴리스틱 기법에 비해 다음과 같은 여러가지의 장점을 가진다고 알려져 있다[2, 3]. 첫째, 휴리스틱 기법은 문제의 유형에 따른 고유한 해법을 일일이 개발해야 하나 GA는 거의 모든 유형의 문제에 대하여 일관성 있게 적용할 수 있다. 둘째, 휴리스틱 기법에서는 주어진 문제의 상황이나 가정이 조금만 달라져도 종전의 해법이 더 이상 적용불가능한 경우가 허다하나 GA에서는 변경내용에 따라 목적함수 및 제약조건의 코딩만 수정하면 계속적인 적용이 가능하다. 셋째, Unimodal 문제로부터 Multimodal 문제, 그리고 Combinatorial 문제에 이르기까지 광범위한 문제영역에

걸쳐, 비록 최적해의 여부는 알 수 없어도 좋은 해를 제공하는 Robustness를 지니고 있다. 따라서 본 연구에서는 최적해의 탐색에 GA를 활용하기로 한다. GA의 절차는 [그림 2]와 같이 크게 3단계로 이루어져 있다. 본 장에서는 앞 장에서 제시한 두 모형에 대해 각각 GA를 적용하는 절차를 [그림 2]의 3단계에 따라 개략적으로 설명한다.



[그림2] 유전 알고리즘의 구조

3.1 기간 평준화

(1) 초기화

정해진 수(Population Size)만큼의 초기 후보해(Candidate)를 생성한다. 각 가능해는 블록수만큼의 비음 정수열(Nonnegative Integer String)로 구성된다. 예를 들어 계획대상블록(예정된 조립착수일자가 계획대상기간에 포함되는 블록)이 10개라면 (0, 0, 1, 0, 2, 3, 1, 2, 0, 1)과 같은 것이 가능해가 될 수 있다. 각 정수는 앞 절의 모형에서 X_i를 표시하므로 예정된 조립착수일보다 몇일이나 빨리 착수되는지를 의미한다. 따라서 정수를 무작위로 선택할 때는 식 (2)의 제약조건을 고려하여 생성하여야 한다. 본 연구에서는 상위계획에서 개략적으로 부하를 평준화한 블록별 조립완료요구일이 주어지므로 이것도 후보해의 일부로 포함시킨다. 즉 이 후보해는 원소가 모두 영(zero)으로 구성된 정수열이 된다.

(2) 평가

정해진 갯수의 후보해가 생성되면 각 후보해를 평가하여야 한다. 우선 제약조건을 고려하여 각 후보해의 가능성(Feasibility)을 조사한다. 여기에서 고려되는 제약조건은 식(3)의 블록종류조건과 식(4)의 면적제약이다. 각 후보해에 의해 각 블록의 조립착수일과 완료일이 정해지면 계획기간내의 각 일자에 어떤 블록들이 작업중인지 알 수 있으므로 이들을 종류별로 집계하여 한계치와 비교하고 그 일자에 작업중인 블록들의 총 투영면적을 합산하여 한계치와 비교할 수 있다. 제약조건을 위배하는 후보해(Infeasible Solution)의 점수는 0점을 부여한다. 제약조건을 만족하는 가능해(Feasible Solution)에 대해서는 각 일자의 부하불균형율을 계산하여 계획기간에 걸친 분산을 구한다. 모든 가능해에 대해 분산을 계산한 다음 그 분산값의 최대를 찾는다. 그러면 각 가능해의 점수로 그 가능해의 부하불균형율 분산과 최대분산과의 차이를 부여할 수 있다. 이것은 우리의 문제가 최소화문제이므로 분산값이 작은 후보해에 높은 점수를 부과하기 위함이다.

(3) 새로운 후보해 집합 생성

① 복제(Reproduction)

앞 단계에서 구한 후보해의 점수를 이용하여 정해진 수만큼의 후보해를 복제한다. 즉, 점수가 높으면 높은 만큼 자신을 복제하여 자식해를 생성하므로 좋은 해는 많이 남고 나쁜 해는 적게 남는 것이다. 또한 비가능해는 점수가 0점이었으므로 자신을 복제할 수 없다.

② 교차(Crossover)

복제에 의한 후보해는 앞 단계와 같은 해들을 가지므로 이들을 이용해 새로운 후보해를 만들어 내야 한다. 본 연구에서는 가장 간단한 형태의 교차방법을 사용한다. 즉, 두개의 후보해를 임의로 선택하여 블록갯수의 범위내에서 무작위로 선택된 정수이후의 정수열을 서로 바꾸는 것이다. 예를 들어 (0, 1, 2, 3, 3, 2, 1, 0, 0, 1)과 (0, 0, 1, 1, 2, 2, 3, 3, 1, 1)의 두 후보해가 선택되고 임의의 정수로 6이 선택되었다면 새로운 두개의 후보해는 (0, 1, 2, 3, 3, 2, 3, 3, 1, 1)과 (0, 0, 1, 1, 2, 2, 1, 0, 0, 1)이 되는 것이다.

교차연산은 모든 후보해에 대해 수행하지는 않는다. 실험적으로 알려진대로[2] 전체의 60%에 대해 교차연산을 수행하여 새로운 후보해 집합을 생성하기로 한다.

③ 돌연변이(Mutation)

탐색의 방향에 변화를 가하기 위해 정수열내의 한 정수를 임의로 바꾸어 준다. 이 과정에서는 1단계의 초기해 생성시와 같이 변수의 범위제약을 고려한 정수값을 선정한다. 그러나 탐색의 방향이

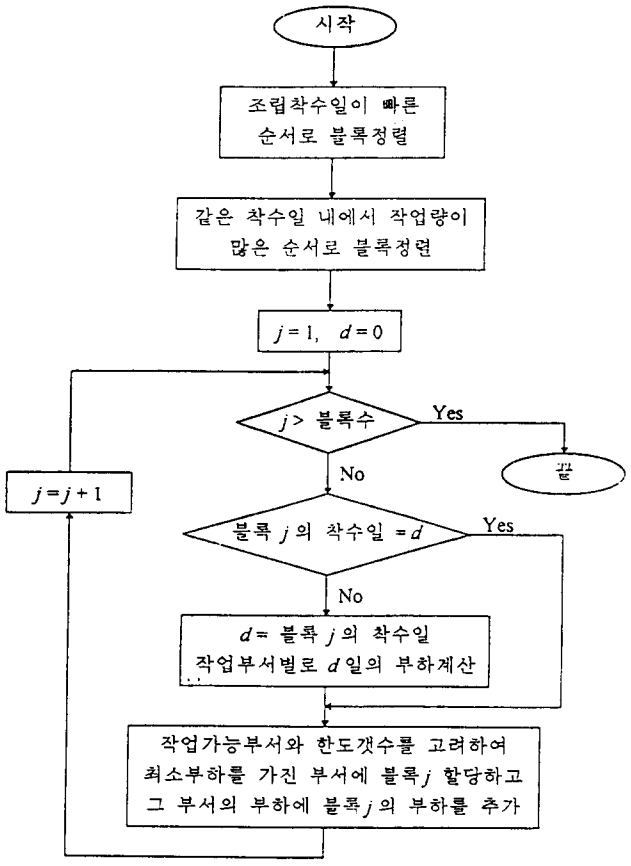
지나치게 무작위적으로 바뀌어 좋은 해에 대한 정보를 분실할 수 있으므로 돌연변이 연산은 조심스럽게 매우 작은 확률로 적용되어야 한다.

3.2 작업장 평준화

작업장 평준화 모형에 대한 GA의 적용도 위의 기간 평준화 모형과 대동소이하다. 서로 다른 부분에 대해서만 설명하면 다음과 같다.

(1) 초기화

각 가능해는 블록수만큼의 양의 정수열(Positive Integer String)로 구성되며 각각의 정수가 의미하는 것은 각 블록의 조립작업이 행해지는 작업부서가 된다. 각 정수값을 생성할 때는 식(6)의 변수범위내에서 식(8)의 우선순위조건을 고려한다. 여기서 만들어지는 후보해는 무작위로 생성된 후보해와 간단한 휴리스틱에 의한 후보해를 포함한다. 사용되는 휴리스틱의 구조는 [그림 3]과 같다. 각 블록의 조립착수일에 대해 작업부서별 작업부하를 계산하여 작업가능한 부서들 중에서 최소의 작업량을 갖는 부서에 그 블록을 할당하는 것이다.



[그림3] 작업장 평준화를 위한 휴리스틱의 구조

(2) 평가

식 (7), (8) 및 (9)의 제약조건을 고려하여 각 후보해의 가능성을 조사한다. 제약조건을 위배하는 후보해에게는 0점을 부여하고 제약조건을 만족하는 가능해에 대해서는 다음과 같은 방법으로 점수를 산정한다. 우선 각 일자의 작업부서별 부하불균형률을 계산하여 그날의 부서간 분산을 구한 뒤 계획기간에 걸친 평균분산을 계산한다. 모든 가능해에 대한 평균분산이 계산되면 그중 최대를 찾는다. 그러면 각 가능해의 점수로 그 가능해의 평균분산과 구해진 최대값과의 차이를 부여한다.

(3) 새로운 후보해 집합 생성

복제, 교차 및 돌연변이 연산은 기간 평준화문제와 동일하다.

4. 결 과

2.5개월(공휴일을 제외하면 60일)에 걸친 실제 자료(계획대상블록의 갯수 : 785개)에 대해 GA를 적용한 결과가 본 장에서 설명된다.

입력자료로는 각 작업장에 대한 사용가능면적과 처리능력, 각 블록에 대한 일일 작업량(부하)과 투영면적 및 블록종류등이 있으며 출력자료는 블록별 조립착수일자 및 작업장이다. 블록들의 종류와 각 종류별 작업가능 작업장과 그 우선순위 및 동시작업 제한갯수는 [표 1]에 있다.

사용된 유전 알고리즘의 파라메터로는 매 세대(Generation)마다 가능해 집단(Population)의 수를 50개로 하였고 실험적으로 알려진 바[2]에 따라 교차(Crossover)확률은 60%를 사용하였으며 돌연변이(Mutation) 확률은 실험을 거쳐 적당한 값을 찾았다.

〈표 1〉 블록 종류별 작업부서 배정 우선순위

	부서 1	부서 2	부서 3	부서 4	부서 5	부서 6	부서 7	*LIMIT*
종류 1				1	2	2	2	3
종류 2		2		1			3	
종류 3	2	1	2	2				3
종류 4	1				2	2	2	2
종류 5	1	1	1	1				∞
종류 6	1	1	1	1	1	1	1	∞

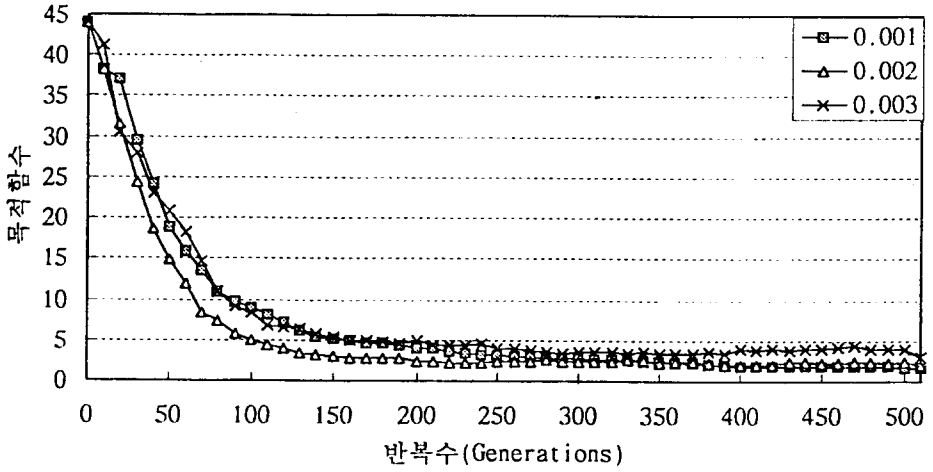
4.1 기간 평준화

우선 몇개의 돌연변이 확률(0.0, 0.001, 0.002, ……, 0.01)에 대해 목적함수의 변화를 관찰하였다. 이 중에서 0.001, 0.002, 0.003에 대해 500세대에 걸친 유전 알고리즘의 적용결과가 [그림 4]에 있다. 이 그림은 매 10회째 세대의 결과만을 표시하며 마지막 결과(그림의 가장 우측 결과)는 500회의 세대중 가장 우수한 결과를 표시한다. 이 그림에 의하면 돌연변이 확률이 0.001일때 가장

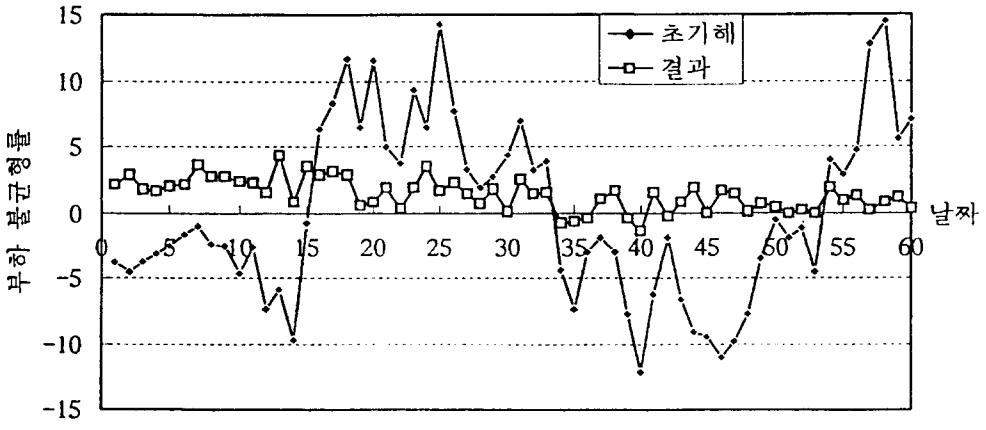
우수한 결과를 찾았지만 확률이 0.002일때 대체로 우수한 결과를 제공해 주었다. 또한 이때 세대수가 200회 정도 진행이 되면 거의 일정한 해에 도달함을 볼 수 있었으며 그때까지의 실행시간은 Windows NT를 탑재한 75Mhz의 Pentium PC에서 CPU Time이 20분 정도였다. 이재동과 홍유신[1]의 연구에서는 1000개 블록에 대해 VAX8800에서 4시간이 소요되었는데 본 연구결과는 비록 블록수가 785개였지만 실행시간 측면에서 상당한 개선이 있었음을 보여준다.

돌연변이 확률이 0. 001일때 초기해(탑재일정계획에 의해 정해진 조립완료요구일에 의한 일정)와 최우수해에 의한 일별작업량의 불균형률은 [그림 5]에 표시되어 있다. 이 그림에 의하면 일별 비균형률이 상당히 개선되어 있음을 알 수 있다.

여기서 구해진 최우수해에 의해 블록의 조립착수시점이 결정되면 다음으로 각 블록의 조립작업이 어느 부서에서 이루어지는 지를 결정하기위해 작업장 균형화 단계로 들어간다.



[그림4] 돌연변이 확률과 목적함수의 변화(기간 평준화)

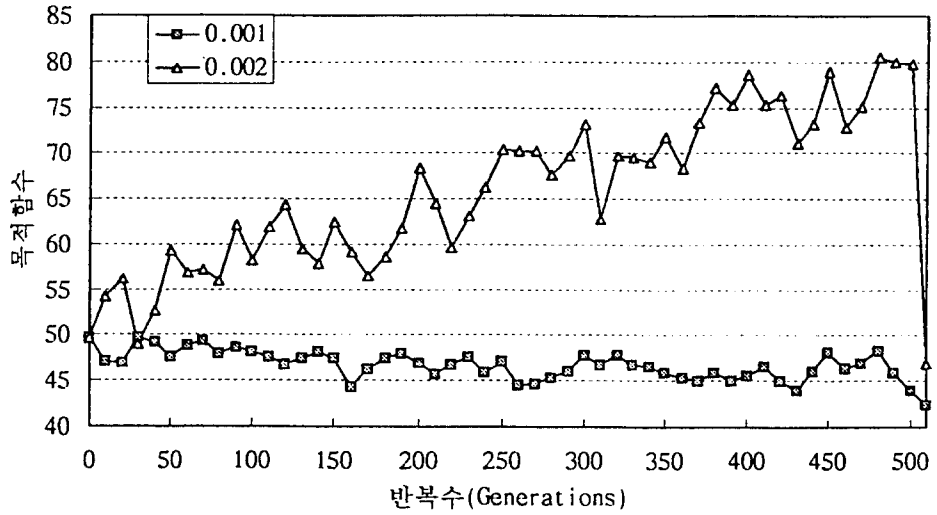


[그림5] GA에 의한 부하균형화의 결과(기간 평준화)

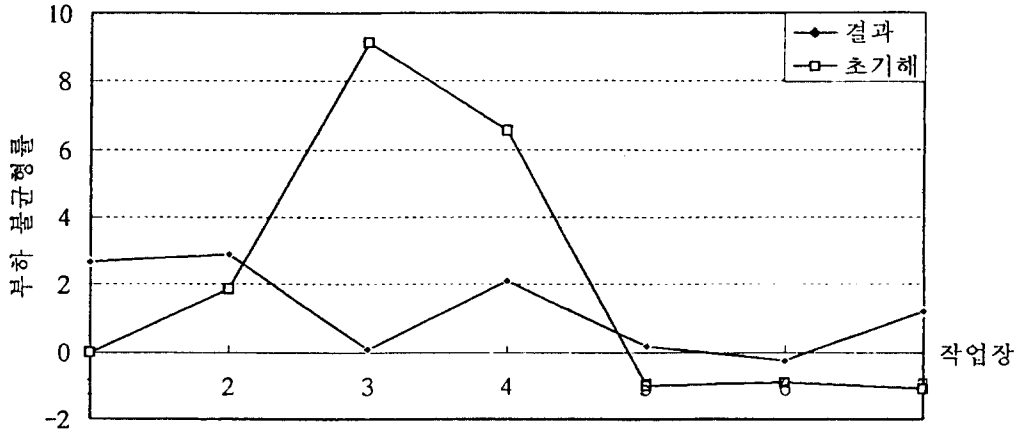
4.2 작업장 평준화

여기에서도 우선 적합한 돌연변이 확률을 찾기 위해 실험을 행하였다. 실험의 결과가 [그림 6]에 그려져 있다. 이 그림은 매 10회째의 결과를 표시하며 마지막 결과는 500회중 최우수해의 결과를 표시한다. 그림에 의하면 돌연변이 확률이 0.001일 때 우수한 결과를 제공해 주며 반복횟수가 300회 정도에 이르면 더이상 발전이 없음을 알 수 있다. 그때까지의 컴퓨터 CPU Time도 20분 내외였다. (이재동과 홍유신[1]의 연구에서는 3시간)

최우수해에 의한 작업부서별 작업할당결과는 [그림 7]에 표시되어 있다. 계획기간에 걸쳐 작업장별로 부하불균형률을 평균한 결과를 볼때 상당히 공정하게 작업량이 배분되어 있음을 볼 수 있었다. 그러나 [그림 6]과 [그림 7]의 결과를 볼때 GA에 의한 초기해의 개선정도가 기간 평준화문제에 비해서 상당히 저조함을 알 수 있다. 이것은 [그림 3]의 휴리스틱이 우수한 초기해를 제공해 준다는 것을 의미한다고 볼 수 있다.



[그림6] 돌연변이 확률과 목적함수의 변화(작업장 평준화)



[그림7] GA이 의한 부하균형화의 결과(작업장 평준화)

5. 결 론

본 연구에서는 다수의 작업부서에서 조립될 수 있는 블록들에 대해 계획기간에 걸친 조립작업장 전체의 작업량을 평준화하도록 조립착수일과 완료일을 결정하고 이 작업량을 작업부서의 능력에 맞게 배분하는 방안을 다루었다.

두 개의 문제를 각각 비선형 정수계획법으로 정식화하고 해를 구하기 위해 유전 알고리즘을 적용하였다. 그 결과 최적해의 보장은 없지만 상당히 우수한 해를 구할 수 있었고 실행시간 측면에 있어서도 적용가능성을 확인하였다.

그러나 본 연구에서는 블록의 작업장내 배치가능성에 대해서는 전혀 고려하지 않았다. 즉, 블록의 면적합을 작업장 면적과 단순비교하는 방법으로 해의 가능성을 조사하였으나 실제로 블록의 형상을 고려하여 작업장내에 블록을 배치할 경우에는 많은 변화를 가질 수 있다. 따라서 블록의 작업장내 배치방법에 대해서도 추가적인 연구가 필요하리라 생각되며 본 연구의 결과는 블록의 실제 형상을 고려한 블록 조립일정계획의 초기해로 사용될 수 있을 것이다.

참 고 문 헌

[1] 이재동, 홍유신, "블록조립공장의 부하평준화를 위한 생산일정계획", [산업공학], 제7권(1994), 제2호, pp. 75-85.

[2] 한용호, 류광렬, "기계-부품군 형성문제의 사례를 통한 유전 알고리즘의 최적화 문제에의 응용", [경영과학], 제12권(1995), 제2호, pp. 105-127.

[3] Goldberg, D. E. , Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.