

유전알고리즘을 이용한 U라인의 작업부하 평활화*

김동묵** · 김여근***

Workload Smoothness in U-Shaped Production Lines Using Genetic Algorithms*

Dong Mook Kim** · Yeo Keun Kim***

■ Abstract ■

In just-in-time production systems, U-shaped production lines rather than traditional straight lines are often adopted since they have some advantages. The advantages of U-lines over straight lines are that the workstations required can be reduced and the necessary number of workers can be easily adjusted when the demand rates are changed. In this paper, we present a new heuristic based on genetic algorithm to improve the workload smoothness in the U-line. In the proposed algorithm, a new genetic representation is developed which is specific to the problem being solved. To enhance the capability of searching good solutions, genetic operators are designed by using the problem-specific information and heuristics. Extensive experiments are carried out on well-known test-bed problems in the literature to verify the performance of our algorithm. The computational results show that our algorithm is a promising alternative to existing heuristics.

1. 서 론

JIT(Just-In-Time)생산 시스템에서, 생산라인의 형태로 U자형의 라인(이하 U라인이라 함)이 자주 사용되고 있다. U라인에서는 흔히 입·출구 작업

이 동일한 숙련작업자에 의해 이루어지면서 제품 한 단위가 출구에서 떠나면 한 단위의 재료가 공정의 입구에 투입된다. 따라서 U라인에서는 라인내의 재공품 수량을 일정하게 유지할 수 있다는 장점을 갖는다. 또한 작업자가 서로 근접하여 작업을

* 본 연구는 한국과학재단 특정기초연구비(98-0200-09-01-3)지원으로 수행되었음.

** 동신대학교 산업공학과

***전남대학교 산업공학과

함으로 가시성이 높아 작업자간의 작업 불균형을 쉽게 파악할 수 있어, 라인관리가 용이하다는 장점을 갖고 있다[17].

본 연구에서는 U라인의 라인밸런싱(U-line line balancing : ULB)문제를 다룬다. 라인밸런싱문제는 라인에 부과된 여러 제약들을 어기지 않고 하나 또는 그 이상의 목적들이 최적이 되도록 작업들을 작업장에 할당하는 문제이다. 전통적인 라인밸런싱은 작업장이 직선형태로 배치된 조립라인(이하 직선라인이라 부름)의 문제를 다루었다. 직선라인의 작업할당에서, 할당가능작업은 모든 선행작업들이 할당된 작업들이다. 그러나 U라인에서 할당 가능한 작업은 모든 선행작업들이 할당된 작업과 함께 모든 후행작업들이 할당된 작업을 포함한다. 따라서 U라인은 직선라인에 비해 작업할당에 대한 유연성이 높게 되고, 이로 인해 좋은 작업할당이 가능하며, 요구되는 작업장의 수는 직선라인의 그것보다 크지 않다[16]. 본 연구는 작업자들이 이동하면서 작업하는 U자형의 생산라인에서 단일모델을 생산하고, 작업시간은 확정적(deterministic)으로 주어지는 경우의 라인밸런싱문제를 다룬다. 이때 작업자의 이동시간은 고려하지 않는다.

ULB문제에 관한 연구는 1994년에 Miltenburg와 Wijngaard[16]에 의해 처음 시작되었다. 그들은 동적계획 모형과 발견적 기법을 제시하였으며, Hwang *et al.*[10]은 시뮬레이티드 어닐링(simulated annealing)을 이용하여 라인밸런싱하였다. 또한 Urban[21]은 정수계획모형을, 김여근 외의 연구[2]에서는 수리 모형과 분지한계법을 제시하였다. 이들 연구에서 라인 밸런싱의 목적은 작업장 수의 최소화로 두었다.

유전알고리즘(genetic algorithm)을 이용한 라인밸런싱에 관한 연구로는, Anderson and Ferris[3], Leu *et al.*[14], Kim *et al.*[11, 12], 김여근 외의 연구[1] 등이 있다. 그러나 이들 연구는 직선라인의 밸런싱문제를 다루고 있다. 유전알고리즘을 이용한 U라인의 밸런싱에 관한 연구는 아직 이루어지지 않고 있다.

본 연구의 목적은 ULB문제를 위한 유전알고리즘을 제시하는데 있다. 라인밸런싱의 목적은 각 작업장에 할당된 작업량의 평활화로 둔다. 이는 작업자의 작업량에 대한 불만을 해소할 수 있고, 최대 작업량을 갖는 작업장의 작업량을 다른 작업장에 분산시켜 사이클타임을 줄이는 효과를 얻을 수 있다. 다루는 문제의 해 공간을 효율적으로 탐색할 수 있는 유전알고리즘을 개발하기 위하여, 새로운 유전표현을 제시하고, 문제가 갖는 특정정보를 이용하고 유지할 수 있는 유전연산자를 개발한다. 그리고 컴퓨터 실험을 통하여 제안한 유전알고리즘의 성능을 입증한다.

2. U라인밸런싱을 위한 유전알고리즘

전통적인 유전알고리즘은 작업 선후행 제약이 존재하는 라인밸런싱과 같이 강한 제약이 존재하는 문제에 적용할 경우, 탐색능력이 낮은 것으로 알려졌다[5, 6]. 이것은 전통적인 유전알고리즘이 문제에 적합한 발견적 기법이나 문제가 갖는 특정정보의 이용보다는 강한 임의성(randomness)을 갖는데 기인한다. 강한 제약을 갖는 여러 조합최적화 문제에 있어서 문제의 특성에 적합한 기법의 결합은 유전알고리즘의 탐색능력을 향상시킨다[5].

본 연구에서는 ULB문제가 갖는 정보와 발견적 기법을 이용하여 해 공간을 좀더 효율적으로 탐색할 수 있는 발견적 유전알고리즘(heuristic genetic algorithm : HGA)을 제안한다. ULB문제가 갖는 정보의 전달과 추출이 용이한, 새로운 개체의 표현방법과 유전연산자를 제시한다. 먼저 본 연구에서 사용되는 기호를 아래와 같이 정의한다.

-
- I : 작업의 집합; $I = \{1, 2, \dots, i, \dots, M\}$.
 - J : 작업장의 집합; $J = \{1, 2, \dots, j, \dots, M\}$.
 - t_i : 작업 i 의 작업시간.
 - R : 삭제되거나 미할당된 작업의 집합.
 - T_j : 작업장 j 의 작업량, 즉 작업장 j 에 할당

- 된 총 작업시간.
- \hat{T}_k : 개체에서 인자 k 에 해당하는 작업들의 작업시간의 합, $k = 1, 2, \dots, 2N-1$.
 - \bar{T} : 작업장에 할당된 작업량의 평균;

$$\bar{T} = \sum_{i \in I} t_i / N.$$
 - $P(i)$: 작업 i 의 선행작업 집합.
 - $S(i)$: 작업 i 의 후행작업 집합.
 - $FR(i)$: 작업 i 의 선행작업들 중 삭제된 작업의 집합; $FR(i) = R \cap P(i)$.
 - $BR(i)$: 작업 i 의 후행작업들 중 삭제된 작업의 집합; $BR(i) = R \cap S(i)$.
 - E_i : 개체에서 $P(i) - R$ 의 작업들의 인자값 중에서 가장 큰 값,
 만약 $P(i) = \emptyset$ 이면 $E_i = 1$.
 - L_i : 개체에서 $S(i) - R$ 의 작업들의 인자값 중에서 가장 작은 값,
 만약 $S(i) = \emptyset$ 이면 $L_i = 2N-1$.
 - $J(i)$: 개체에서 작업 $i \in R$ 에 할당할 수 있는 인자값의 범위;
 $J(i) = \{E_i, E_i+1, \dots, L_i\}$.
 - $T_j(p)$: 개체 p 에서 작업장 j 의 작업량, $j \in J$.
 - $D_j(p)$: $T_j(p)$ 와 평균 작업량의 절대편차;
 $|T_j(p) - \bar{T}|, j \in J$.
 - $B_j(p_1, p_2)$: 개체 p_1 의 작업장 j 의 작업량 편차에서 p_2 의 작업장 j 의 작업량 편차를 뺀 값; $D_j(p_1) - D_j(p_2), j \in J$.

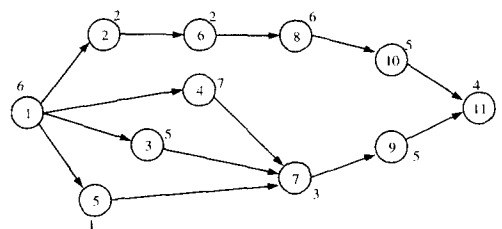
2.1 표현과 초기 모집단

유전알고리즘을 주어진 문제에 적용하기 위해서는 먼저 문제의 특성을 잘 반영할 수 있도록 잠재해를 개체로 표현해야 한다. 본 연구에서 개체표현은 다음과 같이 한다. 작업 i 가 작업장 j 에 할당되었다고 하자. 이 작업 i 가 U라인의 앞에 위치하면 (즉, 선행작업이 모두 할당된 조건하에서 할당되

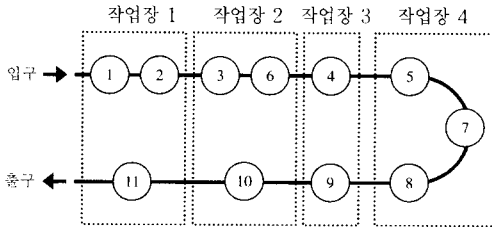
면), 개체에서 i 번째 인자값은 j 로, 그렇지 않고 U라인의 뒤에 위치하면(즉, 후행작업이 모두 할당된 조건하에서 할당되면) 개체의 i 번째 인자값은 $2N-j$ 로 나타난다. 즉, 이 표현에서 작업장 j 에 할당된 작업은 인자값이 j 와 $2N-j$ 를 갖는 위치에 해당하는 작업들이 된다. 이 표현은 일종의 수정그룹번호 표현으로서 개체 해석이 용이할 뿐 아니라 U라인에서 그 작업의 작업위치를 쉽게 알 수 있다.

초기 모집단은 U라인의 작업할당제약을 만족하는 임의의 가능해들로 구성한다. 그 생성절차는 다음과 같다. 초기 가능해는 일단 빈 개체로 시작한다. 작업장 $j=1$ 로 둔다. 선행작업 또는 후행작업이 모두 할당된 작업집합, 즉 할당가능 작업집합을 구한다. 할당가능 작업집합에서 임의로 한 작업 i^* 를 선택하여 이 작업 i^* 가 선행제약을 만족한 작업이면 작업 i^* 의 인자값을 j 로, 후행제약을 만족한 작업이면 $2N-j$ 로 둔다. 그리고 다음 할당을 위하여 할당가능 작업집합을 갱신한다. 이 가능집합에서 임의로 선택된 작업을 할당함으로써 작업장 j 의 작업량이 작업장의 평균작업시간 (\bar{T})보다 크면, j 를 $j+1$ 로 둔다. 그리고 이때 선택된 작업은 작업장 $j-1$ 에 할당한다. 이 과정을 반복하여 $N-1$ 작업장까지 작업할당이 끝나면 남은 작업은 모두 마지막 작업장 N 에 할당한다. 여기서 작업장의 평균작업시간을 기준으로 할당하는 것은 라인밸런싱의 목적이 작업장의 평활화이므로 이를 반영한 것이다.

예로, [그림 1]의 (a)와 같은 선행공정도를 갖는 작업들을 4개의 작업장에 할당한다고 하자. <표 1>은 하나의 초기 개체를 생성하는 과정을 보여주고 있다. 여기서 평균작업시간 $\bar{T} = 11.5$ 이다. 이 초



(a) 선행공정도



(b) U라인의 예

[그림 1] 선행공정도와 U라인

기계의 개체표현은 (1 1 2 3 4 2 4 4 5 6 7)로 되며, 이 개체를 해석하면 [그림 1]의 (b)와 같이 작업장이 편성된다.

2.2 평가함수와 선별

본 연구에서는 작업장간 작업량의 평활성을 평가하는 척도로 Rachamadugu와 Talbot[19]가 제시한 각 작업장의 작업량의 절대편차 평균(mean absolute deviation of workloads : MAD)을 사용한다. 평가함수는 MAD로 다음과 같이 둔다.

$$MAD = \frac{1}{N} \sum_{j=1}^N |(T_j - \bar{T})|$$

선별(selection)은 개체의 적응도(fitness)에 기초하여 다음 세대에 생존할 개체들을 선택하는 과정이다. 선별방법에는 확률바퀴 선별(roulette wheel selection), 순위 선별(ranking selection), 토너먼트 선별(tournament selection) 등이 있다[8, 15]. 본 연구에서는 토너먼트 선별을 사용한다. 토너먼트 선

별은 평가값의 척도를 재구성할 필요가 없고, 토너먼트 크기에 의해 모집단의 선별력을 조정할 수 있다는 장점을 갖는다. 이 선별의 절차는 다음과 같다[9].

- 단계 1. 모집단의 개체들을 임의로 순서로 나열한 순열을 만든다.
 - 단계 2. 순열의 첫 개체부터 l 까지의 개체들의 평가값을 비교하여, 가장 좋은 개체를 선택한다. 그리고 비교된 개체들은 순열에서 삭제한다.
 - 단계 3. 선택된 개체 수가 모집단의 크기와 같으면 종료한다.
 - 단계 4. 순열을 모두 사용하였으면 단계 1로 가고, 그렇지 않으면 단계 2으로 간다.
- 여기서 l 은 토너먼트 크기이고, 흔히 2가 사용된다. 본 연구에서는 $l=2$ 로 두었다.

본 연구에서는 우수개체 보존전략(elitism)을 사용하였다. 이 전략은 가장 좋은 개체가 유전연산을 통해 손실되는 것을 방지하기 위하여 현재까지 생성된 개체 중에서 가장 좋은 개체를 모집단에 보존시키는 전략이다.

2.3 발견적 구조연산자

본 연구에서 제시한 수정그룹번호 표현에서는 그룹(작업장)을 이루는 인자들이 개체내에 분산되어 있으므로, 고전적 교차 방법인 일점(one-point) 교차나 이점(two-point)교차는 성능이 좋은 작업

<표 1> 초기해 생성의 예

작업장	할당가능 작업집합	임의의 작업선택	작업장의 작업량	개체의 인자값	비 고
1	1, 11	1	6	1	선행제약 만족 후행제약 만족 선행제약 만족
	2, 3, 4, 5, 11	11	6+4=10	7	
	2, 3, 4, 5, 9, 10	2	10+2=12 > \bar{T}	1	
2	3, 4, 5, 6, 9, 10	10	5	6	후행제약 만족 선행제약 만족 선행제약 만족
	3, 4, 5, 6, 8, 9	3	5+5=10	2	
	4, 5, 6, 8, 9	6	10+2=12 > \bar{T}	2	
3	4, 5, 8, 9	9	5	5	후행제약 만족 선행제약 만족
	4, 5, 7, 8	4	5+7=12 > \bar{T}	3	
4	마지막 작업장이므로 나머지 작업 5, 7, 8의 모든 인자값은 4가 된다.				

장의 정보(스키마타)를 파괴할 가능성이 높다. 따라서 본 연구에서는 부모의 좋은 정보(평활화가 잘된 작업장)를 자손에게 그대로 상속할 수 있고, ULB문제의 발견적 기법을 효과적으로 이용할 수 있는 발견적 구조교차(heuristic structural crossover)와 발견적 구조 돌연변이(heuristic structural mutation)를 개발하였다. 이 연산자는 Von Laszewski[13]의 구조교차와 구조 돌연변이를 ULB문제에 적합하도록 변형한 것이다.

본 연구에서 제시한 발견적 구조교차의 절차는 아래와 같다. 교차할 두 부모를 p_1, p_2 로 두자.

단계 1. 부모 p_1, p_2 중 임의의 한 부모 (p_1)를 선택하고, $B_j(p_1, p_2), j=1, 2, \dots, N$ 를 계산한다.

단계 2. $B_j(p_1, p_2)$ 가 큰 순서로 $N_q (< N)$ 개의 작업장을 선택하여 집합 Q를 만든다. 여기서 N_q 는 다른 부모 (p_2)로부터 자손에 유전될 작업장의 수를 나타내는 파라미터이다.

단계 3. Q에 속한 작업장의 번호에 해당하는 p_2 의 인자들을 모두 자손 (o_1)의 같은 위치에 복사한다. 자손 o_1 에서 결정되지 않은 인자는 p_1 의 같은 위치에 있는 인자를 상속받는다.

단계 4. p_1 로부터 상속받는 o_1 의 인자 중 Q에 속한 작업장에 해당하는 인자와 선후행관계를 어긴 인자들을 지운다.

단계 5. 단계 4에 의해 미할당된 작업은 재할당한다. 재할당 방법은 2.4절에서 설명된다.

p_1 과 p_2 의 역할을 바꾸어 자손 o_2 를 생산한다.

단계 2는 두 부모로부터 상대적으로 좋은 평가값을 갖는 작업장을 추출하는 과정이다. 여기서 $B_j(p_1, p_2)$ 의 값이 상대적으로 크다는 의미는 작업장 j 의 작업 할당에서 p_1 에 비하여 p_2 가 상대적으로 좋은 정보(평균작업량에 가까운 작업할당)를 갖고 있음을 나타낸다. 단계 3은 단계 2에서 추출된 작업장들을 자손에게 상속하는 과정이다. 예로,

작업장 j^* 가 선택되었다면, p_2 에서 작업장 j^* 에 해당하는 인자값 j^* 와 $2N-j^*$ 가 자손 o_1 의 동일 위치에 복사된다. 단계 4는 부모 p_2 로부터 유전된 인자는 보존하고, 선후행 관계제약을 어긴 인자들을 삭제하는 과정이다. 단계 5는 미할당된 작업들을 선후행 제약을 만족하는 좋은 해를 생산하는 과정이다. 이 절차는 2.4절에서 다룬다.

예를 들어 제안한 교차 방법을 설명하기로 한다. [그림 1] (a)의 문제에서 다음과 같은 두개의 개체가 선택되었다고 하자. 교차할 작업장의 수 $N_q = 1$ 로 둔다

$$p_1 = (1 \ 1 \ 3 \ 3 \ 4 \ 1 \ 4 \ 6 \ 4 \ 6 \ 6)$$

$$p_2 = (1 \ 2 \ 2 \ 3 \ 1 \ 3 \ 5 \ 4 \ 6 \ 4 \ 7)$$

단계 1. 평균 작업량 \bar{T} 는 11.5이므로, $B_1(p_1, p_2) = 1, B_2(p_1, p_2) = 3, B_3(p_1, p_2) = 0, B_4(p_1, p_2) = 2$ 이다.

단계 2. $N_q = 1$ 이므로, $B_j(p_1, p_2)$ 가 가장 큰 j 는 2이다. 즉, $Q = \{2\}$ 이다.

단계 3. p_2 에서 작업장 2에 할당된 작업들(인자 값이 2, 6인 인자들)을 같은 위치에 복사하고, 남은 인자들은 p_1 에서 복사하면 다음과 같이 된다.

$$\begin{matrix} p_1 = & (& 1 & 1 & 3 & 3 & 4 & 1 & 4 & 6 & 4 & 6 & 6 &) \\ & & & & & & & \downarrow & \downarrow & \downarrow & \downarrow & & & \\ o_1 = & (& 1 & 2 & 2 & 3 & 4 & 1 & 4 & 6 & 6 & 6 & 6 &) \\ & & & \uparrow & \uparrow & & & & & \uparrow & & & \\ p_2 = & (& 1 & 2 & 2 & 3 & 1 & 3 & 5 & 4 & 6 & 4 & 7 &) \end{matrix}$$

단계 4. p_1 에서 상속받는 인자중 인자값이 2와 6인 인자를 지운다. 또한 선행제약을 어긴 할당은 작업 6이다. 이 작업의 인자값을 지운다. 이 결과는 다음과 같다.

$$o_1 = (1 \ 2 \ 2 \ 3 \ 4 \ * \ 4 \ * \ 6 \ * \ *)$$

단계 5. 삭제된 작업 6, 8, 10, 11을 재할당한다. 재할당 방법은 2.4절에서 다룬다.

유전알고리즘에서 돌연변이는 다양한 해공간의 탐색을 가능하게 하여, 부분최적에 조기 수렴하는 것을 방지하는 역할을 한다. 제안한 발전적 구조 돌연변이는 돌연변이율에 의해 선택된 인자를 지우고, 지워진 인자들에 해당하는 작업들을 재할당한다. 이때 재할당 방법은 제안한 교차에서와 같은 방법을 사용한다. 돌연변이율은 인자 단위로 적용한다.

2.4 재할당 방법

제안한 유전연산에서 인자값이 결정되지 않는 인자(미할당 작업)들의 인자값을 부여하는 과정이 필요하다. 즉, 삭제된 작업의 재할당이 요구된다. 이때 가능해이면서 좋은 해를 생산하기 위하여 문제가 갖는 특정 정보와 발전적 기법을 이용한다. 재할당의 설명을 위하여 먼저 지시함수(indicator function)를 아래와 같이 정의한다.

$$I(k) = \begin{cases} 1, & k \neq N \text{이면,} \\ 0, & \text{그렇지 않으면.} \end{cases}$$

본 연구에서 개발한 재할당 방법의 절차는 다음과 같다.

단계 1. R 과 $FR(i)$, $BR(i)$, $i \in R$ 를 구하고,

$$\widehat{T}_k, k=1, 2, \dots, 2N-1 \text{과 } \overline{T} \text{를 계산한다.}$$

단계 2. $FR(i)$ 또는 $BR(i)$ 이 공집합인 작업 중 작업시간이 가장 큰 작업, i^* 를 선택한다.

단계 3. 작업 i^* 의 E_{i^*} 와 L_{i^*} 를 구하고, $J(i^*) = \{E_{i^*}, E_{i^*}+1, \dots, L_{i^*}\}$ 로 둔다.

단계 4. a) $A = \{k \mid \widehat{T}_k + I(k)\widehat{T}_{2N-k} + t_{i^*} \leq \overline{T},$

$$k \in J(i^*)\}$$
이 공집합이 아니면

$$B = \{k \mid \max(\widehat{T}_k + I(k)\widehat{T}_{2N-k}), k \in A\}$$

를 구한다.

1) $FR(i^*)$ 이 공집합이면, 집합 B 중에서 인자값이 작은 k 를 k^* 로 한다.

2) 그렇지 않으면, 집합 B 중에서 인자값

이 큰 k 를 k^* 로 한다.

b) 집합 A 가 공집합이면, $C = \{k \mid \min(\widehat{T}_k + I(k)\widehat{T}_{2N-k}), k \in J(i^*)\}$ 를 구한다.

1) $FR(i^*)$ 이 공집합이면, 집합 C 중에서 인자값이 작은 k 를 k^* 로 한다.

2) 그렇지 않으면, 집합 C 중에서 인자값이 큰 k 를 k^* 로 한다.

단계 5. 작업 i^* 에 k^* 를 할당하고 작업 i^* 를 집합 R 에서 삭제한다. $R = \emptyset$ 이면 종료하고, 그렇지 않으면 $FR(i)$, $BR(i)$, $i \in R$ 과 \widehat{T}_k 를 갱신하고 단계 2로 간다.

단계 2에서는 작업시간이 큰 작업일수록 작업장의 작업량 편차에 크게 영향을 미치므로, 할당가능한 작업 중에서 작업시간이 가장 큰 작업 i^* 를 먼저 선택한다. 단계 3은 선행행 제약을 만족하는 할당을 위하여, 선택된 작업 i^* 가 할당될 수 있는 작업장의 범위를 구하는 과정이다. 단계 4에서는 작업 i^* 가 할당되어도 작업장의 작업량이 평균 작업량 이하인 작업장들이 존재하면, 이러한 작업장에 우선적으로 작업을 할당하며, 여러개 존재할 경우는 작업량이 가장 많은 작업장을 선택하여 작업 i^* 를 할당한다. 이렇게 함으로써, 개체가 좋은 구조(잘 할당된 작업장)를 갖도록 한다. 이는 일종의 빌딩블록(building block)을 만드는 것과 같다. 한편, 작업 i^* 가 할당가능한 작업장에 할당될 때 모두 평균작업량이 초과되면, 이때는 작업량이 가장 적은 작업장에 작업 i^* 를 할당하여 편차를 줄인다. 그리고 작업을 할당할 때, 선정된 작업장에서 라인의 앞 작업(인자값 k) 또는 뒷 작업(인자값 $2N-k$)에 모두 할당 가능한 경우가 발생하는 데 이때는 미할당 작업이 할당될 수 있는 작업장의 범위가 최대가 되도록 할당한다. 단계 5는 자료를 갱신하는 과정이다.

2.3절의 예에서 얻은 개체 (1 2 2 3 4 * 4 * 6 * *)에 이 할당 방법을 적용하면 (1 2 2 3 4 2 4 3 6 4

<표 2> 재할당 방법의 예

반복	R	FR(i)	BR(i)	\hat{T}_k	단계 2	단계 3		단계 4	단계 5
						E_{i^*}	L_{i^*}		
1	{6,8,10,11}	FR(6) = ∅ FR(8) = {6} FR(10) = {6,8} FR(11) = {6,8,10}	BR(6) = {8,10,11} BR(8) = {10,11} BR(10) = {11} BR(11) = ∅	$\hat{T}_1=6, \hat{T}_2=7,$ $\hat{T}_3=7, \hat{T}_4=4,$ $\hat{T}_5=0, \hat{T}_6=5,$ $\hat{T}_7=0$	$i^*=11$	6	7	$k^*=7$	작업 11에 7을 할당
2	{6,8,10}	FR(6) = ∅ FR(8) = {6} FR(10) = {6,8}	BR(6) = {8,10} BR(8) = {10} BR(10) = ∅	$\hat{T}_1=6, \hat{T}_2=7,$ $\hat{T}_3=7, \hat{T}_4=4,$ $\hat{T}_5=0, \hat{T}_6=5,$ $\hat{T}_7=4$	$i^*=10$	2	7	$k^*=4$	작업 10에 4를 할당
3	{6,8}	FR(6) = ∅ FR(8) = {6}	BR(6) = {8} BR(8) = ∅	$\hat{T}_1=6, \hat{T}_2=7,$ $\hat{T}_3=7, \hat{T}_4=9,$ $\hat{T}_5=0, \hat{T}_6=5,$ $\hat{T}_7=4$	$i^*=8$	2	4	$k^*=3$	작업 8에 3을 할당
4	{6}	FR(6) = ∅	BR(6) = ∅	$\hat{T}_1=6, \hat{T}_2=7,$ $\hat{T}_3=13, \hat{T}_4=9,$ $\hat{T}_5=0, \hat{T}_6=5,$ $\hat{T}_7=4$	$i^*=6$	2	3	$k^*=2$	작업 6에 2를 할당, R=∅

7)이 된다. <표 2>는 미할당 작업들 6, 8, 10, 11이 재할당되는 과정을 보여 주고 있다. 이 개체를 해석하면 작업장 1, 2, 3, 4에 각각 작업 {1, 11}, {2, 3, 6, 9}, {4, 8}, {5, 7, 10}이 할당되었음을 의미한다.

2.5 HGA의 절차

지금까지 설명한 유전알고리즘의 구성요소를 가지고, HGA의 전체 절차를 보이면 아래와 같다. 단, $P(t)$ 는 t 세대의 모집단, Np 는 모집단의 크기, Ng 는 개체의 인자 수(개체의 길이), Pc 와 Pm 은 각각 교차율과 돌연변이율이다.

- 단계 1. 가능한 개체들을 임의로 발생시켜 초기 모집단 $P(t), t=0$ 를 생성한다. 모집단 내 개체의 적응도를 평가한다.
- 단계 2. 개체의 적응도를 기준으로 $P(t)$ 로부터 Np 개의 개체들을 선별하여, 다음세대의 모집단 $P(t+1)$ 에 복사한다.
- 단계 3. 교차를 위하여, $P(t+1)$ 로부터 $0.5 * Pc * Np$ 쌍의 부모를 임의로 선별한다.
- 단계 4. 교차율 Pc 로 발견적 구조교차를 하여, 이로부터 새롭게 생산된 $Pc * Np$ 개의 자손들을

그들의 부모와 대체한다.

단계 5. $P(t+1)$ 에 있는 개체의 인자 중 $Pm * Np * Ng$ 개를 임의로 선택하여 발견적 구조 돌연변이를 적용한다.

- 단계 6. 새롭게 생산된 개체들의 적응도를 평가한다.
- 단계 7. 종료조건을 만족하면, 종료한다. 그렇지 않으면, $t \leftarrow t+1$ 이고, 단계 2로 간다.

3. 실험 및 분석

제안한 HGA의 성능 분석을 위하여 여러 기준 기법들과 비교 분석하였다. 이를 위하여 실험 문제로는 라인밸런싱의 벤치마크 문제들을 사용하였다. 작업장 수가 많은 경우를 실험하기 위하여 작업시간이 너무 큰 작업들은 작업시간을 조정하였다. Tonge[20]의 문제에서 작업 18의 작업시간은 319에서 143으로, Bartholdi[4]의 문제에서 작업 79의 작업시간은 218에서 111로, 작업 108의 작업시간은 383에서 43으로 바꾸었다. 알고리즘은 C++ 프로그램 언어로 구현하였으며, 300MHz Pentium CPU를 장착한 IBM-PC에서 수행되었다. 실험 결과는 유전알고리즘을 20회 반복 수행하여 얻은 값을 평균

하여 나타냈다.

3.1 파라미터 값의 결정

제안한 HGA에서 파라미터는 모집단의 크기, 토너먼트의 크기, 교차율, 돌연변이율, 그리고 부모로부터 자손에게 유전될 작업장(교차작업장)의 수 N_g 이다. 이들 파라미터 값들은 예비실험을 통하여 얻었다. 모집단의 크기는 30으로 두었다. 이는 실험에서 모집단의 크기를 30보다 크게하는 경우, 모집단의 다양성은 증가되지만, 한정된 수의 개체를 재생산하는 상황아래서 해의 탐색 효율은 오히려 저하되는 결과를 가져왔다. 토너먼트의 크기는 2로 한다. 교차율, 돌연변이율, 그리고 교차 작업장의 수에 있어서 좋은 결과를 보인 파라미터 값의 범위는 <표 3>과 같았다. 이후의 비교실험에서 각 문제에 대한 파라미터의 값은 괄호의 값을 사용하였다. 그리고 제안한 알고리즘은 연속적으로 생산된 개체수가 5000개에 이를 때까지 현재의 최선해를 갱신하지 못하면 종료하였다.

3.2 성능분석

HGA의 성능분석을 위하여 기존의 발견적 기법

과 비교하였다. 비교한 발견적 기법은 할당가능한 작업중에서 작업시간이 가장 큰 것을 할당하는 최대작업시간 규칙(MAXDU)[18], 임의로 할당하는 임의 할당방법(RAND), Gehelein과 Patterson[7]에 의해 수정된 Hoffmann기법을 사용하였다. 수정된 Hoffmann기법에서는 HOFF-2.0을 사용하였다. 이들 발견적 기법은 직선라인의 밸런싱문제를 위하여 개발된 것이다. 따라서 본 연구에서는 U라인 특성에 맞게 할당가능 작업집합을 선행작업이 모두 할당된 작업집합과 모든 후행작업이 할당된 작업집합의 합집합으로 변형하였다. 본 연구는 작업장 수가 주어진 문제를 다루고 있다. 따라서 이들 기법과 제안한 기법을 비교하기 위하여, 사이클타임(CT)을 이론적인 사이클타임($[1/N \sum_{i=1}^M t_i]^+$)부터 1씩 증가시키면서 구해진 작업장 수가 N 과 같아질 때까지 반복 수행하여 해를 구하였다. 여기서 $[x]^+$ 는 x 보다 크거나 같은 가장 작은 정수이다. Rachamadugu와 Talbot[19]는 기존의 발견적 기법에 의해 구한 해를 초기해로 하여, 작업장간 작업 부하의 평활성을 보다 개선시키는 기법(R&T)을 제시했다. 이 기법에서는 직선라인의 작업부하가 보다 평활화될 수 있도록 작업들을 작업장간에 이동 또는 교환한다. 본 연구에서는 U라인의 작업할

<표 3> 유전 파라미터 값

문제	작업 수	작업장 수	교차율	돌연변이율	교차작업장 수
Jackson	11	3,5,7	0.5-0.9 (0.7)	0.2-0.25 (0.2)	1-2 (1)
Dar-El	11	3,5,7	0.5-0.9 (0.7)	0.2-0.25 (0.2)	1-2 (1)
Mitchell	21	3,5,6,7	0.5-0.9 (0.7)	0.2-0.25 (0.2)	1-2 (1)
Sawyer	30	5	0.5-0.9 (0.7)	0.2-0.25 (0.2)	1-2 (1)
		8,10,12	0.5-0.9 (0.7)	0.2-0.25 (0.2)	1-3 (2)
Kibridge & Wester	45	6,7,10	0.5-0.9 (0.7)	0.2-0.25 (0.2)	2-3 (2)
		12,15	0.5-0.9 (0.7)	0.2-0.25 (0.2)	2-3 (3)
Tonge	70	7,12	0.65-0.9(0.8)	0.1-0.15 (0.15)	2-3 (2)
		15	0.65-0.9(0.8)	0.1-0.15 (0.15)	2-4 (4)
		20,21,22	0.7-0.9 (0.8)	0.1-0.15 (0.15)	2-5 (5)
Arcus	83	10,11,12,14	0.7-0.9 (0.8)	0.1-0.15 (0.15)	1-3 (3)
		18	0.7 0.9 (0.8)	0.1-0.15 (0.15)	2-5 (4)
Arcus	111	10,12	0.7-0.9 (0.8)	0.1-0.15 (0.15)	1-3 (3)
		14,15,16	0.7-0.9 (0.8)	0.1-0.15 (0.15)	4-5 (5)
		20	0.7 0.9 (0.8)	0.1-0.15 (0.1)	5-6 (6)
		25,27	0.7-0.9 (0.8)	0.1-0.15 (0.15)	6-7 (7)
Bartholdi	148	10,15,20	0.7-0.9 (0.8)	0.1-0.15 (0.15)	1-3 (2)
		25,30	0.7-0.9 (0.8)	0.1-0.15 (0.15)	2-7 (6)

당 제약하에서 R&T기법의 이동과 교환방법을 사용하였다. R&T기법의 초기해는 RAND, MAXDU, HOFF-2.0에서 구한 해들을 각각 사용하였다. 이와 같이 구한 해 중 가장 좋은 해와 본 연구에서 제안한 HGA의 해를 비교하였다.

<표 4>는 발견적 기법과 HGA의 결과를 보여 주고 있다. 1열은 각 문제의 작업 수를, 2열은 작업장

수를 나타내며 3, 4, 5열은 각각 RAND, MAXDU, HOFF-2.0에 의해 얻은 MAD이다. 단, RAND는 할당가능 작업집합에서 임의로 작업을 할당하므로 제안한 HGA와 공정한 비교를 위하여, 400번 반복 실험 중에서 그 결과가 가장 좋은 것을 나타내었다. 6, 7, 8열은 3, 4, 5열의 결과로 얻은 해를 각각 R&T기법으로 개선한 결과이다. 9와 10열은 HGA

<표 4> HGA와 기존 발견적 기법과의 해의 성능비교

Problem	# of stat.	Heuristics			R&T with			HGA				개선율(%)
		RAND	MAXDU	HOFF-2.0	RAND	MAXDU	HOFF-2.0	Worst	Best	Mean	s.d.	
Jackson 11	3	0.44	0.89	0.89	0.44	0.44	0.89	0.44	0.44	0.44	0.00	0.00
	5	0.96	0.96	0.32	0.64	0.64	0.32	0.32	0.32	0.32	0.00	0.00
	7	1.06	0.61	0.61	0.94	0.61	0.61	0.49	0.49	0.49	0.00	19.67
Dar-El 11	3	1.78	8.44	2.44	1.78	8.44	2.44	0.44	0.44	0.44	0.00	75.28
	5	5.20	10.80	10.80	3.60	3.60	4.00	3.60	3.60	3.60	0.00	0.00
	7	13.10	19.80	19.80	12.41	10.78	10.78	10.78	10.78	10.78	0.00	0.00
Mitchell 21	3	0.00	1.33	0.00	0.00	0.67	0.00	0.00	0.00	0.00	0.00	0.00
	5	0.40	1.20	0.00	0.40	0.80	0.00	0.00	0.00	0.00	0.00	0.00
	6	0.67	1.67	0.83	0.50	0.83	0.83	0.50	0.50	0.50	0.00	0.00
Sawyer 30	5	1.04	3.52	0.32	1.04	0.96	0.32	0.32	0.32	0.32	0.00	0.00
	8	1.38	1.75	0.75	1.25	1.13	0.63	0.50	0.50	0.50	0.00	20.63
	10	1.40	1.08	2.24	1.40	1.08	1.92	0.56	0.48	0.48	0.02	55.56
Kilbridge & Wester 45	6	0.33	2.67	0.00	0.33	1.33	0.00	0.00	0.00	0.00	0.00	0.00
	7	1.06	0.71	0.24	1.06	0.71	0.24	0.24	0.24	0.24	0.00	0.00
	10	0.84	1.32	0.80	0.84	1.12	0.80	0.32	0.32	0.32	0.00	60.00
Tonge 70	12	8.17	13.33	13.67	7.83	8.00	9.00	1.50	1.50	1.50	0.00	80.84
	15	17.97	22.93	23.20	4.29	3.52	21.60	2.43	2.43	2.43	0.00	30.97
	7	3.51	1.22	0.82	3.51	1.22	0.82	0.49	0.49	0.49	0.00	40.24
	12	5.92	1.58	0.92	4.75	1.33	0.75	0.50	0.50	0.50	0.00	33.33
	15	5.87	4.47	1.07	4.93	4.33	1.07	0.40	0.00	0.19	0.12	82.66
Arcus 83	20	10.55	2.15	2.45	3.55	2.15	2.45	0.95	0.50	0.62	0.11	71.40
	21	9.44	7.39	3.88	4.61	4.94	3.51	1.78	0.49	0.83	0.29	76.37
	22	8.40	3.70	1.76	5.13	2.72	1.76	0.93	0.62	0.72	0.10	59.03
	10	46.84	83.82	21.10	46.84	83.82	21.10	26.96	7.78	15.98	5.40	24.29
	11	48.13	134.61	44.96	48.13	132.61	44.96	51.74	16.13	28.01	11.17	37.70
Arcus 111	12	44.58	81.29	30.28	44.58	66.11	30.28	23.97	11.42	15.59	3.27	48.53
	14	57.64	139.17	26.31	57.64	90.36	26.31	32.22	9.49	17.31	7.16	34.21
	18	73.72	134.63	62.62	73.72	132.85	62.62	71.15	24.81	39.35	11.51	37.16
	10	109.90	4.52	3.90	46.44	3.72	3.90	6.68	0.72	2.18	1.47	41.29
	12	124.79	8.00	8.42	17.25	5.71	6.63	17.92	0.96	5.01	4.26	12.18
Bartholdi 148	14	175.73	28.02	11.64	87.85	9.67	8.76	7.95	1.21	2.93	1.92	66.55
	15	194.48	56.37	15.73	69.65	7.17	9.71	11.09	2.51	6.53	2.74	9.00
	16	150.07	33.87	8.82	29.69	9.42	7.81	9.91	2.30	6.00	2.27	23.12
	20	240.55	52.98	42.25	112.84	13.04	29.05	14.72	2.85	7.29	3.07	44.07
	25	303.80	25.25	103.49	119.63	12.28	48.27	25.38	5.64	12.09	5.94	1.54
Bartholdi 148	27	241.75	171.90	89.28	69.95	25.95	58.25	26.62	10.32	15.46	4.81	40.43
	10	20.90	28.64	0.64	20.90	25.00	0.64	0.32	0.32	0.32	0.00	50.00
	15	16.12	19.87	1.16	7.26	19.87	1.16	0.50	0.50	0.50	0.00	56.90
	20	11.56	13.91	8.46	11.16	13.90	6.68	0.29	0.18	0.22	0.05	96.72
	25	7.74	9.03	7.61	7.20	8.96	5.77	0.63	0.21	0.46	0.14	91.99
30	11.28	11.86	10.06	2.94	8.90	3.48	0.73	0.39	0.58	0.11	80.32	

를 20회 수행하여 얻은 가장 나쁜 해와 가장 좋은 해이고, 11과 12열은 20회 수행결과의 평균과 표준편차이다. 마지막으로 13열은 R&T기법으로 구한 해 중 가장 좋은 해($R\&T_{best}$)에서 HGA의 평균(HGA_{mean})을 뺀 값을 R&T로 구한 $R\&T_{best}$ 로 나누어 100을 곱한 값이다. 즉, 개선율 = $[(R\&T_{best} - HGA_{mean}) / R\&T_{best}] \times 100$ 이다.

실험결과, HGA의 성능이 우수함을 알 수 있다. <표 4>에서 전체 문제에 대한 개선율의 평균은 37.6%이었다. 특히, 작업이 70이상인 큰 규모의 문제들에서의 평균개선율은 48.3%이었다. 제안한 알고리즘은 대형문제에서 비교기법들보다 상대적으로 더 좋은 해를 탐색할 수 있음을 알 수 있다. 그리고 구한 해들의 표준편차가 비교적 낮았다. 이는 제안한 기법이 안정적인임을 알 수 있다.

4. 결 론

본 연구에서는 U라인에서 작업장의 작업량 평활화를 위한 유전알고리즘을 제시하였다. U라인의 특성에 맞는 효율적인 유전알고리즘의 개발을 위하여, 해를 자연스럽게 나타낼 수 있는 표현방법과 발견적 정보를 이용한 유전연산자를 개발하였다. 제안한 알고리즘의 성능을 평가하기 위하여 기존의 발견적 기법들과 그 성능을 비교 분석 하였다. 실험결과, 제안한 알고리즘은 비교된 기법들보다 그 성능이 우수하였다. 특히, 해 공간이 큰 문제에서 더욱 우수하였다. 유전알고리즘의 한 장점은 최적화 문제에서 목적함수와 제약식의 변화에 유연하다는 것이다. 따라서 제안한 알고리즘은 작업자의 이동시간을 고려한 경우나 특정제약이 추가된 U라인 밸런싱문제에 쉽게 적용할 수 있다.

참 고 문 헌

- [1] 김여근, 김용주, 김준혁, 박재승, "양면 조립라인 밸런싱을 위한 유전알고리즘", IE Interfaces, Vol.12, No.1(1999), pp.132-142.
- [2] 김여근, 김재윤, 김동목, 송원섭, "U라인 라인 밸런싱을 위한 분지한계법", 한국경영과학회지, 23권 12호(1998), pp.83-101.
- [3] Anderson, E. J. and M. C. Ferris, "Genetic Algorithms for Combinatorial Optimization: the Assembly Line Balancing Problem," *ORSAJ. Computing*, Vol.6(1994), pp.161-173.
- [4] Bartholdi, J.J., "Balancing Two-sided Assembly Lines: A Case Study," *International Journal of Production Research*, Vol.31(1993), pp.2447-2461.
- [5] Eiben, A. E., P. E. Raué, and Zs. Ruttkay, "How to Apply Genetic Algorithms to Constrained Problems," *Practical Handbook of Genetic Algorithms 1: Applications* (edited by Chambers, L.), (1995), pp.307-365. CRC Press, Boca Raton.
- [6] Falkenauer, E., "A New Representation and Operators for Genetic Algorithms Applied to Grouping Problems," *Evolutionary Computation*, Vol.2(1994), pp.123-144.
- [7] Gehrlein, W.V. and J.H. Patterson, "Balancing Single Model Assembly Comments on A Paper by E.M. Dar-El(Mansoor)," *AIIE Trans.* Vol.10, No.1(1978), pp.109-112.
- [8] Goldberg, D. E., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, Reading(1989).
- [9] Goldberg, D. E., K. Bradley, and D. Kayanmoy, "Messy Genetic Algorithms: Motivation, Analysis, and First Results," *Complex Systems*, Vol.3(1989), pp.493-530.
- [10] Hwang, H., J.U. Sun, and T.H. Yoon, "U-line Line Balancing with Simulated Annealing," *Proceedings of the First ASIA-PACIFIC Decision Sciences Institute Conference*, Hong Kong(1996), pp.101-108.
- [11] Kim, Y. K., Y. J. Kim, and Y. K. Cho, "A

- Heuristic-Based Genetic Algorithm for Workload Smoothing in Assembly Lines," *Computers and Operations Research*, Vol.25, No.2(1998), pp.99-111.
- [12] Kim, Y. K., Y. J. Kim, and Y. H. Kim, "Genetic Algorithms for Assembly Line Balancing with Various Objectives," *Computers and Industrial Engineering*, No.30 (1996), pp.397-409.
- [13] Laszewski, V., "Intelligent Structural Operators for The k-Way Group Partitioning Problem," 4th *ICGA*(1991). pp.619-633.
- [14] Leu, Y.Y., L.A. Matheson, and L.P. Rees, "Assembly Line balancing Using genetic Algorithms with Heuristic-Generated Initial Populations and Multiple Evaluation Criteria," *Decision Science*, Vol.25(1994), pp.581-606.
- [15] Michalewicz, Z., *Genetic Algorithms+Data Structures=Evolutionary Programs* (2nd ed.) Springer-Verlag, Berlin(1994).
- [16] Miltenburg, G.J. and J. Wijngaard, "The U-line Line Balancing Problem," *Management Science*, Vol.40(1994), pp.1378-1388.
- [17] Monden, Y., *Toyota Production System* (2nd Ed.), Industrial Engineering and Management Press, Institute of Industrial Engineers, Norcross, GA(1993).
- [18] Moodie, C.L. and H.H. Young, "A Heuristic Method of Assembly Line Balancing for Assumptions of Constant or Variable Work Element Times," *Journal of Industrial Engineering*, Vol.16, No.1(1965), pp.23-29.
- [19] Rachamadugu, R. and B. Talbot, "Improving The equality of Workload Assignments in Assembly Lines," *International Journal of Production Research*, Vol.29(1991), pp.619-633.
- [20] Tonge, F.M., *A Heuristic Program for Assembly Line Balancing*, Prentice-Hall Englewood Cliffs, NT, (1961).
- [21] Urban, T.L., "Note. Optimal Balancing of U-Shaped Assembly Lines," *Management Science*, Vol.44, No.5(1998), pp.738-741.