

목표계획법을 위한 진화알고리즘: 양면조립라인 밸런싱 문제에 적용 An Evolutionary Algorithm for Goal Programming: Application to two-sided Assembly Line Balancing Problems

송원섭* 김여근**

전남대학교 산업공학과 *wonseop.song@gmail.com **kimyk@chonnam.ac.kr

Abstract

This paper presents an evolutionary algorithm for goal programming with preemptive priority. To do this, an evolutionary strategy is suggested which search for the solution satisfying the goals in the order of the priority. Two-sided assembly line balancing problems with multiple goals are used to validate the applicability of the algorithm. In the problems, three goals are considered in the following priority order: minimizing the number of mated-stations, achieving the goal level of workload smoothness, and maximizing the work relatedness. The proper evolutionary components such as encoding and decoding method, evaluation scheme, and genetic operators, which are specific to the problem being solved, are designed in order to improve the algorithm's performance. The computational result is reported.

1. 서론

이 논문에서는 선취적 우선순위(Preemptive Priority)를 가지는 다목표 문제에 적합한 다목표 진화알고리즘(Multi-Goal Evolutionary Algorithm: MGEA)를 제시한다. 목표계획법(Goal Programming: GP)은 여러 형태가 있으나[3, 7, 9], 이 연구에서는 우선 순위가 높은 목표식의 목표 미달치를 우선적으로 최소화한 후 다음 목표의 미달치를 최소화하는 선취적 GP를 고려한다.

진화알고리즘(Evolutionary Algorithm: EA)은 자연 선택(natural selection)과 유전 법칙(genetic rules)의 생물학적 진화과정을 모방한 일종의 확률적 탐색기법이다. GP문제에 EA를 적용한 연구는 현재까지 아주 미미하다. Deb[4]는 다목적 진화알고리즘과 같은 개념으로 각 목표의 편차변수를 최소화하는 문제, 즉 다목적 문제와 동일하게 보고 다양한 비지배해를 찾는 방법을 제안하였다. Zheng et al.[8]은 선취적 비선형 GP문제에서 높은 우선 순위를 만족하는 개체에 더 높은 순위(rank)를 주고, 이 순위에 따라 적응도를

부여하여 개체를 진화시켰다. 이와 같은 진화방법은 목적함수에 선취적 우선 순위 계수를 부여하여 적응도를 계산하는 방법과 유사하다. 이 연구에서는 높은 우선 순위부터 차례로 해 공간을 탐색하는 새로운 MGEA를 제안한다. 이를 위해 GP에 적합한 진화구조와 적응도 부여방법을 개발한다. 다목표를 갖는 양면조립라인 밸런싱(two-sided Assembly Line Balancing: two-ALB)문제를 사용하여, 제안한 MGEA의 적용성과 우수성을 검증한다.

2. 다목표 진화알고리즘

GP에서 높은 순서로 우선순위 목표 $g=1, 2, \dots, G$ 가 있다고 하자. 그리고 우선순위 g 의 목표식을 $f_g, g=1, 2, \dots, G$ 로 두자. 그리고

우선순위 1의 목표: Min. f_1

우선순위 2의 목표: Min. f_2

⋮

우선순위 G 의 목표: Min. f_G

가 된다. 그리고 우선순위 g 의 계수를 R_g 로 두면 선 취적 GP의 목적함수는

$$\text{Min. } F = R_1 f_1 + R_2 f_2 + \dots + R_G f_G$$

가 된다. 위 식에서 R_g 는 R_{g+1} 보다 아주 큰 수이다.

제안하는 MGEA는 우선순위가 높은 목표부터 차례로 해 공간을 탐색해 나간다. 이때 더 높은 우선순위를 만족하는 해가 사라지지 않으면서 낮은 우선 순위를 만족하는 해를 탐색할 수 있도록 설계되었다. 이를 위해 발견된 좋은 해를 저장하는 보관소를 운영한다.

제안하는 MGEA의 절차는 다음과 같다. 먼저 사용되는 기호를 정의한다.

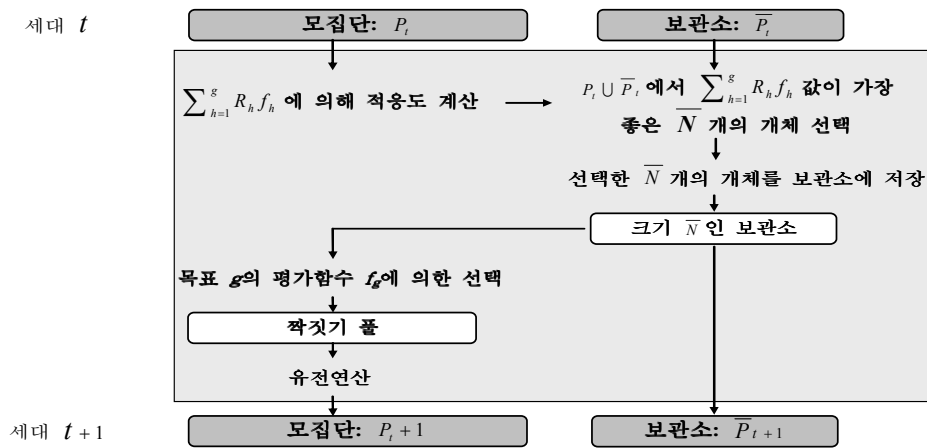
P_t : 세대 t 의 모집단

\bar{P}_t : 세대 t 의 보관소

N : 모집단의 크기

\bar{N} : 보관소 크기

NT_g : 우선 순위 g 에 대한 진화 종료 세대



[그림 1] MGEA의 흐름도.

- 단계 1: $t=0$, $g=1$ 로 둔다. 그리고 초기모집단 P_0 를 생성하고 보관소 $\bar{P}_0 = \phi$ 로 둔다.
- 단계 2: 평가함수 $\sum_{h=1}^g R_h f_h$ 에 기초하여, P_t 에 있는 개체의 적응도를 계산한다.
- 단계 3: $W_t = P_t \cup \bar{P}_t$ 에서 평가함수 $\sum_{h=1}^g R_h f_h$ 에 기초하여, 적응도가 가장 좋은 \bar{N} 의 개체를 \bar{P}_{t+1} 에 복사한다.
- 단계 4: $W_t = P_t \cup \bar{P}_t$ 의 개체에 대해, 목표 g 에 대한 목표식 f_g 에 의해 이진 토너먼트 선택을 사용하여 N 개의 개체로 구성된 짝짓기 풀(mating pool)을 만든다.
- 단계 5: 짝짓기 풀 내의 개체에 유전연산을 적용하여 N 개의 개체를 생산하여 다음 세대 모집단인 P_{t+1} 을 구성한다.
- 단계 6: 평가함수 $\sum_{h=1}^g R_h f_h$ 에 기초하여 P_{t+1} 에 있는 개체의 적응도를 계산한다.
- 단계 7: $t \leftarrow t+1$ 로 둔다. 그리고 $t = NT_g + 1$ 이면, $g \leftarrow$

$g+1$ 로 둔다.

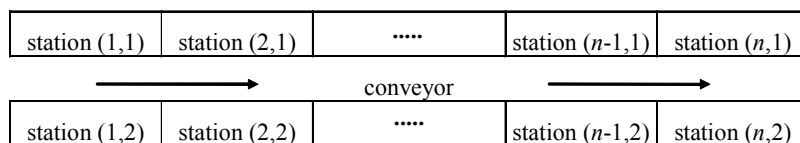
- 단계 8: $g > G$ 이면, $W_t = P_t \cup \bar{P}_t$ 에서 통합평가함수 $\sum_{h=1}^G R_h f_h$ 이 가장 좋은 개체가 해가 되고 끝낸다. 그렇지 않으면 단계 3으로 간다.

단계 3은 지금까지 발견된 좋은 해를 저장하는 과정이고, 단계4와 5는 지금까지 발견된 좋은 해를 가지고 현재의 우선순위 목표를 만족하는 해를 탐색하는 과정이다. [그림 1]은 MGEA의 흐름도이다.

3. 다목표 양면조립라인 밸런싱

3.1 양면조립라인

개발한 MGEA는 two-ALB 문제에 적용한다. 양면 조립라인은 [그림 2]와 같이 작업장이 조립라인의 왼쪽과 오른쪽의 양면에 병렬로 배치되는 생산 라인이다. 양면 조립라인은 트럭이나 버스처럼 조립 대상제품이 크고 작업의 병렬성(대칭성)이 높은 제품의 생산라인에서 흔히 볼 수 있다. [그림 2]에서 작업장 (1,1)과 작업장 (1,2)는 마주보는 두 작업장으로 '작업장 쌍(mated-station)'이라 한다.



[그림 2] 양면 조립라인.

단면 조립라인과 구별되는 양면 조립라인의 특징은 방향 제약을 갖는 작업이 존재한다는 것이다. 양면 조립라인에서 수행되는 작업들은 방향제약이 있는 왼쪽작업, 오른쪽작업, 그리고 왼쪽과 오른쪽 어느 작업장에서나 행할 수 있는 양쪽작업으로 분류할 수 있다. 따라서 양면 조립라인의 효율을 극대화하기 위해서는 작업할당이나, 설비 및 치공구의 배치에 있어서 작업의 방향이 고려되어야 한다. 작업 대칭성이 높고 작업장 내에서 작업자간 작업간섭이 일어나지 않는 경우, 양면 조립라인은 단면 조립라인보다 많은 이점을 갖는다[2].

3.2 two-ALB 문제의 선취적 우선순위와 목표식

이 연구에서 사용된 two-ALB 문제의 선취적 우선순위 목표는 다음과 같다.

(1) 제 1 우선순위 목표: 작업장쌍 수의 최소화

작업장 쌍 수의 최소화는 주어진 사이클타임에 대하여 작업이 할당된 작업장 쌍의 수를 최소화하는 것으로, 라인에 소요되는 공간과 노무비를 줄일 수 있다. 제 1 우선순위 목표 작업장 쌍의 수 최소화 목표식 f_1 은 다음과 같이 정의한다.

$$\text{Min. } f_1 = n \tag{1}$$

(2) 제 2 우선순위 목표: 작업부하의 평활화

작업부하의 평활화는 각 작업장에 가능한 균등하게 작업을 할당하여 작업자에게 공평한 여유시간을 주고 작업량에 대한 불만을 해소할 수 있다.

이 논문에서는 각 작업장에 할당된 작업의 최대작업량 (W_{\max})과 최소작업량 (W_{\min})의 차이를

평균작업부하 (\bar{w} = 총작업시간/2n)의 일정 수준(α , $0 < \alpha < 1$) 이내가 되도록 한다. 작업부하 평활화 인덱스(Index of Workload Smoothness: IWS)는 다음과 같은 척도에 의해 부여 받는다.

$$WR = W_{\max} - W_{\min}$$

$$IWS = \begin{cases} 0 & , \text{if } WR \leq \alpha \bar{w} \\ (WR - \alpha \bar{w}) / \bar{w} & , \text{if } WR > \alpha \bar{w} \end{cases}$$

작업부하의 평활화의 목표식 f_2 는 다음과 같다.

$$\text{Min. } f_2 = IWS \tag{2}$$

(3) 제 3 우선순위 목표: 작업관련성 최대화

작업관련성을 최대화하는 것은 직선행관계로 연결되어 있는 관련 작업들을 가능한 동일 작업장에 할당되도록 하여, 작업자의 작업효율을 높이고

조립라인의 설계자에게 설비와 작업장을 배치하는데 있어서 보다 큰 유연성을 제공한다.

작업장 쌍의 수가 n 일 때 한 작업장에 할당된 작업들간의 관련성을 평가하기 위하여, 아래와 같은 작업관련성 인덱스(Index of Work Relatedness: IWR)를 사용한다.

$$IWR = 1 - \frac{2n}{\sum_{j=1}^n \sum_{k=1}^2 SN_{jk}}$$

여기서 j 는 작업장 쌍을, k 는 왼쪽작업장(=1) 또는 오른쪽 작업장(=2)을 나타내는 인덱스이고, G_{jk} 는 작업장 (j, k)에 할당된 작업의 선행관계를 나타내는 네트워크이며, SN_{jk} 는 G_{jk} 에 나타난 연결된 네트워크 수이다. IWR은 [0, 1) 범위의 값을 갖고 값이 작을수록 작업관련성이 높은 작업할당으로 평가되는데 작업관련성 최대화 목표식 f_3 는 다음과 같다.

$$\text{Min. } f_3 = IWR \tag{3}$$

3.3 다목표 two-ALB 문제의 목적함수

two-ALB 세가지 목표는 식 (1), (2), (3)과 같은 각 목표식 f_g 를 갖는다. 따라서 다목표 문제의 목적함수 F 는 다음과 같이 정의된다.

$$\text{Min. } F = R_1 f_1 + R_2 f_2 + \dots + R_G f_G \tag{4}$$

이 연구에서는 $R_1=10^4, R_2=10^3, R_3=10^0$ 로 둔다.

4. two-ALB을 위한 진화요소

4.1 개체 표현

이 논문에서는 작업들이 작업장에 할당되는 상대적인 순서를 나타내는 순열표현을 사용하였다. 제시한 표현방법에서 개체의 인자 값은 작업번호를 나타내며 인자의 위치는 할당순서가 아니고 작업할당에 있어서 상대적 우선순위를 나타낸다. 예로, 표현 (3 2 6 5 1 12 9 8 4 11 7 10)은 첫번째 할당 가능 작업이 {1, 2, 3}인 경우, 할당규칙을 적용할 때 개체에 나열된 상대적 순서는 3, 2, 1이므로 작업 3을 우선적으로 선택한다는 것이다.

4.2 개체 해석

제안된 방법으로 표현된 개체는 작업들의 작업방향에 관한 정보를 포함하고 있지 않아, 해석 방법에 따라 다양한 해로 해석될 수 있다. 따라서 각 개체에 대해 작업장에 할당된 작업이 가능하면

빨리 종료될 수 있도록 양쪽작업의 작업방향과 작업장에 할당된 모든 작업의 작업순서를 결정할 필요가 있다.

개체 해석 절차는 다음과 같다.

단계 1: $j=1$ 로, $UT=I$ 로 둔다.

단계 2: UT 에 있는 작업 중 모든 선행작업이 할당된 작업집합 F 를 구한다. 그리고 각 작업 $i \in F$ 를 작업장 쌍 j 에 할당할 때 가장 이른 완료시간을 구하여, 이 시간이 사이클 타임 CT 보다 작은 작업들의 집합 FT 를 구한다. 만일 $FT = \emptyset$ 이면 단계 5로 간다.

단계 3: FT 에서 개체 표현의 가장 왼쪽에 나타난 작업을 i^* 로 선택한다.

단계 4: 작업 i^* 가 작업 방향을 가지면 해당 방향의 작업장에 할당한다. i^* 가 양쪽작업이면 시작 가능시점이 빠른 작업장에 할당한다. 이때 양면에서 이 시점이 같으면 작업장 쌍 j 에서 직선행 작업이 할당된 작업장에 할당한다. 이러한 작업장이 없거나, 양쪽 모두 할당되어 있으면 임의로 할당한다. 그리고 $UT \leftarrow UT - \{i^*\}$ 로 두고 단계 2로 간다.

단계 5: $UT \neq \emptyset$ 이면 $j \leftarrow j+1$ 로 두고 단계 2로 간다.

단계 6: f_g 와 F 를 계산하고 알고리즘을 종료한다.

4.3 선택과 유전연산자

MGEA의 단계 4에서 다음 세대에 생존할 개체는 토너먼트로 선택된다.

교차는 부분사상교차(Partially Mapped Crossover: PMX)를 사용한다. PMX는 임의로 두 절단점을 선택하여 두 절단점 사이에 있는 각 부모의 같은 위치에 있는 인자를 사상시켜 각 부모에서 사상된 인자와 상호 교환하는 교차이다[1].

돌연변이는 돌연변이율에 의해 선택된 개체에 대해 교환 돌연변이를 적용하였다. 교환 돌연변이는 선택된 개체내에서 임의의 두 인자를 선택하여 서로 교환하여 새로운 자손을 생산하는 방법이다.

5. 비교알고리즘과 실험 결과

MGEA의 성능분석을 위하여 작업의 수에 따라 각각 작업의 수가 65(P65)[6], 148(P148)[2]인 두 문제에 대해 사이클타임을 변화하여 실험하였다. 진화알고리즘에 필요한 파라메타는 예비실험을 통해 모집단과 보관소의 크기 N 과 \bar{N} 는 200으로, 교차율은 0.9, 개체돌연변이율은 0.4로 두었다. P65,

P148의 문제에 대해 각기 총 400, 800 세대까지 진화시켰다. 또한 NT_1 과 NT_2 는 각각 총 세대 수 $\times 0.5$ 와 총 세대 수 $\times 0.8$ 로, 작업부하 평활화의 목표 $\alpha = 0.05$ 로 두었다. 연구에서 제안한 MGEA의 성능은 비교 알고리즘을 통해 보였다. 모든 알고리즘은 C++언어를 사용하여 작성하였고, Pentium 4 CPU 3.20GHz를 가진 IBM-PC에서 실험하였다.

5.1 비교알고리즘

이 연구에서 제안한 MGEA의 성능은 (1) 새로이 제안한 발견적 기법(Heuristic Method: HM), (2) 단일 평가함수를 가지는 진화알고리즘(Evolutionary Algorithm with Single evaluate function: EAWS)과 비교를 통해 보였다. HM의 절차는 다음과 같다.

단계 1: $j=1$ 로, $UT=I$ 로 둔다.

단계 2: UT 에 있는 작업 중 모든 선행작업이 할당된 작업집합 F 를 구한다. 그리고 각 작업 $i \in F$ 를 작업장 쌍 j 에 할당할 때 가장 이른 완료시간을 구하여, 이 시간이 사이클 타임 CT 보다 작은 작업들의 집합 FT 를 구한다. 만일 $FT = \emptyset$ 이면 단계 5로 간다.

단계 3: FT 에서 임의의 할당규칙을 사용하여 작업 i^* 를 선택한다.

단계 4: 작업 i^* 가 작업 방향을 가지면 해당 방향의 작업장에 할당한다. i^* 가 양쪽작업이면 시작 가능시점이 빠른 작업장에 할당한다. 이때 양면에서 이 시점이 같으면 작업장 쌍 j 에서 직선행 작업이 할당된 작업장에 할당한다. 이러한 작업장이 없거나, 양쪽 모두 할당되어 있으면 임의로 할당한다. 그리고 $UT \leftarrow UT - \{i^*\}$ 로 두고 단계 2로 간다.

단계 5: $UT \neq \emptyset$ 이면 $j=j+1$ 로 두고 단계 2로 간다.

단계 6: IWS 그리고 IWR 을 구한다. 그리고 식 (4)의 F 값을 해로 두고 알고리즘을 종료한다.

HM의 단계 4에서는 다양한 작업 순서를 유도하기 위하여 이들 작업 중에서 (1) 최대 작업시간 (2) 최대 순위위치 가중치(작업 i 와 작업 i 의 후행작업의 작업시간의 합) (3) 후행작업의 수 (4) 최대 순위위치 가중치의 평균(최대순위위치 가중치/후행작업의 수) (5) 임의 선택의 할당 규칙을 사용하여 작업을 선택한다. 그리고 문제별로 10^5 번, 2×10^5 번 반복하여 가장 좋은 값을 해로 두었다.

EAWS는 식(4)의 목적함수 F 를 사용하여 개체를

<표 1> 알고리즘의 성능 비교

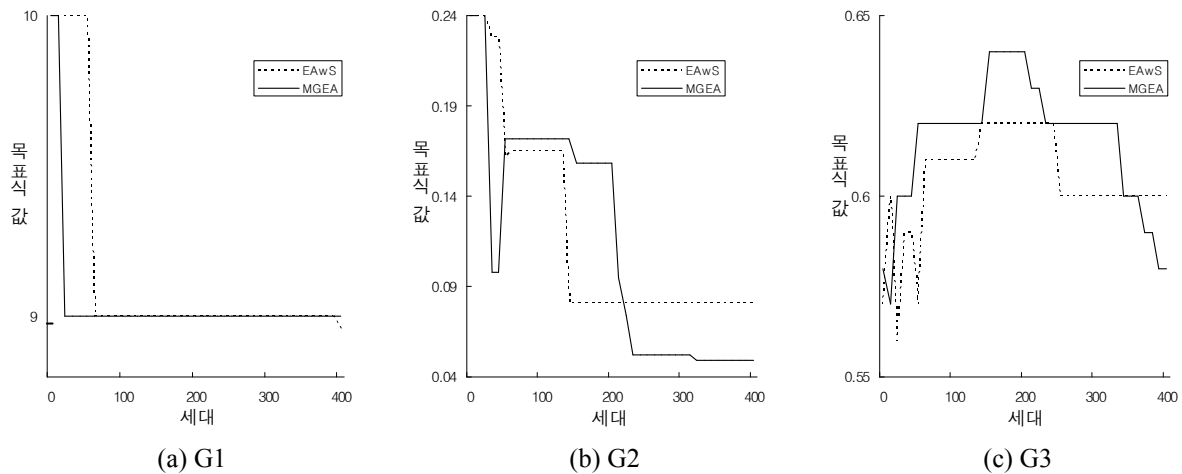
Problem	Cycle Time	\bar{n}	HM						EAwS						MGEA					
			Best			Mean			Best			Mean			Best			Mean		
			G1	G2	G3	G1	G2	G3	G1	G2	G3	G1	G2	G3	G1	G2	G3	G1	G2	G3
P 65	275	10	10.0	0.07	0.63	10.0	0.08	0.62	10.0	0.10	0.58	10.0	0.12	0.56	10.0	0.01	0.60	10.0	0.11	0.58
	300	9	9.0	0.04	0.65	9.0	0.05	0.64	9.0	0.06	0.58	9.0	0.10	0.59	9.0	0.02	0.61	9.0	0.06	0.59
	325	8	8.0	0.00	0.65	8.6	0.12	0.66	9.0	0.06	0.60	9.0	0.11	0.60	8.0	0.00	0.63	8.9	0.07	0.61
	350	8	8.0	0.06	0.67	8.0	0.09	0.68	8.0	0.04	0.64	8.0	0.09	0.62	8.0	0.00	0.63	8.0	0.04	0.62
	375	7	7.0	0.00	0.67	7.0	0.01	0.70	7.0	0.02	0.67	7.8	0.09	0.63	7.0	0.00	0.61	7.3	0.03	0.65
	400	7	7.0	0.05	0.67	7.0	0.08	0.70	7.0	0.24	0.67	7.0	0.09	0.66	7.0	0.00	0.63	7.0	0.05	0.65
	425	6	7.0	0.13	0.71	7.0	0.19	0.71	7.0	0.04	0.64	7.0	0.08	0.67	7.0	0.00	0.62	7.0	0.06	0.67
	450	6	6.0	0.00	0.73	6.0	0.01	0.73	6.0	0.03	0.67	6.0	0.05	0.68	6.0	0.00	0.65	6.0	0.01	0.69
	475	6	6.0	0.07	0.73	6.0	0.09	0.74	6.0	0.00	0.71	6.0	0.07	0.69	6.0	0.00	0.64	6.0	0.06	0.69
	500	6	6.0	0.15	0.73	6.0	0.19	0.75	6.0	0.05	0.65	6.0	0.07	0.68	6.0	0.00	0.65	6.0	0.04	0.69
P148	175	15	15.0	0.01	0.73	15.0	0.02	0.73	16.0	0.09	0.64	16.0	0.12	0.65	15.0	0.00	0.68	15.2	0.04	0.67
	200	13	13.0	0.01	0.75	13.6	0.11	0.75	14.0	0.08	0.67	14.0	0.10	0.69	13.0	0.00	0.71	13.9	0.07	0.68
	225	12	12.0	0.07	0.77	12.0	0.08	0.77	12.0	0.05	0.71	12.0	0.08	0.71	12.0	0.03	0.73	12.0	0.06	0.72
	250	11	11.0	0.09	0.79	11.0	0.10	0.79	11.0	0.10	0.74	11.0	0.12	0.72	11.0	0.01	0.73	11.0	0.09	0.73
	275	10	10.0	0.08	0.80	10.0	0.10	0.81	10.0	0.10	0.73	10.0	0.12	0.74	10.0	0.07	0.72	10.0	0.10	0.75
	300	9	9.0	0.05	0.81	9.0	0.06	0.82	9.0	0.07	0.75	9.0	0.08	0.76	9.0	0.03	0.75	9.0	0.05	0.75
	325	8	8.0	0.00	0.81	8.0	0.07	0.83	9.0	0.12	0.76	9.0	0.15	0.75	8.0	0.00	0.75	8.2	0.03	0.76
	350	8	8.0	0.11	0.83	8.0	0.12	0.83	8.0	0.10	0.77	8.0	0.13	0.79	8.0	0.00	0.79	8.0	0.11	0.78
	375	7	7.0	0.00	0.85	7.0	0.01	0.84	7.0	0.00	0.80	7.3	0.10	0.79	7.0	0.00	0.77	7.0	0.01	0.79
	400	7	7.0	0.08	0.85	7.0	0.10	0.85	7.0	0.09	0.77	7.0	0.12	0.80	7.0	0.03	0.79	7.0	0.09	0.80

평가하고 선택한다. 보관소를 운영하지 않고 모집단에 대해 선택, 재생산하는 알고리즘으로 여러 목표를 동시에 탐색하게 된다. EAwS에 사용되는 진화 파라미터는 MGEA와 동일하게 두었다.

5.2 알고리즘 성능 비교

HM과 EAwS, MGEA에 대한 실험결과는 <표 1>과 같다. 이 표에서 \bar{n} 는 이론적 작업장 쌍 수이고 Best, Mean은 비교알고리즘을 20회 반복 실험하여 얻은 가장 좋은 해와 그 평균이다. 그리고 G1, G2,

G3는 이들 해의 우선순위 목표 1, 2, 3의 값이다. 실험결과, MGEA가 모든 문제에서 좋은 해를 찾는 것으로 나타났다. 평균값에 대해서는 G1에 대해 MGEA가 HM보다 좋지 못한 경우가 있는데, 이는 HM이 작업 할당시 여유시간을 줄이는 것에 중점을 두고 있기 때문이다. 평균 계산소요시간은 P65, P148에서 HM이 각기 약 7.5초, 35초가, 그리고 EAwS와 MGEA 모두에서 각기 약 10초, 110초가 소요되었다.



[그림 3] EAwS와 MGEA의 목표별 진화속도 비교: P65, cycle time=300.

[그림 3]은 P65이고 사이클타임이 300인 문제에서 진화가 진행됨에 따라 EAwS와 MGEA의 목표 값의 변화를 나타낸 것이다. 이 그림에서 MGEA는 진화하는 동안 목표별 진화 세대 구간 (1~200세대, 201~320세대, 321~400세대)별로 높은 우선순위부터 차례로 해를 개선시킴을 알 수 있다. 그리고 보관소의 효과에 의해 우선 순위가 높은 목표의 해를 유지하며 [그림 3](b)에서 보이듯 G3를 향상시키는 세대에서 G2의 목표가 개선됨을 알 수 있다.

6. 결론

이 연구에서는 선취적 우선 목표를 갖는 목표계획법 문제를 해결하는 새로운 다목표 진화 알고리즘을 제안하였다. 이를 위해 목표계획법에 적합한 진화 구조와 적응도 부여방법을 개발하였다. 제안한 다목표 진화알고리즘은 작업장 쌍 수의 최소화를 제 1 우선순위 목표, 작업장간 작업부하의 평활화를 제 2 우선순위 목표, 작업관련성의 최대화를 제 3 우선순위 목표로 갖는 양면조립라인 밸런싱문제에 적용하여 효율성을 검증하였다. 다양한 실험결과 제안한 알고리즘이 기존의 발견적 기법보다 우수함을 알 수 있었다.

감사의 글: 이 논문은 2007년도 전남대학교 학술연구비 지원에 의해 연구되었음.

References

- [1] 김여근, 윤복식, 이상복 공저, 「메타 휴리스틱」, 영지문화사, 1997.
- [2] Bartholdi, J.J., "Balancing two-sided assembly lines: a case study", *International Journal of Production Research*, Vol.31, No.10(1993), pp.2447-2461.
- [3] Charnes, A., Cooper, W.W., *Management Models and Industrial Applications of Linear Programming*, Wiley, New York, 1961.
- [4] Deb, K., "Nonlinear goal programming using multi-objective genetic algorithms", *Journal of the Operational Research Society*, Vol.52, No.3(2001), pp.291-302.
- [5] Kim, Y.K., Kim, Y., and Kim, Y.J., "Two-sided assembly line balancing: A genetic algorithm approach", *Production Planning & Control* Vol.11, No.1(2000), pp.44-53.
- [6] Lee, T.O., Kim, Y., and Kim Y.K., "Two-sided assembly line balancing to maximize work relatedness and slackness", *Computers & Industrial Engineering*, Vol.40(2001), pp.273-292.
- [7] Tamiz, M., Jones, K., Romero, C., "Goal Programming for decision making: An overview of the current state-of-the-art", *European Journal of Operational Research*, Vol.111(1998), pp.569-581.
- [8] Zheng, DW., Gen, M., and Ida, K., "Evolution program for nonlinear goal programming", *18th International conference on Computers and Industrial Engineering*, Vol.31, No.3/4(1996), pp.907-911.
- [9] Özcan, U., Toklu, B., "Multiple-criteria decision-making in two-sided assembly line balancing: A goal programming and a fuzzy goal programming models", *Computers and Operations Research*, doi: 10.1016/j.cor.2008.06.009(2008).