

진화알고리즘을 이용한 선취적 다목표 양면조립라인 밸런싱*

송원섭** · 김여근**†

Two-Sided Assembly Line Balancing with Preemptive Multiple Goals Using an Evolutionary Algorithm*

Won Seop Song** · Yeo Keun Kim**

■ Abstract ■

This paper considers two-sided assembly line balancing with preemptive multiple goals. In the problem, three goals are taken into account in the following priority order : minimizing the number of mated-stations, achieving the goal level of workload smoothness, and maximizing the work relatedness. An evolutionary algorithm is used to solve the multiple goal problems. A new structure is presented in the algorithm, which is helpful to searching the solution satisfying the goals in the order of the priority. 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 results show that the proposed algorithm is promising in the solution quality.

Keyword : Two-Sided Assembly Line Balancing, Evolutionary Algorithm, Multiple Goals, Preemptive Priority

논문접수일 : 2008년 11월 19일 논문게재확정일 : 2009년 06월 02일

논문수정일(1차 : 2009년 03월 23일, 2차 : 2009년 04월 27일)

* 이 논문은 2007년도 전남대학교 학술연구비 지원에 의해 연구되었음.

** 전남대학교 산업공학과

† 교신저자

1. 서론

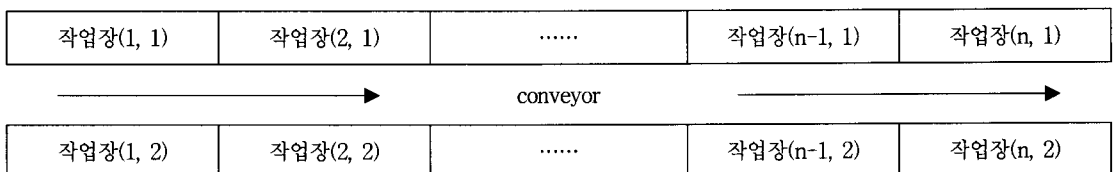
이 연구에서는 양면조립라인 밸런싱(two-sided assembly line balancing : two-ALB) 문제를 다룬다. 양면조립라인은 [그림 1]과 같이 작업장이 라인 진행방향 왼쪽과 오른쪽의 양면에 병렬로 배치되는 생산라인이다. 이러한 조립라인은 트럭이나 버스처럼 조립 대상제품이 크고 작업의 병렬성(대칭성)이 높은 제품의 생산라인에서 흔히 볼 수 있다. [그림 1]에서 작업장(1, 1)과 (1, 2)와 같이 마주보는 두 작업장을 ‘작업장 쌍’이라 부른다. two-ALB 문제는 양면에서 동시에 작업이 이루어지므로 한 면에서만 작업이 이루어 지는 단면조립라인과 비교하여 작업 방향, 상대작업장의 작업간섭 등의 제약이 추가되어 더 복잡하게 된다.

지금까지 two-ALB에 관한 여러 연구가 행하여졌다. Bartholdi[2]는 two-ALB에 관한 최초 연구로 간단한 할당규칙인 First Fit Rule(FFR)에 기초한 대화형(interactive) 프로그램의 설계와 사용에 관하여 연구하였다. Kim et al.[7]은 유전알고리즘을 이용하여 작업장 수의 최소화 문제를 다루었고, Lapierre와 Ruiz[9]는 작업의 높낮이가 서로 다른 두개인 two-ALB에서 작업장 수 최소화 문제를 우선순위에 기초한 발전적 기법(enhanced priority-based heuristic)으로 해결하였다. Baykasoglu and Dereli[3]는 특정 작업은 동일한 작업장 또는 서로 다른 작업장에서만 수행 가능한 지역 제한(zoning constraints)이 있는 작업장 수 최소화 문제를 개미군집이론을, Hu et al.[6]은 Hoffmann기법[5]을 이용하여 작업장 수 최소화 문제를 다루었다. Kim et al.[8]은 사이클 타임 최소화를 위한 수리모형과 진화알고리즘을 제

시하였다. 그러나 이들 연구는 단일 목적을 갖는 two-ALB 문제를 다루고 있다. 한편, Lee et al.[10]은 작업관련성과 작업여유를 최대화 하는 두 목적을 고려한 작업 할당규칙을 제안하였다.

이 논문에서는 선취적 우선순위(preemptive priority)를 가지는 다목적 two-ALB 문제를 위한 진화 알고리즘(Evolutionary Algorithm : EA)을 제안한다. two-ALB 문제에서 여러 목표를 가질 수 있다. 목표로는 작업장 쌍 수 또는 작업장 수(작업자 수)의 목표, 사이클 타임의 목표, 일정 수준 이상의 작업부하 평활화 목표, 가능한 관련작업을 동일 작업장에 할당하는 작업관련성의 목표 등이 있다. 선취적 우선순위는 상위의 목표가 하위의 목표보다 중요도가 아주 높아 하위의 목표를 달성하기 위해 상위의 목표를 희생해서는 안되는 순위이다. 조사에 의하면 지금까지 다목적 two-ALB를 다룬 연구로는 단지 Özcan and Toklu[12]가 있다. Özcan and Toklu [12]는 작업장 쌍 수의 목표, 사이클 타임의 목표, 한 작업장에 할당되는 작업 수의 제한 목표 순으로 선취적 우선순위를 가지는 퍼지 목표계획의 수리모형을 제시하였다. 이 수리 모형으로는 대형 문제를 푸는 데 한계가 있다. 본 연구에서 고려하는 선취적 우선순위는 차례로 (1) 작업장 쌍 수의 최소화 목표, (2) 작업부하 평활화 목표, (3) 작업관련성 목표로 둔다. 이들 우선순위로 이 연구에서는 문제의 크기가 큰 다목적 two-ALB 문제를 해결할 수 있는 선취적 다목적 진화알고리즘(preemptive Multiple Goal Evolutionary Algorithm : pMGEA)을 제안한다.

진화알고리즘은 자연선택과 유전법칙의 생물학적 진화과정을 모방한 일종의 확률적 탐색기법으로 여러 최적화분야에 적용되고 있으나 다목표를 갖는



[그림 1] 양면조립라인

목표계획법에의 적용은 현재까지 아주 미미하다[4, 11]. Zheng et al.[11]은 선취적 비선형 목표계획법 문제를 다루는 데 있어서 단순히 높은 우선순위를 만족하는 개체에 더 높은 순위(rank)를 주고, 이 순위에 따라 적응도를 부여하여 개체를 진화시킨다. Deb[4]는 다목적 진화알고리즘과 같은 개념으로 각 목표의 편차변수를 최소화하는 다양한 비지배해를 찾아 의사결정자의 선호에 의해 해를 선택하는 방법을 제안하였다.

이 논문에서 제안하는 pMGEA는 선취적 다목표 two-ALB를 위해 높은 우선순위부터 차례로 상위의 목표를 손상시키지 않으면서 하위의 목표를 탐색하는 새로운 진화알고리즘의 구조를 갖는다. 또한 새로운 진화 구조와 함께 선취적 다목표 문제에 적합한 적응도 부여방법과 다목표 two-ALB에 적합한 진화 요소들이 제안된다. 그리고 제안한 알고리즘의 성능을 검증한다.

이 논문의 구성은 다음과 같다. 먼저, 제 2장에서는 two-ALB에 관해 설명하고 목표를 기술한다. 제 3장에서는 다목표 two-ALB를 해결하기 위해 개발된 pMGEA를 제시하고, 그 구성요소를 설명한다. 제 4장에서는 알고리즘 성능 분석에 사용될 기법들을 제안한다. 제 5장에서는 실험 결과를 비교 분석하며, 마지막 장인 제 6장에서는 결론을 기술한다.

2. 다목표 양면조립라인 밸런싱

2.1 양면조립라인

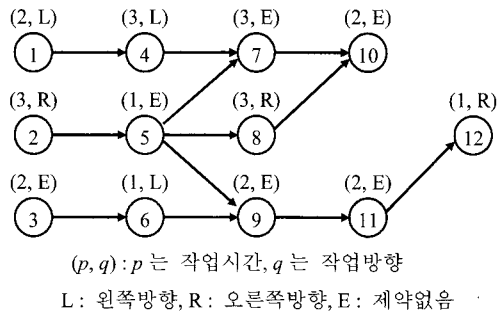
양면조립라인에서 수행되는 작업들은 방향제약이 있는 왼쪽작업(L), 오른쪽작업(R), 그리고 왼쪽과 오른쪽 어느 작업장에서나 행할 수 있는 양쪽작업(E)으로 분류할 수 있다. 예로, A자동차회사의 트럭 라인에서 연료탱크, 공기정화기, 공기상자 장착 작업은 왼쪽에서, 배터리, 공기탱크, 소음기 장착은 오른쪽에서 작업을 해야 한다. 그러나 액슬, 프로펠러 샤프트, 라디에이터 장착 등은 왼쪽 또는 오른쪽 어느 작업자가 하여도 작업 편의상 차이가 없는 작업

들이다.

또한, 양면조립라인에서는 한 작업장 쌍에 있는 두 작업자는 동시에 작업을 하기 때문에 맞은 편 작업장의 작업자에게 할당된 작업의 선행관계에 따라 작업간에 유ힴ시간이 발생할 수 있다. 따라서 양면조립라인에서는 작업할당 시 단면 조립라인과는 달리 작업할당과 함께 할당된 작업들의 작업순서가 고려되어야 한다. 또한 단면 조립라인에서 사이클 타임계약은 각 작업장에 할당된 작업들의 작업시간의 합이 사이클 타임을 넘지 않으면 된다. 그러나 양면조립라인에서는 작업간 유ힴ시간의 발생으로 인하여 각 작업장의 마지막 작업의 완료시점이 사이클 타임보다 작아야 한다.

작업 대칭성이 높은 대형 제품을 조립하는 양면조립라인은 단면 조립라인과 비교하여 여러 이점을 갖는다[2]. 양면조립라인은 단면 조립라인보다 라인 길이를 줄일 수 있고 제품의 완성 소요시간을 단축시킬 수 있다. 그리고 설비나 치공구를 양면에서 공유할 수 있어 비용을 절감할 수 있으며, 작업방향에 따른 작업자의 이동거리와 작업 준비시간을 줄일 수 있다.

[그림 2]는 각 작업의 작업시간과 선호 작업 방향을 나타낸 선행공정도의 예이다.



[그림 2] 작업의 선행공정도

2.2 two-ALB 문제의 선취적 우선순위와 목표

이 연구에서는 two-ALB 문제에서 아래와 같은

우선순위와 목표를 갖는다고 본다.

2.2.1 제 1우선순위 목표 : 작업장 쌍의 수 최소화

작업장 쌍 수의 최소화는 주어진 사이클 타임에 대하여 작업이 할당된 작업장 쌍의 수를 최소화하는 것으로, 라인 길이를 최소화하고 노무비를 줄일 수 있다. 일반적으로 원하는 수준의 작업장 쌍의 수를 목표로 할 수 있지만, 연구자가 양면조립라인을 분석한 결과 작업장 쌍을 최소로 하는 작업할당은 다양하게 존재할 수 있어 최소화를 목표로 사용하였다.

작업장 쌍의 수를 변수 n 로 둘 때, 제 1우선순위의 목표식 f_1 은 다음과 같이 정의된다.

$$\text{Min. } f_1 = n \quad (1)$$

2.2.2 제 2우선순위 목표 : 작업부하의 평활화 목표

작업부하의 평활화는 각 작업장에 작업량을 가능한 균등하게 할당하는 것으로 이는 작업자에게 공평한 여유시간을 주고 작업량에 대한 불만을 해소할 수 있다.

이 논문에서는 제 2우선순위로 작업장에 할당된 작업의 최대작업량, W_{\max} 과 최소작업량, W_{\min} 의 차이가 평균 작업부하량, $\bar{w} = \text{총 작업량}/2n$ 의 일정율, $0 < \alpha < 1$ 이내 둔다. 작업부하 평활화 인덱스(Index of Workload Smoothness : IWS)는 다음과 같은 척도에 의해 부여 받는다.

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

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

그러면, 작업부하의 평활화의 목표식, f_2 는 식 (2)와 같다.

$$\text{Min. } f_2 = IWS \quad (2)$$

2.2.3 제 3우선순위 목표 : 작업관련성 최대화

작업할당시 가능한 작업자에게 작업관련성이 높은 작업을 할당함으로써 작업 효율을 높일 수 있고, 작업 책임감과 함께 작업에 대해 만족감을 높일 수 있다[10]. 예로, 트럭조립라인에서 엔진 관련작업 그룹과 엑셀 관련작업 그룹이 있을 때 작업자는 한 그룹의 작업을 담당하는 것이 위와 같은 이유에서 바람직하다.

관련 작업을 선행공정도에 직접 연결되어 있는 작업으로 정의하자. SN_{jk} 는 작업장 (j, k)에 할당된 작업들의 선행관계를 표현한 연결된 네트워크의 수로 두자. 그리고 작업장 쌍의 수를 n 로 두자. 여기서 j 는 작업장 쌍을, k 는 왼쪽작업장(=1) 또는 오른쪽 작업장(=2)을 나타내는 인덱스이다. 작업관련성 인덱스(Index of Work Relatedness : IWR)를 아래와 같이 정의한다.

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

IWR 은 $[0, 1)$ 범위의 값을 갖고, 이 값이 작을수록 작업관련성이 높은 작업할당으로 평가된다. 예로, [그림 2]의 작업을 2개의 작업장 쌍에 할당한다면 다음 두 경우를 보자.

	작업장(1, 1)	작업장(1, 2)	작업장(2, 1)	작업장(2, 2)
할당 a :	{ 1 4 7 }	{ 2 5 8 10 }	{ 3 6 9 }	{ 11 12 }
할당 b :	{ 1 4 5 }	{ 2 3 7 }	{ 6 9 11 }	{ 8 10 12 }

할당 a는 $SN_{11} = SN_{12} = SN_{21} = SN_{22} = 1$ 이고, 할당 b는 $SN_{11} = 2, SN_{12} = 3, SN_{21} = 1, SN_{22} = 2$ 이다. 따라서 전자의 IWR 은 $\{1 - 4 / (1 + 1 + 1 + 1)\} = 0$ 이고, 후자의 IWR 은 $\{1 - 4 / (2 + 3 + 1 + 2)\} = 0.5$ 가 되어 전자가 후자보다 더 작업관련성이 좋은 작업할당으로 평가된다.

이 연구에서 작업관련성의 목표를 가장 낮은 우선순위로 두었다. 따라서 목표를 일정 수준 이하의

IWR이 아닌, 최소화로 둔다. 작업관련성의 최대화 목표식 f_3 는 식 (3)과 같이 둔다.

$$\text{Min. } f_3 = IWR \quad (3)$$

2.3 two-ALB 다목표 문제의 목적함수

two-ALB의 우선순위 목표 식은 식 (1), 식 (2), 식 (3)과 같다. 따라서 우선순위가 높은 목표를 달성한 해들 중에서 우선순위가 낮은 목표를 탐색하게 되어, 다목표 two-ALB 문제의 목적함수 F 는 식 (4)와 같이 표현할 수 있다.

$$\text{Min. } F = R_1f_1 + R_2f_2 + R_3f_3 \quad (4)$$

위 식에서 R_g 는 우선순위 g 의 선취적 계수로 R_g 는 R_{g+1} 보다 아주 큰 수이다.

3. 다목표 진화알고리즘

3.1 pMGEA의 절차

제안하는 pMGEA는 우선순위가 높은 목표부터 차례로 해 공간을 탐색해 나간다. 이때 보관소를 운영하여 높은 우선순위를 만족하는 해가 사라지지 않으면서 낮은 우선순위를 만족하는 해를 탐색할 수 있도록 알고리즘을 설계한다.

pMGEA의 절차는 다음과 같다. 먼저 사용되는 기호를 정의한다.

P_t : 세대 t 의 모집단

\bar{P}_t : 세대 t 의 보관소 모집단

N : 모집단의 크기

\bar{N} : 보관소 크기

g : 우선순위($g=1, 2, \dots, G$)

NT_g : 우선순위 g 에 대한 진화 종료 세대

단계 1 : $t=0, g=1$ 로 둔다. 그리고 초기모집단

P_0 를 생성하고 보관소 $\bar{P}=\phi$ 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.

pMGEA는 우선순위가 가장 높은 목표($g=1$)부터 차례로 해를 탐색해 나간다. 개체들은 평가함수 $\sum_{h=1}^g R_h f_h$ 에 의해 적응도 값을 부여 받고(단계 2, 단계 6), 지금까지 발견된 좋은 해를 보관소에 저장한다(단계 3). 단계 4는 다음 세대에 생존할 개체를 선택하여 짝짓기 풀을 구성하는 단계로, $P_t \cup \bar{P}_t$ 의 개체에 대해 현재 세대 t 의 진화목표 g 에 대한 목표식 f_g 에 의해 이루어진다. 짝짓기 풀내의 개체를 대상으로 유전연산을 적용함으로써 단계 4와 단계 5는 지금까지 발견된 좋은 해를 이용하여 현재의 우선순위 목표를 만족하는 해를 탐색하게 된다. 즉, 상위 목표를 만족하는 해들이 사라지지 않도록 보관소에 보관하고, 이들 해들에 대해 현재의 우선순위 목표를 개선하는 해를 탐색하도록 한다. 그리고 단계 7은 목표 별로 진화종료조건을, 단계 8은 알고리즘의 종료조건을 검사하는 단계이다. 세대 t 에서

목표 g 가 진화될 때, pMGEA의 흐름도는 [그림 3]과 같다.

3.2 개체 표현

진화알고리즘을 주어진 문제에 적용하기 위해서는 먼저 문제의 특성을 잘 반영할 수 있도록 해를 개체로 표현해야 한다. 개체표현은 유전연산과 관련되어 알고리즘의 성능에 크게 영향을 준다.

이 논문에서는 작업들이 작업장에 할당되는 상대적인 순서를 나타내는 순열표현을 사용한다. 이 표현에서 개체의 인자 값은 작업번호를 나타내며, 나열된 순서는 작업할당의 상대적 우선순위를 나타낸다. 여기서 할당의 상대적 우선순위의 의미는 개체의 해석 과정(3.3 개체 해석'의 단계 3 참조)에서 할당 가능 작업들 중에서 표현에서 가장 왼쪽에 나타난 작업을 선택하여 할당한다는 것이다. 예로, [그림 2]의 문제에서(5 3 4 6 2 7 1 12 9 8 11 10)로 표현된 개체가 있다고 하자. 할당 가능 작업이 {1, 2, 3}인 경우, 표현에서 작업 1, 2, 3 중에서 가장 왼쪽에 나타나는 작업이 작업 3이므로 이 작업을 먼저 할당하게 된다. 다음 할당 가능 작업이 {1, 2, 6}이면, 표현에서 이들 작업 중에서 작업 6이 가장 왼

쪽에 나타나므로 작업 6을 선택하여 할당하게 된다. 선택된 작업을 어떻게 할당할 것인지에 대해서는 다음 절에서 다룬다.

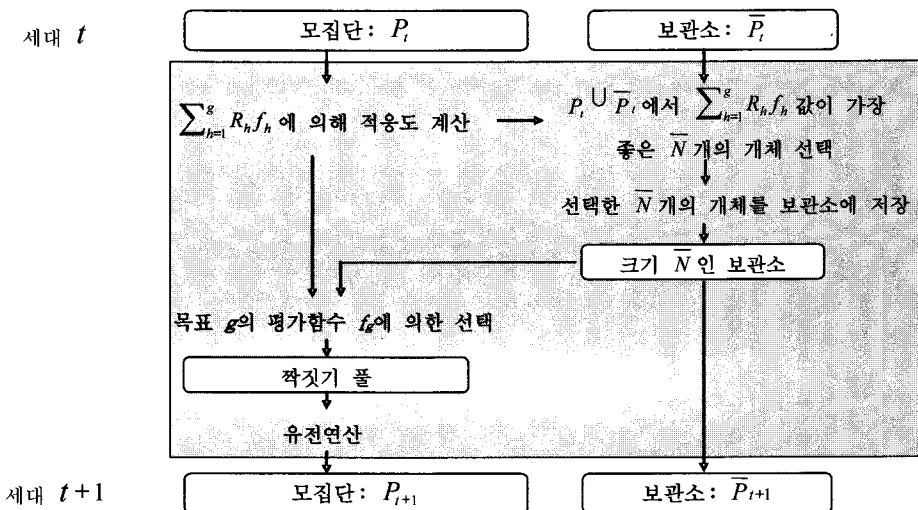
3.3 개체 해석

제안된 방법으로 표현된 개체는 양쪽작업들의 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. 개체의 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.



[그림 3] pMGEA의 흐름도

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 직 선행 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: 완성된 할당에 의해 개체를 평가하고 종료한다.

two-ALB에서 순열표현과 위와 같은 해석방법은 조립라인 밸런싱에서의 선행제약과 사이클타임제약을 만족하는 할당(해석절차의 단계 2와 단계 3 참조)과 양면 조립라인에서 작업 방향제약을 고려한 할당(단계 4 참조)을 가능하게 한다. 또한 이와 같은 표현과 해석은 순열표현에서 사용할 수 있는 다양한 유전연산자를 수정 또는 보수 없이 사용할 수 있다는 장점을 갖는다.

3.4 평가함수와 선택

단계 3에서는 평가함수 $\sum_{h=1}^g R_h f_h$ 에 의해 평가한 적응도에 의해, $W_i = P_i \cup \bar{P}_i$ 에서 가장 좋은 N 의 개체를 선택하여 \bar{P}_{i+1} 에 저장하고, 단계 4에서는 $W_i = P_i \cup \bar{P}_i$ 의 개체에 대해, 현재의 목표 g 에 대한 목표식 f_g 에 의해 평가한 적응도를 사용하여 이진 토너먼트 선택하여 다음 세대의 개체를 생산한다. 이는 상위의 목표를 달성한 개체들을 저장하는 한편, 현재 탐색하는 목표를 개선하기 위한 진화를 하게 하기 위함이다.

3.5 유전연산자

유전연산자에는 교차(crossover)와 돌연변이(muta-

tion)가 있다. 이 연구에서는 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. 예로 [그림 2]의 작업에서 두 부모와 절단점을 다음과 같이 두면, 두 절단점 부분 사이의 인자들은 $1 \leftrightarrow 3, 9 \leftrightarrow 6, 8 \leftrightarrow 7$ 이 된다. 부모 P1에서 사상된 인자들을 교환하면 자손 O1을 얻는다.

$$P1 = (3 \ 2 \ 6 \ 5 \mid 1 \ 9 \ 8 \mid 4 \ 11 \ 7 \ 10 \ 12)$$

$$P2 = (2 \ 5 \ 1 \ 4 \mid 3 \ 6 \ 7 \mid 8 \ 9 \ 11 \ 12 \ 10)$$

$$O1 = (1 \ 2 \ 9 \ 5 \ 3 \ 6 \ 7 \ 4 \ 11 \ 8 \ 10 \ 12)$$

같은 방법으로 아래 자손 O2를 얻는다.

$$O2 = (2 \ 5 \ 3 \ 4 \ 1 \ 9 \ 8 \ 7 \ 6 \ 11 \ 12 \ 10)$$

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

순열 표현을 위한 여러 형태의 교차와 돌연변이 방법이 있으나, 조립라인 밸런싱과 같은 그룹문제에서는 그룹 간 인자가 교환 또는 삽입될 수 있는 부분사상교차와 교환 돌연변이가 알고리즘의 성능에 좋은 결과를 보여[1] 이들 유전 연산자를 사용하였다.

4. 비교알고리즘

연구에서 제안한 pMGEA의 성능 검증을 위해 (1) 새로이 제안한 발견적 기법(Heuristic Method : HM)과 (2) 단일 평가함수를 가지는 진화알고리즘(Evolutionary Algorithm with Single evaluation function : EAWS)과 비교하였다.

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.

위 발견적 기법에서는 선행제약을 만족하는 작업을 구하여, 이들 작업 중에서 현재의 작업장 쌍에 할당가능한 작업들의 집합 FT 를 구한다(단계 2). 그리고 FT 에 있는 작업 중에서 할당규칙을 사용하여 할당할 작업을 선택한다(단계 3). 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. 선택된 작업은 작업방향을 고려하여 가장 빠른 시작 시점에 할당한다(단계 4). 그리고 위의 절차를 반복하여 가장 좋은 값

을 HM 의 해로 둔다.

우선순위의 순으로 진화하는 pMGEA의 진화구조와 적응도 평가방법의 효과를 검증하기 위하여, 식 (4)의 목적함수를 사용하여 적응도 값을 부여하고 전통적 진화 절차를 따르는 진화알고리즘(EAWS)과 성능을 비교 분석한다. EAWS의 절차는 다음과 같다.

단계 1: (초기 모집단) $t=0$ 으로 두고, 초기모집단 $P(0)$ 를 구성한다.

단계 2: (평가) $P(0)$ 의 모든 개체를 식 (4)에 의해 평가한다.

단계 3: (선택) $P(t)$ 에서 식 (4)의 평가값에 의해 $|P(t)|$ 개의 개체를 토너먼트 선택하여, $P(t+1)$ 에 복사한다.

단계 4: (교차 연산) 교차율에 따라 $P(t+1)$ 에서 교차에 참여할 두 부모 개체를 선택하고 교차연산을 적용하여 자손을 생산한다. $P(t+1)$ 에서 교차된 부모 개체를 제거하고 생산된 자손을 $P(t+1)$ 에 삽입한다.

단계 5: (돌연변이 연산) 개체 돌연변이율에 따라 $P(t+1)$ 에서 임의의 개체를 선택하여, 돌연변이 연산을 적용한다.

단계 6: (평가) $P(t+1)$ 의 모든 개체를 평가한다.

단계 7: (우수개체보존 전략) 만일 $P(t)$ 에서 가장 좋은 개체의 평가값이 $P(t+1)$ 에서 가장 좋은 개체의 평가값보다 더 좋다면, $P(t+1)$ 의 가장 나쁜 개체를 $P(t)$ 의 가장 좋은 개체로 대체한다.

단계 8: $t \leftarrow t+1$ 로 둔다. 그리고 종료조건을 만족하면 끝내고, 그렇지 않으면 단계 3으로 간다.

5. 실험 결과

5.1 실험설계 및 파라미터 설정

pMGEA의 성능분석을 위하여 작업의 수가 65개인 문제(P65)[10]와, 148개인 문제(P148)[2]에 대해

사이클 타임과 α 를 변화하여 실험하였다. 진화알고리즘에 필요한 파라미터는 예비실험을 통해 모집단과 보관소의 크기는 200으로, 교차율은 0.9, 개체 돌연변이율은 0.4로 두었다. 알고리즘의 종료조건은 문제 별로 해공간의 차이가 있으므로 P65, P148의 문제에 대해 각각 400, 800세대에서 종료하였다. 또한 NT_1 과 NT_2 는 각각 총 세대 수 \times 0.5와 총 세대 수 \times 0.8로, 작업부하 평활화의 목표의 계수, $\alpha = 0.05, 0.07, 0.10$ 로 두었다. 그리고 선취적 우선순위 계수는 $R_1 = 10^4, R_2 = 10^3, R_3 = 10^0$ 로 두었다. HM에서는 EA의 재생산된 개체 수와 비슷하게 각각 10^5 번, 2×10^5 번 반복하여 얻은 가장 좋은 값을 해로 두었다. 모든 알고리즘은 C++언어를 사용하여 작성하였고, Pentium 4 CPU 3.20GHz를 가진 IBM-PC에서 실험하였다.

5.2 알고리즘 성능 비교

제안한 pMGEA과 비교 기법인 HM과 EAWS에

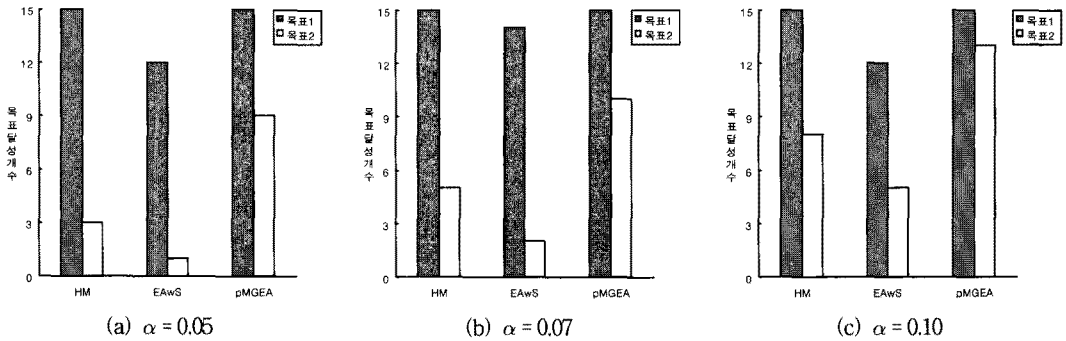
대한 실험결과는 <표 1>과 같다. 이 표에서 $\alpha = 0.05$ 이고, \bar{n} 는 이론적 작업장 쌍 수이고, 'Best', 'Mean'은 20회 반복 실험하여 얻은 가장 좋은 해와 그 평균이다. 그리고 G1, G2, G3는 이들 해의 우선순위 목표 1, 2, 3의 값이다.

실험결과, HM과 pMGEA는 모든 실험 문제의 'Best'해에서 우선순위 목표 1에 대해서는 이론적 최소 작업장 쌍 수를 찾았다. 이는 우선순위 목표 1을 성취하였음을 의미한다. 'Best'에 대해서는 모든 실험 문제에서 pMGEA가 비교 알고리즘보다 좋은 해를 찾았다. 'Mean'에 대해서는 15실험문제 중에서 3문제가 HM보다 pMGEA이 좋지 못하였다. 이는 HM이 작업 할당시 여유시간을 줄여 우선순위 목표 1에 중점을 두고 있기 때문인 것으로 판단된다. EAWS는 가장 좋지 않은 성능을 보였다. 이로부터 선취적 다목표 문제를 하나의 평가함수를 사용하여 해 공간을 탐색하는 데는 한계가 있음을 알 수 있다

비교 알고리즘의 평균 계산소요시간은 P65, P148

<표 1> 알고리즘의 성능 비교

Problem	Cycle Time	\bar{n}	HM						EAWS						pMGEA					
			Best			Mean			Best			Mean			Best			Mean		
			G1	G2	G3	G1	G2	G3	G1	G2	G3	G1	G2	G3	G1	G2	G3	G1	G2	G3
P65	300	9	9	0.04	0.65	9.0	0.05	0.64	9	0.06	0.58	9.0	0.10	0.59	9	0.02	0.61	9.0	0.06	0.59
	350	8	8	0.06	0.67	8.0	0.09	0.68	8	0.04	0.64	8.0	0.09	0.62	8	0.00	0.63	8.0	0.04	0.62
	400	7	7	0.05	0.67	7.0	0.08	0.70	7	0.24	0.67	7.0	0.09	0.66	7	0.00	0.63	7.0	0.05	0.65
	450	6	6	0.00	0.73	6.0	0.01	0.73	6	0.03	0.67	6.0	0.05	0.68	6	0.00	0.65	6.0	0.01	0.69
	500	6	6	0.15	0.73	6.0	0.19	0.75	6	0.05	0.65	6.0	0.07	0.68	6	0.00	0.65	6.0	0.04	0.69
P148	175	15	15	0.01	0.73	15.0	0.02	0.73	16	0.09	0.64	16.0	0.12	0.65	15	0.00	0.68	15.2	0.04	0.67
	200	13	13	0.01	0.75	13.6	0.11	0.75	14	0.08	0.67	14.0	0.10	0.69	13	0.00	0.71	13.9	0.07	0.68
	225	12	12	0.07	0.77	12.0	0.08	0.77	12	0.05	0.71	12.0	0.08	0.71	12	0.03	0.73	12.0	0.06	0.72
	250	11	11	0.09	0.79	11.0	0.10	0.79	11	0.10	0.74	11.0	0.12	0.72	11	0.01	0.73	11.0	0.09	0.73
	275	10	10	0.08	0.80	10.0	0.10	0.81	10	0.10	0.73	10.0	0.12	0.74	10	0.07	0.72	10.0	0.10	0.75
	300	9	9	0.05	0.81	9.0	0.06	0.82	9	0.07	0.75	9.0	0.08	0.76	9	0.03	0.75	9.0	0.05	0.75
	325	8	8	0.00	0.81	8.0	0.07	0.83	9	0.12	0.76	9.0	0.15	0.75	8	0.00	0.75	8.2	0.03	0.76
	350	8	8	0.11	0.83	8.0	0.12	0.83	8	0.10	0.77	8.0	0.13	0.79	8	0.00	0.79	8.0	0.11	0.78
	375	7	7	0.00	0.85	7.0	0.01	0.84	7	0.00	0.80	7.3	0.10	0.79	7	0.00	0.77	7.0	0.01	0.79
400	7	7	0.08	0.85	7.0	0.10	0.85	7	0.09	0.77	7.0	0.12	0.80	7	0.03	0.79	7.0	0.09	0.80	



[그림 4] 우선순위 목표 1과 2를 달성한 실험문제 수

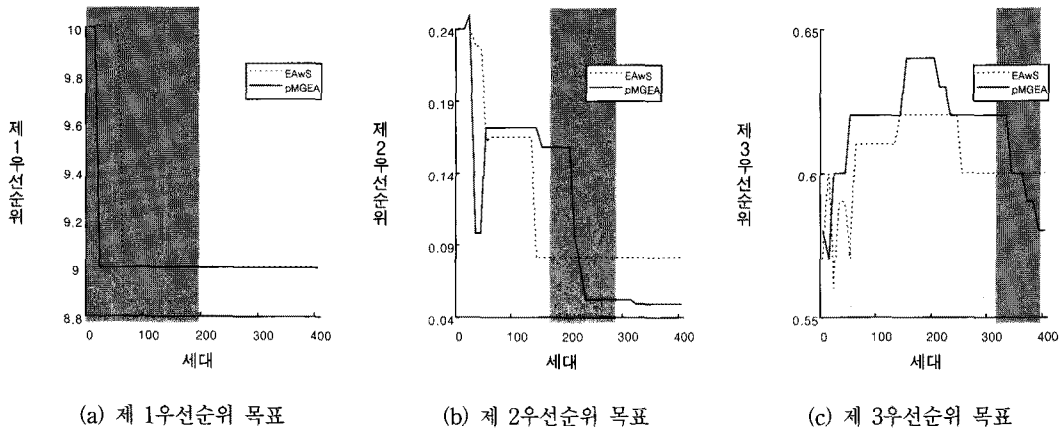
에서 HM이 각각 약 7.5초, 35초가, 그리고 EAWS와 pMGEA 모두에서 약 10초, 110초가 각각 소요되었다. 진화알고리즘 기반의 EAWS와 pMGEA에서는 모집단을 운영하면서 선택과 유전연산을 통해 해공간을 탐색하므로 단순한 여러 작업할당규칙을 사용하여 해를 반복적으로 생성하는 HM보다 계산 시간이 더 소요된다. 하지만 조립라인밸런싱은 라인의 설계 및 운영에 관한 계획문제이므로 이 연구에서 제안한 pMGEA의 계산소요시간은 실제 활용에서 수용될 수 있는 범위의 계산시간으로 판단된다.

[그림 4]는 $\alpha = 0.05$, $\alpha = 0.07$, $\alpha = 0.10$ 일 때, 15개의 실험문제에 대해 'Best'해가 우선순위 목표 1과 2를 달성한 문제의 수를 나타낸 것이다. [그림 4] (b)와 (c)는 앞의 실험과 동일한 조건으로 α 만을 α

$= 0.07$, $\alpha = 0.10$ 로 변경하여 실험한 결과이다.

이 그림으로부터 제안하는 pMGEA가 HM과 EAWS보다 더 성능이 좋음을 알 수 있다. 선취적 우선순위의 특성에 비추어 상위 우선순위가 달성되지 않은 상태에서 우선순위 목표 3에 대한 분석하는 것은 의미가 없다.

[그림 5]는 P65이고 사이클 타임이 300인 문제에서 진화가 진행됨에 따라 EAWS와 pMGEA의 목표값의 변화를 나타낸 것이다. 여기서 음영으로 나타난 부분은 각 목표별 진화 세대 구간(0~200, 201~320, 321~400세대)을 나타낸다. 이 그림에서 pMGEA는 구간 내에서 높은 우선순위부터 차례로 각 목표의 진화가 이루어짐을 알 수 있다. 또한 하위 목표의 진화가 이루어지는 동안에도 보관소에 의해 상위



[그림 5] EAWS와 pMGEA의 목표별 진화속도 : P65, 사이클 타임 = 300

목표에 대해 좋은 해는 유지되면서 진화함을 알 수 있다.

6. 결 론

이 연구에서는 진화알고리즘을 이용하여 선취적 우선순위를 갖는 다목표 양면조립라인 밸런싱 문제를 다루었다. 우선순위 목표로는 작업장 쌍 수의 최소화, 작업장간 작업부하의 평활화, 작업관련성의 최대화 순으로 3가지 목표를 고려하였다. 선취적 우선순위를 갖는 two-ALB에 적합한 진화알고리즘의 개발을 위하여, 선취적 다목표 계획법에 적합한 진화 구조와 적응도 부여방법을 개발하였다. 그리고 two-ALB에 적합한 유전 요소를 제시하였다. 제안한 알고리즘의 성능 분석을 위하여, 새로이 제안한 발견적 기법이 사용되었다. 특히, 제안한 진화알고리즘의 구조가 갖는 효과를 분석하기 위하여 고전적인 진화알고리즘과도 비교하였다. 여러 실험문제를 사용하여 알고리즘의 성능을 비교 분석한 결과, 제안한 알고리즘이 우수함을 알 수 있었고, 이로부터 선취적 다목표 문제에 적합한 진화구조를 가지고 있음을 알 수 있었다. 제안한 알고리즘은 일반화된 선취적 비선형 목표계획 문제에 적용될 수 있다.

참 고 문 헌

- [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] Baykasoglu, A. and T. Dereli, "Two-sided assembly line balancing using an ant-colony-based heuristic," *International Journal of Advanced Manufacturing Technology*, Vol.36, No. 5/6(2008), pp.582-588.
- [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] Hoffmann, T.R., "Assembly line balancing with a precedence matrix," *Management Science*, Vol.9(1963), pp.551-562.
- [6] Hu, X., E. Wu, and Y. Jin, "A station-oriented enumerative algorithm for two-sided assembly line balancing," *European Journal of Operational Research*, Vol.186, No.1(2008), pp.435-440.
- [7] Kim, Y.K., Y. Kim, and Y.J. Kim, "Two-sided assembly line balancing : A genetic algorithm approach," *Production Planning and Control*, Vol.11, No.1(2000), pp.44-53.
- [8] Kim, Y.K., W.S. Song, and J.H. Kim, "A mathematical model and a genetic algorithm for two-sided assembly line balancing," *Computers and Operations Research*, Vol.36, No.3(2009), pp.853-865.
- [9] Lapierre, S.D. and A.B. Ruiz, "Balancing assembly lines : an industrial case study," *Journal of the Operational Research Society*, Vol. 55(2004), pp.589-597.
- [10] Lee, T.O., Y. Kim, and Y.K. Kim, "Two-sided assembly line balancing to maximize work relatedness and slackness," *Computers and Industrial Engineering*, Vol.40(2001), pp.273-292.
- [11] Zheng, D.W., M. Gen, and K. Ida, "Evolution program for nonlinear goal programming," *18th International conference on Computers and Industrial Engineering*, Vol.31, No.3/4(1996), pp.907-911.
- [12] Özcan, U. and B. Toklu, "Multiple-criteria decision-making in two-sided assembly line balancing : A goal programming and a fuzzy goal programming models," *Computers and Operations Research*, Vol.36, No.6(2009), pp.1955-1965.