혼합모델 양면조립라인의 밸런싱과 투입순서를 위한 내공생 진화알고리즘*

조준영** · 김여근**†

An Endosymbiotic Evolutionary Algorithm for Balancing and Sequencing in Mixed-Model Two-Sided Assembly Lines

Jun Young Jo** · Yeo Keun Kim**

■ Abstract ■

This paper presents an endosymbiotic evolutionary algorithm (EEA) to solve both problems of line balancing and model sequencing in a mixed-model two-sided assembly line (MMtAL) simultaneously. It is important to have a proper balancing and model sequencing for an efficient operation of MMtAL. EEA imitates the natural evolution process of endosymbionts, which is an extension of existing symbiotic evolutionary algorithms. It provides a proper balance between parallel search with the separated individuals representing partial solutions and integrated search with endosymbionts representing entire solutions. The strategy of localized coevolution and the concept of steady-state genetic algorithms are used to improve the search efficiency. The experimental results reveal that EEA is better than two compared symbiotic evolutionary algorithms as well as a traditional genetic algorithm in solution quality.

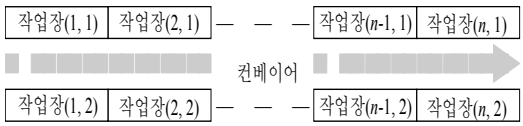
Keyword : Two-Sided Assembly Line, Mixed Models, Balancing and Sequencing Problem, Endosymbiotic Evolutionary Algorithm. Endosymbionts

논문접수일 : 2012년 04월 19일 논문게재확정일 : 2012년 07월 24일
논문수정일(1차 : 2012년 06월 14일)
* 이 논문은 2010년도 전남대학교 학술연구비 지원에 의하여 연구되었음.
** 전남대학교 공과대학 산업공학과
† 교신저자

1. 서 론

혼합모델 양면조립라인(mixed-model two-sided assembly line : MMtAL)은 [그림 1]과 같이 컨베이어의 왼쪽과 오른쪽의 양면에 작업장이 배치되는 생산라인에서 유사한 여러 모델의 제품을 생산하는 조립라인이다. MMtAL은 트럭이나 버스처럼 조립대상제품이 크고 작업의 병렬성(대칭성)이 높은 제품의 생산라인에서 흔히 볼 수 있다. 또한 냉장고와 세탁기를 생산하는 조립라인의 특정 일부 라인에서도 사용되고 있다. MMtAL에서 마주보는 두 작업장을, 예로 [그림 1]에서 작업장(1, 1)과 작업장(1, 2)를, '작업장쌍(mated-stations)'이라 부르고, 작업장쌍의 각 작업장에서 마주 보는 작업장을 '상대작업장'이라 부르자.

양면조립라인은 한 면에서만 작업하는 단면조립라인과 비교하여 작업이 방향성을 갖고 상대작업장 작업으로 인한 작업간섭이 발생할 수 있다. 또한 혼합모델조립라인은 단일모델라인과는 달리 모델에 따라 생산량과 총 작업시간이 다르고 부분적으로 작업내용, 작업시간, 작업방법, 자재 및 이용하는 설비와 치공구가 다를 수 있다.



[그림 1] 양면조립라인

MMtAL에서 밸런싱문제와 투입순서문제는 라인의 효율적 이용에 중요한 문제이다[25]. 조립라인 밸런싱은 제품 생산에 요구되는 작업들을 작업장에 할당하는 문제이고, 투입순서 결정은 조립라인에 투입하는 모델의 순서, 즉 생산순서를 결정하는 문제이다. 작업할당에 따라 모델의 적절한 투입순서가 달라지고, 투입순서에 따라 효율적인 작업할당이 달라진다. 따라서 밸런싱문제와 투입순서문제는 상호 밀접한 관련성을 갖는다[2].

양면 조립라인 밸런싱에 관한 연구는 Bartholdi [5]에 의해 최초로 연구되었다. Bartholdi는 양면 조립라인 밸런싱을 위한 대화형(interactive) 프로그램의 설계와 사용에 관하여 연구하였다. 이 연구[5]에서는 간단한 할당규칙인 First Fit Rule을 양면조립라인 밸런싱문제에 적용하였다. 이후 Kim et al.[14]에서는 양면조립라인 밸런싱에서 양면작업장 수의 최소화 문제를 위한 유전알고리즘을 제시하였으며, Lee et al.[17]은 작업관련성과 작업여유성의 척도를 제시하고, 이들의 최대화를 위한작업그룹 단위의 할당규칙을 제안하였다. Baykasoğlu and Dereli[6]와 Simaria and Vilarinho[24]은 개미군체최적화에 기반을 둔 양면조립라인 밸런싱을 위한 발견적 기법을 제안하였고, Lapierre et al.[16]와 Őzcan and Toklu [20]는 Tabu search를 이용하였다. Hu et al.[9]은 작업장 기준의 열거형 할당규칙을 제시하고, 이 할당규칙과 Hoffmann 기법을 이용하여 양면조립라인을 밸런싱하였다. Kim et al.[15]은 양면조립라인의 사이클타임 최소화를 위한 수리 모형을 최초로 제시하였으며, 이 수리모형에 기초하여 Őzcan and Toklu [21]는 다기준을 갖는 양면조립라인 밸런싱을 위한 목표계획 수리 모형을 제시하였다. 또한 송원섭, 김여근[3]은 선취적 우선순위를 갖는 다목표 양면조립라인 밸런싱을 위한 진화알고리즘을 제안하였다. 이들 연구는 모두 한 모델만 생산하는 단일모델 양면조립라인 밸런싱 문제를 다루고 있다.

여러 모델을 생산하는 혼합모델의 양면조립라인 밸런싱에 관해서는 시뮬레이티드 어닐링(simulated annealing) 기법을 사용한 Őzcan and Toklu[22]의 연구가 있다. 하지만 그는 모든 모델에 대해 하나의 일정한 사이클 타임이 주어진 상태에서 작업장쌍 수의 최소화를 목적으로 하고 있다.

이 연구에서는 혼합모델 양면조립라인에서 밸런싱문제와 투입순서문제를 동시에 고려한다. 앞에서 언급했듯이 MMtAL에서 밸런싱과 모델 투입순서는 라인의 효율적 활용을 위하여 중요한 문제이고 상호 관련되어 있다. 하지만 저자의 조사에 의하면, MMtAL에서 밸런싱과 투입순서를 동시에 다룬 연구는 아직까지 이루어 지지 않고 있다. 이 연구에서

는 두 문제를 동시에 다룰 수 있는 기법으로 진화알고리즘의 특수한 형태인 내공생 진화알고리즘(endosymbiotic evolutionary algorithm)을 사용한다. 진화알고리즘은 자연선택과 유전법칙의 생물학적 진화과정을 모방한 일종의 메타휴리스틱이다[1]. 따라서 이 연구에서는 내공생 진화알고리즘의 적용을 위하여 다루는 문제에 적합한 진화알고리즘 구조와 진화 방법을 제안하고, 밸런싱과 투입순서를 위한 유전표현, 개체해석, 유전연산자를 소개한다. 또한 여러 실험문제를 가지고 제안한 알고리즘의 성능을 분석한다.

2. 혼합모델 양면조립라인의 밸런싱과 투입순서

2.1 혼합모델 양면조립라인(MMtAL)

이 연구에서 다루는 혼합모델 양면조립라인은 일정한 속도로 이동하는 컨베이어 생산라인으로 유사한 여러 모델의 제품이 일정 간격으로 라인에 투입되며, 컨베이어의 이동에 따라 조립라인의 양면에서 각 작업자가 이동하면서 작업을 하는 라인이다.

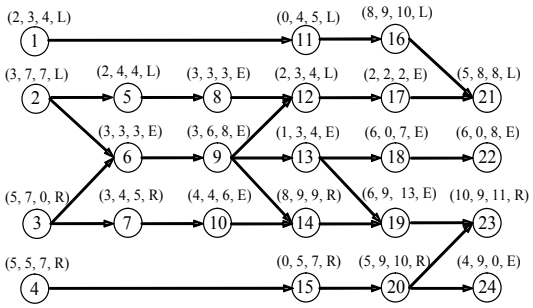
양면조립라인에서 수행되는 작업들은 단면조립라인과 달리 방향 제약을 가지고 있다. 양면조립라인에서는 왼쪽 작업(L), 오른쪽 작업(R), 그리고 왼쪽과 오른쪽 어느 작업장에서나 행할 수 있는 양쪽작업(E)으로 분류할 수 있다.

단면조립라인에서는 작업의 선행관계를 만족하여 작업할당이 이루어지므로 동일작업장에서 행하여 지는 작업 간에는 유휴시간(idle time)이 발생하지 않는다. 그러나 양면조립라인에서는 어떤 작업장의 작업의 미완료로 인해 그 상대작업장의 작업 간에 유휴시간을 발생시킬 수 있다. 예로, 어떤 작업장쌍에서 작업 i의 직선행작업 p가 상대작업장에 할당되면, 작업 i가 시작할 수 있는 시점은 작업 p가 완료된 시점 이후가 된다. 따라서 양면조립라인에서는 작업할당 시 단면조립라인과는 달리 작업할당과 함께 할당된 작업들의 작업순서가 고려되어야 한다.

양면조립라인에서 작업 대칭성이 높은 대형 제품을 생산할 때 단면조립라인과 비교하여 여러 이점을 갖는다[5]. 양면조립라인은 단면조립라인 보다 라인길이를 줄일 수 있고 제품의 완성 소요시간을 단축시킬 수 있다. 그리고 설비나 치공구를 양면에서 공유할 수 있어 비용을 절감할 수 있으며, 작업방향에 따른 작업자의 이동거리를 줄일 수 있다.

혼합모델 양면조립라인은 다음과 같은 상황에서 운영된다고 본다. 첫째, 양면조립라인과 관련된 가정들이다. 직선형 컨베이어 생산라인으로 일정속도(v_c)로 이동하며, 여러 모델의 제품들이 일정 시간간격(γ)으로 투입된다. 라인은 J개의 폐쇄 작업장쌍(closed mated-stations)으로 구성되어 있고, 양 작업장에서는 왼쪽 작업장과 오른쪽 작업장에 각자 1명씩 작업을 수행한다. 폐쇄 작업장은 주어진 각 작업장의 영역을 벗어나서 작업할 수 없는 작업장이다. 둘째, 양면조립라인은 혼류로 생산하는 사이클 생산방식을 취한다. 생산계획기간 동안에 M종류의 모델에 대한 각 수요는 $D_m(m = 1, 2, \cdots, M)$로 두고, 이의 최대 공약수를 h라 하자. 그리고 벡터 $\boldsymbol{d_m} = (d_1, \cdots, d_M)$, $d_m = D_m/h$ 로 두자. 즉, 사이클당 각 모델의 수요($d_1, \cdots, d_M$)를 반복 생산하는 사이클 생산을 사용한다. 여기서 각 모델의 수요, $\boldsymbol{d_m}$를 최소제품집합(minimum part set : MPS)이라 하며, 이를 반복 생산하게 된다[4]. 즉, MPS를 h번 반복 생산하여 총 수요를 만족시킨다. 셋째, M개 모델의 작업 선행관계는 I개의 작업으로 구성된 결합선행공정도(combined precedence diagram)에 의해 표현된다고 본다. 끝으로 작업자의 이동시간은 무시한다.

[그림 2]는 일례의 결합선행공정도로 이 그림은 작업의 선행관계와 작업시간, 선호하는 작업 방향을 나타낸다. 이 결합선행공정도에서 마디(node)는 작업을 나타내고, 마디 위에 있는 수치는 모델별 작업시간과 작업방향을 의미한다. 예로 마디 1위에 있는 수치(2, 3, 4, L)는 모델 A, B, C에서 작업 1의 작업시간이 각각 2, 3, 4임을 나타내고, 작업방향은 왼쪽(L)을 나타낸다, 작업 방향은 오른쪽, 왼쪽, 양쪽을 각각 R, L, E로 나타낸다.



[그림 2] P24의 결합선행공정도

먼저 이 논문에서 MMtAL의 밸런싱과 투입순서 문제를 기술하는 데 사용되는 인덱스와 입력자료의 기호를 정의한다. 이 기호는 Bard et al.[4]가 사용한 기호를 주로 채택하였으며, 그 외의 기호는 설명의 편의를 위해 본문에서 정의하기로 한다.

인덱스

i : 작업; $i=1, \cdots, I$(작업의 총 수는 I)

j : 작업장 쌍; $j=1, \cdots, J$(작업장쌍의 총 수는 J)

k : 작업장의 방향. $k=1$; 왼쪽 작업장, $k=2$; 오른쪽 작업장

m : 모델 형태; $m=1, \cdots, M$(모델의 총 수는 M)

s : 모델의 투입 순서; $s=1, \cdots, S$

m_s : s번째 투입되는 모델

입력 자료

$I, J, M, S(=\sum_{m=1}^{M} d_m)$

v_c : 컨베이어 속도

d_m : MPS 생산 기간의 모델 m의 수요

t_{im} : 모델 m에서 작업 i의 작업시간

$\hat{t}_i$: 작업 i의 MPS 작업시간(= $\sum_{m=1}^{M} d_m t_{im}$)

L_j : 작업장쌍 j의 작업장 길이

γ : 제품의 투입 시간간격

2.2 목적함수

이 연구에서는 제 2.1절에서 언급한 가정하에 가외작업(utility work)을 최소로 하는 MMtAL의 밸런싱과 투입순서를 동시에 구하는 문제를 다룬다. 가외작업은 작업자가 주어진 작업장 내에서 작업을 완수하지 못한 작업량이다. 이 작업은 예비 작업인력, 현실적으로 작업 반장이나 조장이 이를 행한다고 본다. 가외작업의 최소화는 노동 비용뿐 아니라 컨베이어의 정지 위험을 줄이는 데 기여한다. 이는 한정된 길이의 생산라인에서 생산율을 높이는 효과를 갖는다. 또한 가외작업의 최소화는 작업장의 작업량을 평활화하는 경향을 갖는다[13]. 가외 작업은 작업할당과 모델 투입순서에 모두 영향을 받는다.

양면조립라인에서 가외작업은 왼쪽 작업장과 오른쪽 작업장을 모두 고려해야 한다. 작업할당과 투입순서가 주어지면 MPS 생산기간 동안 가외작업은 식 (1)과 표현할 수 있다.

$$Min.\ U=\sum_{j=1}^{J}\sum_{k=1}^{2}\left[\sum_{s=1}^{S}\max\left\{0,(Z_{sjk}+v_c f_{jkm_s}-L_j)/v_c\right\}\right\} + Z_{(S+1)jk}/v_c] \quad (1)$$

여기서 Z_{sjk}는 작업장쌍 j의 작업장 k(k = 1은 왼쪽 작업장, k = 2는 오른쪽 작업장)에서 s번째 투입모델의 작업시작 위치이다. f_{jkm_s}는 작업장쌍 j의 작업장 k에서 s번째 투입모델(m_s)의 작업량이다. 여기서 주의할 점은 이 작업량에는 상대작업장의 선행관계에 의해 발생하는 작업간의 유휴시간이 포함된다. L_j는 작업장쌍 j의 길이이다. 따라서 $Z_{sjk}+v_c f_{jkm_s}$는 작업장쌍 j의 작업장k에서 s번째 투입모델(m_s)의 완료 위치가 되므로 이 값이 작업장쌍 j의 길이 L_j보다 크면, 즉, $Z_{sjk}+v_c f_{jkm_s}-L_j$이 0보다 크면 가외작업이 발생한다. 따라서 $\max\lfloor 0, (Z_{sjk}+v_c f_{jkm_s}-L_j)/v_c \rfloor$은 작업장쌍 j의 작업장 k에서 s번째 투입모델의 가외 작업시간이 된다.

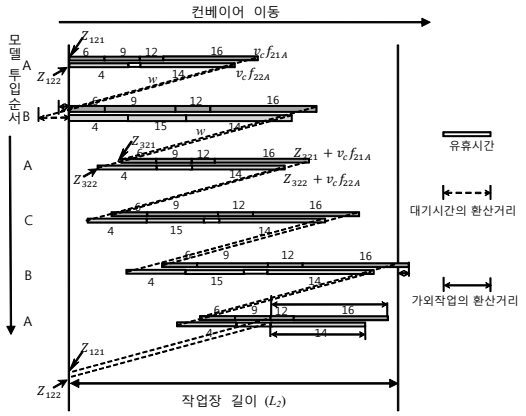
모든 작업장에서 생산 사이클(MPS)의 첫 번째 투입모델의 작업 시작위치는 기준점으로 둔다. 즉, $Z_{1jk}=0,\ \forall j, k$이 된다. 그러나 일반적으로 이 조건은 만족되지 않는다. 따라서 마지막 모델 다음의 시작점이 기준점에 오지 못하면 그 만큼 다음 MPS 생산에서 가외작업이 발생한다. 두 번째 항은 이러한 가

정을 고려하였다.

작업장쌍 j의 작업시작 위치는 아래와 같이 계산된다.

$$Z_{(s+1)jk} = \max\{0,\ \min(Z_{sjk} + v_c f_{jkm_s} - w,\ L_j - w)\}$$
$$s = 1, 2, \cdots, S,\ k = 1, 2 \qquad (2)$$

여기서, $Z_{1jk} = 0,\ j = 1, 2, \cdots, J,\ k = 1, 2$이고, $w(= v_c \gamma)$는 투입간격 동안의 컨베이어의 이동거리이다. 앞에서 언급했듯이 $Z_{sjk} + v_c f_{jkm_s}$는 작업장쌍 j의 작업장 k에서 s번째 투입모델의 작업이 끝나는 지점이므로 이 값이 작업장쌍 j의 길이 L_j보다 크면 가외 작업이 이루어지므로 지점 L_j에서 끝내게 된다. 따라서 다음 모델의 시작 지점은 $\min(Z_{sjk} + v_c f_{jkm_s} - w,\ L_j - w)$이 된다. 그런데 $Z_{sjk} + v_c f_{jkm_s} - w$이 0보다 작으면 앞 작업장의 영역이 되므로 기준점, 즉 지점 0에서 시작하게 된다. 이때 제품이 작업장 영역에 도달할 때까지 대기시간(waiting time)이 발생하게 된다. 이를 수리적으로 표현하면 식 (2)와 같다.



[그림 3] 폐쇄작업장에서의 가외작업

[그림 3]의 결합선행공정도를 사용하여 예를 보자. 하나의 작업할당으로 작업장쌍 2의 왼쪽 작업장에 작업 6, 9, 12, 16의 순으로 오른쪽 작업장에 작업 4, 15, 14의 순으로 작업할당이 되었다 하자. 그리고 모델의 투입순서가(A B A C B A)이고 $v_c = 1$로 두자. 이때 [그림 3]은 가외작업, 유휴시간, 대기시간을 도식한 예이다. 모델 A에서 유휴시간이 발생하며, 그 길이는 투입순서에 따라 달라짐을 알 수 있다.

3. MMtAL의 밸런싱과 투입순서를 위한 내공생 진화알고리즘

공생 진화알고리즘은 여러 부분 문제들이 상호 관련되어 복잡도가 높은 문제를 통합적으로 해결할 수 효율적인 기법으로 알려져 있다[1]. 공생 진화알고리즘은 생물의 공진화 과정을 모방한 탐색 기법으로, 이 알고리즘에서는 종들은 분리된 형태로 상호 작용하고 적응하면서 진화한다[23]. MMtAL에서 밸런싱문제는 분할문제에 속하고 투입순서문제는 일종의 순열문제이다. MMtAL에서 이 두 문제는 상호 관련되어 있다. 이 연구에서는 다루는 문제의 해결을 위해 공생 진화알고리즘을 확장한 내공생 진화알고리즘(endosymbiotic evolutionary algorithm : EEA)[10, 11]을 채용한다.

3.1 내공생 진화알고리즘의 개념과 구조

내공생 진화알고리즘(EEA)은 Lynn Margulis[18, 19]의 내공생(endosymbiosis) 진화이론을 모방한 탐색기법이다. 내공생 진화이론은 원핵생물(prokaryote)에서 진핵생물(eukaryote)로의 진화과정을 설명한다. 이 이론은 비교적 작은 크기의 원핵생물이 보다 큰 숙주 원핵생물로 들어가, 이들이 공생하면서 진핵생물로 진화했다는 것이다. 내공생 진화론은 강력한 정황적 증거로 진핵생물(eukaryote)에서 미토콘드리아(mitochondria)나 엽록체(chloroplast)와 같은 세포 기관을 든다. 오늘날의 미토콘드리아와 엽록체는 원핵생물과 닮은 점이 많다. 그들은 적은 양의 DNA, RNA, 리보좀을 갖고 있는데 이들은 진핵생물보다는 원핵생물에 가까운 특징을 갖고 있다. 또한, 내공생 진화론은 미토콘드리아와 엽록체가 이중막으로 싸여 있는 이유를 설명해 줄 수 있는 가

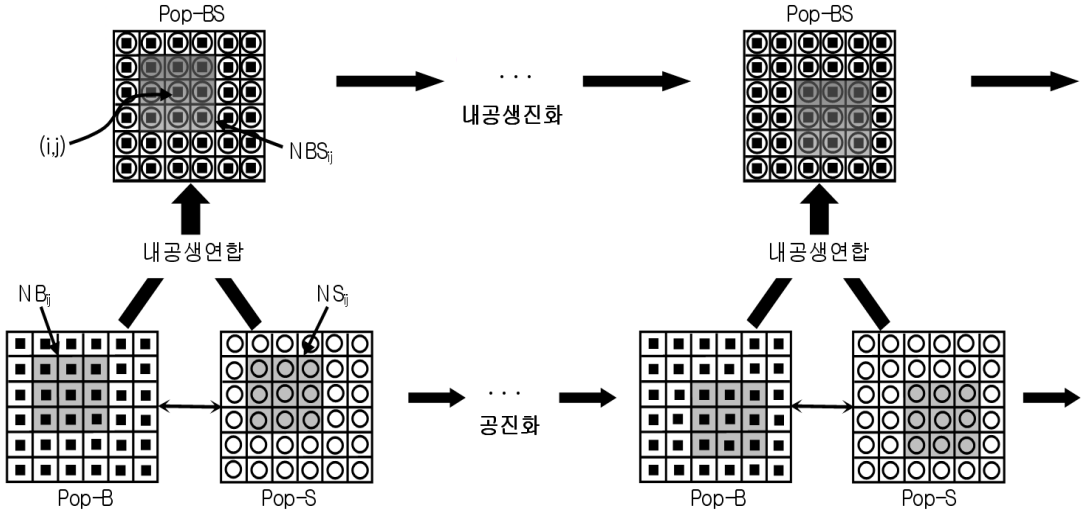
설이기도 하다.

공생진화알고리즘에 관한 대부분의 기존 연구는 공생하는 개체들이 결합되지 않고 분리된 형태로 공생한다고 본다[1]. 이 경우, 개체들이 진화하는 동안 공생자의 변동 빈도가 높게 되어 다양한 해 공간의 탐색은 가능하나 주어진 탐색 방향에서 충분한 탐색이 이루어지지 않는다는 단점을 갖는다. 이러한 단점을 보완하기 위하여 제안한 내공생 진화알고리즘은 각 개체가 자신과 결합하여 높은 적응도를 유도하는 공생자를 만나면, 이들을 결합된 형태로 진화시킨다.

[그림 4]와 같이 내공생 진화알고리즘은 분리된 형태로 공생하는 두 모집단, 즉 밸런싱 개체로 이루어진 모집단 Pop-B와 투입순서의 개체로 이루어진 모집단 Pop-S, 그리고 이들 두 모집단의 개체가 각각 결합한 형태의 개체로 이루어진 모집단 Pop-BS로 구성된다. 즉, Pop-BS는 완전해를 나타내는 내공생자(endosymbiont)로 이루어지며, 이러한 내공생자는 두 공생자의 게놈(genome) 결합에 의해 이루어진다. Pop-BS의 개체들은 결합된 형태로 Pop-B와 Pop-S의 모집단에서 이웃 공진화를 통하여 나온 가장 좋은 적응도를 갖는 개체인 잠재내공생자와 경쟁하며, 동시에 자신의 모집단 내에서 진화한다. 그러므로 내공생 진화알고리즘은 Pop-B와 Pop-S에 있는 부분해들에 의한 병렬탐색과 Pop-BS에 있는 완전해에 의한 통합탐색이 동시에 이루어진다.

또한, 제안하는 내공생 진화알고리즘에서는 모집단 내와 모집단 간의 상호작용이 지역에 국한하여 이루어지는 이웃 진화전략이 사용된다. 이는 모집단의 다양성 유지와 적소(niche) 형성을 돕는다. 이것들은 결국 지역최적으로의 조기 수렴을 막아 장기적으로는 해의 개선을 유도할 수 있다. 지역 내공생 진화를 위하여 모집단과 이웃은 다음과 같이 구성된다. 세 모집단 모두 2차원 토러스(torus) 격자구조를 갖는다. 여기서 이웃은(3×3)의 정사각형으로 한다. NB_{ij}, NS_{ij}와 NBS_{ij}는 각기 모집단 Pop-B, Pop-S와 Pop-BS에서 위치(i, j)에 있는 개체와 그 주위의 8개의 셀(3×3의 정사각형)을 포함하는 이웃을 나타낸다.

초기에 Pop-B와 Pop-S의 각 셀에는 임의로 생성한 부분해로 이루어진 개체들을 할당하고, Pop-BS는 임의로 생성된 완전해로 이루어진 개체들을 할당한다. NB_{ij}와 NS_{ij}에 있는 개체는 공생하면서 진화한다. NB_{ij}와 NS_{ij}의 공생은 공생 진화알고리즘에



[그림 4] 내공생 진화알고리즘

서와 동일한 과정을 수행한다. 이때 공생자는 상대 모집단의 이웃에서 가장 좋은 적응도를 갖는 개체를 사용한다. NB_{ij}와 NS_{ij}에 있는 공생자의 적응도를 평가할 때 발견한 가장 높은 적응도를 갖는 개체 결합을 $a_p b_q$라 하자. 여기서 a_p와 b_q는 각기 NB_{ij}와 NS_{ij}에 있는 한 개체를 나타낸다. $a_p b_q$는 잠재 내공생자가 된다. 잠재 내공생자의 적응도 $f(a_p b_q)$가 NBS_{ij}에 있는 가장 낮은 적응도인 $f(\alpha_v \beta_v)$보다 크면, NBS_{ij}에 있는 $\alpha_v \beta_v$를 $a_p b_q$로 대체한다. 그리고 NBS_{ij}도 진화한다.

이러한 내공생 진화알고리즘의 수행을 위해서는 다루는 문제에 적합한 유전요소가 필요하다. 먼저 제 3.2절에서는 밸런싱 모집단, Pop-B의 진화에 필요한 주요 유전요소로, 유전표현, 개체 해석, 초기모집단 구성, 교차와 돌연변이의 방법을 다루고, 제 3.3절에서는 Pop-S의 진화에 필요한 주요 유전 요소를 다룬다.

3.2 MMtAL에서 밸런싱을 위한 유전요소

3.2.1 유전표현과 개체 해석

진화알고리즘에서 유전표현은 다루는 문제의 특성을 반영하면서 가능한 자연스럽고, 명료하며, 비중복적이어야 한다[1]. 따라서 이 연구에서 밸런싱 모집단 Pop-B에서는 그룹번호표현(group-number representation)을 사용한다. 밸런싱 모집단의 개체 길이는 I(작업의 수)이며, 각 인자는 1과 J(작업장쌍 수)사이의 정수 값을 갖는다. 즉, i번째 인자값이 j이면 작업 i는 작업장쌍 j에 할당된다. 예를 들어 (1 2 3 1 1 2 3 2 4 4 3)으로 표현된 개체는 작업장쌍 1, 2, 3, 4에 각각 {1, 4, 5}, {2, 6, 8}, {3, 7, 11}, {9, 10}의 작업이 할당됨을 나타낸다. 이러한 그룹번호표현은 작업장쌍 수가 주어진 문제의 경우에 적용할 수 있으며, 작업장쌍의 정보를 명확히 나타내는 장점을 가지고 있다. 그러나 이 표현으로는 할당된 작업들의 작업순서와 E-type 작업의 할당 작업장(왼쪽 작업장 또는 오른쪽 작업장) 정보를 알 수 없다. 이 표현은 이를 해결하기 위한 개체 해석(decoding) 방법이 필요하나, 해석을 위한 발견적 기법을 사용하면 국부적으로 좋은 해를 찾을 수 있다.

작업장쌍 그룹번호표현을 위한 아래와 같은 개체 해석 방법을 사용한다.

<개체 해석 절차>

단계 1 : 작업장쌍 j에 할당된 작업집합을 U_j로 둔다. 작업 i의 시간은 MPS 생산기간 동안의 작업시간 $\hat{t}_i$로 보고 계산한다.

단계 2 : U_j에서 모든 선행작업이 할당된 작업집합 Q를 구한다.

단계 3 : 이 집합 Q에서 가장 일찍 시작할 수 있는 작업을 구한다. 이들 작업이 여러 개 있으면, MPS 생산의 최대 순위위치가중치(ranked positional weight)를 사용한다. 선택된 작업을 i^*로 둔다.

단계 4 : 작업 i^*가 방향을 가지면, 즉 L-type이나 R-type이면 그 방향의 작업장에 할당하고 단계 7로 가고, E-type이면서 왼쪽과 오른쪽 작업장의 가장 이른 시작시간이 다르면 이른 시작시점을 갖는 작업장에 할당하고 단계 7로 간다.

단계 5 : 작업 i^*의 모든 직후행작업이 L-type(R-type)이면 왼쪽 작업장(오른쪽 작업장)에 할당하고 단계 7으로 간다.

단계 6 : 미할당 작업들의 L-type 작업과 R-type 작업의 MPS 생산의 총 작업시간을 각각 구하여 작업시간이 작은 쪽에 할당한다. 같으면 임의로 할당한다.

단계 7 : $U_j \leftarrow U_j - \{i^*\}$로 두고 $U_j = \varnothing$이면 종료하고, 그렇지 않으면 단계 2로 간다.

단계 3에서 작업장쌍 j에 할당된 작업i의 MPS 생산의 순위위치가중치는 MPS를 생산하는 데 요구되는 작업 i와 작업 i의 모든 후행작업(작업장쌍 j에 할당된 작업 중에서 후행작업)의 작업시간을 합한 시간을 의미한다. 이 절차에서 가장 이른 시작 시간은 각 작업의 소요시간을 MPS 생산의 작업소요시

간으로 보고 계산한다. 이 해석 기법에서는 특정 작업장쌍에 할당된 작업들의 작업순서와 E-type 작업의 할당 작업장을 결정하는 데, 선행제약을 만족하면서 상대작업장의 작업에 따른(선행제약으로 인해 발생하는) 유휴시간을 줄이면서 왼쪽과 오른쪽 작업장의 작업완료시점의 차를 줄이려는 데 초점을 두고 있다.

3.2.2 초기모집단

초기모집단은 탐색 공간을 넓히기 위하여 가능한 다양한 개체로 구성되는 것이 바람직하다. 초기모집단 개체는 다음과 같은 절차로 생성하며, 이를 모집단 크기만큼 반복한다.

<초기모집단 생성 절차>

단계 1 : 첫 번째 작업장쌍, j = 1을 생성한다. 그리고 SUM = 0로 두고, 선행 작업이 없는 할당가능한 작업들의 집합 AT로 둔다.

단계 2 : 집합 AT에서 임의로 작업 p를 선택하고, $SUM \leftarrow SUM + \hat{t}_p$한다. 만약 $SUM \leq \sum_{i=1}^{m} \hat{t}_p / J$이면 작업 p를 작업장쌍 j에 할당한다. 그렇지 않으면 $j \leftarrow j+1$, SUM = 0로 두고 0.5의 확률로 작업 p를 작업장쌍 j 또는 j-1에 할당한다.

단계 3 : j = J이면 남은 작업을 마지막 작업장쌍 J에 할당하고 끝낸다. 그렇지 않으면 단계 4로 간다.

단계 4 : 미할당 작업집합에서 모든 선행작업이 할당된 작업들의 집합 AT을 만들고 단계 2로 간다.

이와 같은 절차로 밸런싱 모집단을 구성하는 것은 작업의 선행제약을 만족하면서 가능한 작업장에 평균 작업량이 할당되는 다양한 개체를 생산하기 위함이다.

3.2.3 교차와 돌연변이

교차연산자는 문제의 특성을 잘 반영하여 부모 개체가 지닌 좋은 정보를 자손에게 전달 할 수 있어야 한다. 이를 위하여 교차는 부모가 갖는 그룹(작업장쌍)의 정보를 자손에게 전달하는 구조교차(structural crossover)[26]를 기반으로 한다. 밸런싱 문제에서 선행제약을 만족하면서 부모의 할당 정보가 자손에게 전달되는 변형된 구조교차 방법을 사용한다. 이 교차는 다음과 같이 행한다.

<교차 방법>

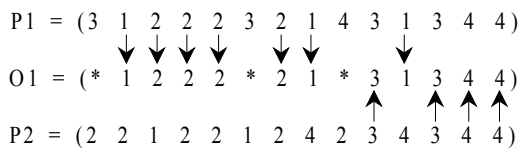
단계 1 : 범위 [1, J]에서 임의의 정수 r을 선택한다.

단계 2 : 부모 P1에서 1부터 r의 값을 갖는 인자를 자손 O1의 동일한 위치에 상속한다.

단계 3 : 자손 O1의 남은 위치에 부모 P2의 같은 위치에서 r+1부터 J의 값을 갖는 인자를 상속한다.

단계 4 : 자손 O1의 비어있는 위치에는 재할당 기법을 이용하여 작업을 배정한다.

다른 자손 O2는 P1과 P2의 역할을 바꾸어 생산한다. [그림 5]는 위 절차를 도식한 것이다. 4개의 작업장쌍 중 r = 2인 경우, 부모 P1에서 1과 2의 값을 갖는 인자들을 자손개체 O1의 같은 위치에 복사하고, 미할당 위치들에서 부모 P2의 인자 값이 3 또는 4인 인자를 같은 위치에 상속한다. [그림 5]에서 작업장쌍이 할당되지 않은 작업 1, 6, 9은 *로 표시하고 재할당 방법을 이용하여 할당한다.



[그림 5] 구조교차

돌연변이는 해공간의 다양성을 유지하고 부분최적에 조기수렴하는 것을 방지 하는 역할을 한다. 밸런싱문제에서는 개체 돌연변이율(R_{im})에 의해 선택된 개체에 대하여 인자 돌연변이율(R_{gm})에 따라 인자를 돌연변이 시킨다. 인자의 돌연변이 방법은 교

차에서 사용하는 재할당방법을 채용한다.

교차와 돌연변이 연산에서 사용하는 재할당방법의 설명을 위해 먼저 사용되는 기호를 정의한다.

$\overline{T}$: 한 MPS 사이클 동안 작업장쌍 부하의 평균 $(=(1/J)\sum_{i=1}^{I}\hat{t}_i)$

T_j^* : 작업장쌍 j에 이미 할당된 MPS 작업시간의 합

UA : 아직 미할당된 작업의 집합

$IP(i)$: 작업 i의 직선행작업의 집합

$S(i)$: 작업 i의 후행작업의 집합

$NR(i)$: $UA \cap IP(i)$(즉, UA에 있는 작업 중에서 작업 i의 직선행작업의 집합)

e_i : $IP(i)$에 있는 작업이 할당된 작업장쌍 중에서 가장 늦은 작업장쌍

$IP(i) = \varnothing$이면, $e_i = 1$

l_i : $S(i)-UA$의 작업이 할당된 작업장쌍 중에서 가장 이른 작업장쌍

$S(i)-UA = \varnothing$이면, $l_i = J$

$AW(i)$: $\{e_i, e_i+1, \cdots, l_i\}$(즉, 작업 i가 할당 가능한 작업장쌍의 집합)

<재할당 절차>

단계 1 : UA와 $NR(i)$, $i \in R$를 구성하고, T_j^*, $j=1, 2, \cdots, J$를 계산한다.

단계 2 : $NR(i) = \varnothing$, $i \in UA$인 작업 중에서 가장 큰 MPS 작업시간을 갖는 작업을 선택한다. MPS 작업시간이 같으면 작업번호가 작은 것을 선택한다. 선택된 작업을 작업 i^*로 둔다.

단계 3 : e_{i^*}와 l_{i^*}를 구하고, $AW(i^*)$을 구성한다.

단계 4 : $T_j^* + \hat{t}_{i^*} \le \overline{T}$와 $j \in AW(i^*)$을 만족하는 작업장쌍이 존재하면, 가장 이른 작업장쌍을 j^*로 둔다. 그렇지 않으면, 가장 작은 T_j^*를 갖는 작업장쌍을 j^*로 둔다. 이때 여러 작업장쌍이 존재하면 가장 작은 작업장쌍 번호를 선택한다.

단계 5 : 작업 i^*를 작업장쌍 j^*에 할당한다. UA에서 작업 i^*를 제거한다. UA가 공집합이면 끝낸다. 그렇지 않으면 $NR(i)$와 T_j^*, j = 1, 2, …, J를 갱신하고 단계 2로 간다.

위 절차에서 단계 1에서 단계 3까지는 선행제약을 만족하는 할당을 위한 과정이다. 특히, 단계 2에서 작업시간이 가장 큰 작업의 선택은 best-fit-decreasing(BFD) 규칙에 기초한 것이다[8]. 단계 4는 작업장의 작업량 평활화를 고려하면서, 다음 작업의 할당 영역을 높이기 위해 가장 빠른 작업장에 할당하고 있다.

3.3 MMtAL에서 투입순서를 위한 유전 요소

이 절에서는 투입순서를 위한 모집단, Pop-S의 진화에 요구되는 유전 요소로 유전표현과 유전연산자를 다룬다.

3.3.1 유전표현

투입순서 결정을 위한 모집단의 개체들은 MPS 생산 동안 라인에 투입하는 순서대로 열거한 순서 리스트로 표현한다. 예로, MPS가 $d_m = (d_A, d_B, d_C) = (3, 2, 1)$이고 투입순서가 A, B, C, B, A, A이면 개체는 (A B C B A A)로 표현된다. 이 표현은 투입 순서대로 표현되어 개체의 해석과 유전자 정보의 이용이 용이하다는 장점이 있다. 초기 모집단은 MPS 제약을 만족하는 개체들을 모집단 크기만큼 임의로 생성하여 구성한다.

3.3.2 교차와 돌연변이

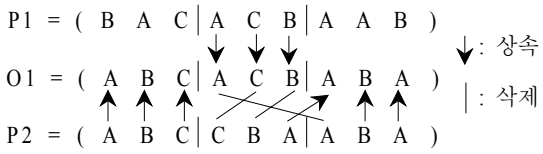
투입순서 문제에서 유전연산자로는 수정 순서교차(Modified order crossover operator)와 역순(inversion) 연산자를 사용한다. 이들 두 연산자는 혼합모델 조립라인의 투입순서 문제에서 좋은 탐색성능을 보였다[12]. 수정 순서교차는 순서교차[7]를 변형한 것으로 부모들의 상대적 순서를 보존한다. 이 교차는 다음과 같이 행한다.

단계 1 : 부모개체에 임의의 두 절단점을 지정한다.

단계 2 : 부모 P1의 두 절단점 사이에 있는 인자를 자손 O1의 같은 위치에 상속 한다

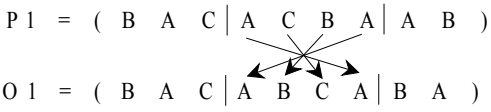
단계 3 : 부모 P2에서 O1에 상속된 인자를 임의로 삭제한 후, 부모 P2에 남은 인자를 앞에서부터 자손 O1의 미할당 인자에 차례로 상속시킨다.

다른 자손 O2는 P1과 P2의 역할을 바꾸어 생산한다. [그림 6]은 위 절차의 예이다.



[그림 6] 수정 순서교차

역순 돌연변이는 부모에서 두 개의 절단점을 임의로 선택하고 절단점 사이의 인자들을 역순으로 하여 자손을 생산하는 방법이다. 예를 들어 [그림 7]과 같이 부모 P1에서 두 절단점이 임의로 선택되면 절단점 사이의 인자들을 역순으로 하여 자손 O1을 생산한다. 이때 돌연변이 될 개체는 개체 돌연변이율(R_{im})에 의해 선택되고 인자 돌연변이율(R_{gm})은 영이 된다.



[그림 7] 역순 연산자

3.4 Pop-BS를 위한 유전 요소

밸런싱과 투입순서를 함께 나타낸, 즉 완전해의 개체로 이루어진 모집단, Pop-BS의 유전요소는 제 3.2절과 제 3.3절에서 다룬 방법을 그대로 채용한다. 즉, 개체 표현은 밸런싱과 투입순서의 개체 표현을 결합하여 사용한다. 이는 [그림 8]처럼 개체가 두 염색체로, 즉 밸런싱을 표현한 염색체와 투입순서를 나타낸 염색체로 구성된다. 두 개의 부분해가 결합되어 표현된다. [그림 8]은 [그림 2]의 선행공정도를 예로 사용한 것이다.

[그림 8]의 표현을 제 3.2절의 개체해석 방법에 의해 해석하면 할당 작업장과 작업순서는 다음과 같이 된다. 왼쪽과 오른쪽 작업장에 작업장쌍1에는 작업 (2, 1, 9)과 (3, 6), 작업장쌍 2에는 작업 (5, 8, 12, 13)과 (7, 10, 14), 작업장쌍 3에는 작업 (11, 16, 17)과 (4, 15, 20), 작업장쌍 4에는 작업 (19, 21, 22)과 (18, 24, 23)이 각각 할당된다.

Pop-BS에서 유전연산은 밸런싱과 투입순서 문제에 해당하는 염색체에 각기 제 3.2절과 제 3.3절에서 사용된 교차와 돌연변이 연산자를 적용한다.

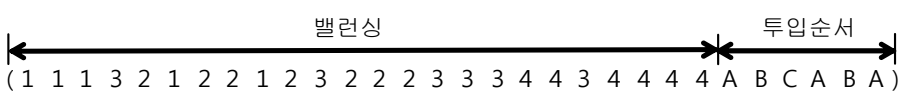
3.5 내공생 진화알고리즘의 절차

제 3.1절에서 설명한 내공생 진화알고리즘의 개념과 구조를, 그리고 제 3.2절~제 3.5절에서 이를 구현하기 위한 진화요소를 다루었다. 이를 바탕으로 제안하는 알고리즘의 절차를 단계별로 기술하면 다음과 같다.

<내공생 진화알고리즘의 절차>

단계 1 : (초기화)

세 모집단 Pop-B, Pop-S, Pop-BS의 모든 셀에 개체를 임의로 생성한다. 그리고 초기 적응도를 평가한다. 이때 Pop-B와 Pop-S의 개체들은 같은 격자 위치에 있는 개체



[그림 8] Pop-BS에 있는 개체의 표현

를 공생자로 하여 평가한다. 평가에서 가장 좋은 해를 f_{best}로 둔다.

단계 2 : (이웃 설정)
임의의 위치(i, j)를 선택하여 이웃 NB_{ij}, NS_{ij}, NBS_{ij}를 정의한다.

단계 3 : (NB_{ij}의 진화)
함수 *genetic_evolution*(NB_{ij}, R_c, R_{im}, R_{gm}, f_{best})를 호출하여 수행한다.

단계 4 : (기존 내공생자와 잠재 내공생자의 경쟁)
NB_{ij}에서 가장 높은 적응도 조합을 잠재 내공생자 a_pb_q라 두자. $f(a_pb_q)$가 NBS_{ij}에 있는 가장 낮은 적응도인 $f(\alpha_v\beta_v)$보다 크면, NBS_{ij}의 $\alpha_v\beta_v$를 a_pb_q로 대체한다. 이때 NB_{ij}와 NS_{ij}내에 있는 각 개체 a_p와 b_q는 제 위치에 그대로 둔다.

단계 5 : (NS_{ij}의 진화)
함수 *genetic_evolution*(NS_{ij}, R_c, R_{im}, R_{gm}, f_{best})를 호출하여 수행한다.

단계 6 : (기존 내공생자와 잠재 내공생자의 경쟁)
NS_{ij}에서 가장 높은 적응도 조합을 잠재 내공생자 a_pb_q라 두자. $f(a_pb_q)$가 NBS_{ij}에 있는 가장 낮은 적응도인 $f(\alpha_v\beta_v)$보다 크면, NBS_{ij}의 $\alpha_v\beta_v$를 a_pb_q로 대체한다. 이때 NB_{ij}와 NS_{ij}내에 있는 개체 a_p와 b_q는 제 위치에 그대로 둔다.

단계 7 : (NBS_{ij}의 진화)
함수 *genetic_evolution*(NBS_{ij}, R_c, R_{im}, R_{gm}, f_{best})를 호출하여 수행한다.

단계 8 : (종료조건)
종료조건을 만족하면 알고리즘을 종료하고, 그렇지 않으면 단계 2로 간다.

위 절차에서, 단계 3, 5, 7의 함수 *genetic_evolution*(P, R_c, R_{im}, R_{gm}, f_{best})은 아래와 같다.

함수 *genetic_evolution* (P, R_c, R_{im}, R_{gm}, f_{best})

부 단계 1 : (선택과 교차) 집단 P에서 토너먼트선택에 의해 재생산에 참여할 개체를 선택한다. 그리고 선택된 각 개체에 대해 범위 [0, 1]에서 난수(random number)를 발생하여, 이 난수가 R_c(교차율)보다 작거나 같으면, 그 개체는 교차연산의 후보자가 된다. 이들 후보개체들에 대해 임의로 쌍을 지어 교차연산을 수행하여 자손개체(s개)를 생산한다.

부 단계 2 : (개체 대체) 집단 P에서 대체될 개체들을 선택한다. 이때 적응도가 가장 낮은 s개의 개체를 확정적으로 선택한다. 새롭게 생산된 자손개체를 선택된 개체들의 위치에 대체한다.

부 단계 3 : (돌연변이) P로부터 R_{im}(개체단위 돌연변이율)보다 낮은 개체들을 선택하고, R_{gm}(인자단위 돌연변이율)에 기초하여 각 인자에 대해 돌연변이를 행한다.

부 단계 4 : (적응도 평가와 최선해 갱신) 새롭게 생산된 개체들의 적응도를 평가한다. 그리고 가장 좋은 개체의 적응도가 f_{best}보다 높으면 f_{best}를 갱신한다.

단계 1은 초기해를 생성하고 초기 적응도를 평가하는 단계이다. 적응도는 가외작업에 의해 평가되고, 가외작업이 작을수록 높은 적응도를 갖는다. 작업할당과 투입순서가 주어져야 가외작업을 측정할 수 있다. Pop-B의 개체는 작업할당을 Pop-S의 개체는 투입순서를 나타낸다. 초기 적응도 평가에서 공생자는 상대 모집단의 같은 위치에 있는 개체로 둔다. 단계 2는 이웃 진화를 위해 특정한 하나의 위치 (i, j)의 이웃을 설정하는 단계이다.

단계 3은 밸런싱 모집단 Pop-B에 있는 일부 개체 집단 NB_{ij}가 진화를 하는 단계이다. 함수 *genetic_evolution*에서 부 단계 1은 이웃 NB_{ij}에서 크기 2의 토너먼트 선택에 의해 $|NB_{ij}|$개의 개체를 선택하고 교차율을 적용하여 교차를 실시하여 자손개체를 생산한다. 부 단계 2에서는 생산된 개체를 NB_{ij}에서 적응도가 낮은 개체와 대체하고, 부 단계 3에서는 NB_{ij}를 대상으로 하여 개체를 돌연변이율에 기초하

여 돌연변이한다. 부 단계 4에서는 교차 또는 돌연변이에 의해 새로이 생산된 개체의 적응도를 평가하고, 현재까지 찾은 최선해를 갱신한다. 적응도를 평가할 때 공생자가 요구된다. 이 공생자는 상대 모집단 Pop-S의 이웃 집단 NS_{ij}에서 가장 높은 적응도를 갖는 개체로 한다. 이 연구에서 이웃 진화는 교차율을 비교적 낮게 하여(예로 0.5), 안정상태 유전알고리즘(steady-state genetic algorithm)의 개념[1]을 도입하였다.

단계 4는 단계 3의 NB_{ij}의 진화 후, NB_{ij}에서 가장 높은 적응도를 갖는 개체가 이를 평가할 때 결합된 잠재내공자 $a_p b_q$와 NBS_{ij}에 있는 개체와 경쟁하는 단계이다. 이 잠재내공자가 Pop-BS 개체로의 변환을, 즉 결합된 형태로 진화하는 것을 허용한다.

단계 5는 단계 3과 같은 방법으로 Pop-S에 있는 일부 개체 집단 NS_{ij}를 진화하는 단계이고, 단계 6은 단계 4와 같은 방법으로 단계 5의 NS_{ij}의 진화 후, 잠재내공자 $a_p b_q$와 NBS_{ij}에 있는 개체와 경쟁하는 단계이다. 단계 7에서는 단계 3과 같은 방법으로 Pop-BS에 있는 일부 개체 집단 NBS_{ij}를 진화시킨다. 이 모집단의 개체는 결합된 형태이므로 공생자는 필요하지 않다.

4. 실험과 분석

4.1 실험문제와 비교 알고리즘

제안한 MMtAL를 위한 내공생 진화알고리즘(EEA)의 성능은 컴퓨터 모의 실험에 의해 평가한다. 실험문제는 양면조립라인의 기존 실험문제로, 24개 작업[14], 65개와 205개 작업[17] 문제를 혼합모델로 확장하여 사용하였다. 24개 작업의 경우에는 기존 작업시간(모델 B)에 각각 범위 [0.5, 1.2]과 [0.9, 1.5] 의 난수를 곱하여 두 모델, 모델 A와 C를 생성하였다. 65개 작업 문제는 기존 작업시간을 갖는 모델 C와 새로운 작업시간을 갖는 4개의 모델을 생성하였다. 새로운 작업시간은 기존 작업시간에 범위 [0.5, 1.5]의 난수와 모델별 가중치를 곱하여 생성하였다. 이 때 모델별 가중치는 {A, B, D, E} = {0.8, 0.9, 1.2, 1.4}로 두었다. 그리고 205개의 문제는 5개의 모델을 각각 기존의 작업시간을 동일하게 두고, 모델별로 특정 작업을 0으로 두고 혼합모델로 확장시켰다. <표 1>는 작업장쌍 수(Number of mated-stations)와 MPS를 다양하게 변화시킨 문제를 보여주고 있다. 실험문제에서 컨베이어의 이동속도(v_c)는 계산의 편의를 위해 1로 두었으며, 제품 투입간격(λ)은 1회의 MPS를 생산하는데 요구되는 총 생산시간($\sum_{m=1}^{M} d_m t_{im}$)을 제품의 총 수와 작업장 수의 곱으로 나눈 값, 즉 $\gamma = \sum_{i=1}^{I} \sum_{m=1}^{M} d_m t_{im} / (S \times J \times 2)$로 두었다. 각 작업장의 길이는 투입시간간격의 1.5배로 두었다.

〈표 1〉 실험문제

Problem	Number of tasks	Number of models	Number of mated-stations	MPS
P24_1	24	3	3	3 2 1
P24_2	24	3	3	5 3 2
P24_3	24	3	3	2 3 5
P65_1	65	4	6	1 2 9 10
P65_2	65	4	6	8 7 5 3
P65_3	65	4	6	5 6 5 6
P65_4	65	5	6	1 2 3 9 10
P65_5	65	5	6	8 7 5 3 2
P65_6	65	5	6	5 5 6 4 6
P65_7	65	5	8	1 2 3 9 10
P65_8	65	5	8	8 7 5 3 2
P65_9	65	5	8	5 5 6 4 6
P205_1	205	4	8	1 2 9 10
P205_2	205	4	8	8 7 5 3
P205_3	205	4	8	5 6 5 6
P205_4	205	5	8	1 2 3 9 10
P205_5	205	5	8	8 7 5 3 2
P205_6	205	5	8	5 6 5 6 4
P205_7	205	5	10	1 2 3 9 10
P205_8	205	5	10	8 7 5 3 2
P205_9	205	5	10	5 6 5 6 4

제안하는 내공생 진화알고리즘(EEA)의 특징과 탐색성능의 비교 분석을 위하여, 비교 알고리즘으로 SNA(separated and neighborhood symbiotic evolutionary algorithm), SPA(separated and population-based symbiotic evolutionary algorithm),

GA(genetic algorithm)의 세 알고리즘을 사용한다. SNA는 밸런싱과 투입순서를 위한 두 모집단으로 분리되어 공진화한다. 이때 이웃 단위로 진화하고 공생자는 이웃에서 가장 좋은 개체로 둔다. 이는 EEA에서 결합된 형태의 내공생 진화가 없는 알고리즘으로, 즉, EEA에서 단계 4, 단계 6, 단계 7를 수행하지 않는 알고리즘이다. SPA는 SNA와 같이 두 모집단으로 분리되어 공진화하나, 이웃 진화하는 SNA와 달리 모집단을 기준으로 진화한다. 이때 공생자는 상대 모집단의 가장 좋은 개체로 둔다. GA는 밸런싱과 투입순서를 위한 두 염색체를 결합한 완전해 형태로 개체를 표현하고, 모집단 단위로 진화하다. 이 절차는 일반적 유전알고리즘의 절차[1]를 따른다.

4.2 파라미터 설정

이 연구에서 사용되는 파라미터는 예비실험을 통하여 결정하였다. 첫째, 모집단은 문제의 종류에 따라 다르게 주었는데 이는 대형문제에서는 모집단의 크기가 작은 경우 해 공간을 탐색능력이 떨어져 모집단의 다양성을 유지하기 어렵기 때문이다. 각 문제의 모집단 크기는 P24문제 49(7×7의 격자구조), P65문제 81(9×9의 격자구조), P205문제 100(10×10의 격자구조)으로 하였다. EEA와 SNA에서 이웃의 크기는 모두 (3×3)의 정방형으로 두었다. 둘째 토너먼트선택을 기반으로, 교차율은 EEA와 SNA는 밸런싱과 투입순서 문제 모두 0.5로 하였고, 밸런싱 개체의 개체 돌연변이율(R_{im})은 0.2, 인자 돌연변이율(R_{gm})은 0.3으로 두고, 투입순서 개체의 개체 돌연변이율은 0.1로 두었다. 그리고 SPA의 경우에는 0.2의 교차율과 밸런싱과 투입순서의 개체돌연변이는 0.05로 두고, 인자 돌연변이율은 0.3으로 두었다. 이들 세 알고리즘에서 비교적 낮은 교차율을 사용한 것은 모집단에서 적응도에 따라 적은 수의 개체를 선택하여 재생산하고 이를 모집단에서 적응도가 낮은 개체와 대체하는 안정상태(steady state) 유전알고리즘의 개념을 채용하였기 때문이다. 여기서 사용된 돌연변이율은 실험을 통해 적절한 값을 채택하였다.

그리고 GA에서는 개체가 완전해로 형성되므로 교차율은 0.95, 밸런싱과 투입순서 문제에서 개체돌연변이율은 0.1, 밸런싱 문제의 인자돌연변이율은 0.3으로 두었다. 이들 파라미터 또한 실험을 통해 적절한 값을 채택하였다. 앞에서 언급했듯이 EEA와 SNA에서 공생자는 상대 모집단의 이웃에서 가장 높은 적응도를 갖는 개체를, SPA에서는 상대 모집단에서 가장 좋은 적응도를 갖는 개체를 사용하였다. 마지막으로 종료조건은 24개의 작업의 경우는 50,000번, 65개와 205개의 문제는 200,000번의 재생산개체수로 비교알고리즘 모두 동일한 종료조건을 적용하였다. 여기서 재생산개체수는 형평성을 위해 GA와 EEA의 Pop-BS 진화에서는 재생산개체수를 그대로 사용하나, 분리된 두 모집단을 갖는 SNA와 SPA, 그리고 EEA의 Pop-B와 Pop-S의 진화에서는 각 모집단에서 생산된 개체 수의 반을 재생산된 개체수로 두었다.

4.3 실험결과와 분석

제안한 EEA의 성능 분석을 위해 SNA, SPA, GA의 세 진화알고리즘과 비교하였다. 모든 알고리즘은 java로 구현하였으며, 2.4GHZ Q6600 CPU를 장착한 IBM-PC에서 수행하였다. 각 실험문제를 20회 반복 수행하였다. <표 2>는 20회 수행에서 가장 좋은 해(Best), 각 반복에서 가장 좋은 해의 평균 값(Mean), 개선율(Improvement rate)를 나타낸다. 개선율의 R1은 {(mean of GA-mean of SPA)/mean of GA}×100%, R2는 {(mean of SPA-mean of SNA)/mean of SPA}×100%, R3은 {(mean of SNA-mean of EEA)/mean of SNA}×100%을 나타낸다.

<표 2>의 실험결과를 분석하면 다음과 같다.

첫째, 모든 실험문제에서 EEA가 가장 좋은 성능을 보였다. 다음은 SNA이다. EEA와 SNA의 차이는 단지 내공생자의 존재에 있다. EEA에서는 부분해에 의한 병렬탐색과 완전해(내공생자)로 이루어

〈표 2〉 EEA와 비교 알고리즘의 성능 비교

problem	Best				Mean				Improvement rate		
	GA	SPA	SNA	EEA	GA	SPA	SNA	EEA	R1	R2	R3
P24_1	14.7	11.5	11.5	11.5	19.6	19.4	16.2	15.9	1.0	15.4	1.9
P24_2	26.3	24.5	16.5	16.5	30.5	29.0	25.2	25.0	4.9	13.1	0.8
P24_3	19.0	18.3	18.8	18.6	23.0	22.3	21.0	20.1	3.0	5.8	4.7
P65_1	1657	1042	867	841	2044	1477	1185	1159	27.7	19.7	2.2
P65_2	1618	1048	982	885	2042	1354	1248	1154	33.6	7.8	7.5
P65_3	1575	903	855	798	1769	1360	1151	1102	23.1	15.3	4.3
P65_4	2640	985	1000	985	3563	1865	1510	1469	47.6	19.0	2.7
P65_5	2836	1601	1466	1427	3324	1986	1908	1885	40.2	3.9	1.2
P65_6	3415	1551	1514	1504	4017	2020	1936	1885	49.7	4.1	2.6
P65_7	4093	2084	1981	1962	4880	2797	2391	2320	42.6	14.5	3.0
P65_8	3773	2129	2060	2051	4235	2701	2399	2329	36.2	11.1	2.9
P65_9	4148	1949	1890	1879	4900	2719	2366	2326	44.5	12.9	1.7
P205_1	13416	6295	5814	5299	16529	9239	7374	6930	44.1	20.1	6.0
P205_2	13866	6640	5883	5867	16893	9994	6916	6762	40.8	30.7	2.2
P205_3	13730	5408	5329	5118	16673	8937	6588	6425	46.3	26.2	2.5
P205_4	16110	8682	6240	6171	18667	11327	8309	8177	39.2	26.6	1.6
P205_5	13771	9045	7153	6815	18449	11236	8895	8345	39.0	20.8	6.2
P205_6	17714	7637	6789	6520	19485	11487	8238	8156	41.0	28.2	1.0
P205_7	11290	5691	5275	4854	16436	8510	7328	7222	48.2	13.8	1.5
P205_8	13701	6986	6126	5566	17868	9487	7879	7599	46.9	16.9	3.6
P205_9	12870	7415	5938	5217	18294	10065	8220	7539	44.9	18.3	8.3

진 통합탐색을 동시에 수행한다. EEA의 이러한 성능은 부분해를 사용하는 공생진화 알고리즘이 갖는 병렬탐색능력과 완전해를 표현한 개체를 사용하는 표준진화 알고리즘이 갖는 빠른 수렴성을 적절히 조화시킨 결과에서 비롯된 것으로 보인다.

둘째, SNA와 SPA를 비교하여 보자. 이들 두 알고리즘의 비교는 공생진화 알고리즘에서 이웃 진화의 효과를 시험하는 것이다. <표 2>에서 개선율 R2로부터 이웃 진화가 해의 탐색효율을 촉진함을 알 수 있다. 이는 이웃 진화가 모집단의 다양성을 유지하여 조기 수렴을 방지하고, 적소(niche)를 형성하는 경향이 있는 데 기인한다[1].

셋째, SPA는 GA보다 좋은 성능을 보인다. 이는 복잡도가 큰 문제에서 부분해를 운영하는 공생진화 알고리즘이 완전해를 운영하는 표준진화 알고리즘보다 탐색효율이 더 우수함을 의미한다. 넷째, 개선율 R1, R2, R3로부터 EEA의 성능이 대형 문제에서 더 좋아 지는 경향이 있음을 알 수 있다. 이는 앞에서 설명한 EEA가 병렬탐색과 수렴성의 적절한 조화를 이용한 데 기인한 것으로 보인다. 그리고 해 개선에 공생진화 알고리즘의 병렬 탐색이, 다음은 이웃 진화, 내공생자의 존재 순으로 영향을 주고 있음을 알 수 있다.

<표 3>은 각 실험문제에 대한 매 실험에서 얻은 가장 좋은 해의 표준편차(s.d.)를 보여 주고 있다. EEA는 다른 비교 알고리즘에 표준 편차가 낮아 알고리즘이 안정적임을 알 수 있다. 다음은 SNA, SPA 순으로 안정성이 떨어졌고, GA가 가장 낮았다. 특히 대형문제에서 GA는 해의 변동폭이 컸다.

알고리즘에 대한 해의 수렴성을 비교해 보자. [그림 9]는 P65_8과 P205_5문제에서 재생산개체수에 따른 가장 좋은 해의 탐색을 보여준다. 이 그림에서 EEA의 수렴속도는 다른 알고리즘에 비해 빠르다. EEA는 진화의 초기 단계를 지나면 다른 비교알고리즘보다 더 좋은 해를 제공함을 알 수 있다. 다른 대형 실험문제에서도 비슷한 결과를 보였다. 끝으

〈표 3〉 EEA와 비교 알고리즘의 해의 표준편차

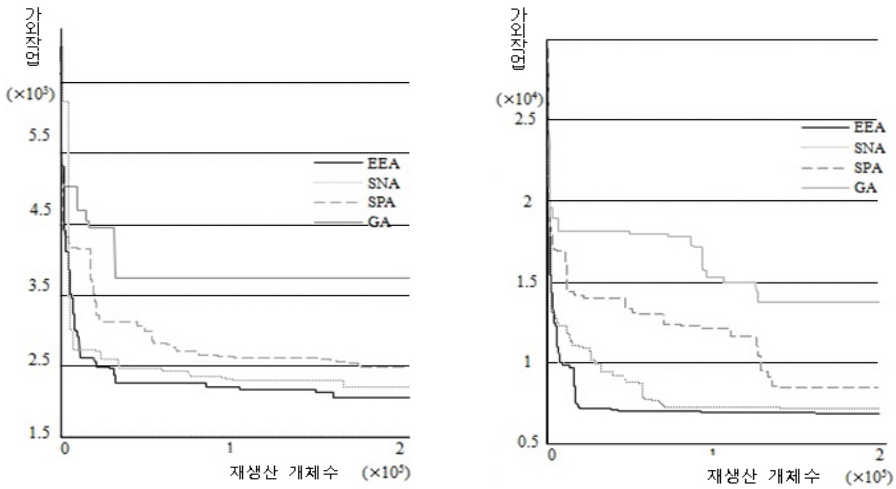
problem	s.d.				problem	s.d.			
	GA	SPA	SNA	EEA		GA	SPA	SNA	EEA
P24_1	2.1	3.0	1.8	1.8	P205_1	1982	1920	775	662
P24_2	3.1	3.1	3.8	3.6	P205_2	2336	1615	757	594
P24_3	2.1	2.3	1.5	1.4	P205_3	1356	1425	745	566
P65_1	183	341	194	173	P205_4	1590	1287	1214	1089
P65_2	249	150	168	156	P205_5	2475	1158	904	882
P65_3	126	301	154	140	P205_6	1614	1484	859	815
P65_4	362	424	257	241	P205_7	2997	1428	1108	1073
P65_5	243	228	193	181	P205_8	2934	1615	1168	1051
P65_6	226	274	185	168	P205_9	3485	1990	1098	1037
P65_7	339	312	241	197					
P65_8	164	255	196	177					
P65_9	328	413	191	174					

로, 컴퓨터 계산소요시간은 GA가 가장 적고, 다음은 EEA, SPA, SNA순이었다. GA, EEA, SPA, SNA는 65개 작업 문제에서 각각 평균 36초, 78초, 124초, 126초의 계산시간이 요구되었고, 205개 작업 문제에서는 각각 평균 294초, 376초, 450초, 522초이 소요되었다. EEA는 일부 결합된 내공생자의 진화로 인해 완전해를 운영하는 GA보다는 계산시간이 많이 소요되나, 부분해로 나뉘어 진화하는 공생진화알고리즘인 SNA와 SPA보다는 더 적게 소요되었다.

5. 결 론

이 연구에서는 혼합모델 양면조립라인(MMtAL)의 밸런싱과 투입순서를 동시에 해결할 수 있는 효율적인 진화알고리즘으로, 내공생 진화알고리즘(EEA)을 제안하였다. 제안한 EEA는 기존의 공생진화알고리즘에 진핵생물의 진화과정을 추가한 형태로 자연의 내공생 진화과정을 모방한 알고리즘이다. EEA에서는 밸런싱과 투입순서를 나타낸 부분해로 구성된 두 모집단과 밸런싱과 투입순서를 결합하여 완전해로 구성된 모집단을 운영한다. 부분해로 구성된 두 모집단의 각 개체는 분리되어 공생 진화하고, 이 진화과정에서 나타난 좋은 조합은 결합된 형태로, 즉 내공생 진화한다. 알고리즘 성능 향상을 위하여 이웃 진화전략과 안정상태 유전알고리즘의 개념을 채용하였다. 또한, MMtAL의 밸런싱과 투입순서에 적합한 유전표현과 유전연산자를 제안하였다. 다양한 실험문제를 사용한 실험에서, 제안한 EEA는 전통적 유전알고리즘은 물론 비교된 두 공생진화알고리즘보다 해의 품질과 수렴속도에서 우수한 결과를 보였다.

제안한 EEA가 우수한 성능을 갖는 이유는 다음과 같이 설명할 수 있다. 첫째, EEA는 부분해에 의



(a) P65_8 문제 (b) P205_5 문제

[그림 9] 재생산 개체 수에 따른 알고리즘 성능 비교

한 병렬탐색과 완전해에 의한 통합탐색을 적절히 조화시킨다. 병렬탐색은 해의 수렴속도를 낮추나 모집단의 다양성에 기여하고 통합탐색은 지역최적에 빠질 수 있으나 수렴속도를 증가시키는 데 기여한다. 둘째, 이웃 공진화전략의 사용과 안정상태 유전알고리즘의 개념의 도입이다. 전자는 적소 형성에 의해 모집단의 다양성을 촉진하여 조기수렴을 방지하는 데, 후자는 좋은 해의 이용(exploitation)을 높이는 데 기여한다. 끝으로, 다루는 문제에 적절한 유전요소, 즉 표현, 교차, 돌연변이, 파라미터의 사용이다.

진화알고리즘은 다양한 형태의 목적과 제약을 다룰 수 있다는 강점을 가지고 있다. 이로부터 제안한 알고리즘의 개념과 구조는 여러 부분문제가 통합된 복잡도가 높은 다양한 분야의 문제 해결에 적용될 수 있을 것으로 기대된다.

참 고 문 헌

[1] 김여근, 진화알고리즘, 전남대학교 출판부, 2011.

[2] 김여근, 이상선, “다목적을 갖는 혼합모델 조립라인의 밸런싱과 투입순서를 위한 공생진화알고리즘”, 「한국경영과학회지」, 제35권, 제3호(2010), pp.25-43.

[3] 송원섭, 김여근, 진화알고리즘을 이용한 선취적 다목적 양면조립라인 밸런싱, 한국경영과학회지, 제34권, 제2호(2009), pp.101-111.

[4] Bard, J.F., E.M. Dar-El, and A. Shtub, “An Analytic Framework for Sequencing Mixed Model Assembly Lines,” *International Journal of Production Research,* Vol.30(1992), pp.35-48.

[5] Bartholdi, J.J., “Balancing two-sided assembly lines : a case study,” *International Journal of Production Research*, Vol.31, No.10 (1993), pp.2447-2461.

[6] Baykasoğlu, A. and T. Dereli, “Two-sided assembly line balancing using an ant-colony-based heuristic,” *International Journal of Advanced Manufacturing Technology*, Vol. 36(2008), pp.582-588.

[7] Davis, L., “Applying adaptive algorithms to epistatic domains,” *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, (1985), pp.162-164.

[8] Falkenauer, E., “A new representation and operators for genetic algorithms applied to grouping problems,” *Evolutionary Computation*, Vol.2(1994), pp.123-144.

[9] 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(2008), pp. 435-440.

[10] Kim, J.Y., Y. Kim, and, Y.K. Kim, “An esdosymbiotic evolutionary algorithm for optimization,” *Applied Intelligence,* Vol.15, No.2 (2001), pp.117-130.

[11] Kim, Y.K., S.J. Kim, and J.Y. Kim, “An esdosymbiotic evolutionary algorithm for the integration of balancing and sequencing in mixed-model U-lines,” *European Journal of Operational Research*, Vol.168(2006), pp.838-852.

[12] Kim, Y.K., C.J. Hyun, and Y. Kim, “Sequencing in mixed model assembly lines : a genetic algorithm approach,” *Computers and Operations Research*, Vol.23(1996), pp.1131-1145.

[13] Kim, Y.K., J.Y. Kim, and Y. Kim, “A coevolutionary algorithm for balancing and sequencing in mixed model assembly lines,” *Applied Intelligence*, Vol.13(2000a), pp.247-258.

[14] 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(2000b), pp.44-53.

[15] 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(2009), pp.853-865.

[16] Lapierre, S.D., A. Ruiz, and P. Soriano, "Balancing assembly lines with tabu search," *European Journal of Operational Research*, Vol.168(2006), pp.826-37.

[17] Lee, T.O., Y.H. 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.

[18] Margulis, L., *Origin of eukaryotic cells*, Yale University Press : New Haven, 1970.

[19] Margulis, L., *Symbiosis in cell evolution*, W. H. Freeman : San Francisco, 1981.

[20] Özcan, U. and B. Toklu, "A tabu search algorithm for two-sided assembly line balancing," *International Journal of Advanced Manufacturing Technology*, Vol.43, No.7/8(2009a), pp.822-829.

[21] 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(2009b), pp.1955-1965.

[22] Özcan, U. and B. Toklu, "Balancing of mixed-model two-sided assembly lines," *Computers and Industrial Engineering*, Vol.57, No.1 (2009c), pp.217-227.

[23] Potter, M.A., *The design and analysis of a computational model of cooperative coevolution*, Ph. D. dossertation, George Mason Univercity, 1997.

[24] Simaria, A.S. and P.M. Vilarinho, "2-ANT BAL : An ant colony optimization algorithm for balancing two-sided assembly lines," *Computers and Industrial Engineering*, Vol. 56, No.2(2009), pp.489-506.

[25] Thomopulos, N.T., Line Balancing-Sequencing for Mixed-Model Assembly, *Management Science*, Vol.14(1967), pp.59-75.

[26] von Laszewski, G., Intelligent structural operators for k-way group partitioning problem, *Proceedigns 4th International on Conference Genetic Algorithms*, (1991), pp.45-52.