

내공생 진화알고리즘을 이용한 혼합모델 조립라인의 작업할당과 투입순서 결정*

김여근** · 손성호***

Balancing and Sequencing in Mixed Model Assembly Lines Using an Endosymbiotic Evolutionary Algorithm*

Yeo Keun Kim** · Seong Ho Son***

■ Abstract ■

This paper presents a new method that can efficiently solve the integrated problem of line balancing and model sequencing in mixed model assembly lines (MMALs). Line balancing and model sequencing are important for an efficient use of MMALs. The two problems of balancing and sequencing MMALs are tightly related with each other. However, in almost all the existing researches on mixed-model production lines, the two problems have been considered separately. In this research, an endosymbiotic evolutionary algorithm, which is a kind of coevolutionary algorithm, is adopted as a methodology in order to solve the two problems simultaneously. This paper shows how to apply an endosymbiotic evolutionary algorithm to solving the integrated problem. Some evolutionary schemes are used in the algorithm to promote population diversity and search efficiency. The proposed algorithm is compared with the existing evolutionary algorithms in terms of solution quality and convergence speed. The experimental results confirm the effectiveness of our approach.

Keyword : Mixed model assembly line, Line balancing, Model sequencing, Endosymbiotic evolutionary algorithm

논문접수일 : 2001년 9월 20일 논문게재확정일 : 2001년 11월 28일

* 본 연구는 2000년도 전남대학교 학술연구비 지원으로 수행되었음.

** 전남대학교 산업공학과 교수

*** 전남대학교 산업공학과

1. 서 론

혼합모델 조립라인(Mixed Model Assembly Line : MMAL)은 공장부지, 시설 등의 투자를 줄이고 시설의 효율적인 이용을 위하여 한 라인에서 유사한 여러 모델의 제품을 생산하는 조립라인이다. MMAL의 효율적 이용을 위하여 작업할당문제와 모델투입순서문제는 중요하게 다루어져야 한다[29]. 작업할당문제는 제품 생산에 요구되는 여러 작업들을 작업장에 할당하는 문제이고, 투입순서문제는 모델들의 생산순서를 결정하는 문제이다. 본 연구에서는 MMAL에서 작업할당과 투입순서를 동시에 결정하는 효율적인 알고리즘을 제시한다.

MMAL에서 작업할당과 모델투입순서는 밀접한 관련성을 갖는다. 즉, 모델의 최적투입순서는 각 모델의 작업할당에 의해 결정되고, 작업할당량은 투입순서에 의해 또한 영향을 받는다. 그러나 기존 대부분의 연구에서는 두 문제를 분리하여 독립적으로 다루었다. 이들 두 문제는 모두 조합 최적화 문제로서 NP-hard로 알려져 있다[11, 31]. MMAL에서 작업할당에 관한 연구는 Thomopoulos[30], Macaskill[19], Chakravarty and Shtub[8] 등에 의해 이루어졌다. 그러나 이들 연구에서 모델의 투입순서는 고려하지 않고 있다. 한편, MMAL에서 투입순서문제는 Kilbridge and Wester[12]에 의해 연구가 시작된 이후, Okamura and Yamashina[23], Yano and Rachamadugu[34], Bard *et al.*[5], Tsai[31], Kim *et al.*[14] 등에 의해 연구가 행하여졌다. 이들 연구는 모두 작업할당이 이루어진 상태에서 모델투입순서문제를 다루고 있다. 두 문제를 동시에 고려한 연구로는 Thomopoulos[29]와 Dar-El and Navidi[9]의 연구가 있다. 이들 연구[9, 29]에서는 단지 계층적 방법에 의해, 먼저 작업할당문제를 풀고, 이로부터 얻은 해를 입력하여 모델의 투입순서문제를 해결하였다. 이러한 계층적 방법은 해공간을 충분히 탐색하는데 한계를 갖는다. MMAL의 작업할당과 모델투입순서의 통합적 접근은 이 논문의 저자들에 의해 일차적인 연구가

행하여졌다[15]. 이 연구[15]에서는 두 문제를 동시에 해결하기 위한 방법론으로 공생 진화알고리즘(symbiotic evolutionary algorithm)을 사용하였다. 공생 진화알고리즘은 자연계의 여러 종들이 주어진 환경 내에서 서로 영향을 주고 받으며 진화하는 공진화 과정을 모방한 확률적 탐색 기법이다[17]. 공생 진화알고리즘의 여러 변형들이 Potter[24], Moriarty and Miikkulainen[22], 그리고 Kim *et al.*[15]에 의해 제안되었다. 제안된 알고리즘들에서 각 부분문제는 서로 다른 종으로 간주되고, 각 부분문제의 잠재해를 나타내는 개체들은 자신들의 모집단을 독립적으로 구성한다. 즉, 각 개체들은 상호 작용하는 다른 개체들과 서로 분리되어 공생하며 진화한다고 보고 있다. 공생 진화알고리즘은 계층적 방법보다 우수하다고 알려져 있다. 그러나 공생 진화알고리즘에서 개체들은 공생파트너(symbiotic partner)와 상관없이 독립적으로 진화하는 경향이 있다. 여기서 공생파트너란, 개체의 적응도 평가에 참여하는 상대모집단(또는 이웃)의 개체를 말한다. 이렇듯 공생하는 개체들이 분리되어 공생하면 진화하는 동안 공생파트너의 변경기회를 증가시켜 한 탐색방향에 대한 충분한 탐색을 방해할 수 있으며, 좋은 해로의 수렴 속도를 감소시킬 수 있다.

이러한 단점을 극복하기 위하여, 자연계의 내공생 진화과정을 모방한 새로운 공생 진화알고리즘이 개발되었다[13]. 본 논문에서는 이를 내공생 진화알고리즘(endosymbiotic evolutionary algorithm : EEA)이라 부르기로 한다. EEA에서는 서로 다른 종의 개체가 분리된 상태에서 공생할 뿐 아니라 결합된 상태에서의 공생도 허용한다. 즉, EEA에서는 어떤 개체들이 자신의 공생파트너의 변동 없이 일정기간 동안 적응할 수 있도록 하여, 해공간의 주어진 탐색방향에 대하여 충분한 탐색이 이루어지도록 하였다. 이러한 방법은 개체가 자신의 적응도를 높게 평가해준 공생파트너를 만나면 이와 안정적으로 적응하면서 진화해 나가도록 하여준다. 본 연구에서는 MMAL에서 작업할당과 모델투입순서

결정의 통합문제 해결을 위한 방법론으로 EEA를 사용하였다. 본 연구를 통해, MMAL에서 작업할당과 모델 투입순서를 동시에 결정할 수 있는 새로운 방법론을 제안하고, 기존 연구결과와의 비교를 통해 제안한 EEA의 우수성을 보이고자 한다.

2. MMAL의 작업할당과 투입순서 결정

본 연구에서 다루는 MMAL은 다음과 같은 상황에서 운영된다고 본다. 첫째, 컨베이어는 일정속도(v_c)로 이동하며, 여러 모델의 제품들은 일정 시간간격(γ)으로 투입된다. 둘째, 라인은 폐쇄작업장으로 구성되어 있다. 그리고 각 작업장은 한 사람의 작업자로 운영되며, 각 작업장에 기준점을 두어 이 위치 앞으로 작업자가 이동하여 작업을 수행할 수 없다고 본다. 셋째, 작업장내에서 완료하지 못한 작업들은 가외작업자에게 할당된다. 넷째, 생산하는 모델들의 결합선행공정도가 주어졌다고 본다. 결합선행공정도는 작업선행관계, 작업구성, 작업시간, 그리고 생산량이 서로 다른 모델들의 선행공정도를 하나로 통합한 것이다[19]. 다섯째, 라인은 사이클 생산 방식(cyclic production)을 사용한다. 전체 계획생산기간 동안 M 종류의 모델이 생산되며, 이 기간동안 모델 m 의 수요를 $D_m(m=1, 2, \dots, M)$ 이라 하자. 그리고 D_1, D_2, \dots, D_M 의 최대공약수가 g 이면, $\mathbf{d}=(D_1/g, D_2/g, \dots, D_M/g)=(d_1, \dots, d_M)$ 로 둔다. 이 \mathbf{d} 를 최소부품집합(minimum part set : MPS)[5]이라 부르고, 이를 반복 생산한다. 즉, MPS를 g 번 반복 생산하여 총 수요를 만족시킨다. 본 연구에서 다루는 모든 투입순서는 MPS를 기준으로 하며, 이는 혼합모델 조립라인[5, 14]과 유연생산시스템[3, 27]에서 흔히 사용하는 방식이다. 그리고 작업자의 이동시간은 무시한다.

MMAL에서 작업할당과 모델투입순서 결정을 위하여 여러 목적들이 사용될 수 있다. 본 연구에서는 가외작업(utility work) 최소화를 목적으로 두

문제를 동시에 해결한다. 가외작업은 각 제품에 대하여 고정된 길이의 작업장 내에서 완료되지 못한 작업의 양을 의미하며, 이를 최소화함으로써 인건비, 라인 길이, 그리고 컨베이어 정지위험 등을 줄일 수 있다[34]. MPS 생산기간동안 작업할당과 투입순서가 주어지면, 다루는 MMAL에서 가외작업은 식 (1)에 의해 구할 수 있다.

$$U = \sum_{j=1}^J [\sum_{h=1}^H \max\{0, (Z_{hj} + v_c T_{j m_h} - L_j) / v_c\} + Z_{(H+1)j} / v_c], \quad (1)$$

여기서, J 는 작업장수, H 는 MPS 생산기간동안 생산되는 제품의 수 ($= \sum_{i=1}^M d_i$)이며, Z_{hj} 는 작업장 j 에서 h 번째 투입모델의 작업시작 위치이다.

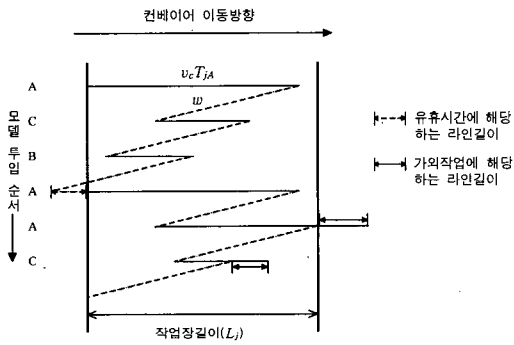
$T_{j m_h}$ 는 작업장 j 에서 h 번째 투입되는 모델 m_h 의 작업시간, L_j 는 작업장 j 의 길이, v_c 는 컨베이어 속도를 나타낸다. 식 (1)의 첫 항에서는 작업시작 위치로부터 조립에 필요한 라인길이를 더한 값이 작업장 길이를 초과하는 길이가 계산된다. 둘째 항은 모든 작업장에서 각 생산 사이클(MPS 생산)의 첫 번째 투입모델의 작업시작 위치는 기준점에서 시작한다는 가정. 즉, $Z_{1j} = 0, \forall j$ 에 의해 계산된 것이다. 즉, 이 항은 각 사이클의 마지막 모델에서 다음 모델을 기준점에서 시작하기 위하여 요구되는 가외작업을 나타낸다. 작업장 j 에서 투입순서에 따른 각 제품의 작업시작 위치는 아래와 같이 계산된다.

$$Z_{(h+1)j} = \max[0, \min(Z_{hj} + v_c T_{j m_h} - w, L_j - w)], \\ h = 1, 2, \dots, H$$

여기서 $w(= \gamma v_c)$ 는 투입시간 간격동안 컨베이어의 이동거리이다.

본 연구에서 다루는 MMAL의 폐쇄작업장을 도식화하면 [그림 1]과 같다. 여기서 작업장에서 수행되어야 할 각 모델의 작업들은 주어져 있고, 모델의 투입순서는 (A C B A A C)라 하자. 작업자는 한 모델에 대한 작업이 끝나면 컨베이어의 이동방향과 반대방향으로 이동한다. 점선은 모델의 투입시간 간격동안 컨베이어의 이동거리(w)를 나

타내며, 모델의 투입시간 간격이 일정하기 때문에 모든 점선은 평행하다. 만약 바로 다음에 작업해야 할 제품이 작업장 내에 있으면 작업자는 작업을 시작할 수 있지만, 그렇지 않으면 작업자는 제품이 작업장의 기준점에 도달할 때까지 기다려야 한다. 위의 예에서는 다섯 번째와 여섯 번째 모델에서 가외작업이 발생한다. 여섯 번째 모델에서 발생하는 가외작업은 다음 MPS에서 첫 번째 투입되는 모델을 기준점에서 시작한다는 가정에서 비롯된 것이다. 그리고 네 번째 모델에서는 작업자의 유휴시간이 발생한다.



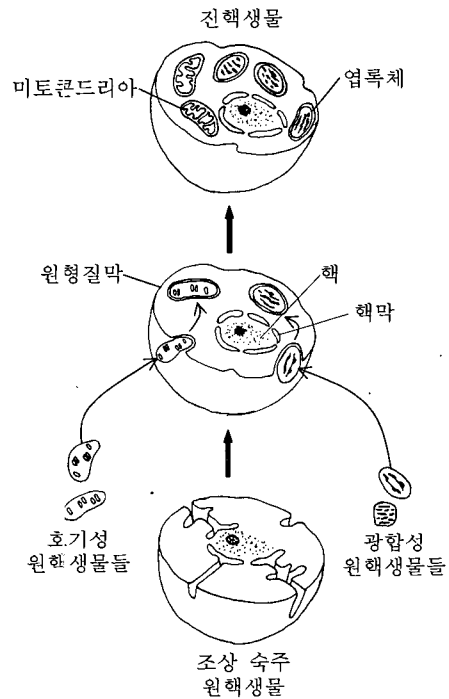
[그림 1] 폐쇄작업장에서의 작업

3. MMAL의 작업할당과 투입순서 결정을 위한 내공생 진화알고리즘

3.1 내공생진화론

내공생진화(endosymbiotic evolution) 이론은 원핵생물(prokaryotes)들이 내공생 과정을 거쳐 진핵생물(eukaryotes)로 진화하였다는 이론으로, 1981년 생물학자인 Lynn Margulis[21]에 의해 제안되었다. 세포소기관인 미토콘드리아(mitochondria)와 엽록체(chloroplast)의 형성과정으로 이를 설명하고 있다. 미토콘드리아의 조상이 되는 원핵생물은 산소를 이용하여 세포호흡을 하는 호기성 세포로, 이들이 더 큰 독립 영양체인 원핵생물(숙주) 내부에 들어가 소화되지 않고 살아남게 되어 진화하였다는 것이다. 이와 유사하게, 엽록체는 몸집이 큰 숙

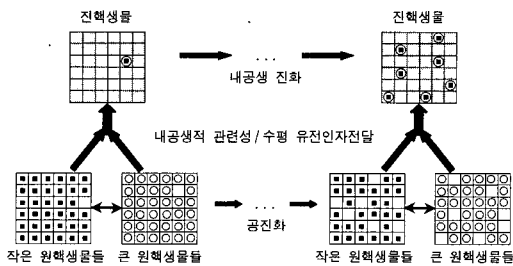
주 세포내에 살게 된 작은 광합성계 박테리아(남조)가 진화하였다는 것이다. 이러한 증거는 미토콘드리아와 엽록체는 각기 자신의 DNA를 가지고 있어 모세포와 독립적으로 증식할 수 있다는 데서 찾았다. [그림 2]는 이러한 내공생진화의 과정을 보이고 있다[7].



[그림 2] 내공생 진화

내공생진화는 원핵생물들이 내공생(endosymbiosis)과정과 수평적 유전인자전달(horizontal gene transfer)의 과정을 통해 진핵생물(eukaryote)로 진화하였다고 보는 이론이다[21]. 수평적 유전인자전달은 어떤 형질이 부모로부터 자식에게 수직적으로 유전되는 것이 아니고, 동종 또는 타종의 개체로 전해져서 진화하는 것을 말한다. 이런 경우, 개체의 생존율이나 번식률을 높이는 형질이 진화한다고 하는 단순 적응진화는 성립하지 않지만, 가능한 다른 개체에 적응하기 쉬운 형질이 진화한다고 하는 의미로서 적응진화의 한 형태라고 알려져 있다. 현재, 많은 생물학자들은 내공생 진화론을 정설

로 받아들이고 있으며, 이 이론으로 지구상의 생명체 기원과 진화를 설명하고 있다[7, 18]. 이러한 내공생진화의 개념을 간략히 추상화하면 [그림 3]과 같다. 이 그림은 두개의 원핵세포가 결합하여 하나의 진핵세포가 되는 과정을 나타낸 것이다.



[그림 3] 인공 내공생진화

3.2 내공생 진화알고리즘

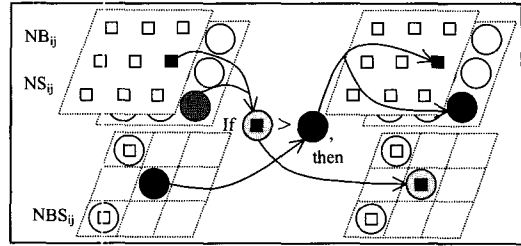
본 연구에서는 MMAL의 작업할당문제와 모델의 투입순서문제를 동시에 해결하기 위하여 내공생 진화알고리즘(EEA)을 방법론으로 사용한다. EEA는 3.1절에서 설명된 내공생 진화론을 모방한 하나의 공생 진화알고리즘이다. EEA의 기본 아이디어는 원핵생물들이 내공생을 통해 진핵생물로 진화하는 과정을 모방한 진화전략을 기존의 공생 진화알고리즘에 포함시킨 것으로서, 기존의 공생 진화알고리즘의 병렬탐색능력과 표준 유전알고리즘의 빠른 수렴성을 적절히 조화시킨 알고리즘이다. 이러한 조화를 위하여 EEA는 부분해에 의한 병렬탐색과 전체해에 의한 통합적 탐색을 동시에 수행한다. 또, EEA에서는 모집단내와 모집단간의 이웃진화전략을 사용하였다. 이러한 전략의 사용은 모집단의 다양성이 유지되도록 한다[15, 16]. 일반적으로 진화알고리즘의 탐색 성능은 모집단의 다양성에 영향을 받는다고 알려져 있다[22, 26]. 다양성이 높은 모집단의 유전 탐색(genetic search)은 재조합을 통하여 새로운 인자구조를 생산할 가능성이 높고, 부분최적에 수렴하는 것을 방지할 수 있다[26]. 따라서, 다양성이 높은 모집단의 유지는 진화 알고리즘의 장기적 성공에 중요한 요소이다.

EEA의 구현을 위해 작업할당 모집단(Pop-B), 투입순서 모집단(Pop-S), 그리고 작업할당-투입순서 모집단(Pop-BS)을 구성한다. Pop-B와 Pop-S는 각각 작업할당과 모델의 투입순서를 나타내는 개체들로 구성된 모집단으로서 전체문제에 대한 부분해를 나타낸다. 두 모집단에 있는 개체들은 서로 분리되어 있지만, 공진화하는 동안 상대 모집단의 개체와 공생한다. Pop-BS는 전체해를 표현하는 개체인 내공생자(endosymbiont)로 구성된다. 내공생자는 두 공생자의 게놈(genome)을 모두 포함한다. 즉, 내공생자는 공생자들의 통합에 의해 형성된 개체로서 작업할당과 모델의 투입순서가 결합된 전체해를 나타낸다. Pop-BS의 개체들은 Pop-B와 Pop-S에 있는 개체들과 적응도를 기준으로 경쟁함과 동시에 Pop-BS 내에서 두 부분해가 결합한 상태를 유지하면서 진화한다.

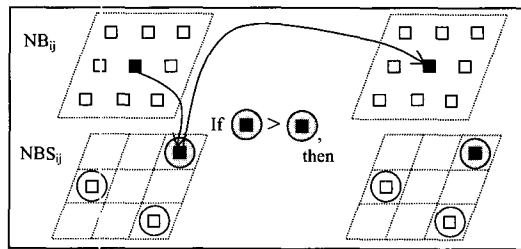
EEA에서 모집단과 이웃은 다음과 같이 구성한다. 세 모집단 Pop-B, Pop-S, 그리고 Pop-BS는 토러스 형태의 2차원 격자구조를 갖는다고 본다. 이웃의 크기와 구조는 다양하게 정의할 수 있다[25]. 본 연구에서는 3×3 격자의 이웃구조를 사용한다. 그리고 NB_{ij} , NS_{ij} , NBS_{ij} 를 각각 Pop-B, Pop-S, Pop-BS에서 모집단의 위치 (i, j) 와 그 주위 8개 개체로 구성된 3×3 격자의 이웃이라고 정의한다. 초기에는 Pop-B와 Pop-S의 각 셀에는 부분해를 나타내는 개체들을 임의로 할당하며, Pop-BS는 공집합으로 둔다.

세 모집단간의 상호작용은 다음 세 가지 형태로 행하여진다. 첫째, NB_{ij} 와 NS_{ij} 의 개체들은 서로 공생관계를 갖는다. NB_{ij} 에 있는 개체들은 NS_{ij} 에 있는 개체와 공생하며, NS_{ij} 에 있는 개체 또한 NB_{ij} 에 있는 개체와 공생한다. 이때 가장 높은 적응도를 갖는 개체들의 결합이 잠재내공생자(potential endosymbiont)가 된다. 이러한 과정을 도식화하면 [그림 4](a)와 같다. 둘째, 잠재내공생자는 NBS_{ij} 의 내공생자와 적응도를 기준으로 경쟁한다. 이 과정은 다음과 같다. es_{new} 를 잠재내공생자라고 하고, es_{old} 를 NBS_{ij} 에 있는 내공생자 중에서 가장

낮은 적응도를 갖는 내공생자라 하자. 만약, es_{new} 의 적응도가 es_{old} 의 적응도보다 높으면, es_{new} 는 NBS_{ij} 에 있는 es_{old} 의 위치로 이동하여 비로소 내공생자가 된다. 그리고 es_{old} 는 두 공생자를 서로 분리하여 NB_{ij} 와 NS_{ij} 에서 es_{new} 를 생성하여 빈셀에 위치시킨다([그림4](b) 참조). 그렇지 않으면 이 과정은 일어나지 않고, 잠재내공생자를 구성하는 공생자들은 분리된 상태로 현재의 위치에 존재한다. 셋째, NBS_{ij} 의 내공생자들은 NB_{ij} 또는 NS_{ij} 에 있는 공생자들과 경쟁한다. NBS_{ij} 에 있는 내공생자의 NB_{ij} 에 해당하는 계놈을 NB_{ij} 에 있는 공생자로 변경하여 내공생자의 적응도가 높아지면 이 계놈을 NBS_{ij} 의 내공생자로 이동시키고, 원래 내공생자를 구성하는 Pop-B의 계놈을 NB_{ij} 로 이동시킨다([그림 4](c) 참조).



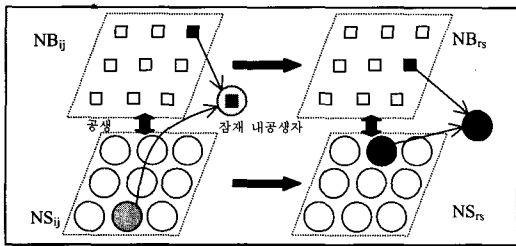
(b) 기존 내공생자와 잠재 내공생자의 경쟁



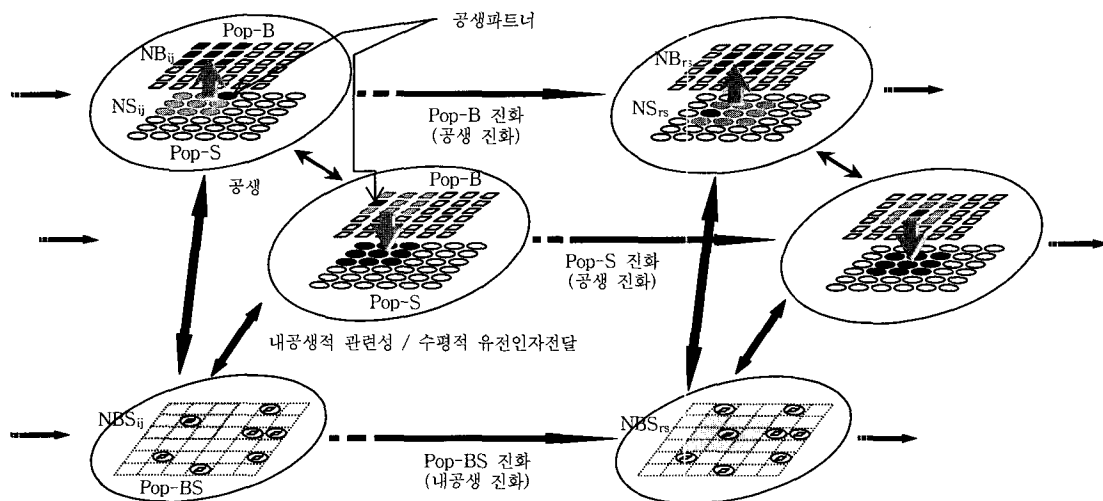
(c) NBS_{ij} 의 내공생자와 NB_{ij} 의 공생자간 경쟁

[그림 4] 부분해와 완전해의 상호작용

이상의 상호작용에서 분리된 부분해들이 전체해로 결합하는 것은 내공생과정을 모방한 것이고, 공생자들과 내공생자들이 상호간의 모집단으로 이동하는 것은 수평적 유전인자전달 개념을 모방한 것이다. 본 연구에서 제안한 EEA의 전체적인 진화 과정을 도식화하면 [그림 5]와 같다.



(a) NB_{ij} 와 NS_{ij} 의 공생과 잠재 내공생자 생성



[그림 5] 내공생 진화알고리즘

3.3 절차

EEA의 단계별 절차는 다음과 같다. 아래 절차에서 b_c 와 s_t 는 각각 Pop-B와 Pop-S의 개체(분리된 공생자)를 나타내고, $\beta_u \sigma_u$ 는 Pop-BS에 있는 개체(내공생자)를 나타낸다. 또한 $f(b|s)$ 는 개체 b 가 자신의 공생파트너 s 와 결합할 때 갖는 적응도를, $f(\beta\sigma)$ 는 내공생자 $\beta\sigma$ 의 적응도를 나타낸다.

단계 1 : 초기화

Pop-B와 Pop-S의 모든 셀에 두 문제의 각 부분해들을 임의로 생성한다. 그리고 두 모집단의 개체들을 격자구조의 같은 위치에 있는 개체들과 짝지어 초기 적응도를 평가하고, 가장 좋은 해를 f_{best} 로 둔다. 모집단 Pop-BS는 공집합으로 둔다.

단계 2 : 이웃설정

임의의 위치 (i, j) 를 선택하여 이웃 NB_{ij} , NS_{ij} , NBS_{ij} 를 정의한다. 만약 NBS_{ij} 가 공집합이면 단계 4로 간다.

단계 3 : NBS_{ij} 와 NB_{ij} , NS_{ij} 간의 경쟁

단계 3.1 : NBS_{ij} 와 NB_{ij} 의 경쟁

NBS_{ij} 의 각 $\beta_u \sigma_u$ 에 대해, $f(b_c \sigma_u)$ 가 $f(\beta_u \sigma_u)$ 보다 큰 NB_{ij} 의 개체 b_c 가 존재하면, NBS_{ij} 에 있는 $\beta_u \sigma_u$ 는 $b_c \sigma_u$ 와 대체하고, NB_{ij} 에 있는 b_c 는 β_u 와 대체한다.

단계 3.2 : NBS_{ij} 와 NS_{ij} 의 경쟁

NBS_{ij} 의 각 $\beta_u \sigma_u$ 에 대해, $f(\beta_u s_t)$ 가 $f(\beta_u \sigma_u)$ 보다 큰 NS_{ij} 의 개체 s_t 가 존재하면, NBS_{ij} 에 있는 $\beta_u \sigma_u$ 는 $\beta_u s_t$ 와 대체하고, NS_{ij} 에 있는 s_t 는 σ_u 와 대체한다.

단계 4 : NB_{ij} , NS_{ij} 간의 협조와 잠재내공생자의 생성

단계 4.1 : NB_{ij} 와 NS_{ij} 개체들의 적응도를 평가한다. 이를 위하여 NS_{ij} 와 NB_{ij} 내에서 각각 공생파트너를 선택한다.

단계 4.2 : 단계 4.1의 과정을 통해 얻은 가장 높

은 적응도 조합을 잠재내공생자 $b_p s_p$ 라 두자. $f(b_p s_p)$ 가 f_{best} 보다 크면, f_{best} 를 갱신한다.

단계 5 : 기존 내공생자와 잠재내공생자 간의 경쟁

단계 5.1 : NBS_{ij} 가 공집합이면, $b_p s_p$ 를 NBS_{ij} 의 (i, j) 번째 위치의 셀에 삽입한다. 그리고 각각 NB_{ij} 와 NS_{ij} 에서 b_p 와 s_p 를 제거하고, 단계 6으로 간다.

단계 5.2 : NBS_{ij} 가 공집합이 아니고, $f(b_p s_p)$ 가 NBS_{ij} 내에 존재하는 가장 낮은 내공생자의 적응도인 $f(\beta_v \sigma_v)$ 보다 크면, $b_p s_p$ 와 $\beta_v \sigma_v$ 를 대체한다. 내공생자 $\beta_v \sigma_v$ 는 각각 β_v 와 σ_v 로 분리하여, b_p 와 s_p 가 위치한 곳에 각기 두고, 이들의 적응도는 $f(\beta_v \cdot) = f(\sigma_v \cdot) = f(\beta_v \sigma_v)$ 로 둔다.

단계 6 : NB_{ij} 의 진화

함수 $genetic_evolution(NB_{ij}, R_c, I_m, G_m, f_{best})$ 를 호출하여 수행한다.

단계 7 : NS_{ij} 의 진화

함수 $genetic_evolution(NS_{ij}, R_c, I_m, G_m, f_{best})$ 를 호출하여 수행한다.

단계 8 : 단기 진화 종료조건

단기 진화 종료조건을 만족하면 단계 9로 가고, 그렇지 않으면 단계 2로 간다.

단계 9 : Pop-BS의 진화

함수 $genetic_evolution(Pop-BS, R_c, I_m, G_m, f_{best})$ 를 호출하여 수행한다.

단계 10 : 장기 진화 종료조건

장기 진화 종료조건을 만족하면 알고리즘을 종료하고, 그렇지 않으면 단계 2로 간다.

위의 EEA 절차 중 단계 6, 단계 7, 단계 9는 다음의 과정을 통하여 수행된다.

함수 $genetic_evolution(P, R_c, I_m, G_m, f_{best})$

부 단계 1 : 먼저, 선별전략에 따라 모집단 P 에서 재생산에 참여할 개체들을 선별한다. 그리고

선별된 개체들에 대하여 난수(random number)를 발생시킨다. 만약, 난수가 R_c (교차율)보다 작거나 같으면, 그 개체는 교차연산의 후보자가 된다. 그리고 후보 개체들의 쌍에 대하여 교차연산을 수행하여 자손 개체들을 생산한다.

부 단계 2 : 대체할 개체들을 모집단 P 로부터 선택한다. 이 때 낮은 적응도를 가진 개체일수록 선택될 확률을 높게 둔다. 선택된 개체들은 새롭게 생산된 자손 개체들과 대체된다.

부 단계 3 : 모집단 P 로부터 I_m (개체단위 돌연변이율)보다 낮은 개체들을 선택하고, G_m (인자단위 돌연변이율)을 기준으로 각 인자에 대하여 돌연변이 연산을 수행한다.

부 단계 4 : 새롭게 생성된 개체들의 적응도를 평가하고, 가장 좋은 개체의 적응도가 f_{best} 보다 높으면 f_{best} 를 갱신한다.

단계 3은 진화가 이루어지고 있는 이웃내에서 좀더 높은 적응도를 갖는 내공생자를 생성하기 위한 것이며, 단계 4는 NB_{ij} 와 NS_{ij} 에 있는 개체들의 적응도를 평가하며 잠재내공생자를 결정하는 단계이다. 단계 4.1에서 NB_{ij} 와 NS_{ij} 에 있는 개체의 적응도를 평가하기 위해서는 공생파트너가 설정되어야 한다. 공생파트너를 선택하는 방법에는 여러 가지가 존재한다[20, 6]. Kim *et al.*[2]의 연구결과에 의하면 공생파트너의 선택방법에 따른 성능비교가 통계적으로 유의한 차이를 보이지 않았다. 그러므로 본 연구에서는 단순하면서 계산시간이 가장 적게 소요되는 전략으로, 상대 모집단(또는 상대이웃)에서 임의로 한 개체를 선택하여 공생파트너로 취하는 전략을 사용한다. 단계 5는 잠재내공생자와 기존 내공생자의 적응도를 비교하여 잠재내공생자가 내공생자가 되어 NBS_{ij} 로 이동할 것인지 결정하는 단계이다. 단계 6과 7은 각각 Pop-B와 Pop-S가 진화하는 단계로서, 본 연구에서는 안정상태 유전알고리즘을 사용한다. 안정상태 유전알고리즘은 단순 유전알고리즘보다 우수한 탐색 효율을 나타내는 것으로 알려져 있다[28, 32]. 단계

8의 조건은 Pop-BS의 진화 시작을 결정하기 위한 것이다. EEA에서는 Pop-BS에 일정 개수 이상의 내공생자들이 만들어지면 진화를 시작하는 것으로 한다. 단계 9는 Pop-BS의 진화를 나타낸다. Pop-BS의 크기는 Pop-B나 Pop-S에 비하여 상대적으로 작기 때문에 이웃진화 전략보다는 전통적인 진화 알고리즘에서 사용되는 모집단진화전략을 사용하기로 한다.

일반적으로, 진화알고리즘을 특정 문제에 적용하려면 유전표현, 유전연산자, 적응도 평가, 유전파라미터 등을 포함한 여러 요소들이 필요하다. Pop-B와 Pop-S는 각각 MMAL에서 서로 다른 특성을 갖는 작업할당과 모델투입순서를 나타내므로 이것을 다루기 위한 효율적인 방법들이 필요로 한다. Pop-B와 Pop-S의 유전표현과 유전연산은 각각 4장과 5장에서 설명하며, Pop-BS의 유전표현과 유전연산은 6장에서 설명한다. 그리고 개체의 적응도 평가시 평가함수로 식 (1)을 사용한다. 모집단과 이웃의 크기, 교차율과 돌연변이율, 종료조건 등은 실험을 통해 결정하였다. 실험에 사용된 파라미터들은 실험결과와 함께 7장에서 언급한다.

4. MMAL의 작업할당을 위한 유전표현과 유전연산

4.1 유전표현과 초기모집단

잠재해들의 유전표현은 주로 해결하고자 하는 문제의 특성에 의해 결정된다. 본 연구에서는 작업할당문제에 대하여 그룹번호표현을 사용한다. 작업할당 모집단(Pop-B)의 한 개체의 길이는 I (작업의 수)이며, 각 인자는 1과 J (작업장 수)사이의 정수값을 갖는다. 만약 i 번째 인자값이 j 이면 작업 i 는 작업장 j 에 할당된다. 예를 들어 작업들의 집합 {1,2,3}, {4,6,7}, {5,8,9,10}이 작업장 1, 2, 3에 각각 할당되었다면 이때 개체는 (1 1 1 2 3 2 2 3 3 3)로 표현된다. 이러한 표현은 별도의 해석 방법이 필요하지 않으며 작업할당의 정보를 쉽게 이용할 수 있

다는 장점이 있다.

4.2 유전연산자

교차연산자는 문제의 특성을 잘 반영할 수 있어야 하며, 부모가 갖고 있는 좋은 정보를 자손에게 전달할 수 있어야 한다. 또한, 재생산된 자손 개체들은 작업할당문제의 가능해 유지 조건을 만족시켜야 한다. 이를 위하여 본 연구에서는 구조교차를 작업할당문제에 적합하게 변형하여 사용하였다. 이 방법의 절차는 다음과 같다.

- 단계 1 : 범위 $[1, J]$ 에서 임의의 정수 r 을 선택한다.
- 단계 2 : 부모 P1에서 작업장 1부터 r 까지 표현한 인자를 자손 O1의 동일한 위치에 상속한다.
- 단계 3 : 자손 O1의 남은 위치에 부모 P2에 있는 $r+1$ 부터 J 까지 표현한 인자를 상속한다.
- 단계 4 : 자손 O1의 비어있는 위치는 아래에 설명된 재할당 방법을 이용하여 할당한다.

다른 자손 O2는 P1과 P2의 역할을 바꾸어 생산한다.

돌연변이는 진화알고리즘에서 탐색 능력을 향상시키는 데 중요한 역할을 한다. 작업할당문제에서는 개체 돌연변이율에 의해 개체를 선택하고 선택된 개체에 대하여 인자 돌연변이율에 의해 인자를 돌연변이시킨다. 이들 인자의 돌연변이는 다음에서 설명될 재할당 절차에 의해 이루어진다.

위에서 설명한 교차와 돌연변이 방법에서 어떤 인자들은 비어있게 된다. 따라서 비어있는 인자들에 작업장을 할당해야 한다. 이것은 재할당 절차에 의해 이루어진다. 제안된 재할당 기법에서는 선행 제약의 만족은 물론 작업장의 작업량이 평활화 되도록 하였다. 재할당 기법의 절차를 보이기 위해서, 먼저 여기에 사용되는 기호를 정의하자.

t_{im} : 모델 m 에서 작업 i 의 생산시간($i=1,2,\dots,I, m=1,2,\dots,M$).

\hat{t}_i : 작업 i 의 MPS 생산시간 ($= \sum_{m=1}^M d_m t_{im}$).

\bar{T} : 한 MPS 사이클 동안 작업장의 작업부하의 평균 ($= (1/J) \sum_{i=1}^I \hat{t}_i$).

R : 미할당되었거나 재할당이 요구되는 작업집합.

$IP(i)$: 작업 i 의 직선행작업집합.

$S(i)$: 작업 i 의 후행작업집합.

$NR(i)$: 집합 R 의 원소 중 작업 i 의 직선행작업집합 ; $NR(i) = R \cap IP(i)$.

E_i : 개체에서 $IP(i)$ 의 작업들에 해당하는 인자값 중 가장 작은 값, 만약 $IP(i) = \emptyset$ 이면, $E_i = 1$.

L_i : 개체에서 $S(i) - R$ 의 작업들에 해당하는 인자값 중 가장 작은 값, 만약 $S(i) - R = \emptyset$ 이면, $L_i = J$.

$AW(i)$: $\{E_i, E_i+1, \dots, L_i\}$ (작업 i 가 할당 가능한 작업장의 집합).

T_j : 작업장 j 에 이미 할당된 작업들의 MPS 생산시간의 합.

단계 1 : 집합 R 과 $NR(i)$ 를 구하고 $T_j, j=1,2,\dots, J$ 를 계산한다.

단계 2 : 집합 R 의 원소이고, $NR(i) = \emptyset$ 인 작업 중 MPS 생산동안의 작업시간이 가장 큰 작업 i^* 를 선택한다.

단계 3 : i^* 의 E_{i^*} 와 L_{i^*} 를 구하고, i^* 의 할당가능 인자값 집합 $AW(i^*)$ 를 구한다.

단계 4 : 집합 $K = \{k | T_k + \hat{t}_{i^*} \leq \bar{T} \text{ and } k \in AW(i^*)\}$ 이라 하자. 만약 $K \neq \emptyset$ 이면, K 내에서 가장 작은 값을 k^* 로 두고, 그렇지 않으면 가장 작은 T_j 를 갖는 작업장을 k^* 로 둔다. 동일한 경우 가장 작은 작업장 번호를 선택한다.

단계 5 : 작업 i^* 의 해당 인자값을 k^* 로 두고 i^* 를 집합 R 에서 삭제한다. $R = \emptyset$ 이면 종료하고, 그렇지 않으면 $NR(i), i \in R$ 와 $T_j, j=1,2,\dots, J$ 를 갱신하고 단계 2로 간다.

위의 재할당 기법은 best-fit-decreasing(BFD) 규칙[10]을 작업할당 문제에 적합하게 수정한 것

이다. BFD기법은 일종의 발견적기법(heuristic)으로 그룹분할문제에서 좋은 결과를 얻는 것으로 알려져 있다.

5. MMAL의 투입순서결정을 위한 유전표현과 유전연산

5.1 유전표현과 초기모집단

투입순서 모집단(Pop-S)의 개체들은 간단히 MPS 생산동안 생산되는 모델들의 투입순서로 표현한다. 예를 들어, MPS가 $d=(d_A, d_B, d_C)=(1, 2, 3)$ 이고 투입순서가 A B C B C C이면 개체표현은 (A B C B C C)가 된다. 이 개체는 투입되는 모델의 순서대로 자연스럽게 표현됨으로써 개체해석과 유전인자의 정보이용이 용이하다. 초기 모집단은 MPS 제약을 만족하는 개체들을 모집단 크기만큼 임의로 생성하여 구성한다.

5.2 유전연산자

본 연구에서는 투입순서문제를 위한 유전연산자로 ISRX(immediate successor relation crossover)와 역순(inversion) 연산자를 사용한다. 혼합모델 생산라인의 투입순서문제에서 이들 두 연산자를 결합하여 사용함으로써 좋은 탐색 성능을 보인 것으로 나타났다[14]. ISRX는 인접인자 재결합 교차(edge recombination crossover)[33]를 변형한 것으로 부모 개체들의 직후행 관계 정보를 이용하여 자손을 생산하는 교차 연산자이다. 인자 A의 직후행인자는 A 바로 다음에 위치한 인자이다. 마지막 인자의 직후행인자는 정의에 의해 첫 번째 인자가 된다. 이러한 관계는 아래의 <표 1>과 같이 요약될 수 있다. 두 부모 개체가 각각 $P1=(A A B B B C C C)$ 와 $P2=(A B C A B C B C C)$ 일 때, 초기 ISRX 표는 <표 1>(a)와 같다. <표 1>(a)에는 각 인자형에 대한 모든 직후행인자들이 나열되어 있다. 예를 들어, 인자형 A는 P1의 첫

번째와 두 번째 위치, P2의 첫 번째와 네 번째 위치에 있으며 각각의 직후행인자는 A, B, B, B이다. 그리고 인자형 A, B, C의 직후행인자들의 개수는 각각 4, 6, 8이므로 인자비율은 2:3:4가 된다. ISRX를 이용하여 자손을 생산하는 절차는 아래와 같다.

단계 1: ISRX 표를 만든다.

단계 2: $i=1$ 로 두고 임의로 인자형(e_i)를 선택한다.

단계 3: 자손의 i 번째 위치에 e_i 를 유전시킨다. 만약 하나의 자손이 완전히 생산되었으면 끝내고 아니면 다음 단계로 간다.

단계 4: 직후행인자의 열에서 인자형 e_i 의 원소 2개를 임의로 삭제하고, 인자비율을 수정한다.

단계 5: i 를 1 증가시키고, 단계 4에서 삭제된 인자형의 행에서 가장 많이 남아있는 인자형을 선택한다. 동일한 경우 인자비율이 작은 것을 선택하고, 이것 또한 동일한 경우 임의로 선택한다. 선택된 인자를 e_i 로 둔다. 단계 3으로 간다.

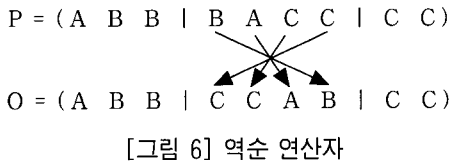
위 절차에서 단계 5는 직후행 관계를 기준으로 다음에 선택될 인자형을 결정하는 단계이다. 인자형 B가 초기에 임의로 선택되었다고 할 때, 처음 4회 반복시행 절차가 <표 1>에 나타나 있다. <표 1>(c)에서 인자형 C는 직후행인자로서 3개의 A, 1개의 B, 3개의 C를 가진다. A와 C의 개수가 같고 인자비율이 2:3이므로 인자형 A가 선택된다. 이는 미할당된 인자형 A와 C의 수가 각각 4개, 6개(인자비율 A:C=2:3)이고 인자형 C의 직후행 인자형 A와 C의 수가 같으므로 인자형 C는 C보다 A에 가깝다고 할 수 있다. 이러한 절차를 통해 자손개체(B C A B C C B C A)를 얻을 수 있다. 다른 자손개체는 다른 초기 인자형을 선택하여 위의 절차를 반복함으로써 얻을 수 있다. ISRX에 의해 생산된 자손개체는 부모개체의 직후행 관계를 상속받게 된다.

역순 돌연변이는 부모에서 두 개의 절단점을 선택하고, 절단점 사이의 인자들을 역순으로 하여 자

〈표 1〉 ISRX 표

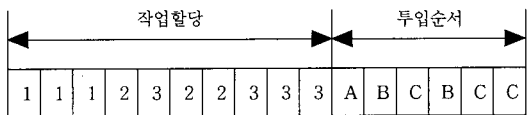
인자형	(a)		(b)		(c)		(d)	
	직후행 인자	인자 비율	직후행 인자	인자 비율	직후행 인자	인자 비율	직후행 인자	인자 비율
A	ABBB	2	ABB	2	ABB	2	ABB	1
B	BBCCCC	3	BCCCC	2	BCCC	2	BCCC	2
C	CCCAABCA	4	CCCAABCA	4	CCAABCA	3	CCBCA	3
선택된 인자형	B(임의선택)		C		A		B	

손을 생산한다. 이러한 역순 돌연변이의 예는 [그림 6]과 같다. 그림에서 P와 O는 각각 부모와 자손 개체이다.



6. Pop-BS의 유전표현과 유전연산

작업할당-투입순서 모집단(Pop-BS)에서 개체는 내공생자로 구성되어 있다. 내공생자는 Pop-B에 있는 개체의 계놈과 Pop-S에 있는 개체의 계놈이 결합된 형태로 이루어져 있다. 즉, 내공생자는 MMAL에서 작업할당과 투입순서가 결합하여 통합된 형태로 표현되어 있다. 이러한 유전표현은 4장과 5장에서 사용된 유전표현을 그대로 사용함으로써 쉽게 나타낼 수 있다. Pop-BS의 개체는 [그림 7]에서 보여지는 것처럼 앞부분은 작업할당을 나타내며 뒷부분은 투입순서를 나타낸다.



[그림 7] Pop-BS의 유전표현

두 내공생자에서 사용되는 표현이 서로 다르기 때문에 Pop-BS 모집단에서 사용되는 유전연산은

두 단계에 의해 실행된다. 첫 단계는 작업할당을 나타내는 개체의 앞부분을 위한 것이며, 두 번째 단계는 모델투입순서를 나타내는 개체의 뒷부분을 위한 것이다. 교차연산을 위해 먼저 개체의 앞부분에는 4.2절에서 기술된 수정된 구조교차를 사용하고, 뒷부분에는 ISRX를 사용한다. 돌연변이는 4.3절에서 기술된 것과 동일한 방법을 개체의 앞부분에 사용하고, 5.2절에서 기술된 역순 돌연변이를 개체의 뒷부분에 사용한다.

7. 실험 및 결과분석

7.1 실험문제와 비교알고리즘

제안한 EEA의 성능은 컴퓨터 모의실험을 통해서 분석하였다. 실험 문제는 19개의 작업을 갖는 Thomopoulos문제[30], 61개의 작업을 갖는 Kim *et al.* 문제[1], 그리고 111개의 작업을 가진 Arcus 문제[4]를 사용하였다. 다양한 문제에 대한 실험을 위하여 <표 2>에서 보는 바와 같이 실험문제의 작업장수와 MPS를 변화시켜 실험문제로 사용하였다. Arcus문제의 경우 작업장수가 많은 경우를 실험하기 위하여 작업 95의 작업시간을 334.91에서 66.15로 변경하였다. 실험문제에서 컨베이어의 이동속도(v_c)는 계산의 편이를 위해 1로 하였으며, 제품 투입시간간격(γ)은 1회의 MPS를 생산하는데 요구되는 총 생산시간 ($\sum_{i=1}^I \sum_{m=1}^M t_{im} d_m$)을 제품의 총수($H = \sum_{i=1}^M d_i$)와 작업장수(J)로 나눈 값,

즉 $\gamma = \sum_{i=1}^I \sum_{m=1}^M t_{im} d_m / (H \times J)$ 로 두었다. 각 작업장의 길이는 제품 투입시간간격의 1.5배로 두었다.

본 연구에서는 다음 5가지 알고리즘을 통해 실험결과를 비교 분석하였다.

HGA : 계층적 유전알고리즘(Hierarchical Genetic Algorithm)

TCoA : 강결합 공진화알고리즘(Tightly-coupled Coevolutionary Algorithm)

• LCoA : 약결합 공진화알고리즘(Loosely-coupled Coevolutionary Algorithm)

SNA : 분리된 이웃단위 공생 진화알고리즘(Separated and Neighborhood-based symbiotic evolutionary Algorithm)

EEA : 내공생 진화알고리즘(Endosymbiotic Evolutionary Algorithm)

HGA는 4장에서 소개한 유전 요소들에 의해 작업량 편차 최소화를 목적으로 작업을 작업장에 할당하고, 이 결과로부터 얻은 가장 좋은 작업할당 정보를 이용하여 가의작업 최소화를 목적으로 5장에서 소개한 유전 요소들에 의해 모델의 투입순서 문제를 해결하였다. TCoA와 LCoA는 Maher와 Poon[20]에 의해 제안된 공진화 모형으로서 모집단 단위로 진화하는 것은 동일하지만, 공생파트너의 선택방법이 서로 다른 모형이다. TCoA는 개체의 공생파트너를 항상 상대 모집단의 같은 위치에 있는 개체로 두는 것으로, 진화하는 동안 공생파트

너의 위치가 변화되지 않는다. 이와 달리 LCoA는 매 세대마다 상대 모집단에서 적용도가 가장 높은 개체를 공생파트너로 갖는 방법으로, 이는 진화하는 동안 공생파트너가 변화하게 된다. 전자는 강결합(tightly-coupled), 후자는 약결합(loosely-coupled) 공진화 모형이라 부른다. SNA는 각 부분해로 구성된 모집단을 운용한다. 각 부분해는 분리되어 있는 상대 모집단의 개체들(공생파트너)에 의해 적용도가 평가된다. 또한 이웃단위 진화 전략을 사용한다. SNA는 EEA와 유사하지만, 단지 부분해를 나타내는 개체들이 결합된 형태의 내공생자와 그 모집단을 운용하지 않는다는 점에서 다르다. 즉, SNA는 EEA에서 내공생자와의 상호작용과 그 진화과정을 제외한 것이다.

7.2 파라미터 결정

유전 파라미터들은 예비실험을 통하여 결정하였다. 첫째, 모집단의 크기는 100(10×10의 격자구조)으로 하였다. 둘째, 선별은 크기 2인 토너먼트 선별을 사용하였다. 셋째, 작업할당문제와 투입순서문제에서 교차율은 모두 0.5로 하였다. 단, SNA에서는 재생산 사이클수를 0.5의 확률로 2~3회 반복하였다. 이는 0.5의 교차율을 사용한 것과 같은 효과를 갖는다. 넷째, 작업할당문제와 투입순서문제에서 개체들연변이율은 모두 0.05로 하였으며, 작업할당 문제에서 돌연변이되는 개체의 인자 돌연변이율은 0.1로 두었다. 다섯째, EEA에서 Pop-BS

〈표 2〉 실험문제

문 제	작업수	모델수	작업장수	MPS	문 제	작업수	모델수	작업장수	MPS
Thom1	19	3	3	1 1 1	Arcus3	111	5	12	1 2 4 5 8
Thom2	19	3	3	3 2 1	Arcus4	111	5	12	1 4 8 3 1
Kim1	61	4	6	1 1 1 1	Arcus5	111	5	15	1 1 1 1 1
Kim2	61	4	6	1 3 4 5	Arcus6	111	5	15	5 3 2 1 1
Kim3	61	4	6	6 4 2 1	Arcus7	111	5	15	1 2 4 5 8
Kim4	61	4	12	1 1 1 1	Arcus8	111	5	15	1 4 8 3 1
Kim5	61	4	12	1 3 4 5	Arcus9	111	5	27	1 1 1 1 1
Kim6	61	4	12	6 4 2 1	Arcus10	111	5	27	5 3 2 1 1
Arcus1	111	5	12	1 1 1 1 1	Arcus11	111	5	27	1 2 4 5 8
Arcus2	111	5	12	5 3 2 1 1	Arcus12	111	5	27	1 4 8 3 1

가 진화하기 위한 조건인 단계 8의 단기 진화 종료 조건은 단계 2의 누적수행횟수를 30으로 하였다. 마지막으로, 알고리즘의 종료조건은 생산된 개체수를 사용하였다. 작업수와 모델수에 따라 해공간이 다르므로, Thomopoulous문제, Kim문제, Arcus 문제에 대하여 모든 모집단의 생산 개체수의 합이 각각 5000개, 15000개, 30000개 일 때 종료하였다. SNA에서 공생파트너의 선택은 EEA에서와 같이 상대이웃에서 임의로 한 개체를 선택하는 전략을 사용하였다.

7.3 성능비교

모든 알고리즘은 C++언어로 구현되었으며, 700 MHz Pentium CPU를 장착한 IBM-PC에서 수행되었다. 실험결과는 <표 3>과 같다. 첫 번째 열은 실험문제, 두 번째 열은 계층적 방법에 의해 얻어진 가외작업이고, 세 번째 열부터 여섯 번째 열까지는 네 가지 공생 진화알고리즘에 의한 가외작업이다.

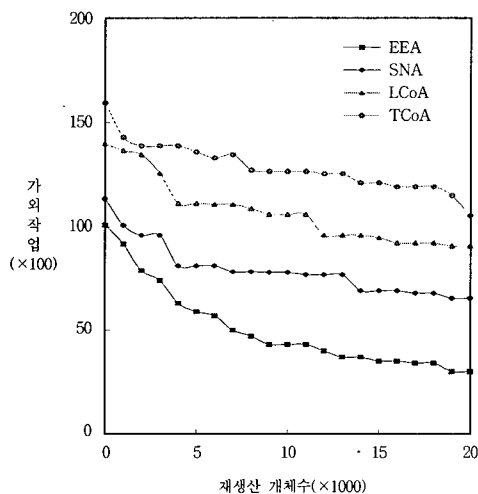
마지막 열은 계층적 방법에 대한 EEA의 개선율로, $\frac{(\text{계층적 방법} - \text{EEA})}{\text{계층적 방법}} \times 100(\%)$ 의 식에 의해 계산하였다. 모든 실험 결과값은 10회 반복 실험의 평균이다.

실험결과로부터 첫째, 모든 실험문제에서 모든 공생 진화알고리즘이 계층적 방법보다는 좋은 해를 탐색하였다. 이것은 부분문제가 통합된 복잡한 문제를 해결하는데 고전적인 방법이 해공간을 효과적으로 탐색하지 못함을 의미한다. 둘째, 모든 문제에 대하여 EEA가 가장 좋은 결과를 보였다. EEA와 SNA의 차이점은 단지 내공생자의 존재 유무이다. EEA는 Pop-B와 Pop-S가 진화하는 동안 얻은 좋은 개체들의 조합인 내공생자를 구성한다. 이러한 내공생자를 구성하는 방법은 전통적인 유전 알고리즘에서 가장 좋은 개체가 항상 생존할 수 있도록 하는 우수개체 보존전략(elitism)과 유사하다. 그러나 EEA에서는 우수한 개체들의 집단이 하나의 모집단을 형성하고 그 자체 모집단이 진화한다

<표 3> EEA와 다른 알고리즘의 비교

문 제	HGA	공진화 알고리즘				개선율(%)
		TCoA	LCoA	SNA	EEA	
Thom1	0.00	0.00	0.00	0.00	0.00	0.00
Thom2	0.40	0.24	0.17	0.00	0.00	100.00
Kim1	1.65	1.25	1.10	0.82	0.59	64.24
Kim2	5.51	4.45	3.26	2.26	1.68	69.51
Kim3	4.76	3.64	2.74	1.53	1.43	69.96
Kim4	6.45	5.01	4.58	3.60	1.08	83.26
Kim5	23.42	18.71	15.14	10.76	2.93	87.49
Kim6	39.71	27.17	23.91	15.13	3.56	91.04
Arcus1	2784.42	1212.75	1181.52	950.57	932.00	66.53
Arcus2	7787.11	4957.09	4846.33	3002.16	2418.19	68.95
Arcus3	2448.67	1166.53	1046.73	968.54	948.96	61.25
Arcus4	9377.36	4792.10	4472.61	3714.65	3003.34	67.97
Arcus5	6477.27	2965.27	2845.91	1845.20	1371.54	78.83
Arcus6	15804.22	10536.33	9097.72	6533.00	4435.69	71.93
Arcus7	4256.50	2634.81	2239.33	1677.44	1520.68	64.27
Arcus8	17362.18	11838.32	10989.59	8934.15	6966.95	59.87
Arcus9	14825.63	7848.08	6939.79	6013.67	5844.50	60.58
Arcus10	46890.72	38971.69	37935.44	26996.73	26607.03	43.26
Arcus11	14013.29	8349.37	7429.33	7343.30	6289.80	55.12
Arcus12	92997.84	48556.41	43722.57	39309.59	37188.74	60.01

는 점에서 우수개체 보존전략과는 다르다. 이러한 내공생자 모집단의 구성으로 내공생자들이 많은 환경의 변화를 겪지 않고 어느 기간동안 안정적으로 공생관계를 유지할 수 있다. 이 논문에서 제안된 알고리즘이 다른 알고리즘에 비해 우수한 성능을 보인 것은 이러한 이유 때문이라고 본다. EEA의 계산시간은 Thomopoulos문제, Kim문제 그리고 Arcus문제에 대하여 평균적으로 각각 2초, 38초, 280초 소요되었으며, EEA가 SNA보다 각 실험 문제에서 10%정도 더 소요되는 것으로 나타났다. 이러한 계산시간의 증가는 내공생자 모집단의 추가에 의한 것으로 보인다. TCoA와 LCoA는 SNA와 계산소요시간에서 큰 차이가 없었다.



[그림 8] 수렴 속도 비교

Arcus4 문제에서 재생산 개체수의 증가에 따른 가장 좋은 적응도 값의 변화를 나타내면 [그림 8]과 같다. 이것을 통해 수렴속도의 관점에서 TCoA, LCoA, SNA, EEA 간의 비교가 가능하다. 그림에서 보여지듯, EEA의 수렴속도는 다른 알고리즘들에 비해 진화 초기부터 빠르게 진행된다. 이러한 결과로부터 EEA는 진화의 어느 시점에 종료하더라도 비교된 기법들보다 항상 좋은 해를 제공함을 알 수 있다. 그리고 이러한 결과는 다른 실험문제에서도 유사하게 나타났다. 특히, EEA와 SNA의

성능비교를 통하여 알고리즘에서 내공생자의 도입이 해의 탐색능력에 미치는 효과를 알 수 있다. 앞에서 언급하였듯이 EEA와 SNA의 차이는 내공생자의 유무이다. 즉, EEA는 SNA의 진화전략에 내공생자를 도입하여 이들 내공생자가 분리된 공생자와 상호작용하면서 진화하는 알고리즘이다. <표 3>과 [그림 8]로 부터 내공생자의 도입이 해를 향상시킴을 알 수 있다.

8. 결 론

본 연구에서는 MMAL의 작업할당과 모델투입 순서를 동시에 결정할 수 있는 새로운 방법론을 제시하였다. 대부분의 기존 연구는 MMAL의 작업할당과 모델투입순서를 분리하여 독립적으로 다루었다. 본 연구에서는 이들 두 문제의 통합을 위한 방법론으로 공생 진화알고리즘의 일종인 내공생 진화알고리즘을 기본 틀로 하였다. 내공생 진화알고리즘은 기존의 공생 진화알고리즘에 원핵세포가 내공생을 통해 진핵세포로 진화하는 과정을 개념적으로 모방한 진화전략을 포함시킨 알고리즘이다. 이와 같은 전략 이외에, 본 연구에서는 모집단의 다양성 유지와 해공간의 탐색 능력의 향상을 위하여 분리된 공생자(부분해)와 내공생자(전체해) 모집단은 안정상태 유전알고리즘을 기반으로 개체를 재생산하였고, 분리된 공생자 모집단들은 이웃단위 진화전략을 사용하였다. 그리고 작업할당과 모델투입 순서에 적합한 유전 연산자를 제시하였다.

실험 결과, 제안된 내공생 진화알고리즘은 고전적 방법인 계층적 접근방법뿐만 아니라 비교된 공생 진화알고리즘들보다 좋은 성능을 나타내었다. 특히, 진화의 초기 단계부터 진화시간의 흐름에 상관 없이 항상 좋은 해를 생산하였다. 이와 같은 주된 이유는 알고리즘에서 내공생자의 상호작용과 진화때문인 것으로 분석되었다. 진화알고리즘이 갖는 적용의 유연성에 의해, 제안한 알고리즘은 다루려는 문제의 적절한 유전표현이 가능하다면 여러 분야에서 상호 관련된 여러 부분문제로 구성된 문

제를 통합적으로 해결하는데 적용 가능할 것이다.

참 고 문 헌

- [1] 김여근, 이수연, 김용주, “혼합모델 조립라인에서 작업부하의 평활화를 위한 유전알고리즘”, 『대한산업공학회지』, 제23권, 제3호(1997), pp. 515-532.
- [2] 김재윤, 김여근, 신태호, “공생진화 알고리즘에서의 공생파트너 선택전략 분석”, 『한국경영과학회지』, 제25권, 제4호(2000), pp.67-80.
- [3] Agnetis, A., C. Arbib, and K.E. Stecke, “optimal two-machine scheduling in a flexible flow system,” *Proceedings of the 2nd International Conference on Computer Integrated Manufacturing*, Rennselaer Polytechnic Institute, Troy, New York, 1990.
- [4] Arcus, A.L., “An analysis of a computer method of sequencing assembly line operations,” Ph.D. dissertation, University of California, 1963.
- [5] 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.
- [6] Bull, L., “Evolutionary computing in multi-agent environments : partners,” *Proceedings 7th International on Conference Genetic Algorithms*, East Lansing, MI, (1997), pp.370-377.
- [7] Campbell, N.A., L.G. Mitchell, and J.B. Reece, *Biology : Concepts & Connections*, Second Edition, Benjamin/Cummings Publishing Company Inc., Redwood City, CA, 1996.
- [8] Chakravarty, A.K. and A. Shtub, “Balancing mixed model lines with in-process inventory,” *Management Science*, Vol.31(1985), pp.1161-1174.
- [9] Dar-El, E.M. and A. Navidi, “A mixed-model sequencing application,” *International Journal of Production Research*, Vol.19(1981), pp.69-84.
- [10] Falkenauer, E., “A new representation and operators for genetic algorithms applied to grouping problems,” *Evolutionary Computation*, Vol.2(1994), pp.123-144.
- [11] Gutjahr, A.L. and G.L. Nemhauser, “An algorithm for the line balancing problem,” *Management Science*, Vol.11(1964), pp.308-315.
- [12] Kilbridge, M. and L. Wester, “The assembly line model-mix sequencing problem,” *Proceeding of the 3rd International Conference on Operations Research*, Oslo, Paris, (1963), pp.247-260.
- [13] Kim, J.Y., Y. Kim, and Y.K. Kim, “An endosymbiotic evolutionary algorithm for optimization,” *Applied Intelligence*, Vol.15 (2001), pp.117-130.
- [14] Kim, Y.K., C.J. Hyun, and Y. Kim, “Sequencing in mixed model assembly lines : a genetic algorithm approach,” *Computers & Operations Research*, Vol.23(1996), pp.1131-1145.
- [15] 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(2000), pp.247-258.
- [16] Kim, Y.K., S.J. Kim, and J.Y. Kim, “Balancing and sequencing mixed-model U-lines with a coevolutionary algorithm,” *Production Planning & Control*, Vol.12(2000), pp.754-764.
- [17] Koza, J.R., *Genetic Programming*, The MIT

- Press, Cambridge, Massachusetts, 1992.
- [18] Lovelock, J. and L. Margulis, "Dr. Lynn Margulis : microbiological collaboration of Gaia," [http : //www.magan.com.au/~prfbrown/gaia_lyn.html](http://www.magan.com.au/~prfbrown/gaia_lyn.html), Mountain Man Graphics, Australia, 1996.
- [19] Macaskill, J.L.C., "Production-line balances for mixed-model lines," *Management Science*, Vol.19(1972), pp.423-434.
- [20] Maher, M.L. and J. Poon, "Modelling design exploration as co-evolution," *Microcomputers in Civil Engineering*, Vol.11(1996), pp.195-210.
- [21] Margulis, L., *Symbiosis in cell evolution*, W.H. Freeman, San Francisco, 1981.
- [22] Moriarty, D.E. and R. Miikkulainen, "Forming neural networks through efficient and adaptive coevolution," *Evolutionary Computation*, Vol.5(1997), pp.373-399.
- [23] Okamura, K. and H. Yamashina, "A heuristic algorithm for the assembly line model-mix sequencing problem to minimize the risk of stopping the conveyor," *International Journal of Production Research*, Vol.17(1979), pp.233-247.
- [24] Potter, M.A., "The design and analysis of a computational model of cooperative coevolution," Ph. D. dissertation, George Mason University, 1997.
- [25] Sarma, J., and K. De Jong, "An analysis of the effects of neighborhood size and shape on local selection algorithm," *Proceedings of the 4th International Conference on Parallel Problem solving from Nature*, (1996), pp.236-244.
- [26] Smith, R.E., S. Forrest, and A.S. Perelson, "Searching for diverse, cooperative populations with genetic algorithms," *Evolutionary Computation*, Vol.1(1993), pp.127-149.
- [27] Stecke, K., and I. Kim, "A study of FMS part type selection approaches for short-term production scheduling," *International Journal of Flexible Manufacturing Systems*, Vol.1(1988), pp.7-29.
- [28] Syswerda, G., "A study of reproduction in generational and steady-state genetic algorithms," *Foundations of Genetic Algorithms*, edited by Gregory J.E. Rawlins, San Mateo, CA, (1991), pp.94-101.
- [29] Thomopoulos, N.T., "Line balancing-sequencing for mixed-model assembly," *Management Science*, Vol.14(1967), pp.59-75.
- [30] Thomopoulos, N.T., "Mixed model line balancing with smoothed station assignment," *Management Science*, Vol.16(1970), pp.593-603.
- [31] Tsai, L., "Mixed model sequencing to minimize utility work and the risk of conveyor or stoppage," *Management Science*, Vol.41(1995), pp.485-495.
- [32] Whitley, D., "The Genitor algorithm and selection pressure : why rank-based allocation of reproductive trials is best," *Proceedings of the 3rd International Conference on Genetic Algorithms and their Application*, (1989), pp.116-121.
- [33] Whitley, D., T. Starkweather, D. Fuquay, "Scheduling problems and traveling salesmen : the genetic edge recombination operator," *Proceedings of the 3rd International Conference on Genetic Algorithms*, (1989), pp.133-140.
- [34] Yaro, C.A. and R. Rachamadugu, "Sequencing to minimize work overload in assembly lines with product options," *Management Science*, Vol.37(1991), pp.572-586.