

진화알고리즘을 이용한 혼합모델 U라인의 작업할당과 투입순서 결정[†] Balancing and sequencing mixed-model U-lines using evolutionary algorithm

김재윤 · 김여근
전남대학교 산업공학과

Abstract

This paper presents a new method that can efficiently solve the integrated problem of line balancing and model sequencing in mixed-model U-lines (MMULs). Balancing and sequencing problem are important for an efficient use of MMULs and 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 evolutionary algorithm, is adopted as a methodology in order to solve the two problems simultaneously. Some evolutionary schemes are used in the algorithm to promote parallel search capability, rapidity of convergence and population diversity. The proposed algorithm is compared with the existing evolutionary algorithms in terms of solution quality. The experimental results confirm the effectiveness of our approach.

1. 서론

본 연구에서는 혼합모델 U라인(mixed-model U-line: MMUL)에서의 작업할당과 투입순서문제를 다룬다. MMUL에서 라인의 효율적인 이용을 위하여 작업할당문제와 투입순서문제는 중요하다(Sparling & Miltenburg, 1998). 작업할당문제는 제품을 생산하기 위한 여러 작업들을 작업장에 할당하는 문제이며, 투입순서문제는 모델의 생산순서를 결정하는 문제이다.

MMUL에서 작업할당과 투입순서는 관련성을 갖는다. 그러나 기존 대부분의 연구에서는 두 문제를 독립적으로 다루거나, 전통적인 혼합모델 직선라인(mixed-model straight line: MMSL)에 관한 연구들이 주로 이루어졌다. 조사한 바에 의하면, MMUL의 작업할당문제는 Sparling & Miltenburg(1998)의 연구가 유일하다. JIT시스템에서의 혼합모델 투입순서에 관한 연구는 Miltenburg(1989), Monden(1993), McMullen(1998) 등의 연구가 있다. 본 연구에서는 MMUL의 작업할당과 투입순서를 동시에 결정하는 새로운 기법을 제안하고자 한다.

MMUL에서 작업할당과 투입순서문제의 통합 연구는 본 논문의 저자들(Kim *et al.*, 2000)에 의해 처음으로 이루어졌다. 이들의 연구에서는 공생(symbiotic) 진화알고리즘이 방법론으로 사용되었다. 공생 진화알고리즘은 진화알고리즘의 특별한 형태로, 공생적 상호관계를 통하여 공진화하는 자연현상을 모방한 확률적 탐색기법이다. 공생 진화알고리즘을 MMUL의 작업할당과 투입순서 문제에 적용하기 위하여, 두 문제는 각각 부분 문제로서 독립된 하나의 종으로 간주되고, 각

모집단은 각 종을 유지하기 위하여 존재한다. 공생 진화알고리즘은 복잡하고 동적인 문제, 특히 특성이 다른 여러 문제들이 관련성을 가지고 결합되어 있을 때, 이를 통합적으로 해결하는데 유용한 기법으로 알려져 있다(Moriarty & Miikkulainen, 1997).

본 연구에서는 MMUL의 작업할당과 투입순서 문제를 해결하기 위한 또 다른 방법론으로 내공생(endosymbiotic) 진화알고리즘을 제안한다. 내공생 진화알고리즘은 자연계의 내공생 진화과정(Margulis, 1980)을 모방한 것이다. 이는 이전 연구에서 소개한 공생 진화알고리즘을 확장한 알고리즘으로, 공생 진화알고리즘에서 발견된 문제점들을 개선하기 위해 개발하였다. 본 연구에서는 내공생 진화알고리즘이 어떻게 MMUL의 작업할당과 투입순서의 통합 문제에 적용되는지 보일 것이며, 해의 길 측면에서 제안한 알고리즘과 기존 진화알고리즘들을 비교할 것이다.

2. MMUL의 작업할당과 투입순서

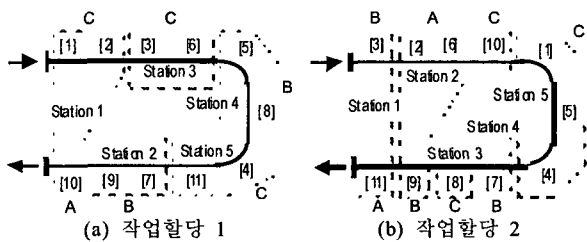
본 연구에서는 MMUL이 다음과 같은 상황에서 운영된다고 가정한다. 첫째, 라인은 고정된 수의 작업장을 가지며, 각 작업장에는 단지 한 사람의 작업자가 할당되어 있다. 둘째, 사이클 생산방식을 취한다. 전체 계획생산기간동안 M 종류의 모델이 생산되며, 이 기간동안 모델 m 의 수요를 $D_m(m=1, 2, \dots, M)$ 이라 하자. 그리고 D_1, D_2, \dots, D_M 의 최대 공약수가 h 이면, $d = (D_1/h, \dots, D_M/h) = (d_1, \dots, d_M)$ 으로 둔다. 이 d 를 최소부품집합(minimum part set: MPS)이라 부르고, 이를 반복생산한다. 즉, MPS를 h 번 반복생산하여 총 수요를 만족한다. 본 연구에서 다루는 모든 투입순서는 MPS를 기준으로 하며, 이는 혼합모델 생산라인에서 흔히 사용한다(Bard *et al.*, 1992). 그리고 작업자의 이동시간은 무시한다.

<그림 1>은 MMUL에서 MPS $d = (d_A, d_B, d_C) = (1, 2, 3)$, 모델의 투입순서는 ABCBCC로 가정하고 보인 각 작업장에 할당된 작업과 생산 모델이다. 특히, <그림 1>은 동일한 투입순서를 갖지만, 서로 다른 작업할당이 이루어진 상황을 보인 것이다. 이 예에서 작업장수는 5개로 고정되어 있고, 작업간의 선후행관계 및 모델별 작업시간은 <표 1>에 표현되어 있다.

MMUL의 작업할당과 투입순서는 다음과 같은 특징을 갖는다. 즉, <그림 1>의 (a)에서 보는 바와 같이 작업장 1의 작업자는 모델 C의 작업 1, 2와 모델 A의 작업 10을 수행한다. 여기서, 작업장 1의 작업 10은 선행작업들이 미할당되었지만 할당된 것이며, 이를 후방향(backward) 할당이라 하고, 작업

[†] 본 연구는 한국과학재단 특정기초연구(과제번호:98-0200-09-01-3) 지원으로 수행되었음.

1과 2는 전방향(forward) 할당이라고 한다. 예제를 통하여 MMUL에서 작업할당문제는 작업할당 뿐만 아니라 작업장에 할당된 모델을 변경할 수 있음을 알 수 있다.



(a) 작업할당 1 (b) 작업할당 2
<그림 1> 작업할당의 두가지 예

<표 1> 작업선행관계 및 작업시간

| 작업 번호 | 직선행 작업 | 작업시간 | | |
|-------|--------|------|----|----|
| | | A | B | C |
| 1 | | 3 | 5 | 4 |
| 2 | | 3 | 3 | 0 |
| 3 | | 6 | 6 | 9 |
| 4 | 1 | 5 | 8 | 4 |
| 5 | 2 | 0 | 0 | 6 |
| 6 | 2,3 | 3 | 5 | 5 |
| 7 | 4 | 8 | 2 | 10 |
| 8 | 5 | 8 | 4 | 6 |
| 9 | 7,8 | 2 | 8 | 5 |
| 10 | 6 | 5 | 0 | 5 |
| 11 | 6 | 3 | 2 | 2 |
| 합계 | | 46 | 43 | 56 |

MMUL은 MMSL과 작업할당 측면에서 몇가지 차이점을 갖고 있다. 첫째, MMUL은 전방향 및 후방향 할당이 가능하지만, MMSL에서는 전방향 할당만을 허용한다. 둘째, MMSL에서는 1회의 생산 사이클동안 각 작업장에서는 하나의 모델만이 작업되지만, MMUL에서는 서로 다른 모델들이 한 사이클동안 동일한 작업장에 위치할 수 있다. 예를 들어, <그림 1>에서 보는 바와 같이 전방향 및 후방향 할당이 동시에 이루어진 작업장((a)의 작업장 1, (b)의 작업장 1, 2, 4)에서는 각 작업장에 서로 다른 모델이 할당될 수 있다. 그러므로, MMUL에서 한 사이클동안 작업장의 작업량은 작업할당과 투입순서에 모두 영향을 받는다.

본 연구에서는 각 사이클동안 작업장의 작업량 평활화를 목적으로 MMUL의 작업할당문제와 투입순서문제를 동시에 해결한다. 작업량 평활화는 작업자간의 작업 형평을 이룰 수 있고, 라인혼잡을 감소시킬 수 있으며, 생산량을 증가시킬 수 있다. 작업량 평활화의 평가 척도는 Sparing & Miltenburg (1998)가 사용한 작업량 절대편차(absolute deviation of workloads: ADW)를 사용한다. 주어진 작업할당과 투입순서에 대하여 ADW는 식 (1)과 같이 계산된다. 그리고 ADW의 계산에 사용된 기호는 아래와 같다.

$$ADW = \sum_{j=1}^J \sum_{s=1}^S |T_{js} - \bar{T}| \quad (1)$$

J : 작업수.
 J : 작업장수.

S : MPS생산동안 생산되는 제품수.
 t_{im} : 모델 m 에서 작업 i 의 작업시간.
 \bar{T} : 이론적 최소사이클타임, 즉 평균 작업량.
 IF_j : U라인에서 작업장 j 의 앞에 할당된 작업집합.
 IB_j : U라인에서 작업장 j 의 뒤에 할당된 작업집합.
 f_j^s : 작업장 j 의 앞에서 s 번째 사이클의 생산 모델.
 b_j^s : 작업장 j 의 뒤에서 s 번째 사이클의 생산 모델.
 T_{js} : 작업장 j 에서 s 번째 사이클에 할당된 작업들의 작업시간 합.

<표 1>과 <그림 1>의 예를 이용하여 각 사이클마다 할당된 작업량의 변화를 보기로 하자. 예를 들어, MPS가 $d = (d_A, d_B, d_C) = (1, 2, 3)$ 이고, MPS 제약을 만족하는 투입순서가 각각 ABCBCC와 CACBCB로 주어졌다고 하자. <표 2>는 작업할당과 투입순서가 주어졌을 때, 1회의 MPS생산동안 작업장 1에서 각 사이클동안 생산되는 모델과 작업량을 보인 것이다. 이로부터 작업할당과 투입순서는 모두 MMUL의 작업량에 영향을 미침을 알 수 있다.

<표 2> 작업장 1의 ADW

| | 작업할당 1 | | | | 작업할당 2 | | | |
|-------|-------------------------|----------|----------|----------------------|-----------------------|----------|----------|----------------------|
| | $IF_1 = IB_1 = \{1,2\}$ | $\{10\}$ | T_{1s} | $ T_{1s} - \bar{T} $ | $IF_1 = IB_1 = \{3\}$ | $\{11\}$ | T_{1s} | $ T_{1s} - \bar{T} $ |
| 사이클 1 | C | A | 9 | 1 | B | A | 9 | 1 |
| 사이클 2 | A | B | 6 | 4 | C | B | 11 | 1 |
| 사이클 3 | B | C | 13 | 3 | B | C | 8 | 2 |
| 사이클 4 | C | B | 4 | 6 | C | B | 11 | 1 |
| 사이클 5 | B | C | 13 | 3 | C | C | 11 | 1 |
| 사이클 6 | C | C | 9 | 1 | A | C | 8 | 2 |
| 합계 | | | 54 | 18 | | | 58 | 8 |

(a) 모델의 투입순서: ABCBCC

| | 작업할당 1 | | | | 작업할당 2 | | | |
|-------|-------------------------|----------|----------|----------------------|-----------------------|----------|----------|----------------------|
| | $IF_1 = IB_1 = \{1,2\}$ | $\{10\}$ | T_{1s} | $ T_{1s} - \bar{T} $ | $IF_1 = IB_1 = \{3\}$ | $\{11\}$ | T_{1s} | $ T_{1s} - \bar{T} $ |
| 사이클 1 | B | C | 13 | 3 | A | C | 8 | 2 |
| 사이클 2 | C | A | 9 | 1 | C | A | 12 | 2 |
| 사이클 3 | A | C | 11 | 1 | B | C | 8 | 2 |
| 사이클 4 | C | B | 4 | 6 | C | B | 11 | 1 |
| 사이클 5 | B | C | 13 | 3 | B | C | 8 | 2 |
| 사이클 6 | C | B | 4 | 6 | C | B | 11 | 1 |
| 합계 | | | 54 | 20 | | | 58 | 10 |

(b) 모델의 투입순서: CACBCB

3. MMUL의 작업할당과 투입순서의 통합을 위한 내공생 진화알고리즘

3.1 내공생 진화알고리즘

본 연구에서는 자연계의 내공생 진화과정을 모방한 Kim *et al.*(2001)의 알고리즘을 기반으로 하여 MMUL의 작업할당과 투입순서를 동시에 해결할 수 있는 내공생 진화알고리즘(endosymbiotic evolutionary algorithm: EEA)을 제안한다. 내공생진화 이론은 Margulis(1980)가 경쟁이 아닌 공생의 상호작용을 통해 생물이 진화했음을 설명한 이론이며, 그 대표적인 예로 복잡한 세포구조를 갖는 진핵생물

(eukaryotes)의 출현은 비교적 구조가 단순한 원핵생물(prokaryotes)들이 부피가 더 큰 또 다른 원핵생물의 몸속에서 공생하면서 진화된 형태라고 본다. 그리고 진핵생물이 또다시 공생의 과정을 통해 고등동물 또는 고등식물로 진화한 것으로 설명하고 있다. EEA의 기본 아이디어는 내공생 진화과정의 모방한 진화전략을 기존의 공생 진화알고리즘에 포함시킨 것이다.

MMUL의 작업할당과 투입순서문제를 EEA로 해결하기 위하여 작업할당 모집단(Pop-B), 투입순서 모집단(Pop-S), 그리고 통합모집단(Pop-BS)의 3개 모집단을 구성하고 유지한다. Pop-B와 Pop-S는 공생자(symbiont)로 구성되어 있고, 이들은 각각 작업할당정보와 투입순서정보를 표현한다. Pop-B와 Pop-S의 개체들은 통합문제의 부분해이다. Pop-BS는 두 공생자가 결합된 내공생자(endosymbiont)들로 구성되어 있다. Pop-BS에 있는 개체들은 통합문제에 대한 전체해이므로, 작업할당과 투입순서의 조합을 표현한다. 그리고 EEA는 모집단내 및 모집단간의 이웃진화전략을 사용한다.

지역적 내공생 진화를 위하여, 모집단과 이웃은 다음과 같이 구성한다. 3개의 모집단은 모두 토러스 형태의 2차원 격자구조로 구성하고, 이웃진화를 위하여 3x3의 이웃구조를 사용한다. 또한, NB_{ij} , NS_{ij} , 그리고 NBS_{ij} 를 각각 3개의 모집단 Pop-B, Pop-S 그리고 Pop-BS에서 개체 (i, j) 와 그 주위의 8개 개체들을 포함하는 이웃이라 정의한다. Pop-B, Pop-S 그리고 Pop-BS의 각 셀에는 부분해와 전체해로 표현된 각 개체들을 임의로 생성하여 초기화한다.

3개의 모집단은 다음과 같이 2가지 형태로 상호작용한다. (1) $NB_{ij}(NS_{ij})$ 에 있는 개체들은 $NS_{ij}(NB_{ij})$ 에 있는 임의의 개체와 공생한다. 그리고 가장 높은 적응도를 갖는 개체는 서로 결합하여 잠재 내공생자를 형성한다. (2) 잠재 내공생자는 NBS_{ij} 에 존재하는 내공생자와 경쟁한다. 만약, 잠재 내공생자의 적응도가 NBS_{ij} 에 존재하는 가장 낮은 적응도를 갖는 내공생자는 잠재 내공생자에 의해 대체된다. 이때 잠재 내공생자를 형성한 두 공생자는 복사되어 Pop-B와 Pop-S에 그대로 생존해 있다. 조건을 만족하지 않으면, 잠재 내공생자와 기존 내공생자간의 대체는 이루어지지 않고, 잠재 내공생자를 구성했던 공생자들은 원래 위치에 그대로 생존한다. 이상의 상호작용들은 자연현상의 모방 측면에서 자연스러우며, 해의 질을 높일 것으로도 기대된다.

EEA의 특징은 다음과 같다. 첫째, Pop-BS의 개체들은 공생파트너(symbiotic partner)와 결합형태로 진화하므로, 진화과정동안 개체들의 공생파트너에 대한 안정적인 적응을 도모할 수 있다. 여기서 공생파트너란, 적응도 평가에 참여하는 다른 모집단의 개체를 말하며, EEA의 Pop-B와 Pop-S에 있는 개체의 적응도를 평가하기 위해서는 공생파트너를 선택하는 과정이 요구된다. 기존 공생 진화알고리즘에서 모든 개체들의 공생파트너는 진화과정동안 서로 분리되어 있고, 개체들은 공생파트너와 독립적으로 진화한다. 이로부터

공생파트너의 잦은 변경을 초래하게 되고, 이로 인하여 어떤 주어진 탐색방향을 충분히 탐색하지 못할 수 있다. 둘째, EEA는 부분해로 구성된 Pop-B, Pop-S와 전체해로 구성된 Pop-BS가 공존하므로, 부분해에 의한 병렬탐색과 전체해에 대한 통합 탐색이 동시에 이루어질 수 있다. 셋째, EEA는 이웃진화전략을 사용함으로써, 모집단의 다양성과 탐색의 효율성을 높일 수 있다. 다양한 모집단을 갖는 유전탐색은 재조합을 통한 새로운 유전구조를 손쉽게 생성해낼 수 있고, 부분최적(local optima)에 빠지는 것을 방지할 수 있다.

3.2. 내공생 진화알고리즘의 절차

EEA의 절차는 다음과 같다. 아래 절차에서 b_c 와 s_i 는 각각 Pop-B와 Pop-S의 개체(공생자)를 의미하고, β, α 는 Pop-BS의 개체(내공생자)를 의미한다. 그리고 $f(b,s)$ 는 개체 b 가 공생파트너 s 와 결합했을 때의 적응도, $f(\beta\sigma)$ 는 내공생자 $\beta\sigma$ 의 적응도를 표현한 것이다.

단계 1: (초기화)

단계 1.1: Pop-B와 Pop-S의 각 셀에 임의의 생성한 개체들을 채운다. f_{best} 를 $-\infty$ 로 둔다.

단계 1.2: Pop-BS의 각 셀에 임의의 개체들을 생성한다.

단계 2: (이웃정의)

임의의 위치 (i, j) 를 선택하고, 이웃 NB_{ij} , NS_{ij} , 그리고 NBS_{ij} 를 정의한다.

단계 3: (NB_{ij} 와 NS_{ij} 의 협조/잠재 내공생자 생성)

단계 3.1: NB_{ij} 와 NS_{ij} 에 있는 개체의 적응도를 평가한다. 각 개체들의 공생파트너는 각각 NS_{ij} 와 NB_{ij} 에서 선택한다.

단계 3.2: 단계 3.1에서 평가한 개체결합중에서 가장 높은 적응도를 갖는 결합을 b_{ps} 라 하고, 이를 잠재 내공생자로 둔다. 만약, $f(b_{ps})$ 가 f_{best} 보다 높으면 f_{best} 를 갱신한다.

단계 4: (기존 내공생자와 잠재 내공생자의 경쟁)

단계 4.1: NBS_{ij} 에 있는 개체들의 적응도를 평가한다. 가장 높은 적응도를 갖는 개체의 적응도가 f_{best} 보다 높으면, f_{best} 를 갱신한다.

단계 4.2: 만약, $f(b_{ps})$ 가 NBS_{ij} 에서 가장 낮은 적응도 $f(\beta, \alpha)$ 보다 높으면 β, α 와 b_{ps} 를 대체한다. 공생자 b_p 와 s_p 는 각각 Pop-B와 Pop-S에 그대로 생존하고, 대체되는 내공생자 β, α 는 사멸한다.

단계 5: (모집단들의 진화)

단계 5.1: $genetic_evolution(NB_{ij}, R_c, I_m, G_m)$ 를 호출한다.

단계 5.2: $genetic_evolution(NS_{ij}, R_c, I_m, G_m)$ 를 호출한다.

단계 5.3: $genetic_evolution(NBS_{ij}, R_c, I_m, G_m)$ 를 호출한다.

단계 6: (진화의 종료조건)

종료조건을 만족하면 끝내고, 그렇지 않으면 단계 2로 간다.

Sub-routine: *genetic_evolution*(P, R_c, I_m, G_m)

부분단계1: P로부터 재생산에 참여할 개체들을 선별한다. 선별된 각 개체들 중에서 R_c (교차율)를 만족하는 개체들에 대하여 교차연산을 수행한다.

부분단계2: P로부터 낮은 적응도를 갖는 개체가 선별될 확률을 높게 하여 대체할 개체들을 선택한다. 그리고 선택된 개체들은 자손개체와 대체된다.

부분단계3: P로부터 I_m (개체단위 돌연변이율)의 확률로 개체들을 선택하고 G_m (인자단위 돌연변이율)을 기준으로 돌연변이한다.

Pop-B, Pop-S, Pop-BS는 단계 5에서 이웃단위로 안정상태(steady-state) 유전알고리즘의 형태를 따르며 진화한다. 이 알고리즘은 세대별 유전알고리즘보다 탐색능력이 우수하다고 알려져 있다(Syswerda, 1991). EEA를 이용하여 특정한 문제를 해결하기 위해서는 공생파트너, 적응도 평가, 해의 표현, 유전 연산자, 그리고 그밖의 유전 파라미터들이 요구된다. Pop-B, Pop-S, 그리고 Pop-BS의 유전표현과 유전연산자는 각각 제4, 5, 6장에서 순차적으로 설명한다. 개체의 적응도는 식 (1)에 의해 평가한다. 그리고 ADW의 값이 적을수록 높은 적응도를 갖도록 정규화하였다. 이밖에 공생파트너의 선택, 모집단과 이웃의 크기, 교차율과 돌연변이율, 종료조건 등은 제7장에서 설명한다.

4. MMUL의 작업할당을 위한 표현과 연산자

4.1 유전표현과 초기 모집단

작업할당문제를 위한 개체표현에서는 그룹번호 표현을 사용한다. 한 개체의 길이는 I (작업수)이며, 각 인자는 $[1, 2J-1]$ 사이의 정수값을 갖는다. 만약, 개체의 i 번째 인자값 j 가 J 보다 작거나 같으면, 이것은 작업 i 가 U라인에서 앞쪽에 위치한 작업장 j 에 할당됨을 의미한다. 그러나 i 번째 인자값이 J 보다 큰 $2J-j$ 이면, 작업 i 는 뒤쪽에 위치한 작업장 j 에 할당된다. 이와 같은 표현은 별도의 해석방법이 필요없고, 각 인자가 갖는 정보를 명확히 나타내 준다는 장점을 갖는다. 반면에, 가능해를 생산하는 유전연산자의 개발이 요구된다. 이에 대해서는 다음 절에서 설명하기로 한다.

초기 가능해는 모든 작업이 할당될 때까지 아래의 과정을 반복하여 생성한다. 우선, 작업장 j ($=1$)에 대하여 할당을 고려하고, 할당가능 작업집합을 구한다. FP 는 모든 선행작업, FS 는 모든 후행작업이 할당된 작업들로 이루어진 작업집합이라 하자. U라인의 할당가능 작업집합은 FP 와 FS 의 합집합이다. 할당가능 작업집합으로부터 임의의 작업 i 를 선택한다. 만약, $i \in FP$ 이면, 작업 i 에 인자값 j 를 할당하고, 그렇지 않으면, 인자값 $2J-j$ 를 할당한다. 그리고 다음 작업을 할당하기 위하여 할당가능 작업집합을 갱신한다. \hat{t}_i 는 MPS생산동안 작업 i 의 총 작업시간, \bar{T} 는 MPS생산동안 각 작업장의 평균작업량으로 정의하자. 만약, MPS

생산동안 작업장에 할당된 총 작업량이 \hat{T} 보다 크면, 다음 작업장 $j+1$ 을 고려한다. 그리고 할당할 후보 작업은 현재 고려중인 작업장이나 이전 작업장중, 하나에 임의로 할당한다. $(J-1)$ 번째 작업장까지 작업을 할당한 후, 남아있는 모든 작업들은 작업장 J 에 할당한다.

4.2 유전연산자

구조교차(von Laszewski, 1991)를 작업할당문제에 적합하게 변형하였다. 절차는 다음과 같다. 그리고 자손 $O2$ 는 $P1$ 과 $P2$ 의 역할을 바꿔 생산한다.

- 단계1: 범위 $[1, J]$ 에서 임의의 정수 r 을 선택한다.
- 단계2: 부모 $P1$ 에서 작업장 1부터 r 까지 표현한 $[1, r]$ 과 $[2J-r, 2J-1]$ 의 인자를 자손 $O1$ 의 동일한 위치에 상속한다.
- 단계3: 자손 $O1$ 의 남아있는 위치에 부모 $P2$ 에 있는 $[r+1, 2J-r-1]$ 의 인자를 상속한다.
- 단계4: 자손 $O1$ 의 비어있는 위치는 재할당방법을 이용하여 할당한다.

작업할당문제에서는 개체 돌연변이율에 의해 개체들을 임의 선택하고, 선택된 개체에 대하여 인자 돌연변이율에 의해 인자를 돌연변이한다. 이들 인자의 돌연변이는 Kim et al.(2000)에 의해 제안된 재할당절차에 의해 이루어진다. 이 연구에서 제안된 재할당기법은 그룹분할문제에서 좋은 결과를 얻는 것으로 알려진 best-fit-decreasing 규칙을 기본으로 하여 작업할당문제에 적합하게 수정한 것이다. 그리고 이 기법에서는 선행제약의 만족은 물론 작업장의 작업량이 평활화되도록 한다. 여기서 작업장의 작업량 평활화는 각 작업장의 작업량 절대편차의 합을 최소화하는 것이다.

5. MMUL의 투입순서를 위한 표현과 연산자

5.1 유전표현과 초기 모집단

투입순서 모집단의 개체들은 MPS 생산동안 생산되어야 하는 모델들을 투입순서의 나열로 표현한다. 예를 들어, MPS가 $d = (d_A, d_B, d_C) = (1, 2, 3)$ 이라 가정하고, 투입순서가 A, B, C, B, C, C라면 개체는 모델의 투입순서대로 (A B C B C C)와 같이 표현한다. 이 표현은 개체의 인자 위치에 투입순서 정보를 나타낼 수 있다. 초기 모집단은 MPS 제약을 만족하는 개체들을 모집단의 크기만큼 임의로 생성하여 구성된다.

5.2 유전연산자

투입순서문제를 위한 연산자로는 수정순서교차(modified order crossover)와 역순(inversion) 연산자를 사용한다. 혼합모델 투입순서문제에서 이들 두 연산자를 결합하여 사용함으로써 좋은 탐색 성능을 보인 것으로 나타났다(Kim et al., 1996).

수정순서교차는 순서교차를 투입순서문제에 적합하게 변형한 것이다. 두 부모로부터 두개의 절단점을 임의로 선택하고, 부모 $P1$ 에 있는 절단점

사이의 인자들을 자손 O1의 동일한 위치에 상속한다. 부모 P2에서는 부모 P1에 있는 절단점 사이의 인자들을 임의로 삭제하고, P2에 남아 있는 인자들을 P2의 순서를 유지하면서 자손 O1의 아직 채워지지 않은 위치로 상속한다. 그리고 P1과 P2의 역할을 바꿔 자손 O2를 생성한다. 수정순서교차는 부모의 상대적 순서가 자손에 유지되는 연산자이다.

역순은 돌연변이 연산자이다. 부모에서 두 개의 절단점을 임의로 선택하고, 절단점 사이의 인자들을 역순으로 하여 자손을 생산한다.

6. Pop-BS를 위한 표현과 연산자

Pop-BS의 개체는 MMUL의 작업할당과 투입순서의 통합된 형태를 표현한다. Pop-BS에 있는 개체들의 유전표현은 다루는 두 문제의 유전표현을 그대로 사용하고, 이들 두 표현을 결합하기만 하면 된다. 즉, 작업할당을 위한 표현을 개체의 앞부분에 두고, 투입순서를 위한 표현을 개체의 뒷부분에 위치하게 하여 결합하면 된다.

두 부분의 표현이 서로 이질적이므로, Pop-BS를 위한 유전연산은 이질적인 두 부분에 대하여 구분하여 적용한다. 즉, 개체의 앞 부분에 대한 유전연산을 적용한 후, 뒷 부분에 대한 유전연산을 적용한다.

7. 실험설계 및 결과분석

7.1 실험문제 및 비교알고리즘

본 연구에서는 알고리즘의 성능 분석 및 비교를 위한 실험문제로, 19개 작업의 Thomopoulos문제(Thomopoulos, 1967), 61개 작업의 Kim문제(Kim *et al.*, 1997), 그리고 111개 작업의 Arcus문제(Arcus, 1963, Appendix X-3)의 3가지 문제를 기본으로 하여 생성하였다. 다양한 실험을 위하여 <표 3>에서 보는 바와 같이 작업장수와 MPS를 달리하여 여러 실험문제를 만들었다. Arcus문제에서는 작업장수가 많은 문제를 실험하기 위하여 작업 95의 작업시간을 33491에서 66.15로 변경하였다.

본 연구에서 제안한 EEA의 탐색성능은 다음 3가지 진화알고리즘과 비교 분석하였다. 즉, 계층적 유전알고리즘(HGA: hierarchical genetic algorithm), 분리된 모집단기반 공생 진화알고리즘(SPA: separated and population-based symbiotic evolutionary algorithm), 그리고 분리된 이웃기반 공생 진화알고리즘(SNA: separated and neighborhood-based symbiotic evolutionary algorithm)이 비교 알고리즘이다.

HGA에서는 먼저 작업할당문제를 해결한 후, 투입순서문제를 해결하였다. 작업할당문제에서 사용한 목적함수는 작업장에 할당된 작업량 편차의 최소화이고, 투입순서문제의 목적함수는 ADW의 최소화이다. SPA에서는 부분해의 형태로 표현된 각 개체들이 각각 Pop-B와 Pop-S를 유지하고, 두 모집단은 독립적이면서 모집단단위로 진화한다. 기존 대부분의 공생 진화알고리즘(Maher & Poon, 1996; Potter, 1997)은 SPA의 형태를 따른다. SNA는

SPA와 유사하지만, 모집단의 진화가 이웃진화전략을 사용한다는 것에 차이를 갖는다(Kimet *et al.*, 2000).

<표 3> 실험문제

| 문제 | 작업수 | 모델수 | 작업장수 | MPS |
|---------|-----|-----|------|-------------|
| Thom1 | 19 | 3 | 3 | 1 1 1 |
| Thom2 | 19 | 3 | 3 | 3 2 1 |
| Thom3 | 19 | 3 | 4 | 1 1 1 |
| Kim1 | 61 | 4 | 6 | 1 1 1 1 |
| Kim2 | 61 | 4 | 6 | 1 3 4 5 |
| Kim3 | 61 | 4 | 6 | 6 4 2 1 |
| Kim4 | 61 | 4 | 12 | 1 1 1 1 |
| Kim5 | 61 | 4 | 12 | 1 3 4 5 |
| Kim6 | 61 | 4 | 12 | 6 4 2 1 |
| Arcus1 | 111 | 5 | 12 | 1 1 1 1 1 1 |
| Arcus2 | 111 | 5 | 12 | 5 3 2 1 1 |
| Arcus3 | 111 | 5 | 12 | 1 2 4 5 8 |
| Arcus4 | 111 | 5 | 12 | 1 4 8 3 1 |
| Arcus5 | 111 | 5 | 15 | 1 1 1 1 1 1 |
| Arcus6 | 111 | 5 | 15 | 5 3 2 1 1 |
| Arcus7 | 111 | 5 | 15 | 1 2 4 5 8 |
| Arcus8 | 111 | 5 | 15 | 1 4 8 3 1 |
| Arcus9 | 111 | 5 | 27 | 1 1 1 1 1 1 |
| Arcus10 | 111 | 5 | 27 | 5 3 2 1 1 |
| Arcus11 | 111 | 5 | 27 | 1 2 4 5 8 |
| Arcus12 | 111 | 5 | 27 | 1 4 8 3 1 |

7.2. 실험파라미터 설정

모든 알고리즘은 C++언어로 구현되었으며, 700MHz Pentium CPU를 장착한 IBM-PC에서 수행되었다. 유전 파라미터들은 예비실험을 통하여 결정하였다. 첫째, 모집단의 크기는 100(10×10)의 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, 다루는 두 문제에서 개체 돌연변이율은 모두 0.05로 하였으며, 작업할당문제에서 돌연변이되는 개체의 인자 돌연변이율은 0.1로 두었다. 마지막으로, 알고리즘의 종료조건은 생산된 개체수를 사용하였다. 작업수와 모델수에 따라 해공간이 다르므로, Thomopoulos문제, Kim문제, Arcus문제에 대하여 모든 모집단의 생산 개체수의 합이 각각 10,000개, 30,000개, 50,000개 일때 종료하였다. 단, HGA에서 두 문제의 종료조건은 각각 이 종료조건의 절반씩 사용하였다. SPA, SNA, EEA에서는 공생파트너를 선택해야 한다. 모든 공생 진화알고리즘에서 공생파트너는 상대 모집단 또는 이웃에서 임의로 한 개체를 선택하는 전략을 사용하였다.

7.3. 성능비교

실험결과는 <표 4>와 같다. 첫번째 열은 실험문제, 두번째 열은 계층적 방법에 의해 얻어진 ADW이고, 세번째 열부터 다섯번째 열까지는 3가지 공생 진화알고리즘에 의한 ADW이다. 마지막 열은 EEA를 제외하고 가장 좋은 결과를 보인 SNA에 대한 EEA의 개선율로, $\{(SNA의 ADW - EEA의 ADW)/SNA의 ADW\} \times 100(\%)$ 으로 계산하였다. 모든 실험 결과값은 10회 반복실험의 평균이다.

실험결과, 모든 실험문제에서 모든 공생 진화알고리즘이 계층적 방법보다는 좋은 해를 탐색하였다. 이것은 통합문제를 해결하는 고전적인 방법인 계층적인 방법이 해공간을 효과적으로 탐색하지 못함을 의미한다. 또한, SPA와 SNA의 비교에서 SNA가 좋은 성능을 보임에 따라 이웃진화전략이 알고리즘의 성능 향상에 기여한 것을 알 수 있다. 마지막으로, 모든 실험문제에서 EEA가 가장 좋은 결과를 보였다. EEA는 Pop-B와 Pop-S가 진화하는 동안 높은 적응도를 갖는 좋은 개체결합을 Pop-B에서 내공생자로 유지한다. 내공생자는 진화하는 동안 자신에게 높은 적응도를 갖게해 준 공생파트너와 함께 결합되어 진화되므로, 공진화의 환경이 안정적이어서 좋은 해로의 탐색이 유리한 것으로 보인다. EEA는 다른 진화알고리즘에 비해 우수한 성능을 보이면서도 계산소요시간은 많이 요구하지 않았다. Thomopoulos문제, Kim문제, Arcus문제를 해결하는데 소요된 평균 계산시간은 2, 29, 207초가 소요되었다.

<표 4> EEA와 다른 알고리즘들간의 성능 비교

| 문제 | HGA | 공생 진화알고리즘 | | | 개선율 (%) |
|---------|----------|-----------|----------|----------|---------|
| | | SPA | SNA | EEA | |
| Thom1 | 4.7 | 2.0 | 1.6 | 1.5 | 7.7 |
| Thom2 | 7.3 | 3.5 | 3.0 | 2.6 | 11.4 |
| Thom3 | 5.9 | 2.6 | 1.9 | 1.4 | 27.4 |
| Kim1 | 68.7 | 37.5 | 26.9 | 26.0 | 3.3 |
| Kim2 | 198.3 | 126.5 | 97.2 | 89.6 | 7.9 |
| Kim3 | 182.8 | 107.2 | 86.7 | 81.9 | 5.5 |
| Kim4 | 98.3 | 64.1 | 49.8 | 40.3 | 19.1 |
| Kim5 | 289.3 | 225.6 | 167.5 | 144.0 | 14.0 |
| Kim6 | 289.7 | 198.5 | 147.4 | 126.6 | 14.1 |
| Arcus1 | 23529.3 | 23339.3 | 17209.5 | 14639.6 | 14.9 |
| Arcus2 | 60118.3 | 57704.3 | 39754.6 | 36378.6 | 8.5 |
| Arcus3 | 104423.8 | 73700.3 | 59014.3 | 53782.4 | 8.9 |
| Arcus4 | 119028.4 | 85118.6 | 57673.3 | 50019.6 | 13.3 |
| Arcus5 | 47436.9 | 34035.9 | 26868.4 | 22071.7 | 17.9 |
| Arcus6 | 88266.9 | 65356.1 | 61361.8 | 50133.8 | 18.3 |
| Arcus7 | 150468.3 | 114288.1 | 101517.7 | 77907.7 | 23.3 |
| Arcus8 | 128876.8 | 102692.1 | 83109.3 | 70295.6 | 15.4 |
| Arcus9 | 96019.0 | 81843.8 | 71427.0 | 53568.2 | 25.0 |
| Arcus10 | 268641.8 | 234265.2 | 173147.4 | 127554.0 | 26.3 |
| Arcus11 | 463784.2 | 373476.7 | 319593.2 | 247918.9 | 22.4 |
| Arcus12 | 321596.5 | 293803.7 | 255036.9 | 186631.9 | 26.8 |

8. Conclusions

본 연구에서는 MMUL의 작업할당과 투입순서를 동시에 결정할 수 있는 방법론으로 진화알고리즘의 특별한 형태인 EEA를 제안하였다. EEA는 기존 공생 진화알고리즘에 자연계의 내공생 진화과정을 모방한 진화전략을 포함시킨 알고리즘이다. EEA에서는 진화알고리즘이 높은 탐색성능을 갖기에 요구되는 요소들 중에서 병렬성, 수렴성, 다양성을 향상시키기 위한 진화전략들을 사용하였다.

실험결과, 제안된 EEA는 계층적 접근방법뿐만 아니라 다른 공생 진화알고리즘들보다 좋은 성능을 나타냈다. 진화알고리즘이 갖는 적용의 유연성에 의해, 제안한 알고리즘은 다루려는 문제의 적절한 유전표현이 가능하다면 여러 분야에서 상호 관련된 여러 부분문제로 구성된 문제를 통합적으로 해결하는데 적용 가능할 것이다.

References

- Arcus, A.L. (1963), An analysis of a computer method of sequencing assembly line operations, Ph.D. dissertation, University of California, USA.
- Bard, J.F., Dar-El, E.M., and Shtub, A. (1992), An analytic framework for sequencing mixed model assembly lines, *International Journal of Production Research*, **30**, 35-48.
- Kim, J.Y., Kim, Y., and Kim, Y.K. (2001), An endosymbiotic evolutionary algorithm for optimization, *Applied Intelligence*, **15**, 117-130.
- Kim, Y.K., Hyun, C.J., and Kim, Y. (1996), Sequencing in mixed model assembly lines: a genetic algorithm approach, *Computers & Operations Research*, **23**, 1131-1145.
- Kim, Y.K., Kim, S.J., and Kim, J.Y. (2000), Balancing and sequencing mixed-model U-lines with a co-evolutionary algorithm, *Production Planning and Control*, **11**, 754-764.
- Kim, Y.K., Lee, S.Y., and Kim, Y.J. (1997), A genetic algorithm for improving the workload smoothness in mixed model assembly lines. Working Paper, Department of Industrial Engineering, Chonnam National University, Korea.
- Maher, M.L. and Poon, J. (1996), Modelling design exploration as co-evolution, *Microcomputers in Civil Engineering*, **11**, 195-210.
- Margulis, L. (1980), *Symbiosis in Cell Evolution*, WH Freeman, San Francisco.
- McMullen, P.R. (1998), JIT sequencing for mixed-model assembly lines with setups using tabu search, *Production Planning & Control*, **9**, 504-510.
- Miltenburg, J. (1989), Level schedules for mixed-model assembly lines in just-in-time production systems, *Management Science*, **35**, 192-207.
- Monden, Y. (1993), *Toyota Production System*. Second Edition. Institute of Industrial Engineers, Georgia.
- Moriarty, D.E. and Miikkulainen, R. (1997), Forming neural networks through efficient and adaptive coevolution, *Evolutionary Computation*, **5**, 373-399.
- Potter, M.A. (1997), The design and analysis of a computational model of cooperative coevolution, Ph. D. dissertation, George Mason University, USA.
- Sparling, D. and Miltenburg, J. (1998), The mixed-model U-line balancing problem, *International Journal of Production Research*, **36**, 485-501.
- Syswerda, G. (1991), A study of reproduction in generational and steady-state genetic algorithms, *Foundations of Genetic Algorithms*, 94-101.
- Thomopoulos, N.T. (1967), Line balancing-sequencing for mixed-model assembly, *Management Science*, **14**, 59-75.
- von Laszewski, G. (1991), Intelligent structural operators for k-way group partitioning problem. *Proc. 4th International Conference Genetic Algorithms*, 45-52.