

내공생 진화알고리즘을 이용한 유연조립시스템의 공정계획과 일정계획의 통합

송원섭 · 신경석 · 김여근*

전남대학교 산업공학과

The Integrated Process Planning and Scheduling in Flexible Assembly Systems using an Endosymbiotic Evolutionary Algorithm

Won Seop Song · Kyoung Seok Shin · Yeo Keun Kim

Department of Industrial Engineering, Chonnam National University, Gwangju, 500-757

A flexible assembly system (FAS) is a production system that assembles various parts with many constraints and manufacturing flexibilities. This paper presents a new method for efficiently solving the integrated process planning and scheduling in FAS. The two problems of FAS process planning and scheduling are tightly related with each other. However, in almost all the existing researches on FAS, the two problems have been considered separately. In this research, an endosymbiotic evolutionary algorithm is adopted as 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 experimental results are reported.

Keywords: flexible assembly systems, process planning, scheduling, endosymbiotic evolutionary algorithm

1. 서론

유연조립시스템(flexible assembly system; FAS)은 조립기능을 수행하는 로봇, 조립품의 운반기능과 저장기능을 수행하는 자동자재 처리설비, 그리고 이들 설비를 제어하는 컨트롤시스템으로 구성되어 여러 종류의 제품을 자동으로 조립할 수 있는 시스템이다(Ran and Stecke, 1994). 본 연구에서는 FAS에서의 공정계획(process planning)과 일정계획(scheduling)을 다룬다. 공정계획은 작업의 효율적 완성을 위해 FAS 자원을 작업에 할당하는 문제이다. 일정계획은 각 제품의 공정계획에 기초하여 작업장에서의 조립순서를 결정하는 문제이다. FAS 자원의 효율적 이용을 위해서는 공정계획문제와 일정계획문제가 중요하

게 다루어져야 한다(Sawik, 1999).

FAS에서 공정계획과 일정계획은 밀접한 상호 관련성을 갖는다. 즉, 각 제품의 공정계획은 일정계획에 영향을 주고, 일정계획 또한 공정계획에 영향을 준다. 본 연구에서는 FAS에서 공정계획과 일정계획을 동시에 결정하는 효율적인 알고리즘을 제시한다.

FAS의 공정계획을 다룬 여러 연구(Graves and Redfield, 1988; Kim and Park, 1995; Sawik, 1999)와 일정계획을 다룬 여러 연구(Kim *et al.*, 1993; Lee *et al.*, 1993; Park and Kim, 2000)가 이루어졌다. 한편, 이 두 문제의 통합문제를 다룬 다음의 몇몇 연구들 또한 행하여졌다. Agnetis *et al.*(1995)은 두 문제를 다루는 데 동적 계획법을 사용하였다. 이때 버퍼와 공구 매거진의 용량은 고

이 논문은 2003년도 한국학술진흥재단 지원에 의하여 연구되었음(KRF-2003-002-D00362).

*연락처 : 김여근 교수, 500-757 광주광역시 북구 용봉동 300번지 전남대학교 산업공학과, Fax : 062-530-1789,

E-mail : kimyk@chonnam.ac.kr

려하고 있으나 주조립(main assembly) 작업과 부분조립(sub-assembly) 작업으로 구성된 공정도를 사용하고 있다. 이는 작업의 순서유연성의 이용을 제한한다. Sawik은 버퍼를 고려하지 않는 경우 (Sawik, 2002)와 버퍼를 고려한 경우(Sawik, 2004)의 FAS 두 문제를 위한 수리모형을 제시하였다. 이들 Sawik 연구에서는 조립가능 순서가 주어지는 등 해공간을 인위적으로 축소하고 있다.

본 연구에서는 FAS의 각 작업장에서 나타날 수 있는 아래와 같은 현실적인 제약과 각 제품이 가질 수 있는 여러 조립순서를 고려한다. FAS의 작업장은 흔히 로봇에 의해 조립이 행하여 지므로 조립 가능한 작업과 그렇지 못한 작업이 있다. 그리고 각 조립작업에 사용되는 설비의 설치에 요구되는 공간으로 인한 제약과 작업 선행제약이 고려된다. 또한 제품조립의 순서유연성과 동일 작업이라도 제품에 따라 다른 작업장에 할당될 수 있는 대안할당(alternative assignment)을 고려함으로써 FAS의 설비효율화를 얻고자 한다. 그리고 작업이 행해지는 작업장에 따라 작업시간이 다르고, 일정 크기의 버퍼가 작업장 사이에 위치하고 있다고 본다.

본 연구에서는 FAS의 공정계획과 일정계획을 동시에 통합적으로 다룰 수 있는 접근방법으로 공진화알고리즘(coevolutionary algorithm)의 한 형태인 내공생 진화알고리즘(endosymbiotic evolutionary algorithm; EEA)을 채용하였다. 내공생 진화알고리즘은 Kim *et al.*(2001)에 의해 제안되었다.

2. 문제정의

본 연구에서 다루는 문제의 이해를 돕기 위하여 문제정의에 앞서, FAS와 전통적인 직선조립라인(straight assembly line; SAL)의 차이를 비교하여 보자. 첫째, 제품조립이 FAS에서는 자동조립기계(예로, 산업용 로봇)와 자동자재처리설비(부품공급장치, 자재처리 로봇 등에 의해 이루어지고, SAL에서는 작업자와 수동(또는 반자동) 조립설비에 의해 이루어진다. 이로 인해, 둘째, FAS에서는 흔히 각 작업의 할당 가능 작업장이 몇 개로 제한되어 있는 반면, SAL에서는 모든 작업이 각 작업장에 할당될 수 있다고 본다. 단지 예외적으로, 설비제약이 있는 작업은 특정 작업장에 할당되어야 한다는 제약을 갖는다. 셋째, FAS 작업장에서는 작업을 위한 피딩 메커니즘(feeding mechanism)의 필요에 의해 이를 위한 설치공간을 필요로 한다. 이로 인해 작업장에 할당될 수 있는 작업의 수는 유한하다. 이것을 흔히 FAS의 유연성 용량(flexibility capacity) 또는 작업장 용량(staging capacity)이라 부른다. 본 연구에서는 이를 작업장의 공간제약이라 부르기로 한다. 이러한 FAS의 형태 및 작업장 구조를 간단히 나타내면 <그림 1>과 같다.

본 연구에서는 아래와 같은 가정을 둔다.

- 1) FAS는 고정된 수의 작업장을 가지며, 각 작업장에는 단지 하나의 조립기계가 있다.

- 2) FAS는 조립제품이 단일 방향으로만 흐르는 작업장으로 구성되어 있다. 즉, 지나온 작업장의 재방문(revisiting)은 불가하나 건너뛰기(bypass)의 전진은 가능하다.
- 3) 각 작업은 여러 작업장에서 작업될 수 있다.

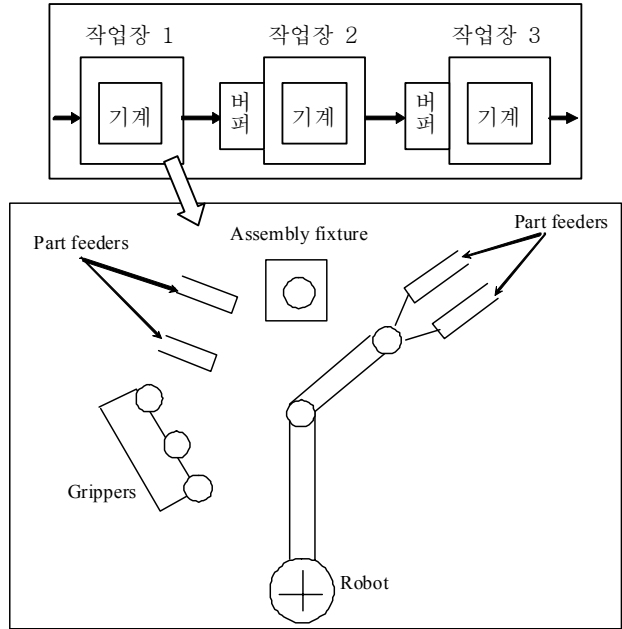


그림 1. FAS 형태 및 작업장 구성요소.

- 4) 모든 제품의 작업 선행관계를 표현한 결합 선행공정도를 사용한다. 이는 <그림 2>와 같이 나타내어지고, 작업들의 선행관계, 작업 가능한 작업장집합, 작업소요시간, 작업에 필요한 공간크기의 정보를 담고 있다.
- 5) 각 제품에 대한 수요는 주어졌다.
- 6) 각 작업의 조립시간은 준비시간, 공구교환시간, 장착/탈착 시간을 포함한다.

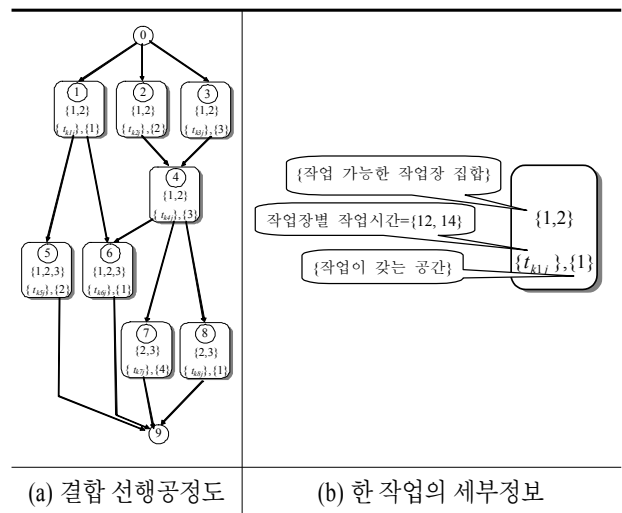


그림 2. 네트워크 표현.

- 7) 동일 작업이라도 수행되는 작업장에 따라 다른 조립시간을 갖는다.
- 8) 버퍼는 유한한 크기로서 첫 작업장을 제외한 각 작업장의 앞에 위치하고 있다.

본 연구에서는 이러한 가정과 제약 아래서, 총 처리시간(makespan) 최소화를 목적으로 공정계획과 일정계획을 동시에 통합적으로 해결한다. 본 연구에서 공정계획은 각 제품의 작업을 작업장에 할당하는 문제이다. 이 때, 동일 작업이라도 제품에 따라 다른 작업장에 할당될 수 있다. 그리고 일정계획은 흐름조립라인이므로 제품의 투입순서를 결정하는 것이다.

FAS의 공정계획과 일정계획에 따라 제품들이 라인에 하나씩 투입되면, 유한한 버퍼 크기로 인해 제품의 흐름은 다음과 같이 이루어진다. 어떤 제품이 작업장 ($j-1$)에서 조립이 완료되어, 다음 작업장 j 에서 작업 가능하기 위해서는 먼저 작업장 j 의 로봇이 사용 가능한지(작업장 j 가 비어 있는지) 살펴보게 된다. 만일 로봇이 사용 가능하면 이동하여 바로 조립이 이루어지며, 그렇지 않다면 작업장 j 의 버퍼에서 대기하게 된다. 하지만 작업장 j 의 모든 버퍼에 제품이 대기하고 있으면, 작업장 ($j-1$)에서 제품을 방출하지 못하고 버퍼 사용이 가능한 시각까지 대기하는 작업장 ($j-1$)의 유ힴ시간(idle time)이 발생하게 된다.

본 연구에서 사용되는 기호를 먼저 정의해 두자.

- I : 작업의 집합, $i \in I = \{1, 2, \dots, m\}$.
- J : 작업장의 집합, $j \in J = \{1, 2, \dots, n\}$.
- K : 제품의 집합, $k \in K = \{1, 2, \dots, p\}$.
- d_k : 제품 k 의 수요.
- t_{kij} : 제품 k 의 작업 i 를 작업장 j 에서 행할 때의 조립시간.
- $\hat{t}_{kij} = d_k \cdot t_{kij}$.
- T_j : 작업장 j 에 할당된 작업의 총 작업시간.
- B_j : 작업장 j 의 공간 크기.

- a_{ij} : 작업 i 를 작업장 j 에서 수행하는 데 요구되는 작업장 공간 크기.
- b_j : 작업장 j 에 이미 할당된 작업들에 의해 요구되는 공간 크기(모든 $a_{ij}=1$ 이면 작업장에 할당된 작업의 수).

3. FAS의 공정계획과 일정계획 통합을 위한 내공생 진화알고리즘

3.1 내공생 진화알고리즘

내공생 진화(endosymbiotic evolution) 이론은 원핵생물(prokaryotes)들이 내공생 과정을 거쳐 진핵생물(eukaryotes)로 진화하였다는 이론으로, 1981년 생물학자인 Lynn Margulis (1981)에 의해 제안되었다. EEA는 이러한 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. 또한 EEA에서는 모집단의 다양성 유지를 위하여 이웃진화전략(Kim *et al.*, 2001)이 사용된다.

EEA는 공정계획 모집단(Pop-P), 일정계획 모집단(Pop-S), 그리고 통합 모집단(Pop-PS)의 3개 모집단을 구성하고 유지한다. Pop-P와 Pop-S는 분리되어 활동하는 공생자(symbiont)로 구성되어 있고, 이들은 각각 공정계획정보와 일정계획정보를 나타낸다. Pop-P와 Pop-S의 개체들은 통합문제의 부분해이다.

Pop-PS는 두 공생자가 결합된 내공생자(endosymbiont)들로 구성되어 있다. 이들은 통합문제에 대한 전체해로서, 공정계획과 일정계획을 결합한 형태로 표현된다. 그리고 EEA는 모집단내와 모집단 간에 있어서 이웃진화전략을 사용한다. 모집단과

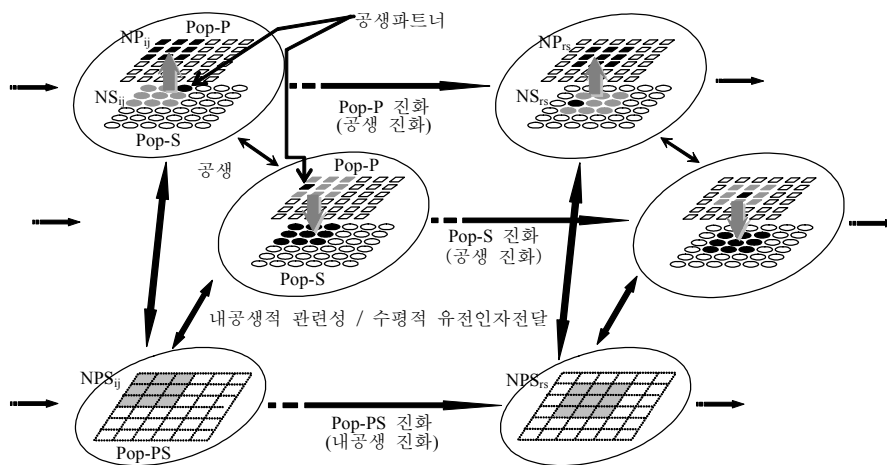


그림 3. 내공생 진화알고리즘.

이웃은 다음과 같이 구성한다. 3개의 모집단은 모두 토러스 형태의 2차원 격자구조로 구성하고, 3×3 의 이웃구조를 사용한다. NP_{ij} , NS_{ij} , 그리고 NPS_{ij} 는 각각 2차원 모집단 Pop-P, Pop-S 그리고 Pop-PS에서 위치 (i, j) 에 있는 개체 (i, j) 와 그 주위의 8개 개체로 구성된 이웃을 나타낸다.

3.2 내공생 진화알고리즘의 절차

단계 1: (초기화)

단계1.1: Pop-P와 Pop-S의 각 셀에 임의의 생성한 개체들을 채운다. 그리고 $f_{best} = -\infty$ 로 둔다.

단계1.2: Pop-PS의 각 셀에 임의의 개체들을 생성한다.

단계 2: (이웃정의)

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

단계 3: (NP_{ij} 와 NS_{ij} 의 잠재 내공생자 생성)

단계3.1: NP_{ij} 와 NS_{ij} 에 있는 개체의 적응도를 평가한다. 각 개체들의 공생 파트너는 각각 NS_{ij} 와 NP_{ij} 에서 선택한다.

단계3.2: 단계 3.1에서 평가한 개체결합 중에서 가장 높은 적응도를 갖는 결합을 $p_{s,u}$ 라 하고, 이를 잠재 내공생자로 둔다.

만약, $f(p_{s,u})$ 가 f_{best} 보다 높으면 f_{best} 를 갱신한다.

단계 4: (기존 내공생자와 잠재 내공생자의 경쟁)

단계4.1: NPS_{ij} 에 있는 개체들의 적응도를 평가한다. 가장 높은 적응도를 갖는 개체의 적응도가 f_{best} 보다 높으면, f_{best} 를 갱신한다.

단계4.2: 만약, $f(p_{s,u})$ 가 NPS_{ij} 에서 가장 낮은 적응도 $f(\rho_i, \sigma_i)$ 보다 높으면 ρ_i, σ_i 와 $p_{s,u}$ 를 대체한다. 공생자 p_u 와 s_u 는 각각 Pop-P와 Pop-S에 그대로 생존하고, 대체되는 내공생자 ρ_i, σ_i 는 사멸한다.

단계 5: (모집단들의 진화)

단계5.1: $genetic_evolution(NP_{ij}, R_c, I_m, G_m)$ 를 호출한다.

단계5.2: $genetic_evolution(NS_{ij}, R_c, I_m, G_m)$ 를 호출한다.

단계5.3: $genetic_evolution(NPS_{ij}, R_c, I_m, G_m)$ 를 호출한다.

단계 6: (진화의 종료조건)

종료조건을 만족하면 끝내고, 그렇지 않으면 단계 2로 간다.

Sub-routine: $genetic_evolution(P_{ij}, R_c, I_m, G_m)$

부분단계1: P_{ij} 로부터 재생산에 참여할 개체를 선택한다. 선택된 각 개체들에 대해 교차율에 따라 교차연산을 수행한다.

부분단계2: P_{ij} 로부터 낮은 적응도를 갖는 개체가 선택될 확률을 높게 하여 대체할 개체들을 선택하고, 이를 부분단계1에서 생성된 자손개체로 대체한다.

부분단계 3: P_{ij} 로부터 I_m (개체단위 돌연변이율)의 확률로 개체들을 선택하고 G_m (인자단위 돌연변이율)의 확률로 인자 돌연변이 연산을 수행한다.

Pop-P, Pop-S, Pop-PS는 단계 5에서 이웃단위로 안정상태 (steady-state) 유전알고리즘의 형태를 따르며 진화한다. 이 알고리즘은 세대별 유전알고리즘보다 탐색능력이 우수하다고 알려져 있다(Syswerda, 1991). EEA를 이용해 특정한 문제를 해결하기 위해서는 공생 파트너, 적응도 평가, 해의 표현, 유전 연산자, 그리고 유전 파라미터들이 요구된다. Pop-P, Pop-S, 그리고 Pop-PS의 유전표현과 유전연산자는 각각 제4, 5, 6장에서 순차적으로 설명한다.

3.3 적응도 함수

개체의 적응도는 총 처리시간 값이 적을수록 높은 값을 가질 수 있도록 하였다. 총 처리시간 값은 가장 마지막 투입제품의 모든 일정계획이 끝난 후, 즉 모든 제품에 대해 일정계획이 끝난 후 마지막 작업장에서의 완료시간이다. 본 연구에서는 다음과 같은 평가 함수 $Eval$ 을 사용한다.

$$Eval = makespan + (C \times WS^a)$$

여기서, $WS = \sum_{j=1}^n \max(0, b_j - B_j)$, a 와 C 는 파라미터이다.

4. FAS 공정계획을 위한 유전요소

4.1 표현 및 초기 모집단

개체는 $(p \times m)$ 행렬로 표현하며 행은 제품을, 열은 작업을 나타낸다. <그림 4>는 이를 보여주는데, 인자 값, c_{ki} 는 제품 k 의 작업 i 가 행하여지는 작업장번호(단, 제품 k 에서 작업 i 가 수행되지 않으면 $c_{ki}=0$)를 나타낸다. 이러한 표현은 일종의 그룹번호 표현으로 작업장 정보(작업장에 할당된 작업)가 자연스럽게 명확히 표현되어 개체의 해석과 정보의 이용이 용이하다는 장점을 갖는다.

초기 모집단은 작업의 선행제약과 조립 가능 작업장 제약을 만족하는 임의의 개체로 구성한다. 이때 작업장 공간제약을 가능한 한 만족하도록 동일 작업은 같은 작업장에 할당한다. 조립 가능 작업장 중에서 공간제약을 만족하는 작업장이 없는 경우, 조립 가능 작업장 중에서 가장 앞에 위치한 작업장에 할당한다.

$k \backslash i$	1	2	3	m
1	c_{11}	c_{12}	c_{13}	c_{1m}
2	c_{21}	c_{22}	c_{23}	c_{2m}
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
p	c_{p1}	c_{p2}	c_{p3}	c_{pm}

그림 4. 개체표현.

4.2 유전연산자

유전연산자에는 교차(crossover)와 돌연변이 (mutation)가 있다. 본 연구에서 교차는 변형된 구조교차(structural crossover)를 사용한다. 사용된 구조교차는 다음과 같다. 범위 $[1, n]$ 의 정수 중에서 임의로 r 을 선택한다. 한 부모(P1)로부터 인자 값이 0부터 r 까지의 인자를 자손(O1)에게 상속한다. 다른 부모(P2)에서 1부터 r 까지의 값을 갖는 인자들을 모두 지운다. 자손(O1)에서 비어 있는 인자들을 다른 부모(P2)에서 복사한다. 자손(O1)에서 인자 값이 결정되지 않은 인자는 발견적 기법에 의해 보수한다. 다른 자손(O2)은 두 부모의 역할을 바꾸어 생산한다. <그림 5>는 $r = 2$ 일 때, 교차의 예를 보이고 있다.

	$k \backslash i$	1	2	3	4	5	6	7	8
P1	1	1	1	0	1	3	0	3	3
	2	1	1	3	2	2	2	0	3
	3	1	0	3	1	2	3	3	0
		↓	↓	↓	↓	↓	↓		
O1	1	1	1	0	1	*	0	3	3
	2	1	1	*	2	2	2	0	3
	3	1	0	*	1	2	3	3	0
							↑	↑	↑
P2	1	1	1	0	1	2	0	3	3
	2	1	1	2	1	3	3	0	3
	3	1	0	2	1	3	3	3	0

그림 5. 교차.

자손 (O1)에서 밑줄이 있는 인자 값은 부모1(P1)로부터, 윗줄이 있는 것은 부모2(P2)로부터 상속받은 것이다. 그리고 * 는 부모로부터 인자 값을 상속받지 못하고 비어 있는 인자를 의미한다. 이 인자들은 다음에 제시하는 발견적 기법인 재할당 방법에 의해 인자 값을 부여받는다. 돌연변이는 개체 돌연변이율에 의해 개체들을 선택하고, 선택된 개체에 대하여 인자 돌연변이율에 의해 인자를 선택한다. 이들 인자들은 본 연구에서 제안하는 재할당방법에 의해 작업장 번호(인자 값)가 부여된다. 이 재할당방법은 문제가 갖는 특정 정보를 이용한다. 재할당 과정과 이에 필요한 기호는 다음과 같다.

- $J(i)$: 작업 i 가 할당 가능한 작업장 집합.
- $I(j)$: 작업장 j 에 할당된 작업의 집합.
- $IP(i)$: 작업 i 의 직선행 작업의 집합.
- $S(i)$: 작업 i 의 후행 작업의 집합.
- $R(k)$: 제품 $k, k=1,2,\dots,p$ 에서 작업장이 미할당된 작업의 집합.
- $AR(k)$: $IP(i), i \in R(k)$ 의 원소가 모두 개체에 할당된 작업집합. 즉, $AR(k) = \{i | \{R(k) \cap IP(i)\} = \phi, i \in R(k)\}$.
- E_i : 개체에서 $IP(i)$ 의 작업들에 해당하는 인자 값 중 가장 큰 값. $IP(i) = \phi$ 이면, $E_i = 1$ 이다.

L_i : 개체에서 $S(i)$ 의 작업들에 해당하는 인자 값 중 0이 아닌 가장 작은 값. $S(i) = R(k) = \phi$ 이면, $L_i = n$ 이다.

$AS(i) = \{j | E_i \leq j \leq L_i, j \in J(i)\}$, $J(i)$ 에서 E_i 보다 이른 작업장과 L_i 보다 늦은 작업장을 제거한 작업장 집합.

$$\bar{a}_{ij} = \begin{cases} a_{ij}, & \text{if } i \in I(j) \\ 0, & \text{otherwise} \end{cases}$$

- 단계 1: T_j 와 $b_j, j=1, 2, \dots, n$ 을 계산하고, $k = 1$ 로 둔다.
- 단계 2: $R(k)$ 와 $AR(k)$ 를 구한다. $R(k) = \phi$ 이면, 단계8로 간다.
- 단계 3: $AR(k)$ 에서 작업소요공간이 가장 큰 작업을 i^* 로 선택한다. 여러 작업이 존재하면 작업시간이 가장 큰 작업을 선택한다.
- 단계 4: E_{i^*} 와 L_{i^*} 를 구하고, 이로부터 $AS(i^*)$ 를 구한다.
- 단계 5: $RS = \{j | b_j + \bar{a}_{i^*j} \leq B_j, j \in AS(i^*)\}$ 로 둔다.
 - 단계 5.1: $RS = \phi$ 이면, $j^* = \arg \min_{j \in AS(i^*)} T_j + \hat{t}_{ki^*j}$ 로 두고, 단계 6으로 간다.
 - 단계 5.2: 그렇지 않으면, $j^* = \arg \min_{j \in AS(i^*)} (b_j + \bar{a}_{i^*j})$ 로 둔다.
- 단계 6: 제품 k 의 작업 i^* 를 작업장 j^* 에 할당하고, 집합 $R(k)$ 에서 i^* 를 제거한다.
- 단계 7: $R(k) \neq \phi$ 이면, $AR(k), T_{j^*}$ 와 b_{j^*} 를 갱신한 후 단계 3으로 간다.
- 단계 8: $k = k + 1$ 로 둔다.
- 단계 9: $k > p$ 이면 종료하고, 그렇지 않으면 $T_{j^*}, b_{j^*}, j=1, 2, \dots, n$ 를 갱신하고 단계 2로 간다.

위 재할당절차에서 작업 소요공간과 그에 따른 작업장 공간 정보는 단계 3과 단계 5.2에서, 작업의 할당 가능한 작업장 정보는 단계 4에서, 작업장 부하정보는 단계 5.1에서 각각 고려하고 있다.

5. FAS 일정계획을 위한 유전요소

5.1 유전표현

일정계획 모집단의 개체들은 제품혼합비율을 나타내는 최소제품집합(minimum part set; MPS)에 있는 제품들의 투입순서의 나열로 표현한다. 예를 들어, MPS가 $d = (d_A, d_B, d_C) = (1, 2, 3)$ 이라 가정하고, 투입순서가 A, B, C, B, C 라면 개체는 제품의 투입순서대로 (A B C B C)와 같이 표현한다.

5.2 유전연산자

본 연구에서는 투입순서문제를 위한 유전연산자로 ISRX (immediate successor relation crossover)와 역순(inversion) 연산자를 사용한다. 혼합모델 생산라인의 투입순서문제에서 이들 두 연

산자를 결합하여 사용함으로써 좋은 탐색성능을 보인 것으로 나타났다(Kim *et al.*, 1996). ISRX는 인접인자 재결합교차(edge recombination crossover)(Whitley, 1989)를 변형한 것으로 부모 개체들의 직후행관계 정보를 이용하여 자손을 생산하는 교차 연산자이다. 인자 A의 직후행인자는 A 바로 다음에 위치한 인자이다. 마지막 인자의 직후행인자는 첫 번째 인자가 된다. 두 부모 개체가 각각 P1=(A A B B B C C C C)와 P2=(A B C A B C B C C)일 때, 초기 ISRX 표는 <표 1>의 (a)와 같다. <표 1>의 (a)에는 각 인자형에 대한 모든 직후행인자들이 나열되어 있다. 자손의 첫 번째 인자형은 임의로 선택한다. 그리고 직후행인자열에서 선택된 인자형을 임의로 두 개 삭제하고 인자비율을 수정한다. 그리고 선택된 인자형의 행에서 가장 많이 남아 있는 인자형을 선택한다. 동일한 경우 인자비율이 작은 것을 선택하고, 이것 또한 동일한 경우 임의로 선택한다. 자손이 완성될 때까지 이를 반복한다. <표 1>은 초기에 임의로 인자형 B를 선택할 때, 처음 4회 반복 시행한 절차를 보인 것이다. 구체적인 절차는 Kim *et al.*(1996)을 참조할 수 있다.

역순 돌연변이는 부모에서 두 개의 절단점을 선택하고, 절단점 사이의 인자들을 역순으로 하여 자손을 생산한다. 이러한 역순 돌연변이의 예는 <그림 6>과 같다. 그림에서 P와 O는 각각 부모와 자손 개체이다.

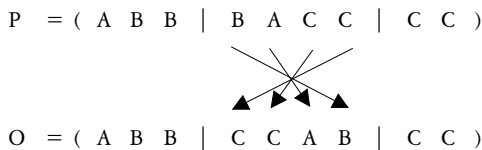


그림 6. 역순 연산자.

6. Pop-PS의 유전표현과 유전연산

작업할당-투입순서 모집단(Pop-PS)에서 개체는 내공생자로 구성된다. 내공생자는 Pop-P에 있는 개체의 계능과 Pop-S에 있는 개체의 계능이 결합된 형태로 이루어진다. 즉, 내공생자는 FAS의 공정계획과 일정계획을 통합한 형태이다. 이러한 유전표현은 4장과 5장에서 사용된 유전표현을 그대로 사용한다. 내

공생자의 유전연산(교차와 돌연변이)도 4장과 5장에서 사용한 유전연산방법을 각 계능의 해당 부분에 그대로 적용하여 연산한다.

7. 실험과 분석

7.1 비교 알고리즘

본 연구에서는 제안한 알고리즘(EEA)을 유희시간 최소화 알고리즘(Route Idle Time Minimization Algorithm; RITM)과 계층적 유전알고리즘(Hierarchical Genetic Algorithm; HGA)을 통해 실험하고 결과를 비교분석한다.

RITM은 Sawik(1999)이 제안한 FAS의 일정계획을 위한 휴리스틱 기법으로, 가중유희시간(weighted idle time)을 계산하여, 이 시간이 작은 제품부터 시스템에 투입한다. 이 기법은 마지막 작업장에서 작업시간이 큰 제품이 먼저 투입되고, 다음은 모든 작업장에서 작업시간이 거의 같은 제품들이 투입되도록 하고, 마지막으로 앞 작업장에서 작업시간이 큰 제품이 투입되도록 가중치를 주어 유희시간을 조정한다. 자세한 절차는 Sawik (1999)을 참고할 수 있다. 여기서 요구되는 제품의 작업할당은 4장에서 설명한 유전요소를 사용한 진화알고리즘에 의해 구하였다.

HGA는 4장에서 소개한 유전요소들에 의해 최대작업량을 갖는 작업장의 작업부하의 최소화를 목적으로 작업을 작업장에 할당하고, 이 결과로부터 얻은 가장 좋은 작업할당정보를 이용하여 총 처리시간의 최소화를 목적으로 5장에서 소개한 유전요소들에 의해 제품의 투입순서문제를 해결하였다.

7.2 실험설계

실험문제는 19개 작업의 Tomopoulos 문제(Thomopoulos, 1967), 61개 작업의 Kim 문제(김여근 외 2인, 1997), 111개 작업의 Arcus 문제(Arcus, 1963)를 기본으로 하였다. 이들 문제의 결합 선행공정도를 이용하고, <표 2>와 같이 제품의 모델 수, 작업장 수, 작업장 공간크기에 따라 문제를 구성하였다. 실험에 필요한 작업시간, 작업공간, 할당 가능 작업장 등의 자료는

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	BCCCC	3	BCCCC	2	BCCC	2	BCCC	2
C	CCCAABCA	4	CCCAABCA	4	CCAABCA	3	CCBCA	3
선택된 인자형	B(임의선택)		C		A		B	

‘http://syslab.choanma.ac.kr/links/data.htm’을 참고할 수 있다. 제안한 알고리즘은 C++프로그램 언어로 구현되었으며, 2.66GHz Pentium CPU를 장착한 IBM-PC에서 수행되었다. 파라미터는 예비실험을 통하여 결정하였다.

표 2. 실험문제

	문제 번호	작업 수	제품 모델 수	작업장 수	작업장 공간
Tomopoulos	Prob01	19	3	3	20
	Prob02	19	3	4	18
Kim	Prob03	61	4	12	35
	Prob04	61	4	18	30
	Prob05	61	6	12	35
	Prob06	61	6	18	30
Arcus	Prob07	111	5	15	121
	Prob08	111	5	24	94
	Prob09	111	5	30	77
	Prob10	111	7	15	125
	Prob11	111	7	24	105
	Prob12	111	7	30	85
	Prob13	111	10	15	127
	Prob14	111	10	24	125
	Prob15	111	10	30	100

7.3 성능 비교

이 절에서는 앞에서 언급한 RITM, HGA, EEA의 성능을 비교 분석한다. <표 3>에서 ‘Best’, ‘Mean’, ‘Std’는 10회 반복실험에서 가장 좋은 값, 평균 값, 표준편차를 각각 나타낸다. <표 3>

표 3. 알고리즘 성능 비교

Problem	RITM			HGA			EEA			Improved rate(%)
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	
Prob01	782.0	835.8	33.3	420.0	462.0	40.8	416.0	429.2	9.1	7.1
Prob02	619.0	661.2	26.7	337.0	365.8	21.4	337.0	348.6	8.5	4.7
Prob03	5525.0	5964.6	382.6	3583.0	3864.8	237.3	3029.0	3187.6	118.4	17.5
Prob04	4535.0	4903.2	189.1	3151.0	3368.0	188.5	2537.0	2665.4	81.8	20.9
Prob05	9287.0	9891.6	474.7	4919.0	5498.2	369.7	4613.0	4764.6	124.2	13.3
Prob06	7099.0	7574.0	412.7	4409.0	4761.8	190.4	3363.0	3713.6	182.0	22.0
Prob07	5587.7	5748.0	186.0	3407.3	3597.6	100.1	2732.0	2856.9	98.7	20.6
Prob08	4377.7	4709.5	200.1	2847.8	3011.3	122.8	2146.5	2300.4	88.0	23.6
Prob09	4053.3	4496.9	256.3	2777.1	2967.1	99.2	2033.1	2150.2	64.4	27.5
Prob10	8866.3	9369.2	403.0	4852.1	5115.6	197.3	4108.5	4253.8	83.3	16.8
Prob11	6835.5	7125.1	249.2	4019.8	4191.9	154.6	3101.3	3221.2	82.9	23.2
Prob12	6413.4	6933.5	439.8	3636.7	4013.3	198.5	2883.8	2970.5	84.2	26.3
Prob13	16399.7	16966.8	736.6	7923.3	8542.1	297.0	7089.5	7292.2	146.8	14.6
Prob14	12022.2	12962.6	674.5	6120.7	6622.4	201.3	5152.8	5442.2	179.7	17.8
Prob15	11677.0	12038.6	285.7	5859.3	6131.1	217.1	4624.3	4827.7	111.6	21.3

의 Prob07~Prob15 문제에서 제시한 값은 결과 값을 1000으로 나눈 것이다. 마지막 열은 HGA에 대한 EEA의 개선율이다. 이는 $\{(HGA의\ Mean - EEA의\ Mean)/HGA의\ Mean\} \times 100(\%)$ 에 의해 계산되었다.

<표 3>에서, EEA는 다른 두 알고리즘보다 모든 문제에서 더 좋은 해를 제공한다. 그리고 HGA가 그 다음으로 좋은 결과를 보였으며, RITM이 가장 낮은 성능을 보였다. 또한 EEA는 비교알고리즘보다 모든 실험문제에서 낮은 표준편차를 갖는다. 이는 제안한 EEA가 안정적으로 좋은 해를 유도할 수 있음을 의미한다. 이것은 통합문제를 해결하는 고전적인 방법 중 계층적 방법인 RITM 및 HGA가 해공간을 효과적으로 탐색하지 못함을 의미한다.

EEA는 Pop-P와 Pop-S가 진화하는 동안 높은 적응도를 갖는 좋은 개체결합을 Pop-PS에서 내공생자로 유지한다. 내공생자는 진화하는 동안 자신에게 높은 적응도를 갖게 해준 공생 파트너와 함께 결합되어 진화하므로, 공진화의 환경이 안정적이어서 좋은 해로의 탐색이 유리한 것으로 보인다. 평균 계산시간은 Thomopoulos 문제, Kim 문제, Arcus 문제에서 각각 5초, 92초, 304초가 소요되었다.

8. 결론

본 연구에서는 유연조립시스템의 공정계획과 일정계획을 동시에 결정할 수 있는 방법론으로 진화알고리즘의 한 형태인 내공생 진화알고리즘을 제안하였다. 제안된 알고리즘은 기존 공생 진화알고리즘에 자연계의 내공생 진화과정을 모방한 진화전략을 포함시킨 알고리즘이다. 이 알고리즘은 진화알고리즘에서 높은 탐색성능을 위해 요구되는 해탐색의 병렬성, 좋은

해로의 수렴성, 모집단의 다양성을 위한 진화전략들을 포함하고 있다.

실험결과, 제안된 내공생 진화알고리즘은 계층적 접근방법 뿐만 아니라 다른 발견적 기법보다 좋은 성능을 보였다. 따라서 제안한 접근방법은 유연조립시스템의 두 계획문제를 통합적으로 해결하는 하나의 유망한 방법론으로 사용될 수 있다.

참고문헌

김여근, 이수연, 김용주 (1997), 혼합모델조립라인에서 작업부하의 평활화를 위한 유전알고리즘, *대한산업공학회지*, 23(3), 515-532.
 Agnetis, A., Nicolo, F., Arbib, C. and Lucertini, M. (1995), Task assignment and subassembly scheduling in flexible assembly lines, *IEEE Transactions on Robotics and Automation*, 11, 1-20.
 Arcus, A.L. (1963), An analysis of a computer method of sequencing assembly line operations, Ph.D. dissertation, University of California.
 Graves, S.C. and Redfield, C.H. (1988), Equipment selection and task assignment for multiproduct assembly system design, *The International Journal of Flexible Manufacturing Systems*, 6, 361-382.
 Kim, D.S., Vincent, A.M. and Peter A.P. (1993), Integrative cycle scheduling approach for a capacitated flexible assembly system, *Decision Sciences*, 24, 126-147.
 Kim, H. and Park, S. (1995), A strong cutting plane algorithm for the robotic

assembly line balancing problem, *International Journal of Production Research*, 33, 2311-2323.
 Kim, J.Y., Y. Kim and Y.K. Kim (2001), An endosymbiotic evolutionary algorithm for optimization, *Applied Intelligence*, 15, 117-130.
 Kim, Y.K., C.J. Hyun and Y. Kim (1996), Sequencing in mixed model assembly lines: a genetic algorithm approach, *Computers & Operations Research*, 23, 1131-1145.
 Lee, C.Y., Cheng, T.C.E. and Lin, B.M.T. (1993), Minimizing the makespan in the 3-machine assembly type flowshop scheduling problem, *Management Science*, 39, 616-625.
 Margulis, L. (1981), *Symbiosis in cell evolution*, W.H. Freeman, San Francisco.
 Park, M.W. and Kim, Y.D. (2000), A branch and bound algorithm for a production scheduling problem in an assembly system under due date constraints, *European Journal of Operational Research*, 123, 504-518.
 Sawik, T. (1999), *Production planning and scheduling in flexible assembly systems*, Springer-Verlag, Berlin.
 Sawik, T. (2002), Monolithic vs. hierarchical balancing and scheduling of a flexible assembly line, *European Journal of Operational Research*, 143, 115-124.
 Sawik, T. (2004), Loading and scheduling of a flexible assembly system by mixed integer programming, *European Journal of Operational Research*, 154, 1-19.
 Thomopoulos, N.T. (1967), Line balancing-sequencing for mixed-model assembly, *Management Science*, 14, 59-75.
 Whitley, D., Starkweather, T. and Fuquay, D. (1989), Scheduling problems and traveling salesmen: the genetic edge recombination operator, *Proceedings of the 3rd International Conference on Genetic Algorithms*, 133-140.



송원섭

전남대학교 산업공학과 학사
 전남대학교 산업공학과 석사
 현재: 전남대학교 산업공학과 박사과정
 관심분야: 생산라인관리, 진화알고리즘, 물류 최적화



신경석

전남대학교 산업공학과 학사
 전남대학교 산업공학과 석사
 현재: 전남대학교 산업공학과 박사과정
 관심분야: 물류최적화, 진화알고리즘 응용



김여근

서울대학교 산업공학과 학사
 서울대학교 산업공학과 석사
 서울대학교 산업공학과 박사
 현재: 전남대학교 산업공학과 교수
 관심분야: 진화알고리즘, 경영과학, 물류최적화, 생산라인관리