

# 비대칭형 다계층 공생 진화알고리즘을 이용한 FMS 공정계획과 일정계획의 통합

김여근<sup>1\*</sup> · 김재윤<sup>2</sup> · 신경석<sup>1</sup>

<sup>1</sup>전남대학교 산업공학과 / <sup>2</sup>전남대학교 고품질전기전자부품 및 시스템연구센터

## The Integration of FMS Process Planning and Scheduling Using an Asymmetric Multileveled Symbiotic Evolutionary Algorithm

Yeo Keun Kim<sup>1</sup> · Jae Yun Kim<sup>2</sup> · Kyoung Seok Shin<sup>1</sup>

<sup>1</sup>Department of Industrial Engineering, Chonnam National University, Gwangju, 500-757

<sup>2</sup>Research Center for High-Quality Electric Components and Systems,  
Chonnam National University, Gwangju, 500-757

This paper addresses the integrated problem of process planning and scheduling in FMS (Flexible Manufacturing System). The integration of process planning and scheduling is important for an efficient utilization of manufacturing resources. In this paper, a new method using an artificial intelligent search technique, called asymmetric multileveled symbiotic evolutionary algorithm, is presented to handle the two functions at the same time. Efficient genetic representations and operator schemes are considered. While designing the schemes, we take into account the features specific to each of process planning and scheduling problems. The performance of the proposed algorithm is compared with those of a traditional hierarchical approach and existing evolutionary algorithms. The experimental results show that the proposed algorithm outperforms the compared algorithms.

**Keywords:** process planning, FMS scheduling, multileveled integration, symbiotic evolutionary algorithm, coevolution

### 1. 서론

유연생산시스템(Flexible Manufacturing System; FMS)은 컴퓨터 통합생산시스템을 물리적으로 구현한 대표적인 형태로서, 수치제어 기계들이 컴퓨터 네트워크와 자동 물류시스템에 의해 연결되고, 기계에 장착된 공구들은 자동적으로 상호교환이 가능한 자동화된 제조시스템을 말한다(Guerroero *et al.*, 1999). FMS는 제품수명의 단축화, 제조기술의 다양화, 그리고 시장 환경의 변화와 같은 생산환경의 불확실성에 효율적으로 대처할 수 있는 제조시스템으로 알려져 있다. FMS의 효율적인 운영과 관련하여 해결해야 할 여러 문제들이 있다. 본 연구에서

는 이들 중에서 공정계획문제와 일정계획문제를 다룬다. 공정계획(process planning)문제는 기술적 제약조건을 만족하면서 각 부품생산에 적합한 작업(operation)경로를 선택하고 각 작업에 기계나 공구 등 생산자원을 할당하는 문제로 정의되고, 일정계획(scheduling)문제는 주어진 생산목표에 따라 생산자원에서 원활한 가공이 이루어질 수 있도록 작업의 시간적 순서를 결정하는 문제로 정의된다. 본 연구에서는 이들 두 문제를 동시에 통합적으로 다루고자 한다.

FMS 공정계획문제와 일정계획문제는 서로 독립적인 문제가 아니고, 밀접한 상호 관련성을 갖는다. 즉, 일정계획을 고려하지 않고 공정계획을 수립하면, 그 공정계획은 비효율적인

본 연구는 한국과학재단 지역대학 우수과학자 지원연구(R05-2002-000-00474-0) 지원으로 수행되었음.

\* 연락처 : 김여근 교수, 500-757 광주시 북구 용봉동 300 전남대학교 산업공학과, Fax : 062-530-1789, E-mail : kimyk@chonnam.ac.kr

2003년 11월 접수; 2004년 3월 수정본 접수; 2004년 4월 게재 확정.

일정계획을 야기하거나, 최악의 경우 공정계획의 결과가 작업 과부하와 기계 고장 등에 의하여 사용 불가능할 수도 있다. 그러나 기존의 많은 연구들에서는 두 문제가 갖는 복잡성 때문에 두 문제를 서로 분리시켜 독립적으로 다루었거나, 많은 가정들을 포함하여 순차적으로 해결하였다. 즉, FMS를 구성하는 기계 및 공구들은 다기능성으로 인하여 여러 작업들을 수행할 수 있으므로, 작업에 대한 자원할당이나 가능한 작업경로의 선택에 있어 많은 대안들이 존재할 수 있다. 따라서, FMS 공정계획문제는 다양한 유연성(flexibility)으로 인하여 복잡도가 매우 높다. 또한, FMS 일정계획문제는 대안의 기계나 공구가 존재하고 공구매겨진 용량이나 가용 공구수 등과 같은 공구 관련 제약들이 고려되어야 하므로, 주문형 생산(job shop) 시스템의 일정계획문제보다 더 복잡한 문제가 된다(Rachamadugu and Stecke, 1994). 구체적인 연구현황들은 다음 장에서 언급한다.

본 연구에서는 FMS 공정계획과 일정계획의 두 문제를 동시에 통합적으로 해결할 수 있는 새로운 방법을 제안한다. 또한, 기존 연구들과 달리 문제의 복잡도를 낮추기 위하여 문제의 해공간을 축소하지 않고, 현실적인 제약들을 적절히 고려하고자 한다. 이들 두 문제의 통합적 해결을 위한 방법론으로 공생 진화알고리즘(symbiotic evolutionary algorithm)을 사용한다. 공생 진화알고리즘은 자연계에서 둘 또는 그 이상의 종들이 공생(symbiosis)의 상호작용을 통해 서로 적응해 나가는 공진화(coevolution) 현상을 모방한 확률적 탐색기법이다(Potter, 1997). 이 알고리즘은 통합문제에서 적용의 우수성을 갖는 것으로 알려져 있다(Potter, 1997; Kim *et al.*, 2000).

본 논문의 구성은 다음과 같다. 제2장에서는 본 연구에서 다루는 FMS 공정계획문제와 일정계획문제를 소개하고, 두 문제의 통합문제가 갖는 특징 및 관련 연구현황들을 기술한다. 제3장에서는 FMS 공정계획과 일정계획의 통합적 해결을 위한 방법론으로 제안하는 알고리즘의 개념, 특징, 그리고 절차에 관하여 설명한다. 제4장에서는 본 연구에서 제안하는 알고리즘의 적용을 위한 유전요소들을 설명하고, 제5장에서는 실험설계와 실험결과를 제시하며 결과에 대하여 분석한다. 마지막으로 제6장은 결론으로 구성되어 있다.

## 2. FMS 공정계획문제와 일정계획문제

### 2.1 FMS 공정계획문제

FMS 공정계획문제는 고려하는 유연성의 종류와 범위에 따라 문제가 갖는 결정변수의 종류와 수, 그리고 복잡도와 구조 등이 결정된다(Guerrero *et al.*, 1999). FMS의 유연성들에 대한 분류 및 정의는 Stecke and Raman(1995)의 연구를 참조할 수 있다. 본 연구에서는 기계유연성(machine flexibility), 공구유연성(tool flexibility), 공정유연성(process flexibility), 그리고 순서

유연성(sequence flexibility)을 고려한다.

기계유연성은 대안기계에 의한 작업 실행의 가능성으로 정의한다. 기계유연성은 다양한 형태의 작업을 가공할 수 있는 기계능력으로 인하여 고려된 것이다. 공구유연성은 대안공구에 의한 작업실행의 가능성 또는 여러 작업유형을 가공할 수 있는 공구의 능력으로 정의한다. FMS는 다기능성과 자동화된 교환장치를 갖는 머시닝센터를 사용하므로, 이들 두 유연성은 중요하게 고려되어야 한다. 특히, 공구유연성을 고려하지 않은 공구할당 결과는 실제 생산현장에 적용하는 데 한계를 가질 수 있다(Modi and Shanker, 1994). 공정유연성은 하나의 같은 특징형상(feature)을 가공하는 데 있어 대안작업 또는 대안공정이 있음을 의미한다. 여기서 공정(process)은 하나 이상의 작업 집합을 나타낸다. 대안작업이나 대안공정은 특징형상의 가공방법, 기계의 다기능성, 대안공구들로 인하여 다양하게 생성될 수 있다. 마지막으로, 순서유연성은 요구되는 작업들의 가공순서가 바뀔 수 있는 가능성을 말한다. 순서유연성을 고려함으로써 병목현상에 의한 자원들의 비효율적인 사용을 방지할 수 있다. 공정계획단계에서 여러 유연성을 고려함으로써 다양한 일정계획의 생성에 유리한 측면이 있고, 평균처리 시간(mean flow time)이나 기계활용성(machine utilization) 측면에서 좋은 성능을 가져올 수 있다(Rachamadugu and Stecke, 1994).

본 연구에서 부품가공정보는 Ho and Moodie(1996)에 의해 제안된 네트워크 표현(network representation) 형태로 주어졌다고 가정한다. 네트워크 표현에서는 위에서 언급한 4가지 유연성이 모두 표현되며, 유연성으로 인하여 형성되는 모든 가능한 경우의 수를 고려할 수 있게 함으로써 FMS에서 유연성이 갖는 강점을 유지할 수 있다. <Figure 1>은 어떤 한 부품의 가공정보를 나타내는 하나의 예이다. <Figure 1(a)>에서 시작노드(starting node)와 끝노드(ending node)는 가상노드(dummy node)로, 제조공정에 있어 시작과 끝을 나타낸다. 중간노드들은 <Figure 1(b)>에서 보는 바와 같이 작업번호, 각 작업을 수행할 수 있는 대안기계 및 대안공구, 그리고 이에 대한 가공시간을 포함하고 있다.

예를 들어, <Figure 1(b)>는 <Figure 1(a)>의 작업 18을 확대해 보인 것이다. 작업 18은 기계 1, 4, 8 중 하나의 기계에서 작업될 수 있고, 3대의 기계들은 각각 여러 대안공구들 중에서 하나를 선택할 수 있다. 이때 가공시간은 기계 및 공구의 성능이나 작업준비시간 등으로 인하여 서로 다를 수 있다. 중간노드에 포함된 내용은 기계유연성과 공구유연성을 설명할 수 있다. 노드를 연결하는 화살표는 작업 간의 선행관계를 나타낸다. 'OR' 관계는 동일한 특징형상을 가공하는데 대안작업 또는 대안공정이 존재할 수 있다는 공정유연성을 표현한다. 만약, 노드의 연결이 'OR' 기호에 나누어지면 한쪽만을 따른다. 예로, 'OR1'의 경우, 대안작업 3 또는 대안공정 {4, 5}를 택하게 된다. 순서유연성은 네트워크의 작업 선행관계에 의해 다양한 작업 가공순서를 가짐으로써 나타난다.

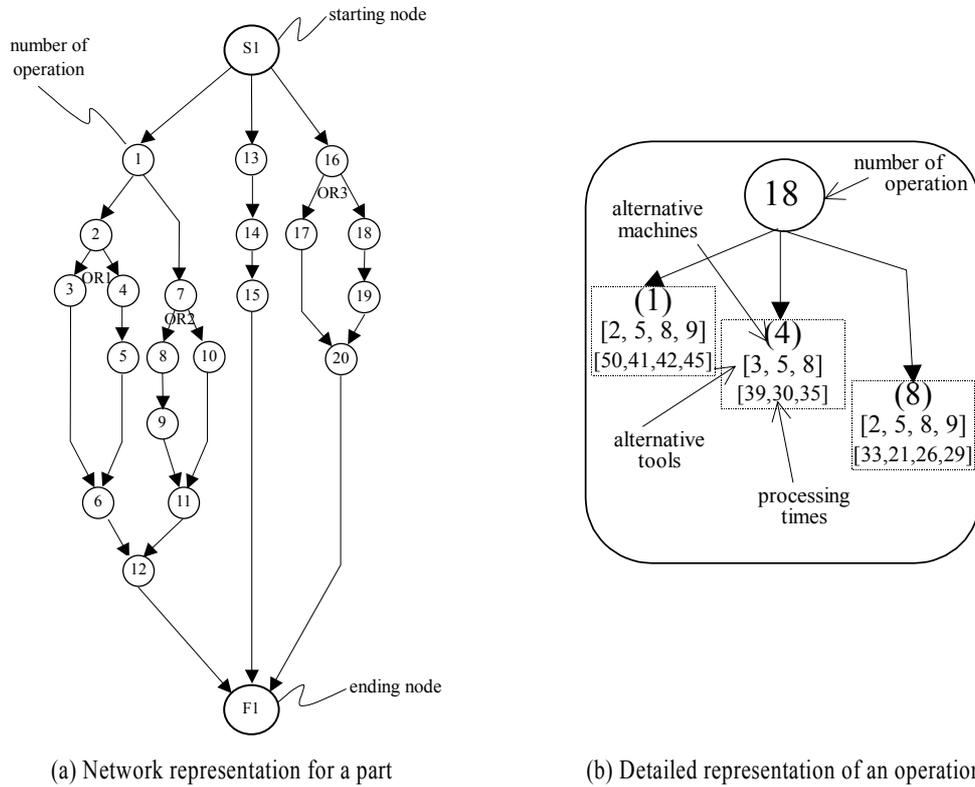


Figure 1. Network representation of FMS process plan.

2.2 FMS 일정계획문제

FMS를 구성하는 기계는 다기능성을 갖는 머시닝센터로 구성되므로, 각 작업마다 많은 수의 대안기계를 갖는다. 또한, 각 기계마다 여러 종류의 공구를 장착하는 것이 가능하다. 이것은 각 부품마다 가공순서가 정해져 있다고 하더라도 작업별로 많은 대안기계 및 대안공구를 갖게 되어 FMS 일정계획문제를 어렵게 한다.

FMS는 상황에 따라 다양하게 문제를 정의할 수 있으므로 다루는 문제의 범위 및 특징이 주어져야 한다. 본 연구에서 다루는 FMS는 다음과 같은 상황에서 운영된다고 본다.

- (1) 생산계획기간과 그 기간 동안 생산해야 할 부품들은 주어져 있다.
- (2) FMS는 머시닝센터로 구성되며, 적절한 공구가 주어지면 다양한 작업을 수행할 수 있다.
- (3) 각 기계의 공구매거진은 용량은 다르고, 그 한계용량이 주어져 있다.
- (4) 생산계획기간에 사용되어질 유형별 가용 공구수의 정보는 주어져 있으며, 한 작업은 다양한 공구유형에 의해 가공되어질 수 있다. 그리고 공구수명은 고려하지 않는다.
- (5) 각 부품 및 지원장치의 준비시간 및 이동시간은 가공시간에 포함된다.

- (6) 부품의 선점(preemption)과 작업의 분할은 허용되지 않는다. 따라서 하나의 작업이 완료될 때까지 기계 및 공구의 변화는 발생하지 않는다.
- (7) 모든 부품은 시작시간부터 동시에 이용될 수 있다.

2.3 통합문제

본 연구에서 해결하고자 하는 문제는 앞에서 언급한 FMS 환경에 부여된 여러 제약들을 위반하지 않고, 총 처리시간(makespan)의 최소화를 목적으로 여러 부품의 생산을 위한 공정계획과 일정계획을 동시에 결정하는 문제로 정의된다. 이때, 각 부품생산을 위한 가공정보에는 기계, 공구, 공정, 그리고 순서유연성을 고려한다. 총 처리시간의 최소화는 기계이용률을 최대화하기 위한 것으로, 공정계획과 일정계획을 통합하여 다음으로써 그 효과가 증대될 수 있다.

이제, FMS 공정계획과 일정계획의 통합문제가 갖는 문제의 구조를 살펴보자. 통합문제는 크게 공정계획과 일정계획의 두 부분문제로 나누어질 수 있다. 공정계획 부분문제는 다시 부품별로 유연성에 따라 (1) 각 작업에 기계할당, (2) 각 작업에 공구할당, (3) 대안작업 또는 대안공정 결정, 그리고 (4) 작업순서 결정 등 4가지의 부분문제로 또다시 나누어질 수 있다. 이를 그림으로 표현하면 <Figure 2>와 같다. 즉, 최상위 계층인

계층 4에는 완전한 통합생산계획 문제가 존재한다. 계층 3은 통합생산계획의 두 부분문제인 공정계획문제와 일정계획문제가 위치한다. 계층 2는 공정계획문제가 분해되어 공정계획의 대상이 되는 부품별 공정계획문제( $Part_i, i=1,2,\dots,n$ )로 구성되고, 최하위 계층인 계층 1에는 각 부품별 기계( $M_i$ ), 공구( $T_i$ ), 공정( $P_i$ ), 순서( $S_i$ )를 결정하는 문제가 위치한다. 일정계획문제는 부품별 일정계획으로 분리될 수 없으므로, 계층 3으로부터 더 이상 분해되지 않는다. 결과적으로, FMS 공정계획과 일정계획의 통합문제는 하위의 부분문제가 결합되어 상위의 문제를 형성하고 이들이 여러 계층을 이루며, 부분문제에 따라 계층의 깊이가 서로 다른 형태를 갖는다. 본 연구에서는 이와 같은 형태의 문제를 비대칭형 다계층 통합문제(asymmetric multileveled integrated problem)라 부르기로 한다. 그리고 계층 3에 있는 공정계획문제의 부분문제들의 모든 가지는 깊이가 같다. 이렇게 통합문제에서 분해되어 나가는 가지의 깊이가 모두 같으면 대칭형 다계층 통합문제(symmetric multileveled integrated problem)라 부르기로 한다.

2.4 관련 연구현황

FMS 공정계획과 일정계획에 관한 많은 연구들이 이루어져 왔다. 여기에서는 문제의 해결기법 측면보다는 다루는 문제의 상황 측면에서 관련 연구현황을 살펴보고자 한다. 특히, 고려하는 유연성이나 문제상황, 그리고 제약에 초점을 두고 설명한다.

FMS 공정계획에 관한 연구는 Modi and Shanker(1994), Stecke and Raman(1995), Kumar and Shanker(2000) 등에 의해 이루어졌다. 그러나 이들 연구에서는 일정계획을 고려하지 않고 있다. FMS 일정계획문제는 Paulli(1995), Liu and Maccarthy(1997), Sabuncuoglu(1998) 등에 의해 연구가 수행되었다. 이들 연구는 모두 하나의 고정된 공정계획이 주어진 상태에서 일정계획문제를 해결하는 기법에 관하여 연구하였다.

FMS 통합문제를 다룬 연구로는 Saygin and Kilic(1999), Morad and Zalzal(1999), Alexander *et al.*(1999) 등의 연구가 있다. Saygin and Kilic(1999)은 각 부품별로 기계유연성과 공정

유연성을 고려하여 하나의 공정계획을 수립하고, 이를 입력자료로 일정계획을 결정하는 방법을 제안하였다. 그리고 구한 일정계획이 시스템의 목적에 적합하지 않으면 재일정계획을 수행함으로써 순차적이고 반복적으로 두 문제의 정보가 피드백(feedback)되는 방법을 제안하였다. Morad and Zalzal(1999)은 진화알고리즘을 이용하여 공정계획과 일정계획문제를 동시에 해결하고자 하였다. 이 연구에서 부품의 작업 가공순서는 고정되어 있으며, 기계유연성만을 고려하였다. 또한, 공구매거진 및 공구 관련 제약을 고려하지 않았다. Alexander *et al.*(1999)은 대안공정계획을 고려하고, 타부 서치(tabu search)를 이용하여 통합문제를 다루었다. 여기서, 대안공정계획이란 사전에 결정된 일정한 개수의 공정계획을 말하고, 이들 중에서 선택된 하나를 입력자료로 하여 일정계획을 수립한다. 또한, 공구유연성과 공구매거진의 한계용량 제약 등을 고려하지 않았다.

이상에서 살펴본 바와 같이, FMS 공정계획문제 및 일정계획문제와 관련된 많은 연구들은 문제 간 상호 관련성을 무시하고, 서로 분리하여 독립적으로 다루었다. 또한, 통합문제에 관한 연구들에서는 문제의 복잡도를 축소하기 위하여 다루는 유연성에 의해 발생하는 경우의 수를 제한하거나 문제를 단순화하였다. 또한, 통합문제가 구조적으로 갖는 특징을 문제해결기법에 반영하지 않았다. 조사한 바에 의하면, 본 연구에서 다루는 문제상황 및 문제구조를 고려하여 FMS 통합문제를 해결한 연구는 아직까지 이루어지지 않았다.

3. 비대칭형 다계층 공생 진화알고리즘

3.1 알고리즘의 개념

통합문제를 위한 해결의 방법론으로 제안되었던 기존 대부분의 공생 진화알고리즘들은 동일한 하나의 계층에 있는 부분 문제들의 통합에만 적용될 수 있었다. 즉, 기존 연구들은 통합적 해결의 관점에 대해서만 관심을 가졌으며, 통합문제가 갖는 구조적 특성을 고려한 연구들은 미흡하다. 다계층 문제를

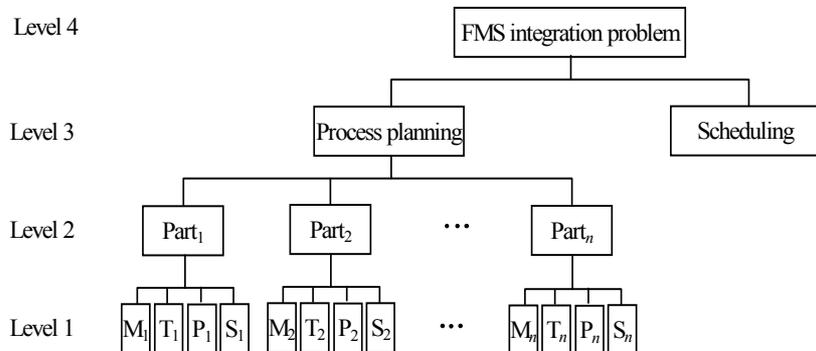


Figure 2. The structure of the FMS integration problem.

단일계층 문제로 변환하여 기존의 알고리즘을 해결의 방법론으로 채택할 수도 있으나, 이러한 경우에 부분문제의 수가 많아지면, 한 계층에 모든 모집단을 두고 동시에 운영해야 하므로 좋은 해에 관한 정보를 추출하고, 이를 유지 또는 진화시키는 데 비효율적일 수 있다. 최근, Kim *et al.*(2003a)은 FMS의 부하할당(load)문제, Kim and Min(2003)은 SCM(supply chain management)에서 생산계획과 분배계획의 통합문제를 위한 다계층 공생 진화알고리즘을 제안하였다. 그러나 이들 연구는 통합문제의 모든 가지의 깊이가 동일한 대칭형 다계층 구조를 대상으로 이루어졌으며, 비대칭형 다계층 구조를 갖는 통합문제를 위한 알고리즘에 관한 연구는 아직까지 이루어지지 않고 있다.

본 연구에서는 자연계의 내공생진화(endosymbiosis evolution) 현상을 모방하여 비대칭형 문제를 효과적으로 해결할 수 있는 알고리즘을 개발하였다. 내공생진화는 공생진화의 특별한 형태로, 생물학자인 Margulis(1970)가 복잡한 세포구조를 갖는 진핵생물(eukaryote)의 출현을 설명하기 위하여 주장한 이론이다. 내공생진화론은 한 생물이 다른 생물의 내부에서 공생하여 마치 하나의 생물처럼 생존하고, 이들이 다시 새로운 생물과 내공생해 나가며 점차로 환경에 적응해 나가 고등생물로 진화해 간다는 이론이다. <Figure 3>은 Margulis가 내공생진화론을 설명하기 위하여 보인 진핵세포의 진화과정이다. 내공생진화론은 현재 대부분의 생물학자들이 지구상의 생명체 기원과 진화를 설명하기 위해 사용하며, 부분이 아닌 전체의 관점에서 문제의 해결방법을 찾으려는 시스템적 사고를 표현하는 하나의 예로 사용되기도 한다(Campbell *et al.*, 1996; Capra, 1996). 본 연구에서는 <Figure 3>이 구조적인 측면에서 <Figure 2>의 비대칭형 다계층 구조와 유사하고, 내공생진화론의 진화메커니즘은 다계층 통합문제에서 계층 내 또는 계층 간 부분문제들의 관련 메커니즘과 비슷하다는 점에 착안하여

알고리즘을 설계하였다.

제안한 알고리즘의 개념은 다음과 같다. 완전한 통합문제에 최상위 계층에 위치한다. 계층의 수준이 내려갈수록 상위계층에 존재하는 문제들이 부분문제로 분해되어 하위계층을 구성해 나간다. 그리고 최하위 계층은 최소단위의 부분문제들로 형성된다. 따라서, 계층의 수준이 높을수록 각 계층에는 하위계층의 문제들이 결합되어 좀더 복잡한 구조를 갖는 문제들이 존재한다. 그리고 각 문제들은 자신이 속해 있는 계층의 다른 문제들과 상호작용하고, 계층 간에는 하위계층의 정보가 상위계층으로 전달되는 형태로 해를 탐색해 나간다. 이때, 비대칭 구조에서 부분문제들은 계층 깊이가 서로 다를 수 있으므로, 이를 효율적으로 다룰 수 있는 진화 메커니즘을 갖는다. 본 연구에서 제안한 알고리즘은 비대칭형 구조를 갖는 다계층 통합문제를 해결하기 위한 것이고 공생 진화알고리즘을 기반으로 하였으므로, 이를 비대칭형 다계층 공생 진화알고리즘(Asymmetric Multileveled Symbiotic Evolutionary Algorithm; AMSEA)이라 부른다.

### 3.2 알고리즘의 특징

비대칭형 다계층 구조는 대칭형 구조를 포함할 수 있다. 예로, <Figure 2>에서 보는 바와 같이 FMS 통합생산계획 문제를 크게 공정계획문제와 일정계획문제로만 분해하면 2개의 계층이 형성되고 부분문제의 계층 깊이가 동일한 대칭형 다계층 통합문제가 된다. 따라서, AMSEA는 Kim *et al.*(2003a)과 Kim and Min(2003)이 제안한 대칭형 다계층 알고리즘이 갖는 특징들 중에서 비대칭형 다계층 문제에 사용 가능한 전략들을 포함하도록 하였다. 그리고 비대칭형 문제구조를 가짐으로써 발생하는 문제점에 대해서는 새로운 특징이 이를 다룰 수 있도록 설계하였다.

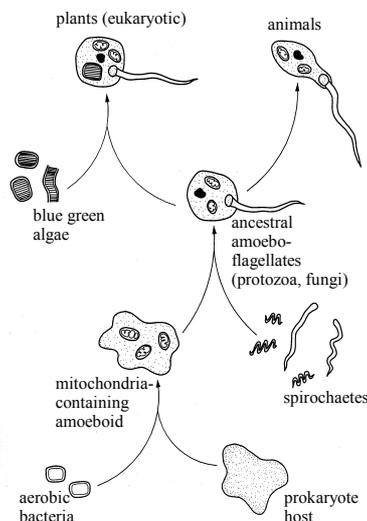


Figure 3. Evolution of eukaryote by endosymbiosis.

AMSEA는 계층구조를 갖지 않은 기존의 공생 진화알고리즘과 비교할 때, 계층구조에 따른 특징적인 전략들을 다음과 같이 갖는다. 첫째, 어떤 계층에 있는 모집단의 개체는 자신의 직하위 모집단들의 개체들을 서로 결합한 형태로 표현한다. 여기서, 직하위 모집단이란 계층도에서 가지가 직접 연결된 하위계층의 모집단을 말한다. 둘째, 계층 간에는 하위계층에서 가장 높은 적응도(fitness)를 갖는 개체의 결합을 상위계층으로 전달하는 상호작용을 한다. 이때, 전달 여부는 적응도를 기준으로 결정된다. 이들 두 특징은 내공생진화를 모방한 것이다. 셋째, 각 계층에 존재하는 모집단들은 이웃단위 진화전략을 사용하여 진화한다. 이상의 세 가지 특징들은 대칭형 다계층 알고리즘의 전략을 그대로 사용한 것이다. 이들의 효과는 Kim *et al.*(2003a) 또는 Kim and Min(2003)의 연구를 참조할 수 있다.

AMSEA의 네 번째 특징은 대칭형 다계층 알고리즘과의 차이점으로, 비대칭형 다계층 구조를 효율적으로 다룰 수 있는 새로운 진화 메커니즘을 갖는다. 이를 통해, 비대칭형 구조에서 각 부분문제의 가지 깊이가 서로 다르므로 인하여 발생하는 여러 문제들을 해결할 수 있다. AMSEA의 진화 메커니즘은 다음과 같다. 먼저, 계층 깊이가 낮은 가지들에 대해서는 새로운 가상 모집단을 구성하여, 모든 가지의 계층 깊이가 최대 계층 깊이와 같도록 그 구조를 재구성한다. 그리고 이러한 구조 하에서 내공생 진화과정을 모방한다.

AMSEA에서 새로운 진화 메커니즘이 필요한 이유는 비대칭형 구조의 문제가 대칭형 구조와 다르게 갖는 어려움들을 살펴봄으로써 알 수 있다. 우선, 비대칭형 구조에서는 상대적으로 계층깊이가 깊은 부분문제의 적응도 평가에 어려움이 있다. 계층구조에서 최상위 계층을 제외한 다른 계층의 개체들은 통합문제에 대한 부분해(partial solution)이다. 흔히, 공생 진화알고리즘에서의 적응도 함수는 통합문제의 목적에 맞도록 설정된다. 따라서, 부분해의 적응도는 다른 부분해들과 결합하여, 통합문제에 대한 완전해(complete solution)를 구성해야만 평가받을 수 있다. 이때, 적응도 평가에 참여하는 다른 모집단의 부분해(개체)를 공생 파트너(symbiotic partner)라고 부르기로 하자. 공생 파트너 선택전략은 다음 절에서 설명한다. 대칭형 구조에서는 각 계층마다 그 계층에 존재하는 모든 모집단에서 공생 파트너를 선택하면 완전해가 구성된다. 그러나, 비대칭형 구조에서는 계층깊이가 깊은 부분문제의 경우, 동일 계층에 있는 모든 모집단에서 공생 파트너들을 선택하더라도 완전해를 구성할 수 없다. 예로, <Figure 2>에서의 계층 1과 계층 2가 이에 해당한다. 따라서 계층깊이가 깊은 부분문제들은 적응도 평가를 위한 별도의 과정이 요구된다. 또한, 비계층형 구조에서는 각 부분문제의 진화 기회에 차이를 가질 수 있다. 만약, 진화가 계층단위로 이루어지는 경우, <Figure 2>에서 공정계획문제는 계층 1, 2, 3이 진화할 때마다 각각 진화하는 것이 되지만, 일정계획문제는 계층 3이 진화할 때에만 진화의 기회를 부여받게 된다. 그러므로, 비계층 구조에서는 계층 깊이가

상대적으로 낮은 부분문제들의 진화기회를 균등하게 하는 진화전략이 필요하다. AMSEA의 진화 메커니즘을 적용하면, 이들 두 문제를 효율적으로 해결할 수 있다.

그러나 비대칭형 구조를 위한 진화 메커니즘은 다양할 수 있다. 대안 가능성이 있는 메커니즘들과 이들의 특징, 그리고 AMSEA에서 위의 메커니즘을 선택한 이유에 대해서는 제5.3 절에서 구체적으로 설명한다. AMSEA의 이러한 특징들은 전체 진화과정이 효율적으로 운영될 수 있도록 하며, 진화연산의 측면에서도 좋은 탐색성능을 갖도록 하기 위한 것이다.

### 3.3 알고리즘의 절차

AMSEA에서 각 계층에 존재하는 모집단들은 토러스(torus) 형태의 2차원 정방형 격자구조를 갖는다고 본다.  $Pop[l, q]$ ,  $l=1,2,\dots,L$ ,  $q=1,2,\dots,Q_l$ 은 계층  $l$ 의  $q$ 번째 모집단,  $NP_{ij}[l, q]$ 는  $Pop[l, q]$ 에서 위치  $(i, j)$ 와 그 주위 8개의 개체로  $(3 \times 3)$  정사각형 형태의 구성된 이웃으로 정의한다.  $Imh(NP_{ij}[l, q])$ 는  $NP_{ij}[l, q]$ 의 직상위 모집단의 이웃이다. PES(Potential EndoSymbiont)는 어떤 이웃의 적응도 평가과정에서 발견된 가장 높은 적응도를 갖는 개체결합, WES(Worst EndoSymbiont)는 한 이웃 내에서 가장 낮은 적응도를 갖는 개체를 나타낸다. AMSEA의 절차는 아래와 같다.

#### 단계 1: 문제구조의 초기화

부분문제 중에서 계층깊이가 낮은 가지에 대해서 가상 모집단을 생성하여, 모든 가지의 계층깊이가 최대 계층깊이를 갖는 가지와 동일하도록 한다.

#### 단계 2: 알고리즘의 초기화 및 초기 적응도 평가

각 계층의 개체표현방법에 따라 개체들을 임의로 생성하여, 모든 계층의 모집단들을 구성한다. 그리고 각 계층의 적응도 평가전략에 따라 모든 개체들의 초기 적응도를 평가하고, 가장 높은 적응도를  $f_{best}$ 로 둔다.

#### 단계 3: 이웃정의

임의의 위치  $(i, j)$ 를 선택하고,  $NP_{ij}[l, q]$ ,  $l=1, 2, \dots, L$ ,  $q=1, 2, \dots, Q_l$ 을 정의한다.

#### 단계 4: 적응도 평가 및 계층 간 상호작용

단계 4.1:  $l=1, q=1$ 로 설정한다.

단계 4.2:  $q'(\neq q)$ 인 모든  $NP_{ij}[l, q']$ 에서 공생 파트너를 선택하여  $NP_{ij}[l, q]$  개체들의 적응도를 평가한다.  $NP_{ij}[l, q]$ 의 PES와  $Imh(NP_{ij}[l, q])$ 의 WES에 대하여,  $f(PES) > f(WES)$ 이면, WES는 PES의 해당 인자와 대체된다. 그리고  $f(PES)$ 가  $f_{best}$ 보다 높으면  $f_{best}$ 를 갱신한다.

단계 4.3:  $q \leftarrow q+1$ 로 둔다. 만약,  $q \leq Q_l$ 이면 단계 4.2로 간다.

단계 4.4:  $l \leftarrow l+1$ 로 둔다. 만약,  $l < L$ 이면  $q = 1$ 로 하고, 단계 4.2로 간다.

단계 4.5:  $NP_{ij}[L, q]$  개체들의 적응도는 공생 파트너 없이 평가하고, 가장 높은 적응도가  $f_{best}$ 보다 높으면  $f_{best}$ 를 갱신한다.

#### 단계 5: 이웃진화

단계 5.1:  $l = 1, q = 1$ 로 설정한다.

단계 5.2:  $NP_{ij}[l, q]$ 에서 적응도가 높은 두 개체를 확률적으로 선택하여 교차한 후, 두 자손개체를 생산한다.

단계 5.3:  $NP_{ij}[l, q]$ 에서 적응도가 낮은 두 개체를 확률적으로 선택하여 단계 5.2에서 생산한 자손개체들과 대체한다.

단계 5.4:  $NP_{ij}[l, q]$ 의 개체들에 대하여 돌연변이율에 따라 돌연변이시킨다.

단계 5.5:  $NP_{ij}[l, q]$ 의 진화종료조건을 만족하면 단계 5.6으로, 그렇지 않으면 단계 5.2로 간다.

단계 5.6:  $q \leftarrow q+1$ 로 둔다. 만약,  $q \leq Q$ 이면 단계 5.2로 간다.

단계 5.7:  $l \leftarrow l+1$ 로 둔다. 만약,  $l \leq L$ 이면  $q = 1$ 로 하고, 단계 5.2로 간다.

#### 단계 6: 종료조건

알고리즘 종료조건을 만족하면 종료하고, 그렇지 않으면 단계 3으로 간다.

단계 1을 위하여 통합문제를 계층별로 분해한 후, 각 계층마다 모든 부분문제의 계층깊이를 조사한다. 그리하여, 각 가지의 깊이가 다른 경우에는 상위계층부터 가상 모집단을 형성하여 그 깊이를 같도록 한다. 이때 가상 모집단은 그 모집단이 위치할 계층의 직상위 계층에 있는 모집단(또는 가상 모집단)과 같은 정보를 나타내도록 한다. 이와 같은 과정을 통합문제의 모든 가지가 최대 계층 깊이와 같아질 때까지 반복한다.

단계 2에서 계층 1의 모든 모집단에는 각 모집단이 나타내려는 부분해의 표현방법에 따라 임의로 개체를 생성한다. 계층 2부터 계층  $L$ 의 초기 모집단은 자신의 직하위 모집단의 표현을 결합한 형태로 임의의 생성한다. 단, 직하위 모집단이 1개인 모집단은 직하위 모집단의 표현과 동일한 형태로 임의의 생성한다. 따라서, AMSEA의 적용을 위해서는 최하위 계층에 존재하는 최소단위의 부분문제들을 위한 모집단의 표현방법만 결정하면 된다. 이러한 결합형태의 표현은 하위계층의 분리된 공생자(symbiont)들이 진화하여 상위계층의 결합된 내공생자(endosymbiont)로 생존해 나간다는 자연현상을 모방한 것이기도 하다. 단계 3은 진화대상이 되는 이웃을 정의하는 단계이다.

단계 4에서는 단계 3에서 선택된 이웃이 계층 내 및 계층 간에 다른 이웃들과 상호작용한다. 계층 내 상호작용은 동일한 계층 내에 있는 개체들을 공생 파트너로 선택하고, 이들 파트너들과의 결합을 통한 적응도 평가에 의해 이루어진다. 공생

파트너 선택전략은 다양할 수 있으나, 본 연구에서는 비교적 단순한 전략으로 동일 계층에 있는 다른 모집단의 이웃에서 임의로 한 개체를 선택하는 전략을 사용한다. 이 전략은 단계 2의 초기 적응도 평가과정에서도 동일하게 적용된다. 단, 최상위 계층  $L$ 에 있는 개체들은 해를 완전하게 표현하고 있으므로, 공생 파트너가 필요없다. 그리고, 계층 간 상호작용은 적응도가 높은 개체결합이 발견되면, 적응도를 기준으로 하여 상위 계층으로 전달하는 상향식 정보전달과정을 통해 이루어진다.

단계 5에서는 모든 계층에 존재하는 모집단의 이웃(가상 모집단 포함)들이 진화한다. 이때 진화는 안정상태 유전알고리즘(steady-state genetic algorithm)의 형태를 따른다. 안정상태 유전알고리즘은 높은 적응도를 갖는 개체가 생산되는 즉시 재생산에 참여하므로, 이 개체의 유전정보를 이용하여 좋은 해의 이용을 높일 수 있다(Syswerda, 1991). 단계 5.5는 선택된 이웃에 대한 진화 강도를 조절하기 위하여 사용된다. 한편, 유전 연산을 위해서는 개체 표현과 유사하게 계층 1에 있는 모집단에 적합한 연산자들만을 개발하면 된다. 계층 2부터 계층  $L$ 에 있는 개체들은 자신의 직하위 모집단들에 적용되었던 유전 연산자들을 그대로 사용한다.

## 4. FMS 통합문제를 위한 유전요소

### 4.1 FMS 공정계획문제를 위한 유전요소

#### 4.1.1 유전표현 및 초기모집단

진화알고리즘에서 개체는 자연스럽게 명확하며 중복되지 않도록 표현되는 것이 바람직하다. 또한 표현은 개체가 갖는 중요 정보가 유전연산에 의해 용이하게 추출되어 자손에게 잘 전파될 수 있어야 한다. 본 연구에서는 특성이 다른 두 부분문제의 통합문제를 다룬다. 여기에서는 부분문제마다 서로 다르게 개발된 유전요소들을 소개한다. 앞에서 언급한 바와 같이 AMSEA는 최하위 계층의 유전요소만을 설계하면, 다른 계층의 유전요소는 최하위 계층의 그것들을 그대로 사용하면 되므로, 여기에서는 각 부분문제별로 최하위 계층에서 요구되는 유전요소들만을 설명한다.

FMS 공정계획문제에 대하여 다음과 같은 개체표현방법을 사용한다. 공정계획문제에 대한 최하위 계층에는 각 부품별로 기계모집단, 공구모집단, 공정모집단, 그리고 순서모집단이 존재한다. 기계모집단의 개체표현은 각 작업별로 할당된 기계 번호를 표현한다. 즉,  $i$ 번째 인자값은 작업  $i$ 를 수행하는 기계 번호를 나타낸다. 기계모집단의 초기해는 각 작업별로 주어진 대안기계 중에서 임의로 생성한다. 공구모집단의 개체들은 기계모집단의 개체와 유사한 방법으로 표현하여,  $i$ 번째 인자값은 작업  $i$ 를 수행하는 공구번호를 나타낸다. 공구모집단의 초기해도 각 작업별로 주어진 대안공구 중에서 임의로 생성한다. 하나의 예로, 여섯 개의 작업들이 <Figure 4(a)>와 같이 각

각 대안기계 및 대안공구를 가질 때, 기계모집단과 공구모집단의 개체표현의 한 예는 <Figure 4(b)>와 같을 수 있다.

공정모집단 개체는 그 모집단이 나타내는 부품의 네트워크 표현에서 보이는 ‘OR’ 기호 개수만큼의 개체길이를 갖는다. 인자의 위치는 ‘OR’ 기호의 번호를 나타내고, 인자값은 그 위치에 해당하는 ‘OR’ 기호에 의해 분지되는 최대가지수를  $B$  라 한다면  $[1, B]$  구간의 정수값을 갖는다. 만약, 그 값이  $b$  라면 왼쪽에서  $b$  번째 가지의 대안작업(또는 대안공정)을 수행함을 의미한다. 이 모집단의 초기해는 각 인자위치마다 취할 수 있는 정수값 내에서 임의로 선택하여 생성한다. <Figure 1(a)>를 기준으로 공정모집단의 개체를 표현하면, (1 1 2)로 나타낼 수 있다.

순서모집단의 개체는 순열표현을 사용하여 하나의 부품을 완성하기 위한 가공순서들을 표현한다. 이때, 이 순열은 모든 가능한 작업들을 포함하고, 작업의 선행관계를 만족하도록 한다. 해당부품의 가공순서는 이 순열에서 공정모집단의 개체가 갖는 정보로부터 선택되어지지 않는 ‘OR’ 노드의 작업들을 삭제하면 결정된다. 순서모집단의 초기해는 순서제약을 만족하는 가공순서를 갖는 개체들을 임의로 생성하여 구성한다. 순서제약을 만족하는 순서모집단의 초기해를 생성하기 위한 방법은 다음과 같다. 먼저, 선행작업이 없는 작업들의 집합  $A$ 를 생성하고, 집합  $A$ 에서 임의로 한 작업을 선택하여 순서열에 추가한다. 집합  $A$ 에서 선택된 작업을 삭제하고, 선행작업이 모두 선택된(할당된) 작업을 집합  $A$ 에 삽입한다. 이와 같은 과정을 집합  $A$ 가 공집합이 될 때까지 반복한다. 예를 들어, <Figure 1(a)>의 선행공정도에서 순서모집단의 한 개체는 (1 2 7 13 14 15 16 3 8 9 10 4 5 6 18 19 17 11 12 20)로 표현될 수 있다.

계층 2, 3, 4에 있는 모집단의 개체는 자신의 직하위 모집단들의 개체들을 서로 결합한 형태로 표현된다. 따라서, 계층 2에 있는 부품별 모집단들의 개체는 기계모집단의 개체( $M_i$ ), 공구모집단의 개체( $T_i$ ), 공정모집단의 개체( $P_i$ ), 그리고 순서모집단의 개체( $S_i$ )를 결합한 형태로 표현된다. 마찬가지로, 계층 3의 개체는 부품별 모집단의 개체들을 결합한 형태로 표현한다. 마지막으로, 계층 4에 있는 모집단의 개체는 계층 3에 있는 공정계획 모집단의 개체와 제4.2절에서 설명할 일정계획 모집단의 개체가 결합되어 표현된다.

#### 4.1.2 유전연산자

교차와 돌연변이를 포함하는 유전연산자는 문제의 특성을 반영할 수 있는 정보를 두 부모로부터 추출하여 자손을 생성할 수 있어야 하며, 재생산된 자손 개체의 가능해가 유지되도록 해야 한다. 유전연산자도 최하위 계층에 대해서만 설명하고, 다른 계층의 유전연산은 개체표현과 동일하게 하위계층에서 사용한 연산자들을 순차적으로 적용하면 된다.

계층 1에 있는 기계모집단과 공구모집단의 교차는 이점(two-point)교차를 사용한다. 공정모집단의 개체에 대해서는 개체길이가 비교적 짧으므로, 일점(one-point)교차를 사용한다. 세 모집단에 대한 돌연변이는 모두 같은 방법을 사용한다. 즉, 개체돌연변이율에 따라 돌연변이될 개체를 선택하고, 선택된 개체의 각 인자에 인자돌연변이율을 적용하여 그 값에 따라 대안기계, 대안공구, 대안작업 또는 대안공정(‘OR’ 기호의 가지) 중에 임의로 선택하는 방법을 택하였다.

순서모집단의 개체들은 수정(modified)이점교차를 사용한다. 수정이점교차는 <Figure 5(a)>에서 보인 바와 같으며, 구체적으로 다음의 과정을 따른다. 먼저 두 개의 임의의 절단점을 선정한다. 두 절단점에 의하여 나누어진 앞부분, 중간부분, 뒷부분에 있는 인자 중 앞부분과 뒷부분은 한 부모( $P1$ )로부터 자손( $O1$ )의 같은 위치에 복사하고, 복사된 인자들과 같은 다른 부모( $P2$ )의 인자들을 지운다.  $P2$ 에 남아 있는 인자들은  $P2$ 에 나타나는 순서대로  $O1$ 의 채워지지 않은 위치에 순서대로 복사한다. 수정이점교차는 새롭게 생성된 자손개체가 작업 간 선행 순서제약을 만족하도록 한다. 순서모집단내 개체의 돌연변이 방법은 <Figure 5(b)>처럼 진행된다. 먼저, 부모개체로부터 한 인자를 임의로 선택한다. 돌연변이 후의 개체가 선행관계를 어기지 않도록 하기 위하여, 선택된 인자의 모든 직선행 작업과 직후행 작업을 조사하여 인자의 삽입 가능 범위를 구한다. 이 범위 내에서 선택된 인자를 임의의 위치에 삽입한다.

### 4.2 FMS 일정계획문제를 위한 유전요소

#### 4.2.1 유전표현 및 초기모집단

일정계획은 각 기계에 대하여 고려하는 모든 부품들의 완성에 필요한 작업들의 시간적 순서를 결정하는 것이다. 일정계

operation	alternative machines	alternative tools
1	1, 3, 4, 5, 10	1, 4, 6, 8
2	2, 5, 6, 8, 9	2, 5, 7
3	1, 3, 9, 10	2, 3, 5, 9
4	4, 5, 6, 7	7, 10
5	5, 6, 7, 9, 10	3, 6, 7, 9
6	1, 2, 4, 8, 9, 10	4, 7, 8

(a) Alternative machines and tools for each operation

Individual of machine population:

(3 5 9 5 7 10)

Individual of tool population:

(6 7 2 10 6 8)

(b) Representations of machine and tool populations

Figure 4. Example of the representations for machine and tool populations.

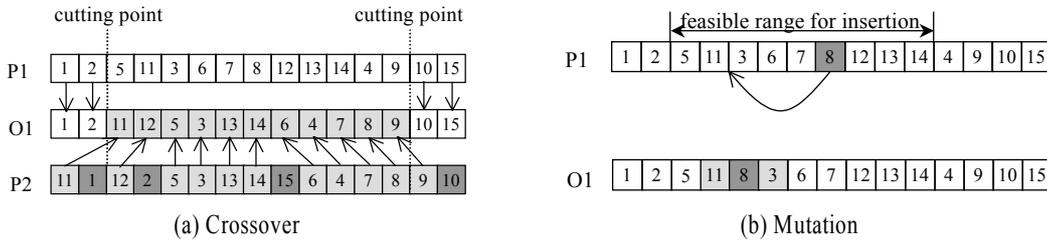


Figure 5. Genetic operators for sequence population.

확문제를 표현하는 모집단의 개체들이 의미를 갖기 위하여, 공정계획문제를 위한 모집단에서 공정계획들을 제공한다고 가정한다.

일정계획 모집단의 개체는 순열표현을 사용한다. 이것은 일정계획을 하나의 순서로 표현하는 것이 자연스럽고, 진화알고리즘의 측면에서 볼 때 부모에게 포함되어 있는 중요한 정보가 자손에게 손쉽게 전달될 수 있기 때문이다. 개체의 길이는 일정계획의 대상이 되는 모든 부품의 총 작업수의 합이다. 예를 들어, 일정계획을 하고자 하는 부품이 2개이고, 각각의 작업수가 15개, 9개라 하자. 그러면, 개체의 길이는 24가 되고, 인자값에서 1~15까지는 부품 1의 작업, 16~24까지는 부품 2의 작업을 나타낸다. 일정계획 모집단의 초기해는 부품수 및 총 작업수에 따라 순열표현에 의해 임의로 생성하여 구성한다.

순열표현은 직접적으로 가능한 일정계획을 표현하지 못하므로, 일정계획으로의 해석절차가 요구된다. 순열표현을 일정계획으로 해석하는 절차는 다양할 수 있다. Bierwirth and Mattfeld(1999)는 순열표현을 semi-active, active, non-delay, 그리고 혼합(hybrid) 일정계획으로 해석하는 절차들을 소개하였다. 특히, 이 연구에서는 active와 non-delay 일정계획을 결합하여 생성하는 혼합 일정계획을 제안하였다. 실험에 의하면, 혼합일정계획은 해의 질을 향상시키고 동시에 계산시간을 줄일 수 있는 것으로 나타났다. 본 연구에서는 이 절차를 채택하여 사용한다.

혼합 일정계획을 생성하는 절차는 아래와 같다. 아래 절차에서,  $o_{ij}$ 는 부품  $i$ 의  $j$ 번째 작업,  $\sigma_{ij}$ 는 작업  $o_{ij}$ 의 가장 이른 시작 시간,  $p_{ij}$ 는 작업  $o_{ij}$ 의 가공시간, 그리고  $\tau_{ij}$ 는 작업  $o_{ij}$ 의 가장 이른 완료시간을 의미한다. 즉,  $\tau_{ij} = \sigma_{ij} + p_{ij}$ 이다. 위에서 언급한 바와 같이, 일정계획문제를 위한 유전표현은 각 부품에 대하여 하나의 고정된 공정계획이 주어졌다는 가정하에 사용된다. 그 역할은 공정계획문제를 위한 모집단에서 선택된 공생 파트너가 하며, 이를 통해  $o_{ij}$ 가 유일하게 결정된다.

단계 1: 모든 시작작업의 집합을 만든다.

$$A := \{o_{ij} \mid 1 \leq i \leq n\}$$

단계 2:  $\tau^* = \min\{\tau_{ij} \mid o_{ij} \in A\}$ 인 기계  $m^*$ 를 결정한다.

단계 3: 집합  $A$ 의 작업 중 기계  $m^*$ 에서 가공될 수 있는 모든 작

업들로 집합  $B$ 를 만든다. 그리고

$$\sigma^* = \min\{\sigma_{ij} \mid o_{ij} \in B\}$$

단계 4: 파라미터  $\theta$ 에 의해  $C$ 를 만든다.

$$C := \{o_{ij} \in B \mid \sigma_{ij} \leq \theta\tau^* + (1-\theta)\sigma^*, 0 \leq \theta \leq 1\}$$

단계 5: 집합  $C$ 에 있는 순열에서 가장 왼쪽에 위치한 작업  $o_{ij}^*$ 를 선택하고, 이를 집합  $A$ 에서 삭제한다.

$$A := A \setminus \{o_{ij}^*\}$$

단계 6: 작업  $o_{ij}^*$ 를 일정계획에 추가하고, 그 시작시간과 완료 시간을 계산한다.

단계 7: 작업  $o_{ij}^*$ 의 후행작업  $o_{i,j+1}^*$ 가 있으면, 이를 집합  $A$ 에 추가한다.

$$A := A \cup \{o_{i,j+1}^*\}$$

단계 8: 집합  $A$ 가 공집합이 아니면 단계 2로 간다. 그렇지 않으면 종료한다.

위의 절차에서, 파라미터  $\theta$ 는 기계가 유휴상태를 유지할 수 있는 한계를 정의하는 데 사용될 수 있다. 극단적인 경우로,  $\theta = 0$ 이면 non-delay 일정계획을 생성해 내고,  $\theta = 1$ 이면 active 일정계획을 생성해 낸다. Bierwirth and Mattfeld(1999)는  $\theta = 0.5$ 일 때 좋은 결과를 보인다고 보고하였다. 본 연구에서도 같은 값을 사용한다.

#### 4.2.2 유전연산자

교차는 Bierwirth and Mattfeld(1999)에 의해 제안된 PPX (Precedence Preservative Crossover)를 사용한다. 그 절차는 다음과 같다. 먼저, 개체의 길이와 같은 크기의 배열을 생성하고, 배열의 인자에 1 또는 2를 임의로 넣는다. 이러한 배열에서 인자값이 1인 경우는 부모 1에서 가장 왼쪽에 있는 인자를 복사하고, 이에 해당하는 인자를 부모 1, 2에서 모두 지운다. 인자값이 2인 경우에는 부모 2에서 가장 왼쪽에 있는 인자를 복사하고, 이에 해당하는 인자를 부모 1, 2에서 모두 지운다. 이와 같은 방법으로 개체의 모든 인자값이 채워질 때까지 반복한다. PPX는 두 부모개체가 갖는 작업의 상대적 순서를 거의 동일한 비율로 유지한다. 일정계획문제에서 각 기계에 할당된 작업의 순서가 특별히 중요하다는 것은 명백하다. 돌연변이는 돌연변이될 개체를 선택하고, 선택된 개체의 모든 인자에 대

해 돌연변이 여부를 판단하고, 선택된 인자들을 임의의 위치에 삽입하는 삽입 돌연변이를 사용한다.

### 4.3 적응도 평가

적응도 평가함수는 다루는 문제의 목적함수 자체를 흔히 사용한다. 그러나 본 연구에서 다루는 문제와 같이, 다양한 제약을 갖는 문제에서는 목적함수 자체로 적응도를 평가하기 어렵다. 따라서 본 연구에서 각 부품의 대안기계제약, 대안공구제약, 대안공정제약 및 작업 순서제약은 각기 기계모집단, 공구모집단, 공정모집단, 그리고 순서모집단의 표현 및 유전연산에서 만족되도록 한다. 앞에서 제시한 각 모집단의 표현과 유전연산은 이들 제약을 모두 충족시킬 수 있도록 설계되었다. 그러나 이러한 표현과 유전연산은 공구 매거진 제약 및 공구의 슬롯 제약을 항상 만족시킬 수 없다는 한계를 갖는다. 따라서 이들 제약에 대해서는 제약 위반 시 벌금을 부과한다.

$$g_{it}(a, b) = mkspan_{it}(a, b) + c_1 \sum_t TP(t)^\alpha + c_2 \sum_m MP(m)^\beta \quad (1)$$

식 (1)은 이웃  $NP_{ij}[l, q]$ 의 위치  $(a, b)$ 에 있는 개체의 총 처리시간과 제약어긋에 대한 벌금을 합한 값을 계산하는 식이다. 식 (1)의 첫 번째 항은 그 개체의 총 처리시간을 나타낸다. 다음 두 항은 각각 가용공구수 제약(Tool Penalty; TP)과 공구매거진의 한계용량제약(Magazine Penalty; MP)에 대한 벌금함수를 표현한 것이다. 함수  $TP(t)$ 는 공구  $t$ 의 가용 공구수 제약을 어긋 정도를 계산하는 함수로, 가용 공구수의 초과량으로 나타낸다. 예를 들어, 공구  $t$ 의 가용 공구수가 7개인데 위치  $(a, b)$ 에 있는 개체에 의해 할당된 공구  $t$ 의 개수가 10개라면,  $TP(t)$ 는 3의 값을 갖는다. 만약, 가용 공구수 제약을 초과하지 않으면 함수  $TP(t)$ 는 0의 값을 갖는다. 함수  $MP(m)$ 은 기계  $m$ 에 장착된 공구매거진의 한계용량제약을 어긋 정도를 표현한 것이며, 함수  $TP(t)$ 와 같은 방법으로 계산한다. 위 식에서 벌금계수  $c_1, c_2, \alpha, \beta$ 는 벌금의 양을 조절하는 파라미터이다. 벌금은 가

능해와 비가능해가 적절하게 차별화될 수 있도록 적응도에 부과되어야 한다. 이를 위하여, 실험을 통하여 벌금계수들의 값을 적절히 결정하였다. 본 연구에서 다루는 문제는 최소화 문제이므로, 개체의 적응도는 식 (1)의 값이 낮을수록 높은 적응도를 부여받을 수 있도록 정규화하여 사용하였다(Kim *et al.*, 2003a).

## 5. 실험 및 분석

### 5.1 실험설계

FMS 유연성은 본 연구에서 해결하려는 문제의 해공간 크기를 결정하는 중요한 요소이다. 따라서, AMSEA의 특성 및 성능을 합리적으로 평가하기 위한 실험문제는 각 유연성이 다양한 수준을 갖도록 설계하는 것이 중요하다. 따라서, 4가지 유연성의 정도를 판단하는 척도를 제안하고 이 척도에 따라 다양한 실험문제를 생성하였다. 본 연구에서는 단순한 척도가 사용된다. 기계유연성(MF)의 평가척도는 작업당 평균 대안기계수로 하였다. 즉, 하나의 부품을 생산하기 위한 모든 작업의 대안기계수 총합을 작업수로 나눈 값을 사용한다. 공구유연성(TF)은 작업당 평균 대안공구수로 하였고, 기계유연성과 유사한 방법으로 계산하였다. 공정유연성(PF)은 대안공정들로 인한 가능한 가지의 수로 평가하였다. 순서유연성(SF)은 다음과 같이 계산하였다. 먼저, 한 부품의 생산을 위해 수행되어야 할 최대 작업수( $mo$ )와 그 작업들로 이루어진 집합 B를 구한다. 다음으로, 집합 B에 있는 각 작업의 후행 작업수를 구하고, 이들의 합을 S로 둔다. 이로부터, 순서유연성은  $1 - \{S / (mo(mo-1)/2)\}$ 로 계산한다. 이는 조립라인 밸런싱(assembly line balancing)에서 선행관계의 유연성을 평가할 때 종종 사용되는 방법으로, 구체적인 절차는 Kim *et al.*(2003b)의 연구를 참조하기 바란다.

실험문제는 <Table 1>과 같이 다양한 수준의 유연성을 갖는 18개의 부품들을 조합하여 생성하였다. 본 논문에서는 지면

Table 1. Degrees of flexibility for each part

Part No	TO	Degrees of flexibility				Part No	TO	Degrees of flexibility			
		MF	TF	PF	SF			MF	TF	PF	SF
1	12	5.38	6.88	4	0.58	10	11	4.91	2.18	2	0.39
2	14	5.00	4.86	2	0.61	11	15	9.78	3.22	3	0.80
3	19	6.68	4.58	4	0.84	12	18	3.78	2.22	2	0.74
4	16	3.94	1.94	4	0.38	13	18	6.50	3.33	5	0.33
5	18	6.17	2.06	3	0.51	14	13	5.85	4.00	4	0.58
6	20	4.65	2.35	3	0.72	15	15	3.33	3.40	4	0.73
7	21	5.57	2.86	9	0.33	16	21	6.24	4.76	6	0.38
8	20	3.50	3.05	10	0.41	17	22	2.86	4.55	12	0.69
9	20	6.30	3.20	8	0.72	18	17	5.12	4.82	8	0.72

계약상 부품의 네트워크를 보이진 않았으며, 구체적인 부품들의 네트워크 표현은 Kim(2002)에서 참조할 수 있다. <Table 1>에서 'TO'는 각 부품의 대안공정의 작업들을 포함한 총 작업수를 의미한다. <Table 1>에서 보인 18개의 부품들을 적절히 조합하여 <Table 2>와 같이 18개의 실험문제를 구성하였다. 실험문제들은 문제크기(부품 및 작업수)와 유연성 측면에서 다양하게 설계되었다.

또한, 총 가용기계는 10대, 총 가용공구는 20개의 유형을 갖

는 것으로 가정하였다. 각 기계별 공구매거진의 한계용량, 각 공구유형별 가용공구수 및 공구당 슬롯 요구량은 <Table 3>과 같이 사용하였다. 이때, 각 공구유형별 가용 공구수는 다양한 값을 사용하여 제약의 강약이 표현될 수 있도록 하였다.

모든 알고리즘은 C++ 프로그램 언어로 구현되었으며, 1.7GHz Pentium CPU를 장착한 IBM-PC에서 수행되었다. 다른 파라미터는 예비실험을 통하여 비교적 우수한 성능을 보인 것으로 결정하였다. 식 (1)의 벌금계수  $c_1, c_2$ 는 모두 10으로 두었

**Table 2.** Test problems

Problem No	No. of parts	No. of total operations	Part Number
Prm01	6	89	1, 2, 3, 10, 11, 12
Prm02	6	121	7, 8, 9, 16, 17, 18
Prm03	6	111	3, 5, 9, 11, 13, 16
Prm04	6	76	7, 8, 9, 13, 14, 15
Prm05	6	105	1, 2, 3, 16, 17, 18
Prm06	9	142	1, 2, 3, 5, 6, 10, 11, 12, 15
Prm07	9	168	4, 7, 8, 9, 13, 14, 16, 17, 18
Prm08	9	160	2, 3, 6, 9, 11, 12, 15, 17, 18
Prm09	9	159	4, 5, 6, 7, 8, 9, 10, 11, 12
Prm10	9	151	1, 2, 3, 13, 14, 15, 16, 17, 18
Prm11	12	189	1, 2, 3, 4, 5, 6, 10, 11, 12, 13, 14, 15
Prm12	12	221	4, 5, 6, 7, 8, 9, 13, 14, 15, 16, 17, 18
Prm13	12	211	2, 3, 5, 6, 8, 9, 11, 12, 14, 15, 17, 18
Prm14	12	199	1, 2, 4, 6, 7, 8, 10, 12, 14, 15, 17, 18
Prm15	12	212	1, 2, 3, 7, 8, 9, 13, 14, 15, 16, 17, 18
Prm16	15	262	2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18
Prm17	15	266	1, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18
Prm18	18	310	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18

**Table 3.** Tool and tool magazine constraints

Machine		Tool			
No	Tool magazine capacity	Type	No. of available copies (required slots)	Type	No. of available copies (required slots)
1	28	1	6(1)	11	7(1)
2	31	2	7(2)	12	10(2)
3	38	3	6(2)	13	9(3)
4	28	4	6(2)	14	8(2)
5	36	5	10(1)	15	6(2)
6	37	6	6(2)	16	10(2)
7	29	7	10(3)	17	9(1)
8	28	8	9(2)	18	6(2)
9	29	9	7(1)	19	8(2)
10	26	10	5(2)	20	7(3)

고,  $\alpha$ ,  $\beta$ 는 모두 0.5로 두었다. 모집단의 크기는 100(10×10의 격자구조)으로 하였다. 단계 5.5에서 요구되는 한 이웃의 진화 종료조건은 0.5의 확률로 2회 또는 3회 반복하였다. 돌연변이율과 관련하여, 공정계획문제와 관련된 모든 개체들에 대하여 개체 돌연변이율은 0.15, 인자 돌연변이율은 0.3으로 하였다. 단, 공정모집단의 개체길이는 다른 모집단의 개체길이에 비해 비교적 짧으므로, 공정모집단의 인자 돌연변이율은 0.05로 하였다. 일정계획문제와 관련된 모든 개체들에 대해서는 개체 및 인자 돌연변이율을 각각 0.05와 0.2로 두었다. 알고리즘의 종료조건은 각 모집단들의 개체 평가횟수로 하였다. 각 실험 문제별로 Prm01~Prm05는 7.0E+03, Prm06~Prm10은 1.0E+04, Prm11~Prm15는 1.1E+04, Prm16~Prm18은 1.3E+04일 때 알고리즘을 종료하였다. 각 문제의 종료조건은 해공간의 차이를 반영하기 위하여 달리 두었다.

### 5.2 알고리즘의 성능비교

AMSEA의 탐색성능은 다음 세 가지 알고리즘과 비교하였다. 첫 번째 기법은 계층적 진화알고리즘(Hierarchical Evolutionary Algorithm; HEA)이다. HEA는 먼저 공정계획의 해를 구하고, 이 해를 입력자료로 하여 일정계획문제를 해결한다. 공정계획문제에서 목적함수는 기계부하량의 절대편차 최소화 로하였고, 일정계획문제의 목적함수는 총 처리시간의 최소화

로 하였다. 공정계획문제는 계층 2에 존재하는 부품별 모집단 들로 구성하여, 단일계층을 갖는 일반적인 공생 진화알고리즘 을 기반으로 해결하였고, 일정계획문제는 하나의 모집단을 운영 하는 진화알고리즘으로 해결하였다. 각 문제를 위한 유전요소 들은 제4장에서 설명한 것들을 그대로 사용하였다. HEA의 종료조건은 두 문제에 각각 절반의 종료조건을 사용하였다. 두 번째 기법은 하나의 개체에 표현하고자 하는 모든 정보를 표현하고, 단일 모집단을 운영하는 TEA(Traditional Evolutionary Algorithm)이다. TEA는 AMSEA에서 최상위 계층(계층 4)만을 운영하는 것과 같다. 이 기법은 개체를 다중 염색체 형태로 표현 한 것으로, 복수개의 특성을 갖는 문제를 해결할 때 사용하는 일반적인 방법이다. 세 번째 기법은 Kim *et al.*(2003b)이 제안한 SEA(Symbiotic Evolutionary Algorithm)이다. SEA는 부분 문제 또는 문제의 특성에 따라 복수개의 모집단을 구성하지만, 이들이 단일계층을 이루고 있는 형태이다. SEA는 제안한 알고리즘에서 계층 1만을 운영하여 사용하였다. SEA에 대한 구체적인 절차는 Kim *et al.*(2003b)의 연구를 참조하기 바란다. 비교 알고리즘에서 사용된 진화전략, 파라미터 및 유전요소들은 AMSEA의 파라미터 및 요소들과 동등한 조건으로 설정하였다.

<Table 4>는 HEA, TEA, SEA, AMSEA에 의해 구한 각 문제 의 총 처리시간을 나타낸 것이다. 아래 표에서 ‘Best’, ‘Mean’, ‘Std’는 각각 실험문제마다 10회 반복실험을 통해 얻은 값에

**Table 4.** Performance comparison of the AMSEA and other algorithms

Problem	Evolutionary algorithms												Improved rate(%) [Min,Max]
	HEA			TEA			SEA			ASMEA			
	Best	Mean	s.d.	Best	Mean	s.d.	Best	Mean	s.d.	Best	Mean	s.d.	
Prm01	500.0	506.7	11.7	284.0	318.5	26.8	282.0	306.0	17.0	280.0	292.2*	9.8	[ 4.5, 75.5]
Prm02	385.0	391.0	8.0	273.0	302.0	24.1	236.0	280.5	25.9	234.0	246.6*	7.1	[12.1, 52.9]
Prm03	381.0	382.0	2.1	268.0	316.9	24.7	267.0	306.9	27.5	278.0	291.9*	8.9	[ 4.9, 33.6]
Prm04	317.0	319.5	4.2	257.0	280.7	19.8	224.0	274.0	24.3	230.0	243.3*	12.1	[11.2, 29.6]
Prm05	342.0	349.0	9.9	299.0	314.0	15.2	264.0	310.6	33.6	269.0	286.3*	12.9	[ 7.8, 21.0]
Prm06	432.0	441.8	8.4	377.0	429.3	30.6	368.0	418.4	30.0	365.0	379.6*	10.5	[ 9.3, 16.5]
Prm07	362.0	394.2	21.0	358.0	379.0	9.0	323.0	361.2	18.8	309.0	335.1*	16.5	[ 7.2, 16.5]
Prm08	438.0	480.2	21.3	436.0	456.9	18.9	426.0	435.5	9.3	397.0	421.9*	14.5	[ 3.1, 13.4]
Prm09	400.0	428.6	21.9	390.0	413.1	16.0	384.0	405.1	8.4	370.0	383.2*	7.5	[ 5.4, 11.6]
Prm10	380.0	384.7	4.1	340.0	380.2	30.4	336.0	363.0	16.8	335.0	343.1*	7.3	[ 5.5, 12.2]
Prm11	548.0	549.6	2.6	482.0	526.7	21.5	478.0	509.6	27.0	448.0	465.1*	9.8	[ 8.7, 17.5]
Prm12	485.0	514.1	17.5	455.0	477.3	13.0	446.0	457.1	9.3	407.0	426.5*	12.4	[ 6.7, 19.3]
Prm13	534.0	587.5	27.3	524.0	550.7	23.3	495.0	503.1	5.1	484.0	500.2*	7.4	[ 0.6, 16.7]
Prm14	482.0	531.3	31.2	448.0	504.0	28.5	450.0	457.8	7.0	422.0	448.7*	13.7	[ 2.0, 18.4]
Prm15	487.0	518.2	30.7	477.0	496.2	8.3	467.0	487.7	10.3	415.0	440.3*	14.4	[ 9.7, 16.3]
Prm16	611.0	638.1	21.7	607.0	639.4	21.9	585.0	603.7	14.8	554.0	580.2*	13.7	[ 3.9, 9.5]
Prm17	573.0	602.1	11.6	563.0	601.8	42.7	530.0	553.4	11.7	499.0	526.6*	13.0	[ 4.8, 13.4]
Prm18	714.0	753.0	32.7	691.0	752.1	36.5	676.0	699.7	10.8	639.0	669.2*	19.2	[ 4.4, 12.1]

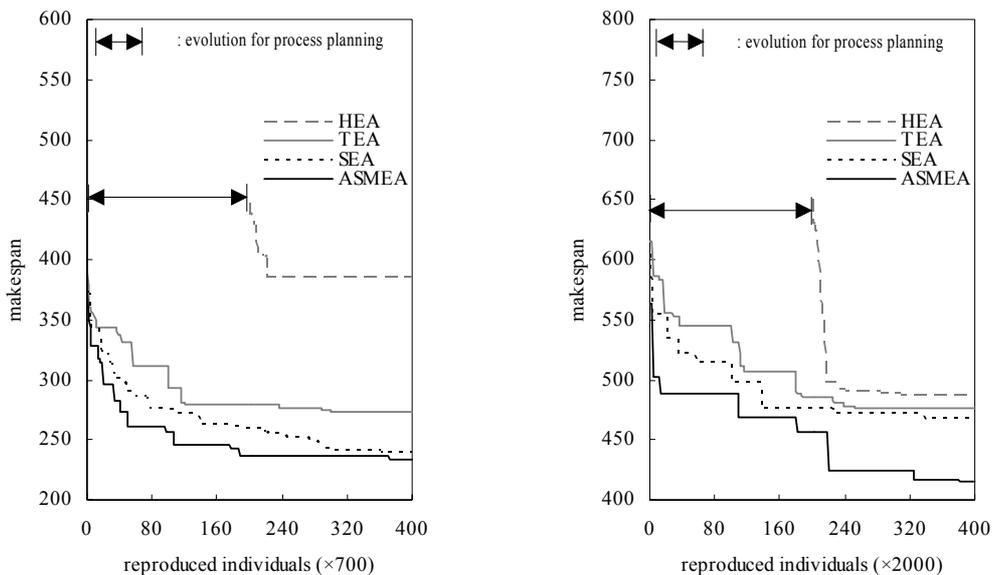
대한 가장 좋은 값, 평균값, 그리고 표준편차를 의미한다. 마지막 열은 각 문제마다 AMSEA와 비교되는 세 알고리즘에 대한 개선율의 최소값과 최대값을 보인 것이다. 예를 들어, Prm01에서 개선율의 최소값은 AMSEA를 제외하고 가장 높은 성능을 보인 것이 SEA이므로  $\{(SEA의\ mean - AMSEA의\ mean) / SEA의\ mean\} \times 100(\%)$ 으로 계산하였고, 최대값은 가장 낮은 성능을 보인 HEA를 기준으로 같은 방법에 의하여 계산하였다. 그리고 아래 표에서 ‘\*’는 각 문제에서 평균값을 기준으로 가장 높은 성능을 보인 결과를 표시한 것이다.

실험결과, 모든 실험문제에서 AMSEA가 가장 좋은 성능을 보였고, 그 다음으로 SEA가 좋은 결과를 보였으며, TEA와 HEA의 순으로 낮은 성능을 보였다. 또한, 반복실험에 대한 표준편차 측면에서도 AMSEA는 다른 알고리즘들에 비해 높지 않은 수준을 보임으로써 반복실험의 과정에서 안정적으로 해를 탐색해 나갈 수 있다. HEA가 해의 질 측면에서 가장 좋지 않은 결과를 보인 것은 통합적 접근방법과 비교할 때, 순차적 접근이 갖는 탐색의 한계를 보인 것이라고 말할 수 있다. 한편, TEA는 HEA에 비하여 좀더 많은 공정계획을 고려할 수 있으므로, HEA보다는 좋은 결과를 보였다. 그러나 TEA는 개체가 다중 염색체 형태로 표현되며 단일 모집단 내에서 운영된다. 이것은 SEA나 AMSEA에서와 같이 복수 개의 모집단을 운영하고, 이들 모집단 간의 여러 결합이 허용되는 것보다는 좋지 않은 결과를 보인 것으로 나타났다.

한편, SEA는 통합문제를 단일계층으로 표현하였고, AMSEA는 다계층으로 표현하였다. 문제의 정보를 다양하게 표현했다는 것은 해공간의 탐색을 다양하게 할 수 있다는 것을 의미할 수 있다. 또한, AMSEA는 SEA와 달리 진화과정 동안 하위계

층에서 발견된 좋은 해들이 선별적으로 상위계층의 모집단으로 이동한다. 그리고 이들은 동일 계층 내에서 또다시 진화하므로, 좋은 해로의 탐색을 촉진할 수 있다. 이는 AMSEA의 기본 프레임이 되는 내공생 진화알고리즘의 개념이기도 하며, Kim *et al.*(2001)의 연구에서도 이러한 상호작용이 SEA보다 우수한 성능을 보임을 확인할 수 있다. 본 연구에서 제안한 내공생 진화모형은 계층 내에서의 진화와 계층 간의 내공생 진화에 의해, 빌딩블록(building blocks)들을 생성하고 유지하며, 이들을 결합시켜 더 좋은 해로의 탐색을 유도한다고 볼 수 있다. AMSEA가 갖는 탐색의 우수성은 부품수, 유연성의 종류나 정도 측면에서도 크게 영향을 받지 않은 것으로 나타났다.

AMSEA는 다른 알고리즘에 비해 약간 많은 계산소요시간을 요구하였다. AMSEA는 SEA보다 많은 모집단과 계층을 운영하고, 추가되는 모집단과 계층별로 유전연산과 진화를 위한 계산시간이 요구된다. 또한, AMSEA는 TEA보다 SEA와의 비교에서 불리한 요소들과 함께 개체의 적응도 평가 시 공생 파트너를 선택하는 데 필요한 계산시간이 요구된다. AMSEA의 계산시간을 평균적으로 보면, Prm01~Prm05는 63초, Prm06~Prm10은 225초, Prm11~Prm15는 437초, Prm16~Prm18은 1,676초 정도 소요되었다. SEA는 AMSEA보다 15%정도 덜 소요되었고, TEA는 AMSEA보다 25% 정도, HEA는 AMSEA보다 30% 정도 덜 소요되었다. 그러나 <Table 4>의 마지막 열에서 보듯이, 본 연구에서 다루는 FMS 통합생산계획문제와 같이 복잡도가 높은 문제에서는 높은 수준의 개선율이 계산시간의 증가에 따른 불리함을 상쇄시키는 효과로 제시될 수 있을 것이다. 또한, AMSEA는 진화의 초반부터 다른 알고리즘보다 우수한 성능을 보이므로, 계산시간의 부담은 종료조건의 조절



(a) Prm02

(b) Prm15

Figure 6. Comparison of search capability and evolution speed.

을 통하여 감소시킬 수 있다. 이는 알고리즘별 성능 및 수렴성을 보인 <Figure 6>을 통해 확인할 수 있다. 다른 문제에서도 이와 유사한 결과들을 보였다.

### 5.3 비대칭 다계층 통합문제의 진화메커니즘

비대칭 다계층 통합문제에서, 좋은 해를 효율적으로 탐색하기 위해서는 상대적으로 계층깊이가 낮은 부분문제의 진화와 계층 깊이가 깊은 부분문제의 적응도 평가를 위한 적절한 메커니즘이 요구된다. 이 절에서는 AMSEA의 설계 시 고려했던 진화메커니즘들을 보이고, 이들이 알고리즘 성능에 미치는 영향을 분석한 실험 결과들을 제시하고자 한다.

본 연구에서는 세 가지 형태의 진화메커니즘들을 고려해 보았다. 이를 도식적으로 나타내면 <Figure 7>과 같다. <Figure 7> (a)의 진화메커니즘 1은 계층 깊이가 낮은 일정계획 모집단에 대하여 각각 계층 1과 2에 계층 3의 일정계획 모집단과 동일한 크기와 형태의 가상 모집단(Dummy Sch.)을 두고, 각 모집단들은 계층별로 계층 내에서 진화하고 계층 간에는 하위 계층에서 발견된 좋은 해가 직상위 계층으로 이동하는 메커니즘이다. 이때, 계층 1과 계층 2의 가상 일정계획은 각각 계층 3에 존재하는 일정계획 모집단의 개체 생성방법과 동일하게 임의로 생성한다. <Figure 7(b)>의 진화메커니즘 2는 별도로 가상 모집단을 두지 않고, 계층 1과 2의 적응도 평가 시 계층 3의 일정계획 모집단에서 공생 파트너를 제공한다. 또한, 계층 1과 계층 2의 진화 시에는 동일한 기회로 계층 3의 일정계획 모집단을 진화시키는 메커니즘이다. 즉, 하나의 일정계획 모집단이 계층 위치를 유연하게 이동해 가며 계층 1, 2, 3에서의 적응도 평가 및 진화에 참여하는 것이다. <Figure 7(c)>의 진화메커니즘 3은 일정계획 모집단에서 공생 파트너만을 제공하고, 일정계획 모집단은 계층 1과 계층 2가 진화할 때에는 진화하지 않고 계층 3이 진화할 때에만 진화하는 메커니즘이다. 이들은 모든 계층에서 완전해의 구성이 용이하여 적응도 평가가 손쉽게 이루어질 수 있다는 공통점을 갖지만, 계층깊이가 낮은 일정계획 문제에 대한 가상 모집단의 구성이나 진화기회 측면에서는 차이점을 갖는다.

성능실험을 위하여, 각 메커니즘은 다음과 같이 전개된다.

진화메커니즘 1을 채택하면 3.3절의 AMSEA가 된다. 다른 두 메커니즘은 AMSEA를 적절히 수정하여 구현하였다. 진화메커니즘 2를 위하여 AMSEA의 단계 1에서는 일정계획의 하위 계층 1, 2에 비어 있는 모집단을 만든다. 단계 2에서 계층 3의 일정계획 모집단을 위한 개체를 생성한 후, 계층 1과 2에 이 개체들을 복사한다. 계층 1, 2, 3의 일정계획 모집단은 모두 같은 계층 간 적응도 비교는 의미가 없으므로, 단계 4.2에서 일정계획의 계층 1과 2의 적응도 평가 시에는 계층 간 적응도 비교가 생략된다. 단계 5에서는 계층 1과 2의 진화 이후마다 일정계획 가지에 있는 해당 계층의 모집단이 직상위 계층으로 복사되도록 하였다. 진화메커니즘 3은 AMSEA의 단계 1을 생략하고, 단계 4.2에서 현재 적응도를 평가하는 계층과 무관하게 계층 3의 일정계획 모집단에서 항상 공생 파트너를 선택하도록 하였다. 또한 진화메커니즘 3에서도 계층 1과 2를 위한 단계 4.2의 상하위 계층 간 적응도 비교는 생략된다.

<Figure 8>은 이들 3가지 메커니즘에 의해 구한 알고리즘의 성능 결과를 보인 것이다. 실험문제에서 각 부품수에 따라 그 경향이 유사하여, <Figure 8>에서는 대표적인 문제들의 결과만을 보이기로 한다. 이 그림은 각 메커니즘마다 10회 반복실험의 평균값을 막대 그래프로 보인 것이고, 10회 반복실험 중 가장 좋은 해와 가장 나쁜 해를 편차의 개념으로 하여 오차선으로 보였다.

실험결과는 다음 몇 가지로 요약될 수 있다. 첫째, 부품수가 상대적으로 적은 문제(Prm04 ~ Prm10)에서는 각 메커니즘의 실험결과들이 서로의 편차범위에 속하여 성능 차이가 크게 존재한다고 말하기 어렵다. 그러나 부품의 수가 증가할수록 진화메커니즘 1의 평균값은 다른 두 메커니즘들의 값보다 점점 더 낮아졌으며, 진화메커니즘 1의 가장 나쁜 해도 다른 진화메커니즘의 가장 좋은 해보다 더 낮은 값을 보였다. 따라서 문제 크기가 커질수록 진화메커니즘 1이 다른 메커니즘들에 비해 더욱 우수하다고 말할 수 있다.

둘째, 진화메커니즘 1과 진화메커니즘 2의 성능 차이를 분석해 보면, 두 메커니즘에서 일정계획 모집단은 진화의 기회 또는 정도가 동일하므로, 부분문제들이 각각의 계층깊이에 큰 영향을 받지 않고 비교적 균등하게 진화의 기회를 가질 수 있다. 그러나 진화메커니즘 1은 진화메커니즘 2와 달리 계층 1,

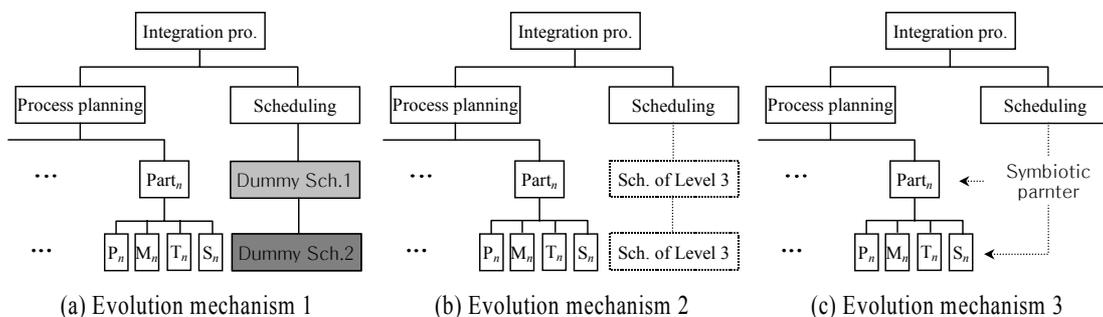


Figure 7. Evolution mechanisms for the asymmetric multileveled integrated problem.

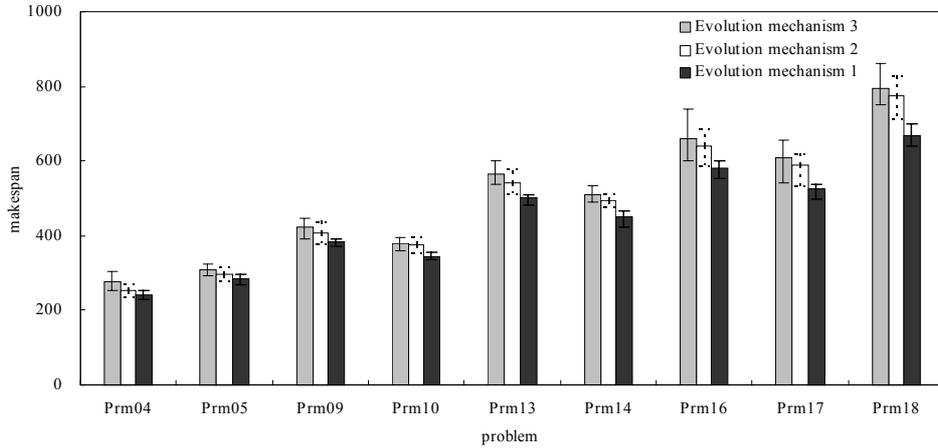


Figure 8. Performance comparison of evolution mechanisms .

2, 3에 서로 다른 일정계획의 개체들이 존재하므로, 다양성 측면에서 유리할 수 있다. 또한, 진화메커니즘 1은 계층간 상호작용으로 인해 하위계층보다 상위계층에 적응도가 높은 개체들이 존재하게 되어, 계층별로 해의 탐색방향을 달리 형성하는 데 유리한 것으로 판단된다. 이와 같은 진화메커니즘 1의 특징이 좋은 해로의 탐색에 영향을 미친다고 말할 수 있다.

셋째, 예상대로 일정계획 모집단의 진화기회가 다른 것들에 비해 적은 진화메커니즘 3이 가장 낮은 성능을 보였다. 진화메커니즘 3이 가장 낮은 성능을 보임에 따라 일정계획 모집단의 진화도 FMS 통합문제의 해에 상당한 영향을 미침을 알 수 있다. 일정계획문제는 공정계획문제에 비하여 계층수준이 낮지만, 어떤 부분문제의 계층수준이 낮다고 하여 문제의 복잡도가 낮다고 말할 수는 없다. 결국, 각 문제가 갖는 복잡도와 해에 대한 영향력을 고려하여 진화기회를 부여하는 것이 요구된다. AMSEA에서는 FMS 통합문제에 대한 이상의 실험결과를 바탕으로 하여, 진화메커니즘 1을 채택하였다.

## 6. 결론 및 토의

본 연구에서는 FMS 공정계획과 일정계획의 통합적 해결을 위한 새로운 방법론으로 문제구조를 해법에 반영한 AMSEA를 제안하였다. FMS 통합문제에서는 기계, 공구, 공정, 순서유연성을 고려하였고, 유연성으로 인해 형성되는 가능한 경우의 수를 모두 고려하였다. 실험결과, 다양한 실험문제에서 제안한 알고리즘은 기존 알고리즘들보다 우수한 성능을 보였다. 특히, AMSEA는 유연성의 종류나 정도 측면에서도 크게 영향을 받지 않고 좋은 성능을 보였으며, 실험문제의 해공간이 커지더라도 다른 알고리즘과의 성능 차이에 큰 변화를 보이지는 않았다.

AMSEA는 다음 몇 가지 특징들을 갖는다. 첫째, 계층별로

다른 개체표현을 사용한다. 이는 다양한 길이를 갖는 부분해와 이들의 결합된 해들을 하나의 알고리즘에 공존시켜 탐색의 다양성을 기대하기 위함이다. 둘째, 하위계층의 적응도 평가 과정에서 발견된 높은 적응도를 갖는 개체결합을 상위계층으로 이동시키고, 이들이 자신이 속한 계층에서 더 좋은 개체로 진화할 기회를 갖는 계층 간 상호작용이 이루어진다. 셋째, 다양한 해공간의 탐색에 유리한 이웃진화전략을 사용하였다. 넷째, 비대칭형 다계층 통합문제를 적절히 해결하기 위하여 계층깊이가 낮은 부분문제에 가상 모집단을 두는 진화 메커니즘을 갖는다. 다섯째, AMSEA는 FMS의 유연성 확장 또는 FMS의 다양한 생산계획의 통합문제를 해결하는 데 손쉽게 적용될 수 있다. 또한, 제안한 알고리즘은 복잡한 문제구조를 계층화하여 다루고자 하는 문제를 효과적으로 해결할 수 있으며, 계층의 깊이나 모집단수, 그리고 대칭형과 비대칭형에 제한을 받지 않고 다양한 문제에 적용 가능하다.

일반적으로 통합문제에서 각 부분문제는 해공간의 상대적 크기, 통합해에 대한 영향력, 그리고 해의 탐색노력 등에 있어 차이를 가질 것으로 예상된다. 또한, 계층구조상에서 보인 부분문제의 계층깊이가 이들을 결정하는 절대적 기준일 수는 없다. 따라서, 각 부분문제의 진화가 통합의 관점에서 적절히 균형을 갖도록 하는 것이 중요하다. 이에 따라 본 연구에서는 FMS의 통합문제를 예제로 하여 AMSEA를 설계할 때, 이를 반영하는 진화메커니즘을 설계 및 제안하고자 하였다. 각 부분문제 간 조화로운 진화가 이루어 질 수 있는 메커니즘에 대한 연구가 앞으로 더 깊이있게 이루어져야 할 것으로 보인다.

이와 함께, AMSEA의 계산소요시간 단축을 위해 진화전략, 계층 간 또는 계층 내의 상호작용에 관하여 추후 연구가 요구된다. 그리고 FMS 공정계획 또는 일정계획을 포함한 다른 생산계획문제와의 통합적 해결도 중요한 연구주제가 되며, FMS 유연성의 수준에 따라 해의 질에 영향을 미치는 요소들에 관한 연구분석도 수행될 수 있다.

## 참고문헌

- Alexander, W., Denis, C., Thom, H., Russell, K., James, W., and Andrew, Z. (1999), Scheduling with alternatives: a link between process planning and scheduling, *IIE Transactions*, **31**, 1093 - 1102.
- Bierwirth, C. and Mattfeld, D.C. (1999), Production scheduling and rescheduling with genetic algorithms, *Evolutionary Computation*, **7**, 1-17.
- Campbell, N.A., Mitchell, L.G., and Reece, J.B. (1996), *Biology: Concepts & Connections*, Second Edition, Benjamin/Cummings Publishing Company Inc.: Redwood City, CA.
- Capra, F., (1996), *The web of life*, Anchor Books: New York.
- Guerrero, F., Lozano, S., Koltai, T. and Larraneta, J. (1999), Machine loading and part type selection in flexible manufacturing systems, *International Journal of Production Research*, **37**, 1303-1317.
- Ho, Y.C. and Moodie, C.L. (1996), Solving cell formation problems in a manufacturing environment with flexible processing and routing capabilities, *International Journal of Production Research*, **34**, 2901-2923.
- Kim, J.Y., Kim, Y.H. and Kim, Y.K. (2001), An endosymbiotic evolutionary algorithm for optimization, *Applied Intelligence*, **15**, 117-130.
- Kim, Y.K. (2002), A set of data for the integration of process planning and scheduling in FMS, available at <http://syslab.chonnam.ac.kr/links/FMSdata-pp&s.doc>.
- Kim, Y.K., Kim, J.Y. and Kim, Y. (2000), A coevolutionary algorithm for balancing and sequencing in mixed model assembly lines, *Applied Intelligence*, **13**, 247-258.
- Kim, Y.K., Kim, J.Y. and Lee, W.K. (2003a), A multi-level symbiotic evolutionary algorithm for FMS loading problems with various flexibilities, *Journal of the Korean Institute of Industrial Engineers*, **29**(1), 66-78.
- Kim, Y.K. and Min, Y.J. (2003), An integrated planning of production and distribution in supply chain management using a multi-level symbiotic evolutionary algorithm, *Journal of the Korean Operations Research and Management Science Society*, **28**(2), 1-15.
- Kim, Y.K., Park, K. and Ko, J. (2003b), A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling, *Computers & Operations Research*, **30**, 1151-1171.
- Kumar, N. and Shanker, K. (2000), A genetic algorithm for FMS part type selection and machine loading, *International Journal of Production Research*, **38**, 3861-3887.
- Liu, J. and Maccarthy, B.L. (1997), A global MILP model for FMS scheduling, *European Journal of Operational Research*, **100**, 441-453.
- Margulis, L. (1970), *Origin of eukaryotic cells*, Yale University Press: New Haven.
- Modi, B.K. and Shanker, K. (1994), Models and solution approaches for part movement minimization and load balancing in FMS with machine, tool and process plan flexibilities, *International Journal of Production Research*, **33**, 1791-1816.
- Morad, N. and Zalzal, A. (1999), Genetic algorithms in integrated process planning and scheduling, *Journal of Intelligent Manufacturing*, **10**, 169 - 179.
- Paulli, J. (1995), A hierarchical approach for the FMS scheduling problem, *European Journal of Operational Research*, **86**, 32-42.
- Potter, M.A. (1997), The design and analysis of a computational model of cooperative coevolution, Ph.D. dissertation, George Mason University.
- Rachamadugu, R. and Stecke, K.E., (1994), Classification and review of FMS scheduling procedures, *Production Planning and Control*, **5**, 2-20.
- Sabuncuoglu, I. (1998), A study of scheduling rules of flexible manufacturing systems: a simulation approach, *International Journal of Production Research*, **36**, 527-546.
- Saygin, C. and Kilic, S.E. (1999), Integrating Flexible Process Plans with Scheduling in Flexible Manufacturing systems, *International Journal of Advanced Manufacturing Technology*, **15**, 268-280.
- Stecke, K.E. and Raman N. (1995), FMS planning decisions, operating flexibilities, and system performance, *IEEE Transactions on Engineering Management*, **42**, 82-90.
- Syswerda, G., (1991), A study of reproduction in generational and steady-state genetic algorithms, *Foundations of Genetic Algorithms*, San Mateo, CA, pp.94-101.