

## 유연제조시스템의 공정계획을 위한 다목적 진화알고리즘\*

김여근\*\* · 신경석\*\* · 김재윤\*\*\*

### A multiobjective evolutionary algorithm for the process planning of flexible manufacturing systems\*

Yeo Keun Kim\*\* · Kyoung Seok Shin\*\* · Jae Yun Kim\*\*\*

#### ■ Abstract ■

This paper deals with the process planning of flexible manufacturing systems (FMS) with various flexibilities and multiple objectives. The consideration of the manufacturing flexibility is crucial for the efficient utilization of FMS. The machine, tool, sequence, and process flexibilities are considered in this research. The flexibilities cause to increase the problem complexity. To solve the process planning problem, in this paper an evolutionary algorithm is used as a methodology. The algorithm is named multiobjective competitive evolutionary algorithm (MOCEA), which is developed in this research. The feature of MOCEA is the incorporation of competitive coevolution in the existing multiobjective evolutionary algorithm. In MOCEA competitive coevolution plays a role to encourage population diversity. This results in the improvement of solution quality and, that is, leads to find diverse and good solutions. Good solutions means near or true Pareto optimal solutions. To verify the performance of MOCEA, the extensive experiments are performed with various test-bed problems that have distinct levels of variations in the four kinds of flexibilities. The experiments reveal that MOCEA is a promising approach to the multiobjective process planning of FMS.

Keyword : FMS, Process Planning, Multiobjective Evolutionary Algorithm, Competitive Coevolution

논문접수일 : 2004년 1월 17일    논문게재확정일 : 2004년 4월 3일

\* 이 논문은 2002년도 전남대학교 학술연구비 지원에 의하여 연구되었음.

\*\* 전남대학교 산업공학과

\*\*\* 전남대학교 고품질전기전자부품 및 시스템 연구센터

## 1. 서 론

유연제조시스템(FMS : flexible manufacturing system)은 수치제어 기계들이 네트워크와 자동 물류시스템에 의해 연결되고, 기계에 장착된 공구들은 자동적으로 상호교환이 가능한 자동화된 제조 시스템을 말한다. FMS는 흐름생산(flow shop) 시스템의 생산성과 주문생산(job shop) 시스템의 유연성을 동시에 구현하여 다양해진 사용자의 요구에 부합하고, 수명이 점점 짧아지고 있는 제품을 경제적으로 생산하기 위한 것이다[7, 16]. FMS의 효율적인 운영을 위해서는 해결해야 할 여러 문제들이 있다[24]. 특히 FMS 공정계획문제는 생산설비들의 효율적인 활용을 위해 산업현장에서 중요하게 다루어진다. 본 연구에서는 FMS의 공정계획문제를 다룬다.

FMS의 공정계획문제는 부품의 형상, 원자재, 공차 등의 부품관련 정보와 가공기술적 제약을 포함한 기계관련 정보를 이용하여 구체적인 가공방법 및 가공경로를 결정하는 것이다. 이때 생산시스템의 유연성은 변화하는 시장 환경에 효율적으로 대처하는 데 매우 중요한 요소로 인식되고 있다[16]. 그러나 이 유연성의 증대는 공정계획문제의 복잡도를 증가시킨다. 이러한 복잡도로 인하여 기존의 많은 연구들은 다양한 유연성을 동시에 고려하지 못하고 있다[19, 27]. 본 연구에서는 다양한 유연성을 고려한다.

FMS 공정계획에서 추구하는 목적들은 다양하다. 즉 기계 이용률, 부품 이동, 공구 및 기계변경 등과 관련된 여러 목적들이 있을 수 있다. 이들 목적들은 상호 상충될 수 있다. 이러한 상황에서 의사결정자에게 다양한 파레토 최적해(Pareto optimal solutions)를 제안하고, 이들 중에서 상황에 가장 적합한(의사 결정자가 가장 선호하는) 대안을 선택하게 하는 것도 하나의 방법이다. 파레토 최적해는 모든 목적들에 대해 이 해보다 더 우수한 해가 존재하지 않는 해이다. 파레토 최적해를

비지배해(nondominated solution)라 부르기도 한다. 해공간이 넓은 조합최적화문제에서는 파레토 최적해를 구하기가 쉽지 않다. 본 연구에서는 용어의 혼동을 피하기 위하여 앞으로 파레토 최적해는 이론적인 실제 파레토 최적해를 의미하고, 비지배해는 구한 여러 대안들 중에서 지배당하지 않는 해를 나타내기로 한다. 본 연구에서는 여러 유연성을 고려한 다목적 FMS 공정계획에서 다양하고 좋은 계획들을 찾는 방법론을 제시한다. 방법론으로는 진화알고리즘(evolutionary algorithm)을 채용한다.

진화알고리즘은 개체들의 집합인 모집단을 운영함으로써, 즉 여러 해를 운영함으로써 다목적 문제에서 다양한 대안을 동시에 구하는 한 방법론으로 사용될 수 있다. 다목적 문제를 다룬 진화알고리즘을 다목적 진화알고리즘(MultiObjective Evolutionary Algorithm : MOEA)이라 부른다[30]. 다목적 문제에서는 의사결정자의 선호가 다를 수 있어 다양한 대안(해)들을 제안해야 한다. 그리고 이들 해는 가능한 좋은 해이어야 한다. 따라서, 다목적 진화알고리즘은 좋은 해를 탐색할 수 있으면서도 다양한 해를 얻을 수 있도록 구현되어야 한다. 이를 위한 여러 형태의 다목적 진화알고리즘이 제안되었다[26]. 본 연구에서는 종(種)간의 경쟁에 기초한 새로운 다목적 진화알고리즘을 제안한다. 제안한 알고리즘을 FMS 공정계획문제에 적용할 수 있도록 문제에 맞는 여러 유전요소 등을 개발하고 컴퓨터 모의실험을 통해 기존 다목적 진화알고리즘과 성능을 비교 분석한다.

본문의 구성은 다음과 같다. 제2장에서는 본 연구에서 다루는 FMS 공정계획문제를 설명한다. 제3장에서는 다목적 진화알고리즘의 일반론과 제안한 진화알고리즘에 대해 설명한다. 제4장에서는 알고리즘 적용을 위한 유전요소를 설명한다. 그리고 제5장에서는 실험설계와 실험결과를 제시하고, 그 결과에 대해 분석한다. 마지막으로 제6장은 결론으로 본 연구의 내용을 요약 정리한다.

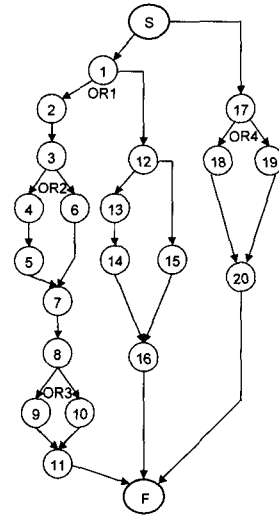
## 2. FMS의 공정계획 문제

### 2.1 다목적 FMS의 공정계획

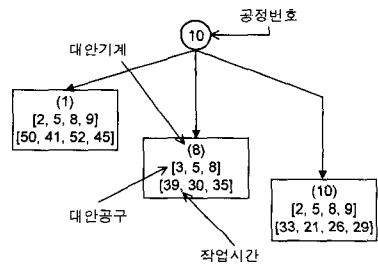
FMS 공정계획문제는 고려하는 유연성의 종류와 범위에 따라 결정변수의 종류와 수, 그리고 문제의 복잡도와 구조 등이 결정된다[7]. FMS의 유연성들에 대한 분류 및 정의는 Steckle and Raman[25]의 연구를 참조할 수 있다. 본 연구에서는 기계유연성, 공구유연성, 순서유연성, 공정유연성의 4가지 유연성을 고려한다. 기계유연성은 대안기계에 의한 작업 실행의 가능성으로 정의한다. 기계유연성은 다양한 형태의 공정을 가공할 수 있는 기계능력으로 인하여 고려된 것이다. 공구유연성은 대안공구에 의한 작업 실행의 가능성 또는 여러 작업유형을 가공할 수 있는 공구의 능력이라고 정의한다. FMS는 다기능성과 자동화된 교환장치를 갖는 머시닝 센터를 사용하므로, 기계유연성과 공구유연성은 중요하게 고려되어야 한다. 특히, 공구유연성을 고려하지 않은 공구할당 결과는 실제 생산현장에 적용하는데 한계를 가질 수 있다[18]. 공정유연성은 하나의 특징형상(feature)을 가공하는데 있어 대안작업, 또는 대안가공경로가 있음을 말한다. 대안가공경로는 기계의 다기능성, 대안기계, 대안공구들로 인하여 다양하게 생성될 수 있다. 마지막으로, 순서유연성은 제조 공정간의 순서가 바뀔 수 있는 가능성을 말한다. 순서유연성을 고려함으로써 병목현상에 의한 자원들의 비효율적인 사용을 방지할 수 있다.

본 연구에서 다루는 유연성을 설명하기 위해 Ho and Moodie[8]에 의해 제안된 네트워크 표현을 사용한다. [그림 1]은 어떤 한 부품의 가공을 위한 유연성을 나타낸 네트워크 표현의 한 예이다. [그림 1](a)에서 보는 바와 같이 네트워크 표현은 시작노드, 중간노드, 끝노드로 구성되어 있다. 시작노드와 끝노드는 가상노드(dummy node)이며, 제조공정에 있어 시작과 끝을 나타낸다. 중간노드에는 작업번호, 각 작업을 수행할 수 있는 대안기

계 및 대안공구, 그리고 이에 대한 가공시간을 포함하고 있다. 중간노드에 포함된 내용은 기계유연성과 공구유연성을 설명할 수 있다.



(a) 부품의 네트워크 표현



(b) 작업의 세부 표현

[그림 1] FMS의 유연성을 위한 네트워크 표현

노드를 연결하는 화살표는 작업간의 선행관계를 나타낸다. 'OR'관계는 대안 가공경로를 나타내기 위해 사용된다. 만약, 노드의 연결이 'OR'기호에 의해 나누어지면 두 경로중 어느 한쪽 경로를 따른다. 이를 통해 공정유연성을 표현할 수 있다. 'OR'기호와 연결되지 않는 노드들은 반드시 수행되어야 함을 뜻하고, 다양한 가공순서를 포함하므로 순서유연성이 표현된다. [그림 1](b)는 (a)의 10번 작업을 확대해 보인 것이다. 10번 작업은 1,

8, 10의 기계 중 어느 하나에서 작업될 수 있고, 기계 1은 공구 2, 5, 8, 9 중 하나를 선택할 수 있다. 이 때 가공시간은 기계와 공구에 따라 다를 수 있다.

## 2.2 문제상황 및 목적함수

FMS는 상황에 따라 다양한 문제가 정의될 수 있다. 본 연구에서 다루는 FMS는 다음과 같은 상황에서 운영된다고 가정한다.

- (1) 생산계획기간(planning period)이 주어져 있다.
- (2) 각 부품 및 지원장치(물류시스템, 팔레트, 치공구 등)의 준비시간과 이동시간은 가공시간에 포함된다.
- (3) 각 부품의 작업들은 하나의 기계에 할당된다. 즉, 작업 분할은 허용되지 않는다.
- (4) 시스템은 머시닝센터로 구성되고, 적절한 공구가 주어지면 다양한 작업을 수행할 수 있다.
- (5) 각 기계의 공구매거진은 용량이 서로 다르고, 그 한계용량은 주어져 있다.
- (6) 한 작업은 여러 형태의 공구에 의해 가공될 수 있다.
- (7) 생산계획기간에 사용될 수 있는 형태별 가용 공구수와 이를 공구매거진에 장착시키기 위한 슬롯 량이 주어져 있다.
- (8) 각 공구 수명은 고려하지 않는다.

본 연구에서 다루는 FMS 공정계획문제는 위에서 언급한 가정하에서, 기계유연성, 공구유연성, 순서유연성, 공정유연성을 동시에 고려하면서 다 목적을 가지고 각 공정에 기계 및 공구를 할당하고, 가공경로 및 가공순서를 결정하는 문제이다. 이 문제에서 추구하는 여러 목적들이 있을 수 있다. 예로, 부하균형화, 총가공시간 최소화, 부품이동 최소화, 준비비용 최소화 등이 있다[7, 27]. 본 연구에서는 고려하는 유연성과 관련이 있으면서 서로 상충적인 관계에 있는 다음 두가지 목적을

사용한다.

목적 1 : 준비비용의 최소화

목적 2 : 기계부하량의 절대편차(absolute deviation of machine workloads : ADMW)의 최소화

준비비용은 작업을 위해 각 기계에서 요구되는 준비 또는 기계내에서 작업의 변경에 따른 공구의 준비와 관련된 비용을 포함한다. 이러한 비용의 최소화는 생산물의 원활한 흐름과 생산시간의 단축에 중요하다[18, 24]. 반면, ADMW의 최소화는 기계 이용률 측면에서 좋은 효과를 보일 수 있다 [1]. 하지만 이들 두 목적은 상충적인 관계에 있을 수 있다. 즉, 준비비용의 최소화를 위한 공정계획은 기계 부하의 불균형을 초래할 수 있고, 반대로 ADMW 최소화를 위한 공정계획은 잦은 기계 변경과 공구 변경으로 인해 준비비용의 증대를 야기할 수 있다.

목적 1의 준비비용(setup cost)은 식 (1)과 같이 표현한다.

$$\begin{aligned} \text{Setup Cost} = & MSC \times \sum_i \sum_j \phi(M_{i(j+1)}, \\ & M_{ij}) + \\ & TSC \times \sum_i \sum_j \{ [1 - \phi(M_{i(j+1)}, \\ & M_{ij})] \times \phi(T_{i(j+1)}, T_{ij}) \} \quad (1) \end{aligned}$$

여기서,

$$\phi(x, y) = \begin{cases} 1, & \text{if } x \neq y \\ 0, & \text{otherwise} \end{cases}$$

여기서, MSC는 기계변경에 따른 단위당 준비비용,  $M_{ij}$ 는 부품  $i$ 의  $j$ 번째 작업을 수행하는 기계번호, TSC는 공구변경에 따른 단위당 준비비용,  $T_{ij}$ 는 부품  $i$ 의  $j$ 번째 작업을 수행하는 공구번호를 각각 나타낸다.

목적 2의 기계부하량 절대편차(ADMW)는 식 (2)와 같다.

$$ADMW = \sum_{m=1}^M |W_m - \bar{W}| \quad (2)$$

여기서,  $W_m$ 은 기계  $m$ 에 할당된 작업량,  $\bar{W}$ 는 기계 부하량의 평균을 각각 나타낸다.  $\bar{W}$ 는 각 기계에 작업을 할당한 후, 현재 작업 할당량의 총합을 할당 가능한 기계수로 나누어 계산한다.

### 3. 다목적 FMS의 공정계획 문제를 위한 진화알고리즘

#### 3.1 다목적 진화알고리즘

다목적 문제를 위한 전통적인 기법들은 하나의 해를 개선시켜 가면서 하나의 비지배해를 찾는다. 알고리즘을 여러 번 수행하여 여러 비지배해를 얻을 수 있다. 그러나 이와 같은 방법은 구한 해들이 해 공간에서 넓게 분포되는, 즉 다양하게 존재하는 것을 보장하지 못한다[2].

진화알고리즘은 자연의 진화과정을 모방한 확률적 탐색기법으로 여러 해(모집단)를 운용하면서 해공간을 탐색한다. 진화알고리즘에서 모집단 운용은 한 번의 알고리즘 실행으로 여러 비지배해를 구할 수 있게 한다. 이러한 이유로 진화알고리즘은 다목적 최적화문제를 해결하는 데 매우 효율적인 기법으로 인식되고 있으며 이에 관한 많은 연구가 이루어져 왔다[26]. 다목적 진화알고리즘(MOEA)의 목표는 파레토 최적해 또는 이들 해에 가까운 해들을 다양하게 찾는 것이다[3]. 따라서 MOEA 연구의 초점은 두가지로, 그 하나는 (1) 파레토 최적해로의 수렴이고, 다른 하나는 (2) 다양한 해를 찾는 것(즉, 모집단의 다양성 유지)에 두었다[3]. MOEA에 관한 최초 알고리즘은 Schaffer[22]에 의해 제시된 VEGA(Vector Evaluated Genetic Algorithm)이다. 이 방법은 각 목적에 부분모집단을 운영하여 각 목적별(부분모집단별) 적응도 평가와 선택(selection), 유전연산을 반복하면서 비

지배해들을 찾아 나간다. 하지만 이 방법은 구한 해가 각 목적에 편중되는 단점을 보였다. 이러한 단점을 보완하기 위해, Goldberg[6]는 적소 메커니즘에 의한 적응도 배분을 제안하였다. 적소 메커니즘은 주위 개체들의 밀집도에 따라 적응도를 다르게 부여한다. 즉 주위 밀집도가 낮은 개체의 적응도를 높게 줌으로써, 모집단의 다양성을 유지하는 방법이다. 이 개념을 바탕으로 한 대표적 알고리즘으로는 MOGA(MultiObjective Genetic Algorithm[5]), NSGA(Non-dominated Sorting Genetic Algorithm[23]), NPGA(Niched Pareto Genetic Algorithm[9]) 등이 있다. 즉, 이들 연구는 적소 개념에 의해 다양한 모집단을 구하고 유지하는 데 중점을 두었다.

한편, 최근에는 엘리티즘(elitism)전략을 이용하여, 다양한 해와 함께 좋은 해로의 수렴속도를 향상시키는 연구가 이루어 졌다[20, 31]. 다목적 진화 알고리즘에서 '엘리티즘전략'이란 지금까지 발견된 비지배해를 보관하고 운용하는 전략을 의미한다. 예로, SPEA(Strength Pareto Evolutionary Algorithm[32]), PAES(Pareto-Archive Evolution Strategy[15]), NSGA-II[4], SPEA2[33] 등이 있다. SPEA는 현재까지 발견된 비지배해들을 외부 모집단(external population)에 유지하는 엘리티즘전략을 사용하였다. 후에 SPEA2에서 외부 모집단 운영방법과 적응도 평가방법을 개선하였다. PAES는 진화전략(Evolution Strategy : ES)을 응용한 방법으로, 하나의 부모 개체로 자손을 생산한 후 생산된 자손을 비지배해 집합(archive)과의 지배관계 및 밀집거리를 비교하면서 비지배해 집합을 갱신하고, 또한 부모개체와 비교하여 자손을 생산할 부모개체를 갱신한다. 이 과정을 통해 수렴성과 다양성을 동시에 추구하는 방법이다. NSGA-II는 NSGA의 많은 계산량에 관한 단점을 보완하고 엘리티즘을 위해 진화전략(ES ; Evolution Strategy[17])의 한 방법인  $(\mu + \lambda)$ -ES를 사용하였다.  $(\mu + \lambda)$ -ES란  $\mu$  개의 개체를

갖는 모집단에서  $\lambda$  개의 자손을 생산하여, 모집단과 생산된 자손 개체 ( $\mu + \lambda$ )개에서 가장 좋은  $\mu$  개의 개체를 선택하여 다음 세대의 모집단을 구성하는 전략이다. NSGA-II에서는 개체의 파레토 순위를 우선적으로 고려하고 순위가 동일하면 밀집도(crowding distance)를 계산하여 밀집도가 낮은 개체가 자손 생산을 위해 선택되거나, 다음 세대 모집단을 구성하는 개체로 선택된다.

대부분의 기존 연구들은 해의 수렴성과 다양성을 높이기 위하여 모집단의 운영과 적응도 평가에 많은 계산 노력을 필요로 한다. 본 연구에서는 이러한 기존 알고리즘의 단점을 보완하기 위하여 새로이 종(species)간의 경쟁 공진화 개념을 도입한 알고리즘을 제안한다. 경쟁 공진화는 생태계에서 숙주와 이에 기생하는 기생충의 관계처럼 진화적 무기경쟁(evolutionary arms race)을 통해 계속적으로 상호변화에 적응하면서 공진화하는 현상을 모방한 탐색기법으로 적대문제(adversarial problem)에 주로 적용되어 왔다[12].

### 3.2 제안한 알고리즘의 개념 및 절차

MOEA에서 다양한 좋은 해의 효율적 탐색은 모집단 운영, 적응도 평가, 선택 방법들과 관련이 깊다. 특히 모집단의 다양성 유지를 위한 적응도 평가와 모집단 운영 방법이 중요하게 인식되고 있다[28]. 본 연구에서는 기존의 적소 또는 거리개념이 아닌 경쟁공진화의 개념을 바탕으로 하는 새로

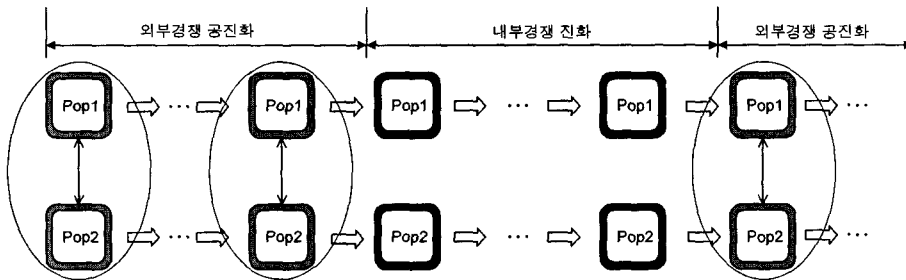
운 적응도 평가방법과 그와 관련한 모집단 운영 전략을 제안한다.

제안한 알고리즘을 다목적 경쟁진화알고리즘(Multi-Objective Competitive Evolutionary Algorithm ; MOCEA)이라 부르기로 한다. MOCEA는 두개의 모집단을 운영한다. 진화는 크게 모집단(종)간의 경쟁(외부경쟁)과 모집단 내에서, 즉 같은 종의 개체들의 경쟁(내부경쟁)으로 나뉜다. 즉, 외부경쟁은 다른 모집단에 있는 개체들과의 경쟁에 의해 적응도가 계산되고, 내부경쟁은 자신이 속한 모집단의 다른 개체들과의 경쟁에 의해 적응도가 계산된다. 외부경쟁 공진화에서는 모집단간의 경쟁을 통해 상대 모집단이 탐색하지 못하는 영역에 위치하는 개체에 높은 적응도를 부여하여 다양한 비지배해를 탐색할 수 있도록 한다. 반면, 내부경쟁 진화에서는 모집단 내 개체들이 서로 경쟁하여 파레토 최적해로의 탐색과 그 속도를 증가시키고자 한다.

[그림 2]는 MOCEA의 진화 과정을 보여주고 있다. 먼저 같은 종류의 두 모집단을 생성하고 일정 세대동안 외부경쟁을 통해 공진화한다. 이 과정은 초기 두 모집단이 넓은 탐색공간을 갖도록 한다. 그 후 각 모집단은 내부 개체들간의 경쟁을 통해 진화한다. 이는 알고리즘의 수렴성을 높여 질 좋은 비지배해의 탐색을 돕는다. 이러한 외부경쟁과 내부경쟁을 반복하면서 진화한다.

MOCEA의 주요 절차는 다음과 같다.

#### 단계 1 : 초기모집단 생성



[그림 2] MOCEA의 개념

두 모집단 Pop1, Pop2를 만들고 각 모집단에 모집단 크기(popsize)의 수만큼 임의로 개체를 생성한다. 초기 진화방법을 결정하는 조건은 외부경쟁 조건으로 둔다.

#### 단계 2 : 진화조건 판단

진화조건을 판단한다. 진화조건이 외부경쟁 조건이면 단계 3으로, 내부경쟁 조건이면 단계 4로 간다.

#### 단계 3 : 외부경쟁에 의한 공진화

두 모집단 Pop1, Pop2간의 경쟁에 의해 적응도를 평가하고, 평가된 적응도에 기초한 선택, 유전 연산을 통해 각 모집단의 개체를 진화시킨다. 이때 적응도를 기준으로 상위  $\delta\%$ 의 엘리트 개체는 다음 세대에 생존시킨다. 진화조건을 갱신하고 단계 5로 간다.

#### 단계 4 : 내부경쟁에 의한 진화

두 모집단 Pop1, Pop2는 각각 독립적으로 유전 연산에 의해 자손을 생산하고,  $(\mu + \lambda)$ -ES를 통해 진화시킨다. 진화조건을 갱신하고 단계 5로 간다.

#### 단계 5 : 종료조건

알고리즘 종료조건을 만족하면 끝내고, 그렇지 않으면 단계 2로 간다.

단계 1은 유전표현에 의해 개체를 생성하여 초기모집단을 구성하는 단계이다. 단계 2는 외부경쟁 공진화 또는 내부경쟁 진화를 선택하기 위해 진화방법에 대한 조건을 판단한다. 진화방법을 선택하는 전략은 여러 가지가 있을 수 있다. 예를 들어 미리 정해진 세대 수, 각 모집단의 진화상태가 서로 균형을 이루는지의 여부 등이 그것이다. 본 연구에서는 일정 세대를 주기로 외부경쟁 공진화와 내부경쟁 진화를 반복하는 전략을 사용하였다. 단계 3과 4는 각각 외부경쟁 공진화와 내부경쟁 진화 과정이다. 각 진화방법에 대한 구체적 절차는 4.4절과 4.5절에서 다룬다.

## 4. FMS 공정계획을 위한 다목적 경쟁진화알고리즘

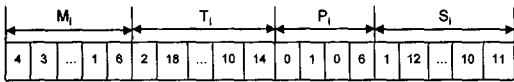
### 4.1 유전표현과 초기모집단 생성

진화알고리즘에서의 개체의 표현은 자연스럽게 명확하며, 중복되어 표현되지 않는 것이 바람직하다. 본 연구에서 다루는 FMS 공정계획 문제의 경우, 각 부품별로 기계, 공구, 공정, 그리고 순서 유연성이 적절하게 표현될 수 있어야 한다.

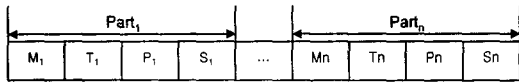
개체 표현은 각 부품 별로 기계, 공구, 공정, 그리고 작업순서를 위한 부분으로 나타내지고, 이러한  $n$ 개의 부품별 표현이 결합되어 하나의 완전한 개체를 이룬다. 개체가 부품별로 다른 부분으로 이루어진 이유는 부품마다 각 작업이 할당되는 기계, 공구가 다르고, 수행되는 공정경로가 다르며, 또한 작업순서가 다르기 때문이다. [그림 3](a)는  $n$ 개의 부품으로 이루어진 문제에서 부품  $i$ 의 표현이다. 기계, 공구를 표현한 부분에서의 위치는 부품  $i$ 가 수행될 작업을 나타내고 인자값은 각 작업이 할당될 기계, 공구번호를 각각 나타낸다. 대안 공정을 표현하는 부분에서 위치는 'OR'기호의 번호를 나타내고, 인자값은 부품마다 'OR'기호에 의해 분지되는 최대가지수를  $B$ 라 한다면  $[0, B-1]$  구간의 정수값을 갖는다. 그 값이  $b$ 라면 왼쪽에서  $(b+1)$ 번째 가지의 작업들을 수행함을 뜻한다. 본 연구에서는 하나의 'OR'기호에 의해 분지되는 최대 가지수를 2로 가정하였다. 따라서 실제로 공정을 위한 부분의 인자값은 0 또는 1의 값을 갖는다. 순서를 위한 부분에서는 부품  $i$ 를 완성하기 위한 작업들의 가공순서를 나타낸다. 물론 개체에 나열된 작업들의 순서는 작업 선행 관계를 만족해야 한다. 이때 이 순열은 대안공정들로 인해 실제로 수행되지 않을 수 있는 작업들도 포함한 모든 작업들로 이루어진다.

초기 모집단을 구성할 때, 작업의 선행 관계를

만족하는 순서의 생성 방법은 다음과 같다. 먼저, 선행작업이 없는 작업들의 집합 A를 구하고, 집합 A에서 임의로 한 작업을 선택하여 순서열에 추가한다. 집합 A에서 선택된 작업을 삭제하고, 삭제된 작업의 직후행 작업들 중에서 모든 선행 작업이 할당된 작업을 집합 A에 추가한다. 그리고 갱신된 집합 A에서 임의로 하나의 작업을 선택하여 순서열에 추가한다. 이와 같은 과정을 모든 작업의 순열이 형성될 때까지 반복한다. n 개의 부품을 갖는 문제는 각 부품의 표현들이 모여서 [그림 3](b)와 같은 완전한 개체가 된다.



(a) 부품 i의 개체표현



(b) n개의 부품을 갖는 문제의 개체표현

[그림 3] 다목적 FMS 공정계획을 위한 개체표현

4.2 목적함수 벡터

본 연구에서 다루는 문제에는 각 부품의 작업간 순서제약, 대안기계제약, 대안공구제약, 대안가공 경로제약, 공구매거진 한계용량제약, 공구의 가용 공구수제약이 존재한다. 본 연구에서는 이 중에서 대안기계, 대안공구, 대안공정, 대안순서 제약은 개체표현에 직접 반영하여, 개체들이 이들 제약을 항상 만족하면서 진화하도록 한다. 이를 위해 이들 제약을 항상 만족하는 유전표현 및 유전연산을 개발하여 사용한다.

공구매거진의 한계용량제약과 각 공구의 가용공구수 제약은 제약의 위반정도에 따라 벌금을 부과하는 벌금함수로서 처리하였다. 즉, 제약의 고려없이

계산된 원 목적함수 벡터  $f = (f_1, f_2, \dots, f_L)$ 를 이 두 제약을 고려하여 벌금이 부과된 새로운 목적함수 벡터  $f' = (f'_1, f'_2, \dots, f'_L)$ 로 변환한다. 이 변환된 새로운 목적함수 벡터를 이용하여 적응도를 평가한다. 본 연구에서는 다음과 같은 변환식을 사용하였다.

$$f'_l = f_l + c_1 \sum_m MP_l(m)^{\alpha} + c_2 \sum_t TP_l(t)^{\beta}, \quad l = 1, 2, \dots, L \quad (2)$$

여기서,

$$MP_l(m) = \begin{cases} \frac{(b_m - B_m)f_l}{B_m}, & \text{if } b_m > B_m \\ 0, & \text{otherwise} \end{cases}$$

$$TP_l(m) = \begin{cases} \frac{(h_t - H_t)f_l}{H_t}, & \text{if } h_t > H_t \\ 0, & \text{otherwise} \end{cases}$$

$B_m$ 와  $b_m$ 는 각각 기계 m의 공구매거진 용량과 계획에 의해 필요로 하는 요구량을,  $H_t$ 와  $h_t$ 는 공구 t의 총 가용량과 계획에 의해 요구되는 사용량을 각각 나타낸다.  $MP_l(m)$ 와  $TP_l(t)$ 는 목적 l에 대해 각각 공구매거진과 공구 가용량 한계에 따른 벌금함수를 나타낸다.  $c_1, c_2, \alpha, \beta$ 는 상수로써 벌금함수의 강도를 조절하는 파라미터이다.

이와같이 변환된 목적함수 벡터에 의해 각 개체들을 비교하여(경쟁하여) 지배 또는 비지배관계를 파악한다. 이러한 개체들간의 지배관계에 기초하여 적응도를 부여한다. 이는 4.4절과 4.5절에서 구체적으로 다룬다.

4.3 교차와 돌연변이

교차는 문제의 특성을 반영하는 유전 정보를 부모로부터 추출하여 자손에게 잘 전달할 수 있어야 한다. 교차와 돌연변이의 유전연산은 개체에서 각

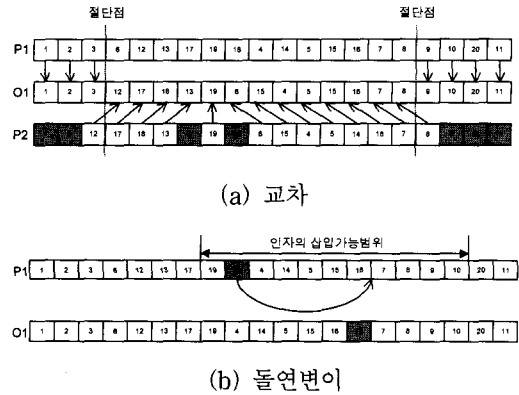


부품의 공정계획을 표현한 네 부분, 즉 기계, 공구, 공정, 그리고 작업순서에 대해 각기 그 특성에 맞게 행하여진다.

개체에서 기계, 공구를 나타내는 인자 부분의 교차는 이점(two-point)교차를 사용한다. 그리고 이 두 부분의 돌연변이는 각 인자에 돌연변이율에 따라 대안기계와 대안공구를 임의로 각각 선택하는 방법을 사용한다. 개체의 공정표현은 비교적 짧은 길이를 가지므로 일점(one-point)교차를 사용한다. 공정표현의 돌연변이는 인자값이 0이면 1로, 1이면 0으로 값을 변경한다.

개체의 작업순서를 나타낸 부분의 교차는 수정 이점교차를 사용한다. 교차 방법은 다음과 같다. 먼저 두개의 임의 절단점을 선택한다. 두 절단점에 의해 나누어진 앞, 중간, 뒷부분에 있는 인자 중 앞부분과 뒷부분은 한 부모(P1)로부터 자손(O1)의 같은 위치에 복사한다. 그리고, 다른 부모(P2)에서 복사된 인자들과 같은 값을 갖는 인자를 찾아서 지운다. 부모 P2에 남아있는 인자들은 순서대로 O1의 빈 위치에 차례로 복사한다 [그림 4](a)는 이 교차의 예를 보인 것이다. 이 교차에서 선행제약을 만족하는 부모(P1)의 앞과 뒤 부분이 자손에 상속되어 이 부분은 작업선행조건을 만족하고, 중간 부분 또한 선행제약을 만족하는 부모(P2)의 작업순서를 상속 받음으로써 생산된 자손 개체는 작업선행제약을 만족하게 된다.

한편, 돌연변이는 아래와 같이 수행된다. 먼저, 부모개체로부터 한 인자(작업)를 임의로 선택한다. 그리고 작업 선행관계를 어기지 않도록 하기 위해, 선택된 인자의 삽입가능범위를 구한다. 삽입가능범위는 개체에 나타나는 순서에서 해당 작업의 직선행 작업 중 마지막 작업 이후부터 직후행 작업 중 첫번째 작업 앞까지이다. [그림 1](a)에서 18번 작업의 직선행 작업은 17번 작업이고 직후행 작업은 20번 작업이다. 따라서 [그림 4](b)의 18번 작업이 삽입 가능한 범위는 17번 작업 이후부터 20번 작업 앞까지가 된다.



[그림 4] 작업 순서를 표현한 인자들의 유전연산

#### 4.4 외부경쟁에 의한 공진화

외부경쟁 공진화에서 개체의 적응도는 상대 모 집단 개체들과의 경쟁에 의해 평가된다. 즉 상대 모집단에 있는 개체에 쉽게 지배되지 않는 개체에 더 높은 적응도를 부여함으로써 상대 모집단에 대해 강한 개체가 살아남도록 한다. 이는 상대 모집단의 개체에 대해 지배되지 않는 영역에 위치한 개체가 높은 적응도를 갖게 되어, 상대 모집단이 탐색하지 못한 영역으로의 탐색을 유도한다. 따라서 이 과정을 통해 다양한 비지배해를 탐색하기 위해 사용되던 기존 연구의 적소 매커니즘이나 거리개념에 의한 다양성 유지 기법과 유사한 효과를 얻고자 한다. 또한 엘리트즘 전략을 사용한다. 이 과정은 3장의 알고리즘 절차에서 단계 3에 해당되고 그 구체적 절차는 다음과 같다.

##### [외부경쟁 공진화 : 단계 3]

단계 3.1  $Pop1$ 과  $Pop2$ 의 개체는 각기 상대 모집단에 있는 개체와 경쟁을 통해 적응도를 구한다.

단계 3.2  $F=0$ ,  $TP = \emptyset$ ,  $EP = Pop1$ 으로 둔다.

단계 3.3  $EP$ 에서 적응도가 가장 높은  $\delta\%$ 를 임시모집단  $TP$ 에 복사하여 둔다.

단계 3.4  $EP$ 에서 토너먼트 선택, 유전연산을 통해 자손 모 집단  $O$ 를 생산한다.

단계 3.5  $O$ 에 있는 임의 개체와  $TP$ 의 개체를 대체하여, 모집단  $O$ 를 갱신한다.

단계 3.6  $F=0$ 이면  $Pop1=O$ ,  $F=1$ ,  $TP=\phi$ ,  $EP=Pop2$ 로 두고 단계 3.3으로 간다. 그렇지 않으면 단계 3.7로 간다.

단계 3.7  $Pop2=O$ 로 두고, 경쟁공진화의 종료 조건을 만족하면(단계 3을 끝내고) 단계 4로 간다. 그렇지 않으면 단계 3.1로 간다.

단계 3.1에서  $Pop1$ 의 개체  $i$ 의 적응도는 다음과 같이 계산한다.

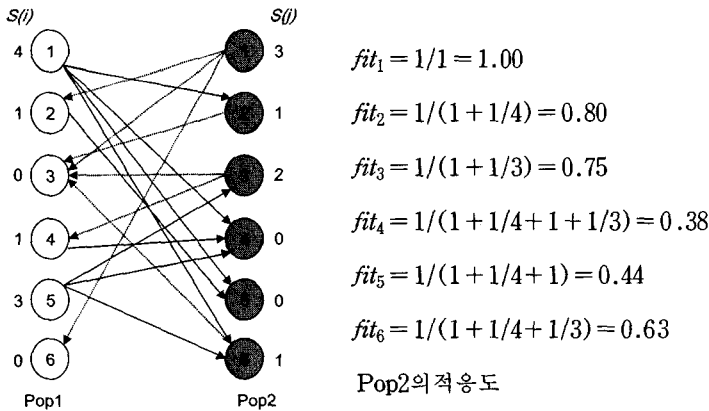
$$fit_i = \frac{1}{1 + \sum_{(j \in Pop2, j > i)} \frac{1}{s(j)}}, \forall i \in Pop1 \quad (3)$$

여기서,  $j > i$ 는 개체  $j$ 가 개체  $i$ 를 지배함을 나타낸다.  $s(j)$ 는  $Pop2$ 의 개체  $j$ 가 지배하는  $Pop1$ 의 개체 수를 나타내고, 이때  $Pop2$ 에 있는 개체  $j$ 는  $Pop1$ 에 있는 개체  $i$ 를 지배하는 개체이다. 식 (3)에서 분모에 1을 더해 준 이유는 분모가 0이 되는 경우를 피하기 위함이다. 이 방법은 같은 수의 개체에 의해 지배된다 할지라도 약한 개체에 의해 지배되는 개체보다 강한 개체에 지배되는 개체에 상대적으로 높은 적응도가 부여된다. 이것은

각 모집단의 탐색방향을 상대 모집단과의 경쟁에서 서로 지배되지 않는 영역으로 유도하여 결국 두 모집단이 다양한 영역을 탐색할 수 있도록 한다.

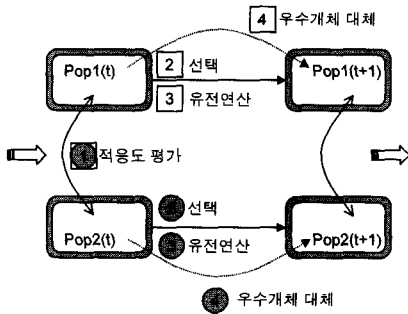
[그림 5]는 모집단  $Pop2$ 의 개체가 모집단  $Pop1$ 과의 경쟁을 통해 적응도를 부여받는 과정을 보여주고 있다. 그림에서  $a \rightarrow b$ 는 개체  $a$ 가 상대모집단에 있는 개체  $b$ 를 지배함을 의미한다.  $Pop2$ 의 개체 5, 6을 비교해보자. 5, 6번의 두 개체는 모두  $Pop1$ 의 2개 개체에 의해 지배되고 있다. 즉,  $Pop2$ 의 5번 개체는  $Pop1$ 의 1, 2번 개체에 의해 지배되고,  $Pop2$ 의 6번 개체는  $Pop1$ 의 1, 5번 개체에 의해 지배된다. 이 때  $Pop1$ 의 5번 개체가  $Pop1$ 의 2번 개체보다  $s(i)$ 값이 높다. 이는 5번 개체가 2번 개체보다 더 강함을 뜻한다. 따라서 같은 수에 의해 지배를 받지만 상대적으로 약한 개체에게 지배를 받는  $Pop2$ 의 6번 개체가  $Pop2$ 의 5번 개체보다 더 높은 적응도를 부여받게 된다.

적응도 평가 후 각 모집단은 구해진 적응도에 따라 선택과 유전연산을 통해(단계 3.4) 다음 세대로 진화한다. 선택방법은 토너먼트 선택으로, 토너먼트 크기는 2로 두었다. 유전 연산 방법은 4.3절에서 이미 언급하였다. 그리고 우수한 적응도를 갖는 개체  $\delta\%$ 를 다음 세대 모집단의 임의 개체와 대체하는 엘리트즘 전략을 사용한다(단계 3.3



[그림 5] 외부경쟁 공진화에서 적응도 평가

과 단계 3.5). 이러한 경쟁공진화에 의한 한 주기 (cycle) 진화의 종료 조건으로는 공진화의 세대수로 두었다. [그림 6]은 이러한 외부경쟁 공진화의 과정을 보여준다.



[그림 6] 외부경쟁 공진화 과정

#### 4.5 내부경쟁에 의한 진화

앞에서 언급하였듯이 MOCEA에서 내부경쟁 진화의 주목적은 파레토 최적해로의 탐색 강화와 수렴속도를 높이기 위함이다. 외부경쟁 공진화에서 적응도는 상대 모집단에 의해 평가되지만, 내부경쟁 진화에서는 자신이 속한 모집단내 개체와의 경쟁에 의해 적응도가 평가된다. 모집단 내 다른 개체보다 상대적으로 우수한 개체가 다음 세대에 살아남아, 파레토 최적해로의 탐색을 유도할 수 있도록 한다. 이는 3장의 알고리즘 절차에서 단계 4에 해당되고 그 구체적 절차는 다음과 같다.

##### [내부경쟁 진화 : 단계 4]

- 단계 4.1  $F=0$ ,  $EP=Pop1$ 로 둔다.
- 단계 4.2  $EP$ 에서, 부모 쌍을 임의로 짝지어, 이들을 유전연산하여  $\lambda$ 개의 자손 집합  $O$ 를 생성한다.
- 단계 4.3  $TP=EP \cup O$ 로 둔다.
- 단계 4.4  $TP$ 에서  $(\mu + \lambda)$ -ES에 의해  $EP$ 의 다음세대 모집단을 구성한다.
- 단계 4.5  $F=0$ 이면  $F=1$ ,  $EP=Pop2$ 로 두고 단계 4.2로 간다.

단계 4.6 내부경쟁진화의 종료조건을 만족하면 (단계 4을 끝내고) 단계 5로 간다. 그렇지 않으면 단계 4.1로 간다.

최근 연구들은 엘리티즘 전략이 MOEA의 성능에 매우 중요한 역할을 하고 있음을 보여준다[20, 31]. MOEA에서 엘리티즘의 적용은 외부모집단 (external population 또는 archive)을 두어 발견된 비지배해를 보관하고 계속 갱신해 가는 방법과  $(\mu + \lambda)$ -ES를 통해 발견된 비지배해가 사라지지 않도록 하는 방법이 주로 사용된다. 전자의 경우는 SPEA2에서, 후자의 경우는 NSGA-II에서 사용하였다. 본 연구에서는  $(\mu + \lambda)$ -ES를 채용한다. 이때  $(\mu + \lambda)$ 개에서 가장 좋은  $\mu$ 개의 선택은 NSGA-II에서와 같은 방법을 사용한다. 단계 4.2에서 유전 연산은 4.3절에서 언급한 방법을 사용하였으며, 단계 4.6에서의 한 주기의 내부경쟁 종료조건은 외부경쟁 공진화에서와 같이 세대 수를 사용하였다.

## 5. 실험 및 결과

### 5.1 실험 문제 및 실험 설계

본 연구에서 제안한 MOCEA의 성능 평가를 위해 가공 작업 및 작업경로가 서로 다른 18개의 부품들을 조합하여 생성한 20개의 문제를 사용하여 실험하였다. 각 문제의 부품수와 총 작업수는 <표 1>과 같다. 실험문제들은 문제크기(부품 및 작업수)와 유연성 측면에서 가능한 다양한 범위를 갖도록 설계하였다. 본 논문에서는 지면의 제약상 [그림 1]과 같은 각 문제의 네트워크 표현은 보이지 않았다. 구체적인 부품들의 가공정보와 문제 생성방법은 Kim[13]과 Kim et al.[14]를 참고할 수 있다.

각 문제에 대해 총 가용 기계수는 10대, 총 가용공구는 20종류로 두었다. 각 기계별 공구매거진의 한계용량, 각 공구유형별 가용공구수 및 공구

〈표 1〉 실험문제

문 제 번 호	부 품 수	문제별 총작업수
Prob 1 ~ Prob 8	6	121, 99, 102, 109, 103, 111, 105, 76
Prob 9 ~ Prob 15	9	142, 168, 150, 160, 155, 155, 151
Prob 16 ~ Prob 20	12	221, 201, 211, 199, 207

〈표 2〉 기계 및 공구관련 제약

기 계		공 구					
번 호	공구매거진 용 량	번 호	가 용 공구수	슬 롯 요구량	번 호	가 용 공구수	슬 롯 요구량
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

당 슬롯요구량은 <표 2>와 같이 사용하였다. 이때, 공구매거진의 용량은 모든 유형의 공구가 갖는 슬롯요구량의 총합을 초과하지 않도록 하였고, 각 공구유형별 가용공구수는 다양한 값을 사용하여 제약의 강약을 표현할 수 있도록 하였다.

제안한 알고리즘은 성능 비교를 위해 최근에 제안된 NSGA-II, SPEA2와 비교되었다.

실험을 위한 모든 알고리즘은 동일한 프로그램 언어로 구현되었으며, 동일한 CPU를 장착한 IBM 계열 PC에서 수행되었다. 다른 파라미터는 예비 실험을 통해 비교적 우수한 성능을 보인 것으로 결정하였다. 식 (1)에서 기계의 단위당 준비비용인 MSC는 5, 공구의 단위당 준비비용인 TSC는 1로 하였다. 이들은 모두 기계나 공구의 유형과는 무관하게 일정한 값으로 두었다. 벌금함수와 관련된 파라미터,  $C_1$ ,  $C_2$ 는 30, 그리고  $\alpha$ ,  $\beta$ 는 0.5로 설정하였다. 외부경쟁진화에서 다음 세대로 전달되

는 우수개체의 비율인  $\delta$ 는 20%로 두었다. 또한 본 연구의 실험에서는 25세대를 주기로 외부경쟁진화와 내부경쟁진화를 반복하도록 하였다. 모집단의 크기는 알고리즘들의 구조를 고려하여 서로 다르게 설정하였다. 즉 두 모집단을 운영하는 MOCEA는 각 모집단 크기는 100으로, 합이 200으로 두었다. NSGA-II와 SPEA2 역시 각각 200으로 두었다. 유연연산과 관계된 파라미터인 교차율과 돌연변이율은 타 연구[25]에서 비교적 좋은 성능을 보이는 0.7, 0.01을 각각 사용하였다. 세 알고리즘 모두 200세대 후에 종료하였으며 종료 후 얻어진 비지배해 집합을 비교 대상으로 삼았다.

## 5.2 실험결과 및 분석

비지배해 집합을 가능한 다양하게 찾고자 하는 목표를 갖는 MOEA는 다양한 측면에서 그 성능을 평가할 수 있다. 즉, 파레토 최적해로의 수렴속

도, 모집단의 다양성 유지, 최종적으로 얻은 비지배해의 파레토 최적해와의 근사성과 확산성 (spread) 및 균일성 등이 평가척도로 사용될 수 있다. 여러 연구에서 비교를 위한 척도를 제시하고 있지만 제한된 측면만을 반영하는 척도에 머물고 있다[26]. 이러한 이유로 MOEA 비교척도에 관한 연구 역시 MOEA 분야의 향후 연구과제로써 인식되고 있다[3].

<표 3>은 본 연구에서 제안한 MOCEA와 NSGA-II, SPEA2의 성능을 질적, 양적 측면에서 비교하기 위해 Hyun et al.[10]에서 사용한 척도으로써 계량화한 결과이다. 설명을 위해 먼저 사용되는 기호를 정의하자. 알고리즘 A, B에 의해 얻은 최종 비지배해 집합을  $Set_A, Set_B$ 라 하고, 그 크기를 각각  $N_A(=|Set_A|), N_B(=|Set_B|)$ 라고 하면,  $N_A, N_B$ 는 알고리즘 A와 B의 양적 지표가 된다. 또한  $Set_A, Set_B$ 의 합집합에서 추출된 새

로운 비지배해 집합  $Set_{AB}$ 를 만들고 그 크기를  $N_{AB}(=|Set_{AB}|)$ 라고 하자.  $Set_{AB}$ 에서  $Set_A$ 에 속한 개체수,  $Set_B$ 에 속한 개체수를 각각  $N_{A/AB}(=|Set_A \cap Set_{AB}|), N_{B/AB}(=|Set_B \cap Set_{AB}|)$ 라고 놓으면  $N_{A/AB}, N_{B/AB}$ 값은 알고리즘 A, B에 의해 수행된 최종 비지배해의 질적 비교척도로 사용될 수 있다. 알고리즘 MOCEA, NSGA-II, SPEA2를 간단히 M, N, S로 나타낸다. <표 3>에 나타난 값은 10회 반복실험의 평균 값이다.

그리고 <표 3>의 실험결과에 대한 통계적 분석을 행하였다. <표 4>는 MOCEA와 기존 알고리즘간의 유의한 차이가 있는지를 알아보기 위한 t-검정 결과이며, 표의 수치는 유의확률이다. 각 문제별로 검정 통계량을 구한 후 유의확률이 유의수준보다 낮으면 비교한 두 알고리즘은 통계적으로 성능 차이가 있음을 의미한다. 유의수준은 5%로 두

<표 3> 알고리즘별 성능비교

Prob.	Algorithms			MOCEA vs NSGA-II			MOCEA vs SPEA2			NSGA-II vs SPEA2		
	MOCEA	NSGA-II	SPEA2	$N_{MN}$	$N_{M/MN}$	$N_{N/MN}$	$N_{MS}$	$N_{M/MS}$	$N_{S/MS}$	$N_{NS}$	$N_{N/NS}$	$N_{S/NS}$
	$N_M$	$N_N$	$N_S$									
Prob 1	9.6	5.2	9.2	9.3	8.2	1.1	9.5	7.6	1.9	7.2	3.6	3.6
Prob 2	9.7	6.4	10.0	9.1	6.1	3.0	9.7	7.7	2.0	9.4	5.5	3.9
Prob 3	10.9	6.4	12.7	11.1	7.1	4.0	12.4	9.5	2.9	11.0	5.3	5.7
Prob 4	11.3	5.8	11.3	9.4	4.6	4.8	13.4	9.7	3.7	8.8	5.7	3.1
Prob 5	9.9	6.7	9.6	11.0	9.0	2.0	10.6	6.8	3.8	10.4	5.2	5.2
Prob 6	11.5	6.8	9.9	9.5	6.4	3.1	12.9	9.9	3.0	9.4	6.0	3.4
Prob 7	8.8	5.4	10.9	7.8	4.8	3.0	10.7	6.5	4.2	9.1	4.1	5.0
Prob 8	11.4	6.1	7.9	8.7	5.8	2.9	12.5	9.7	2.8	9.0	5.6	3.4
Prob 9	10.9	6.6	9.6	9.2	3.7	5.5	10.9	8.5	2.4	8.5	6.4	2.1
Prob10	9.1	6.3	10.1	8.8	6.5	2.3	9.9	7.6	2.3	9.7	4.7	5.0
Prob11	9.0	7.0	11.9	9.3	8.0	1.3	10.1	7.7	2.4	11.4	4.7	6.7
Prob12	8.5	6.0	10.3	8.7	3.8	4.9	9.1	6.4	2.7	9.3	5.7	3.6
Prob13	8.6	7.0	11.3	7.7	4.3	3.4	10.5	6.7	3.8	10.4	5.2	5.2
Prob14	10.6	6.9	11.3	10.8	8.2	2.6	12.1	8.5	3.6	11.1	4.9	6.2
Prob15	9.6	6.5	9.2	9.2	8.3	0.9	10.4	7.3	3.1	9.6	5.0	4.6
Prob16	7.0	7.5	10.9	7.1	7.0	0.1	9.0	5.1	3.9	11.1	1.5	9.6
Prob17	8.9	6.7	10.6	8.9	8.5	0.4	9.9	6.3	3.6	9.9	1.6	8.3
Prob18	9.4	7.8	10.9	9.4	8.8	0.6	10.4	7.4	3.0	10.7	3.8	6.9
Prob19	9.6	6.3	10.6	9.8	8.1	1.7	11.9	5.5	6.4	10.3	2.7	7.6
Prob20	9.6	6.2	13.1	9.1	8.8	0.3	12.5	8.3	4.2	11.6	1.6	10.0

〈표 4〉 문제별 알고리즘간 유의확률

Prob.	양 적 비 교		질 적 비 교	
	MOCEA vs NSGA-II	MOCEA vs SPEA2	MOCEA vs NSGA-II	MOCEA vs SPEA2
Prob 1	0.001*	0.806	0.000*	0.000*
Prob 2	0.009*	0.857	0.009*	0.000*
Prob 3	0.001*	0.190	0.053	0.001*
Prob 4	0.000*	1.000	0.901	0.000*
Prob 5	0.002*	0.778	0.000*	0.042*
Prob 6	0.001*	0.202	0.055	0.000*
Prob 7	0.004*	0.112	0.163	0.112
Prob 8	0.000*	0.019*	0.006*	0.000*
Prob 9	0.001*	0.403	0.302	0.000*
Prob 10	0.024*	0.513	0.001*	0.000*
Prob 11	0.085	0.039**	0.000*	0.000*
Prob 12	0.029*	0.308	0.342	0.024*
Prob 13	0.241	0.020**	0.271	0.029*
Prob 14	0.014*	0.585	0.003*	0.001*
Prob 15	0.006*	0.763	0.000*	0.001*
Prob 16	0.685	0.001**	0.000*	0.348
Prob 17	0.013*	0.109	0.000*	0.050
Prob 18	0.260	0.329	0.000*	0.000*
Prob 19	0.009*	0.496	0.000*	0.660
Prob 20	0.002*	0.030**	0.000*	0.001*

\* 유의수준 5%에서 알고리즘간 유의한 차이가 있으며 MOCEA가 우수.

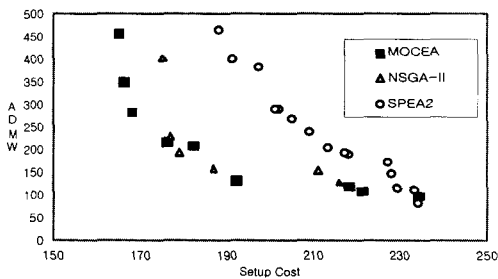
\*\* 유의수준 5%에서 알고리즘간 유의한 차이가 있으며 MOCEA가 열등.

었다. 표에서 \*는 MOCEA가 유의수준에서 비교 알고리즘보다 우수함을 나타내며, (\*\* )는 MOCEA가 유의수준에서 비교 알고리즘보다 열등함을 의미한다.

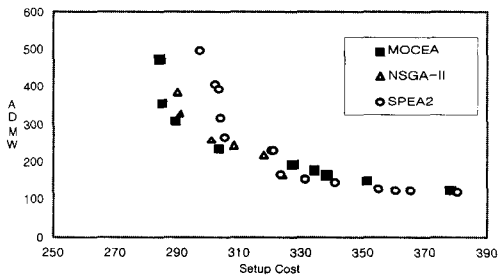
〈표 3〉과 〈표 4〉로부터, 다음과 같은 결과를 얻었다. 먼저, 해의 양적 척도,  $N_M$ ,  $N_N$  과  $N_S$ 를 비교하여 보면, MOCEA는 NSGA-II 보다는 좋고, SPEA2와는 비슷한 결과를 보였다. 그리고 MOCEA를 NSGA-II와 질적으로( $N_{M/MN}$ 과  $N_{N/MN}$ 의 비교), 그리고 SPEA2와 질적으로( $N_{M/MS}$ 과  $N_{S/MS}$ 의 비교) 비교하면, MOCEA는 이들 두 비

교 알고리즘 보다 많은 실험문제에서 질적으로 우수함을 알 수 있다. 특히, 규모가 큰 문제에서 MOCEA는 NSGA-II보다 질적 척도에서 더 우수한 성능을 보였다. SPEA2는 MOCEA와 거의 같은 수의 비지배해를 찾지만, 이들 해 중에서 상당수는 MOCEA에 의해 구한 비지배해들에 의해 지배된다는 것을 알 수 있다. NSGA-II와 SPEA2의 비교한 경우 양적( $N_N$ 와  $N_S$ 의 비교)으로 SPEA2가 더 좋은 결과를 얻었고 질적 측면( $N_{N/NS}$ 과  $N_{S/NS}$ 의 비교)에서는 두 알고리즘이 유사한 성능을 보였다.

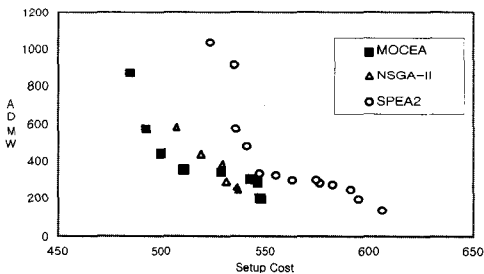
[그림 7]은 Prob 3, Prob 14, Prob 18에서 얻은 최종 비지배해들을 보여 주고 있다. 그림에서 보는 바와 같이 SPEA2는 다양한 탐색을 보인 반면 파레토 최적해에 대한 수렴성이 낮고, NSGA-II의 경우 수렴성은 SPEA2보다 높지만 다양한 탐색에는 단점을 보였다. MOCEA는 이런 두 알고리즘의 단점들을 어느 정도 보완하고 있음을 알 수 있다.



(a) Prob 3



(b) Prob 14



(c) Prob 18

[그림 7] 해의 질과 다양성 비교

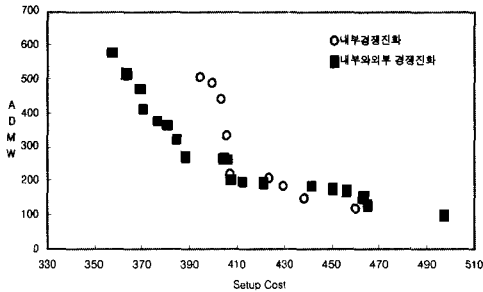
한편, 비교 알고리즘들의 컴퓨터 계산소요시간은 아래와 같다. 이 시간은 각 문제에 대해 10회 반복한 실험의 평균 CPU 시간이다. MOCEA, NSGA-II, SPEA2의 순서로 Prob 1은 4.02분, 4.59분, 14.36분, Prob 9는 5.37분, 6.23분, 18.4분, 그리고 Prob 16은 6.96분, 7.85분, 22.80분이 평균 소요되었다. 문제 크기에 따라 계산소요시간이 거의 같았으므로 대표적인 3문제만을 제시하였다. 다양한 비지배해의 유지를 위해, NSGA-II와 SPEA2는 개체 선택시 각기 개체간 거리를 계산하여야 한다. 반면, MOCEA는 개체간의 경쟁 결과에 의해 개체가 선택되므로, NSGA-II와 SPEA2보다는 상대적으로 계산량이 적어 계산시간이 약간 적게 소요되었다. 일반적으로, CPU 계산소요시간은 알고리즘 구현 과정에서 자료구조, 메모리 관리, 프로그래밍 기술 등의 여러 요인에 의해 약간 달라질 수 있다.

### 5.3 외부경쟁 공진화의 효과

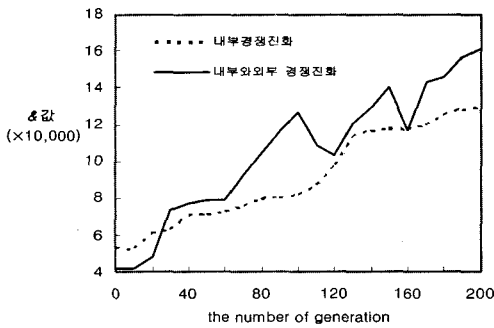
MOCEA에서 외부경쟁이 알고리즘 성능에 미치는 영향을 살펴보자. [그림 8]은 이를 보여 주고 있다. Prob 9 문제에 대해 MOCEA를 25세대 주기로 외부경쟁 공진화와 내부경쟁진화를 반복한 경우와 MOCEA에서 외부경쟁 공진화를 수행하지 않고 내부경쟁진화만을 수행한 결과를 비교하였다. [그림 8](a)는 두 경우에 대한 200세대 후의 최종 비지배해를 그래프에 표시한 것이다. 이로부터 외부경쟁 공진화는 다양하고 좋은 해를 유도하는데 기여함을 알 수 있다. 그리고 진화과정에 따른 해의 질과 다양성의 변화를 살펴보자. 이를 위하여 Zitzler and Thiele[33]가 제안한 지배된 영역의 크기(size of the dominated space: 이 크기를 &로 표시함)를 척도로 사용하였다. & 값은 비지배해들이 지배하는 영역의 크기로 정의되며, 비지배해 수가 많을수록, 파레토 최적해에 가까울수록 커진다는 특성이 있다.

진화 초기에는 외부경쟁에서 보다는 내부경쟁에

서 빠른 수렴속도를 보이므로 내부경쟁 진화가 더 높은 & 값을 갖는다. 하지만 초기 외부경쟁에 의한 공진화 후 내부경쟁에 의한 진화가 이루어지면 점차 내부와 외부경쟁이 결합된 형태의 진화가 (실선으로 표시됨) 더 높은 & 값을 유지한다. 이것은 외부경쟁 공진화가 다양하고 좋은 해탐색에 효과적이고 효율적임을 나타낸다.



(a) 진화방법에 따른 최종 비지배해



(b) 진화방법에 따른 효율 비교

[그림 8] 외부경쟁 공진화의 효과

외부와 내부 경쟁진화간의 변환 주기는 하나의 파라미터이다. 주기가 너무 짧으면 충분한 경쟁이 이루어지지 않아 해공간의 탐색이 불충분할 수 있고, 반면에 주기가 너무 길면 내부경쟁진화에서는 국부 최적해로 조기에 수렴할 위험이 있으며, 외부경쟁진화에서는 파레토 최적해로의 수렴을 방해할 수 있다.

## 6. 결 론

본 연구에서는 유연제조시스템(FMS)의 다양한 유연성과 여러 목적을 동시에 고려하는 다목적 공정계획문제를 다루고, 이를 해결하는 방법론으로 다목적 진화알고리즘(MOEA)의 한 형태인 다목적 경쟁진화알고리즘(MOCEA)을 제안하였다.

MOCEA는 기존 MOEA 연구들의 해의 수렴성과 다양성을 높이기 위해 모집단의 운영 및 적응도 평가에 많은 노력을 필요로 하는 단점을 보완하는 새로운 진화 방법, 즉 외부경쟁 공진화와 내부경쟁 진화를 도입하였다. 외부경쟁 공진화에서는 모집단 간 외부경쟁을 통해 모집단의 다양한 해탐색을 가능하게 하고, 내부경쟁 진화에서는 모집단 내 경쟁을 통해 파레토 최적해로의 탐색강화와 수렴속도를 증가시킨다. 또한 MOCEA는 발견된 비지배해를 보관하고 이용하기 위한  $(\mu + \lambda)$ -ES의 엘리티즘 전략을 채용하여 알고리즘의 성능을 향상시켰다.

MOCEA의 성능을 평가하기 위해 유연성의 수준을 달리하여 생성한 실험문제에 제안한 알고리즘을 적용하고 그 결과를 최근에 제안된 NSGA-II, SPEA2와 비교하여 그 적용성을 검증하였다. 실험 결과, MOCEA는 해의 양적 척도에서 비교 알고리즘보다 우수하거나 비슷하였고, 질적 척도에서는 비교한 두 알고리즘보다 많은 실험문제에서 우수한 성능을 보였다. 이러한 실험결과는 제안한 알고리즘이 하나의 유망한 다목적진화알고리즘임을 보여 주고 있다.

## 참 고 문 헌

[1] Brandimarte, P. and M. Calderini, "A hierarchical bicriterion approach to integrated process plan selection and job shop scheduling," *International Journal of Production Research*, Vol.33, No.1(1995), pp.161-181.



- [2] Coello, C.A.C., "A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques," *Knowledge and Information Systems*, Vol.1, No.3(1999), pp.269-308.
- [3] Deb, K., "Multi-objective Genetic Algorithms : Problem Difficulties and Construction of Test Problems," *Evolutionary Computation*, Vol.7, No.3(1999), pp.205-230.
- [4] Deb, K, S. Agrawal, A. Pratap, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization : NSGA-II," In M. S. et al.(Ed.), *Parallel Problem Solving from Nature PPSN VI*, Berlin, Springer, 2000, pp. 849-858.
- [5] Fonseca, C.M. and P.J. Fleming, "Genetic algorithm for multiobjective optimization, formulation, discussion and generalization," In Forrest, S. (ed.) *Genetic Algorithms : Proceeding of the Fifth International Conference*, Morgan Kaufmann, San Mateo, CA. 1993, pp.416-423.
- [6] Goldberg, D.E., "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison Wesley, Reading, Massachusetts, 1989.
- [7] Guerrero, F., S. Lozano, T. Koltai, and J. Larraneta, "Machine loading and part type selection in flexible manufacturing systems," *International Journal of Production Research*, Vol.37(1999), pp.1303-1317.
- [8] Ho, Y.C. and C.L. Moodie, "Solving cell formation problems in a manufacturing environment with flexible processing and routing capabilities," *International Journal of Production Research*, Vol.34(1996), pp. 2901-2923.
- [9] Horn, J., N. Nafpliotis, and D.E. Goldberg, "A niched Pareto genetic algorithm for multi objective optimization," *IEEE International Conference on Evolutionary Computation*, Vol.1(1994), pp.82-87.
- [10] Hyun, C.J., Y.H. Kim and Y.K. Kim, "A genetic algorithm for multiple objective sequencing problems in mixed model assembly," *Computers & Operations Research*, Vol.25(1998), pp.657-690.
- [11] Khare, V., X. Yao and K. Deb, "Performance Scaling of Multi-objective Evolutionary Algorithms," in Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb and Lothar Thiele(editors), *Evolutionary Multi-Criterion Optimization*, Second International Conference, EMO 2003, pp.376-390, Springer. Lecture Notes in Computer Scienc., Vol.2632, Faro, Portugal, April, 2003.
- [12] Kim, J.Y., Kim, Y.K. and Kim, Y.H., "Tournament Competition and its Merits for Coevolutionary Algorithms," *Journal of Heuristics*, Vol.9. Issue 3(2003), pp.249-268.
- [13] Kim, Y.K., "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>. 2002.
- [14] Kim, Y.K., K.T. Park and J.S. Ko, "A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling," *Computers & Operations Research*, Vol.30(2003), pp.1151-1171.
- [15] Knowles, J.D. and D.W. Corne, "The Pareto archived evolution strategy : A new baseline algorithm for multi-objective optimization," *IEEE International Con-*

- ference on Computation*, pp.98-105. 1999.
- [16] Kumar, N. and K. Shanker, "A genetic algorithm for FMS part type selection and machine loading," *International Journal of Production Research*, Vol.38(2000), pp.3861-3887.
- [17] Michalewics, Z., "Genetic Algorithms+Data Structures=Evolution Programs," Second, Extended Edition, Springer-Verlag, 1992.
- [18] Modi, B.K. and K. Shanker, "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*, Vol.33(1994), pp.1791-1816.
- [19] Nayak, G.K. and D. Acharya, "Part type selection, machine loading and part type volume determination problem in FMS planning," *International Journal of Production Research*, Vol.36(1998), pp.1801-1824.
- [20] Parks, G.T. and I. Miller, "Selective breeding in a multiobjective genetic algorithm," In A.E.E. et al. (Ed.). *Parallel Problem Solving from Nature PPSN V*, Berlin, pp.250-259, Springer, 1998.
- [21] Rudolph, G., "On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set," In *proceedings of the Fifth IEEE Conference on Evolutionary Computation*, pp.511-516, IEEE Press, Piscataway, New Jersey, 1998.
- [22] Schaffer, J.D., "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," In *Genetic Algorithms and their Applications : Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum, 1985, pp.93-100.
- [23] Srinivas, N. and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, Vol.2, No.3(1994), pp.221-248.
- [24] Stecke, K.E., "Formulation and solution of nonlinear integer production planning problem for flexible manufacturing systems," *Management sciences*, Vol.29(1983), pp.273-288.
- [25] Stecke, K.E. and N. Raman, "FMS planning decisions, operating flexibilities, and system performance," *IEEE Transactions on Engineering Management*, Vol.42(1995), pp.82-90.
- [26] Tan, K.C., T.H. LEE and E.F. Khor, "Evolutionary Algorithms for Multi-Objective Optimization : Performance Assessments and Comparison," *Artificial Intelligence Review*, Vol.17(2002), pp.253-290.
- [27] Tiwari, M.K. and N.K. Vidyarthi, "Solving machine loading problems in a flexible manufacturing system using a genetic algorithm based heuristic approach," *International Journal of Production Research*, Vol.38(2000), pp.3357-3384.
- [28] Toffolo, A. and E. Benini, "Genetic Diversity as an Objective in Multi-Objective Evolutionary Algorithms," *Evolutionary Computation*, Vol.11, No.2(2003), pp. 151-167.
- [29] Veldhuizen, D.A.V. and G.B. Lamont, "Evolutionary computation and convergence to a pareto front," In J.R. Koza, W. Banzhaf, K. chellapilla, K. Deb, M. Dorigo, D.B. Fogel, M.H. Garzon, D.E. Goldberg, H. Iba, and R. Riolo (Eds.), *Genetic Programming 1998 : Proceedings of the Third Annual Con-*

- ference*, San Francisco, CA, pp.22-25. Morgan Kaufmann.
- [30] Veldhuizen, D.A.V. and G.B. Lamont, "Multiobjective evolutionary algorithms : Analyzing the state-of-the-art," *Evolutionary Computation*, Vol.8, No.2(2000), pp.125-147.
- [31] Zitzler, E., K. Deb and L. Thiele, "Comparison of multiobjective evolutionary algorithms : Empirical results," *Evolutionary Computation*, Vol.8, No.2(2000), pp. 173-195.
- [32] Zitzler, E. and L. Thiele, "Multiobjective evolutionary algorithms : A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, Vol.3, No.4(1999), pp.257-271.
- [33] Zitzler, E., M. Laumanns and L. Thiele, "SPEA2 : Improving the Strength Pareto evolutionary Algorithm," Technical Report 103, Computer Engineering and Networks Laboratory(TIK), Swiss Federal Institute of Technology(ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May, 2001.