

진화알고리즘을 이용한 유연조립시스템의 다목적 공정계획

신경석 · 김여근[†]

전남대학교 산업공학과

A Multiobjective Process Planning of Flexible Assembly Systems with Evolutionary Algorithms

Kyoung Seok Shin · Yeo Keun Kim

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

This paper deals with a multiobjective process planning problem of flexible assembly systems(FASs). The FAS planning problem addressed in this paper is an integrated one of the assignment of assembly tasks to stations and the determination of assembly routing, while satisfying precedence relations among the tasks and flexibility capacity for each station. In this research, we consider two objectives: minimizing transfer time of the products among stations and absolute deviation of workstation workload(ADWW).

We place emphasis on finding a set of diverse near Pareto or true Pareto optimal solutions. To achieve this, we present a new multiobjective coevolutionary algorithm for the integrated problem here, named a multiobjective symbiotic evolutionary algorithm(MOSEA). The structure of the algorithm and the strategies of evolution are devised in this paper to enhance the search ability. Extensive computational experiments are carried out to demonstrate the performance of the proposed algorithm. The experimental results show that the proposed algorithm is a promising method for the integrated and multiobjective problem.

Keywords: flexible assembly systems, process planning, multiobjective optimization, multiobjective evolutionary algorithm.

1. 서론

유연조립시스템(Flexible Assembly System; FAS)은 조립기능을 수행하는 로봇, 조립품의 운반기능과 저장기능을 수행하는 자동자재 처리설비, 그리고 이들 설비를 제어하는 컨트롤 시스템으로 구성되어 여러 종류의 제품을 자동으로 조립할 수 있는 시스템을 말한다(Lee and Johnson, 1991). 이러한 시스템은 자동차 조립공정이나 SMT(surface mount technology)라인, PCB(printed circuit board) 조립시스템 등 많은 산업분야에 적용되고 있다. 또한 카세트, 자명종, 볼 베어링, 컴퓨터 디스켓 등 고객의 요구가 다양한 제품을 생산하는 시스템에서 폭넓게 사용되고 있다(Groover, 2001).

본 연구에서 FAS의 공정계획문제는 제품들을 조립함에 있어 각 제품별로 수행할 각각의 작업들을 작업장에 할당하고,

각 제품의 조립경로를 결정하는 문제로 정의된다. FAS 공정계획문제는 흔히 많은 제약과 결정변수를 가지며, 이로 인해 복잡도가 높은 문제이다(Lee and Johnson, 1991; Sawik, 1997). 이때 추구하는 목적으로는 조립제품의 이동시간 최소화, 총 가공시간 최소화, 그리고 작업장 작업부하의 균형과 같은 다양한 목적이 있을 수 있다. 또한 이들 목적은 서로 상충될 수 있으며, 이 경우 서로 상충되는 목적들을 동시에 최적화하는 해는 일반적으로 존재하지 않는다. 이러한 상황에서 의사결정자에게 다양한 파레토 최적해(Pareto optimal solutions)를 제안하고, 이들 중에서 상황에 적합한 해를 선택하도록 하는 것도 문제 해결의 좋은 방법이다. 여기서 파레토 최적해란 모든 목적들에 대해 이 해보다 더 우수한 해가 존재하지 않는 해로 비지배해(nondominated solutions)라 부르기도 한다. 해공간이 넓고 복잡도가 높은 조합최적화 문제에서는 파레토 최적해를 구하

이 논문은 2004년도 전남대학교 특별연구사업비 지원에 의해 연구되었음.

[†] 연락저자 : 김여근 교수, 500-757 광주광역시 북구 용봉동 300 전남대학교 산업공학과, Fax : 062-530-1789, E-mail : kimyk@chonnam.ac.kr
2005년 1월 접수; 2005년 4월 수정본 접수; 2005년 5월 게재 확정.

기가 쉽지 않다. 이후 본 연구에서는 용어의 혼동을 피하기 위해 파레토 최적해는 실제 파레토 최적해를 의미하고, 비지배해는 파레토 최적해의 여부와 상관없이 현재 구한 대안해들 중에서 지배당하지 않는 해를 나타내기로 한다.

본 연구에서는 다목적성을 갖는 FAS 공정계획에서 다양하고 좋은 공정계획을 찾는 방법론을 제안한다. 다목적 문제의 해결방법론은 의사결정자의 선호가 다를 수 있어 가능한 한 다양하고 좋은 비지배해들을 제시할 수 있어야 한다. 진화알고리즘은 여러 해로 구성된 모집단을 운영함으로써 다양한 대안해를 동시에 고려할 수 있는 장점을 가지고 있다(Tan *et al.*, 2002). 다목적 문제에 적용된 진화알고리즘을 다목적 진화알고리즘(Multi-Objective Evolutionary Algorithm; MOEA)이라 부른다.

본 연구에서 다루는 FAS 공정계획문제는 제품들의 각 작업을 작업장에 할당하는 작업할당과 제품의 작업순서 결정문제의 두 부분문제로 구성된 통합문제이다. 이러한 통합문제를 분리하여 순차적으로 해결하지 않고 동시에 해결하는 방법론으로 흔히 진화알고리즘의 한 형태인 공생 진화알고리즘이 적용되고 있다(Potter, 1997; Kim, *et al.*, 1999, Kim, *et al.*, 2000; Kim, *et al.*, 2001). 공생 진화알고리즘은 생물계에서 공생(symbiosis) 관계를 갖는 서로 다른 종들이 상호작용하고 상호 적응하며, 공진화하는 과정을 모방한 확률적 탐색기법이다(Potter, 1997). 공생 진화알고리즘에서는 공생관계를 갖는 여러 종들이 각각 모집단을 구성하고, 각 모집단은 다른 모집단과 상호 적응해 나간다. 이 때 각 모집단은 해결하고자 하는 하나의 완전한 문제에 대한 부분문제를 나타낸다. 본 연구에서는 다목적 FAS 공정계획문제를 해결하기 위해 공생 진화알고리즘을 적용한 새로운 알고리즘을 제안한다. 이 알고리즘에서는 문제의 해를 하나의 개체로 표현하는 기존 MOEA 방법과는 달리 개체를 부분개체로 분리하여 여러 모집단을 운영함으로써 다양하고 우수한 비지배해를 효율적으로 탐색할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 본 연구에서 다루는 FAS의 다목적 공정계획을 2장에서 소개하고, 3장에서는 제안한 알고리즘의 개념과 절차를 설명한다. 4장에서는 알고리즘 적용에 요구되는 유전요소들을 설명하고, 5장에서는 제안한 알고리즘의 성능분석을 위한 실험과 결과들을 비교분석한다. 마지막으로 6장에서 연구성과를 요약하고 결론을 맺는다.

2. FAS 공정계획 문제

2.1 FAS의 구조와 특성

FAS는 일반적으로 시스템의 배치, 자재흐름유형, 품종의 다양성과 생산량 등에 따라 여러 형태로 분류할 수 있다(Sawik, 1999). 본 연구에서 다루는 FAS의 구조와 특징은 개략적으로

다음과 같고, 이에 대한 자세한 사항은 Kim *et al.*(2004)을 참고할 수 있다.

첫째, 전통적인 컨베이어 흐름조립 라인과는 달리 FAS에서 작업장 간 조립작업의 흐름은 직선적인 단방향 이동 외에 후진, 건너뛰기 등 여러 형태의 작업장 이동이 가능하다

둘째, 다기능성 또는 범용기계를 사용함으로써 각 작업장에서 여러 종류의 작업이 가능하여 높은 유연성을 갖는다.

셋째, 각 작업장에서는 작업을 위한 자동조립기계와 관련 설비의 설치로 요구되는 작업장 공간으로 인해 할당되는 작업의 수에 제약이 받는다. 이와 같이 작업장에서 수행할 수 있는 작업의 수와 관련된 작업장 크기를 유연성 용량(flexibility capacity) (Lee and Stecke, 1996) 또는 작업장 공간용량(staging capacity) (Park and Kim, 1995)이라고 한다. 본 연구에서는 공간용량이라 부르기로 한다.

넷째, 본 연구에서는 공구와 관련된 제약은 무시한다. FAS에서 사용되는 조립용 공구는 유연제조시스템(Flexible Manufacturing System; FMS)의 절삭용 공구보다 소모가 적어 공구관리 및 공구교환에 따른 노력과 시간이 상대적으로 적게 요구되어 공구에 대한 제약은 미미하다고 볼 수 있다.

다섯째, 동일한 작업이라 할지라도 각 작업장에 설치된 조립기계 또는 관련 장비의 성능차이로 인해 작업장마다 소요되는 조립시간이 다르다고 본다.

여섯째, 전통적인 혼합모델 조립라인의 작업할당과는 달리, 같은 작업이 제품에 따라 다른 작업장에서 행해질 수 있다. 이를 작업의 중복할당(duplicate assignment)이라 부른다(Sawik, 1999).

본 연구에서 사용되는 제품들의 조립작업의 선행관계는 <Figure 1>과 같은 네트워크 표현에 의한 결합선행 공정도를 사용한다.

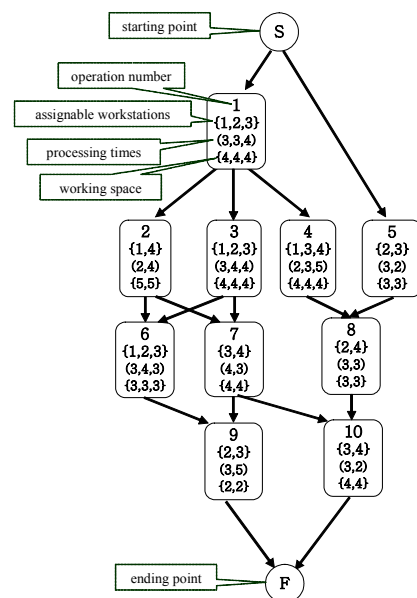


Figure 1. Network for FAS process planning.

네트워크에서 노드(node)는 작업을, 호(arc)는 선행관계를 나타낸다. 그리고 각 노드에는 해당 작업을 행할 수 있는 작업장, 작업시간, 그리고 이 작업을 위해 필요한 공간이 표시된다. 노드 S, F는 각각 시작 노드와 끝 노드를 나타내는 가상(dummy) 노드이다.

이후 본 연구에서 다루는 FAS 공정계획문제와 제안한 알고리즘의 설명을 위해 사용되는 기호를 정리하면 다음과 같다

- i : 조립작업의 첨자, $i=1, 2, \dots, l$.
 j : 작업장의 첨자, $j=1, 2, \dots, m$.
 k : 제품의 첨자, $k=1, 2, \dots, n$.
 $I(j)$: 작업장 j 에 할당된 작업들의 집합.
 a_{ij} : 작업장 j 에 작업 i 가 할당될 때 요구되는 공간용량.
 b_j : 작업장 j 에 할당된 작업들의 총 소요 공간용량,

$$\text{즉 } b_j = \sum_{i \in I(j)} a_{ij}.$$

- B_j : 작업장 j 의 최대 허용 공간용량.
 W_j : 작업장 j 에 할당된 조립작업 부하량(조립 작업시간의 합).
 \overline{W} : 작업장의 평균 작업 부하량(전체 작업 부하량을 작업장 수로 나눈 값).

2.2 문제상황 및 목적함수

본 연구에서 다루는 공정계획은 작업의 선행제약, 작업장 공간제약이 주어진 조건하에서 각 제품의 작업할당과 조립경로를 동시에 결정하는 것이다. 본 연구에서는 다음과 같은 가정 및 제약을 둔다.

1. 각 조립작업은 적어도 하나의 작업장에 할당되어야 한다.
2. 각 작업들은 한 개 이상의 작업장에 할당 가능하다(중복 할당 가능).
3. 작업장에 할당된 작업들의 소요 공간용량의 합은 그 작업장의 최대 허용 공간용량을 넘을 수 없다.
4. 조립라인의 조립기계와 장비, 자동자재 처리설비 등은 이미 결정되어 있다.
5. 조립품의 작업장 간 이동시간은 주어져 있다.
6. 각 작업들의 조립시간은 준비시간과 공구교환시간을 포함한다. 그리고 기계 또는 작업의 불량, 고장은 없다고 본다.

본 연구에서는 작업장의 공간제약, 제품의 작업순서와 관련이 있으면서 서로 상충적인 관계에 있는 다음 두 가지 목적을 사용한다.

- 목적 1: 조립제품의 총 이동시간의 최소화
 목적 2: 작업장 부하 절대편차(absolute deviation of worksta-

tion workload; ADWW)의 최소화

조립제품의 이동시간은 각 조립제품의 작업장 간 이동시간의 합이다. FAS에서는 흔히 작업처리시간이 짧고, 작업장 간 이동의 무방향성으로 인해 조립품의 이동시간이 작업처리시간보다 상대적으로 많이 소요되는 경향이 있다. 따라서 조립품의 이동시간은 전체 시스템의 수행도 평가에 있어서 중요한 요소가 될 수 있다. 이동시간을 최소로 하기 위한 공정계획은 다른 작업장 간의 부하 불균형이 발생하여 시스템 효율을 저하시킬 수 있고, 반대로 작업장 부하 편차의 최소화를 기준으로 한 공정계획은 작업들을 여러 작업장으로 분산시켜 작업을 위한 제품의 작업장 간 이동시간을 크게 할 수 있다. 따라서 두 목적은 서로 상충적인 관계에 있다. 목적 2의 작업장 부하 절대편차는 다음과 같은 식으로 표현된다.

$$ADWW = \sum_j^m |W_j - \overline{W}|$$

FAS와 관련된 연구로 본 연구에서 다루는 문제와 유사한 연구들은 다음과 같다. Lucertini *et al.*(1996)은 부품의 작업장 간 이동횟수를 최소화하는 문제에 대해 작업장의 작업부하를 고려하지 않은 유형과 작업장의 작업부하를 고려한 유형의 문제를 다루었다. Khouja *et al.*(2000)은 작업장 평활화를 목적으로 각 로봇의 성능을 고려한 작업할당문제를 다루었지만 조립순서문제는 고려하지 않고 있다. Markus and Robert(1989)는 제품별 조립공정도를 이용하여 총 처리시간 최소화를 위한 조립순서 결정과 일정계획문제를 해결하는 발견적 기법을 제안하였다. 이상의 연구들(Lucertini *et al.*, 1996; Khouja *et al.*, 2000; Markus & Robert, 1989)은 모두 단일목적에 갖는 FAS 문제를 다루고 있다.

다목적 문제를 다룬 대표적인 연구로 Sawik(1999, 2000)이 있다. 이 연구에서는 최대 작업 부하량과 최대 이동시간의 최소화를 목적으로 두고, 작업할당과 조립경로를 동시에 결정할 수 있는 정수계획모형을 제시하였다. 그러나 이 정수계획모형은 많은 변수와 제약으로 인해 규모가 큰 문제를 해결하는 데 한계가 있다. 또한 이들 연구에서는 가능 조립경로가 네트워크 상태로 주어진 것이 아니라 한정된 몇 개의 경로만을 대상으로 하고 있다. 이 외에 Ammons *et al.*(1985)은 FAS의 작업부하 평활화와 작업들의 작업장 방문횟수의 두 목적을 갖는 문제를 다루었고, Park and Kim(1995)은 한 방향을 갖는 FAS에서 부품선택의 가중합의 최대화와 최대 작업 부하량의 최소화를 목적으로 작업장의 공간용량을 고려한 부품선택과 작업할당 문제를 다루었다. 이 중 Park and Kim(1995), Sawik(1999, 2000)은 작업장의 공간용량을 고려한 반면 Ammons *et al.*(1985)은 고려하지 않고 있다. 다목적에 다루는 이들 연구들(Sawik, 1999, 2000; Ammons *et al.*, 1985; Park and Kim, 1995)은 모두 목적들을 일차 결합하여 단일 목적문제로 변환하여 해결하고 있다.

3. FAS의 다목적 공정계획을 위한 진화알고리즘

3.1 다목적 진화알고리즘

서로 상충되는 목적을 갖는 다목적 문제에서는 모든 목적에 최적인 해는 일반적으로 존재하지 않는다. 이 경우 의사결정자의 선호도와 상황에 따라 여러 대안 중 하나를 선택해야 한다. 하지만 의사결정자의 가치기준과 선호도를 계량화하여 하나의 최적 대안을 찾기는 쉽지 않다. 따라서 다목적 문제를 해결하는 하나의 방법으로 비지배해를 가능하면 많고 다양하게 찾아서 의사결정자에게 제시하는 것을 그 목표로 한다.

진화알고리즘은 한 개의 해를 개선시켜 하나의 비지배해를 찾는 전통적인 기법과는 달리, 다수 개의 해(개체)들로 이루어진 모집단을 운용하여 한번의 알고리즘 수행으로 다양한 비지배해를 구할 수 있다. 이러한 특성으로 인해 진화알고리즘은 다목적 문제를 해결하는 효율적인 도구로 인식되어 왔으며, 그에 관한 여러 연구가 이루어졌다(Tan *et al.*, 2002).

다목적 진화알고리즘(MOEA)에 관한 최초의 연구는 VEGA(Vector Evaluated GA) (Schaffer, 1985)로, 각 목적에 대한 모집단을 운영하여 각 목적별로 모집단을 진화시키는 방법이다. 하지만 이 방법은 다양한 비지배해가 아닌 각 목적에 편중되는 비지배해를 찾는 단점을 보였다. Goldberg(1989)는 다양한 비지배해를 탐색하기 위해 적소(適所; niche) 개념에 의한 적응도 부여방법을 제안하였다. MOGA(MultiObjective Genetic Algorithm) (Fonseca and Fleming, 1993), NSGA(Non-dominated Sorting Genetic Algorithm) (Srinivas and Deb, 1994), NPGA(Niched Pareto Genetic Algorithm) (Horn *et al.*, 1994) 등은 적소 개념을 구현한 대표적 알고리즘에 속한다.

MOEA에 관한 비교적 최근의 연구로는 SPEA(Strength Pareto Evolutionary Algorithm) (Zitzler and Thiele, 1999), PAES(Pareto-Archive Evolution Strategy) (Knowles and Corne, 1999), NSGA-II(Deb *et al.*, 2000), SPEA2(Zitzler *et al.*, 2001) 등이 있다. 이들 연구의 특징은 각 알고리즘에 맞는 엘리티즘 전략, 즉 발견된 좋은 해를 보관하고 이용하는 전략을 채용하여 탐색 효율성을 높이고 있다. SPEA와 SPEA2는 발견된 좋은 해를 보관하기 위한 별도의 모집단을 운영하여 적응도 평가와 선택에 이용한다. NSGA II와 PAES는 μ 개의 개체를 갖는 모집단에서 λ 개의 자손을 생산하여, 모집단과 생산된 자손개체 ($\mu + \lambda$)개에서 우수한 μ 개의 개체를 선택하는 방법인 ($\mu + \lambda$)-ES(Evolutionary Strategy)를 응용하는데, 특히 PAES는 ($\mu + \lambda$)-ES의 특별한 형태인 (1+1)-ES를 사용하고 있다. MOEA의 각 알고리즘에 관한 특징과 기법, 그리고 평가척도에 관한 자세한 내용은 Coello(1999), Tan *et al.*(2002)를 참조할 수 있다.

기존 MOEA에 관한 연구는 모두 전체문제를 하나의 개체로 표현하여 이들 개체로 이루어진 모집단을 운영하고 있다. 본 연구에서는 전체문제를 부분문제로 나누어 각 부분문제에 맞는 개체로 이루어진 모집단을 다수 개 운영하는 공생 진화알

고리즘을 제안한다. 복잡도가 높고 부분문제로 이루어진 통합 문제에 효율적인 공생 진화알고리즘의 개념을 다목적 문제로 확장하여 적용한 이 알고리즘을 다목적 공생 진화알고리즘(Multi Objective Symbiotic Evolutionary Algorithm; MOSEA)으로 명명한다.

3.2 제안한 알고리즘의 개념과 절차

MOSEA의 전체 구조는 <Figure 2>와 같이 부분문제를 표현하는 여러 모집단이 분리되어 있는 형태를 갖는다. 그림에서 $P[q]$ 는 세대 t 에서 q 번째 모집단을 나타낸다. 모집단 $P[q]$, $q = 1, 2, \dots, n+1$ 은 모든 제품의 작업할당을 나타내는 하나의 작업할당 모집단 $P[n+1]$ 과 각 제품의 조립경로를 나타내는 n 개의 조립경로 모집단 $P[q]$, $q = 1, 2, \dots, n$ 로 구성된다. 작업할당문제에서 개체표현은 모든 제품의 작업할당을 하나의 개체에 나타내는 형태로서, 이 문제를 위해 하나의 모집단으로 운영한다. 그렇지 않고 제품별로 모집단을 운영하면 공간제약을 충족시키기가 쉽지 않아 비효율적인 탐색이 될 수 있다. 한편, 경로문제의 모집단은 모든 제품의 경로를 하나의 개체로 표현하여 하나의 모집단으로 운영할 수도 있고, 제약을 처리하기 위한 특별한 장치 없이 제품별로 개체를 표현, 제품별 모집단을 운영할 수도 있다. 부분문제의 분리수준결정은 문제가 갖는 특성에 따라 달라질 수 있으며, 이는 알고리즘 성능에 영향을 줄 수 있다(Potter, 1997).

제안한 MOSEA의 개념과 특징을 다음과 같이 정리할 수 있다.

첫째, MOSEA에서 부분문제에 대한 각 모집단은 분리된 종들이 공생하는 과정을 모방한 공생 진화알고리즘이 수행된다. 즉 통합문제를 이루는 부분문제 각각을 하나의 종으로 간주하여 모집단을 구성한다. 이들 모집단의 개체는 다른 모집단의 개체와 짝을 지어 적응도를 부여받고, 이 적응도에 기초하여 독립적으로 진화한다. 이러한 공생진화는 병렬탐색을 강화하여 넓은 해공간의 효율적 탐색을 유도한다.

둘째, 각 모집단의 진화는 ($\mu + \lambda$)-ES를 사용하여 독립적으로 진화한다. 즉, 부분 모집단은 생산된 자손과 부모와의 합집합에서 적응도를 기준으로 다음 세대 모집단을 구성하게 된다. 이 방법은 엘리티즘(elitism) 전략의 하나로 부모세대에서의 우수한 개체가 사라지는 것을 방지하여, 우수한 형질이 다음 세대에 전파되어 수렴속도를 향상시키고자 하기 위함이다(<Figure 3> 참조).

셋째, 모집단 TNS(Temporary Non-dominated Set)는 각 모집단이 진화함에 따라 매 세대마다 발견된 비지배해 개체결합을 보관하기 위한 임시 모집단이다. MOSEA에서는 다루는 문제의 완전한 해를 표현하는 개체가 아닌 부분문제 개체로 이루어진 다수의 모집단을 운영하므로 찾아진 비지배해 개체결합

을 따로 보관하여 유지할 필요가 있다(<Figure 2>와 <Figure 3>).

넷째, 각 모집단에서 개체 적응도는 파트너 선택전략에 의해 계산된 목적함수 벡터에 대해 비지배순위(rank)와 밀집거리를 고려하여 평가된다. 이 방법은 NSGA-II(Deb *et al.*, 2000)에서 사용된 개념을 본 알고리즘에 맞게 채용한 방법으로 3.3절의 적응도 평가부분에서 자세하게 설명한다.

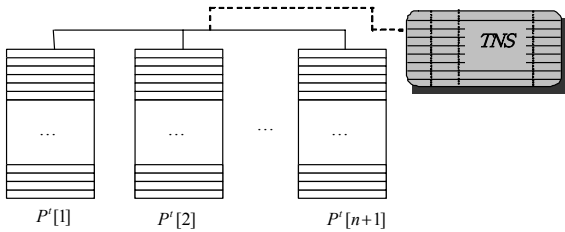


Figure 2. Structure of MOSEA.

MOSEA의 구체적인 절차는 다음과 같다.

단계 1 (초기화)

- 단계 1.1: $TNS = \emptyset$, $t = 1$, $q = 1$ 로 둔다.
- 단계 1.2: 각 부분문제를 위한 부분 모집단 $P'[q]$ ($q = 1, 2, \dots, n+1$)를 임의로 생성한다.
- 단계 1.3: 각 부분 모집단($P'[q]$, $q = 1, 2, \dots, n+1$)에 대해, 상대 모집단($P'[s]$, $s = 1, 2, \dots, n+1$, $s \neq q$)에서 임의로 선택한 개체와 짝지어 목적함수 벡터를 계산하고, 적응도를 평가한다. 발견된 비지배해 개체결합을 TNS 에 복사한다.

단계 2 (자손생산)

- 단계 2.1: $P'[q]$ 로부터 $|P'[q]|$ 개의 자손을 생산하여, 자손 모집단 $O'[q]$ 를 만든다. 여기서 선택은 토너먼트 선택을 사용한다.

단계 3 (적응도 평가)

- 단계 3.1: $O'[q]$ 에 있는 각 개체는 모집단 $P'[s]$ ($s = 1, 2, \dots, n+1$, $s \neq q$)에서 비지배순위가 1인 개체를 임의로 선택, 짝지어 목적함수 벡터를 계산한다 (3.3.1절 참조).

- 단계 3.2: 목적함수 벡터를 이용하여 $O'[q]$ 개체의 적응도를 구한다(3.3.2절 참조).

- 단계 3.3: 단계 3.1의 개체결합에서 비지배해를 TNS 에 복사하고, TNS 를 비지배해의 집합이 되도록 갱신한다.

단계 4 (모집단 진화)

- 단계 4.1: $P'[q] \cup O'[q]$ 에서 적응도가 가장 높은 $|P'[q]|$ 개의 개체를 선택하여 다음 세대 모집단 $P^{t+1}[q]$ 를 구성한다.

- 단계 4.2: $q < n+1$ 이면, $q \leftarrow q+1$ 로 두고 단계 2로 간다.

단계 5 (종료조건)

- 단계 5.1 종료조건을 만족하면 끝내고, 그렇지 않으면 $t \leftarrow t+1$, $q = 1$ 로 두고 단계 2로 간다.

각 모집단 개체의 목적함수 벡터 계산을 위해서는 공생 파트너가 요구된다. 이는 단계 1.3과 단계 3.1에 해당한다. 단계 1.3 초기화 단계에서 공생 파트너는 상대 모집단에서 임의로 개체를 선택한다. 반면 단계 3.1 진화단계에서는 상대 모집단의 부모개체의 비지배 개체 중에서 임의로 선택하는 전략을 사용한다. 이는 前 세대의 우수한 개체를 다음 세대를 위한 개체의 공생 파트너로 선택함으로써 탐색성능을 보다 향상시키는 역할을 할 것으로 기대된다. <Figure 3>은 한 부분 모집단이 진화하는 과정으로 단계 2에서 단계 4.2까지의 과정을 보여주고 있다. 단계 5의 종료조건으로는 진화세대 수, 진화과정에서 새로이 생산된 개체 수, 일정 수준 이상 해를 개선시키지 못한 세대 수 또는 생산개체 수 등이 흔히 사용된다.

결국 MOSEA의 개념은 공생 진화알고리즘이 갖는 병렬탐색능력은 다양한 해공간의 탐색을 가능하게 하고, 적응도 평가 시 공생 파트너 선택전략과 모집단 진화에서의 $(\mu + \lambda)$ -ES는 좋은 해의 집중적 탐색을 강화하여 MOEA가 추구하는 우

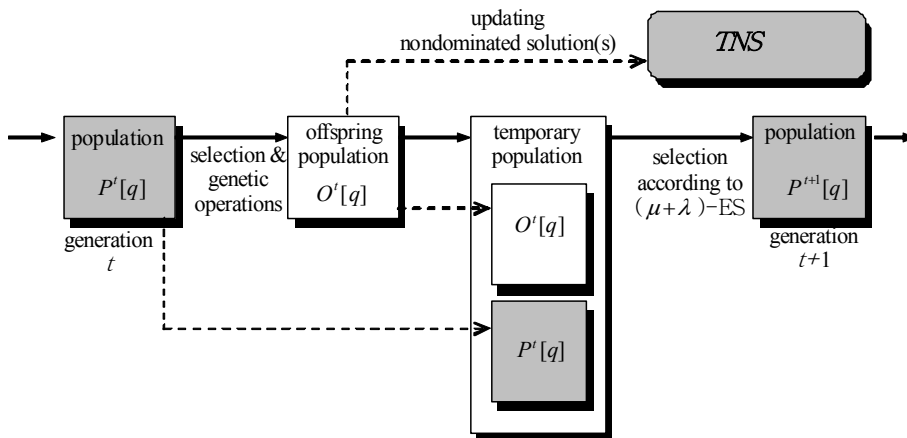


Figure 3. Evolution process of the subpopulation.

수하고 다양한 비지배해를 구할 수 있도록 하는 것이다.

3.3 적응도 평가

앞서 언급했듯이 MOEA에서의 목표는 가능한 우수하고 다양한 비지배해를 탐색하고자 하는 것이다. 따라서 MOEA에서 적응도는 단일목적문제와는 달리, 해의 질적 향상을 위한 수렴성과 다양한 해를 구하기 위한 다양성의 두 요소를 적절히 반영할 수 있어야 한다.

MOSEA에서는 부분문제를 표현하는 여러 모집단을 운영하므로 적응도 평가를 위해서는 먼저 공생 파트너 선택에 의해 각 개체에 대한 목적함수 벡터가 계산되어야 한다. 이렇게 구해진 목적함수 벡터를 기준으로 적응도를 부여하게 된다.

3.3.1 목적함수 벡터 계산

개체를 해석하여 목적함수를 계산할 때 개체의 가능성(feasibility) 여부를 고려하여야 한다. 본 연구에서 다루는 문제의 제약 중 작업 간 순서제약과 대안작업장 제약은 이를 항상 만족하는 개체표현과 유전연산을 채용함으로써 이들 제약을 어기지 않도록 한다. 반면 허용공간에 대한 제약은 그 위반 정도에 따라 벌금함수를 사용하여 처리한다. 제약만족과 상관없이 계산된 2.2절의 목적1과 목적2의 목적함수 값을 각각 f_1' , f_2' 로 둔다. 그리고 목적함수 벡터 $\mathbf{f}' = (f_1', f_2')$ 에 제약을 고려하여 벌금이 부과된 새로운 목적함수 벡터를 $\mathbf{f} = (f_1, f_2)$ 로 둔다. 변환된 목적함수 벡터 \mathbf{f} 를 이용하여 적응도를 평가한다. 그 변환식은 다음과 같다.

$$f_g = f_g' + c \sum_j PS_g(j)^\alpha, \quad g=1,2.$$

여기서,

$$PS_g(j) = \begin{cases} \frac{(b_j - B_j)f_g'}{B_j}, & \text{if } b_j > B_j \\ 0, & \text{otherwise} \end{cases}$$

이다. $PS_g(j)$ 는 목적 g 에 대해 공간용량의 위반에 따른 벌금함수로서, 목적 g 에 대해 초과된 공간량의 정도를 벌금으로 부과한다. 상수 c, α 는 파라미터이다.

3.3.2 적응도 계산

다양한 비지배해를 찾는 다목적 진화알고리즘에서의 적응도는 한 개의 해를 찾는 단일목적 진화알고리즘과는 달리 해의 수렴과 다양한 탐색을 동시에 고려하여 평가되어야 한다. 이를 위해 본 연구에서는 개체 i 의 적응도를 다음의 평가값 $eval_i$ 를 이용하여 계산한다.

$$eval_i = r(i) + \frac{1}{1 + d(i)}$$

여기서 $r(i), d(i)$ 는 개체 i 의 비지배순위와 동일한 비지배

순위에 있는 다른 개체들과의 밀집거리를 각각 나타낸다. 따라서 개체의 평가값은 비지배순위가 높을수록, 밀집거리가 클수록 작은 값을 갖게 된다. 개체 i 의 적응도는 계산된 평가값이 작을수록 큰 값을 갖도록 하였다. 이러한 방법은 해의 수렴을 위해 개체의 비지배순위를 우선적으로 고려하여 적응도를 부여하고, 비지배순위가 같을 경우 밀집거리가 큰 개체에 더 높은 적응도가 부여되어 다양한 해공간을 탐색할 수 있도록 한다. 비지배순위를 구하는 데 있어 본 연구에서는 좀더 계산이 적게 요구되는 Deb *et al.*(2000)의 방법을 사용한다. 이 방법은 각 개체 간의 지배관계 정보를 구하여 보관한 후, 이를 이용하여 비교적 적은 계산시간으로 비지배순위를 구할 수 있다. 개체의 밀집거리 또한 Deb *et al.*(2000)에 의해 채용된 방법을 사용한다. 비지배순위와 밀집거리를 구하는 이들 절차는 <부록>으로 두었다.

4. 진화요소

MOSEA를 구현하기 위해서는 각 부분문제의 유전표현 표현된 개체에 적합한 유전연산방법 등이 요구된다. 사용되는 진화요소들은 알고리즘의 성능에 중요한 영향을 주므로 문제의 특성을 잘 반영할 수 있도록 설계되어야 한다.

4.1 유전표현과 초기 모집단 생성

본 연구에서 사용하는 공생 진화알고리즘은 1개의 작업할당 모집단과 n 개의 조립경로 모집단으로 구성된다. 작업할당 모집단의 개체는 $(n \times l)$ 행렬로 표현한다. 행렬에서 행은 제품을, 열은 작업을 나타낸다. 이 표현에서 위치 (k, i) 에 있는 인자 값은 제품 k 의 작업 i 가 할당되는 작업장 번호를 의미한다. 이 표현은 그룹번호 표현의 일종으로 바로 해석(decoding)이 가능하며 중복 표현이 되지 않는다는 장점이 있다(Kim *et al.*, 1997b). <Figure 4>는 작업할당 모집단의 개체표현의 한 예를 보여 주고 있다. 여기서 인자 값 0은 해당 작업이 행하여 지지 않음을 뜻한다. 각 제품에 대해 분리하지 않고 하나의 개체로 표현한 이유는 공간제약조건을 만족하는 해공간탐색이 용이하기 때문이다. 예비실험 결과, 각 제품별로 개체를 분리하여 운영하면, 모집단의 개체 중에서 공간제약을 만족하는 가능해의 수가 적어서 탐색의 효율성이 저하됨을 알 수 있었다.

$k \backslash i$	1	2	3	4	5	6	7	8	9	10
1	1	0	1	3	2	2	3	2	2	4
2	3	1	2	0	2	0	3	2	3	4
3	1	1	1	3	0	2	4	0	3	1

Figure 4. Encoding for task assignment.

작업할당 모집단의 초기 모집단은 중복할당은 허용하지 않고 모든 제품의 동일 작업이 같은 작업장에 할당되도록 구성한다. 이는 초기 모집단에 가능한 많은 가능해 개체를 생성하기 위함이다. 그 구체적 절차는 <부록>을 참조하기 바란다.

조립경로 모집단의 개체는 순열표현으로 나타낸다. 즉, 각 제품에 대해 조립되는 작업순서를 차례로 나타낸다. 이 때 제약은 조립의 선행관계이다. <Figure 1>의 예에서 제품 1을 위한 개체표현이 <Figure 5>와 같이 표현되었다면 제품 1은 (1→2→4→5→3→6→8→7→9→10) 순서로 작업이 이루어짐을 의미한다. 공생 진화알고리즘에서는 각 제품에 대한 작업순서의 개체를 하나로 모아 하나의 모집단을 형성할 수도 있고, 각각 분리하여 각 제품을 위한 순서 모집단, 즉 제품의 수만큼 모집단을 운영할 수도 있다. 본 연구에서는 작업할당을 위한 개체와 달리 조립경로 모집단은 제품의 수만큼 분리하였다. 이와 같은 모집단의 분리는 각 제품별 조립경로를 탐색할 수 있어 다양한 해공간 탐색을 촉진한다. 조립경로는 목적 1의 이동시간과 관련된다. 조립경로를 나타내는 초기 모집단은 아래와 같은 절차에 의해 생성된다.

1	2	4	5	3	6	8	7	9	10
---	---	---	---	---	---	---	---	---	----

Figure 5. Encoding for routing.

<조립경로 초기 모집단의 개체생성 절차>

- 단계 1: 선행 작업이 없는 작업들의 집합 B를 만들고, 비어 있는 순서열을 만든다.
- 단계 2: 만약 B가 비어 있으면 끝내고, 그렇지 않으면 단계 3으로 간다.
- 단계 3: 집합 B에서 임의로 한 작업을 선택하여 순서열의 왼편부터 차례로 삽입한다.
- 단계 4: 선택된 작업은 집합 B에서 삭제하고, 선행 작업이 모두 할당된 작업은 집합 B에 포함시킨다. 단계 2로 간다.

각 제품에 대해 개체를 위 과정을 통해 생성하여 제품의 수만큼 모집단을 구성한다. 위 절차에서 선·후행 제약은 단계 1, 4에서 고려되고, 단계 3의 집합 B에서 임의로 작업을 선택하도록 하여 다양한 개체가 생산될 수 있도록 하고 있다.

4.2 유전연산자

유전연산자는 흔히 교차(crossover)와 돌연변이(mutation)로 나눌 수 있다. 교차는 문제의 특성을 반영하는 유전정보를 부모로부터 추출하여 자손에게 전달하는 역할을 하고, 돌연변이는 모집단의 다양성을 유지하는 역할을 한다. FAS 공정계획 문제를 구성하는 두 부분문제의 개체에 대해 서로 다른 유전연산자를 사용한다. 먼저 작업할당 모집단에서의 교차는 수정구조교차(Kim et al., 1997a)를 사용한다. 그 절차는 다음과 같다.

단계 1: 범위 [1, m]에서 임의의 정수 r를 선택한다.

단계 2: 부모 P1에서 인자 값 0부터 r까지 표현한 인자를 자손 O1의 동일한 위치에 상속한다.

단계 3: 자손 O1의 남은 위치에 부모 P2의 동일 위치에 인자 값이 r+1부터 m까지 표현된 인자를 상속한다.

단계 4: 자손 O1에 비어 있는 위치는 재할당방법을 이용한다. 이 방법은 발견적 기법으로 아래에서 설명한다.

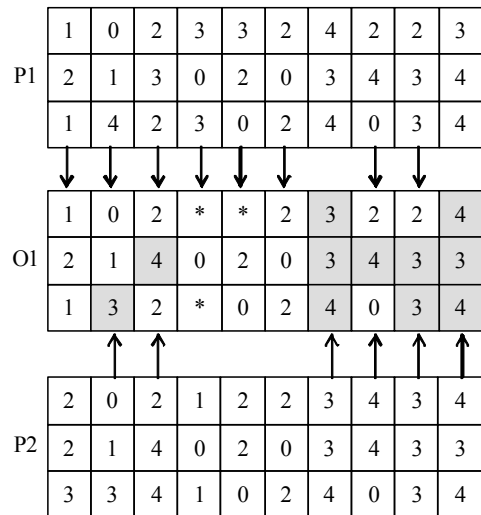


Figure 6. Crossover for task assignment.

자손 O2는 위 단계에서 두 부모의 역할을 바꾸어 생산한다. 이 구조교차는 두 부모가 가지고 있는 위치와 인자 값 정보를 자손에게 상속한다는 특징을 갖는다. <Figure 6>은 m=4, r=2인 경우의 구조교차를 보이고 있다. 여기서 * 표시는 미할당인자이다. 이런 미할당인자들에 대해서는 적절한 재할당기법에 의해 인자 값을 정해주어야 한다. 이들 인자의 할당을 위한 재할당기법은 4.3절에서 다시 자세히 설명한다. O1에서 음영으로 표시된 인자는 P2로부터 상속받은 것이다.

작업할당 모집단의 돌연변이는 돌연변이율에 의해 인자 값 (작업장 번호)을 선택하고, 선택된 인자의 위치에 해당되는 작업을 행할 수 있는 작업장 중에서 임의로 선택하여 인자 값을 바꾸는 방법으로 수행한다.

조립경로 모집단에 대한 유전연산자는 순열표현개체에 적합한 연산자로서 교차의 경우 수정이점교차(Kim et al., 1997a)를 사용한다. 먼저 임의의 절단점을 정한다. 두 절단점에 의하여 나누어진 앞부분, 중간부분, 뒷부분에 있는 인자 중 앞부분과 뒷부분은 한 부모(P1)로부터 자손(O1)의 같은 위치에 복사한다. 그리고 다른 부모(P2)에서 복사된 인자 값을 갖는 인자들을 지운다. P2에 남아 있는 인자들을 개체에서 나타나는 순서대로 O1의 중간부분에 복사한다. <Figure 7>은 이를 보여주고 있다. 자손 O2는 두 부모의 역할을 바꾸어 생산한다. 이 교차방법은 작업의 선행관계를 만족하고(Kim et al., 1997a), 한 부모로부터는 위치와 순서정보를, 다른 부모로부터는 상대적

순서정보를 상속받는다.

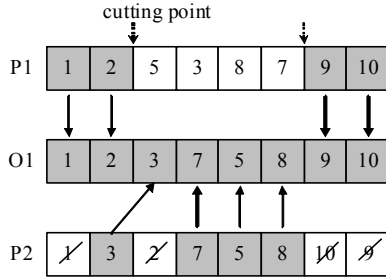


Figure 7. Crossover for routing.

조립경로 모집단에서 돌연변이는 수정된 삽입방법을 사용한다. 이는 돌연변이율에 의해 선택된 인자 값(작업)이 이동할 수 있는 범위 내의 임의의 위치에 삽입하는 방법이다. 여기서 이 범위는 작업의 선행관계에 의해 구해진다. 즉, 선택된 인자 값을 선행 제약을 어기지 않는 임의의 위치에 삽입한다.

4.3 재할당기법

작업할당 모집단의 수정구조교차에서 인자 값이 결정되지 않는 인자(미할당작업)들에 그 값을 부여하는 과정으로, Kim et al.(2004)에서 사용한 발견적 기법을 수정하여 사용한다. 여기서, C 는 미할당된 인자의 집합, 즉 *로 표시된 위치 (k, i) 들의 집합을 나타낸다. 위치 (k, i) 는 제품 k 의 작업 i 를 나타낸다. 이 기법에 사용되는 용어를 먼저 정의하자.

i_k : 제품 k 의 작업 i .

$AS(i_k)$: 제품 k 의 작업 i 가 할당 가능한 작업장 집합.

$FJ(i_k)$: 제품 k 에서 작업 i 의 직선행과 직후행 작업들이 할당된 작업장 집합.

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

단계 1: $AS(i_k)$, $i_k \in C$ 와 $b_j, \forall j$ 를 구한다.

단계 2: 집합 C 에서 임의로 i_k^* 를 선택한다.

단계 3: $R = \{j | b_j + \bar{a}_{i_k^*j} \leq B_j, j \in AS(i_k^*)\}$ 로 둔다. $R \neq \emptyset$ 이면, R 에서 임의로 선택한다. 그렇지 않으면, 즉 $R = \emptyset$ 이면 $AS(i_k^*)$ 에서 임의로 작업장 j^* 를 선택한다.

단계 4: 작업 i_k^* (즉 위치 (k, i))에 작업장 (인자 값) j^* 를 할당한다. 그리고 작업 i_k^* 를 집합 C 에서 삭제한다. $C = \emptyset$ 이면 종료하고, 그렇지 않으면 b_{j^*} 를 갱신하고 단계 2로 간다.

위 절차에서 미할당작업의 작업장은 해당 작업의 할당 가능 작업장을 대상으로 공간제약을 만족하는 작업장이 존재하면 만족하는 작업장 중에서 임의로 선택하고, 공간제약을 만족하는 작업장이 존재하지 않으면 모든 할당 가능 작업장에서 임의로 선택한다.

5. 실험과 결과분석

5.1 실험설계

본 연구에서 제안한 MOSEA의 성능평가를 위해 복잡도가 다른 다양한 문제를 가지고 실험을 수행하였다. 실험문제는 19

Table 1. Test-bed problems

problem	no. of operations	no. of products	no. of workstations	
Thomopoulos	Prob. 1	19	3	4
	Prob. 2	19	3	5
	Prob. 3	19	4	4
	Prob. 4	19	4	5
	Prob. 5	19	4	6
Kilbrudge and Wester	Prob. 6	45	4	5
	Prob. 7	45	4	6
	Prob. 8	45	4	7
	Prob. 9	45	5	5
	Prob. 10	45	5	6
	Prob. 11	45	5	7
Kim	Prob. 12	61	4	6
	Prob. 13	61	4	7
	Prob. 14	61	5	7
	Prob. 15	61	5	8
	Prob. 16	61	6	8
	Prob. 17	61	6	9

개의 작업을 갖는 Thomopoulos 문제(Thomopoulos, 1967), 45개 작업을 가진 Kilbridge와 Wester 문제(Kilbridge and Wester, 1964), 그리고 작업이 61개인 Kim 문제(Kim and Kwak, 1993)의 결합선행공정도를 이용하였다. 여기에 제품의 수와 작업장 수를 달리하여 같은 수의 작업을 갖는 문제라도 서로 다른 복잡도를 갖도록 문제를 생성하였다. 각 작업의 조립시간과 공간용량은 임의로 추출하였으며, 조립시간은 조립이 행해지는 작업장에 따라 서로 다르게 두었다. 또한 작업장간 이동시간은 한 작업의 조립시간에 비해 상대적으로 큰 값을 갖도록 하여 임의로 추출하였다. 각 실험문제에 대한 구체적인 자료는 Shin(2005)을 참조할 수 있다.

<Table 1>은 실험문제로서 각 문제에 대해 작업 수, 제품 수, 작업장 수를 보여준다.

생성된 실험문제에 대해 제안한 알고리즘은 최근의 MOEA 연구로서 비교적 좋은 성능을 보이는 알고리즘인 NSGA-II 그리고 SPEA2와 비교하였다. 각 알고리즘은 모두 JAVA 언어로 구현되었으며, Intel Pentium4 3.2GHz CPU를 장착한 IBM 계열 PC에서 수행되었다. MOSEA, NSGA-II, 그리고 SPEA2에 필요한 각 파라미터는 예비실험을 통해 우수한 성능을 보이는 파라미터로 결정하였다.

벌금과 관련된 파라미터 c 와 α 는 각각 10, 0.5로 설정하였고, 교차율과 돌연변이율은 세 알고리즘 모두 0.6, 0.006을 각각 사용하였다. MOSEA에서 각 부분 모집단의 크기를 100, NSGA-II의 모집단의 수는 200, 그리고 SPEA2의 경우 모집단과 외부 모집단의 수를 각각 100으로 두어 합이 200이 되도록 하였다. 종료조건은 자손의 생산개수로 두고 15,000개의 자손이 생산되면 종료하도록 하였고, 종료 후 얻어진 비지배해 집합을 비교대상으로 삼았다.

5.2 실험결과와 분석

비지배해 집합을 찾고자 하는 MOEA는 다양한 측면에서 그 성능을 평가할 수 있다. 예를 들면, 파레토 최적해로의 수렴속도, 모집단의 다양성 유지, 최종적으로 얻은 비지배해 집합의 파레토 최적해와의 근사성, 구해진 비지배해 집합의 크기(비지배해 개체 수), 각 알고리즘에서 구해진 비지배해 집합들의 상호 지배관계, 비지배해 집합의 분포영역 등이 대표적인 평가를 위한 지표들이다. 이들 지표들을 계량화하기 위해 많은 평가척도들이 개발되어 사용되었다. 하지만 각 평가척도는 제한된 측면만을 반영하는 척도에 머물고 있다(Tan *et al.*, 2002). 더욱 많은 측면을 반영할 수 있는 평가척도의 개발은 그 자체로써 향후 연구주제로 인식되고 있다(Deb, 1999).

본 연구에서는 Hyun *et al.*(1998)이 제안한 평가척도를 사용하여 NSGA-II, SPEA2의 성능을 비교하였다. 사용된 척도는 각 알고리즘에서 얻어진 최종 비지배해 집합에 대한 양적, 질적 비교를 위한 지표로 사용될 수 있다. 설명을 위해 사용되는 기호를 정의하자. 알고리즘 A, B에 의해 얻은 최종 비지배해 집합을 각각 Set_A , Set_B 라 하고, 그 크기를 각각 $N_A(=|Set_A|)$, $N_B(=|Set_B|)$ 라고 하면, N_A , N_B 는 비지배해 집합의 개체 수를 뜻하므로 알고리즘 A와 B의 양적 지표가 된다. 하지만 이 N_A , N_B 의 지표는 단순한 개체의 숫자이므로 지배관계, 즉 질적인 비교에 대해서는 어떤 정보도 주지 못한다. 따라서 질적 비교를 위해 Set_A , Set_B 의 합집합에서 새로이 추출한 비지배해 집합 Set_{AB} 를 만들고 그 크기를 $N_{AB}(=|Set_{AB}|)$ 라고 두자. Set_{AB} 에서 Set_A 에 속한 개체 수, Set_B 에 속한 개체 수를 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 의

Table 2. Performance comparison of MOSEA and NSGA-II

problem	N_M	N_N	N_{MN}	N_{MMN}	N_{NMN}	significance probability
Prob. 1	8.4	5.3	6.5	3.3 (51.3%)	3.2 (48.7%)	0.779
Prob. 2	7.6	5.8	7.2	3.6 (49.3%)	3.7 (50.7%)	0.873
Prob. 3	9.5	6.5	8.3	4.7 (57.0%)	3.6 (43.0%)	0.160
Prob. 4	8.3	6.4	7.7	4.1 (53.9%)	3.5 (46.1%)	0.465
Prob. 5	10.8	7.0	9.3	5.1 (55.0%)	4.2 (45.0%)	0.329
Prob. 6	13.7	11.5	16.2	12.5 (77.0%)	3.7 (23.0%)	0.000*
Prob. 7	15.8	16.6	19.3	14.2 (73.4%)	5.1 (26.6%)	0.000*
Prob. 8	14.5	15.6	18.5	14.2 (76.6%)	4.3 (23.4%)	0.000*
Prob. 9	15.3	12.5	17.7	13.8 (77.8%)	3.9 (22.2%)	0.000*
Prob. 10	17.5	19.3	21.8	14.9 (68.5%)	6.9 (31.5%)	0.000*
Prob. 11	16.4	19.9	21.9	14.3 (65.2%)	7.6 (34.8%)	0.000*
Prob. 12	19.4	16.7	24.4	17.8 (73.1%)	6.6 (26.9%)	0.000*
Prob. 13	12.5	17.6	19.1	12.1 (63.6%)	6.9 (36.4%)	0.000*
Prob. 14	15.0	22.9	23.9	14.1 (59.0%)	9.8 (41.0%)	0.002*
Prob. 15	14.9	16.0	22.7	12.8 (56.5%)	9.9 (43.5%)	0.006*
Prob. 16	12.7	16.9	19.1	11.6 (60.7%)	7.5 (39.3%)	0.000*
Prob. 17	15.9	18.6	24.7	14.4 (58.3%)	10.3 (41.7%)	0.004*

해 수행된 최종 비지배해의 질적 비교척도로 사용될 수 있다. $N_{A|AB} > N_{B|AB}$ 은 알고리즘 A에 의해 얻어진 비지배해들이 알고리즘 B보다 질적으로 우수하거나 또는 더 많은 비지배해를 구했음을 의미한다.

<Table 2>와 <Table 3>은 위에서 설명한 척도를 이용하여 본 연구에서 제안한 알고리즘 MOSEA와 비교 알고리즘으로 사용된 NSGA-II, SPEA2와의 결과에 대해 위에서 설명한 척도로서 계량화한 값을 각각 보여준다. 표에 나타난 값은 각 문제당 30회 반복 실험의 평균값이다. 또한 MOSEA와 기존 알고리즘과의 유의한 차이가 있는지를 알아보기 위한 t 검정결과의 유의확률을 나타내었다. 각 문제별로 검정통계량을 구한 후 유의확률이 유의수준보다 낮으면 비교한 두 알고리즘은 통계적으로 성능차이가 있음을 의미한다. 유의수준은 5%로 두었다. 표에서 *는 MOSEA가 유의수준에서 비교 알고리즘보다 우수함을 뜻한다. 표에서는 알고리즘 MOSEA, NSGA-II, 그리고 SPEA2를 간단히 M, N, S로 각각 표시하였다.

<Table 2>와 <Table 3>으로부터 MOSEA는 NSGA-II, SPEA2 두 알고리즘과의 비교에서 모두 유사한 결과를 보였다. 먼저 양적 척도(N_M, N_N)의 경우 문제에 따라 차이가 있어, 양적인 면에서 알고리즘의 우열은 판단하기 어려웠다. 하지만 질적 척도(N_{MMN}, N_{NMN})에 있어서 대부분의 문제에 대해 MOSEA가 우수한 결과를 보였다. 이는 NSGA-II와 SPEA2가 MOSEA보다 더 많거나 비슷한 비지배해를 구했다 할지라도, 구해진 해들의 많은 부분이 MOSEA가 구한 해들에 의해 지배당하고 있음을 의미한다. 또한 실험결과는 문제의 규모가 큰 문제에서 MOSEA의 성능이 더 우수함을 보여주고 있다. 이 결과는 MOSEA의 구조가 큰 규모의 통합문제를 부분문제로 분리하여 진화하는 구조로, 복잡도가 높고 동적인 문제에 효율적인

것으로 알려진 공생 진화알고리즘의 특성과 부합한다. 질적 비교(N_{MMN}, N_{NMN})에 있어서 작업 수가 45개, 61개를 갖는 대부분의 실험문제에서 작업 수 19개인 문제의 결과와 비교할 때 MOSEA의 비율이 더 높게 나타났다. 또한 그 유의수준이 5% 이하를 보여주고 있어, 이는 알고리즘 간에 유의한 차이가 있으며 MOSEA가 우수하다고 판단할 수 있다.

<Figure 8>은 특정 문제(prob. 8)에 대해, 세대가 진행됨에 따라 각 알고리즘이 구한 비지배해들의 분포 및 수렴성을 나타낸 것이다. 세대의 변화는 자손의 생산개수로 표시하였다. 그림에서 비지배해의 분포는 알고리즘 간 큰 차이를 보이지 않지만, 수렴속도 측면에서는 같은 세대진행에서 MOSEA가 타 알고리즘보다 향상된 모습을 보여주고 있다. 이는 동일한 기간 동안 MOSEA의 비지배해 탐색능력이 비교된 타 알고리즘보다 우수하다고 볼 수 있다.

한편 컴퓨터 계산시간은 MOSEA, NSGA-II, SPEA2의 순으로 prob.1~5가 23초, 9초, 26초, prob.6~11은 26초, 15초, 30초, 그리고 prob.12~17에서 31초, 20초, 35초가 평균 소요되었다. NSGA-II의 경우 새로운 비지배분류 방법을 사용하여 기존 알고리즘이 갖는 과도한 계산시간을 개선한 알고리즘으로 계산 시간이 가장 빠른 반면, MOSEA의 경우 부분 모집단의 운영으로 각 모집단에서 개체의 공생 파트너 선택과 적응도 평가에 필요한 계산소요량으로 인해 NSGA-II보다 상대적으로 많은 시간을 요구하는 것으로 보인다.

6. 결론

본 연구에서는 다목적성을 갖는 FAS 공정계획문제를 다루었다.

Table 3. Performance comparison of MOSEA and SPEA2

problem	N_M	N_S	N_{MS}	N_{MMS}	N_{SMS}	significance probability
Prob. 1	8.4	5.5	7.0	3.6 (51.2%)	3.4 (48.8%)	0.808
Prob. 2	7.6	5.6	7.0	3.1 (44.5%)	3.9 (55.5%)	0.228
Prob. 3	9.5	5.4	7.6	4.4 (57.7%)	3.2 (42.3%)	0.151
Prob. 4	8.3	5.6	7.2	3.7 (52.1%)	3.4 (47.9%)	0.669
Prob. 5	10.8	7.7	10.5	6.4 (61.0%)	4.1 (39.0%)	0.023*
Prob. 6	13.7	12.7	15.8	12.4 (78.5%)	3.4 (21.5%)	0.000*
Prob. 7	15.8	16.5	18.5	14.2 (76.4%)	4.4 (23.6%)	0.000*
Prob. 8	14.5	15.0	19.2	13.8 (71.8%)	5.4 (28.2%)	0.000*
Prob. 9	15.3	12.7	17.5	13.4 (76.2%)	4.2 (23.8%)	0.000*
Prob. 10	17.5	17.9	22.7	15.3 (67.2%)	7.5 (32.8%)	0.000*
Prob. 11	16.4	15.6	21.1	13.6 (64.5%)	7.5 (35.5%)	0.000*
Prob. 12	19.4	14.2	23.4	17.4 (74.4%)	6.0 (25.6%)	0.000*
Prob. 13	12.5	19.2	19.0	12.1 (63.9%)	6.9 (36.1%)	0.000*
Prob. 14	15.0	20.4	22.6	13.9 (61.4%)	8.7 (38.6%)	0.000*
Prob. 15	14.9	15.5	22.5	13.1 (58.1%)	9.4 (41.9%)	0.000*
Prob. 16	12.7	16.6	19.1	11.6 (60.6%)	7.5 (39.4%)	0.000*
Prob. 17	15.9	15.8	23.7	14.6 (61.7%)	9.1 (38.3%)	0.000*

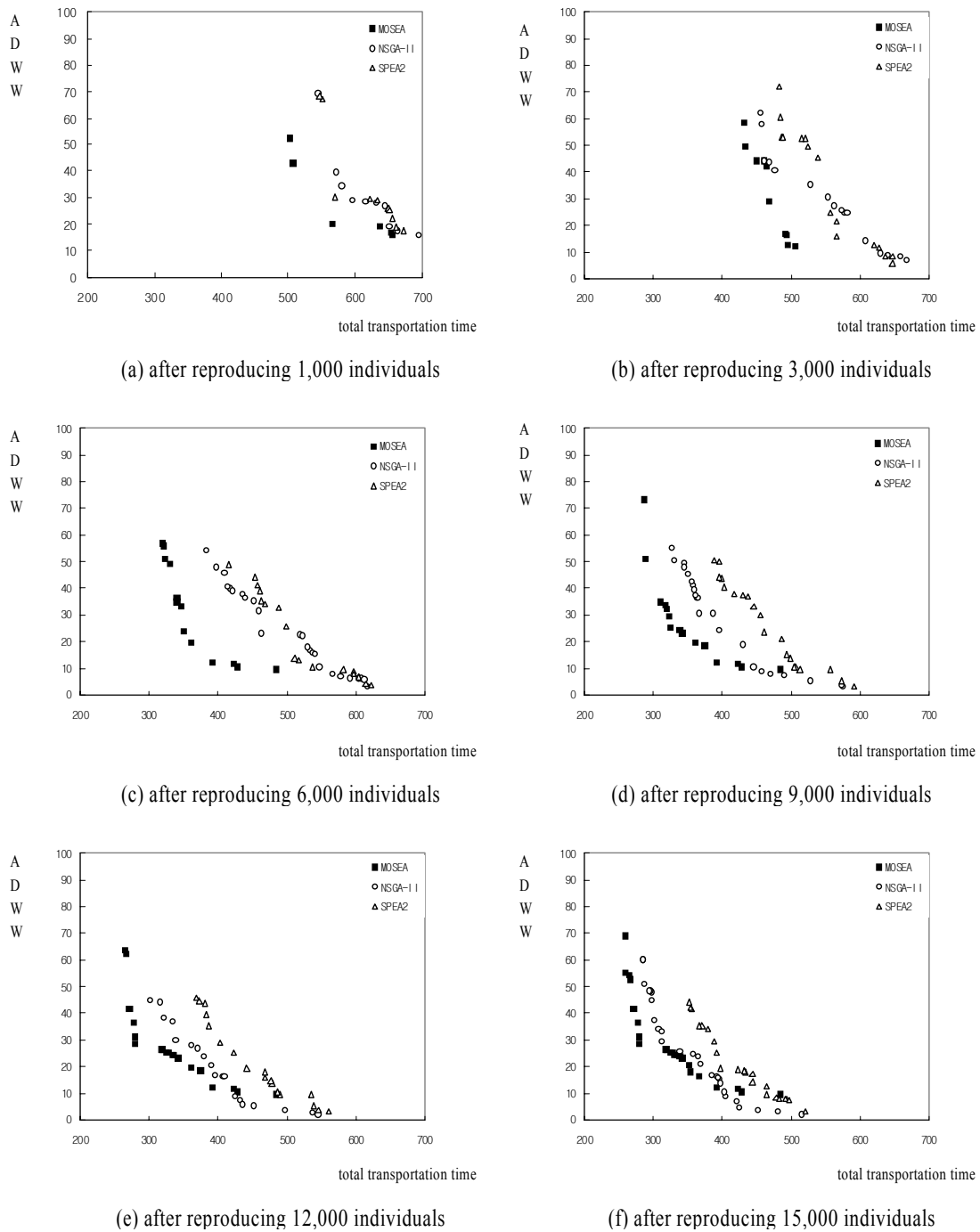


Figure 8. Distribution of non-dominated solutions and convergence speed.

FAS 공정계획문제는 두 부분문제로 이루어진 통합문제로서, 이를 순차적인 방법이 아닌 통합적으로 접근하여 해결하기 위한 방법론으로 공생 진화알고리즘을 응용한 다목적 공생 진화 알고리즘(MOSEA)을 제안하였다.

MOSEA는 기존 MOEA 연구에서 문제를 하나의 개체로 표현하여 모집단을 운영하는 방법 대신, 부분문제에 대한 개체를 표현하고 각 부분개체로 이루어진 모집단을 운영한다. 모

집단의 개체는 상대 모집단 개체와의 결합을 통해 각 목적들이 계산되고 또한 적응도가 평가된다. 각 모집단은 평가된 적응도를 기준으로 독립적으로 진화하는데, 이 때 우수한 비지배해가 진화하는 동안 사라지지 않도록 ($\mu + \lambda$)-ES의 엘리티즘을 적용한다. 이렇듯 MOSEA는 공생 진화알고리즘이 갖는 병렬탐색과 엘리티즘을 구현한 모집단의 진화가 적절하게 조화되어 다양하고 우수한 비지배해를 구할 수 있도록 한다.

복잡도가 서로 다른 여러 실험문제를 생성하여, MOSEA와 비교적 최근에 개발된 NSGA-II와 SPEA2 알고리즘의 성능을 비교 분석하였다. MOSEA가 비교된 두 알고리즘보다 우수하거나 비슷한 결과를 보였다. 특히 복잡도가 높은 문제일수록 제안한 MOSEA가 보다 우수한 성능을 보였다. 따라서 제안한 MOSEA는 다양한 구성요소로 이루어진 현실적인 시스템에서 발생하는 통합형태의 다목적 문제에서 해결을 위한 효율적 방법론으로 응용될 수 있을 것이라 기대된다.

부 록

A1. 개체 집합 Q에서 개체 i의 비지배순위 r(i)를 구하는 절차

fast nondominated sorting (Q, r(i))

단계 1: $F_1 = \emptyset$ 로 둔다.

단계 2: 집합 Q의 각 개체 p에 대해,

단계 2.1: $S_p = \emptyset, n(p) = 0$ 으로 둔다.

단계 2.2: 각 개체 q ($q \in Q, q \neq p$)에 대해, $p > q$ 이면 $S_p \leftarrow S_p \cup \{q\}$, $p < q$ 이면 $n(p) \leftarrow n(p) + 1$ 로 한다.

단계 2.3: $n(p) = 0$ 이면 $r(p) = 1$ 로 두고 $F_1 \leftarrow F_1 \cup \{p\}$ 로 한다.

단계 3: $l = 1$ 로 둔다.

단계 3.1: 집합 $H = \emptyset$ 로 둔다.

단계 3.2: 집합 F_l 에서 개체 j를 선택한다.

단계 3.3: S_j 에 있는 각 개체 k에 대해 $n(k) \leftarrow n(k) - 1$ 로 한다. 만약 $n(k) = 0$ 이면 $H \leftarrow H \cup \{k\}$, $r(k) = l + 1$ 로 한다.

단계 3.4: F_l 에서 개체 j를 제거한다. $F_l \neq \emptyset$ 이면 단계 3.2로 간다.

단계 3.5: $l = l + 1, F_l = H$ 로 둔다.

단계 3.6: $F_l = \emptyset$ 이면 종료하고, 그렇지 않으면 단계 3.1로 간다.

여기서, $p > q$ 는 개체 p가 개체 q를 지배함을 의미한다. 단계 1과 단계 2는 각 개체들의 비지배관계를 판단하고 최상위 비지배순위를 구하는 과정이다. $n(p)$ 는 개체 p를 지배하는 개체 수이고, S_p 는 개체 p가 지배하는 개체들의 집합이다. $n(p) = 0$ 인 개체 p는 자신을 지배하는 개체 수가 0이므로 비지배해이다. 따라서 단계 2에서 집합 F_1 은 Q의 개체들 중 비지배해들로 이루어진 개체집합이다. 단계 3은 위 단계에서 구해진 $n(p)$ 와 S_p 를 이용하여 차례로 비지배순위를 결정하는 과정이다. 현재의 비지배순위 j에 속하는 개체 $\{j \in F\}$ 가 지배하는 개체집합은 S_j 이다. 따라서 S_j 내 각 개체 k ($k \in S_j$)의 $n(k)$ 값을 1씩 감소시켜 $n(k) = 0$ 이 되면, 개체 k는 현재

비지배해 외에 자신을 지배하는 개체가 없으므로 현재의 순위보다 한 단계 낮은 비지배순위 ($l+1$)를 갖게 된다. 이러한 과정을 반복하여 개체의 비지배순위를 차례로 결정한다.

A2. 동일한 비지배순위를 갖는 개체집합 S에서 개체 i의 밀집거리 d(i)를 구하는 절차

crowding distance assignment(S, d(i))

$f_m(i)$: 개체 i의 m번째 목적의 함수 값

$m[j]$: m번째 목적의 함수 값을 기준으로 정렬한 후의 j번째 개체

L: 고려하는 목적의 수

단계 1: $m = 1, d(i) = 0, i = 1, 2, \dots, |S|$ 로 둔다.

단계 2: m번째 목적의 함수값을 기준으로 오름차순으로 정렬한다.

$d[m[1]] = d[m[|S|]] = \infty$ 로 둔다.

단계 3: $j = 2, 3, \dots, |S| - 1$ 에 대해 $d[m[j]] \leftarrow \{f_m(m[j+1]) - f_m(m[j-1])\}$ 을 계산한다.

단계 4: $m = L$ 이면 끝내고, 그렇지 않으면 $m \leftarrow m + 1$ 로 두고 단계 2로 간다.

개체 i의 밀집거리 $d(i)$ 는 개체 i와 동일한 비지배순위를 갖는 개체 중에서 i와 각 목적별로 거리가 가장 가까운 개체들과의 거리를 합한 값이다. 예를 들어 2개의 최소화 목적을 갖는 경우, 밀집거리는 <Figure 9>와 같다.

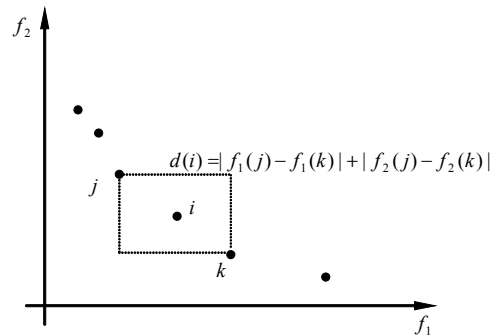


Figure 9. Crowding distance.

A3. 작업할당 초기 모집단의 개체생성 절차

단계 1: 작업들의 집합 I를 생성한다. $b_j = 0, \forall j$ 로 둔다.

단계 2: 집합 I에서 임의로 작업 i^* 를 선택한다.

단계 3: $j^* = \arg \max_{j \in AS(i^*)} \{B_j - b_j\}$ 인 작업장 j^* 에 모든 제품의 작업 i^* 를 할당한다. j^* 가 여러 개 존재하면 임의로 선택한다. 여기서 $AS(i^*)$ 는 작업 i^* 가 할당될 수 있는 작업장 집합이다.

단계 4: $b_{j_i} \leftarrow b_{j_i} + a_{i_j}$ 로 갱신하고, 집합 I 에서 i^* 를 삭제한다. $I = \emptyset$ 이면 종료하고, 그렇지 않으면 단계 2로 간다.

참고문헌

- Ammons, J.C., Lofgren, C.B. and McGinnis, L.F.(1985), A large scale machine loading problem in flexible assembly, *Annals of Operations Research*, **3**, 319-332.
- Coello, C.A.C.(1999), A Comprehensive Survey of Evolutionary Based Multiobjective Optimization Techniques, *Knowledge and Information Systems*, **1**(3), 269-308.
- Deb, K.(1999), Multi objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems, *Evolutionary Computation*, **7**(3), 205-230.
- Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T.(2000), 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, pp. 849-858. Springer.
- Fonseca, C.M. and Fleming, P.J.(1993), 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, 416-423.
- Goldberg, D.E.(1989), Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, Reading, Massachusetts.
- Groover, M.P.(2001), Automation, production systems and computer integrated manufacturing, Prentice-Hall.
- Horn, J., Nafpliotis, N. and Goldberg D.E.(1994), A niched Pareto genetic algorithm for multiobjective optimization. *IEEE international Conference on Evolutionary Computation*, **1**, 82-87.
- Hyun, C.J., Kim, Y.H. and Kim, Y.K.(1998), A genetic algorithm for multiple objective sequencing problems in mixed model assembly. *Computers & Operations Research*, **25**, 657-690.
- Khouja, M., Booth, D.E., Suh, M. and Mahaney, J.K.(2000), Statistical procedures for task assignment and robot selection in assembly cells, *International Journal of Computer Integrated Manufacturing*, **13**, 95-106.
- Kilbridge, M.D. and Wester, L.(1964), A heuristic method of assembly line balancing, *Journal of Industrial Engineering*, **12**.
- Kim, Y.K. and Kwak, J.S.(1993), Mixed model assembly line balancing with the related task consideration, *Journal of the Korean Operations Research and Management Science Society*, **18**(2), 1-22.
- Kim, Y.K., Euy, J.M., Shin, K.S. and Kim, Y.J.(2004), Process Planning in Flexible Assembly Systems Using a Symbiotic Evolutionary Algorithm, *IE Interfaces*, **17**(2), 208-217.
- 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, J.Y., Kim, Y. and Kim, Y.K.(2001), An endosymbiotic evolutionary algorithm for optimization, *Applied Intelligence*, **15**, 117-130.
- Kim, Y.K., Kim, S.J., Kim, J.Y. and Kwak, J.S.(1999), A Coevolutionary Algorithm for Balancing and Sequencing Mixed Model U-Lines, *Journal of the Korea Institute of Industrial Engineers*, **25**(4), 411-420.
- Kim, Y.K., Yun, B.S. and Lee, S.B. (1997a), Metaheuristics, Yeongji Moonhwa, Seoul, Korea.
- Kim, Y.K., Lee, S.Y., Kim, Y.J.(1997b), A Genetic Algorithm for Improving Workload Smoothness in Mixed Model Assembly Lines, *Journal of the Korea Institute of Industrial Engineers*, **23**(3), 515-532.
- Knowles, J.D. and Corne, D.W.(1999), The Pareto archived evolution strategy: A new baseline algorithm for multi-objective optimization. *IEEE International Conference on Evolutionary Computation*, 98-105.
- Lee, H.F. and Johnson, R.V.(1991), A Line Balancing strategy for designing flexible assembly systems, *The International Journal of Flexible Manufacturing Systems*, **3**, 91-120.
- Lee, H.F. and Stecke, K.E.(1996), An integrated design support method for flexible assembly system, *Journal of Manufacturing Systems*, **15**, 13-32.
- Lucertini, M., Pacciarelli, D. and Pacifici, A.(1996), Optimal flow management in flexible assembly systems: The minimal part transfer problem, *Systems Science*, **22**(2), 69-80.
- Markus, D. and Robert, J.G.(1989), Flexible assembly systems; test result for an approach for near real time scheduling and routing of multiple products, *International Journal of Production Research*, **27**, 215-227.
- Moriarty, D.E. and Miikkulainen, R.(1997), Forming neural networks through efficient and adaptive coevolution, *Evolutionary Computation*, **5**, 373-399.
- Park, M.W. and Kim, Y.D.(1995), A heuristic for setting up a flexible assembly system, *International Journal of Production Research*, **33**, 2295-2310.
- Potter, M.A.(1997), The design and analysis of a computational model of cooperative coevolution, Ph.D. dissertation, George Mason University.
- Sawik, T.(1997), An interactive approach to bicriterion loading of a flexible assembly system, *Mathematical and Computer Modeling*, **25**, 71-83.
- Sawik, T.(1999), Production planning and scheduling in flexible assembly systems, Springer-Verlag, Berlin.
- Sawik, T.(2000), An LP based approach for loading and routing in a flexible assembly line, *International Journal of Production Economics*, **64**, 49-58.
- Schaffer, J.D.(1985), Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, In Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms, 93-100, Lawrence Erlbaum.
- Shin, K. S.(2005), A multi objective process planning of flexible assembly systems with evolutionary algorithms, available at <http://syslab.chonnam.ac.kr/links/M-FASData.doc>.
- Srinivas, N. and Deb, K.(1994), Multiobjective optimization using nondominated sorting in genetic algorithms, *Evolutionary Computation*, **2**(3), 221-248.
- Tan, K.C., LEE, T.H. and Khor, E.F.(2002), Evolutionary Algorithms for Multi-Objective Optimization: Performance Assessments and Comparison, *Artificial Intelligence Review*, **17**, 253-290.
- Thomopoulos, N.T.(1967), Line balancing-sequencing for mixed

model assembly, *Management Science*, **14**, 59-75.

Zitzler, E. and Thiele L.(1999), Mutlobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, **3**(4), 257-271.

Zitzler, E., Laumanns M. and Thiele, L.(2001), 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.