

경쟁 공진화알고리즘을 이용한 다목적 Job Shop 일정계획 Multi-objective job shop scheduling using a competitive coevolutionary algorithm

이현수 · 신경석 · 김여근

전남대학교 산업공학과

Abstract

Evolutionary algorithm is recognized as a promising approach to solving multi-objective combinatorial optimization problems. When no preference information of decision makers is given, multi-objective optimization problems have been commonly used to search for diverse and good Pareto optimal solution. In this paper we propose a new multi-objective evolutionary algorithm based on competitive coevolutionary algorithm, and demonstrate the applicability of the algorithm. The proposed algorithm is designed to promote both population diversity and rapidity of convergence. To achieve this, the strategies of fitness evaluation and the operation of the Pareto set are developed. The algorithm is applied to job shop scheduling problems (JSPs). The JSPs have two objectives: minimizing makespan and minimizing earliness or tardiness. The proposed algorithm is compared with existing evolutionary algorithms in terms of solution quality and diversity. The experimental results reveal the effectiveness of our approach.

1. 서론

최근, 다목적 문제를 해결하는 데 진화 알고리즘이 흔히 채용되고 있다(Carlos, 1996; Greenwood, 1996; Zitzler, 2001; Deb et al., 2002). 이는 진화알고리즘이 모집단을 운용함으로써, 의사 결정자의 선호 정보가 없는 상태에서 여러 해를 제시할 수 있다는 데 기인 한다. 이러한 연구들은 다양하고 좋은 해를 구하는 방법론에 중점을 두고 있다. 본 연구에서는 좀 더 효과적으로 이러한 해를 구할 수 있는 새로운 기법을 제시한다. 이 기법은 경쟁 공진화 알고리즘을 기반으로 한다.

Hillis (1990)에 의해 제안된 경쟁 공진화 알고리즘은 생태계의 경쟁(competition) 또는 기생(parasitism) 관계를 모방한 탐색적 기법이다. 숙주(host)가 자신에게 기생하는 기생충(parasite)을 이기기 위하여 진화하고, 기생충 또한 이에 적응하기 위하여 진화한다. 이와 같은 과정을 반복하면서 경쟁적으로 진화하는 것을 진화적 무기경쟁(evolutionary arms race)이라 부른다(Nolfi & Floreano, 1998). 경쟁 공진화 알고리즘의 특성과 그 응용에 관한 여러 연구가 행하여지고 있다(Nolfi & Floreano, 1998; Kim et al., 2000; Kim et al., 2002; Buason, 2002).

본 연구에서는 경쟁 공진화 알고리즘의 특성을 이용해 다목적 조합 최적화 문제를 해결하기에 적절한 다목적 경쟁 공진화 알고리즘을 개발한다. 또한, 개발한 알고리즘을 job shop 일정계획문제에

적용한다.

Job shop 일정계획 문제(JSP)는 부품생산을 위한 공정간 순서계약과 기계계약아래서 각 기계에 효과적으로 각 부품의 공정을 할당하는 문제이다.

JSP에서 흔히 사용하는 목적으로는 총처리시간(makespan), 평균처리시간(mean flowtime), 또는 납기지연(tardiness) 등의 최소화이다(Raman & Talbot, 1993; Carlos, 1996; Lee et al., 2002). 총처리시간은 모든 부품의 처리를 완료하는 시간을 의미하고, 평균처리시간은 공정의 혼잡도와 재공재고의 수준을 나타내는 척도로 사용된다(Carlos 1996). 또한, 납기지연은 납기에 대한 고객 만족도, 납기지연 벌금과 재공재고와 관련되어 있다. 그러나 이들 목적들은 서로 다른 목적들에 영향을 미치거나, 각 목적을 달성하는데 있어 상충관계를 가질 수도 있다.

본 연구에서는 경쟁공진화 알고리즘에 기반을 둔 다목적 문제를 위한 알고리즘을 개발하고, 이를 JSP에 적용한다. 그리고 해의 질과 다양성, 그리고 계산소요시간의 측면에서 기존의 알고리즘들과 비교한다.

2. Job shop 일정계획

job shop 일정계획은 제품 또는 부품을 생산하기 위해 필요한 자원을 배분하고 기계를 할당하여 작업을 수행하기 위한 일정표를 작성하는 것이다. 본 연구에서 다루는 JSP의 가정은 아래와 같이 둔다.

- 가공할 부품은 주어져 있고, 시작시점에 모든 부품의 가공이 가능하다.
- 한 기계는 한번에 하나의 부품만을 가공할 수 있다.
- 각 공정은 하나의 특정 기계에서 가공된다.
- 공정 선택은 허용되지 않는다.
- 가공을 위한 기계간의 이동시간은 무시한다.
- 공정에 대한 준비시간은 가공 시간에 포함된다.

본 연구에서는 이러한 가정 하에서, 총처리시간 최소화과 납기 지연 또는 이른 납기의 최소화라는 두 목적을 갖는 일정계획을 찾고자 한다.

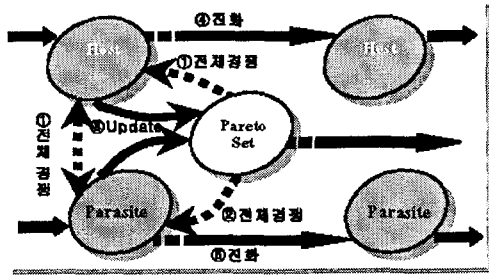
3. 다목적 JSP를 위한 경쟁 공진화 알고리즘

3.1 알고리즘의 개념

본 연구에서 제안하는 다목적 경쟁 공진화 알고리즘(Multi-Objective Competitive Coevolutionary Algorithm: MOCCA)의 흐름도는 [그림 1]과 같다.

경쟁 공진화의 개념 도입은 숙주 모집단과 기생충

모집단의 지속적인 무기경쟁을 통하여 빠른 시간에 다양한 파레토 해를 찾는데 있다.



[그림 1] MOCCA의 개념적 진화과정

숙주 모집단(host population: Pop_H)은 job shop 일정계획을 위한 모집단으로 각 개체는 일정계획을 표현한다. 개체표현에 대한 설명은 4.1절에서 다룬다. 기생충 모집단(parasite population: Pop_P)을 구성하는 개체는 숙주 모집단의 그것과 표현이 같다. 적응도는 상대 모집단과의 경쟁에 의해 결정된다. 재생산은 적응도를 기반으로 하여 이루어진다.

본 연구에서는 효율적인 알고리즘 개발을 위하여 첫째, 다목적에 맞는 경쟁 공간화 알고리즘을 설계하고, 둘째, 배분적응도(Rosin & Belew, 1997)에 기초한 적응도 부여방법을 제안한다. 그리고 마지막으로, 진화과정 동안에 발견된 좋은 해를 보관하고, 모집단의 개체들을 좀 더 좋은 해들로 유도하기 위해서 파레토 집합(Pareto set: PS)을 운영한다.

3.2 알고리즘 절차

MOCCA의 절차는 아래와 같다.

단계1: 초기화

숙주 개체와 기생충 개체로 이루어진 두 모집단 Pop_H 와 Pop_P 를 구성한다. $PS = \emptyset$.

단계2: 적응도 평가

2.1: Pop_H 의 개체들은 Pop_P 와 PS 의 모든 개체들과의 경쟁을 통하여 적응도 평가를 받는다.

2.2: Pop_P 의 개체들은 Pop_H 와 PS 의 모든 개체들과의 경쟁을 통하여 적응도 평가를 받는다. 단, Pop_H 와 경쟁은 단계 2.1의 결과를 사용한다.

단계3: 파레토 집합의 갱신

단계3.1: Pop_H 와 Pop_P 에서 비지배된 개체를 각각 복사하여 PS 에 추가한다.

단계3.2: PS 에서 새롭게 추가된 개체에 의하여 지배되는 개체들을 제거한다.

단계4: 진화

단계4.1: Pop_H 에서 적응도를 기준으로 재생산에 참여할 개체를 선별한다.

단계4.2: 교차와 돌연변이를 통해 자손을 생성한다.

단계4.3: Pop_P 에서 적응도를 기준으로 재생산에 참여할 개체를 선별한다.

단계4.4: 교차와 돌연변이를 통해 자손을

생성한다.

단계5: 종료 판단

종료조건을 만족하면 알고리즘을 종료하고, 그렇지 않으면 단계 2로 간다.

단계 2에서 적응도 평가방법은 4.2절에서 설명한다. 단계 3에서 파레토 집합은 모든 비지배해들로서 구성된다. Pop_H 와 Pop_P 모집단에서 새롭게 추가된 개체는 어떤 개체에게도 지배되지 않는 개체이다. 따라서 새롭게 추가된 개체에 의하여 지배된 파레토 집합내의 개체들만을 제거 시켜주면 된다.

4. 진화요소

4.1 개체표현

진화 알고리즘으로 특정문제를 해결할 때, 어떻게 개체를 표현 할 것인가 하는 것은 중요한 문제이다. 개체를 어떻게 표현하느냐에 따라 해의 질이나 진화속도가 달라질 수 있다.

JSP를 위한 개체표현에는 작업기반 표현(job based representation), 공정기반표현(operation based representation), 그리고 우선순위 기반표현(priority based representation)등이 사용된다(Gen & Cheng, 1997). 본 연구에서는 우선순위 기반표현을 사용한다.

일정계획 생성자(active scheduler, nondelay scheduler 등)를 사용함에 있어 같은 자원(기계)을 요하는 동일한 공정이 두 개 이상일 수 있다. 이때 어느 공정을 선택할 것인가를 결정하는 데 우선순위를 사용할 수 있다. 우선순위 기반표현에서 개체의 길이는 모든 공정의 수와 같고 인자의 값은 우선순위 규칙을 의미하며, 인자의 위치는 해당규칙의 적용시점을 나타낸다. 우선순위 기반표현의 장점은 가능 해를 유지하기 위한 특별한 유전연산이 필요하지 않다는 것이다.

Active 일정계획을 생성할 때, 우선 순위에 의한 방법은 아래와 같다.

기호정의

- o_{ij} : 부품 i 의 j 번째 공정
- σ_{ij} : 공정 o_{ij} 의 가장 이른 시작시간
- p_{ij} : 공정 o_{ij} 의 가공시간
- τ_{ij} : 공정 o_{ij} 의 가장 이른 완료시간, i.e. $\tau_{ij} = \sigma_{ij} + p_{ij}$

Active 일정계획의 생성절차

단계 1: 모든 시작 공정의 집합 A 를 만든다.

$$A = \{o_{ij} | 1 \leq i \leq N\}.$$

단계2: $\tau^* = \min\{\tau_{ij} | o_{ij} \in A\}$ 인 기계 m^* 를 결정한다.

이때, τ^* 가 2개 이상이면 임의로 하나를 선택한다

단계3: 집합 A 의 공정중 기계 m^* 에서 가공될 수 있는 공정들로 집합 C 를 만든다.
 $C = \{o_{ij} \in A | \sigma_{ij} < \tau^*\}.$

단계4: 집합 C 에 우선순위규칙을 적용하여, 공정 o_{ij} 를 선택하고, 공정 o_{ij} 를 일정계획에 포함시킨다. 공정 o_{ij} 를 A 에서 삭제한다.
 $A = A \setminus \{o_{ij}\}.$ 이때, o_{ij} 가 두 개 이상이면 임의로 하나를 선택한다.

단계5: 공정 o_{ij}^* 의 후행공정 $o_{i,j+1}^*$ 이 있으면, 이를 집합 A 에 추가한다. $A := A \cup \{o_{i,j+1}^*\}$.

단계6: 집합 A 가 공집합이 아니면 단계 2로 간다. 모든 공정의 일정계획이 완료될 때까지 이 과정을 반복한다.

본 연구에서 사용된 우선순위 할당규칙들은 <표 1>과 같다. 그리고 여기에서 간략히 active 일정계획에 대한 표현의 예를 들 수 있다.

기호정의

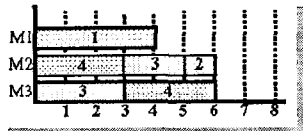
- d_i : 부품 i 의 납기일.
- r_i : 부품 i 의 남아있는 가공시간.
- p_{ij} : 부품 i 의 j 번째 공정의 가공시간.
- t : 현 시점.

<표 1> 우선순위 할당규칙

Priority Rule		
① EDD	Earliest Due Date	Min d_i
② (MIN)SLACK	Slack	Min($d_i - r_i - t$)
③ SPT	Shortest Processing Time	Min p_{ij}
④ CR	Critical Ratio	Min($d_i - t$) / r_i
⑤ MWRK	Most Work Remaining	Max r_i
⑥ LWRK	Least Work Remaining	Min r_i
⑦ RND	Random	

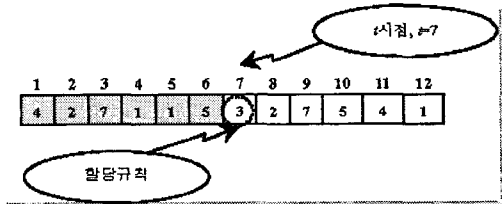
<표 2> JSP의 예제(왼쪽-공정시간, 오른쪽-기계)

	Operation				Operation		
	1	2	3		1	2	3
Job1	4	3	2	Job1	1	2	3
Job2	1	4	4	Job2	2	1	3
Job3	3	2	3	Job3	3	2	1
Job4	3	3	1	Job4	2	3	1



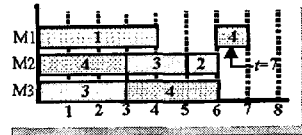
[그림 2] Active 일정계획상에서 부분 일정계획

<표 2>와 [그림 2]는 우선순위 기반표현에 대한 이해를 돕기 위한 각각의 표와 그림이다. <표 2>는 부품에 대한 가공시간과 작업 가능한 기계를 나타낸다. [그림 2]는 <표 2>의 전체 공정중 6개의 공정에 대한 부분일정계획이 수립되었다고 가정한다.



[그림 3] 우선순위 기반 표현

이때, $t=7$ 에 해당하는 다음 일정계획은 Active 일정계획의 적용절차에 따라, 집합 $A = \{(1, 2), (2, 2), (3, 3), (4, 3)\}$, $\tau^* = \tau_{43}$, 그리고 $m^* = 1$ 을 각각 구할 수 있다. $m^* = 1$ 일 때, 집합 $C = \{(2, 2), (3, 3), (4, 3)\}$ 이 된다. 또한, [그림 3]과 같은 인자로 구성된 개체라면, 집합 C 에는 SPT 규칙이 적용된다. 따라서, $o_{ij}^* = o_{43}$ 이 되고, [그림 4]에서 보이는 것처럼 기계 1에 부품 4와 3번째 공정이 일정계획에 추가된다. 일정계획이 완료된 공정 o_{43} 은 집합 A 에서 삭제하고, o_{43} 의 직후행 공정이 집합 A 에 추가한다. 모든 공정의 일정계획이 완료될 때까지 이 과정을 반복한다.



[그림 4] Active 일정계획상에서의 부분 일정계획

4.2 적용도 함수

본 연구에서 사용한 다목적함수들은 아래와 같다.

$$f_1(\sigma) = \max_k \max_i c_{ik} \quad (1)$$

$$f_2(\sigma) = \frac{1}{N} \sum_{i=1}^N |C_i - d_i| \quad (2)$$

여기서, σ 는 가능해인 active 일정계획이고, c_{ik} 는 기계 k 에서 작업 i 의 완성시간, C_i 는 작업 i 의 마지막 공정이 완료되는 시간, 그리고 d_i 는 작업 i 의 납기일을 나타낸다. 목적함수식 (1)은 총처리시간의 최소화, 식 (2)는 납기일로부터 이른납기와 납기 지연의 최소화를 위한 식들이다.

4.3 경쟁전략

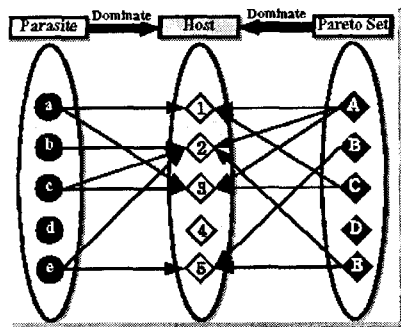
다목적 진화알고리즘에서 개체의 적용도는 부여 방법에 따라 다양한 해와 좋은 해의 탐색에 큰 영향을 준다. 단일목적 최대화문제에서는 하나의 목적함수에 의해 적용도를 쉽게 구할 수 있어, 이 적용도를 근거하여 선별이 이루어진다. 그러나 다목적 문제에서는 여러 목적함수가 존재하므로 각 개체의 적용도를 어떤 기준에 의해 측정하는가가 문제이다. 앞에서 언급했듯이, 한 목적을 최대로 만족하는 해와 다른 목적을 최대로 만족하는 해가 서로 다를 수 있기 때문에 개체의 적용도를 하나의 척도로 제량화하기가 어렵다. 따라서 다목적 진화알고리즘에서의 연구 초점은 주로 선별방법에 있다.

본 연구에서는 Rosin & Belew (1997)의 연구에서 사용된 배분 적용도를 다목적 문제에 맞도록 적절히 변형하여 사용한다. Rosin & Belew가 제안한 배분 적용도는 단순 적용도 평가 방법에 의해 고유 적용도를 구하지 않고 경쟁상대의 난이도(difficulty)를 기준으로 하여 즉, 경쟁하여 승리하기 어려운 상대를 때배시킨 개체에 대해 상대적으로 높은 적용도를 부여하는 방법이다. 다목적 문제를 해결하기 위해 변형된 배분 적용도는 경쟁상대와

경쟁에서 상대 개체에게 지배된 정보를 이용한다. 이때, 숙주 개체 i 의 적응도, f_i 는 식 (3)과 같이 계산된다.

$$f_i = \frac{1}{1 + \left(\sum_{j \in X} \frac{1}{N_j} + \sum_{k \in Y} \frac{1}{N_k} \right)} \quad (3)$$

여기서, X 는 숙주개체 i 를 지배한 기생충 개체 j 의 집합을 나타내고, N_j 는 기생충 개체 j 가 지배한 숙주 개체의 수를 나타낸다. 또한, Y 는 숙주 개체 i 를 지배한 파레토 집합의 개체 k 를 의미하고, N_k 는 파레토 집합의 개체 k 가 지배한 숙주개체의 수를 나타낸다. 한편, 기생충 개체의 적응도는 숙주와 기생충의 역할을 교환하여 계산한다.



[그림 5] 경쟁상대에게 지배된 정보

[그림 5]는 숙주가 기생충과 파레토집합 내의 개체와의 경쟁의 한 예를 보인다. 여기서, 화살표는 경쟁에서 상대 개체에게 패한 것을 의미한다. 예로, 숙주 1은 경쟁에서 기생충 a에게, 파레토 집합의 개체 A와 C에게 패하였음을 나타낸다. 또한, 기생충 a는 경쟁 상대인 숙주들과의 경쟁에서 2개의 숙주를 지배하고, 파레토 집합의 개체 A와 C는 각각 숙주 3개와 2개를 지배하였다. 이때, 숙주 개체 1의 적응도는 식 (4)와 같이 계산된다.

$$f_1 = \frac{1}{1 + \left\{ (1/2) + (1/3 + 1/2) \right\}} = \frac{3}{7} \quad (4)$$

본 연구에서 배분 적응도의 개념은 강한 개체에게 패한 개체에게는 상대적으로 높은 적응도를 부여하고, 상대적으로 약한 개체에게 패한 개체에게는 비교적 낮은 적응도를 부여하기 위함이다.

4.4 파레토 집합의 운영

파레토 집합은 모집단의 진화과정에서 발견된 비지배해들을 보관하는 일종의 저장소이다. 파레토 집합을 운영하는 이유는 크게 두 가지로 나눌 수 있다.

첫째는 강한 숙주 또는 기생충을 보존하여, 이들이 경쟁에 참여하게 함으로써 공진화 과정에서

순환이 발생하는 것을 억제시킬 수 있다. 둘째는 파레토 집합내의 개체들의 경쟁 참여는 다양한 해의 유도를 촉진한다. 파레토집합의 개체들과의 경쟁결과, 파레토집합내의 개체들이 밀집되어 있는 곳보다 그렇지 않은 곳에 위치한 숙주 또는 기생충 개체들이 더 높은 적응도를 부여 받을 가능성이 높다. 따라서, 이는 탐색이 활발히 이루어 지지 않은 부분에 대한 탐색을 유도할 수 있다.

5. 실험설계와 결과분석

5.1 실험설계 및 실험파라미터

제안된 MOCCA는 성능이 좋은 것으로 알려진 두 알고리즘, NSGA II (Deb et al., 1996)와 SPEA II (Zitzler et al., 2001)가 비교 평가된다. 이를 위하여, 실험 문제로는 JSP에서 흔히 벤치마크 (<http://mscmga.ms.ic.ac.uk/info.html>)로 사용되는 6(부품수)×6(공정수, 기계수), 10×5, 10×10, 20×10, 20×20, 그리고 50×10의 문제를 사용한다. Raman et al. (1989)은 납기의 촉박성에 따라 높은 경우, 보통인 경우, 그리고 낮은 경우로 분류하였다. 본 연구에서 납기는 Raman et al. (1989) 촉박성이 보통인 경우를 이용하여 구하였다. 또한, 각 알고리즘들의 성능비교는 해의 질, 해의 다양성, 그리고 계산 시간 측면에서 비교되었다.

<표 3> 공통 유전요소 및 실험 파라미터

문제크기	모집단 크기	종료조건	교차율	돌연변이율
6×6	100	8,000	0.8	0.01
10×5	100	16,000	0.8	0.01
10×10	100	16,000	0.8	0.01
20×10	100	24,000	0.8	0.01
20×20	200	80,000	0.8	0.01
50×10	200	112,000	0.8	0.01

알고리즘들의 비교 분석을 위한 관련 파라미터는 예비 실험을 통해 결정하였다. 사용된 파라미터는 <표 3>과 같으며, 모든 실험 문제에서 이를 사용하였다. 여기서 종료조건은 재생산 개체수이다. 모든 알고리즘은 JAVA로 구현되었으며, 1.7GHz Pentium CPU를 장착한 IBM-PC에서 수행되었다.

5.2 실험결과분석

(1) 비지배해의 양과 질

먼저, 비교 알고리즘에서 얼마나 많은 비지배해들이 탐색되는지, 그리고 이들 비지배해들의 질이 얼마나 좋은 지를 실험하였다. <표 4> (a)는 NSGA II, SPEA II, 그리고 MOCCA를 사용하여 각각 10회 반복 실험한 후, 탐색한 비지배해들의 평균을 나타낸다. <표 4> (b)는 각 알고리즘이 탐색한 모든 비지배해들을 하나의 집합으로 두고, 이 집합에서 다시 비지배해를 구하여, 이를 해가 어느 알고리즘으로부터 구한 것인가를 판별하여, 그 수를 나타낸 것이다. <표 4> (a)는 비지배해의 양의 관점에서, (b)는 비지배해의 질의 관점에서 본 것이다. <표 4> (a)로부터 50×10문제를 제외한 모든

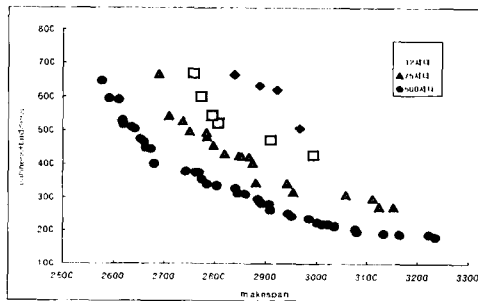
문제에서 MOCCA가 다른 알고리즘들에 비해 많은 비지배해를 찾음을 알 수 있다. 또한, <표 4> (b)로부터, MOCCA가 좋은 해를 가장 많이 찾음을 알 수 있다.

<표 4> 비지배해의 양과 질

문제 크기	(a)			(b)		
	NSGA II	SPEA II	MOCCA	NSGA II	SPEA II	MOCCA
6×6	23.2	21.2	27.1	7.6	11.4	17.6
10×5	26.6	24.8	33.6	1.5	15.8	18.0
10×10	19.0	24.2	37.8	0.0	16.2	28.0
20×10	23.6	32.0	37.4	0.0	20.4	31.0
20×20	20.4	25.4	35.2	0.0	18.3	28.4
50×10	35.0	47.6	46.6	0.0	31.4	35.2

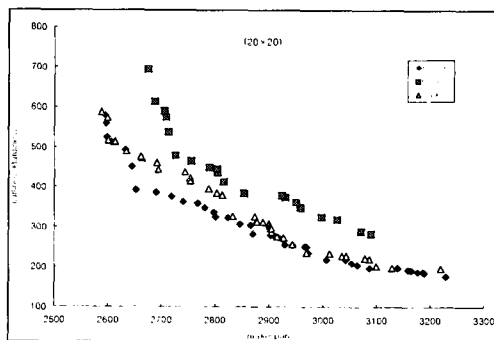
(2) 해의 다양성

[그림 6]은 MOCCA가 세대가 지남에 따라 다양한 비지배해를 탐색함을 보여 주고 있다. 이는 적소(niche)의 개념 대신, 경쟁 공진화 개념을 사용해서도 넓은 영역에서 다양한 비지배해들을 탐색할 수 있음을 알 수 있다. 즉, 경쟁 공진화 알고리즘이 경쟁을 통하여 모집단간의 약점을 상호 보완해 주는 것으로 이해된다.



[그림 6] MOCCA에서 비지배해의 탐색과정

SPEA II는 해의 다양성 측면에서 NSGA II보다 아주 좋은 것으로 알려져 있다(Zitzler, 1999). [그림 7]은 SPEA II, NSGA II와 MOCCA로부터 각기 얻은 비지배해들을 보여 주고 있다. 이로 부터, MOCCA는 SPEA II와 함께 다양한 비지배해를 생산함을 알 수 있다.



[그림 7] 해의 다양성 비교

(3) 계산소요시간

<표 5>은 각 알고리즘이 각 문제크기에 따라 종료조건까지 소비한 CPU 계산시간을 나타낸다.

<표 5> 계산소요시간

문제크기	NSGA II	SPEA II	MOCCA
6×6	6.08	20.23	5.87*
10×5	15.99	36.88	14.06*
10×10	29.78	47.27	20.50*
20×10	176.78	164.94	129.03*
20×20	639.55	842.09	421.24*
50×10	11079.67	6434.78	5845.30*

이로부터 MOCCA는 모든 실험문제에서 가장 적은 계산시간이 소요되었다. 이는 NSGA II나 SPEA II에서 엘리티즘과 다양성을 강조하기 위해 여러 기법들을 사용하는 반면, MOCCA는 단지 모집단간의 무기경쟁을 유지하기 위한 경쟁 전략만을 사용하고 있기 때문으로 보인다.

6. 결론

본 연구에서는 경쟁공진화 알고리즘을 이용하여 다목적JSP를 다루었다. 효율적인 다목적 진화알고리즘을 개발하기 위하여, 경쟁 공진화의 개념을 도입하여, 해의 수렴 속도와 모집단의 다양성을 촉진하고자 하였다. 그리고 해의 순환 방지와 모집단 다양성의 향상을 위하여 파레토 집합을 운영하였다. 또한 개체가 적절히 평가되어 좋은 해의 생성을 유도하는 적응도 평가방법을 제시하였다.

JSP에서 목적은 총처리시간 최소화과 이른납기 또는 납기지연의 최소화로 두었다. 제안한 알고리즘은 비지배해의 양과 질, 비지배해의 다양성, 그리고 계산시간의 관점에서, 기존의 두 알고리즘과 비교 분석되었다. 실험결과, 제안한 알고리즘은 해의 질과 다양성, 그리고 계산소요시간 면에서 비교알고리즘들에 뒤지지 않았다.

본 연구에서 다목적 진화알고리즘에 경쟁 공진화의 개념을 처음으로 도입하였다. 추후, 해의 수렴속도와 비지배해의 다양성을 높이고, 계산의 복잡도를 낮추는 연구가 요구된다.

감사의 글

본 연구는 한국과학재단 목적기초연구(R05-2002-000-00474-0)지원으로 수행되었음.

참고문헌

Baker, K. R. (1974), Introduction to sequencing and scheduling, John Wiley & Sons Inc., New York.
Baker, K. R. (1984), Sequencing Rules and Due Date Assignment in a job shop, *Management Science*, 30, 1093-1104.
Buason, G. (2002), Competitive co-evolution of sensory-motor system, *Masters*, University of Skovdo, Sweden.
Carlos, A.C. (1996), An Empirical study of evolutionary techniques for multiobjective optimization in

- engineering design, Ph.D. thesis, Tulane University.
- Chang, Y. L., Sueyoshi, T. and Sullivan, R. S. (1996), Ranking dispatching rules by data envelopment analysis in a job shop environment, *IIE Transactions*, **28**, 631-642.
- Deb, K., Agrawal, S. and Meyarivan, T. (2002), A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, *KanGAL Report No.200001*.
- Esquivel, S., Ferrero, S., Gallard, R., Salto, C., Alfonso, H. and Schutz, M. (2002), Enhanced evolutionary algorithms for single and multiobjective optimization in the job shop scheduling problem, *Knowledge-Based Systems*, **15**, 13-25.
- Gen, M. and Cheng, R. (1997), *Genetic Algorithms and Engineering Design*, John Wiley & Sons, Inc., New York
- Greenwood, G. W. (1996), Fitness function for multiple objective optimization problem: Combining preference with pareto ranking, *Foundations of Genetic Algorithms 4*, 455.
- Hart, E. and Ross, P. (PPSN V), A heuristic combination method for solving job-shop scheduling problems, *Parallel Problem Solving from Nature*, **5**.
- Hillis, W. D. (1990), Co-evolving parasites improve simulated evolution as an optimization procedure, *Physica D*, **42**, 228-234.
- Kim, J. Y., Kim, Y. K. and Kim, Y. H. (2002), Tournament competition and its merits for coevolutionary algorithms, *in press Journal of Heuristics*.
- Kim, S. J., Kim, Y. K., Kim, J. Y. and Kwak, J. S. (2000), A Competitive Coevolutionary Algorithm with Tournament Competitions, *Journal of the Korean Institute of Industrial Engineers*, **26**, 101-109.
- Lee, H. T., Chen, S. H. and Kang, H. Y. (2002), Multicriteria scheduling using fuzzy theory and tabu search, *International Journal of Production Research*, **40**, 1221-1234.
- Peter, J. M., Emile, H. L. and Jan, K. L. (1992), Job shop scheduling by simulated annealing, *Operations Research Society of America*, **40**, 113-125.
- Raman, N., Talbot, F. B. and Rachamadugu, R. V. (1989), Due date based scheduling in a general flexible manufacturing system, *Journal of Operations Management*, **8**.
- Raman, N. and Talbot, F. B. (1993), The job shop tardiness problem: A decomposition approach, *European Journal of Operations Research*, **69**, 187-199.
- Rosin, C. D. and Belew, R. K. (1997), New methods for competitive coevolution, *Evolutionary Computation*, **5**, 1-29.
- Srinivas, N. and Deb, K. (1995), Multi-objective Optimization Using Nondominated Sorting in Genetic Algorithms, *Evolutionary Computation*, **2**, 221-248.
- Zitzler, E. (1999), *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, Ph.D. dissertation, Swiss Federal Institute of Technology Zurich.
- Zitzler, E., Deb, K. and Thiele, L. (2000), Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary Computation*, **8**, 173-195.
- Zitzler, E., Laumanns, M. and Thiele, L. (2001), SPEA2: Improving the Strength Pareto Evolutionary Algorithm, *TIK-Report*, **103**, May.