

▣ 연구논문

Job Shop 일정계획을 위한 병렬 유전 알고리즘

- A Parallel Genetic Algorithms for Job Shop Scheduling Problems -

박병주*

Park, Byung-Joo

김현수**

Kim, Hyun-Soo

Abstract

The Job Shop Scheduling Problem(JSSP) is one of the most general and difficult of all traditional scheduling problems. The goal of this research is to develop an efficient scheduling method based on single genetic algorithm(SGA) and parallel genetic algorithm (PGA) to address JSSP. In this scheduling method, new genetic operator, generating method of initial population are developed and island model PGA are proposed. The scheduling method based on PGA are tested on standard benchmark JSSP. The results were compared with SGA and another GA-based scheduling method. The PGA search the better solution or improves average of solution in benchmark JSSP. Compared to traditional GA, the proposed approach yields significant improvement at a solution.

1. 서론

생산관리 기능에서 생산계획은 기대 수요를 만족시키기 위해 사용될 자원들을 가장 잘 활용할 수 있도록 결정한다. 생산 계획은 계획 기간에 따라 장기, 중기, 단기 계획으로 나뉜다. 단기 계획에 속하는 일정계획(scheduling)은 생산계획이 생산활동으로 전환되기 전의 마지막 단계로 장기계획과 중기계획이 자원의 획득에 관해 다루는데 반해, 운영 목적을 충족시키는 최선의 순서결정과 특정 작업들에 필요한 자원들의 할당과 관련이 있다. 일정계획의 중요성은 제품 비용과 서비스 수준의 고려에서부터 비롯된다. 잘못된 일정계획은 자원의 비효율적인 사용으로 제품의 비용을 높이고 시장에서의 경쟁력을 낮추게 되고, 비효율적인 일정계획은 종종 납기를 지연시키게 되어 고객을 만족시키지 못하는 결과를 초래하여 신용상의 문제를 유발한다. 그러므로 제한된 자원 하에서 이루어지는 작업들의 일정계획은 생산관리의 효율성 측면에서 매우 중요하다.

생산현장마다 다른 형태를 갖는 일정계획 문제들을 해결하기 위한 오랜 연구에도 불구하고 간단한 일정계획 문제에서만 최적해를 구해주는 알고리즘이 존재할 뿐 현장에서 이루어지는 실제적인 규모의 일정계획 문제를 위한 최적화 알고리즘은 제시되지 않고 있다. JSSP는 잘 알려져 있는 것처럼 NP-hard 문제일 뿐만 아니라 가장 어려운 조합 최적화 문제중 하나이다[6]. 복잡성과 넓은 탐색영역 때문에 수리 계획법, 동적 계획법, 분지한계법 등과 같은 전통적인 최

* 동아대학교 에이전트 기반 전자상거래팀(BK21) Post-Doc.

** 동아대학교 경영정보과학부 교수

적화 방법들은 계산적으로 실행 불가능할 경우가 많다. 그래서 적절한 계산시간 내에 최적해에 가까운 해를 찾아낼 수 있는 휴리스틱 탐색법에 관심이 집중되고 있다. 일반적으로 JSSP는 한정된 기계들의 사용으로 상충되는 목표를 동시에 만족시키려는 문제이고, 각 job은 선후제약에 따른 작업순서를 가진다. job과 기계들에 대한 주요 제약은 하나의 기계는 같은 시점에 단 하나의 job만을 처리할 수 있고, 어떤 job이 기계에서 처리되면 중간에 정지시킬 수 없다는 것이다. 또한 기계에 도착되는 job의 가공시간은 미리 알려져 있다고 가정한다.

본 연구에서는 JSSP를 보다 효과적으로 해결하기 위해 유전 알고리즘에 기반한 일정계획 기법을 제시한다. 이 기법에서는 새로운 유전 연산자, 초기 모집단 구성방법을 사용한다. 또한 이 기법을 확장하여 병렬 유전 알고리즘을 기반으로 한 일정계획 기법을 제시한다.

2. 유전 알고리즘

유전 알고리즘은 자연진화의 법칙인 적자생존과 자연도태의 원리를 바탕으로, 자연계의 현상인 교차(crossover)와 돌연변이(mutation) 과정에서 환경에 적합한 개체는 생존확률이 높고, 부적합한 개체는 도태되어 간다는 진화론을 일반 최적화 문제에 적용한 것이다. Holland[10], Goldberg[9], Bäck[2]는 유전 알고리즘의 이론적인 분석에 대한 내용을 담고있다. 유전 알고리즘을 사용하기 위해서는 문제에 대한 특성을 먼저 분석한 다음 그 문제에 적합한 표현방법, 평가함수, 초기 모집단 구성방법, 유전연산자, 유전 파라미터 등이 결정되어야 한다. 유전 알고리즘의 과정은 우선 초기 모집단을 구성하고, 그 집단에서 선택을 통해 다음 세대로 유전 형질을 물려줄 개체들을 선택하고, 이들 개체들에 유전 연산자를 적용시켜 다음 세대의 집단을 형성한다. 선택에서는 적합도 함수를 통해 우수한 형질을 가지는 개체를 선택한다. 부여된 종료조건이 만족 될 때까지 이러한 과정을 반복한다.

2.1 표현

Job Shop 일정계획 문제를 해결하기 위해서는 먼저 문제의 해를 염색체로 표현해야 한다. 표현은 공정 중심(operation-based) 표현 방법을 사용한다. 이는 job 번호를 공정의 수만큼 반복시키는 순열형태로 표현한다. 하나의 유전인자는 하나의 공정을 의미하고 표현형태는 처리되는 공정의 job 번호로 표현한다. 이처럼 job 번호의 반복을 통해 표현한 염색체는 job의 공정들이 스케줄 되어지는 순서를 나타낸다. 예를 들어 <표 1>의 3×3 문제가 순열 형태의 염색체 [3 2 2 1 1 2 3 1 3]으로 표현되었다면, 세 번씩 반복된 숫자는 job의 번호를 나타낸다. job 번호가 세 번씩 반복되는 것은 각 job들이 3개의 공정을 가지고 있기 때문인데, job 번호의 첫 번째 반복은 그 job의 첫 공정을 두 번째 반복은 두 번째 공정을 의미한다. 염색체의 3번째 유전인자가 2인데 이 유전인자는 job 2의 두 번째 공정에 대응된다. 왜냐하면 2의 표현이 염색체에서 두 번 반복되었기에 이는 job 2의 두 번째 공정을 의미한다.

<표 1> 3×3||Cmax 문제

Job	기계순서 (가공시간)		
1	3(1)	1(3)	2(6)
2	2(8)	3(5)	1(4)
3	3(5)	2(4)	1(8)

염색체 표현에서 job의 공정들과 처리되는 기계들의 관계는 다음과 같다.

염색체	[3 2 2 1 1 2 3 1 3]
기계	[3 2 3 3 1 1 2 2 1]

염색체의 생성은 Giffler & Thompson (G&T) 알고리즘[8]을 이용한다. G&T 알고리즘은 단계 4의 G 집합 내에 속해 있는 공정들을 하나씩 선택하는 과정을 전체 공정의 수만큼 수행하여 스케줄하게 되는데, 이때 하나씩 선택되어지는 공정을 포함하는 job의 번호를 선택 순서대로 전체 공정 수만큼 연결하여 기입함으로써 염색체가 생성된다.

2.2 초기 모집단

국소 탐색법에서 초기 값은 해에 많은 영향을 미친다. 기존의 JSSP에 적용된 유전 알고리즘은 대부분 초기 모집단을 임의대로 발생시켜 사용해 왔다. 이로 인해 해를 찾는데 걸리는 시간은 길어지고 좋은 해를 찾을 가능성이 줄어든다. 본 연구에서는 값이 좋은 개체로 구성된 초기 모집단을 생성시키기 위해 G&T 알고리즘을 사용한다. 모집단 구성에서 중요한 것은 다양성이라 할 수 있는데, G&T 알고리즘은 job shop 일정계획 문제에서 다양한 active 스케줄을 가장 쉽게 산출할 수 있는 알고리즘이다. G&T 알고리즘으로 active, active', non-delay 스케줄을 산출하여, 이 스케줄들로 초기 모집단을 구성한다. active와 non-delay 스케줄은 G&T 알고리즘을 이용하여 생성하고, active' 스케줄은 단계 3에서 G집합을 구성하는 방법을 변형한 G&T 알고리즘을 사용한다. 그 알고리즘은 다음과 같다.

2.2.1 기호

- n : job의 수
- m : 기계의 수
- j : job의 번호(j=1, 2, . . . , n)
- k : job의 공정번호 (k=1, 2, . . . , m)
- $P_{j(k)}$: job j의 k번째 공정의 가공시간
- $M_{j(k)}$: job j의 k번째 공정의 가공기계
- $r_{j(k)}$: job j의 k번째 공정의 시작시간

2.2.2 G&T 알고리즘(active 스케줄)

단계1) 각 job들 중에서 가장 먼저 스케줄 해야 할 공정들을 집합 C로 둔다. 집합 C 내의 모든 공정 (j, k)에 대한 시작시간 $r_{j(k)} = 0$ 이라 둔다.

단계2) $t(C) = \min_{(j, k) \in C} \{r_{j(k)} + d_{j(k)}\}$ 를 계산한다.

그리고 t(C)가 최소가 되는 기계 $M_{j(k)}$ 를 m^* 로 둔다.

단계3) 기계 m^* 상에서 $r_{jm^*} < t(C)$ 인 모든 공정 (j, m^*)들을 집합 G로 둔다.

단계4) 집합 G에서 하나의 공정을 임의대로 선택하고 그것을 스케줄 한다.

단계5) C에서 그 공정을 삭제한다. 그리고 job에서 그 공정의 후행 공정을 집합 C에 포함시킨다. C에서 $r_{j(k)}$ 를 수정하고 모든 공정이 스케줄될 때까지 단계 2)로 돌아간다.

non-delay 스케줄은 다음 알고리즘으로 얻어진다.

단계1) 각 job들 중에서 가장 먼저 스케줄 해야 할 공정들을 집합 C로 둔다. 그리고 C 내의 모

든 공정 (j, k)에 대한 시작시간 $r_{j(k)}=0$ 이라 둔다.

단계2) $t(C) = \min_{(j,k) \in C} \{r_{j(k)}\}$ 를 계산한다.

그리고 $t(C)$ 가 최소가 되는 기계 $M_{j(k)}$ 를 m^* 로 둔다.

단계3) 기계 m^* 상에서 $t(C)=r_{jm^*}$ 인 모든 공정 (j, m^*) 들을 집합 G로 둔다.

단계4) 집합 G에서 하나의 공정을 임의대로 선택하고 그것을 스케줄 한다.

단계5) C에서 그 공정을 삭제한다. 그리고 job에서 그 공정의 후행 공정을 집합 C에 포함시킨다. C에서 $r_{j(k)}$ 을 수정하고 모든 공정이 스케줄 될 때까지 단계 2)로 돌아간다.

2.2.3 수정된 G&T 알고리즘(active' 스케줄)

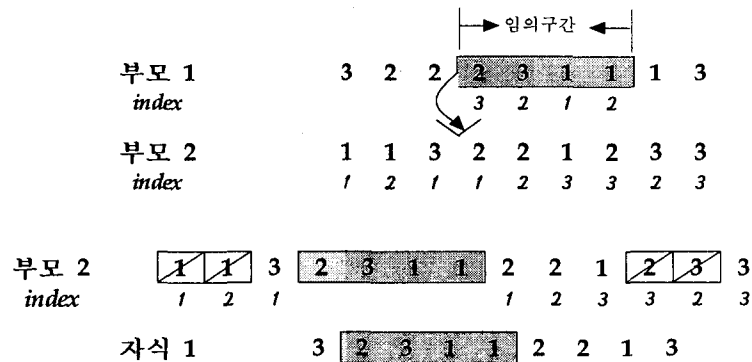
수정된 G&T 알고리즘은 active 스케줄을 산출하는 알고리즘의 단계 3이 아래와 같이 수정되고, 여기서 얻어지는 스케줄을 active' 스케줄이라 한다.

단계3) 기계 m^* 상에서 $r_{jm^*} < t(C)$ 인 모든 공정 (j, m^*) 들 중에서 r_{jm^*} 이 가장 작은 공정들을 집합 G로 둔다.

2.3 교차연산자(crossover)

교차연산자는 G&T 알고리즘으로 얻은 염색체들이 좋은 스케줄을 가지고 있기에 가능하면 그들 순서관계를 유지하면서 조금씩 개선시켜 나갈 수 있는 연산자가 필요하다. 그래서 순서관계를 유지하고 이를 유전시킬 수 있는 교차 연산자를 개발하였다.

제안된 교차연산자는 먼저 임의 구간을 산출한 뒤 그 구간 내 유전인자들을 부모 2에 삽입한다. 삽입 위치는 부모 1의 임의 구간 시작위치 바로 앞이다. 만약 첫 번째 부모에서 임의구간의 시작 위치가 4번째라면 삽입 위치는 부모 2의 3번째 유전인자 다음이 된다. 그리고 나서 임의 구간내의 유전인자와 속성이 같은 인자들을 부모 2에서 삭제한다. 이들 과정을 부모 1과 2를 바꾸어서 수행하여 두 개의 자식 개체를 생성한다. 그리고 두 자식 개체 중 평가기준에 적합한 한 개체만을 다음 세대로 보낸다. 그 과정은 <그림 1>과 같다.



부모 1과 2를 바꾸어 같은 구간으로 두 번째 자식을 생성한다.

자식 2: 3 2 2 1 2 3 1 1 3

<그림 1> 교차연산자 과정

2.4 돌연변이 연산자

유전 알고리즘 내의 돌연변이 연산자는 또 다른 중요 연산자이다. 돌연변이 연산자는 염색체에 변화를 주어 집단내의 다양성을 유지하기 위하여 사용한다. 본 연구에서는 이웃 탐색 기법에 근거한 돌연변이 연산자를 사용한다. <그림 2>는 돌연변이 연산의 예를 보여준다. 이 돌연변이 연산자는 case 2에서 case 6까지의 부모와 다른 5가지 이웃으로 비교하여 가장 좋은 것을 유전시키는 방법과 현 부모와 5개의 이웃 모두를 비교해서 가장 좋은 것을 다음 세대로 전달하는 두 개의 형태로 사용한다. 전자의 경우는 일반적인 돌연변이 과정에서 사용하고 후자의 경우는 교차 연산자 수행 후 더 나은 개체 생성 가능성을 확인하기 위해 사용하는 돌연변이 연산자로 사용한다.

부모 염색체	1	2	3	1	2	3	1	2	3
	↓			↓				↓	
이웃해 염색체									
case 1	1	2	3	1	2	3	1	2	3
case 2	2	2	3	1	1	3	1	2	3
case 3	2	2	3	1	3	3	1	2	1
case 4	3	2	3	1	2	3	1	2	1
case 5	3	2	3	1	1	3	1	2	2
case 6	1	2	3	1	3	3	1	2	2

<그림 2> 돌연변이 연산의 예

2.5 선택

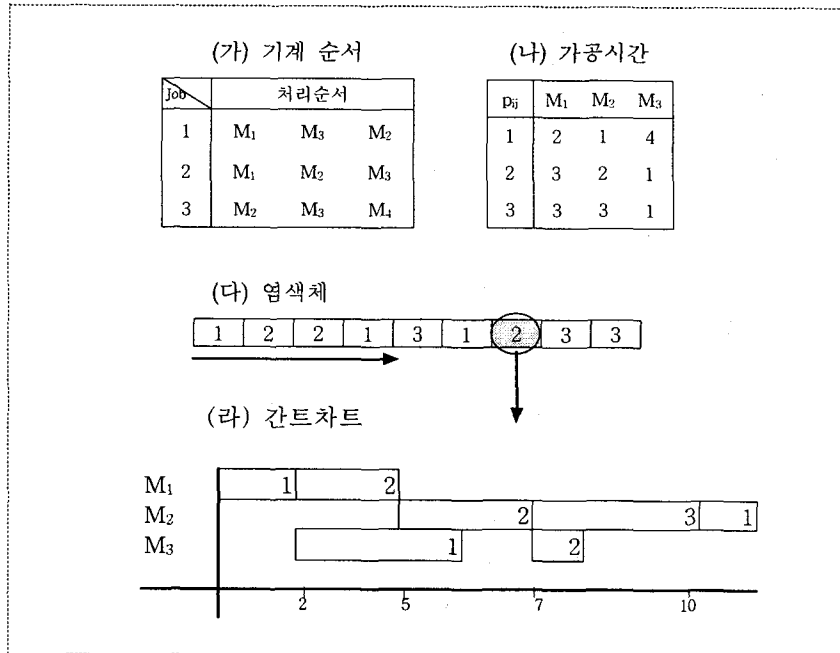
선택은 Brindle[1]이 제시한 경쟁선택을 사용한다. 경쟁선택은 집단으로부터 두 개의 개체들을 임의대로 선택한 다음, 난수 r을 0과 1사이에서 발생시켜 만약 $r < k$ 이면 두 개의 개체 중 더 적합도가 높은 개체를 부모로 선택한다. 그렇지 않으면 덜 적합한 개체를 선택한다. 그런 다음 이 두 개체는 다시 원래의 개체집단에 되돌려지고 다시 선택되어질 수 있다. 여기서 k는 선택 확률을 나타내는 파라미터이다.

2.6 교체

다음 세대의 구성은 현 세대에서 선택과 유전 연산자들을 이용하여 새롭게 구성한다. 새로운 개체들을 초기 모집단의 개수만큼 생성하여 다음 세대를 구성하고 난 뒤 엘리티즘을 적용하여 나쁜 개체는 엘리티즘 적용 개수만큼 좋은 개체로 다시 대체한다. 또한 교차율과 돌연변이율에 따라 일부 개체들은 유전 연산자를 거치지 않고 그대로 다음 세대로 이동하도록 한다.

2.7 목적함수

JSSP에서 최소의 makespan을 가진 스케줄은 종종 기계의 높은 효율을 의미한다. 대부분의 정적 JSSP 벤치마크 문제의 목적들이 makespan을 최소로 하는 것이므로, makespan은 JSSP에서 일정계획 기법의 비교나 평가를 위해 선택한다. 순열 형태의 염색체로 표현하였을 때 makespan은 왼쪽에서 오른쪽으로 유전인자를 읽어, job의 선후관계를 지키면서 기계에 할당하여 구한다. 그 과정은 <그림 3>과 같다.



<그림 3> 검색체에서 makespan 구하는 과정

3. 병렬 유전알고리즘

전통적인 유전 알고리즘에서 일반적인 문제는 조기 수렴이다. 유전 알고리즘은 다른 국소 탐색 절차 보다 조기 수렴을 방지하면서도 이의 영향을 받는다. 유전 알고리즘에서 조기 수렴을 감소시키기 위한 방법은 초기 모집단을 여러 부 집단으로 분리하는 유전 알고리즘의 병렬화이다. 병렬 유전 알고리즘(Parallel Genetic Algorithms: PGA)는 더욱 독립적으로 전개되는 분리된 부 집단을 유지함으로써 조기 수렴을 막을 수 있다. PGA는 각 부 집단을 위한 코드의 시간 분할 실행으로, 병렬 프로세서 또는 하나의 프로세서에서 수행될 수 있다. PGA의 이점은 병렬 프로세서의 사용을 통해 수행 속도를 줄일 수 있다는 것이다. 일반적으로 PGA에는 섬 모델과 이웃 모델의 두 종류의 모델이 있다.

3.1 섬 모델 PGA

섬 모델 PGA는 여러개의 부 집단을 가진다. 그들 각각은 SGA로 활동한다[11]. 그리고 어떤 구간내의 일부 개체들은 하나의 부 집단으로부터 다른 부 집단으로 이주할 수 있다. 섬 모델 PGA는 이주 방법, 연결 구조, 부 집단의 동질성에 따라 분류된다. 이주 방법으로 이주가 없는 고립된 섬 모델 PGA, 부 집단 사이의 이주가 동시에 이루어지는 동시성 섬 모델 PGA, 동시에 이루어지지 않는 비동시성 PGA로 나눌 수 있고, 연결 구조로는 부 집단들 사이의 연결구조가 시작에서 설정되어 수정되지 않는 정적 연결 구조와 변화하는 동적 연결 구조 PGA로 나눈다. 부 집단의 동질성으로 각 부집단이 같은 파라미터를 사용하는 동종(同種) 섬 모델 PGA와 다른 파라미터를 사용하는 이종(異種) 섬 모델 PGA로 나눌 수 있다.

3.2 제안 PGA

본 연구에서는 섬 모델 PGA를 제안한다. 이 섬 모델 PGA에서 부 집단들의 연결은 정적 연결

구조를 가지고, 이종(異種) 섬모델 PGA로 구성된다. 이주는 동시에 이루어지도록 하였다.

이종 섬모델 PGA에서 부 집단이 개수가 2개인 경우와 4개인 경우 두 가지로 설계하는데 처음의 경우를 2S(two Subpopulation)-PGA라 하고 후자를 4S(four Subpopulation)-PGA라 한다. 여기서 2S-PGA에서 2개로 분리된 초기 부 집단은 active' 스케줄과 non-delay 스케줄로 구성한다. 그리고 각 부 집단에 SGA에서의 유전 연산자와 선택방법을 적용한다. 두 부 집단이 50세대 진행되면 동시에 일정 개체 수를 이주하도록 하였다. 4S-PGA에서는 4개로 분리된 초기 부 집단을 active', active, non-delay, random 스케줄로 구성하여 각각의 부 집단에 SGA에서의 유전연산자를 적용하고, 2S-PGA와 같이 50세대 진행 후 이주하도록 하였다. 이주개체 선별과 대체 방법은 이주시기에 각 부 집단의 개체를 정한 개수만큼 임의대로 선별해서 서로 교환하는 방법과 각 부 집단에서 정해진 순위 안에 드는 개체를 복사하여 연결되어 있는 다른 부 집단의 나쁜 개체를 교체하는 방법 중 좋은 결과를 산출한 후자의 이주 방법을 사용한다.

4. 실험 및 분석

제안알고리즘을 두 개의 유명한 벤치마크 문제, FT(Fisher & Thompson) 문제[12]와 YN 문제[15]에 적용한다. FT 문제는 제안되는 거의 모든 job shop 일정계획 기법들이 벤치마크 문제로 사용하고 있고, YN 문제는 가장 큰 규모의 벤치마크 문제라서 선택하였다. 실험에 사용한 컴퓨터는 펜티엄 II 350, 메모리 64M의 기종을 사용하였다.

4.1 유전 파라미터의 설정

제시한 알고리즘에 적용할 교차율, 선택확률, 엘리티즘 크기는 FT10 문제에 하나의 초기 모 집단을 구성하고, 파라미터에 따라 100번 실행하여 가장 좋은 값과 평균값을 얻어내는 값으로 하였다. 교차율이 0.6, 0.7, 0.8일 경우, 선택확률이 0.7, 0.75, 0.8일 경우, 엘리티즘의 크기가 10, 20, 30일 경우에 대해 실험하였다. 돌연변이율은 0.1로 하였다.

<표 2>에서 교차율 0.8, 선택확률 0.75, 엘리티즘 크기를 10으로 했을 경우가 평균해가 가장 좋음을 알 수 있다.

<표 2> 파라미터 실험 결과

파라미터			평균해	최고해
엘리티즘	교차율	선택확률		
10	0.6	0.75	960.27	951
	0.7	0.75	958.76	945
		0.7	959.69	945
		0.8	959.26	951
	0.8	0.75	957.49	937
20	0.6	0.75	961.27	935
	0.7	0.75	958.83	937
		0.7	960.42	951
		0.8	960.54	951
	0.8	0.75	960.67	951
30	0.6	0.75	962.57	940
	0.7	0.75	962.53	951
		0.7	963.62	937
		0.8	959.52	951
	0.8	0.75	960.05	951

4.2 유전알고리즘의 성능평가

실험의 결과는 각 문제마다 새로운 초기 모집단으로 50번씩 수행하여 가장 좋은 해로 하였다. 그리고 SGA는 집단 크기 200, 세대수 1000으로 하고, PGA는 2S-PGA일 경우 집단 크기 100, 세대수 1000, 이주크기 10, 4S-PGA일 경우 집단크기 50, 세대수 1000, 이주크기 5로 하였다. 이전 연구자들이 한번의 실행에서 150,000개에서 300,000개의 염색체를 평가하고 있어, 200,000개로 이전 연구자들과 비교 횟수를 맞추기 위해 이런 집단 크기와 세대수로 정하였다.

4.2.1 FT 벤치마크 문제

일정계획 기법의 수행도를 평가하기 위해 먼저 가장 유명한 세 개의 벤치마크 문제 FT6×6, FT10×10와 FT20×5를 테스트하였다. 이들 FT 문제들은 제안되는 거의 모든 job shop 일정계획 기법들이 벤치마크 문제로 사용하고 있다.

FT 문제에 SGA, PGA를 기반으로 한 일정계획 기법을 적용시켰다. 두 기법 모두 FT6 문제에서는 55로 최적해를 얻을 수 있었으며 FT10, FT20 문제에서는 각각 930, 1173으로 최적해와 근접해를 구해낼 수 있었다. <표 3>은 세 개의 FT 문제에서 이전의 연구에 의해 얻어진 최고의 결과들이다. 이 표에서 이전 연구의 결과들에 비해 제시한 기법이 현저한 해의 개선을 이루어 낼 수 있다. P_c 는 교차율, P_m 은 돌연변이율을 나타낸다.

<표 3> FT 벤치마크 문제의 파라미터와 결과

비교논문	FT6 (6×6)	FT10 (10×10)	FT20 (20×5)	집단크기	세대수	P_c	P_m
최적해	55	930	1165	-	-	-	-
Nakano & Yamada[13]	55	965	1215	1000	150	-	-
Yamada & Nakano[15]	55	930	1184	2000	100	-	-
Paredis[14]	-	1006	-	-	-	-	-
Gen[7]	55	962	1175	60	5000	0.4	0.3
Fang[5]	-	949	1189	-	-	-	-
Dorndorf1 & Pesch[4]	55	960	1249	200	-	0.65	0.001
Dorndorf2 & Pesch[4]	55	938	1178	40	-	0.65	-
Croce[3]	55	946	1178	300	2971	1	0.03
Cheng[7]	55	948	1196	40	2000	0.4	0.4
제안 SGA	55	930	1173	200	1000	0.7	0.1
제안 2S-PGA	55	930	1178	100	1000	0.7	0.1
제안 4S-PGA	55	930	1173	50	1000	0.7	0.1

SGA, PGA 기반 일정계획 기법의 최고해는 거의 차이가 없으나 50번의 수행으로 얻은 평균 해는 PGA기반 일정계획 기법이 우수한 것을 <표 4>에서 볼 수 있다. 이는 짧은 시간의 수행에서는 더 나은 해를 구해낼 수 있음을 보여준다.

<표 4> FT 문제에서 SGA와 PGA 기반 일정계획기법의 결과 비교

문제	SGA		2-PGA		4-PGA	
	최고해	평균해	최고해	평균해	최고해	평균해
FT10	930	977.48	930	972.38	930	970.12
FT20	1173	1217.24	1178	1211.29	1173	1204.44

4.2.2 YN 벤치마크 문제

Yamada와 Nakano[15]에 의해 만들어진 YN문제는 가공시간이 구간 (10, 50)에서 일양분포로 산출된 4개의 20×20 문제이다. Yamada와 Nakano는 G&T 알고리즘 기반 연산자인 GA/GT 교차연산자를 설계하였다. 이들은 4개의 YN 문제에 이 연산자를 적용하였다. 계산시간은 언급되어 있지 않다. 그들은 각 문제에서 임의대로 산출된 400,000개의 active 스케줄과 GA/GT의 결과들을 비교하였다. <표 5>는 SGA, PGA와 GA/GT 그리고 임의대로 산출된 active 스케줄의 결과를 보여준다. 파라미터는 FT문제와 같이 하였다. 3개의 문제에서 제시한 일정계획 기법이 GA/GT 보다 우수한 것을 볼 수 있다. 그리고 PGA가 SGA 보다 우수함을 볼 수 있다.

<표 5> YN 벤치마크 문제의 계산 결과

문제	문제크기	active 스케줄	GA/GT	SGA	PGA
		최고해	최고해	최고해	최고해
YN1	20×20	1126	967	925	925
YN2	20×20	1104	945	966	944
YN3	20×20	1107	951	941	936
YN4	20×20	1202	1052	1027	1027

5. 결론

제한된 자원 하에서 이루어지는 작업들의 일정계획은 생산관리의 효율성 측면에서 매우 중요하다. 본 연구에서는 JSSP를 위해 유전 알고리즘을 기반으로 한 일정계획 기법을 제안하였다. 제안한 일정계획 기법에서 표현은 공정의 순서를 job의 번호로 코드화 하여 항상 해가 실행 가능하도록 하였고, 이를 G&T 알고리즘과 연결하여 초기 모집단을 구성하였다. 그리고 교차 연산자와 돌연변이 연산자는 스케줄에서 일시적인 관계를 보다 좋게 유전시키도록 하여 초기의 좋은 스케줄이 계속 진화될 수 있도록 하였다. 이 기법을 표준 벤치마크 JSSP에 적용하여 좋은 결과들을 산출할 수 있었다. 이는 초기 모집단의 생성과 유전 연산자의 성공적인 통합을 의미하는 것으로 제시된 일정계획 기법의 효율성을 보여준다.

SGA에서 사용한 연산자와 선택방법을 가지고 이종(異種) 섬모델 PGA도 제시하였다. 섬 모델 PGA에서 섬의 수는 단순히 집단 크기를 증가시키는 것보다 더 큰 효과를 얻을 수 있다. 본 연구에서 제시한 PGA를 기반으로 한 일정계획 기법은 벤치마크 문제에서 SGA를 기반으로 한 기법과의 비교를 통해 평균 해와 해의 개선을 확인할 수 있었다. 이는 조기 수렴의 문제를

완화시켜 결과들을 개선하고, 더 적은 평가로도 SGA와 같은 수준의 결과를 얻을 수 있음을 보여주는 것이라 할 수 있다. PGA를 기반으로 한 기법의 수행도는 이주 세대, 이주 크기, 이주 방법 등의 개선으로 인해 더 높아질 것으로 기대된다.

참고문헌

- [1] Brindle, A., Genetic Algorithms for Function Optimization, Unpublished Ph. D. thesis, University of Alberta, Edmonton, Canada, 1981.
- [2] Bäck, T., Evolutionary Algorithms in Theory and Practice: Evolutionary Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University Press, 1996.
- [3] Croce, F.D., Tadei, R., and Volta, G., "A Genetic Algorithm for the Job Shop Problem", Computer & Operations Research, 22(1): 15-24, 1995.
- [4] Dorndorf, U. and Pesch, E. "Evolution based Learning in a Job Shop Scheduling Environment", Computers & Operations Research, 22(1): 25-40, 1995.
- [5] Fang, H., Ross, P. and Corne, D., "A Promising Genetic Algorithm Approach to Job-Shop Scheduling, Rescheduling, and Open-Shop Scheduling Problems", Proc. Fifth Int'l Conf. on Genetic Algorithms, Morgan Kaufmann, San Mateo, pp.375-382 1993.
- [6] Garey, M.R., Johnson, D.S., and Sethi, R., "The Complexity of Flowshop and Job shop Scheduling", Math. Oper. Res., 1: 117-129, 1976.
- [7] Gen M, Cheng R., Genetic algorithms and engineering design, New York, John Wiley & Sons, 1997.
- [8] Giffler, J. and Thompson, G.L., "Algorithms for Solving Production Scheduling Problems", Operations Research, 8: 487-503, 1960.
- [9] Goldberg, D., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
- [10] Holland, H.J., Adaptation in Natural and Artificial Systems, MIT Press, Cambridge, 1992.
- [11] Lin, S. C, Punch, W.F. and Goodman, E.D., "Coarse-Grain Parallel Genetic Algorithms: Categorization and New Approach", IEEE SPDP, pp.28-39, 1994.
- [12] Muth, J.F. and Thompson, G.L., Industrial Scheduling, Prentice-Hall, Englewood Cliffs, N.J., 1963.
- [13] Nakano, R. and Yamada, T., "Conventional Genetic Algorithms for Job Shop Problems", Proc. Fourth Int'l Conf. on Genetic Algorithms, Morgan Kaufmann, San Mateo, pp.474-479, 1991.
- [14] Paredis, J., "Exploiting Constraints as Background Knowledge for Genetic Algorithms: a Case-study for Scheduling", Parallel Problem Solving from Nature, 2, North-Holland, Amsterdam, pp.229-238, 1992.
- [15] Yamada, T. and Nakano, R., "A Genetic Algorithm Applicable to Large-Scale Job Shop Problems", Parallel Problem Solving from Nature, 2, North-Holland, Amsterdam, pp.281-290, 1992.