

## 傳統적인 Job Shop 日程計劃을 위한 혼합유전 알고리즘의 開發 A Development of Hybrid Genetic Algorithms for Classical Job Shop Scheduling

정종백<sup>1)\*</sup>, 김정자<sup>1)</sup>, 주철민<sup>2)</sup>

<sup>1)</sup>동아대학교 산업시스템공학과, <sup>2)</sup>동서대학교 산업공학과

jbjung@shinbiro.com, jjkim@mail.donga.ac.kr, cmjoo@dongseo.ac.kr

### Abstract

Job-shop scheduling problem(JSSP) is one of the best-known machine scheduling problems and essentially an ordering problem. A new encoding scheme which always give a feasible schedule is presented, by which a schedule directly corresponds to an assigned-operation ordering string. It is initialized with G&T algorithm and improved using the developed genetic operator; APMX or BPMX crossover operator and mutation operator. and the problem of infeasibility in genetic generation is naturally overcome. Within the framework of the newly designed genetic algorithm, the NP-hard classical job-shop scheduling problem can be efficiently solved with high quality. Moreover the optimal solutions of the famous benchmarks, the Fisher and Thompson's 10×10 and 20×5 problems, are found.

### 1. 서론

요즘 기업에서는 실제적인 스케줄이나 CIM 환경하의 동적인 재스케줄 방법 개발에 관심을 보이고 있다. 이는 간단한 스케줄을 개선함으로써 제조 원가를 줄일 수 있고 작업효율을 향상시킬 수 있기 때문이다. 제한된 가용 자원으로 다양한 목적을 만족시키는 조합최적화 문제를 풀기 위해 부분적인 최적 알고리즘이 개발되어졌지만 JSSP는 문제의 크기에 따라 해결에 필요한 시간이 지수적으로 증가하는 NP-hard 문제이다. 전통적인 JSSP는 가장 어려운 조합 최적화 문제중의 하나이다[1][2].

전통적인 JSSP는 다음과 같이 설명될 수 있다.  $m$ 개의 다른 기계와  $n$ 개의 다른 job들이 스케줄된다. 각 job은 operation들의 집합으로 이뤄지며 기계에서의 operation 순서가 정해지게된다. 각 operation은 요구되는 기계와 고정된 가공시간을 가진다. job과 기계 사이에는 아래의 제약이 있다:

- 어떠한 job도 선점되지 않는다,
- operation간에 우선순위가 주어진다,
- 각 기계는 한번에 한 작업만 처리할 수 있다,
- 시작시간과 종료시간이 명시되지 않는다.

이 문제의 목적은 makespan을 최소화하고 기계에서의 operation 순서를 결정하는 것이다.

이 문제는 1960년대 후반에 B&B 기법으로 다뤄졌으며, Carlier and Pinson[2]는 Fisher and Thompson(1963)이 제시한 세 가지 유명한 벤치마크에 대한 최적해를 증명하였다. 1980년대 중반이후부터 유전 알고리즘(GA)에 대한 연구가 활발히 이루어져, 많은 연구자들이 스케줄링 문제에 다양한 스키마를 적용하였다. Nakano and Yamada (1991)가 최초로 전통적인 JSSP 해결을 위해 설계하였지만 양질이 아닌 근사해를 얻을 수 있었다. 그리고 Kobayashi et. al.(1995)는 'subsequence exchange crossover'를 이용하여 더 좋은 결과를 얻지만 상대적으로 오랜 계산시간을 필요로 했다.

그 외 다양한 문헌들이 있지만 이들의 주요 쟁점은 encoding method에 있었다.

AI기법중의 하나이며, 자연 진화설을 바탕으로 한 유전 알고리즘은 다수 탐색점을 사용하여, 적절한 다양성(diversity)을 유지하면서 우수한 후보해들 간의 정보를 교환하기 때문에 최적해에 도달할 확률이 높다. 또한 최적해를 발견하지 못하는 경우에도 이 알고리즘은 좋은 해를 제공할 수 있는 견실성(robustness)도 지니고 있다. 이러한 장점들 때문에 유전 알고리즘은 NP-hard 문제에 대한 탐색 방법으로 효율적이라 알려져 있다[3][4].

따라서 본 논문 2장에는 유전 알고리즘에 대해 살펴보고, 3장에서는 개발한 혼합유전 알고리즘, 4장에서는 전통적인 JSSP형태의 유명한 세 가지 벤치마크 문제에 적용하는 실험을 통해 우수성을 검증하고 마지막 5장에서 결론을 맺는다.

### 2. 유전 알고리즘[5]

유전 알고리즘은 탐색과 최적화 문제를 해결하기 위한 방법 중의 하나이며, 생물의 진화 원리로부터 착상을 얻은 일종의 학습 알고리즘이다.

유전 알고리즘은 변수 자체를 표현하는 것이 아니라 변수를 코드화(coding)하여 유전자(gene)을 만든 다음 단일해가 아닌 여러 해들의 집단인 첫 세대를 무작위 발생시키고 이들 중에서 다음 세대를 만들기 위한 부11모해들을 선택한다. 우수한 해는 부모해로 선택될 확률이 높다. 일단 선택된 부모해는 유전 알고리즘의 일반적인 연산자인 재생과 교차와 돌연변이를 통해 다음 세대인 자손 세대를 만든다. 이 과정을 유전집단이 수렴할 때까지 반복한다. 즉, 설계변수를 코드화하고 다수 설계점을 동시에 폭넓게 탐색하고 기본 모델의 수정이 용이하다는 장점이 있다[5].

### 3. 개발한 혼합 유전 알고리즘

위와 같은 유전알고리즘의 장점에도 불구하고 기존의 유전 해법을 job shop 일정계획에 적용시킬 경우 근본적으로 다음과 같은 문제점을 내포하고 있다.

- (1) 유전알고리즘은 좋은 초기집단과 적당한 집단의 크기를 선택함으로써 효율성을 크게 향상시킬 수 있지만 유전알고리즘은 근본적으로 난수를 이용하여 탐색을 수행하기 때문에 문제를 풀 때마다 해가 달라지고, 적용된 유전 연산자의 종료와 적용확률에 따라 해가 달라지는 등의 임의성이 너무 많이 존재한다.
- (2) 유전알고리즘은 설계공간내의 여러 점들을 동시에 탐색하므로 국부적인 최적점에 빠질 가능성이 적으나, 탐색점의 증가로 인하여 계산시간이 급격히 증가하는 특성을 갖고 있다.
- (3) 유전알고리즘의 표현방법은 연산자를 사용하여 가능해 공간을 효율적으로 탐색할 수 있도록 흔히 많은 문제에 이진수로 표현하고 있으나 job shop 일정계획문제와 같은 순서문제는 이진수로 표현하기가 어렵고 연산시에 가능해 유지의 어려움을 내포하고 있다.

따라서, 본 연구에서는 이와 같은 문제점을 해결하기 위해 우선 유전 알고리즘의 초기 해를 Giffler and Thompson(G&T)의 휴리스틱을 이용해 구한 다음 이 초기해 개선을 위해 유전알고리즘을 적용시킴으로써 전체 탐색시간을 줄이는 방법을 개발하였다. 이 방법은 좋은 초기해를 제공하는 G&T 알고리즘과 국부적인 최적점에 빠질 가능성이 적은 유전 알고리즘을 연결하여 수행하면, 탐색시간을 줄이고 전체적인 최적점을 보다 효율적으로 얻을 수 있다는 기본개념에서 출발하였다. 또한 일반적인 유전 알고리즘과는 다른 할당된 작업순서 기반(assigned-operation ordering based) 표현기법을 사용함으로써 계속된 진화에도 가능해를 유지할 수 있도록 하였다. 먼저 G&T 알고리즘의 절차는 다음과 같다[6]. G&T 알고리즘은 active 스케줄을 생성하는 휴리스틱이다.

- 단계1. 각 job들 중에서 가장 먼저 스케줄 해야 할 작업(operation)들을 집합 C로 둔다. 그리고 C 내의 모든 작업 (j, m)에 대해 시작시간  $r_{jm}=0$  이라 둔다. ( $r_{jm}$  은 작업 (j, m)이 시작 할 수 있는 가장 빠른 시작시간)
- 단계2.  $t(C) = \min_{(j,m) \in C} \{r_{jm} + p_{jm}\}$  를 계산한다. ( $p_{jm}$ 은 job j의 기계 m에서 가공시간) 그리고  $t(C)$ 가 최소가 되는 기계를  $m^*$ 로 둔다.
- 단계3. 기계  $m^*$  상에서  $r_{jm^*} < t(C)$  인 모든 작업 ( $j, m^*$ )들을 집합 G로 둔다.
- 단계4. 집합 G에서 하나의 작업을 임의대로 선택하고 그것을 스케줄 한다.
- 단계5. C에서 그 작업을 삭제한다. 그리고 그 job에서 그 작업을 바로 뒤따르는 작업을 C에 포함시킨다. C에서  $r_{jm}$ 을 수정하고 모든 작업이 스케줄 될 때까지 단계2.로 돌아간다. (여기서 새롭게 수정되는  $r_{jm}$ 은 선행 작업의 완성시간과 사용해야할 기계의 가장

늦은 완료시간 중 큰 것으로 한다). 제안된 알고리즘의 흐름은 아래 그림 1.과 같다.

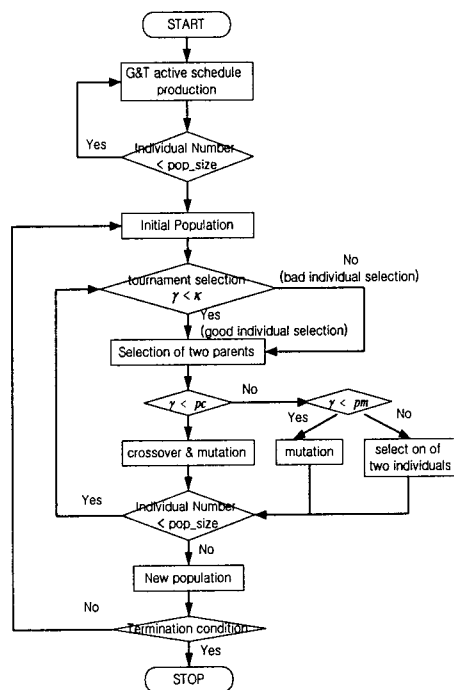


그림 1. 개발한 유전 알고리즘의 흐름도

선별(Selection)은 토너먼트방법을 사용며 K 값에 따라 정렬된 초기집단의 영역을 결정하고 그 영역내에서 임의로 선별하는 방법을 사용하였다.

#### 2.1 유전자 표현(Representation)

아래의 그림 2를 보면 알 수 있듯이, 3번 기계에서 3의 작업시간이 소요되는 3번 job의 첫 번째 operation이 먼저 스케줄 되고 그 다음 2번 기계에서 1의 공정시간이 소요되는 2번 job의 첫 번째 operation이 할당되어 하나의 스케줄을 구성한 것을 나타낸다. 할당한 operation의 job 번호를 순서대로 나열한 이 표현 방법은 진화가 되더라도 항상 실행가능한 스케줄을 구성할 수 있어서 기존의 실행불가능해 산출의 문제를 해결 할 수 있다.

Job	Machine Routing (Processing Time)		
1	1(1)	2(3)	3(6)
2	2(1)	3(2)	1(3)
3	3(3)	2(3)	1(1)

chromosome 

3	2	2	2	3	1	1	1	3
---	---	---	---	---	---	---	---	---

- 3번 부품의 1번째 작업
- 2번 부품의 1번째 작업
- 2번 부품의 2번째 작업

그림 2. 개발된 유전자 표현

#### 2.2 교차(Crossover)

본 논문에서 개발한 두가지 연산자는 Goldberg and Lingle(1985)에 의해 제안된 PMX(partially mapped crossover; 부분사상교차)연산자를 변형한

것이다. PMX는 두 절단점 사이의 각 인자들을 사상시켜, 사상되는 인자쌍의 위치를 상호교환 하여 자손을 생성하는 방법이다. 따라서 순서문제는 인자들의 교환이 해의 개선에 상당히 영향을 미치는 경우이므로 부분사상 교차의 사용은 의미가 있을 것이다.

2.2.1 APMX(After PMX) 교차 연산자

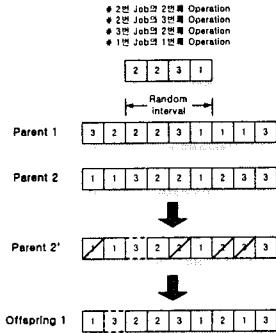


그림 3. APMX 교차의 예

- 단계 1. 부모 1에서 임의의 구간 난수를 발생시킨다.
- 단계 2. 부모 1에서 선택한 임의구간(2 2 3 1)을 부모 2와 비교 후, 그 의미와 일치하는 job의 operation들을 순서대로 지운다.
- 단계 3. 임의구간의 첫 작업이 부모 1의 3번째 유전자였으므로 지운 후 세번째 위치에 그 구간이 위치하도록 삽입하여 자손을 생성한다.

2.2.2 BPMX(Before PMX) 교차 연산자

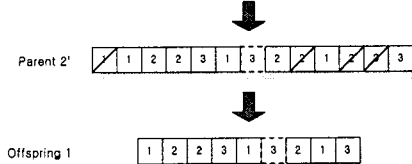


그림 4. BPMX 교차의 예

- 단계 1. 부모 1에서 임의의 구간을 선정한다.
- 단계 2. 부모 2에서 임의구간의 의미에 해당 하는 유전자를 삭제하기 전에 삽입한다 (Parent 2'). 이는 그 위치를 유지시키기 위한 PMX 교차 정의를 준수하기 위함이다.
- 단계 3. 임의구간의 의미대로 순서대로 삭제한다.

2.3 돌연변이(Mutation)

돌연변이 연산자는 집단내의 다양성을 유지하기 위하여 염색체에서의 조금의 변화를 만들기 위해 사용된다.

2.3.1 Pairwise 돌연변이 연산자

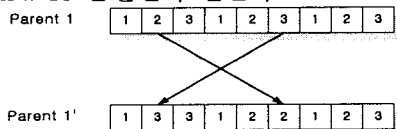


그림 5. Pairwise 돌연변이의 예  
Lin은 블록상의 두 공정들을 임의대로 선택하고

교환하는 방법을 사용하였다. 이 방법을 적용함으로써 실행불가능한 해가 생길 수 있지만 본 논문에서는 항상 가능한 표현방법을 사용함으로써 이 문제를 근본적으로 해결하였다.

2.3.2 Neighborhood 돌연변이 연산자

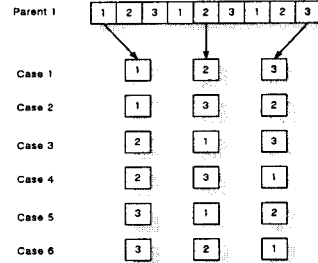


그림 6. Neighborhood 돌연변이의 예

임의의 서로 다른 값을 가지는 세 위치의 유전자로 생길 수 있는 염색체의 평가함수에 따라 채택 여부를 결정한다.

3. 실험결과

본 논문에서는 JSSP의 가장 유명한 벤치마크 문제인 FT6×6, FT10×10 그리고 FT20×5에 대해 개발한 알고리즘을 적용하는 실험을 하였다.

3.1 파라미터에 따른 결과치 비교

먼저, 파라미터를 정하기 위한 실험을 하였다. 표 1은 200개의 염색체로 이루어진 모집단으로 1000세대 수행한 결과의 예를 나타낸다. 그 결과 확인한 차이는 없었지만 연산자 BPMX가 약간 더 우수함을 알 수 있었다.

표 1. 파라미터에 연산자의 비교

Operator	Parameter				Best	Frequency 930-939	Average
	Elitism	Cross-over rate	mutation rete	Selection probability			
APMX	20	0.7	0.1	0.75	936	4	980.91
APMX	10	0.7	0.1	0.7	936	2	980
APMX	10	0.8	0.1	0.75	937	2	982.68
BPMX	10	0.8	0.1	0.75	930	4	977.48
BPMX	20	0.6	0.1	0.75	935	3	980.89
BPMX	20	0.7	0.1	0.75	938	2	981.18

아래의 표 2에서 알수 있듯이 FT6×6과 FT10×10 문제에서는 최적해 55와 930을 구했으며 FT20×5문제에서도 최적해는 아니지만 상당히 좋은 해 1173을 구할 수 있었다. 그림 7은 최적해를 구한 두 경우의 수렴곡선을 나타내고 있다.

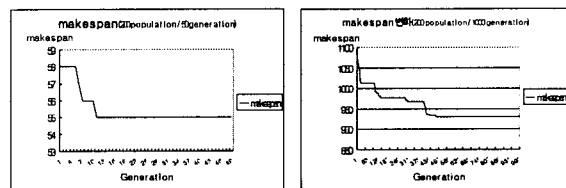


그림 7. 두 벤치마크에서 얻은 최적해의 수렴곡선

표 2. FT 벤치마크문제에 적용한 결과치와 파라미터 비교

Authors	6×6	10×10	20×5	pop size	max gen	p <sub>c</sub>	p <sub>m</sub>
Optimal	55	930	1165	-	-	-	-
Nakano and Yamada(1991)	55	965	1215	1000	150	-	-
Yamada and Nakano(1992)	55	930	1184	2000	100	-	-
Paredis(1992)	-	1006	-	-	-	-	-
Gen et al. (1994)	55	962	1175	60	5000	0.4	0.3
Fang et al. (1993)	-	949	1189	-	-	-	-
Dorndorf1 and Pesch(1995)	55	960	1249	200	-	0.65	0.001
Dorndorf2 and Pesch(1995)	55	938	1178	40	-	0.65	-
Croce et al. (1995)	55	946	1178	300	2971	1	0.03
Cheng et al. (1995)	55	948	1196	40	2000	0.4	0.4
Jung(1999)	55	930	1173	200	1000	0.7	0.1

3.2 초기 모집단에 의한 결과치 비교

표 3에서 G&T 알고리즘으로 초기모집단을 구성하여 수행하는 제안 알고리즘이 우수하고 타당한 방법론임을 확인할 수 있었다.

표 3. 초기 모집단에 의한 결과치 비교

Crossover	Initial population	Active schedule	Random schedule
	BPMX	Average	981.12
	Best Solution	937	945

3.3 Bierwirth 알고리즘과의 비교

그림 8과 그림9에서 알 수 있듯이 제안 알고리즘은 아주 나쁜값을 도출하지는 않고 중·상의 값을 도출하는 특징있는 알고리즘임을 알 수 있었으며 Bierwirth의 알고리즘[7][8]보다 평균적으로 우수함을 확인할 수 있었다.

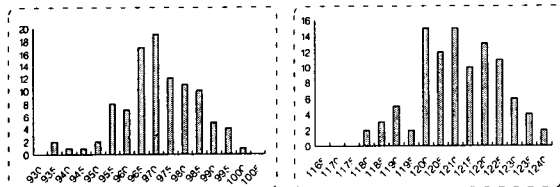


그림 8. Bierwirth의 알고리즘에 의한 100개 해의 분포

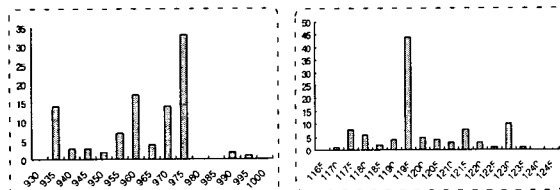


그림 9. 제안 알고리즘에 의한 100개 해의 분포

아래 그림 10은 염색체는 10×10문제에서 얻은 최적해의 스케줄을 나타낸다.

```

6 4 6 4 7 9 10 1 2 7 6 9 5 8 4 6 2 5 4 5 7 9 7 5 6 10 9 8 4
3 7 6 10 2 8 6 7 9 5 4 7 1 2 4 10 3 8 10 5 6 6 3 1 9 8 6 1
3 5 1 9 10 2 7 3 10 5 9 8 1 5 7 7 3 8 10 9 2 3 1 8 2 1 4 2
3 4 1 8 10 8 9 5 2 2 10 3 4 3 1 ⇒ optimal makespan(930)
    
```

그림 10. FT10×10문제에서 얻은 최적해의 예

4. 결론

본 연구를 통하여, JSSP에 적용할 수 있는 효과적인 혼합 유전알고리즘을 제안하였다. 실험을 통하여 임의로 모집단을 구성하는 것 보다 기존의 G&T 휴리스틱과의 접목이 더 효율적임을 확인할 수 있었다. 무엇보다도 순서 결정 문제에 유전 알고리즘을 적용할 경우 세대가 진행되거나 유전 연산자를 적용할 때 실행 불가능한 해를 유발할 수 있는 문제점을 효과적인 스키마를 사용함으로써 근본적으로 해결하였다. 그리고 두가지 효과적인 연산자를 개발하였다.

한편, 본 논문에서는 일반적으로 사용되는 돌연변이 연산자를 활용하여 neighborhood 연산이 약간 좋은 결과를 얻었다. 따라서 향후 연구로는 역위(inversion), 전이(translocation)등의 다른 유전 연산자를 이용하여 작업 일정 계획에 적용하여 유전 알고리즘의 성능을 향상시키기 위한 연구가 필요하다. 또한 본 연구에서 제시된 방법을 실제 생산 현장에 적용한 연구가 필요할 것으로 사료되며 다른 연구의 확장에 기여하리라 본다.

참고문헌

- [1]. J. Adams, E. Balas, and D. Zawack. The shifting bottleneck procedure for job shop scheduling. *Manage. Sci.*, 23(3)391-401, (1988).
- [2]. J. Carlier and E. Pinson, An algorithm for solving the job-shop problem. *Manag. Sci.* 34, 164-176, (1989).
- [3]. David Beasley, David R. Bull, Ralph R. Martin, "An overview of Genetic Algorithms : Part 1, Fundamentals", *University Computing*, 15(2), pp. 58-69, (1993).
- [4]. David E. Goldberg, "Genetic Algorithms in search, optimization & machine learning", Addison Wesley Co, (1989).
- [5]. Holland, J., "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, (1975).
- [6]. Giffler, J. and Thompson, G.L., "Algorithms for Solving Production Scheduling Problems, "Operations Research, Vol. 8, pp. 487-503, (1960).
- [7]. Bierwirth, C., Kopfer, H., Mattfeld, D.C. and Rixen, I., "Genetic Algorithm based Scheduling in a Dynamic Manufacturing Environment", *IEEE Conf. on Evolutionary Computation*, IEEE Press, pp.439-443 (1995).
- [8]. Mattfeld, D.C., Kopfer, H. and Bierwirth, C., "Control of Parallel Population Dynamics by Social Like Behavior of GA-Individuals", *Parallel Problem Solving from Nature*, 3, Springer-Verlag, Berlin, Heidelberg, pp.15-24 (1994).
- [9]. Guoyong. Shi, "A genetic algorithm applied to a classic job-shop scheduling problem", *International Journal of Systems Science*, Vol. 28, No 1, pp. 25-32, (1997).