

A Genetic Algorithm for Minimizing Total Tardiness with Non-identical Parallel Machines

Yu-Jun Choi[†]

Department of Business Administration, Dong-A University

이종 병렬설비 공정의 납기지연시간 최소화를 위한 유전 알고리즘

최 유 준[†]

동아대학교 경영대학 경영학과

This paper considers a parallel-machine scheduling problem with dedicated and common processing machines using GA (Genetic Algorithm). Non-identical setup times, processing times and order lot size are assumed for each machine. The GA is proposed to minimize the total-tardiness objective measure. In this paper, heuristic algorithms including EDD (Earliest Due-Date), SPT (Shortest Processing Time) and LPT (Longest Processing Time) are compared with GA. The effectiveness and suitability of the GA are derived and tested through computational experiments.

Keywords : Non-Identical Parallel-Machine Scheduling, Total-Tardiness, Genetic Algorithm

1. 서 론

본 논문은 제조공정에서 전용설비와 범용설비가 혼재되어 있는 생산시스템에서 납기지연시간 총 합의 최소화를 위한 일정계획 수립을 유전 알고리즘으로 접근하기 위한 연구이다.

제조기업의 경쟁우위 수단으로서 정해진 납기에 공급할 수 있는 능력, 설계 규격에 대한 적합성(conformance with specification)이나 고객 요구조건에 대한 적합성(conformance to requirements), 공정의 표준화 등을 들 수 있다. 경쟁우위 수단을 가진 중·소 제조 기업이라 하더라도 설비에 대한 투자나 추가적인 설비증설을 빈번히 할 수 없기 때문에 기존 설비의 활용을 극대화하지 않을 수 없다.

특히 제조공정에서 전용설비와 범용설비를 혼용하여 운영하는 경우는 그 설비의 활용을 극대화 할 필요가 있다. 이러한 경우 제조에 필요한 설비는 동일한 작업을 처리할 수 있는 기계가 하나 이상 있고, 설비 능력이나 효율, 용량 등이 상이한 기계로 구성된 경우를 이종 병렬설비라 한다. 이종 병렬설비 제조방식은 제조 기업에서 흔히 발견될 수 있는 설비유형이다. 이종 병렬설비는 한 단위의 작업이 끝나면 동일하지 않은 다음 작업을 수행하려면 준비과정이 필요하다. 준비 작업을 위한 작업 준비시간(setup time)이 소요되는데 작업 준비시간이 적게 소요되도록 이종 병렬설비의 효과적인 관리와 효율적인 일정 계획이 필요하다.

본 연구의 사례기업 T사의 면취기 공정(beveling process)은 출하된 강관(steel pipes)을 연결시킬 때 연결부위의 접합을 위해 이음새 부분을 나선형 곡선으로 양 끝을 그라인딩으로 갈아내는 작업이다. 강관류의 형상에 따라 각 제품은 전용설비에 할당되고, 철의 부피와 두께에 따

라 설비가 처리할 수 있는 능력을 벗어나는 제품은 범용 설비에서 작업을 처리하게 된다. 하지만 전·범용 설비는 한 종류의 제품을 처리한 후 다른 형상의 작업을 위해 금형을 교체하는 작업 준비시간이 필요하다. 각 설비에 어떠한 순서로 작업이 할당하는가에 따라 완제품 납기일에도 큰 영향을 미치므로 주문의 납기일을 만족시킬 수 있는 최적화된 일정 수립의 필요성을 확인하였다.

여기서 작업 준비시간은 한 제품을 생산한 후 다른 종류의 제품을 생산하기 위해 발생하는 준비시간으로 라인 교체나 성형틀 교체 등의 setup을 말하는 것으로 준비작업을 하는 시간에는 제품을 생산할 수 없기 때문에 최적의 로트 크기를 정하거나 최적의 작업순서를 정하여 작업교체횟수를 줄여야 한다.

이러한 병렬설비 일정계획의 기존 연구는 오랫동안 많은 연구의 대상이 되었고, 목적함수에 따라 설비에 작업(제품)을 할당하여 총 작업완료시간(makespan) 최소화 문제, 작업(제품) 납기지연 수 최소화 문제, 납기 조기달성과 납기지연(earliness/tardiness) 최소화 문제 등으로 분류된다[1, 4].

Piersma and Van Dijk[22]의 연구에서는 병렬설비의 일정계획 문제를 세 가지 유형으로 분류하고 있다. 단, 아래에서 여기서 p_{ij} 는 작업 j 를 설비 i 에서 처리하는 시간이고, p_j 는 작업 j 의 작업시간을 나타낸다.

- (1) 동일한 성능(처리속도)의 설비(identical machines) :
 $p_{ij} = p_j$ for all i and j ; i 는 설비를 나타내는 인덱스,
 j 는 작업을 나타내는 인덱스
- (2) 일정한 비율의 성능차이를 지닌 설비(uniform machines) :
 $p_{ij} = p_j/v_i$ for all i and j ; v_i 는 기계 i 의 처리속도
- (3) 서로 다른 성능차이를 지닌 설비(unrelated machines) :
 p_{ij} for all i and j .

작업을 처리할 수 있는 설비가 여러 대 있을 경우를 병렬설비를 지닌 공정이라고 한다. 병렬설비 공정에서 설비의 능력이 동일하거나 그 차이의 정도에 따라 각각의 유형으로 분류하고 있다.

일반적인 단일설비의 일정계획 문제에서 작업의 수가 n 개가 있다면 이를 처리하기 위한 작업일정(스케줄) 갯수는 $n!$ 개이며, 작업을 처리할 수 있는 설비의 수가 m 개가 된다면 작업일정의 수는 $(n!)m$ 개가 된다. 그러므로 병렬설비의 일정계획 문제에서 동일한 성능을 가정한 경우라 하더라도 문제의 복잡도는 NP-hard로 알려져 있다[23, 15, 9].

납기지연시간 총 합을 최소화 문제는 Lenstra et al.[16]에 의해 NP-hard로 알려져 있고, Du et al.[17]의 연구에서와 같이 단일 설비의 문제라 하더라도 NP-hard 영역의 복잡도를 가지는 것으로 알려져 있고, 전통적인 최적화 기법은 문제의 복잡도가 높아질수록 최적의 대안을 찾는

것이 어렵다[19]. 이러한 문제를 해결하기 위하여 다양한 휴리스틱 기법이 제안되었고, 여기에는 시물레이티드 어닐링(SA : simulated annealing), 타부서치(TS : tabu search), 분지한계(B&B : branch and bound), 신경망 네트워크(NN : neural network), 유전 알고리즘(GA : genetic algorithm) 등이 있다[2].

이러한 다양한 방법을 활용한 연구들을 살펴보자면 Lee and Pinedo[14]는 WSPT(Weighted Shortest Processing Time) 규칙과 LSR(Least Slack Remaining)규칙을 결합한 ATS(Apparent Tardiness Cost) 알고리즘에서 작업 준비시간을 추가한 ATCS(Apparent Tardiness Cost with Setups)를 고안하여 순서 의존적인 작업 준비시간이 요구되는 단일 및 이중 병렬설비의 총 가중 납기지연 시간을 최소화하는 휴리스틱 기법을 제안하였다. 이는 ATCS 규칙의 적용을 위해 각 작업들의 총 처리시간, 결정모수, 통계적인 특성을 고려하였으며, 작업을 기계로 할당하여 시물레이티드 어닐링 방법을 사용하여 보다 개선된 해를 구하였다.

Kanet[12]은 모든 작업이 공통의 납기일자가 작업완료 시간보다 클 경우의 단일설비에서 MAD(Mean Absolute Deviation)를 최소화하는 일정계획 문제 대하여 연구하고, 최적 알고리즘을 제안하였다. Hall[10]은 Kanet의 연구를 확장하여 한 문제에 대한 여러 해가 존재함을 밝히고, 해를 구하는 알고리즘을 제시하였다. 또한 Lawler[13]은 pseudo-Polynomial 알고리즘을 이용하여 전체 작업의 납기지연을 줄이는 방법을 제시하였고, Monma and Potts[20]은 Batch Setup 시간을 갖는 두 대의 동일 설비에 대한 문제에서 총 작업시간, 최대 지연시간(maximum lateness), 지연작업의 수(number of late jobs), 전체 가중 완료시간(total weighted time)의 pseudo-Polynomial 알고리즘을 개발하여 알고리즘을 m 개의 설비에 대한 문제로 확장하였다.

Wilkerson and Irwin[25]은 납기지연 시간을 최소화하기 위하여 단일설비에서 EDD(Earliest Due-Date)규칙을 확장하는 알고리즘을 제안하였고, Dobson and Nabinadom[7]은 각각의 작업들이 서로 다른 작업 완료시간의 가중치를 요구하는 작업군을 갖는 작업들을 처리하는 방법을 연구하였다. Schutten and Leussink[24]은 출하날짜(release dates)와 납기일, 작업군별 작업 준비시간을 고려하여 최대 지연시간을 최소화하는 분지한계 알고리즘을 제안하였다.

Leung and Young[17]은 동일하지 않은 작업 처리시간을 가지는 단일설비에서 작업의 총 납기지연 최소화를 위한 알고리즘을 제안하였고, Park[21]은 작업의 처리시간이 모든 기계에서 동일하고, 납기 및 납기 가중치가 주어졌을 때 납기지연 가중 합을 최소화를 위해 유전 알고리즘을 제안 및 MDD(Modified Due Date) 알고리즘과 비교 분석하였고, Jeon and Park[11]는 단일기계 문제에서 총 납기지연시간을 최소화하기 위해서 두 개 작업씩 쌍으로 비교

하는 JPC(Job Pair Comparison)할당규칙을 제시하고, 적용에 있어 이행성을 증명하였고, Bilge[3]는 일정한 비율로 성능의 차이를 보이는 병렬기계에서 순서의존적 작업 준비시간을 가지는 작업의 총 납기 지연시간 최소화를 위해 기존 연구의 해의 검색 절차를 개선하는 Tabu search 알고리즘을 제안하였고, Logendran[18]은 타부서치 알고리즘을 활용하여 이중 병렬설비의 가중 납기지연시간의 최소화를 위한 알고리즘을 제안하였다.

본 연구에서는 현실적인 문제의 상황을 해결하기 위하여 설비의 능력이 각각 다르며, 설비는 전용과 범용 설비로 구분됨을 가정하였다. 또한 작업 종류에 따라 지정된 전용설비에서 작업되고, 범용설비는 모든 제품을 작업할 수 있다. 각 작업이 완료되면 작업교체나 예방정비를 위한 준비시간을 고려한 납기지연시간 총 합 최소화 일정계획을 수립하고자 한다.

그러므로 본 연구에서는 각 작업(제품)은 각기 다른 납기일을 가지고 있다. 이러한 작업을 전·범용설비로 구성된 공정에서 처리할 경우 작업 완료시간과 납기 지연시간을 어떻게 최소화할 것인가 하는 문제를 다룬다. 위의 상황에서의 문제를 해결하기 위하여 유전 알고리즘을 제안하고 EDD(Earliest Due-Date), LPT(Longest Processing Time), SPT(Shortest Processing Time) 기법과 비교하여 알고리즘의 성능을 평가하고자 한다.

본 논문은 5장으로 구성되어 있으며 각 장의 구체적인 내용은 다음과 같다. 제 2장에서는 연구의 대상 문제를 정의하고, 제 3장에서는 문제의 분석을 통하여 유전 알고리즘 적용시켜 보았다. 제 4장에서는 제안된 유전 알고리즘의 성능을 평가하였으며, 제 5장에서는 추후 연구 및 결론에 대하여 논하였다.

2. 문제의 정의

본 연구에서의 납기지연 총 합의 최소화 문제는 설비별로 작업 준비시간과 작업시간이 상이하고 각기 다른 납

기일 d_j 가 존재하는 n 개의 작업을 전용설비, 범용설비에 따라 다른 성능을 가지는 m 대의 설비에 납기지연시간의 총 합을 어떻게 최소화할 것인가 하는 문제이다. 즉, 작업은 전용설비의 처리시간과 작업 준비시간이 있고, 범용설비에서도 처리시간과 작업 준비시간이 있다. 또한 작업 j 는 상이한 납기일 d_j 가 존재하며 작업 완료시간을 C_j 라 하면 이 문제에서의 목적함수는 납기지연시간 총 합(Total Tardiness : $\sum_{j=1}^n \max\{0, C_j - d_j\}$)을 최소화 하고자 하는 것이다. 문제에서 고려하는 가정은 다음과 같다.

- (1) 투입된 작업은 완료 전까지 중단되거나, 투입될 수 없다.
- (2) 작업은 하나의 설비에만 투입되어야 한다.
- (3) 제품의 종류에 따라 전용 설비에서만 작업이 가능하다. 단, 범용설비는 모든 작업이 가능하다.
- (4) 전용설비에 투입 가능하지 못한 작업은 범용에서만 가능하다.
- (5) 작업별 주문량은 서로 다른 설비에 분할되어 처리될 수 없다.

j : 작업 index

i : 기계 index

a : 주문량 index

d : 납기일 index

C_j : 작업 j 의 작업완료시간

a_j : 작업 j 의 주문량

d_j : 작업 j 의 납기일

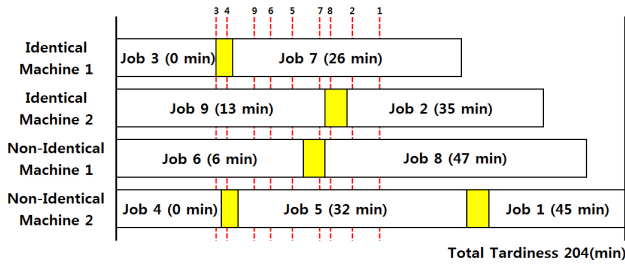
p_{ij} : 작업 j 가 i 번째 설비에서 작업할 때 처리시간

s_{ij} : 작업 j 가 i 번째 설비에서 작업할 때 준비시간

본 연구의 문제 상황에 관한 이해를 돕기 위해 다음의 예제 문제를 활용하여 설명하고자 한다. <Table 1>에서와 같이 사례기업 T사의 면취기 공정의 작업정보가 있고, 각 작업은 납기일이 정해져 있으며, 설비의 처리시간이 각각 다른 전용·범용설비가 2대, 2대 있다고 가정한다.

<Table 1> Information of Beveling Process

		Job 1	Job 2	Job 3	Job 4	Job 5	Job 6	Job 7	Job 8	Job 9
Identical Machine	Processing Time(min)	X	6	4	5	X	2	6	X	5
	Set-up Time(min)	X	4	6	3	X	3	3	X	2
Non-Identical Machine 1	Processing Time(min)	6	5	5	6	9	3	7	6	5
	Set-up Time(min)	4	3	5	4	3	2	4	3	5
Non-Identical Machine 2	Processing Time(min)	5	6	8	3	7	5	4	5	4
	Set-up Time(min)	4	3	5	4	3	2	4	3	5
Tardiness(min)		50	45	20	22	34	30	39	41	27
Orders		5	6	5	7	6	12	7	8	8
Type		1	2	1	1	2	1	1	2	2



<Figure 1> Example of Beveling Process

<Figure 1>을 통하여 전용(Identical)설비 2대, 범용(Non-Identical)설비 2대인 이중 병렬설비의 납기지연시간 총합 최소화 목적을 위한 일정계획 문제의 예제를 참조할 수 있다. 그림에서 붉은색 점선으로 표시된 것은 각 작업별 납기일을 표시한 것이며, 이를 초과하면 납기지연이 발생하는 것이다. “작업 $j(T_j)$ ”는 작업 j 가 전용설비에 처리되는 납기지연시간으로 $T_j = \max\{0, C_j - d_j\}$ 이다. C_j 는 설비에 작업이 투입되어 처리된 후 종료되는 시점을 의미하고, 첫 번째 작업 j 의 작업 종료시점인 p_{ij} 의 처리시간과 주문량 a_j 의 곱으로 계산할 수 있다. 예를 들어 전용설비 1의 경우 “작업 3(0분)”의 의미는 작업 3의 납기지연시간이 0분을 의미한다. d_3 는 작업 3의 납기일로 T_3 은 $\max\{0, (p_{13} \times a_3) - d_3\}$ 이며 작업 완료시점인 20분에서 납기일 20분을 뺀 0분이 작업 3의 납기 지연시간이 된다. 작업 3과 작업 7의 사이에 표시된 노란색은 작업 7의 작업 준비시간 $s_{37} = 3$ 를 의미하며, $T_7 = \{0, C_7 - d_7\}$ 은 작업 j_7 의 작업 종료시점 C_7 은 $(p_{33} \times a_3) + s_{37} + (p_{37} \times a_7)$ 이 된다. 그러므로 전용설비 1의 납기지연시간의 합은 26분이 되며, 총 납기지연시간($\sum_{j=1}^n T_j$)은 전용·범용설비의 납기지연시간의 모두 더한 합으로 $\sum_{j=1}^n \max\{0, C_j - d_j\} = 204$ 가 된다.

3. 유전알고리즘의 적용

본 연구에서 제안하는 유전 알고리즘 절차는 Choi et al.[5, 6]의 알고리즘을 문제의 상황에 따라 수정 및 적용하였으며, 염색체의 해석과 전용·범용 설비의 작업할당을 위하여 설비할당절차를 제안하였다.

3.1 개체의 표현

유전 알고리즘의 첫 단계는 문제의 잠재해를 어떻게 표현하는지 결정하는 것으로 이때, 염색체의 표현방법에 따라 알고리즘의 성능이 달라진다. 본 연구에서는 <Figure 2>와 같이 각 염색체 비트에 작업을 표현하도록 하였다.

i_1	i_2	i_3	i_4	i_5	...	i_n
-------	-------	-------	-------	-------	-----	-------

<Figure 2> Chromosome Structure

<Figure 3>은 염색체 구조의 예제로 작업이 7개 있으며, 작업 2가 면취기 공정에 처음으로 할당되고 작업 4는 두 번째로, 마지막으로 작업 5의 순서로 할당됨을 의미한다.

2	4	7	6	1	3	5
---	---	---	---	---	---	---

<Figure 3> Chromosome Example

3.2 초기 모집단 및 적합도 평가

염색체 개체들로 구성된 모집단을 운영하기 위한 본 연구에서는 100개의 염색체를 임의로 생성하여 구성하였다. 또한 적합도(fitness)는 개체의 생존능력을 의미하고, 모집단에서 다음 세대로 생존할 개체를 선별하기 위하여 목적함수($\sum_{j=1}^n T_j$)를 사용하였다.

3.3 선별기법 및 교차연산

선별기법은 개체의 적합도에 기초하여 모집단에서 다음 세대로 생존 또는 유전에 필요한 개체를 선별하는 과정으로 모집단의 다양성과 좋은 해를 효율적으로 사용하기 위하여 큰 적합도를 가지는 개체의 생존 가능성을 높이는 확률적 과정인 roulette wheel selection 기법을 사용하였다. 이는 모든 개체의 적합도를 계산하여 우수한 개체를 그렇지 못한 개체에 비하여 비율적으로 넓은 룰렛의 범위가 결정되도록 하여 개체의 생존가능성을 높였다.

교차연산에는 선별기법을 통하여 선택된 두 개의 개체에서 공통적으로 두 개의 절단점을 임의로 선택하여 교차하는 것 이점교차를 사용하였다. 이는 두 절단점을 기준으로 앞, 중간, 뒤의 세 부분으로 구분하였다. 앞과 뒤 부분의 인자는 다른 자손에게 교차되어 상속되고, 중간 부분의 인자는 자손 개체에 보존되어 상속되도록 하였다.

Parents 1	1	2	3	4	5	6
Parents 2	6	5	4	3	2	1
	Cutting point →				Cutting point ←	
Children 1	2	1	3	4	6	5
Children 2	5	6	4	3	1	2

<Figure 4> Crossover Example

<Figure 4>는 교차연산의 예제로 두 개의 절단점을 선정하고 부모의 유전인자를 자손에게 상속받도록 하였다. 교차연산을 수행하다 자손의 염색체에 동일한 인자가 중복적으로 출현하는 경우가 발생할 수 있다. 이 경우 3.4의 교정연산을 통하여 실행 불가능한 염색체를 유효하게 만들도록 하였다. 자손 염색체의 적합도 계산을 통하여 모집단의 가장 열등한 염색체와 비교하여 우월한 자손은 다음 세대의 진화에 참여시키고, 그렇지 못한 개체는 제외시키도록 하였다.

3.4 돌연변이 연산 및 교정연산

돌연변이 연산은 하나의 염색체를 부분적으로 조작해서 새로운 염색체를 만드는 연산으로 교차연산을 통해 생성할 수 없는 염색체를 인위적으로 만들어 내는 연산이다. 선택된 개체의 스트링 수 이내로 난수를 발생시켜, 난수에 해당되는 염색체 값을 인위적으로 변화를 주어 새로운 개체를 만들어 내도록 하였다.

초기 모집단 구성을 위한 임의생성, 교차, 돌연변이 연산에서 작업이 염색체에 중복 표현되는 경우가 발생할 수 있다. 이러한 경우는 염색체의 일부가 연구의 기본적인 가정을 만족하지 못하는 것으로 모든 작업이 염색체에 출현할 수 있도록 염색체로 변형하였다.

[교정연산 절차]

- step 1 : 모든 작업이 할당되었으면 종료.
- step 2 : 염색체에 표현되지 않은 작업 j 를 선택.
- step 3 : 중복 표현된 작업 j' 를 선택.
- step 4 : 작업 j' 를 삭제하고 그 자리에 작업 j 를 할당.
- step 5 : 모든 작업의 할당이 완료되면 종료, 그렇지 않으면 step 1로 이동.

위의 '교정연산 절차'는 염색체에 모든 작업이 출현할 수 있도록 작업을 강제로 할당하는 방법이다.

3.5 설비할당 절차

염색체는 작업들이 투입되는 순서 정보에 해당되는 것으로 염색체 해석을 통하여 전용·범용 설비에 적절히 작업 할당을 결정해야 한다. 본 연구에서는 다음과 같은 작업할당 절차를 활용하여 각 설비에 작업을 할당하였다.

염색체의 작업순서 정보를 활용하여 전용설비에 투입이 불가능한 작업을 우선적으로 범용설비에 할당하는 규칙으로 범용설비를 효율적으로 사용하도록 하였다. 이는 다음과 같은 세부절차를 따른다.

[설비할당 절차]

- step 1 : 전용설비에서 작업시간이 0인 작업을 선택한다.
- step 2 : 염색체에 표현된 순서대로 나열하고 선택한다.
- step 3 : 범용설비 작업완료시간이 가장 짧은 설비에 작업을 할당한다.
- step 4 : 모든 작업이 할당되었으면 step 5로 이동, 그렇지 않으면 step 2로 이동한다.
- step 5 : 전용설비의 작업 처리시간이 0인 염색체를 삭제한다.
- step 6 : 전용설비의 처리시간이 범용설비의 처리시간보다 작은 경우 염색체의 순서에 따라 작업을 전용설비에 할당한다. 그렇지 않다면 step 7로 이동한다.
- step 7 : 범용설비 작업완료시간이 가장 짧은 설비에 작업을 할당한다.
- step 8 : 모든 작업의 할당이 완료되면 종료한다, 그렇지 않으면 step 6으로 이동한다.

4. 성능 평가

본 장에서는 납기 지연시간 총 합을 최소화 문제에 대하여 제 3장에서 제안된 유전 알고리즘의 성능을 평가하고 분석한다. 본 연구에서는 복잡한 납기 지연문제에서 단시간에 효과적인 해를 도출하는 휴리스틱 방법인 EDD, SPT, LPT 규칙과 비교분석 하고자 한다.

유전 알고리즘은 1000세대까지 유전시켰으며, 각 문제당 20가지의 문제를 임의 생성하였고, 10회에 걸쳐 유전 알고리즘으로 문제를 실험하였다. <Table 2>는 작업과 설비의 유형, 각 설비 수의 변화에 따라 유전 알고리즘이 얼마나 좋은 성능을 보이는지에 대한 성능평가이다. 이 실험에서는 작업의 수가 30, 40, 50, 60, 70, 80개이고, 설비의 수는 전용설비 3~6대, 범용설비 3~6대이며, 작업 j 의 설비별 작업 처리시간을 20~50, 작업 준비시간을 20~50, 주문량을 20~50사이에서 임의 생성하였다. 또한 작업들의 납기일 d_j 는 $U(T_l \times R_L, T_u \times R_L)$ 로부터 발생시켰다. 여기서 R_L 은 작업완료시간(C_{max})의 이론적 하한값을 의미하며, (T_l, T_u) 값의 범위는 (0.4, 0.8)로 설정하였다.

여기서 Gap1(%)은 유전 알고리즘의 연산을 통해 제시된 가능해와 EDD와 오차를 의미하며, Gap 2(%)은 유전 알고리즘의 가능해와 LPT와 오차를 의미하며, Gap 3(%)은 유전 알고리즘의 가능해와 SPT와 오차를 의미한다. Average Gap(%)은 Gap(%)의 오차 평균값을 의미한다.

Gap 1~3(%)은 다음과 같은 식에 의해 구할 수 있다.

$$Gap\ 1(\%) = \{(EDD - GA) / EDD\} \times 100$$

$$Gap\ 2(\%) = \{(LPT - GA) / LPT\} \times 100$$

$$Gap\ 3(\%) = \{(SPT - GA) / SPT\} \times 100$$

<Table 2>는 작업과 설비의 유형, 각 설비 수의 변화에 따라 유전 알고리즘이 얼마나 좋은 성능을 보이는지에 대한 실험이다. 작업의 수는 30~70개, 전용·범용설비를 각 3~5대에 대하여 실험하였다. Average Gap 1(%)에서 작업의 수와 설비의 수가 증가할수록 즉, 문제의 복잡도가 증가됨에 따라서 EDD와의 차이가 최소 23%까지 줄어들었다. 작업의 수에 비해 설비의 수가 작은 경우는 EDD보다 최대 53% 개선된 해를 제공해 주었다.

Average Gap 2(%)는 유전 알고리즘과 LPT 해와의 차이로 설비의 수가 작은 경우 최대 74% 유전 알고리즘이 우수한 해를 제공해 주었으며, 문제의 복잡도가 증가될수록 최소 34% 유전 알고리즘의 가능 해와의 차이가 줄어드는 것을 확인할 수 있었다.

Average Gap 3(%)는 유전 알고리즘과 SPT 해와의 차

<Table 2> Performance Test of Genetic Algorithm 1

Job	Machines		Average Gap 1 (%)	Average Gap 2 (%)	Average Gap 3 (%)	Time
	Identical	Non-Identical				
30	3	3	53.550	60.367	56.783	0.603
		4	55.986	63.746	55.511	0.624
		5	53.895	61.825	57.446	0.688
	4	3	53.624	64.306	56.105	0.593
		4	49.561	52.806	52.450	0.599
		5	48.426	54.931	50.678	0.603
	5	3	51.550	65.707	60.813	0.591
		4	50.073	58.065	53.611	0.597
		5	43.709	42.389	42.188	0.601
50	3	3	51.289	74.638	72.870	1.473
		4	41.123	69.887	63.237	1.506
		5	41.993	67.237	61.772	1.519
	4	3	43.246	55.625	51.863	1.470
		4	43.709	50.120	43.426	1.540
		5	39.340	50.676	41.598	1.493
	5	3	41.919	54.144	49.802	1.485
		4	41.870	50.322	45.734	1.294
		5	37.637	43.138	37.874	1.308
70	3	3	35.110	73.254	68.422	2.072
		4	34.192	62.993	52.866	2.602
		5	34.193	66.440	61.862	2.612
	4	3	37.745	47.396	39.212	2.579
		4	23.810	43.299	32.825	2.602
		5	35.541	41.661	35.690	2.612
	5	3	25.853	49.174	40.790	2.579
		4	23.036	34.660	27.502	2.596
		5	35.162	38.921	34.251	2.601

이로 설비의 수가 작은 경우 최대 72% 유전 알고리즘이 우수한 해를 제공해 주었으며, 문제의 복잡도가 증가될수록 최소 27% 유전 알고리즘의 가능 해와의 차이가 줄어드는 것을 확인할 수 있었다.

Average Gap 1(%)은 납기일을 기준으로 한 휴리스틱 방법으로 LPT, SPT 방법보다 비교적 개선된 해를 제공하였고, 공통적으로 Average Gap2(%)과 Average Gap 3(%)은 납기일 최소화 관점의 휴리스틱 방법이 아니기 때문에 유전 알고리즘의 해와 많은 차이를 보였으며, EDD, LPT, SPT와 비교하여 유전 알고리즘은 우수한 가능해를 제공함을 확인할 수 있었다.

<Table 3>, <Table 4>, <Table 5>은 작업의 수를 30, 40, 50, 60, 70, 80개, 설비의 대수는 전용설비 4대, 범용설비 5대인 경우를 대상으로 납기일 d_j 는 $U(T_l \times R_L, T_u \times R_L)$ 로부터 발생시켰다. (T_l, T_u)는 (0.4, 0.8) 사이에서 임의 생성하였으며, 전용설비에 투입될 수 없는 작업의 생성을 위한 비율은 전체 작업의 20%로 가정하고 임의생성 하였다. 작업 준비시간, 주문량, 작업 처리시간 범위의 변동에 관한 실험으로 Gap(%)은 <Table 2>에서 실험한 휴리스틱 방법 중 비교적 좋은 가능해를 제공하는 EDD와의 비교하였다.

<Table 3> Performance Test of Genetic Algorithm 2 (Variation of Set-up Time)

Set-up Time Range	Job	Average Gap (%)	Time	Set-up Time Range	Job	Average Gap (%)	Time
[20, 40]	30	46.026	0.548	[20, 50]	30	47.792	0.593
	40	45.761	0.798		40	47.462	0.885
	50	46.517	1.193		50	43.786	1.337
	60	32.780	1.686		60	28.775	1.940
	70	42.890	2.360		70	23.000	2.610
	80	25.170	3.222		80	26.860	3.577
[20, 60]	30	48.446	0.566	[40, 70]	30	46.201	0.540
	40	41.256	0.827		40	44.274	0.807
	50	36.790	1.163		50	39.476	1.218
	60	28.279	1.671		60	34.128	1.766
	70	33.660	2.331		70	30.249	2.376
	80	27.165	3.204		80	22.595	3.250
[20, 80]	30	43.493	0.573	[60, 90]	30	46.810	0.540
	40	36.393	0.811		40	42.672	0.788
	50	40.077	1.172		50	42.393	1.171
	60	33.874	1.686		60	32.654	1.690
	70	34.101	2.362		70	32.806	2.359
	80	31.457	3.288		80	31.687	3.223
[20, 100]	30	46.961	0.547	[80, 110]	30	53.019	0.539
	40	45.341	0.801		40	38.731	0.789
	50	37.133	1.180		50	32.229	1.162
	60	37.247	1.691		60	35.900	1.677
	70	30.085	2.364		70	28.022	2.350
	80	28.579	3.241		80	33.357	3.272

<Table 4> Performance Test of Genetic Algorithm 3
(Variation of Orders)

Orders Range	Job	Average Gap (%)	Time	Orders Range	Job	Average Gap (%)	Time
[20, 40]	30	47.874	0.541	[20, 50]	30	46.886	0.064
	40	40.719	0.791		40	40.782	0.089
	50	39.057	1.170		50	35.318	0.124
	60	34.272	1.681		60	34.857	0.171
	70	29.531	2.340		70	35.425	0.248
	80	26.838	3.221		80	27.926	0.314
[20, 60]	30	44.193	0.540	[40, 70]	30	45.385	0.062
	40	47.400	0.789		40	42.302	0.087
	50	37.015	1.170		50	35.055	0.127
	60	34.933	1.678		60	28.227	0.173
	70	31.622	2.356		70	27.822	0.240
	80	29.166	3.211		80	27.535	0.325
[20, 80]	30	48.688	0.542	[60, 90]	30	42.274	0.062
	40	39.592	0.792		40	40.303	0.090
	50	41.724	1.169		50	38.938	0.124
	60	38.552	1.685		60	36.385	0.173
	70	36.311	2.347		70	28.535	0.236
	80	31.875	3.211		80	31.048	0.344
[20, 100]	30	49.847	0.543	[80, 110]	30	41.168	0.062
	40	43.902	0.796		40	40.806	0.093
	50	43.210	1.187		50	37.090	0.127
	60	39.466	1.732		60	34.633	0.185
	70	28.546	2.409		70	30.743	0.243
	80	28.630	3.221		80	30.602	0.315

<Table 5> Performance Test of Genetic Algorithm 4
(Variation of Processing Time)

Processing Time Range	Job	Average Gap (%)	Time	Processing Time Range	Job	Average Gap (%)	Time
[20, 40]	30	50.250	0.541	[20, 50]	30	50.798	0.064
	40	36.179	0.793		40	40.161	0.089
	50	37.187	1.173		50	33.785	0.125
	60	33.676	1.685		60	36.824	0.173
	70	37.667	2.382		70	30.956	0.236
	80	22.474	3.216		80	27.657	0.338
[20, 60]	30	48.317	0.540	[40, 70]	30	45.923	0.067
	40	44.132	0.807		40	43.686	0.089
	50	28.461	1.218		50	35.675	0.127
	60	40.083	1.766		60	33.640	0.175
	70	24.506	2.377		70	34.890	0.242
	80	24.547	3.250		80	23.383	0.320
[20, 80]	30	45.885	0.540	[60, 90]	30	49.821	0.063
	40	39.597	0.788		40	37.846	0.088
	50	35.636	1.171		50	41.266	0.125
	60	38.010	1.690		60	33.402	0.172
	70	35.239	2.359		70	28.749	0.237
	80	29.289	3.223		80	28.634	0.317
[20, 100]	30	43.005	0.538	[80, 110]	30	47.401	0.063
	40	44.448	0.789		40	35.589	0.090
	50	39.711	1.162		50	36.906	0.125
	60	39.645	1.677		60	37.023	0.173
	70	31.510	2.349		70	29.331	0.237
	80	31.322	3.273		80	25.465	0.312

<Table 3>에서는 최소 작업 준비시간을 20으로 고정할 실험에서는 작업 준비시간의 범위가 증가되더라도 알고리즘의 성능은 최소 25%에서 최대 48%, 작업 준비시간의 범위를 30으로 고정시킨 실험에서는 최소 22%에서 최대 53%로 EDD와 비교하여 유전 알고리즘의 해가 우수한 가능해를 제공해 주었다. <Table 4>, <Table 5>도 공통적으로 작업의 수가 증가하여 문제가 복잡할수록 EDD와의 차이가 줄어드는 것을 확인할 수 있다. 이는 EDD 방법이 복잡한 문제에서도 수렴가능한 해를 찾아주지만 최적해나 최적해와 근사해를 제공하는 메타휴리스틱 기법의 유전 알고리즘 보다 좋은 해를 제공해 주지는 못함을 확인할 수 있었다.

<Table 6>는 작업 j 의 납기일 범위의 변동에 따른 실험으로 납기일 범위가 (0.4, 08)인 경우 (0.6, 1.0)인 경우보다 더 개선된 가능해를 제공해주고 있다. 이는 각 작업의 납기일의 초과한 기간이 더 긴 (0.6, 1.0)보다 초과 기간이 짧은 경우 더욱 안정적인 가능해를 제공해주는 것으로 해석할 수 있다.

<Table 6> Performance Test of Genetic Algorithm 5
(Variation of Tardiness)

Tardiness	Job	Average Gap(%)	Time
[0.4, 0.8]	30	48.426	0.603
	40	45.948	0.949
	50	39.34	1.493
	60	41.952	1.949
	70	35.541	2.612
	80	26.679	3.559
[0.6, 1.0]	30	54.319	0.567
	40	45.141	0.928
	50	46.079	1.365
	60	40.326	1.974
	70	38.059	2.634
	80	38.752	3.509

<Table 7>은 유전 알고리즘의 전용설비에서 작업 불가능한 작업의 비율에 대한 실험으로 작업의 수가 증가할수록 EDD와의 차이가 줄어들며, 전용설비 불가 작업의 비율이 0.5에서 가장 좋은 가능해를 제공해 주었다. 이는 전용설비에서 처리 불가능한 작업이 전체 작업 50%인 경우로써 80%인 경우보다 전·범용 설비에 작업을 균등하게 할당하는 것으로 판단된다. 즉, 작업을 비대칭적으로 할당하는 80%인 경우보다 전용설비와 범용설비를 효율적으로 배분하여 작업을 할당하기 때문에 처리시간의 차이가 줄어드는 것으로 판단된다.

<Table 7> Performance Test of Genetic Algorithm 6 (Impossible Processing Job Ratio in Identical Machines)

Impossible Ratio	Job	Average Gap(%)	Time
0.5	30	52.439	0.603
	40	48.519	0.880
	50	42.300	1.297
	60	43.018	1.880
	70	41.554	2.625
	80	41.358	3.574
0.6	30	54.907	0.642
	40	53.545	0.965
	50	40.496	1.432
	60	41.755	2.262
	70	31.886	2.683
	80	37.276	3.977
0.7	30	48.888	0.665
	40	46.808	0.911
	50	33.785	1.339
	60	38.756	1.906
	70	32.729	2.659
	80	22.714	3.638
0.8	30	48.426	0.603
	40	45.948	0.949
	50	39.340	1.493
	60	32.035	2.022
	70	35.541	2.612
	80	26.679	3.559

<Table 8>는 작업 j 의 규모를 비현실적으로 크게 한 경우의 실험으로 작업의 수가 증가함에 따라 알고리즘의 성능은 EDD와 최대 33%에서 최소 26% 차이를 보인다. 본 연구에서 제안하는 유전 알고리즘이 규모가 큰 문제에 대해서도 안정적인 가능해를 제공해 주는 것으로 판단되며, Average Gap 2(%) 최소 57%에서 최대 75%, Average Gap 3(%) 최소 23%에서 최대 50%로 LPT, SPT보다 개선된 가능해

를 제공하는 것으로 확인되었다. 하지만 작업의 수를 증가시킬수록 실험시간도 증가됨을 확인할 수 있는데, 유전 알고리즘의 염색체 비트가 작업의 수를 의미하고, 이는 유전과정에서 염색체 해석과 유전연산에 소요되는 시간이 길어지는 것으로 해석할 수 있다.

<Table 8> Performance Test of Genetic Algorithm 6 (Number of Large-scale Jobs)

Job	Average Gap 1(%)	Average Gap 2(%)	Average Gap 3(%)	Time
100	33.476	75.736	50.460	6.764
200	25.251	71.174	47.166	44.434
300	25.353	57.387	23.647	143.927
400	26.191	59.913	32.919	335.376

5. 결론

본 논문에서는 사례기업의 실제 생산시스템을 반영한 연구로써 전용설비와 범용설비가 혼재되어 제품을 제조하는 먼취기 공정상황으로 이종병렬설비의 납기지연시간 총합의 최소화 문제를 유전 알고리즘을 활용하여 계획하였다. 유전 알고리즘의 성능을 향상시키고자 염색체의 작업순서를 각각의 설비에 할당하는 ‘설비할당 절차’와 해의 가정을 만족시키는 ‘교정절차’를 제시하였다. 알고리즘에서 제안하는 가능해의 검증에 위해 EDD, SPT, LPT 휴리스틱 알고리즘과 비교하였고, 가능해의 우수성을 분석하였다.

추후 유전 알고리즘의 성능에 대한 다양한 접근과 평가를 위해 이론적 하한값, 메타 휴리스틱 알고리즘의 개발이 필요하며, 다단계 공정상황의 이종병렬설비 대상으로의 연구 확장, 실제 기업에서 발생 가능한 복합적인 상황들을 고려하는 연구가 필요하다.

References

- [1] Baker, K.R. and Baker, K.R., Introduction to sequencing and scheduling, New York : Wiley; 1974. Vol. 31.
- [2] Balin, S., Non-identical parallel machine scheduling using genetic algorithm. *Expert Systems with Applications*, 2011, Vol. 38, No. 6, pp. 6814-6821.
- [3] Bilge, U., Kirac, F., Kurtulan, M., and Pekgun, P., A tabu search algorithm for parallel machine total tardiness problem. *Computers and Operations Research*, 2004, Vol. 31, No. 3, pp. 397-414.
- [4] Cheng, T.C.E. and Sin, C.C.S., A state-of-the-art review

- of parallel-machine scheduling research. *European Journal of Operational Research*, 1990, Vol. 47, No. 3, pp. 271-292.
- [5] Choi, Y.J., Song, H.S., and Lee, I.S., A Genetic Algorithm for Minimizing Completion Time with Non-identical Parallel Machines. *Korean management science review*, 2013, Vol. 30, No. 3, pp. 81-97.
- [6] Choi, Y.J., Yu, J.D., Song, H.S., and Lee, I.S., The Batch Scheduling for Heat-Treatment Process with Parallel Machines Environments. *Journal of the Korean Production and Operations Management Society*, 2011, Vol. 22, No. 4, pp. 451-467.
- [7] Dobson, G. and Nabinadom, R.S., *The batch loading and scheduling problem. Research Repot, Simon Graduate School of Business Administration, University of Rochester*, 1992.
- [8] Du, J. and Leung, J.Y.T., Minimizing total tardiness on one machine is NP-hard. *Mathematics of operations research*, 1990, Vol. 15, No. 3, pp. 483-495.
- [9] Garey, M.R. and Johnson, D.S., *Computer and intractability : A guide to the theory of NP-completeness*, San Francisco : W.H. Freeman, 1979.
- [10] Hall, N.G., Single-and multiple-processor models for minimizing completion time variance. *Naval Research Logistics Quarterly*, 1986, Vol. 33, No. 1, pp. 49-54.
- [11] Jeon, T.J. and Park, S.H., A Proof of Transitivity of Job-Pair Comparison Rule to Minimize Total Tardiness in a Single Machine. *Korean Management Science Review*, 2001, Vol. 18, No. 1, pp. 147-153.
- [12] Kanet, J.J., Minimizing the average deviation of job completion times about a common due date. *Naval Research Logistics Quarterly*, 1981, Vol. 28, No. 4, pp. 643-651.
- [13] Lawler, E.L., A pseudo-polynomial algorithm for sequencing jobs to minimize total tardiness. *Annals of Discrete Mathematics*, 1997, Vol. 1, pp. 331-342.
- [14] Lee, Y.H. and Pinedo, M., Scheduling jobs on parallel machines with sequence-dependent setup times. *European Journal of Operational Research*, 1997, Vol. 100, No. 3, pp. 464-474.
- [15] Lenstra, J.K. and Rinnooy Kan, A.H.G., Complexity of scheduling under precedence constraints. *Operations Research*, 1978, Vol. 26, pp. 22-35.
- [16] Lenstra, J.K., Kan, A.R., and Brucker, P., Complexity of machine scheduling problems. *Ann. Discrete Math*, 1977, Vol. 1, pp. 343-362.
- [17] Leung, J.Y.T. and Young, G.H., Minimizing total tardiness on a single machine with precedence constraints. *ORSA Journal on Computing*, 1990, Vol. 2, No. 4, pp. 346-352.
- [18] Logendran, R., McDonell, B., and Smucker, B., Scheduling unrelated parallel machines with sequence-dependent setups. *Computers and Operations Research*, 2007, Vol. 34, No. 11, pp. 3420-3438.
- [19] Min, L. and Cheng, W., A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines. *Artificial Intelligence in Engineering*, 1999, Vol. 13, pp. 399-403.
- [20] Monma, C.L. and Potts, C.N., On the complexity of scheduling with batch setup time. *Operations Research*, 1989, Vol. 37, No. 5, pp. 798-804.
- [21] Park, M.W., A Genetic Algorithm for the Parallel-Machine Total Weighted Tardiness Problem. *Journal of the Korean Institute of Industrial Engineers*, 2000, Vol. 26, No. 2, pp. 183-192.
- [22] Piersma, N. and van Dijk W., A local search heuristic for unrelated parallel machine scheduling with efficient neighborhood search. *Mathematical and Computer Modelling*, 1996, Vol. 24, No. 9, pp. 11-19.
- [23] Rinnooy, Kan A.H.G., *Machine scheduling problems : Classification, complexity and computations*. The Hague : Martinus Nijhoff, 1976.
- [24] Schutten, J.M.J. and Leussink, R.A.M., Parallel machine scheduling with release dates, due dates and family setup times. *International Journal of Production Economics*, 1996, Vol. 46, pp. 119-125.
- [25] Wilkerson, L.J. and Irwin, J.D., An improved method for scheduling independent tasks. *AIIE Transactions*, 1971, Vol. 3, No. 3, pp. 239-245.

ORCID

Yu-Jun Choi | <http://orcid.org/0000-0002-1426-9612>