

# 작업준비시간이 없는 이중 병렬설비에서 총 소요 시간 최소화를 위한 미미틱 알고리즘 기반 일정계획에 관한 연구

이 태 희\* · 유우 식\*\*

\*인천대학교 산업경영공학과 석사 과정 · \*\*인천대학교 산업경영공학과

## A Study on Memetic Algorithm-Based Scheduling for Minimizing Makespan in Unrelated Parallel Machines without Setup Time

Tehie Lee\* · Woo-Sik Yoo\*\*

\*Incheon National University Graduate School, Industrial & Management Engineering

\*\*Incheon National University, Dept. of Industrial & Management Engineering

### Abstract

This paper is proposing a novel machine scheduling model for the unrelated parallel machine scheduling problem without setup times to minimize the total completion time, also known as “makespan”. This problem is a NP-complete problem, and to date, most approaches for real-life situations are based on the operator’s experience or simple heuristics. The new model based on the Memetic Algorithm, which was proposed by P. Moscato in 1989, is a hybrid algorithm that includes genetic algorithm and local search optimization. The new model is tested on randomly generated datasets, and is compared to optimal solution, and four scheduling models; three rule-based heuristic algorithms, and a genetic algorithm based scheduling model from literature; the test results show that the new model performed better than scheduling models from literature.

**Keywords :** Scheduling, Unrelated parallel machine scheduling, Minimize makespan, Scheduling algorithm, Memetic algorithm

## 1. 서론

### 1.1 서론

본 논문은 작업준비시간이 없는 작업을 위한 이중 병렬 설비 일정계획 문제에 대하여 작업이 모두 완료되기까지의 시간인 총 소요 시간(Makespan)을 최소화하는 일정 계획을 수립하는 것을 목적으로 한다.

현대 산업에 있어 병렬기계 생산 시스템에서의 일정 계획은 생산성을 높이는 데에 핵심적인 영향을 끼치는 조합 최적화 문제다. 그러나 이 문제는 크기가 커질수록 난이도

가 크게 오르는 NP-완전 문제로 알려져 있다. 이러한 이유로 많은 현장에서는 작업자의 경험을 바탕으로, 또는 간단한 규칙 기반 휴리스틱 알고리즘을 사용하여 일정계획을 수립한다.

본 연구에서는 작업 수  $n$ , 설비 수  $m$ 의 이중 병렬설비 일정계획 문제에서 Makespan을 최소화하기 위한 기존 일정계획 방법론들을 구현하여 이들이 수립하는 일정 계획의 성능을 비교하고, 새로운 일정계획 알고리즘을 개발하고, 성능을 평가하며, 그러한 성능을 보인 원인을 파악한다.

†본 논문은 인천대학교 2021년도 자체 연구비 지원에 의하여 연구되었음.

†Corresponding Author : Woo-Sik Yoo, Industrial and management Engineering, INCHEON UNIVERSITY, E-mail: wsyoo@inu.ac.kr  
Received March 20, 2023; Revision June 13, 2023; Accepted June 22, 2023

### 1.2 기존 연구와의 비교

이중 병렬설비 일정계획 문제는 오랜 시간 동안 많이 연구되었다. 대표적으로 E. Horowitz와 S. Sahni(1976) [12], O.H. Ibarra와 C.E. Kim(1977) [1], A.A.E.A. Elhafiz(2017) [2, 3]에서는 해당 문제 상황에서  $C_{max}$  을 최소화하는 일정계획을 수립하기 위한 규칙 기반 휴리스틱 알고리즘을 제시하였다. N. Piersma와 W. Van Dijk (1996) [13]에서는 해당 문제 상황을 위해 기존 지역 탐색 알고리즘을 개선하여 제시하였으며, B. Srivastava (1998) [14]에서는 타부서치 알고리즘을 활용한 일정 계획 방법론을 제시하였고, M. Ghirardi와 C.N. Potts (2005) [15]에서는 Recovering Beam Search 방법에 기반한 휴리스틱 알고리즘을 제시하였다.

R. Tavakkoli-moghaddam et al.(2009) [16]에서는 할당 순서에 의존적인 작업준비시간이 있는 이중 병렬설비에서 납기를 어긴 작업 수와 Makespan 두 비용을 최소화하기 위한 유전 알고리즘을 제안하였다. S. Balin(2011) [8]에서는 작업준비시간이 없는 이중 병렬설비에서 Makespan을 최소화하기 위한 유전 알고리즘을 제안하였으며, 그 후 S.Balin(2011) [9]에서 작업의 생산 시간이 퍼지 이론적 분포를 따른다는 조건을 포함시킨 동종 병렬설비에서 Makespan을 최소화하기 위한 유전 알고리즘을 제안하였다. J. Ding 외 4인(2020) [17]에서는 이중 병렬설비에서

작업을 할당할수록 설비가 열화되는 문제에 대하여 유전 알고리즘과 분할정복 방법론에 기반한 지역 탐색 알고리즘을 혼합한 미미틱 알고리즘을 제안하였다.

이들 중 O.H. Ibarra와 C.E. Kim(1977) [1]에서는 작업 및 자원마다 수행시간이 모두 다른 작업 일정계획 문제에 대하여 자원 수를 기준으로 다양한 문제 조건에서 규칙 기반 휴리스틱 알고리즘을 제안하며, 그중 자원 수  $m > 2$  인 경우에서 5개의 알고리즘을 제안하였다. 그러나 제안한 알고리즘으로 산출된 결과로써 성능을 검증 및 비교하

는 실험은 부재하다. A.A.E.A. Elhafiz(2017) [2]에서는 작업 및 자원마다 수행시간이 모두 다른 작업 일정계획 문제에 대하여 규칙 기반 휴리스틱 알고리즘으로 Average of The Completion Time Algorithm(ACTA)을 제안하였다. 성능을 검증하기 위한 10가지의 문제 조건을 가정하여 문제를 생성하였고, ACTA 알고리즘과 기존 발표된 규칙 기반 휴리스틱 일정계획 알고리즘 다섯 가지의 성능을 비교하였다. A.A.E.A. Elhafiz(2017) [3]에서는 또한 A.A.E.A. Elhafiz(2017) [2]에서 다른 문제와 동일한 문제에 대하여 규칙 기반 휴리스틱 알고리즘으로 Half the Average Scheduling Algorithm(HASA)을 제안하였다. 검증을 위한 문제 조건을 A.A.E.A. Elhafiz(2017) [2]에서 가정한 것과 다르게 가정하여 문제를 새로 생성하고, A.A.E.A. Elhafiz(2017) [2]에서 제안한 ACTA 알고리즘 한 가지의 성능과 비교하였다. S. Balin(2011) [8]은 작업준비시간이 없는 이중 병렬설비 일정계획 문제에 대하여 유전 알고리즘에 기반한 일정계획 모델을 제안하였으며, 규칙 기반 휴리스틱 알고리즘 중 LPT 알고리즘과 비교하였다. 또한 교차 연산이 규칙 기반 휴리스틱 알고리즘을 따른다는 특징을 가지고 있다.

지금까지 설명한 기존 연구 중 [1, 2, 3, 8]과 본 연구에 대한 비교표를 작성하면 <Table 1>과 같다.

## 2. 본 론

### 2.1 문제 상황

본 연구는 이기종 설비 환경에서의 일정계획 문제 중에서도 작업준비시간이 존재하지 않는 문제를 다룬다.  $R_m \parallel Makespan$  환경이라 표현되는 본 문제에서 모든 작업은 선후공정이 없는 독립적인 작업이며, 분리하여 작업할 수 없다. 또한 납기와 작업준비시간이 없으며 모든

<Table 1> Comparison of the previous and this studies

	Problem Size (Jobs × Machines)	Algorithm
<i>Ibarra &amp; Kim</i> (1977) [1]	No Example	Rule-based Heuristic
<i>A.A.E.A. Elhafiz</i> (2017) [2]	{5 levels} × 100 5000 × {5 levels}	Rule-based Heuristic
<i>A.A.E.A. Elhafiz</i> (2017) [3]	{10 levels} × 100 5000 × {10 levels}	Rule-based Heuristic
<i>S. Balin</i> (2011) [8]	9 × 4	Genetic with Rule-based Heuristic
<i>T.H. Lee &amp; W.S. Yoo</i>	{50, 100, 500} × {5, 10}	Memetic

작업은 모든 설비에 할당할 수 있다. 모든 설비는 고장이 발생하지 않으며, 예정된 정비가 없고 할당된 작업은 완료 될 때까지 중단할 수 없다. 이기중 설비이기 때문에 동일한 작업이라도 어느 설비에 할당되는가에 따라 생산시간이 달라진다. 이러한 환경에서 본 연구는 모든 작업을 완료하기까지 소요된 시간, 즉 Makespan을 최소화하는 것을 목적으로 한다.

## 2.2 수리모형

본 연구에서 다룰 문제 상황을 수리적으로 표현하기 위해 사용하는 변수들은 아래와 같다.

- $J_j$  : j번째 작업(Job)
- $M_i$  : i번째 설비(Machine)
- $Pt_{ij}$  :  $J_j$ 가  $M_i$ 에 할당되었을 때의 생산시간(Processing time)
- $C_{ij}$  :  $J_j$ 가  $M_i$ 에 할당되어 생산되기까지 총 소요된 시간(Completion Time)
- $x_{ij}$  :  $J_j$ 가  $M_i$ 에 할당되었는가의 여부
- n : 작업의 개수 | m : 설비의 개수

위 변수를 이용하여 수리모형을 수립할 시, 아래와 같은 목적함수 및 제약식으로 표현할 수 있다.

$$\text{minimize Makespan} = \max_{i=1}^m \sum_{j=1}^n C_{ij} = \max_{i=1}^m \sum_{j=1}^n x_{ij} * Pt_{ij} \tag{1}$$

$$\text{subject to } \sum_{i=1}^m x_{ij} = 1 \forall j = 1, 2, \dots, n \tag{2}$$

$$x_{ij} = 0 \text{ or } 1 \forall i = 1, 2, \dots, m \mid j = 1, 2, \dots, n \tag{3}$$

## 2.3 규칙 기반 휴리스틱 알고리즘

규칙 기반 휴리스틱 알고리즘은 일정계획 방법론에 자주 활용된다. 작업의 생산시간, 설비의 가용성 등 각 문제에 최적화된 우선순위 규칙을 기준으로 구현하기 쉬우며 계산 속도가 빨라 적은 시간을 소모하고도 일정계획을 수립할 수 있기 때문이다. 그러나 이 방법론은 최적해를 보장할 수 없다는 단점이 있다.

본 연구에서 비교군으로써 인용 및 구현한 규칙 기반 휴리스틱 알고리즘 일정계획 모델(규칙 기반 모델)은 O.H. Ibarra와 C.E. Kim(1977) [1]에서 “Algorithm D”라는 이름으로 제안하는 “Min-Min” 알고리즘, A.A.E.A.

Elhafiz(2017) [2]에서 제안하는 “ACTA” 알고리즘과 A.A.E.A. Elhafiz(2017) [3]에서 제안하는 “HASA” 알고리즘으로 총 네 가지이다. 각 알고리즘이 현 문제 상황에 적용되는 방법은 아래와 같다.

### 2.3.1 Min-Min

할당되지 않은 작업들에 대하여 설비 상관없이 최단 소요 시간을 가지는 작업을 탐색한 후, 그 작업을 최단 소요 시간을 가지는 설비에 할당한다. 모든 작업이 할당될 때까지 반복한다.

### 2.3.2 ACTA

할당되지 않은 작업들에 대하여 설비 상관없이 작업별 최단 소요 시간의 평균을 목표값으로 구한다. 목표값과 가장 가까운 최단 소요 시간을 가지는 작업을 탐색한 후, 그 작업을 목표값과 가장 가까운 소요 시간을 가지는 설비에 할당한다. 모든 작업이 할당될 때까지 반복한다.

### 2.3.3 HASA

할당되지 않은 작업들에 대하여 설비 상관없이 작업별 최단 소요 시간의 평균의 절반 값을 목표값으로 구한다. 목표값과 가장 가까운 최단 소요 시간을 가지는 작업을 탐색한 후, 그 작업을 목표값과 가장 가까운 소요 시간을 가지는 설비에 할당한다. 모든 작업이 할당될 때까지 반복한다.

## 2.4 탐색 기반 휴리스틱 알고리즘

조합 최적화 문제에 있어 탐색 기반 휴리스틱 알고리즘은 규칙 기반 휴리스틱 알고리즘에 비해 시간은 오래 걸리지만, 최적해와 더욱 가까운 값을 얻을 수 있는 방법론으로 많이 사용된다. 그 중 유전 알고리즘(G.A.)은 생물학적 진화를 모방한 확률적 탐색 기법으로 하나의 해를 이용하는 타 알고리즘과 달리 해 집단을 운용해 공간을 탐색한다[10, 11]. G.A.의 연산 과정에는 진화에 관여할 해를 선택하는 선택 연산[6], 선택된 해 집단을 기준으로 해 공간을 탐색하는 교차 연산, 확률적으로 교차 연산을 거친 해를 변형시켜 새로운 해 공간을 제공하는 변이 연산 세 과정을 순서대로 진행하며, 이를 진화 연산이라고도 한다. 이 진화 연산은 연구자가 지정한 실험 조건하에 반복된다. G.A.에 대한 수많은 연구와 함께 성능을 보강하기 위해 다른 방법론을 병행하여 운용하는 혼합형 알고리즘이 제안되어 왔는데, P.

Moscato(1989) [4]에서 제안된 미미틱 알고리즘(M.A.)이 그중 하나다. M.A.의 핵심은 G.A.의 과정을 거친 해 집단에 대하여 해마다 지역 탐색 연산을 통해 지역 최적화를 유도한 후 다음 반복으로 전달하는 것이다.

본 연구에서 비교군으로써 인용 및 구축한 탐색 기반 휴리스틱 알고리즘 일정계획 모델은 S. Balin(2011) [8]에서 제안하는 구조를 따르는 G.A. 기반 일정계획 모델이다. 또한 본 연구에서는 제안하는 일정계획 모델은 M.A. 기반 일정계획 모델(제안 모델)이다.

### 2.4.1 유전 알고리즘

S. Balin(2011) [8]에서 제안하는 G.A. 기반 일정계획 모델을 본 연구에서 활용한 방법에 대해 기술한다. 먼저, 현 문제 상황에서 S. Balin(2011) [8]에서 제안한 적합도 계산법은 수식 (4)와 같다.

$$fitness_{(k)} = \alpha * e^{-\beta * Makespan(k)} \quad (4)$$

이 중  $\alpha$ 와  $\beta$ 의 값은 연구자가 임의로 지정한 값으로, 본 연구에서는 동일한 계산법을 적용한 S. Balin(2017) [9]에서 지정한  $\alpha=100$ 을 주었다. 또한 [9]에서는  $\beta=0.10$ 을 지정하였지만, 실험을 진행하며 겪은 수행착오의 결과 문제의 크기가 클수록 적합도가 너무 작아 진화 연산에 지장을 주었기 때문에  $\beta=0.01$ 로 조정하였다. 한 세대당 유전자 수(Population Size)는 기존과 다른 500을 지정하였다. S. Balin(2011) [8]에서 다룬 문제는 작업 수  $n=9$ , 설비 수  $m=4$ 인 작은 크기이기 때문에 현 문제의 크기에 맞춘 조정이다. 초기해 생성 방법은 각 행, 즉 작업에 대하여 표준정규분포 하에  $m$ 개의 난수를 생성한 후 가장 큰 수를 가지는 설비에 할당한다. S. Balin(2011) [8]에서의 Population Size는 10으로 작은 수이었기 때문에 변이 연산은 수행하지 않았으나, 본 연구에서의 Population Size는 500이기 때문에 S. Balin(2011) [9]에서 제시한 변이법을 문제 상황에 맞게 조정 및 적용하여 임의의 50개의 유전자에 대하여 임의의 행 하나를 임의 변형하도록 변이 연산을 수행하였다.

### 2.4.2 미미틱 알고리즘

본 연구에서 제안하는 알고리즘은 M.A.에 기반을 두었기에 전체적인 흐름은 G.A.와 비슷하나, 각 연산 방법론을 보았을 때에는 G.A. 기반 일정계획 모델과 많은 부분이 상이하기 때문에 제안 모델의 방법론을 모두 기술한다. 각 유전자의 적합도는 Makespan 자체를 평가지표로 사용한

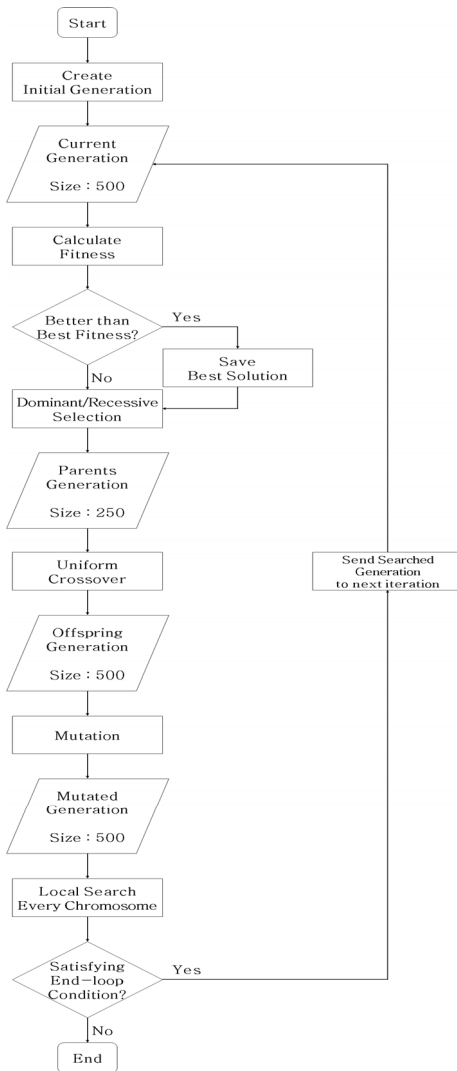
다. 따라서 수립된 일정계획의 Makespan, 즉 평가지표가 낮을수록 높은 성능을 보이는 유전자로 여긴다. Population Size는 500이며, 유전자 하나는  $(n \times 1)$  매트릭스 구조이다. G.A.와 마찬가지로 작업 할당 순서에 대한 정보는 불필요하므로 행의 순서는 작업의 순서와 같다. 한 행이 담고있는 정보는 정수 형태로써 해당 작업이 할당될 설비의 번호를 나타낸다.

초기해 생성 방법은 이를 위한 규칙 기반 휴리스틱 알고리즘을 따른다. 표준정규분포 하에  $n$ 개의 난수를 생성한 후 내림차순으로 작업 할당순서를 결정한다. 그리고 그 순서에 따라 최단 소요 시간을 가지는 설비에 작업을 할당한다. 해당 알고리즘 결과는 하나의 유전자가 생성되며, 유전자의 수가 Population Size만큼 생성될 때까지 반복한다. 선택법은 유전자의 비연속적 진화를 유도함으로 지역 최적해에 수렴하는 것을 방지하기 위하여 [7] 열성 유전자 교합법을 따른다. 현세대 유전자군을 구성하는 모든 유전자에 대하여 성능이 높은 절반을 우성 유전자군, 낮은 절반을 열성 유전자군이라 칭한다. 이후 Population size의 절반 대비 연구자가 임의로 지정하는 비율에 따라 각 유전자군의 상위 유전자를 부모 유전자군에 포함한다.

교차 연산을 수행하기 전, 엘리트주의적 선택 연산[5]을 적용하여 부모 유전자군에서 Population Size 대비 연구자가 임의로 지정하는 비율에 따라 상위 유전자군을 다음 세대 유전자군에 포함시킨다. 교차법은 균일 교차법을 따른다. 두 부모 유전자를 임의로 선택하며 동일한 행, 즉 동일한 작업에 대하여 50%의 확률로 부모 유전자의 정보를 첫 번째 또는 두 번째 자녀 유전자의 동일한 작업에 담는다. 모 유전자의 동일한 작업에 대한 정보는 부모 유전자의 정보를 받지 못한 자녀 유전자에 담는다. 모든 행에 대하여 독립시행하며, 교차 연산 한 번의 결과는 자녀 유전자 2개이다. Population Size만큼의 자녀 유전자로 구성된 다음 세대 유전자군을 생성한 이후, 구성되어있는 유전자마다 연구자가 임의로 지정하는 확률에 따라 연구자가 임의로 지정하는 확률에 따라 돌연변이가 발생하며, 발생할 시 그 변이법은 유전자가 담는 정보 전체를 균일분포 하에  $[0, m)$  범위 정수 형태의 난수를 생성한다.

변이 연산이 완료된 다음 세대 유전자군의 모든 유전자에 대하여 지역 탐색 알고리즘을 수행한다. 지역 탐색을 위한 이웃으로 원본 유전자를 기준으로 첫 번째 작업이 할당될 설비와 두 번째 작업이 할당될 설비를 뒤바꾼 유전자와 두 번째 작업이 할당될 설비와 세 번째 작업이 할당될 설비를 뒤바꾼 유전자 등 최근접 유전정보를 교환한 유전자들을 선정한다. 해당 방법으로 지역 탐색을 수행한 결과, 적합도 평가지표인 Makespan을 단축시킨 유전자가 있다면 원본 유전자를 제외하고 해당 유전자를 다음 세대 유전자군에 포함한다. 지역 탐색 연산까지 완료된 다음 세대 유전자군이

한 세대 진화의 결과이며, 이 유전자군은 다음 진화 과정을 위한 부모 유전자군 후보인 현재대 유전자군으로써 사용한다. 지금까지의 내용을 연구자가 임의로 정한 조건 내에서 반복하며, 이를 흐름도로 표현하면 [Figure 1]와 같다.



[Figure 1] Flow Chart of Memetic Algorithm-based Scheduling Method

제안 모델의 기반으로 M.A.를 채택한 근거를 마련하기 위하여 지역 탐색 연산을 하는가에 따른 자체 비교 실험을 수행하였다. 모델의 구동 시간에 대한 제약은 없으며, 반복 연산 횟수를 1,000회로 제약하였다. 지역 탐색 연산의 여부에 따라 산출된 최고점 유전자가 수립하는 일정계획의 결과를 <Table 2>에 정리하였다. 실험 결과를 통해 지역 탐색 연산이 전체적인 연산 속도는 늦추지만, 선택과 교차, 변이 연산을 완료한 각 유전자와 지역 최적해의 거리를 감소시켜주어 다음 진화 연산에서의 탐색을 보조해주는 역할을 함을 확인할 수 있다. 따라서 본 연구에서는 G.A.를 개선하여 지역 탐색 연산 과정을 추가한 미미틱 일정계획 알고리즘을 개발하였다.

### 3. 성능 평가 실험

#### 3.1 실험 방법

실험을 위한 데이터 세트는 본 연구에서 지정한 문제의 크기마다 10개씩 생성하였다. 본 연구에서는 문제의 크기를 결정하는 요인으로 작업 수  $n = \{50, 100, 500\}$  과 설비 수  $m = \{5, 10\}$  으로 지정하였으므로 그 종류는 6가지이다. 데이터 세트의 형태는  $(n \times m)$  매트릭스 구조이다. 설비에 따라  $\{1, 2, 3\}$ 의 생산 속도값을 보이며, 해당 값이 클수록 느린 속도를 의미하고, 설비의 색인값이 높을수록 느린 속도를 가지도록 임의 지정하였다. 생산시간은 각 작업의 주문량과 설비별 생산 속도값의 곱으로 결정되어 해당하는 항목에 기입된다.

각 모델이 수립한 일정계획에 대한 평가지표를 얻기 위하여, 2.2에서 수립한 수리모형을 토대로 IBM사의 수학적 최적화 소프트웨어 'CPLEX'에 구현하여 모든 데이터 세트의 Makespan을 최소화하는 최적해(Opt.)를 구하였다.

모든 일정계획 모델은 프로그래밍 언어인 Python을 기반으로 구축되었다. 규칙 기반 모델은 모든 데이터 세

<Table 2> Result Table of n Jobs×m Machines Problem comparing with/without Local Search

Problem Size	50×5		100×5		500×5	
Local Search	Average of Makespan	Gap	Average of Makespan	Gap	Average of Makespan	Gap
With	438.82	1.0123	836.36	1.0075	4273.48	1.0000
Without	442.59	1.0210	842.97	1.0155	4321.62	1.0113
Opt.	433.5	1.0000	830.2	1.0000	N/A	N/A
Problem Size	50×10		100×10		500×10	
With	232.85	1.0390	456.24	1.0193	2298.13	1.0000
Without	238.63	1.0648	462.3	1.0328	2303.8	1.0025
Opt.	224.1	1.0000	447.6	1.0000	N/A	N/A

트에 대하여 모델이 수립한 일정계획의 Makespan을 기록하였다. G.A. 기반 일정계획 모델은 진화 연산 속도가 매우 빠른 모델이며, 수립하기까지의 진화 횟수가 문제의 크기에 따라 달랐기 때문에 모든 데이터 세트에 대하여 진화 횟수에 제한을 두지 않고 실제 현장에서의 활용성을 고려하여 프로그램 구동 시간을 1시간으로 제한하여 해당 조건을 어기면 프로그램이 종료되도록 구축하였으며, 모든 세대를 통틀어 최고 성능을 보인 유전자의 Makespan을 기록하였다. 제안 모델은 모든 데이터 세트에 대하여 진화 횟수를 1,000회 혹은 프로그램 구동 시간을 1시간으로 제한하였으며, 두 조건 중 하나를 어기면 프로그램이 종료되도록 구축하였다. G.A. 기반 일정계획 모델과 달리 진화 횟수에 대한 제약을 추가한 이유는 다수의 실험을 통해 문제의 크기 중 작업 수  $n = \{50, 100\}$  일 때는 1,000회 진화연산을 수행하기 이전에 해 탐색이 수렴하는 경향이 파악되었기 때문이다. 최고 성능을 보인 유전자로 수립되는 일정계획에서 Makespan을 기록하였다. G.A. 기반 일정계획 모델과 제안 모델은 연산 수행에 있어 임의성이 존재하기 때문에 모든 데이터 세트에 대하여 10회씩 반복 실험을 수행하여 각 실험 결과의 평균을 기록하였다.

### 3.2 성능 비교

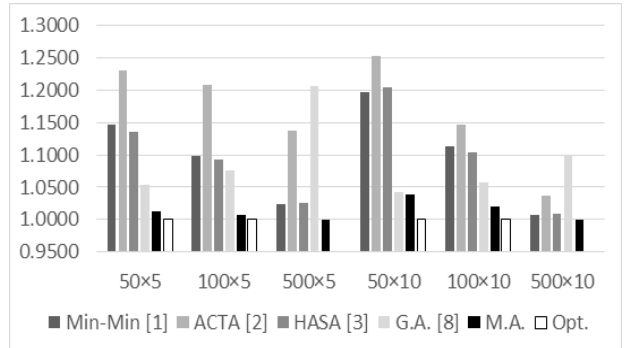
실험 결과를 작성하기에 앞서 실험 결과표의 서식은 <Table 3>와 같다.

<Table 3> Formation Table of Experiment Results

Problem Size	$n \times m$	
Method	Average of Makespan	Gap
Method 1	Average Makespan of 10 Datasets	$Method \div Opt. \text{ or } M.A.$
Method 2		
...		
Opt.		

Opt.의 경우, 작업 수  $n = 500$ 인 문제의 크기가 큰 상황에서는 장시간 가동하여도 최적해를 찾지 못하였기 때문에 Not Available(N/A)로 표기하였다. 비교군 및 제안 모델이 수립한 일정계획을 상대 비교하기 위한 지표로 Opt.와의 차이(Gap)을 사용하였으며, Opt.를 구하지 못하는 문제의 크기에서는 제안 모델이 수립한 일정계획에서의 Makespan을 Gap의 기준으로 계산하였다. <Table 3>와 <Table 4>는 실험 결과를 정리한 표이며, [Figure 2]는 결과 중 상대 비교지표인 Gap을 막대그래프를 통해

시각적으로 표현한 것이다.



[Figure 2] Chart of Comparing Scheduling Models by Gap values

## 4. 결론

본 연구에서는 설비 제약과 작업준비시간이 없고, 분리할 수 없는 작업에 대하여 고장 또는 예방정비가 없는 이중 병렬설비 일정계획 문제에서 Makespan을 최소화하기 위한 미미틱 알고리즘 기반 일정계획 모델을 제안하였다.

제안 모델의 성능 비교를 위해 세 가지 작업 수와 두 가지 설비 수로 총 여섯 가지 문제의 크기에 대하여 각각 열 개의 데이터 세트를 임의 생성하고, 규칙 기반 휴리스틱 알고리즘 일정계획 모델 세 가지[1, 2, 3]와 유전 알고리즘 기반 일정계획 모델[8]로 총 네 가지 비교군 모델이 수립한 일정계획에서 Makespan을 구하였다.

제안 모델은 미미틱 알고리즘 적용을 위해 유전자별 지역 탐색 연산을 수행하였다. 그 외에도 엘리트주의적 선택 연산을 통해 선택된 유전자를 변이 연산 대상에 포함시키는 등, 기존 유전 알고리즘과의 차이점을 두었다. 이렇게 구축된 모델은 네 가지 비교군 모델에 비해 Makespan 측면에서 높은 성능을 보임으로써 좋은 알고리즘임을 입증하였다.

본 연구에서 개발된 알고리즘은 작업준비시간이 없는 이중 병렬설비 환경에의 적용을 목적으로 두고 진행되었기 때문에 작업을 할당하는 순서를 결정하지 않아 작업준비시간이나 설비제약이 존재하는 문제 상황에서는 적용할 수 없다. 따라서 작업준비시간이 있는 문제 상황에서의 방법론 연구 등 추가적인 연구가 필요하다.

<Table 4> Result Table of n Jobs×5 Machines Problem using Scheduling Methods

Problem Size	50×5		100×5		500×5	
Method	Average of Makespan	Gap	Average of Makespan	Gap	Average of Makespan	Gap
Min-Min [1]	496.7	1.1458	911.3	1.0977	4403.9	1.0240
ACTA [2]	533.1	1.2298	1002.8	1.2079	4893.3	1.1378
HASA [3]	492	1.1349	906.6	1.0920	4414.3	1.0264
G.A. [8]	456.9	1.0540	892.8	1.0754	5188.1	1.2064
M.A.	438.82	1.0123	836.36	1.0074	4300.6	1.0000
Opt.	433.5	1.0000	830.2	1.0000	N/A	N/A

<Table 5> Result Table of n Jobs×10 Machines Problem using Scheduling Methods

Problem Size	50×10		100×10		500×10	
Method	Average of Makespan	Gap	Average of Makespan	Gap	Average of Makespan	Gap
Min-Min [1]	268.2	1.1968	498.5	1.1137	2336.7	1.0074
ACTA [2]	280.9	1.2535	513.4	1.1470	2405.4	1.0370
HASA [3]	269.9	1.2044	494.4	1.1046	2340.9	1.0092
G.A. [8]	233.7	1.0428	473.6	1.0581	2549.7	1.0992
M.A.	232.85	1.0390	456.24	1.0193	2319.53	1.0000
Opt.	224.1	1.0000	447.6	1.0000	N/A	N/A

## 5. References

- [1] O. H. Ibarra, C. E. Kim(1977), "Heuristic algorithms for scheduling independent tasks on nonidentical processors." *Journal of the ACM*, 24:280-289.
- [2] A. A. E. A. Elhafiz(2017a), "ACTA: Average of completion time algorithm." *International Journal of Computer Applications*, 172(8):18-22.
- [3] A. A. E. A. Elhafiz(2017b), "HASA: Half the average scheduling algorithm." *Circulation in Computer Science*, 2(9):35-39.
- [4] P. Moscato(1989), "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms." Technical Report C3P 826, Caltech Con-Current Computation Program 158-79, California Institute of Technology, Pasadena, CA.
- [5] S. Baluja, R. Caruana(1995), "Removing the genetics from the standard genetic algorithm." *Machine Learning Proceedings*, 1995:38-46.
- [6] A. Shukla, et al.(2015), "Comparative review of selection techniques in genetic algorithm." 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), pp. 515-519.
- [7] T. Schnier, J. Gero(1998), "Dominant and recessive genes in evolutionary systems applied to spatial reasoning." *Advanced Topics in Artificial Intelligence*, pp. 127-136.
- [8] S. Balin(2011a), "Non-identical parallel machine scheduling using genetic algorithm." *Expert Systems with Applications*, 38(6):6814-6821.
- [9] S. Balin(2011b), "Parallel machine scheduling with fuzzy processing times using a robust genetic algorithm and simulation." *Information Sciences*, 181(17):3551-3569.
- [10] J. H. Holland(1975), *Adaptation in natural and artificial system*. Cambridge: MIT Press.
- [11] Y. G. Kim(2017), *METAHEURISTICS*. Chonnam National University Press.
- [12] E. Horowitz, S. Sahni(1976), "Exact and approximate algorithms for scheduling non-identical processors." *Journal of the ACM*, 23(2): 317-327.
- [13] N. Piersma, W. Van Dijk(1996), "A local search heuristic for unrelated parallel machine scheduling with efficient neighborhood search." *Mathematical and Computer Modeling*, 24(9):11-19.

- [14] B. Srivastava(1998), "An effective heuristic for minimizing makespan on unrelated parallel machines." *Journal of the Operational Research Society*, 49(8):886-894.
- [15] M. Ghirardi, C. N. Potts(2005), "Makespan minimization for scheduling unrelated parallel machines: A recovering beam search approach." *European Journal of Operational Research*, 165(2): 457-467.
- [16] R. Tavakkoli-Moghaddam, et al.(2009), "Design of a genetic algorithm for bi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints." *Computers & Operations Research*, 36(12):3224-3230.
- [17] J. Ding, et al.(2020), "A hybrid memetic algorithm for the parallel machine scheduling problem with job deteriorating effects." *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(3):385-397.

## 저자 소개



### 이 태 희

인천대학교 산업경영공학과에서 학사 취득. 현재 인천대학교 일반대학원 산업경영공학과에서 석사과정 이수 중  
관심분야 : 스마트팩토리, 산업인공지능 등



### 유 우 식

서울대학교 산업공학과와 과학기술원 산업공학과에서 석사, 박사를 취득, 현재 인천대학교 산업경영공학과에서 교수로 재직 중  
관심분야 : CAD/CAM, 제조시스템공학, 물류정보시스템 등