뻐꾸기 탐색법을 이용한 동남㈜ 공정계획 최적화 기법 제안

2016314001

IT경영학과 고광종

CONTENTS

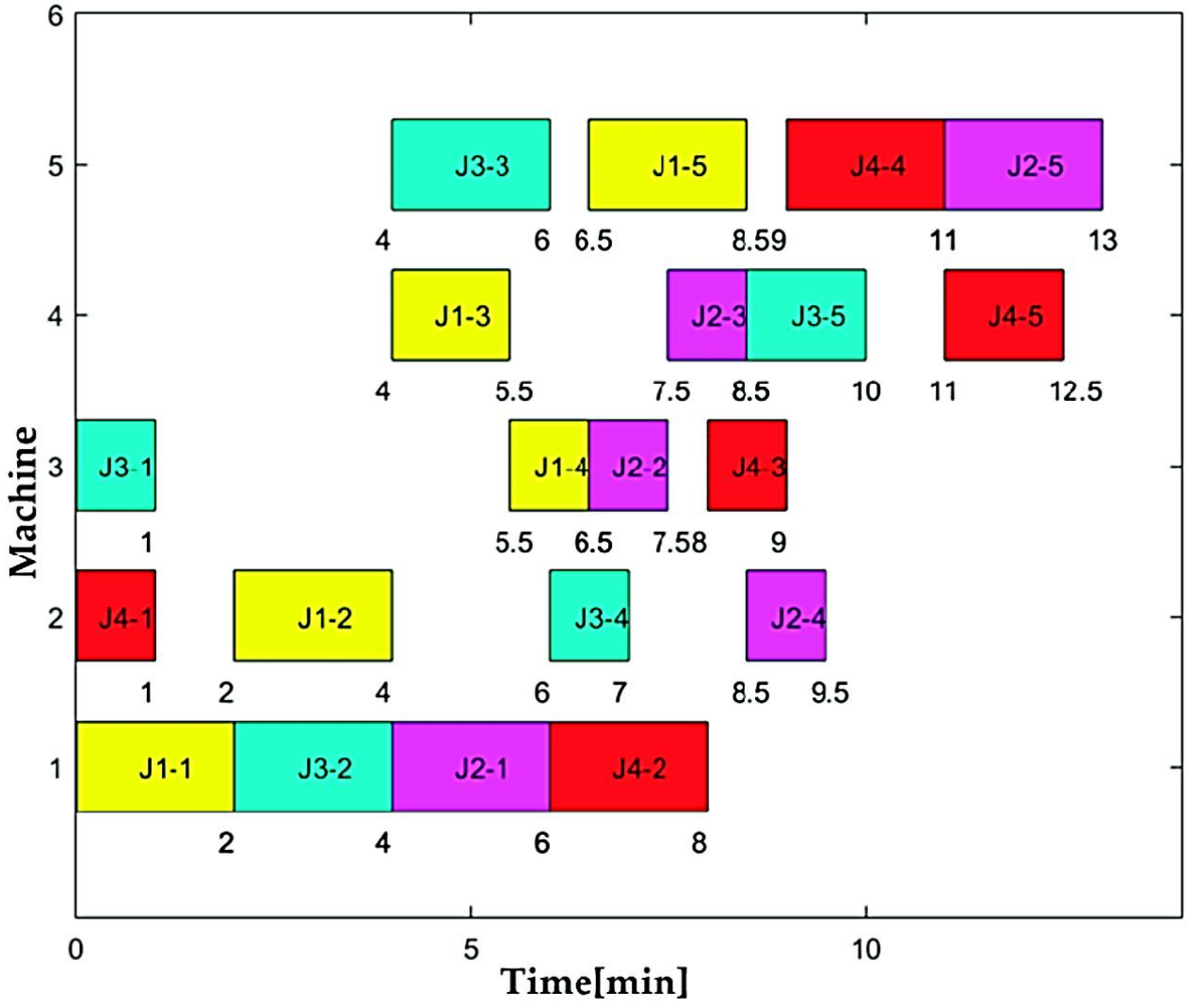
문제 상황

1. 동남㈜의 공정계획문제

- 주식회사 동남(이하 동남)은 경기도 시흥시에 위치한 정밀가공부품 회사
- 동남은 2021년 6월 스마트팩토리 기반 수주 관리 시스템(MES)을 도입하는 등 생산성 향상을 위해 스마트팩토리를 적극적으로 도입하고 있음
- 그러나 복잡성과 비용 문제 때문에 공정계획 최적화에는 스마트팩토리를 도입하지 못하고 있음

문제 상황

1. 동남㈜의 공정계획문제



- 공정계획은 간단하게 “어떤 제품을 어떠한 기계에서 어떠한 순서대로 생산하는가?”를 결정하는 문제
- 공정계획은 수율과 가동률로 이어지고, 이는 생산성과 직결되기 때문에 제조현장에서 매우 큰 이슈
- 공정계획문제를 공학적으로 해결하기 위해 일반화시킨 형태를 FJSP(Flexible Job-Shop Scheduling)이라고 함
  - Job : 제조 제품
    Operation : 제품 제조의 각각의 단계
    Machine : 제조를 하는 기계
    Setup : Job 변경에 대응하기 위한 설정시간
- FJSP는 매우 복잡한 조합 최적화 문제로써, NP-Hard Problem이기 때문에 일반적인 수학적 방법으로 최적해 도출이 불가능하다는 특징을 가짐

문제 상황

1. 동남㈜의 공정계획문제

“

따라서, 본 연구에서는 동남의 제조 스마트팩토리 도입을 위한
뻐꾸기 탐색법(Cuckoo Search) 기반의 제조공정계획 최적화 기법을 제안함

제안 알고리즘

1. FJSP 및 Simulator 설계

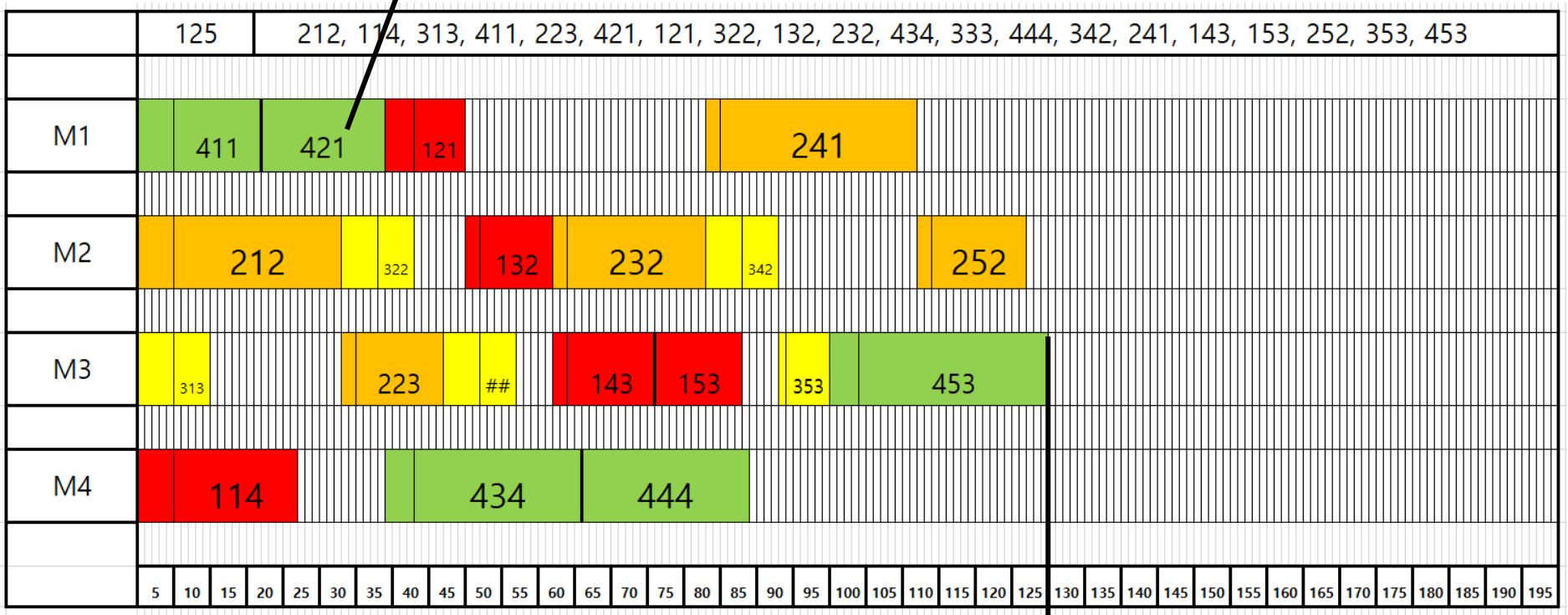
		M1	M2	M3	M4
J1	O1	20	15	18	17
	O2	7		8	
	O3	12	10		15
	O4	14		12	12
	O5	11		12	12
J2	O1		23		32
	O2	22		12	
	O3	23	19		
	O4	27			25
	O5		13	16	
J3	O1	16	5	5	4
	O2		5		5
	O3	12	5	5	7
	O4		5		5
	O5	14	5	6	7
J4	O1	12		14	
	O2	17		18	
	O3		20		21
	O4	22			23
	O5		12	26	
		J1	J2	J3	J4
	J1	5	2	1	4
	J2	4	5	5	4
	J3	2	2	5	4
	J4	4	3	1	5

- 동남은 크게 4가지 파트의 제품을 생산하며, 통합된 선반&밀링 머신을 사용하고 있음.
- 따라서, 총 4개의 Job이 각각 5개의 Operation으로 이루어진다고 가정하며, 이를 생산하는 4대의 Machine이 존재한다고 가정
- 각 Operation들은 1개 이상의 Machine에 할당 가능하며, Machine은 한 번에 하나의 Operation만 할당 받을 수 있다고 가정
- 좌측의 표와 같이 각 Operation이 특정 Machine에서 소요되는 시간과 할당된 Job이 변경될 때의 Setup Time을 설정하며 J1 ~ J4(20개의 Operation)을 모두 끝내면 공정 종료
- **Example**
  - 맨 처음 M3에 J31이 할당될 경우…
    - Setup(5) + Processing Time(5) = 10 소요

제안 알고리즘

1. FJSP 및 Simulator 설계

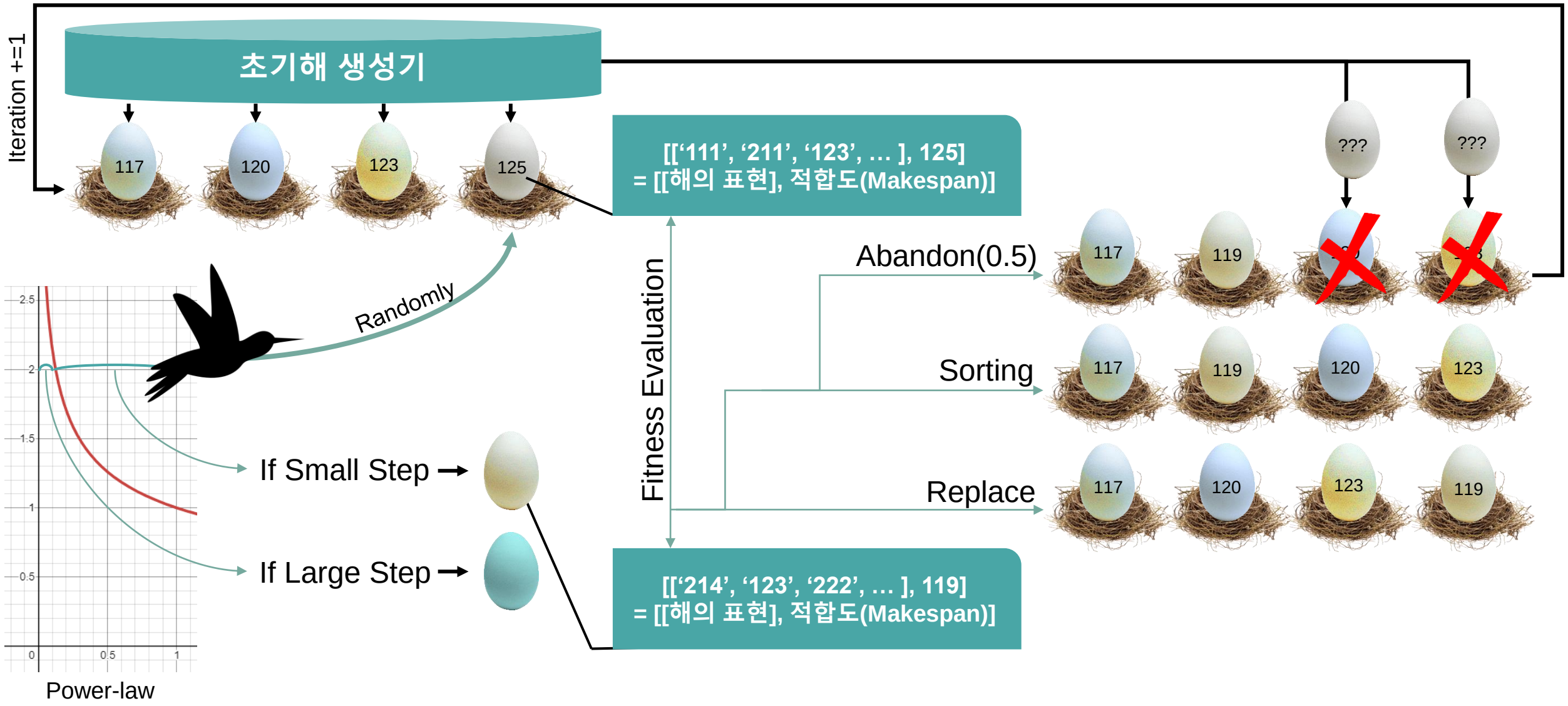
- Job – Operation – Machine으로 이루어진 Solution Encoding으로 해 표현 구현
- Feasible Solution에 대한 제약이 반드시 지켜져야 함



가장 마지막에 끝나는 작업 시간을 기준으로
Makespan 산출 (소요시간이므로 낮을수록 좋음)

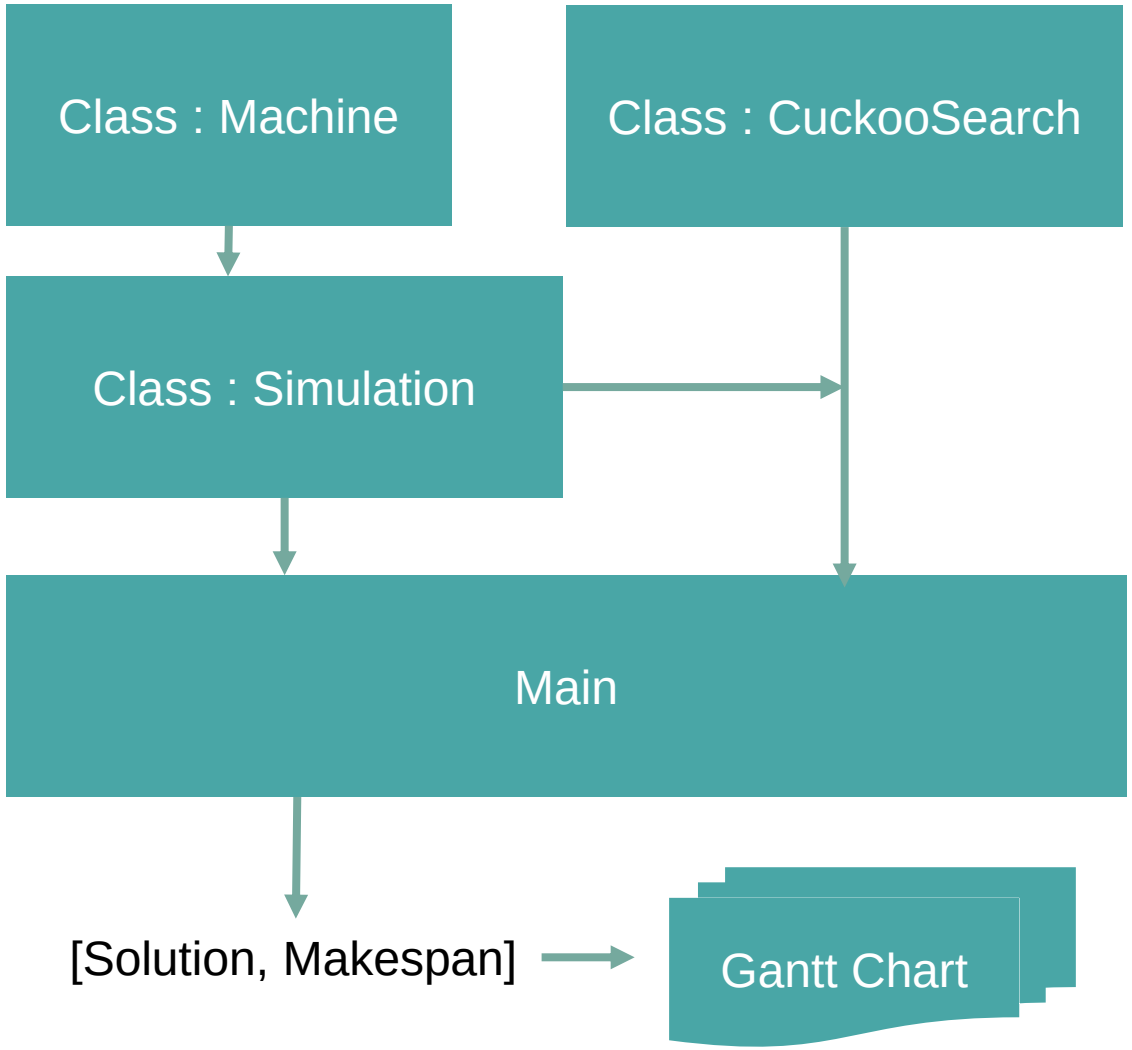
제안 알고리즘

2. Cuckoo Search를 응용한 조합최적화 알고리즘



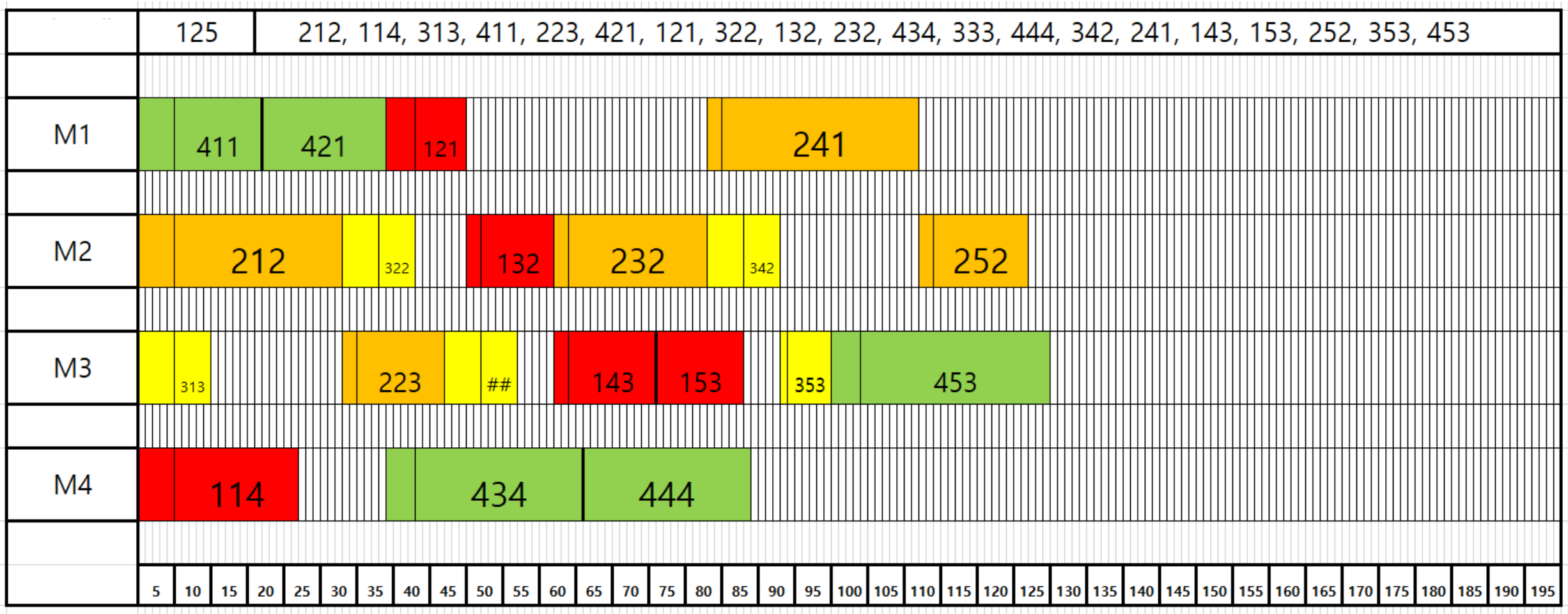
구현 및 결과

1. 구현 환경 및 Hyper-Parameter 세팅

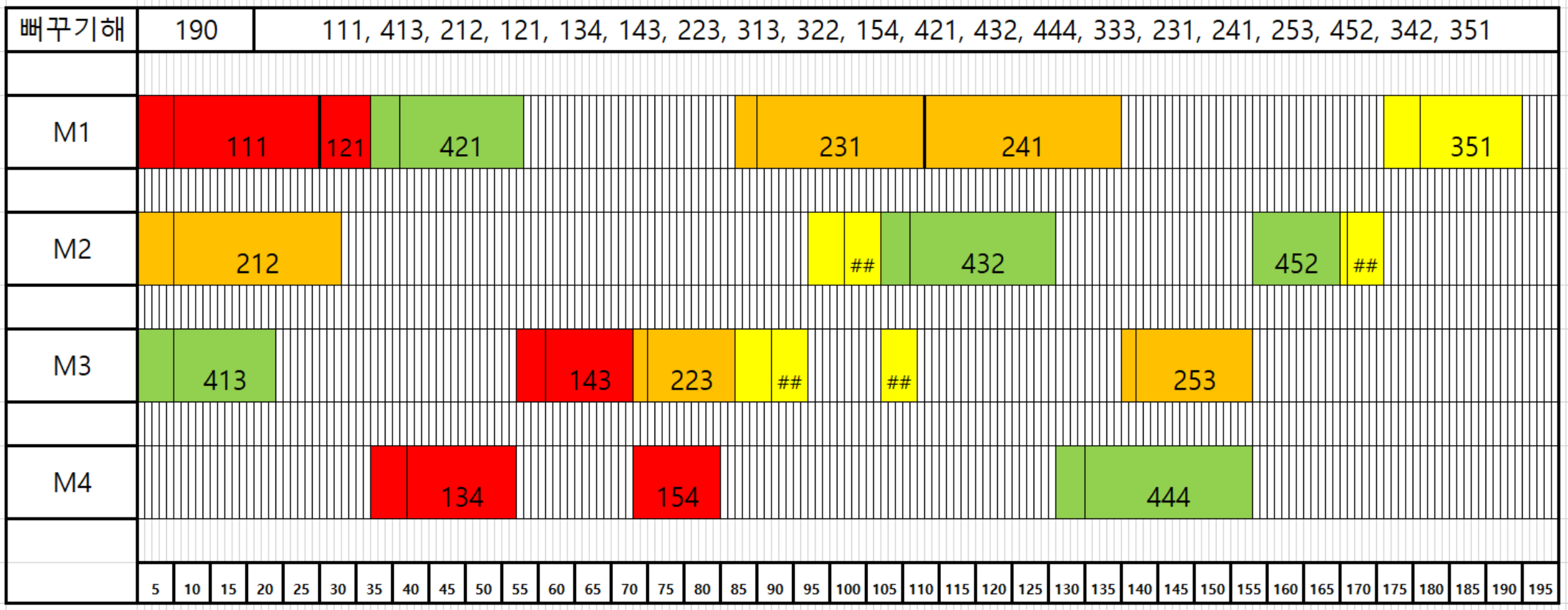


- Hardware : Intel i5 9400F / 16GB RAM
- Programming Language : Python 3.7
- Gantt Chart : Excel
- Hyper-Parameter Setting:
  - Population Size : 20
  - Abandon(Pa) : 0.25
  - Intelligence(Pc) : 0.5
  - Max Iteration : 100000
- 기존의 공정계획은 해당 FJSP에 대해서 생산계획 관련전공 학우가 도출해낸 Heuristic Solution을 적용함

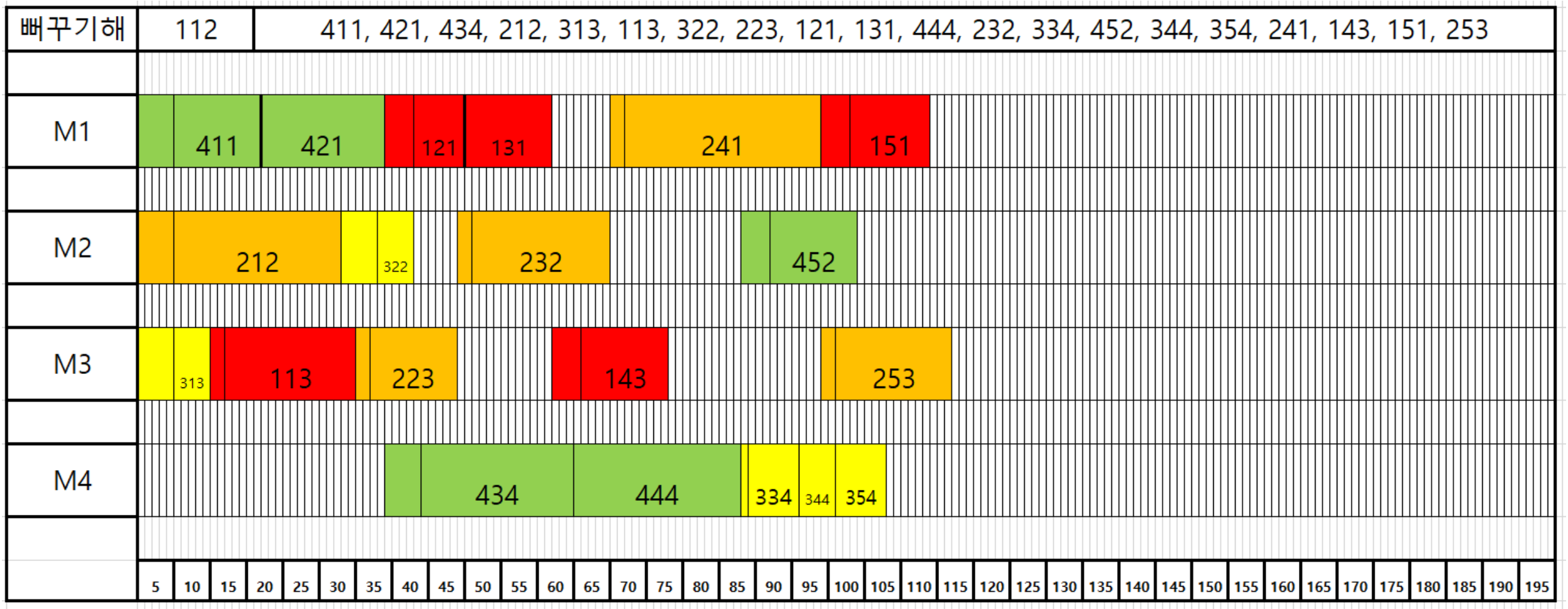
2. 결과 비교 – 생산관리자의 수학적 기법 및 직감을 이용한 Heuristic Solution



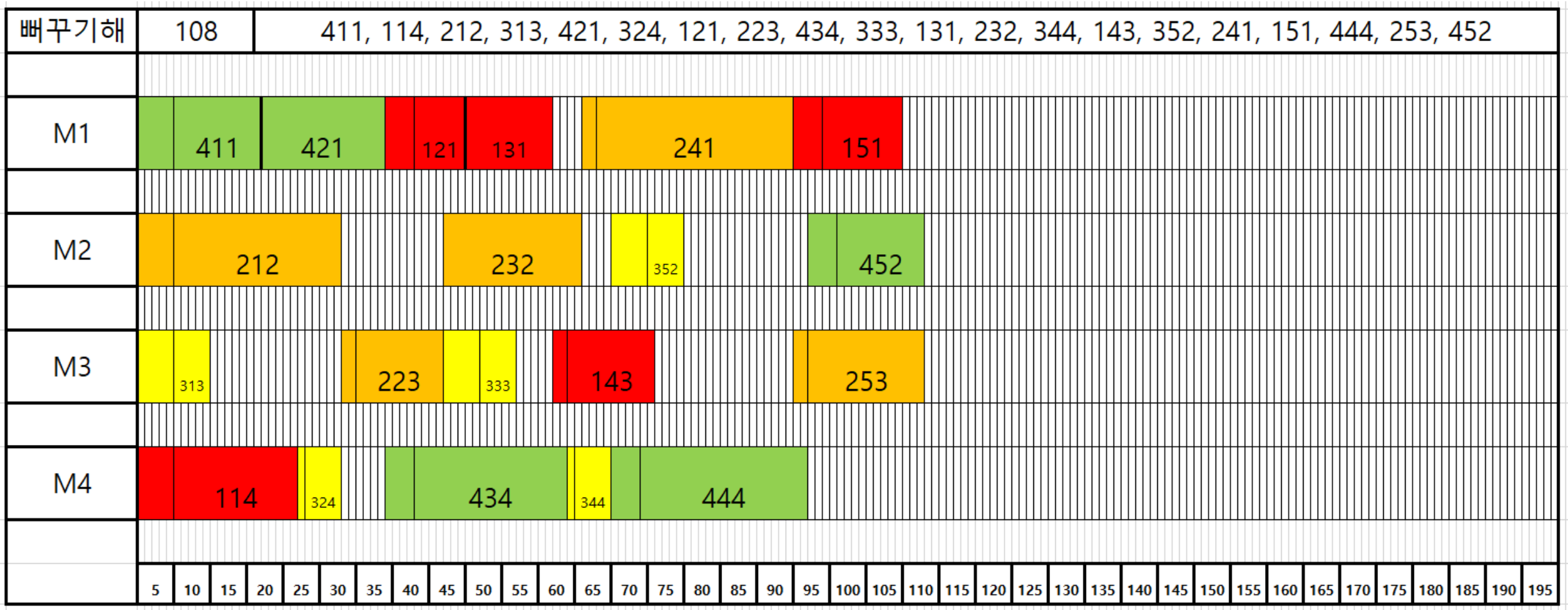
2. 결과 비교 – Cuckoo Search의 초기해 (Iteration : 0/100000)



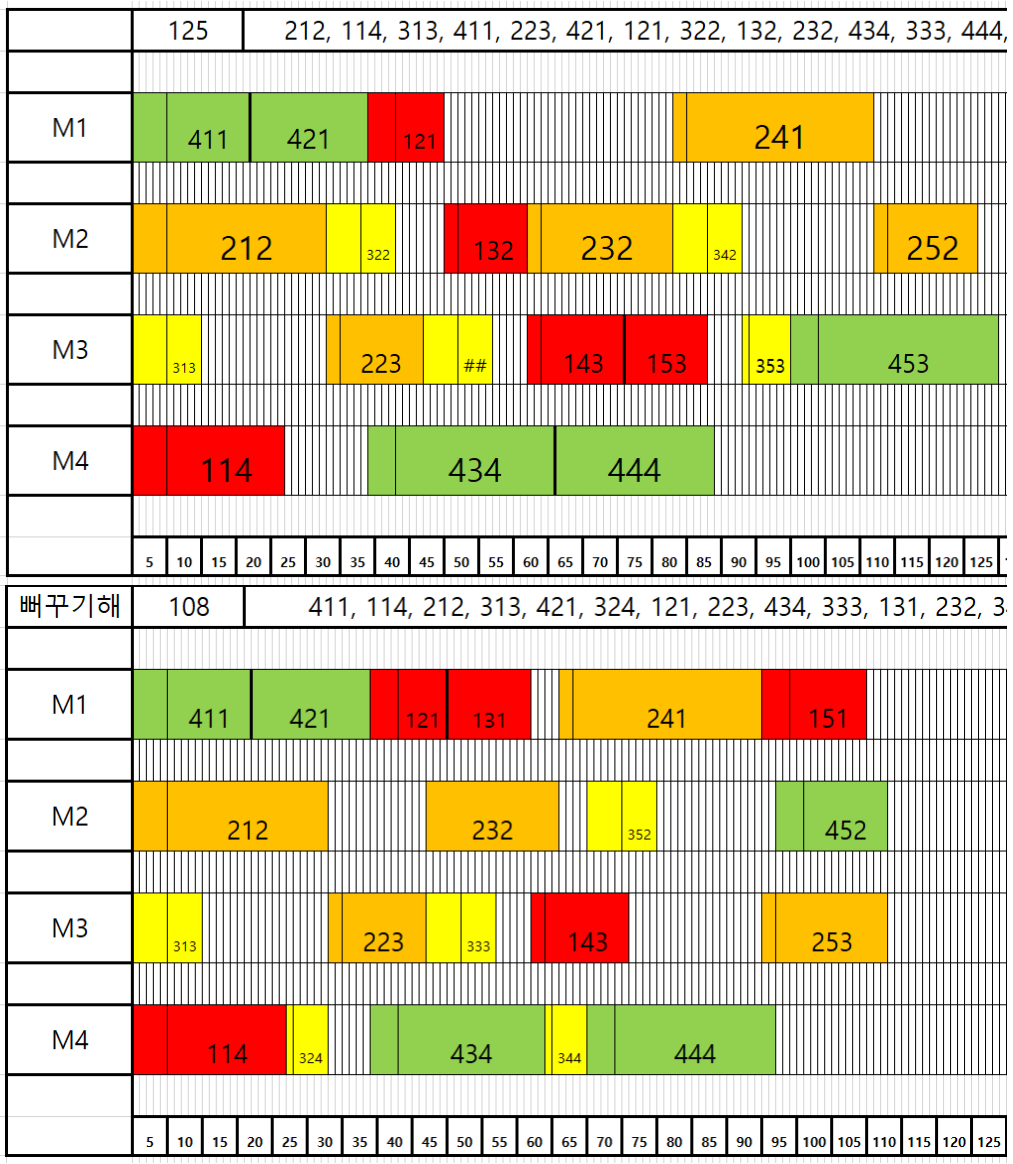
2. 결과 비교 – Cuckoo Search의 중간 과정해 (Iteration : 10000~/100000)



2. 결과 비교 – Cuckoo Search의 수렴해 (Iteration : 80000~/100000)



2. 결과 비교 – Heuristic Search 해 VS Cuckoo Search 해



- Cuckoo Search로 산출한 솔루션의 Makespan은 Heuristic Search 해보다 **15% 이상 더 좋은 성능**을 보였다.

- 생산관리전공 학생은 3번의 시도에서 각 135, 125, 128의 Makespan을 갖는 해를 찾았으며, 평균적으로 한번의 시도 당 약 25분의 시간을 소요하였다.
  - Heuristic Search 기대값 :
    - 기대 Makespan : 129.3
    - 기대소요시간 : 25분

- Cuckoo Search는 20번의 시도에서 108~114의 Makespan을 갖는 해를 찾았으며, 평균적으로 한번의 시도 당 약 3분의 시간을 소요하였다.
  - Cuckoo Search 기대값 :
    - 기대 Makespan : **112**
    - 기대소요시간 : **3분**

추후 연구

- Large-sized FJSP를 추가적으로 구현 후 적용한다면 제안 알고리즘을 더 엄밀하게 검증할 수 있을 것
- Machine Release-time, Lag-time, Arrival-time과 같은 Entity를 추가해서 실효성을 놓일 수 있을 것

- 더 효율적인 해 표현을 통해 Computational Time을 획기적으로 줄일 수 있을 것
- 각 Step별 해 연산자들을 더 개발하여 탐색과 수렴을 더 강력히 수행할 수 있을 것
- Hyper-parameter 튜닝을 통해 전체적인 성능을 향상시킬 수 있을 것

참고 문헌

- Fantahun M. Defersha & Danial Rooyani. (2020). **An efficient two-stage genetic algorithm for a flexible job-shop scheduling problem with sequence dependent attached/detached setup, machine release date and lag-time.** Computers & Industrial Engineering 147.

- Xin-She Yang & Suash Deb. (2009). **Cuckoo Search via Levy Flights.** 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC 2009).

- Aziz Ouaarab, Belaid Ahiod, & Xin-She Yang. (2013). **Discrete cuckoo search algorithm for the travelling salesman problem.** Neural Computing & Applications 24.

- Guohui Zhang, Liang Gao, & Yang Shi. (2011). **An effective genetic algorithm for the flexible job-shop scheduling problem.** Expert Systems with Applications 38.

Thanks