

# 혼합모델 조립라인에서 부품사용의 일정률 유지를 위한 생산순서 결정 : 유전알고리즘 적용

현 철 주\*

\*전북과학대학

## Sequencing Problem to Keep a Constant Rate of Part Usage in Mixed Model Assembly Lines : A Genetic Algorithm Approach

Chul Ju Hyun\*

\*Jeonbuk Science College

### Abstract

This paper considers the sequencing of products in mixed model assembly lines under Just-In-Time (JIT) systems. Under JIT systems, the most important goal for the sequencing problem is to keep a constant rate of usage every part used by the systems.

The sequencing problem is solved using Genetic Algorithm. Genetic Algorithm is a heuristic method which can provide a near optimal solution in real time. The performance of proposed technique is compared with existing heuristic methods in terms of solution quality. Various examples are presented and experimental results are reported to demonstrate the efficiency of the technique.

Keywords : Genetic Algorithm, Sequencing, Mixed Model Assembly Line

### 1. 서론

혼합모델 조립라인은 공장부지 및 시설의 효율적인 이용을 위하여 한 라인에서 유사한 여러 모델의 제품을 생산하고 있다. 이러한 라인의 대표적인 예는 자동차 조립라인으로 각 라인에는 생산제품의 특성과 크기 등이 유사한 여러 종류의 제품이 생산되고 있다. 예로서 승용차 라인에서는 한 라인에서 여러 가지 차종을 생산하고 있으며, 하나의 차종에 대해서도 엔진, 문(door)의 형태, 자동변속기 또는 차의 색상 등의 옵션이 조합되어 각각 다른 모델이 된다.

혼합모델 조립라인에서는 재고비용을 줄이고 수요를 적시에 만족시키기 위하여 모델의 종류와 양을 평균화하여 골고루 혼류로 생산하는 평균화 생산방식을 택하

는 경우가 많다. 평균화 생산방식은 모델별로 작업내용, 작업방법 및 작업시간 등이 상이하여 생산계획과 통제가 복잡하게 된다. 혼합모델 조립라인의 효율적인 이용을 위해서는 라인밸런싱과 모델의 생산순서를 결정하는 문제가 중요하다.

라인밸런싱 문제는 혼합모델을 조립하기 위해 필요한 작업들을 선행제약을 어기지 않고 각 작업장에 균형적으로 할당하는 문제이며, 생산순서 문제는 모델의 수요량에 대한 모델의 생산순서를 다루는 문제이다. 생산순서문제는 라인밸런싱 문제보다 단기적인 측면을 가지고 있으며, 각 모델의 수요율은 자주 변하기 때문에 주기적으로 해결해야 할 필요성이 있다. 본 연구에서는 혼합모델 조립라인에서 모델의 생산순서 문제를 다루고 있다.

혼합모델 조립라인의 생산순서에 관한 연구는 Kilbridge와 Wester[12]에 의해 처음으로 제시된 이래 라인길이의 최소화[6], 가외작업의 최소화[2, 15], 라인정지의 최소화[10], 부품사용의 일정률 유지[7, 8, 13] 등 다양한 목적에 관해 여러 연구가 이루어 졌다. 특히 JIT 생산 시스템에서는 재고비용을 줄이고 고객의 다양한 수요를 적시에 만족시키기 위해 부품사용의 일정률 유지에 관한 목적을 흔히 사용하고 있다.

이 목적은 완제품 및 재공품의 재고량을 가능한 한 최소로 유지시키면서 제품의 수요를 만족시키고자 하는 것이다. 부품 사용의 일정률을 유지하는 제품의 생산순서 결정은 외주업체가 부품의 일정량을 일정간격으로 납품할 수 있고, 완제품 및 재공품 재고의 수준을 최소로 할 수 있으며, 수요의 변화에 빠르게 대처할 수 있다는 장점을 가진다[14].

혼합모델 조립라인의 생산순서 문제는 조합최적화 문제로 수요변화, 결품으로 인한 생산순서의 재결정에는 실시간 처리가 요청된다. 본 연구에서는 혼합모델 조립라인에서 부품의 사용률을 일정하게 유지하기 위한 생산순서를 실시간에 효율적으로 구하기 위해 유전 알고리즘(Genetic Algorithm)을 이용하고자 한다.

유전 알고리즘은 자연계의 적자생존과 생물학적 진화과정을 모방하여 개발된 탐색 기법으로, 순서결정문제, Scheduling 문제, 경로문제 뿐만 아니라 VLSI설계, 패턴인식 등 다양한 분야에 활발히 적용되어 왔으며 [16], 특히 외판원 문제, Flow-shop 순서문제 등과 같은 조합최적화 문제를 해결하는 데 적합한 기법으로 알려져 있다. 유전 알고리즘을 외판원 문제, Flow shop 등의 순서문제에 적용하기 위하여 적합한 표현방법, 생존방법 그리고 유전연산자의 개발을 위해 많은 연구가 이루어져 왔다[11, 16, 18].

유전 알고리즘이 가지고 있는 특성 중의 하나는 빠른 시간에 최적 해에 가까운 해를 찾을 수 있다는 점이다. 이러한 특성으로 인하여 생산순서 문제에 유전 알고리즘을 적용함으로써 효율적으로 준 최적 해를 찾을 수 있다. 본 연구에서는 혼합모델 조립라인 생산순서 문제에 유전 알고리즘을 적용하여 기존에 제시된 발견적 기법들과 해의 탐색성능을 비교분석하고자 한다.

본 연구의 구성은 아래와 같다. 제 2장에서 혼합모델 조립라인의 생산순서문제에 대한 수리모형을 살펴보고, 제 3장에서는 다루는 문제에 적용하고자 하는 유전알고리즘에 필요한 요소들을 다룬다. 그리고 제 4장에서는 실험을 통하여 유전 파라미터 값을 찾고, 유전알고리즘과 기존의 발견적 기법들을 비교분석한다. 제 5장은 결론으로 구성되어 있다.

## 2. 혼합모델 생산순서 문제

혼합모델 조립라인은 한 라인에서 유사한 여러 모델의 제품을 생산하고 있는 라인이다. 혼합모델 조립라인의 생산순서문제는 라인길이의 최소화, 라인정지의 최소화, 부품사용의 일정률 유지, 작업준비비용의 최소화 등 다양한 목적들이 제시되었다[3, 4, 6].

특히 JIT 생산 시스템에서는 부품사용의 일정률을 유지하기 위한 목적이 다른 목적보다 중요시되고 있으며, 이에 관한 최적의 생산순서를 구하기 위한 여러 가지 기법들이 제시되어 왔다. Monden[17]은 부품 사용을 가능하면 일정하게 유지하기 위해 Toyota 자동차 회사에서 사용하는 목표추적법(Goal-Chasing Method)이라는 생산계획의 평준화방법을 설명하였다.

또한 Miltenburg[7]는 JIT 생산시스템에서 단위시간당 부품의 사용을 가능하면 일정하게 유지하는 생산순서를 구하는 수리모형과 발견적 기법을 제시하였고, Miltenburg와 Sinnamon[8]는 Miltenburg[7]의 연구를 확장하여 제품뿐만 아니라 제품을 구성하고 있는 부품들의 실제 생산량과 수요량의 편차를 최소로 하는 생산순서를 구하는 수리모형과 두 가지 발견적 기법(이하 MS\_H1과 MS\_H2로 표기)을 제시하였다.

MS\_H1 방법은 현재 단계에서 실제 생산량과 수요량의 편차 합을 최소로 하는 모델을 투입하는 방법이며, MS\_H2 방법은 현재 단계뿐만 아니라 다음 단계의 실제 생산량과 수요량의 편차 합을 최소로 하는 모델을 투입하는 방법이다.

현철주[1]은 부품의 사용률을 일정하게 유지하기 위한 생산순서를 구하기 위해 Tabu Search를 제안하였으며, Cho et al.[9]은 준비 작업시간의 최소화과 부품 사용 일정률 유지에 관한 목적을 가중치 방법으로 결합하여, 이를 최적화하는 생산순서를 구하기 위해 Simulated Annealing 기법을 적용하였다.

또한 Giovanni et al.[5]는 목표 추적법을 풀기위한 발견적 기법을 다양한 상황에서 비교분석하였으며, Ding et al.[3]은 JIT 생산 시스템에서 사용하는 여러 가지 투입순서의 목적을 비교분석하고 있다.

JIT 생산 시스템에서는 평준화생산을 위해 계획기간(Planning horizon)동안 생산될 모델의 개수를 반복 생산하여 계획기간의 수요를 만족시키는 사이클 생산을 흔히 사용하고 있다.

본 연구에서는 사이클 생산에서 제품뿐만 아니라 제품을 구성하고 있는 반제품, 부품, 원재료들의 사용량을 일정하게 유지하기 위한 생산순서를 구하기 위해 Miltenburg와 Sinnamon[8]이 제시한 수리모형을 사용한다.

## 2.1 사용된 기호

- L : 수준의 수
- n<sub>j</sub> : 수준 j 의 생산품 수
- d<sub>ij</sub> : 수준 j 에서 생산품 i 의 수요
- DT<sub>j</sub> =  $\sum_{i=1}^{n_j} d_{ij}$  j = 1, ..., L : 수준 j 의 총 수요
- r<sub>ij</sub> = d<sub>ij</sub>/DT<sub>j</sub> : 수준 j 에서 생산품 i 의 수요율
- k : 단계 수
- x<sub>ijk</sub> : 단계 1에서 k 동안에 생산된 수준 j 의 생산품 i 의 단위 수
- XT<sub>jk</sub> =  $\sum_{i=1}^{n_j} x_{ijk}$  : 단계 1에서 k 동안에 수준 j 에서 생산된 총 단위 수
- w<sub>j</sub>, j = 1, ..., L : 수준 j 의 가중치

## 2.2 수리모형

수리모형에서 목적함수는 제품뿐만 아니라 제품을 구성하고 있는 반제품, 부품, 원재료들의 실제 생산량 ( $x_{ijk}$ )과 수요량 ( $XT_{jk}r_{ij}$ )의 편차 합을 나타내며, 이를 최소로 하기 위한 생산순서( $x_{ilk}$ )를 구하고자 한다.

$$\text{MIN. } \sum_{k=1}^{DT_1} \sum_{j=0}^4 \sum_{i=0}^{n_j} w_j (x_{ijk} - XT_{jk}r_{ij})^2$$

$$\text{S.T. } XT_{1k} = k, \quad k=1, 2, \dots, DT_1$$

$$0 \leq x_{ilk} - x_{il(k-1)} \leq 1$$

$$x_{ilk} \text{ 는 정수, } i=1, 2, \dots, n_1, \\ k=1, 2, \dots, DT_1$$

목적함수에서  $w_j$ 는 각 수준(level)의 중요도를 나타낸다. 첫 번째 제약 식은 k개의 제품이 정확히 k 단계 동안에 생산되어야 함을 나타내며,  $DT_1$ 은 계획기간 동안에 생산되어야 하는 총 제품수를 나타낸다. 두 번째와 세 번째 제약 식은 각각 주어진 단계에서 각 제품은 한 단위를 초과하여 생산하거나 분할하여 생산할 수 없다는 정수조건을 나타낸다.

## 3. 유전 알고리즘

유전 알고리즘은 해를 표현하는 개체로 구성된 모집단을 변화시키면서 최적 해를 탐색한다. 모집단의 각 개체는 평가함수에 의해 평가되며, 이를 기초로 적합한 개체가 생존되어 새로운 모집단을 형성한다. 새로 형성

된 모집단의 개체는 교차변이 또는 돌연변이의 유전연산을 통해 새로운 개체를 생산한다. 이 과정을 종료조건이 만족할 때까지 반복하게 된다.

유전알고리즘을 적용하기 위해서는 먼저 문제에 대한 특성을 분석하여 적합한 유전알고리즘을 선택하고, 그에 따른 표현방법(Representation), 평가함수(Evaluation Function), 생존(Selection), 유전연산자(Genetic operator) 및 유전파라미터의 결정 등이 이루어져야 한다.

### 3.1 수정 유전 알고리즘

유전 알고리즘의 효율을 높이고 계산시간의 단축을 위해 Modified GA, Messy GA, Parallel GA 등 다양한 유전 알고리즘이 개발되었다[18].

본 연구에서는 Michalewicz [18]이 제시한 수정 유전 알고리즘을 기본으로 한다. 수정 유전알고리즘의 절차를 단계별로 정리하면 다음과 같다. P(t)는 세대 t에서의 모집단을 의미하고, pop\_size 는 모집단의 개수를 표시한다.

- 단계 1 : 초기모집단(P(t=0))을 생성하고 평가한다.
- 단계 2 : 모집단 P(t)에서 k 의 개체를 선택한다.
- 단계 3 : 단계 2에서 선택된 개체들을 교차변이 또는 돌연변이를 시켜 새로운 k 개의 개체를 생산한다.
- 단계 4 : 모집단 P(t+1)에 생산된 k 의 개체를 복사한다.
- 단계 5 : 모집단 P(t)에서 pop\_size -k 의 서로 다른 개체를 선택하여 모집단 P(t+1)에 복사한다.
- 단계 6 : 모집단 P(t+1)을 평가한다.
- 단계 7 : 종료조건이 만족하면 알고리즘의 수행을 마치고, 그렇지 않으면 t = t+1 로 두고 단계 2로 간다.

수정 유전알고리즘의 중요한 특징은 다음과 같다. 수정 유전알고리즘에서는 높은 평가함수 값을 갖는 초개체(Super individual)가 다음 세대에 여러 개 나타나는 것을 방지함과 동시에 상대적으로 평가함수 값이 좋은 개체가 생존가능성이 높아지며(단계 3), 이 개체들이 다른 개체와 재결합하여 더 좋은 개체를 생산할 확률이 높게 된다(단계 5). 또한 수정 유전알고리즘에서는 돌연변이가 개체단위로 처리되어 선택된 k개의 개체 중에서 몇 개의 개체는 교차변이, 나머지 개체는 돌연변이를 사용하여 자손을 생산한다.

유전 알고리즘을 다루는 문제에 적용하기 위해서 첫째, 문제에 대한 해를 나타내기 위한 개체의 표현방법 둘째, 환경에 대한 적응도를 측정하는 평가함수 셋째, 다음 세대에 생존시키기 위한 생존방법 넷째, 새로운

해를 탐색해 내는 유전연산자 다섯째, 유전 알고리즘에서 사용된 여러 가지 파라미터 값의 결정 등에 대해 살펴본다.

본 연구에서는 개체의 표현방법으로 사이클 생산 내에서 생산되어야 하는 모델들의 목록을 생산 순서대로 나열하는 방법을 사용하고자 한다. 예로 한 사이클 동안 모델 A, B, C가 각각 2, 3, 4개 생산될 때, 생산순서가 A, C, B, C, B, A, C, C, B라면 개체의 표현방법은 모델의 생산 순서대로 ( A C B C B A C C B )와 같이 표현한다. 또한 초기 모집단의 개체들은 해 공간을 가능한 넓게 탐색할 수 있도록 각 모델의 수요량을 만족하는 해를 임의로 발생하여 사용한다.

평가함수는 각 개체의 생존에 대한 적응 척도로서, 모집단 내의 각 개체들은 평가함수에 의해 평가되며, 평가된 값을 기초로 하여 다음 세대에 재생산시킬 개체들이 선택된다. 다루는 문제에서 평가함수는 모델의 생산순서에 따라 발생하는 생산량과 수요량의 편차 합을 나타내며, 제 2장의 목적함수에 의해 구해진다.

생존방법은 비례 재생산(Proportionate reproduction), 토너먼트 생존(Tournament selection), 순위생존(Ranking selection) 등으로 나눌 수 있다. 생존은 좋은 해의 효율적 탐색(Exploitation)과 해공간의 효율적 탐색(Exploration)에 커다란 영향을 미치는 요소로, 평가함수 값에 비례하여 생존하는 비례적 재생산 방법을 흔히 사용하고 있다. 그러나 다루는 문제와 같이 최소화 문제이거나 각 개체의 평가함수 값이 별로 차이가 없을 때에는 Roulette wheel을 구성하기 위하여 척도(Scaling)을 만들어야 하는 어려움이 있다. 본 연구에서는 순위에 따라 각 개체의 생존확률이 결정되는 순위생존방법[18]을 사용한다.

순위 생존방법은 모집단의 모든 개체를 평가함수 값이 좋은 순서로 나열하고 순위에 따라 각 개체의 생존 확률을 선형 또는 비선형 함수를 사용하여 결정하는 방법이다. 본 연구에서는 평가함수 값이 가장 좋은 개체부터 차례로 나열하여 순위를 1,2,3,...으로 주고, 각 개체의 생존 확률을 아래와 같은 기하분포를 사용하여 모집단을 재생산하기로 한다.

$$Prob(rank) = q(1 - q)^{rank - 1},$$

$$rank = 1, 2, 3, \dots, pop\_size$$

여기서 q는 생존확률 파라미터로  $0 < q \leq 1$  의 값을 갖는다.

유전연산자는 새로운 해를 탐색해 가는 방법으로, 문제의 특성을 잘 반영할 수 있는 정보를 부모로부터 효

율적으로 추출하여 자손을 생산할 수 있어야 하며, 연산은 다루는 문제의 가능해가 유지되도록 해야 한다.

유전알고리즘에서 연산자는 교차변이와 돌연변이로 나누고 있으며, 본 연구에서 교차변이는 수정 순서교차변이를, 돌연변이는 역순[16]을 사용하기로 한다. 사용된 유전연산자는 다루는 문제의 가능 해를 생산해 낸다.

수정 유전 알고리즘에서 결정해야 할 파라미터는 모집단의 크기(pop\_size), 연산자 적용비율(k), 교차변이와 돌연변이의 비율 등이 있다. 본 연구에서는 파라미터에 대한 기존의 연구결과를 토대로, 실험을 실시하고 결과 분석을 통해 파라미터 값을 결정하기로 한다.

## 4. 실험 결과

이 장에서는 다양한 문제를 통해 유전알고리즘과 기존에 제시된 Tabu Search[1], Miltenberg와 Sinnamon[8]이 제시했던 두 가지 탐색적 방법( MS\_H1, MS\_H2)을 비교분석하고자 한다.

비교분석을 위해 유전알고리즘은 C++ 언어를 사용하여 실험하였으며, 실험의 신뢰도를 높이기 위하여 매 실험마다 초기 해를 달리하여 30회 반복 실험하였다.

유전알고리즘은 많은 예비실험을 통해 연산자의 적용 개체 수(k)는 모집단 크기의 50%로, 교차변이와 돌연변이의 비율은 5 : 5로, 생존확률 파라미터(q)는 각 개체의 평균 생존확률( $1/pop\_size$ )의 5배로 두었다.

또한 해의 성능향상을 위해 한 세대에서 가장 좋은 개체가 다음 세대에서도 항상 보존되는 우수개체(Elitism) 생존전략을 사용하였다. 이 전략은 지금까지 탐색한 가장 좋은 해를 생존과정에서 잃어버리지 않게 방지한다.

### 4.1 예제 1

본 연구에서는 첫 번째 예제로 Miltenburg와 Sinnamon[8]이 제시한 예제를 사용하였다. 이 예제는 3종류의 모델을 각각 6000, 6000, 1000개를 생산한다. 따라서 한 사이클 당 생산되는 모델별 생산제품 수는 각각 6, 6, 1개가 되며, 한 사이클 당 생산되는 총 제품 수(DT1)는 13개가 된다.

예제의 부품구성은 4수준(1수준 : 제품, 2수준 : 반제품, 3수준 : 부품, 4수준 : 원재료)으로 구성되어 있다. 즉, 각 제품은 3개의 반제품으로, 각 반제품은 4개의 부품으로, 또한 각 부품은 3개의 원재료로 이루어져 있다. 이에 관한 부품 구성표는 <표 1>과 같다.

<표 1> 예제 1에 대한 부품구성표

		제 품			
		1	2	3	
반 제 품	1	1	0	0	
	2	1	1	1	
	3	0	0	4	
		반 제 품			
		1	2	3	
부 품	1	1	0	1	
	2	1	1	0	
	3	0	1	0	
	4	0	0	4	
		부 품			
		1	2	3	4
원 재 료	1	1	0	0	1
	2	0	1	0	1
	3	1	0	1	0

<표 2>는 본 연구에서 제시한 유전알고리즘과 MS\_H1, MS\_H2, 그리고 Tabu Search(TS)를 사용하여 구한 목적함수 값을 나타내고 있다. 비교분석을 위해 유전알고리즘과 Tabu Search에서 탐색한 이웃해의 개수는 같게 하였으며, 결과 값은 초기 해를 달리하여 30회 실험한 결과의 평균값을 사용하였다.

문제의 크기가 작은 예제에서는 가중치에 상관없이 유전알고리즘과 Tabu Search기법이 MS\_H1, MS\_H2 방법보다 같거나 좋은 해를 찾아내었다.

4.2 예제 2

두 번째 예제에서는 문제의 크기를 확장하여 여러 가지 기법에 대한 해의 효율을 비교분석하였다. <표 3>은 모델의 수와 한 사이클 당 생산되는 모델별 생산제품 수에 따라 분류한 9개의 문제를 나타내고 있다.

<표 3>에서 문제 II-2의 경우에는 5개의 모델을 생산하며, 한 사이클 당 생산되는 모델별 생산제품 수는 각각 10, 5, 3, 1, 1개로, 한 사이클 당 생산되는 총 제품 수는 20개가 되며, 가능해의 총 수는  $9.3117 \times 10^8$  개가 된다.

<표 2> 해의 성능 비교분석

문제	가중치	MS_H1	MS_H2	TS	유전알고리즘
I - 1	(1, 1, 1, 1)	338.183	324.033*	324.033*	324.033*
I - 2	(0, 1, 1, 1)	352.310	325.662	315.026*	315.026*
I - 3	(0, 0, 1, 1)	382.357	342.883	291.369*	291.369*
I - 4	(0, 0, 0, 1)	120.173	116.844	114.180*	114.180*
I - 5	(1, 0, 0, 0)	5.077	4.615*	4.615*	4.615*
I - 6	(0, 1, 0, 0)	19.380*	19.380*	19.380*	19.380*

\*가장 좋은 해

<표 3> 성능평가를 위한 문제의 설계

문제	생산모델															총수요 (DT <sub>i</sub> )	가능해의 개수
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
II - 1	16	1	1	1	1											20	$1.1628 \times 10^5$
II - 2	10	5	3	1	1											20	$9.3117 \times 10^8$
II - 3	5	4	4	4	3											20	$2.4443 \times 10^{11}$
II - 4	21	1	1	1	1	1	1	1	1	1						30	$5.1917 \times 10^{12}$
II - 5	7	6	5	3	3	2	1	1	1	1						30	$8.4602 \times 10^{20}$
II - 6	4	3	3	3	3	3	3	3	3	2						30	$3.2900 \times 10^{24}$
II - 7	26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	40	$2.0231 \times 10^{21}$
II - 8	9	6	5	4	3	2	2	2	1	1	1	1	1	1	1	40	$2.2589 \times 10^{34}$
II - 9	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	40	$4.2167 \times 10^{38}$

모든 문제의 부품 구성표는 4수준(1수준 : 제품, 2수준 : 반제품, 3수준 : 부품, 4수준 : 원재료)으로 구성되어 있으며, 각 수준의 구성 부품 수는 모델 수와 동일하게 두었다.

<표 4>는 각 수준별 부품 구성표를 나타내고 있다. 여기서 문제 II-1, 2, 3의 경우는 모델 수가 5개 이므로 각 수준별 부품 구성표에서 처음 (5x5)행렬 값을 사용하며, 문제 II-4, 5, 6의 경우는 모델 수가 10개이므로 각 수준별 부품 구성표에서 처음 (10x10)행렬 값만을 사용하고, 문제 II-7, 8, 9의 경우는 모델 수가 15개이므로 각 수준별 부품 구성표에 제시된 데이터를 모두 사용하게 된다.

<표 4> 예제 2에 대한 부품 구성표

		제 품																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
반 제 품	1	1	0	2	0	1	0	1	0	1	0	0	0	0	2	0	0	0
	2	1	0	1	1	0	1	0	1	0	1	0	2	0	0	0	0	0
	3	0	2	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
	4	1	1	0	0	1	0	1	0	0	0	1	0	0	2	0	0	0
	5	2	0	1	0	2	0	0	1	0	0	1	0	0	1	0	0	1
	6	0	0	1	0	0	0	0	1	0	1	0	0	1	2	1	0	0
	7	0	0	0	1	0	1	2	0	0	0	0	1	1	0	0	0	0
	8	0	0	1	0	0	0	0	0	1	0	0	0	2	0	1	0	0
	9	0	1	0	0	0	1	2	0	0	0	0	1	0	0	2	0	0
	10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	11	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
	12	0	0	1	0	0	0	2	0	2	0	1	0	0	0	0	0	0
	13	0	0	0	2	1	1	0	2	0	0	0	0	0	0	1	0	0
	14	0	0	1	0	0	0	2	0	0	1	0	2	1	0	0	0	0
	15	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0

		반 제 품																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15			
부 품	1	2	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	
	2	1	0	2	1	0	0	0	0	0	1	0	0	1	0	1	0	1	
	3	0	0	2	1	0	0	0	1	1	1	1	1	0	0	1	0	0	
	4	0	2	0	0	1	1	2	0	0	0	0	1	0	0	0	0	0	
	5	1	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0	
	6	0	0	1	1	2	0	0	0	1	1	0	0	2	1	0	0	0	
	7	0	1	0	0	0	0	2	0	0	0	0	0	0	1	0	1	0	
	8	0	1	0	1	0	2	0	0	0	1	0	1	0	0	0	0	0	
	9	0	0	0	2	0	0	0	0	0	0	0	1	0	1	0	0	0	
	10	1	0	2	0	0	0	0	0	0	0	0	0	0	1	0	2	0	
	11	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0
	12	0	2	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0
	13	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	14	0	0	0	0	1	0	0	0	1	0	0	2	0	0	0	0	0	0
	15	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

		부 품																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
원 재 료	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0
	2	0	1	2	0	1	0	2	0	0	0	2	1	0	0	2	0	0
	3	0	0	0	2	1	0	0	0	0	0	2	0	2	0	1	0	0
	4	0	1	0	1	0	0	2	0	0	0	1	0	0	0	0	0	0
	5	1	2	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
	6	2	0	0	0	0	0	0	2	0	1	1	0	0	0	0	0	0
	7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0
	8	1	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0
	9	2	0	0	0	2	0	2	0	0	1	0	0	0	0	0	0	0
	10	0	0	1	0	1	2	0	0	0	0	1	0	2	0	1	0	0
	11	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0
	12	1	1	0	2	0	0	1	0	0	1	0	1	0	0	0	0	0
	13	0	0	1	0	0	1	0	0	0	0	0	0	0	0	2	0	0
	14	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
	15	1	0	2	0	0	0	0	0	0	0	1	0	2	0	0	0	0

<표 5>는 각 기법들을 사용하여 구한 탐색성능을 나타내고 있다.

유전알고리즘과 Tabu Search에서 탐색한 이웃해의 수는 문제 II-1, 2, 3의 경우 10000개, II-4, 5, 6의 경우 20000개, II-7, 8, 9는 30000개를 탐색하였다.

<표 5> 각 기법들의 탐색성능 비교분석

가중치 문제	MS_H1	MS_H2	TS	GA	
	II-1	207.934	207.934	161.205*	161.205*
	II-2	188.107	161.149	160.933*	160.933*
	II-3	183.331	180.922	148.778*	148.778*
	II-4	1140.365	1008.236	965.789*	970.721
(1,1,1,1)	II-5	1219.044	999.720	1037.700	1017.120*
	II-6	1643.501	1382.668	1133.401*	1137.211
	II-7	7304.571	6822.721	5745.936	5683.211*
	II-8	8593.404	5796.565	5952.324	5763.412*
	II-9	8453.959	6266.656	6496.030	6242.127*
	II-1	192.934	192.934	145.705*	145.705*
	II-2	172.607	145.363*	145.363*	145.363*
	II-3	197.894	166.040	135.028*	135.028*
	II-4	1077.532	940.789	919.055	902.153*
(0,1,1,1)	II-5	1339.701	933.298	955.599	931.591*
	II-6	1523.790	1220.459	1098.549*	1104.591
	II-7	7135.196	7441.983	5558.928	5421.981*
	II-8	8395.979	5613.925	5922.759	5772.179*
	II-9	8126.084	6703.086	6096.799	6021.279*
	II-1	141.807	141.807	124.017*	124.017*
	II-2	187.834	139.081	113.377*	113.377*
	II-3	180.059	130.413	107.142*	107.144
	II-4	966.423	841.437	818.386	804.526*
(0,0,1,1)	II-5	1148.216	868.961*	891.694	872.941
	II-6	1262.327	1032.967	978.616	955.165*
	II-7	6577.894	6963.029	5388.710	5331.211*
	II-8	7404.955	5212.886	5598.141	5495.114*
	II-9	7330.500	6868.005	5970.467	5770.473*
	II-1	110.305	110.305	97.915*	97.915*
	II-2	128.986	128.631	88.512*	88.512*
	II-3	115.846	113.015	84.179*	84.212
	II-4	787.501	601.283	580.601	571.635*
(0,0,0,1)	II-5	902.665	794.111	643.562	623.534*
	II-6	804.652	727.248	698.515	681.578*
	II-7	6359.024	5798.445	4464.511	4394.511*
	II-8	6785.713	5356.486	4630.325	4556.325*
	II-9	6457.931	4993.135	4992.739	4821.791*
	II-1	13.500*	13.500*	13.500*	13.500*
	II-2	10.000*	10.000*	10.000*	10.000*
	II-3	11.350*	11.350*	11.350*	11.350*
	II-4	45.500*	45.500*	45.500*	45.500*
(1,0,0,0)	II-5	31.289	31.156	30.889*	30.889*
	II-6	40.844*	40.844*	40.844*	40.844*
	II-7	93.375*	93.375*	93.375*	93.375*
	II-8	62.075*	62.075*	62.075*	62.075*
	II-9	76.625*	76.625*	76.625*	76.625*
	II-1	17.793	17.793	17.520*	17.520*
	II-2	20.358	19.358*	19.675	19.413
	II-3	23.985	21.956	17.774*	17.788
	II-4	72.567	70.326	62.515*	62.515*
(0,1,0,0)	II-5	83.471	83.139	65.814	63.813*
	II-6	103.652	85.463	66.991	65.915*
	II-7	185.392	179.820	187.250	178.203*
	II-8	200.431	196.036	195.267	193.163*
	II-9	234.076	210.913	200.661	197.161*

유전알고리즘과 Tabu Search의 결과는 30회 실험한 결과의 평균값을 사용하였으며, 문제의 크기가 커질수록, 가중치가 각 수준별로 골고루 주어질 때, 목적함수

값이 커짐을 알 수 있다.

9개의 문제에 대해 각각 가중치를 변화시키면서 MS\_H1, MS\_H2, Tabu Search, 그리고 유전알고리즘을 비교한 결과, 유전알고리즘이 총 54개의 문제 중 46개 문제에서, Tabu Search는 27개의 문제에서, MS\_H2는 11개의 문제에서 같거나 더 좋은 결과를 나타내어 해의 질 면에서는 유전알고리즘이 상당히 좋음을 알 수 있다.

문제의 크기가 작은 II-1, 2, 3의 경우 유전알고리즘과 Tabu Search 방법은 거의 최적 해를 찾아감을 알 수 있다. 또한 문제의 크기가 작은 경우에는 유전알고리즘과 Tabu Search가 비슷한 성능을 보이지만 문제의 크기가 커질수록 본 연구에서 제시한 유전알고리즘이 더 나은 성능을 보임을 알 수 있다.

문제에서 제품만의 실제 생산량과 수요량의 편차를 최소로 하는 생산순서를 구하는 형태인 가중치 (1, 0, 0, 0)의 경우 문제가 단순해짐으로써 목적함수 값의 많은 중복을 가져와 최적 해와 같은 해가 다수 발생하여 모든 방법이 최적 해를 쉽게 찾을 수 있었다.

### 5. 결론

본 연구에서는 혼합모델 조립라인의 생산순서에 관한 여러 형태의 목적 중에서 JIT 시스템에서 특히 중요시하는 목적인 부품 사용의 일정률을 유지하기 위한 목적을 다루었다. 다루는 목적에 대해 유전알고리즘을 적용하여 실시간에 효율적인 생산순서를 구할 수 있는 방법을 제시하고, 기존에 제시된 발견적 기법과 해의 효율측면에서 비교 분석하였다.

다양한 문제를 통해 각 기법들을 비교 분석한 결과, 예제에 따라 정도의 차이는 있으나 본 연구에서 제시한 유전알고리즘이 해의 효율이 더 좋았다.

본 연구에서 제시한 유전알고리즘은 재고수준을 최소로 유지시키면서 제품의 수요를 만족시키고자 하는 JIT 시스템에서 현장의 상황 변화에 따라 생산순서를 재결정하고자 할 때 다루는 목적에 대해 최적에 가까운 생산순서를 실시간에 구할 수 있다.

### 6. 참고 문헌

[1] 현철주, "자동차 조립라인에서 부품사용의 일정률 유지를 위한 투입순서 결정, 안전경영과학회지, 제 4권 제 3호, pp. 95-105, 2002.  
 [2] C. J. Hyun, Y. K. Kim and Y. Kim, "A Genetic Algorithm for Multiple Objective Sequencing

Problems in Mixed Model Assembly Lines", Computers & Ops Res, 25(7-8), pp.675-690, 1998.

[3] Fong-Yuen Ding, Jin Zhu, Hui Sun, "Comparing two weighted approaches for sequencing mixed-model assembly lines with multiple objectives", Int. J. Production Economics, pp. 108-131, 2006.

[4] F. Kolahan and M. Liang, "An adaptive TS approach to JIT sequencing with variable processing times and sequence-dependent setups", Eur. J. Operational Research, pp. 142-159, 1998.

[5] Giovanni Celano, Antonio Costa, and Sergio Fichera, "A comparative analysis of sequencing heuristics for solving the Toyota Goal Chasing problem", Robotics and Computer-Integrated manufacturing, Vol 20, pp. 573-581, 2004

[6] J. F. Bard, A. Shtub and S. B. Joshi, "Sequencing mixed-model assembly lines to level parts usage and minimize line length", Int. J. Prod. Res., Vol. 32, pp. 2431-2454, 1994.

[7] J. Miltenburg, "Level schedules for mixed-model assembly lines in just-in-time production systems", Mgmt. Sci., Vol. 35, pp. 192-207, 1989.

[8] J. Miltenburg and G. Sinnamon, "Scheduling mixed-model multi-level just-in-time production systems", Int. J. Prod. Res., Vol. 27, pp. 1487-1509, 1989.

[9] Hyung-Soo Cho, Chun-Hyun Paik, Hang-Mook Yoon, and Ho-Gyun Kim, "A robust design of simulated annealing approach for mixed-model sequencing", Computers & IE, Vol 48, pp. 753-764, 2005

[10] K. Okamura and H. Yamashina, "A heuristic algorithm for the assembly line model-mix sequencing problem to minimize the risk of stopping the conveyor", Int. J. Prod. Res., Vol. 17, pp. 233-247, 1979.

[11] K. Takashi, K. Hiroaki and N. Masakazu, "A hybrid Search for genetic algorithms : Combining genetic algorithms, TABU Search and Simulated annealing", Proceedings of the fifth International Conference on Genetic Algorithms, pp. 641, 1993.

[12] M. Kilbridge and L. Wester, "The assembly line model-mix sequencing problem", Proc. of the Third Int. Conf. Op. Res., Oslo, Paris: English Universities Press, pp. 247-260, 1963.

[13] P. R. McMullen, "An efficient frontier approach to addressing JIT sequencing problems with setups via search heuristics", Computers & IE, pp. 335-353, 2001.

[14] R. T. Surnichrast, R. S. Russell and B. W. Taylor III, "A comparative analysis of sequencing procedures for mixed-model assembly lines in a just-in-time production system", Int. J. Prod. Res. Vol. 30, pp. 199-214, 1992.

[15] T. Tamura, H. Long and K. Ohno, "A sequencing problem to level part usage rates and work loads for a mixed-model assembly line with a bypass subline", International Journal Of Production Economics, pp. 557-564, 1999.

[16] Y. K. Kim, C. J. Hyun and Y. Kim, "Sequencing in Mixed Model Assembly Lines : A Genetic Algorithm Approach", Computers & Ops Res., Vol.23, No. 12, pp. 1131-1145, 1996.

[17] Y. Monden, Toyota Production System, An Integrated Approach to JIT, Third Edition, Engineering and Management Press, 1998.

[18] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, Berlin, 1992.

## 저 자 소 개

현 철 주



전남대학교 산업공학에서 학사, 석사, 박사학위를 취득하였다. 현재 전북과학대학에서 조교수로 재직 중이며, 주요 관심분야는 최적화기법 및 영상검색 분야의 Genetic Algorithm과 Tabu Search 응용 및 관련기법 개발 등이다.

주소: 전북 정읍시 시기동 9-28