

자동차 조립라인에서 총 가외작업을 최소로 하는 투입순서 결정

- Sequencing to Minimize the Total Utility Work in Car Assembly Lines -

현철주*

Hyun Chul Ju

Abstract

The sequence which minimizes overall utility work in car assembly lines reduces the cycle time, the number of utility workers, and the risk of conveyor stopping. This study suggests mathematical formulation of the sequencing problem to minimize overall utility work, and present a genetic algorithm which can provide a near optimal solution in real time.

To apply a genetic algorithm to the sequencing problem in car assembly lines, the representation, selection methods, and genetic parameters are studied. Experiments are carried out to compare selection methods such as roulette wheel selection, tournament selection and ranking selection. Experimental results show that ranking selection method outperforms the others in solution quality, whereas tournament selection provides the best performance in computation time.

1. 서론

자동차 조립라인은 공장부지, 시설 등의 투자를 줄이고 시설의 효율적인 이용을 위하여 한 라인에서 유사한 여러 모델의 제품을 생산하고 있다. 예로서 승용차 라인에서는 여러 가지 차종을 생산하고 있으며, 하나의 차종에 대해서도 엔진, 문의 형태, 자동 변속기 등의 옵션이 조합되어 각각 다른 모델이 된다.

* 정인대학 품질관리과

자동차 조립라인에서는 재고비용을 줄이고 수요를 적시에 만족시키기 위하여 모델의 종류와 양을 평준화하여 끌고루 혼류로 생산하는 평준화 생산방식을 택하는 경우가 많다. 평준화 생산방식에서는 모델에 따라 작업장에서 수행하는 작업과 작업시간이 다르기 때문에 어떤 작업장에서 모델에 대한 작업시간은 사이클 타임보다 클 수 있으며, 이러한 제품이 연속적으로 투입되면 작업혼잡이 발생하여 컨베이어 정지위험이 커지고 사이클 타임이 길어진다. 라인의 효율적인 이용을 위해서는 모델의 투입순서를 결정하는 문제가 중요하게 다루어 져야 한다.

자동차 조립라인의 투입순서에 관한 연구는 라인길이의 최소화[7], 컨베이어 정지위험의 최소화[9], 가외작업(Utility work)의 최소화[11, 12], 부품사용의 일정률 유지[1, 8] 등 다양한 목적들에 대해 여러 연구가 이루어 졌다. 가외작업을 최소로 하는 투입순서의 결정은 작업혼잡을 피할 수 있으며, 라인 상에 필요한 가외 작업자의 수 또는 컨베이어의 정지위험을 낮게 하여 생산 능력을 높일 수 있게 한다. 본 연구에서는 자동차 조립라인에서 발생하는 총 가외작업의 최소화를 목표로 연구를 수행하였다.

자동차 조립라인의 투입순서 문제는 생산 모델의 개수가 많아지면 최적해를 구하기 어려운 NP-hard 문제이다. 또한 모델의 수요 변화, 부품의 결품 등에 따라 투입순서의 재결정에는 실시간 처리가 필요하다. 본 연구에서는 자동차 조립라인에서 발생하는 가외작업을 최소로 하는 투입순서를 위한 수리모형을 제시하고, 유전 알고리즘을 이용하여 효율적인 투입순서를 실시간으로 결정하고자 한다.

유전 알고리즘(Genetic Algorithm)은 자연계의 적자생존과 생물학적 진화과정을 모방하여 개발된 탐색 기법으로, 순서결정문제, Scheduling 문제, 경로문제 뿐만 아니라 VLSI설계, 패턴인식 등 다양한 분야에 활발히 적용되어 왔으며[2], 특히 외판원 문제, Flow-shop 순서문제 등과 같은 조합최적화 문제를 해결하는 데 적합한 기법으로 알려져 있다[14]. 유전 알고리즘을 외판원 문제, Flow shop 등의 순서문제에 적용하기 위하여 적합한 표현방법, 선별방법 그리고 유전연산자의 개발을 위해 많은 연구가 이루어져 왔다[5, 13]. 유전 알고리즘이 가지고 있는 특성 중의 하나는 빠른 시간에 최적해에 가까운 해를 찾을 수 있다는 점이다. 이러한 특성으로 인하여 투입순서 문제에 유전 알고리즘을 적용함으로써 효율적으로 준 최적해를 찾을 수 있다.

본 연구에서는 유전 알고리즘의 성능에 큰 영향을 미치는 선별방법에 대해 실험을 통해 비교·분석하여 적합한 선별방법을 찾고, 다루는 문제에 유전 알고리즘을 적용하여 실시간으로 효율적인 투입순서를 결정하고자 한다.

본 연구의 구성은 아래와 같다. 제 2장에서 자동차 조립라인의 총 가외작업을 최소로 하는 수리모형을 제시하고, 제 3장에서는 선별방법을 중심으로 적용하고자 하는 유전알고리즘에 필요한 요소들을 다룬다. 그리고 제 4장에서는 실험을 통하여 유전 파라미터 값을 찾고, 선별방법들을 비교분석한다. 제 5장은 결론으로 구성되어 있다.

2. 모형화

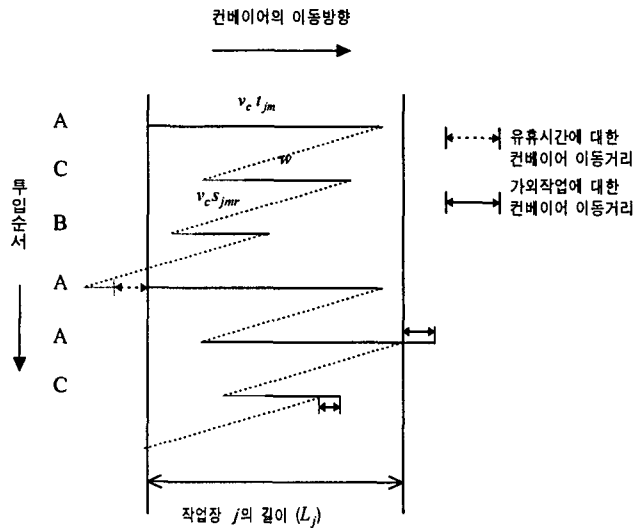
2.1 문제의 설정

본 연구에서 다루는 자동차 조립라인은 일정한 속도(v_c)로 이동하는 컨베이어 시스템으로 유사한 여러 모델의 제품들이 일정시간 간격으로 투입되고 있으며, 컨베이어의 이동에 따라 작업자가 이동하면서 제품을 조립하는 라인이다. 라인은 J 개의 작업장으로 구성되어 있으며, 각 작업장은 한 사람의 작업자로 운영된다. 또한 각 작업장은 작업을 위한 설비나 장비의 제약으로 인하여 경계가 존재하며, 작업장에 할당된 작업은 그 작업장 내에서 수행되어야 한다. 각 작업장에서는 모델 변화에 따른 작업준비가 발생할 수 있으며[6], 이러한 준비시간은 모델 변화에 종속되어 조립시간에 포함시킬 수 없다고 본다[10].

자동차 조립라인에서 계획기간(Planning horizon)동안 생산될 각 모델별 개수를 각각 D_1, \dots, D_M 이라고 하고, D_1, \dots, D_M 의 최대공약수를 h 라 하자. 각 모델의 수요 $(D_1/h, \dots, D_M/h) = (d_1, \dots, d_M)$ 를 최소부분집합(Minimum Part Set : MPS) 이라고 한다 [2]. 자동차 조립라인에서는 MPS를 h 번 반복 생산함으로써 계획기간의 수요를 만족시키는 사이클 생산을 흔히 사용하고 있다. 사이클 생산은 단 한번의 사이클에 대한 생산량의 투입순서를 결정함으로써 사이클이 반복될 때마다 작업자의 유휴시간이 발생하거나, 첫 번째 제품의 시작점(Z_{1j})이 그 작업장의 원점보다 더 먼 위치에 있을 수 있어 주어진 작업장 내에서 작업을 완료할 수 없는 경우가 발생한다. 본 연구에서는 사이클 생산에서 가외작업을 최소로 하는 제품의 투입순서를 결정하고자 한다.

작업자는 컨베이어의 이동에 따라 조립작업을 실시한 후, 다음 제품의 작업준비가 필요하다면 컨베이어 이동과 무관하게 작업준비를 한다. 작업자가 조립작업과 다음 제품의 작업준비를 마친 시점에 작업장 경계내에 제품이 도착한 경우에는 조립작업을 수행하고, 그렇지 않는 경우에는 제품이 도착할 때까지 작업자는 유휴하게 된다.

가외작업은 제품이 작업장의 경계에 이르러 할당된 작업을 마치지 못한 양을 의미한다. 가외작업은 조립작업의 종료위치가 작업장의 실제길이(L_j)보다 큰 경우에 발생하며, 가외작업자에 의해 수행된다. <그림 1>은 작업장 j 에서 발생하는 가외작업을 나타낸다. 그림에서 A, C, B, A, A, C 모델 순으로 생산하는 경우, 굵은 실선은 조립작업에 필요한 작업장 길이를 나타내고, 가는 실선은 작업준비에 필요한 작업장 길이를 나타낸다. 또한 점선은 제품간의 간격을 나타내고, 화살표로 표시된 점선과 실선은 각각 작업자의 유휴시간과 가외작업 시간동안 움직인 컨베이어의 이동거리를 나타낸다. 그림에서 3번째 제품인 B모델을 조립하고 4번째 제품인 A모델의 작업준비를 마친 후, A모델이 작업장의 경계 내에 도달할 때까지 작업자는 유휴하게 된다. 또한 작업장의 경계를 벗어나 작업을 마치는 모델, 즉 5번째 A모델에서 가외작업이 발생하게 되며, 투입순서에서 마지막 제품인 모델 C는 다음 사이클에서 첫 번째 제품의 시작점이 작업장의 원점보다 더 먼 위치에 있기 때문에 발생하는 가외작업이다.



<그림 1> 작업장 j에서 발생하는 가외작업

본 연구에서 제품의 투입시간 간격(v)은 각 모델의 평균 작업시간(v_1)과 각 모델 변화에 따른 평균 준비시간(v_2)의 합으로 둔다.

2.2 수리모형

수리모형에 사용된 기호를 정의하면 다음과 같다.

첨자

- i = 제품의 투입위치; $i = 1, 2, \dots, I$.
- j = 작업장; $j = 1, 2, \dots, J$.
- m, r, p = 모델; $m, r, p = 1, 2, \dots, M$.

입력 자료

I = 제품의 총 수; $I = \sum_{m=1}^M d_m$.

J = 작업장의 수.

M = 모델의 수.

v_c = 컨베이어의 속도.

d_m = 모델 m 의 수요량.

t_{jm} = 작업장 j 에서 모델 m 의 조립시간.

s_{jmr} = 작업장 j 에서 모델 m 에서 모델 r 로 변경에 따른 준비시간.

$$T = \text{총 조립시간} ; T = \sum_{j=1}^I \sum_{m=1}^M t_{jm} d_m .$$

$$S = \text{총 준비시간} ; S = \sum_{j=1}^I \sum_{m=1}^M \sum_{m \neq r, r=1}^M s_{jmr} d_m d_r .$$

$$v = \text{투입시간 간격} ; v = v_1 + v_2,$$

$$\text{여기서 } v_1 = T / (I \times J), \quad v_2 = S / (I \times (I-1) \times J).$$

$$w = \text{연속된 두 제품 사이의 거리} ; w = v_c v.$$

$$L_j = \text{작업장 } j \text{의 실제 길이.}$$

결정변수

$X_{imr} = 1$, 투입순서에서 모델 m 이 i 번째에 위치하고, 모델 r 이 $(i+1)$ 번째에 위치한 경우,
 $= 0$, 그외의 경우.

$Z_{ij} =$ 작업장 j 에서 i 번째 제품의 작업 시작 위치.

자동차 조립라인에서 발생하는 가외작업을 최소로 하는 투입순서를 위한 수리모형을 개발하기 위하여 작업자의 이동시간은 무시할 수 있으며, 각 작업장은 라인벨런싱 되었다고 가정한다. 가외작업을 최소로 하는 투입순서를 구하기 위한 수리모형은 다음과 같다.

$$\text{Min. } \sum_{j=1}^I \left(\sum_{i=1}^I U_{ij} + Z_{(i+1)j} / v_c \right) \tag{0}$$

$$\text{S.T. } \sum_{m=1}^M \sum_{r=1}^M X_{imr} = 1 \quad \forall i \tag{1}$$

$$\sum_{m=1}^M X_{imr} = \sum_{p=1}^M X_{(i+1)rp} \quad \forall i, r \tag{2}$$

$$\sum_{i=1}^I \sum_{m=1}^M X_{imr} = d_r \quad \forall r \tag{3}$$

$$Z_{(i+1)j} = \max \left[0, \min \left(Z_{ij} + v_c \sum_{m=1}^M \sum_{r=1}^M X_{imr} (t_{jm} + s_{jmr}) - w, L_j - w \right) \right] \quad \forall i, j \tag{4}$$

$$U_{ij} = \max \left[0, \left(Z_{ij} + v_c \sum_{m=1}^M \sum_{r=1}^M X_{imr} t_{jm} - L_j \right) / v_c \right] \quad \forall i, j \tag{5}$$

$$X_{imr} = 0 \text{ or } 1 \quad \forall i, m, r \tag{6}$$

$$Z_{ij} = 0, Z_{ij} \geq 0 \quad \forall i, j \quad (7)$$

$$U_{ij} \geq 0 \quad \forall i, j \quad (8)$$

목적함수(Objective function)

전체 작업장에서 모든 생산 제품에 걸쳐 발생하는 총 가외작업을 최소화 하는 투입 순서를 구하는 것을 목적으로 한다. 목적함수에서 두 번째 항은 한 사이클이 끝난 후 다음 사이클이 시작될 때, 첫 번째 제품의 시작위치가 원점보다 먼 위치에 있어서 발생하는 가외작업을 고려한 것이다.

제약식(Constraints)

제약식 (1)은 투입순서의 각 위치에는 단 하나의 모델만이 있어야 한다는 위치제약 이고, 제약식 (2)는 투입된 모델은 라인상에서 모델이 변할 수 없음을 나타낸다. 제약 식 (3)은 각 모델의 수요를 만족시켜야 하는 수요제약을 나타내고 있다. 제약식 (4)는 작업장 j 에서 $(i+1)$ 번째 제품의 작업시작 위치에 관한 제약이고, 제약식 (5)는 j 작업장에서 i 작업에 대한 가외작업(U_{ij})을 나타낸다. 제약식 (6)은 0 또는 1의 결정변수를 나타내고, 제약식 (7)에서 $Z_{ij} = 0, \forall j$ 는 각 작업장의 첫 번째 제품의 작업시작 위치가 0 이라는 것을 의미하고, 다른 것들은 비음제약을 나타낸다.

3. 유전 알고리즘

유전 알고리즘은 해를 표현하는 개체로 구성된 모집단을 변화시키면서 최적해를 탐색한다. 모집단의 각 개체를 평가함수에 의해 평가하여, 이를 기초로 적합한 개체를 선별하여 새로운 모집단을 형성한다. 새로 형성된 모집단의 개체를 교차변이 또는 돌연변이를 시켜 새로운 개체를 생산한다. 이 과정을 종료조건이 만족할 때까지 반복하게 된다.

3.1 수정 유전 알고리즘

유전 알고리즘의 효율을 높이고 계산시간의 단축을 위해 Modified GA [14], Messy GA [4], Parallel GA [14] 등 다양한 유전 알고리즘이 개발되었다. 본 연구에서는 Michalewicz [14]이 제시한 수정 유전 알고리즘을 기본으로 한다. 수정 유전알고리즘의 절차를 단계별로 정리하면 다음과 같다. $P(t)$ 는 세대 t 에서의 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 k 개의 개체 중에서 몇 개의 개체는 교차변이, 나머지 개체는 돌연변이를 사용하여 자손을 생산한다.

유전 알고리즘을 문제에 적용하기 위해서 첫째, 문제에 대한 해를 나타내기 위한 개체의 표현방법 둘째, 환경에 대한 적응도를 측정하는 평가함수 세째, 다음 세대에 생존시키기 위한 선별방법 네째, 새로운 해를 탐색해 내는 유전연산자 다섯째, 유전 알고리즘에서 사용된 여러가지 파라미터 값의 결정 등이 이루어져야 한다.

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

평가함수는 각 개체를 선별하기 위한 척도로서, 모집단 내의 각 개체들은 평가함수에 의해 평가된다. 다루는 문제에서 평가함수는 모델의 투입순서에 따라 발생하는 총가의작업을 나타내며, 제 2장의 목적함수에 의해 구해진다.

선별방법은 유전 알고리즘의 성능에 커다란 영향을 미치는 요소로, 다루는 문제에 적절한 선별방법을 선택해야 한다. 본 연구에서는 다루는 문제에 적합한 선별방법을 찾기 위해 실험을 통해 비교분석한다. 이에 관해서는 3.2절에서 구체적으로 다룬다.

유전연산자는 새로운 해를 탐색해 가는 방법으로, 교차변이와 돌연변이로 나누고 있다. 본 연구에서 교차변이는 수정 순서교차변이를, 돌연변이는 역순[13]을 사용하기로 한다. 사용된 유전연산자는 다루는 문제의 가능해를 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 연구결과

를 토대로, 실험을 실시하고 결과분석을 통해 파라미터 값을 결정하기로 한다.

3.2 선별방법

선별은 각 개체의 평가함수 값을 기초로 다음 세대에 생존할 개체를 결정한다. 선별은 좋은 해의 효율적 탐색(Exploitation)과 해공간의 효율적 탐색(Exploration)에 커다란 영향을 미치는 요소로, 이들을 잘 조화시킬 수 있어야 한다.

선별방법은 비례적 재생산(Proportionate reproduction), 토너먼트 선별(Tournament selection), 순위 선별(Ranking selection) 등으로 나눌 수 있다[5]. 비례적 재생산 방법은 각 개체의 평가함수 값에 비례하여 각 개체의 선별 확률을 결정하는 방법으로, 확률바퀴 선별(Roulette wheel selection) 방법이 흔히 쓰인다. 확률바퀴 선별방법에서 각 개체가 선별될 확률은 그 개체의 평가함수 값을 모든 개체들의 평가함수 값의 합으로 나눈 값이 된다. 그러나 최소화 문제이거나 각 개체의 평가함수 값이 별로 차이가 없을 때에는 확률바퀴를 구성하기 위하여 척도(Scaling)를 만들어야 하는 어려움이 있다. 본 연구에서 다루는 문제는 최소화 문제이므로 다음과 같이 평가함수 값을 변환하여 사용하였다.

$$\text{변환된 평가함수값}(i) = \text{모집단의 최대값} + \text{모집단의 최소값} - \text{실제 평가함수값}(i)$$

토너먼트 선별방법[4]은 먼저 모집단의 개체를 임의의 순서로 나열한 후, 차례로 s 개씩 비교하여 가장 좋은 개체를 선별하는 방법으로 자세한 절차는 다음과 같다.

단계 1 : 토너먼트의 크기($s \geq 2$)를 결정한다.

단계 2 : 모집단의 개체들을 임의의 순열로 나열한다.

단계 3 : 나열된 순열에서 처음부터 차례로 s 개의 개체를 비교하여, 평가함수 값이 가장 좋은 개체를 선별한다.

단계 4 : 나열된 순열을 모두 사용했으면 임의의 순열로 다시 나열한다.

단계 2 : 모집단의 크기만큼 선별하였다면 종료하고, 그렇지 않으면 단계 3으로 간다.

토너먼트 선별방법은 평가함수 값의 척도를 재구성할 필요가 없으며, 토너먼트 크기(s)로 개체의 선별력(Selection pressure)을 조정할 수 있다.

순위 선별방법은 모집단의 모든 개체를 평가함수 값이 좋은 순서로 나열하여 순위에 따라 각 개체의 선별 확률을 선형 또는 비선형 함수를 사용하여 결정하는 방법이다[13]. 본 연구에서는 평가함수 값이 가장 좋은 개체부터 차례로 나열하여 순위를 1,2,3,...으로 주고, 각 개체의 선별 확률을 아래와 같은 기하분포를 사용하여 모집단을 재생산하기로 한다.

$$\text{Prob}(\text{rank}) = q(1 - q)^{\text{rank}-1}, \text{rank} = 1, 2, 3, \dots, \text{pop_size}$$

여기서 q 는 선별확률 파라미터로 $0 < q \leq 1$ 의 값을 갖는다.

본 연구에서는 유전 알고리즘의 성능에 큰 영향을 미치는 이들 선별방법에 대해 4장에서 실험을 통해 비교분석한다.

4. 실험 결과

자동차 조립라인에서 발생하는 가외작업을 최소로 하는 투입순서 결정에 적합한 유전 알고리즘을 제시하기 위하여, 선별방법들에 대해 해의 효율과 계산시간의 측면에서 비교분석하고자 한다. 유전 알고리즘은 C++ 언어로 프로그램되어 각 문제마다 초기해를 달리하여 20회 반복 실험하였다.

4.1 실험 예제

실험에 사용한 예제는 <표 1>과 같다. <표 1>은 생산할 제품 수를 3수준, 작업장의 수를 3수준, 준비시간의 분포를 2수준으로 분류한 18개의 문제를 나타내고 있다. 준비시간 분포에서 0은 준비시간이 없음을, $U(a, b)$ 는 $a \leq s_{jmr} \leq b$, 모든 $j, m \neq r$, 범위를 갖는 일양분포에서 준비시간을 발생하였음을 의미한다.

각 문제에서 모델의 수는 생산 제품 수의 1/2로 두었다. 즉 생산 제품 수가 20, 30, 40인 경우, 모델의 수는 각각 10, 15, 20이 된다. 또한 한 사이클 동안 생산 모델별 제품 수(MPS)는 모델의 개수가 10, 15, 20인 경우 각각 (2,1,3,1,4,3,1,3,1,1), (2,1,3,1,5,3,1,4,1,2,2,1,2,1,1), (3,1,4,3,5,2,1,2,1,2,1,2,1,2,2,1,1,3,2,1)로 두었다. j 작업장에서 m 모델에 대한 조립시간(t_{jm})은 $20 \leq t_{jm} \leq 40$, 모든 j, m , 범위를 갖는 일양분포에서 발생하였다. 각 작업장의 실제 길이는 각 작업장에서 생산하는 모델의 조립시간 중 최대의 값으로 두었으며, 컨베이어 속도는 계산의 편의를 위해 1로 두었다.

<표 1> 실험을 위한 문제의 설계

문제	생산제품의 수	작업장의 수	준비시간의 분포	문제	생산제품의 수	작업장의 수	준비시간의 분포
I - 1	20	10	0	II - 1	20	10	U(0,3)
I - 2	20	20	0	II - 2	20	20	U(0,3)
I - 3	20	30	0	II - 3	20	30	U(0,3)
I - 4	30	10	0	II - 4	30	10	U(0,3)
I - 5	30	20	0	II - 5	30	20	U(0,3)
I - 6	30	30	0	II - 6	30	30	U(0,3)
I - 7	40	10	0	II - 7	40	10	U(0,3)
I - 8	40	20	0	II - 8	40	20	U(0,3)
I - 9	40	30	0	II - 9	40	30	U(0,3)

4.2 선별방법의 비교분석

확률바퀴, 토너먼트, 그리고 순위 선별방법을 해의 효율과 계산시간의 측면에서 비교분석하고자 한다. 이를 위해 먼저 예제 문제에 적절한 유전 파라미터 값을 결정하였다. 많은 예비실험을 통해 모집단의 크기는 100으로, 연산자의 적용 개체 수(k)는 모집단 크기의 50%로, 교차변이와 돌연변이의 비율은 5 : 5로 결정하였다. 또한 유전 알고리즘의 수행은 200세대에서 종료하였으며, 해의 성능향상을 위해 한 세대에서 가장 좋은 개체가 다음 세대에서 항상 보존되는 우수개체(Elitism) 선별전략을 사용하였다. 이는 지금까지 탐색한 가장 좋은 해를 선별과정에서 잃어버리지 않게 방지한다.

<표 2>는 선별방법에 따른 해의 성능을 각 문제별로 나타낸다. 이때 토너먼트 선별방법에서는 토너먼트 크기를 3으로 두고, 순위 선별방법에서는 선별확률 파라미터(q)를 0.05로, 즉 각 개체의 평균 선별확률($1/pop_size$)의 5배로 두었다. 표에서 CV의 열은 표준편차를 평균치로 나눈 값을 백분율로 표시한 변동계수(Coefficient of variation)를 나타낸다.

<표 2> 선별방법에 대한 해의 탐색성능 비교

문제	확률바퀴 선별				토너먼트 선별				순위 선별			
	Worst	Average	Best	CV	Worst	Average	Best	CV	Worst	Average	Best	CV
I - 1	151	141.6	133	3.46	130	126.9	122*	1.81	135	128.8	122*	2.80
I - 2	531	521.9	504	1.28	500	495.0	490*	0.65	507	500.5	495	0.82
I - 3	619	601.4	585	1.66	581	569.0	560*	0.83	577	569.7	562	0.82
I - 4	284	274.3	256	2.44	246	236.8	228*	1.90	245	236.1	231	1.78
I - 5	682	659.2	627	1.85	618	602.0	589*	1.33	624	603.7	592	1.34
I - 6	1082	1052.3	1016	1.82	984	969.5	962	0.57	983	963.1	948*	0.88
I - 7	308	294.8	276	3.22	256	245.1	235	2.24	250	240.6	234*	2.12
I - 8	974	936.5	888	2.16	844	824.3	804	1.29	835	815.3	793*	1.41
I - 9	1165	1134.5	1083	1.98	1041	1020.2	1003	0.98	1032	1015.8	1002*	0.77
II - 1	166	159.1	152	2.64	148	142.9	137*	1.96	151	144.9	139	2.48
II - 2	600	589.1	568	1.41	568	557.2	548*	1.11	570	559.0	549	1.06
II - 3	705	689.5	674	1.31	658	644.2	631*	0.95	663	649.6	636	1.08
II - 4	320	305.2	293	2.33	271	262.7	247*	2.63	278	264.5	251	2.84
II - 5	776	763.8	738	1.28	698	681.7	669	1.10	706	681.0	661*	1.63
II - 6	1253	1220.7	1185	1.38	1140	1119.7	1102	0.93	1141	1118.9	1093*	1.13
II - 7	374	357.3	338	2.29	306	296.7	286	2.12	302	291.9	279*	2.12
II - 8	793	747.5	717	2.72	644	629.0	599	1.89	647	623.8	598*	2.15
II - 9	1393	1358.0	1297	1.84	1256	1215.1	1180	1.51	1239	1204.4	1169*	1.48

* : 문제별 가장 좋은 값

문제의 크기가 작은 경우는 토너먼트 선별방법이, 문제의 크기가 큰 경우는 순위 선별방법이 해의 질이 좋게 나타났으며, 모든 문제에서 확률바퀴 선별방법은 해의 성능이 좋지 않게 나타났다. 또한 변동계수가 토너먼트와 순위 선별방법이 확률바퀴 선별방법보다 낮게 나타났다. 확률바퀴 선별방법은 평가함수 값의 척도를 다른 방법으로 재구성해야 할 필요가 있으며, 토너먼트 선별 또는 순위 선별방법을 사용시 해의 질을 높이기 위해서는 문제의 크기에 따라 토너먼트 크기와 선별확률 파라미터를 조정할 필요가 있다.

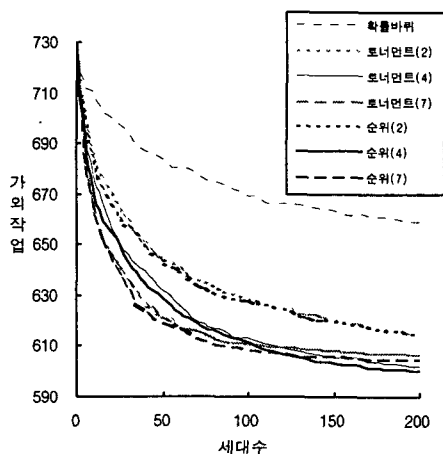
토너먼트 선별과 순위 선별에서 다루는 문제에 대해 적합한 토너먼트 크기(s), 선별확률 파라미터(q)를 찾기 위하여 s 와 q 를 변화시키면서 실험하였다. <표 3>은 s 를 2, 3, 4, 5, 6, 7로 변화시키면서, q 를 각 개체의 평균 선별확률($1/pop_size$)의 2, 3, 4, 5, 6, 7로 변화시키면서 20회 반복한 결과의 평균치를 나타내었다. 표에서 괄호 안의 값은 가장 좋은 탐색성능을 나타낸 s 와 q 를 나타내고 있다.

<표 3> 선별방법에 대한 해의 탐색성능 비교

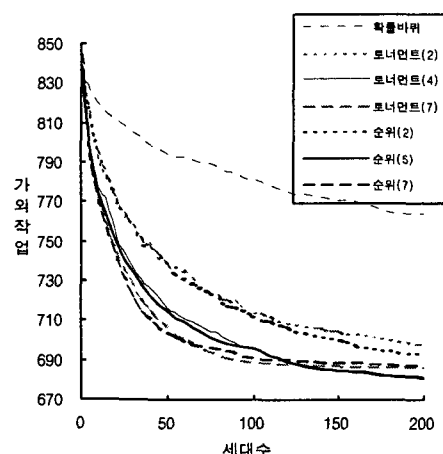
문제	토너먼트	순위	문제	토너먼트	순위
I - 1	126.9* (3)	127.4 (4)	II - 1	142.9* (3)	144.1 (3)
I - 2	495.0* (3)	496.7 (3)	II - 2	557.2 (3)	556.9* (4)
I - 3	569.0 (3)	568.7* (4)	II - 3	644.2 (3)	643.2* (3)
I - 4	234.8 (4)	233.9* (4)	II - 4	262.0 (5)	261.2* (4)
I - 5	602.0 (4)	600.0* (4)	II - 5	681.7 (4)	681.0* (5)
I - 6	960.8* (5)	963.1 (5)	II - 6	1118.8* (4)	1118.9 (5)
I - 7	239.9 (7)	239.6* (7)	II - 7	289.2* (5)	290.2 (4)
I - 8	812.8* (6)	815.3 (5)	II - 8	623.3 (6)	619.0* (6)
I - 9	1010.6 (7)	1009.2* (6)	II - 9	1204.9 (7)	1202.9* (7)

* : 문제별 가장 좋은 값

해의 질은 순위 선별방법이 토너먼트 선별방법보다 약간 좋게 나타났으며, 문제의 크기가 커질수록 토너먼트의 크기와 선별확률 파라미터가 커지는 경향이 있음을 알 수 있다.



(a) 문제 I-5



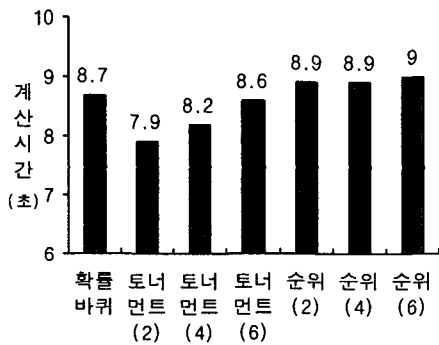
(b) 문제 II-5

<그림 2> 선별방법에 대한 해의 탐색성능 비교

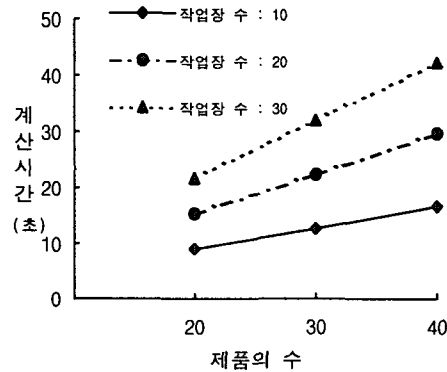
<그림 2>의 (a)와 (b)는 문제 I-5와 문제 II-5에서 각 선별방법에 대한 해의 탐색 성능을 나타내고 있다. 그림에서 괄호 안의 값은 토너먼트의 크기와 선별확률 파라미터를 나타낸다.

<그림 2>에서 토너먼트의 크기 또는 선별확률 파라미터 값이 커지면 좋은 해를 빨리 탐색 가능하나 조기 수렴할 확률이 커지고, 반면에 토너먼트의 크기 또는 선별확률 파라미터 값이 작아지면 해를 효율적으로 탐색하기가 힘들다는 것을 알 수 있다.

선별방법들을 계산시간의 측면에서 비교분석하였다. <그림 3>의 (a)는 문제 I-1에서 선별방법에 따른 유전 알고리즘의 계산시간을, (b)는 각 개체의 평균 선별확률의 5배인 순위 선별방법을 사용하였을 때, 제품의 수와 작업장 수에 따른 유전 알고리즘의 계산시간을 나타낸다.



(a) 선별방법의 변화



(b) 제품의 수와 작업장 수의 변화

<그림 3> 계산시간의 비교

다루는 문제에 있어서 토너먼트 선별방법을 적용한 유전 알고리즘이 계산시간이 비교적 작게 나타났으나, 각 선별방법에 따른 유전 알고리즘의 계산시간은 거의 비슷함을 알 수 있다. 또한 제품의 수와 작업장 수에 따라 계산시간의 변화가 상대적으로 크게 나타나, 유전 알고리즘의 계산시간은 각 개체의 평가에 대부분 소요됨을 알 수 있다.

5. 결론

본 연구에서는 자동차 조립라인에서 모델 변화에 따른 준비시간을 고려하여 총 가외작업을 최소로 하는 자동차 조립라인의 투입순서를 위한 수리모형을 제시하였다. 또한 유전 알고리즘을 사용하여 실시간으로 투입순서를 결정하였으며, 실험을 통해 유전 파라미터를 분석하여 적절한 값을 찾고, 확률바퀴 선별, 토너먼트 선별, 그리고 순위

선별방법을 비교하였다. 실험결과, 다루는 문제의 경우 순위 선별방법이 토너먼트 선별방법보다 해의 질이 약간 우수하게 나타났으며, 확률바퀴 선별방법은 적합한 척도를 만들기 어려워 해의 질이 가장 좋지 않게 나타났다. 또한 문제의 크기가 커질수록 토너먼트의 크기와 선별확률 파라미터가 커지는 경향이 있음을 알 수 있었다. 계산시간 측면에서는 토너먼트 선별방법을 적용한 유전 알고리즘이 계산시간이 작게 나타났으며, 제품의 수와 작업장 수에 따라 계산시간이 상대적으로 크게 나타났다.

6. 참 고 문 헌

- [1] 현철주, "자동차 조립라인에서 부품사용의 일정률 유지를 위한 투입순서 결정", 안전경영과학회지, 4권 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] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, 1989.
- [4] D. E. Goldberg, B. Korb and K. Deb, "Messy Genetic Algorithms : Motivation, Analysis, and First Results", *Complex Systems*, Vol.3, pp.493-530, 1989.
- [5] D. E. Goldberg and K. Deb, "Comparative Analysis of Selection Schemes Used in Genetic Algorithms", Foundation of Genetic Algorithms, Edited by Gregory J. E., Morgan Kaufmann publishers, pp.69-93, 1991
- [6] 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.
- [7] 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.
- [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] 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.
- [10] P. R. McMullen, "An efficient frontier approach to addressing JIT sequencing problems with setups via search heuristics", *Computers & IE*, pp. 335-353, 2001.
- [11] R. T. Sumichrast, 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.

- [12] 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.
- [13] 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.
- [14] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Second Editions, Springer-Verlag, 1994.

저 자 소 개

현 철 주 : 전남대학교에서 학사, 석사, 박사를 취득.

현재 정인대학 품질관리과 조교수로 재직중.

주요 관심분야는 최적화문제에 Genetic Algorithm과 Tabu Search의 응용 및 관련기법 개발 등.