

혼합모델 조립라인의 생산순서 결정을 위한 유전알고리즘*

Genetic Algorithms for Mixed Model Assembly Line Sequencing*

김여근**, 현철주**

Yeo Geun Kim** and Chul Ju Hyun**

Abstract

This paper considers the genetic algorithms(GAs) for the mixed model assembly line sequencing(MMALS) in which the objective is to minimize the overall line length.

To apply the GAs to the MMALS, the representation, selection, genetic sequencing operators, and genetic parameters are studied. Especially, the existing sequencing binary operators such as partially map crossover(PMX), cycle crossover(CX), and order crossover(OX) are modified to be suitable for the MMALS, and a new sequencing binary operator called immediate successor relationship crossover (ISR) is introduced. These binary operators mentioned above and/or unary operators such as swap, insertion, inversion, displacement, and splice are compared to find operators which work well in the MMALS.

Experimental results indicate that 1) among the binary operators ISR operator is the best, followed by the modified OX, and the modified PMX, with the modified CX being the worst, 2) among the unary operators inversion operator is the best, followed by displacement, swap, and insertion, with splice being the worst, and 3) in general, the unary operators perform better than the binary operators for the MMALS

* 이 논문은 1993년도 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

** 전남대학교 공과대학 산업공학과

1. 서론

혼합모델 조립라인은 공장부지, 시설 등의 투자에 따른 제약과 시설의 효율적인 이용을 위하여 한 라인에서 유사한 여러 모델의 제품을 생산하는 조립라인이다. 이러한 라인의 대표적인 예는 자동차 조립라인으로 승용차 라인, 중형차 라인, 대형버스 라인, 대형트럭 라인 등 생산제품의 특성과 크기에 따라 각 라인에서 유사한 여러 종류의 제품을 생산하고 있다.

혼합모델 조립라인에서는 재고비용을 줄이고 수요를 적시에 만족시키기 위하여 모델의 종류와 양을 평준화하여 골고루 혼류로 생산하는 사이클 생산방식을 택하는 경우가 많다. 사이클 생산방식은 모델별로 작업내용, 작업방법 및 작업시간, 자재, 이용하는 설비 등이 상이하여 생산계획과 통제가 복잡하게 된다. 이러한 복잡성 때문에 흔히 작업자를 과잉투입하거나, 작업장의 길이를 필요이상으로 길게 하는 경향이 있다. 라인의 효율적인 이용을 위해서는 모델의 생산순서(Sequencing)를 결정하는 문제가 중요하게 다루어 져야 한다.

사이클 생산에서 모델의 생산순서를 결정하는 데는 여러 목적이 있으나, 본 연구에서는 라인밸런싱되었거나 설비·치공구의 제약에 의해 모델별로 각 작업장의 작업시간이 주어진 경우, 전체 라인길이를 최소로 하는 생산순서를 결정하고자 한다. 이러한 생산순서의 결정은 공장부지 및 시설 등의 활용을 높일 수 있으며, 라인상의 유희작업 구간을 제거하여 생산 능력을 높일 수 있고, 설비의 배치계획수립을 가능하게 한다.

유전알고리즘(Genetic Algorithms)은 일종

의 인공지능 기법으로 자연의 진화과정을 모방하여 개발한 탐색 알고리즘(Search Algorithm)이다. 혼합모델 조립라인의 생산순서 결정문제는 일종의 순서결정문제(Sequencing Problem)이다. 유전알고리즘은 순서결정문제, 예로 외판원 문제(Traveling Salesman Problem), Flow-shop 문제에 적용하는데 적합한 기법으로 이에 관한 여러 연구[12, 15, 18, 28, 30, 34]가 이루어져 왔으나, 혼합모델 조립라인의 생산순서 결정에 응용한 연구는 아직 이루어지지 않았다.

유전알고리즘의 성능은 유전알고리즘의 표현(Representation)과 생존(Selection), 교차변이(Crossover)와 돌연변이(Mutation)와 같은 유전연산자(Genetic Operators), 교차변이율, 돌연변이율, 모집단의 크기 등의 유전파라미터(Genetic Parameters)에 따라 영향을 크게 받는다[6, 14, 16].

본 연구의 목적은 사이클생산을 하는 혼합모델 조립라인에서 전체 라인길이를 최소로 하는 생산순서를 결정하는 유전알고리즘을 개발하는 데 있다. 즉 이러한 생산순서 결정 문제에 유전알고리즘을 적용하기 위하여 적합한 유전알고리즘의 표현방법을 찾고, 연산자들을 개발하며, 실험을 통하여 연산자와 파라미터를 비교분석하는 데 있다.

혼합모델 조립라인의 생산순서에 관한 연구는 Kilbridge와 Webster[21]에 의해 1963년 처음으로 발표된 이후, 여러 연구가 이루어졌다. Thomopoulos[32]는 라인에서 작업자의 비효율성(예로 유희시간, 작업혼잡등)을 벌과금(Penalty)으로 환산하여 임의의 생산순서를 한정된 수 만큼 나열한 후 그 중에서 가장 낮은 벌과금을 갖는 순서를 선택하는 방

법을 제시하고 있다. Okamura와 Yamashina [27]는 컨베이어의 정지위험을 최소로 하는 생산순서에 관한 연구로, 모델의 생산순서를 상호 교환하거나 삽입하는 방법을 제시하고 있다. Dar-EI과 Cother[9]의 전체라인 길이를 최소로 하는 생산순서를 결정하는 알고리즘은 라인길이를 조금씩 증가시키면서 매번 새로이 순서를 구하고 있어 그 절차가 복잡하고 효과적이지 못하다. Dar-EI과 Bard, Shtub [5]는 생산순서를 결정하는데 있어서 투입규칙(Launching Discipline), 작업장 제약 등의 여러 요인들을 고려하여 라인길이를 최소로 하는 생산순서와 완성시간(Throughput Time)을 최소로 하는 생산순서에 대하여 각각 수리 모형을 제시하였다. 그러나 제시된 모형은 NP-Hard 문제로서 모델과 작업장의 수가 많아지면 실제 문제를 해결하기가 어렵게 된다.

그밖에 Burns와 Daganzo[7]는 작업준비시간과 생산능력을 고려하여 생산 순서를 결정하는 방법을 제시하였으며, MacCormick[22]은 작업장 사이에 완충재고(Buffer Stock)가 있는 경우, 사이클 타임을 최소로 하는 생산순서를 결정하는 수리 모형과 발견적 기법을 제시하고 있다.

앞에서 언급한 라인길이를 최소로 하는 생산순서에 관한 기존의 여러 수리모형은 NP-Hard 문제로 실제 문제에 적용하기 어렵고, 정확한 절차를 제시한 발견적 기법은 Dar-EI과 Cother[9]만이 제시하고 있으나 효율적이지 못하다.

유전알고리즘은 Holland[17]에 의해 연구되기 시작하여 여러 분야에서 활발히 연구되어 왔으며, 특히 조합최적화 문제(Combinato-

rial Optimization Problems)를 해결하는 데 적합한 기법으로 알려져 있다. 이 기법을 외판원문제에 적용하기 위하여 외판원문제의 특성에 맞는 유전알고리즘의 표현방법과 생존방법, 유전연산자인 교차변이, 돌연변이 등에 관하여 여러 연구[12, 20, 28, 30]가 행하여졌다. 또한 외판원문제와는 별개로 모집단의 크기, 교차율, 돌연변이율, 생존 전략 등의 유전과라미터에 관한 연구[6, 16, 29]가 이루어졌다.

본 연구에서는 혼합모델 조립라인의 생산순서문제에 적합한 유전알고리즘, 그에 따른 표현방법 및 모집단을 재생산하기 위한 생존방법에 관하여 연구하고 본 문제에 적합한 유전연산자를 개발한다. 또한 유전과라미터의 비교분석을 통해 최적의 유전과라미터 값을 찾고, 개발된 유전연산자의 성능을 비교분석하고자 한다.

본 연구의 구성은 아래와 같다. 제 2장에서 혼합모델 조립라인의 길이를 구하는 방법을 제시하고, 제 3 장에서는 사용하고자 하는 유전알고리즘과 그에 따른 요소들을 다룬다. 그리고 제 4장에서는 적합한 유전연산자를 제시하고, 제 5장에서는 실험을 통하여 유전과라미터 값과 유전연산자를 비교분석하며, 제 6장은 결론으로 구성되어 있다.

2. 혼합모델 조립라인의 길이

본 연구에서 다루는 혼합모델 조립라인은 일정한 속도로 이동하는 컨베이어 시스템으로 유사한 여러모델의 제품들이 일정시간 간격으로 투입(Fixed Rate Launching)되고 있으며 컨베이어의 이동에 따라 작업자가 이동하

면서 작업하는 라인이다. 라인의 길이는 컨베이어의 속도와 제품의 생산순서에 의해 결정되는데, 라인상의 제품간의 거리를 제품투입 시간간격으로 나누면 컨베이어의 속도가 된다. 본 연구에서는 작업장의 라인길이 계산의 편의를 위해 컨베이어의 속도를 단위시간당 단위거리로 가정하고, 작업자가 다음제품의 작업을 하기 위하여 돌아오는 시간은 무시하기로 한다. 그러면 제품투입 시간간격과 라인상에 제품간의 간격이 같게 되고, 또한 작업자가 작업하는 시간과 작업자의 위치변동이 같게 된다.

작업장은 폐쇄 작업장과 개방 작업장의 경우를 모두 다루고자 한다. 두가지 형태의 작업장 모두 인접 작업자와의 간섭이 발생되어서는 안된다고 본다. 폐쇄 작업장이란 도장 작업장, 열처리 작업장 등과 같이 각 작업장의 경계가 존재하는 작업장으로, 작업자연으로 후행 작업장에서 작업을 하거나, 선행 작업장에서 미리 작업을 하는 것이 불가능하며, 주어진 작업장내에서만 작업을 해야 하는 작업장이다. 개방 작업장은 작업장의 경계가 명확히 구분되지 않고, 앞뒤 작업장의 공유가 가능한 작업장이다.

그림 2.1과 그림 2.2는 폐쇄작업장과 개방 작업장에서 작업시간과 그에 따른 작업장의 소요라인길이를 각각 나타내고 있다. 그림에서 A, C, A, A, B, B 모델순으로 생산하는 경우 실선은 작업자의 위치변동(작업시간)을 나타내며, 점선은 제품간의 간격(투입 시간간격)을 나타낸다.

2.1 폐쇄작업장의 라인길이

폐쇄작업장의 라인길이는 그림 2.1과 같이

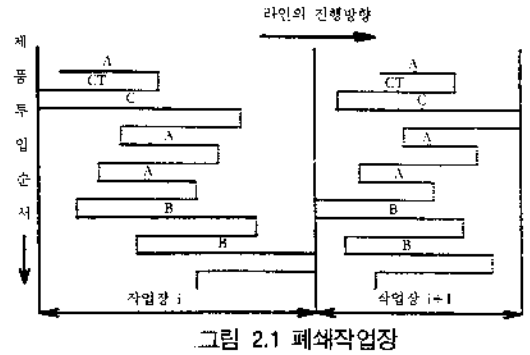


그림 2.1 폐쇄작업장

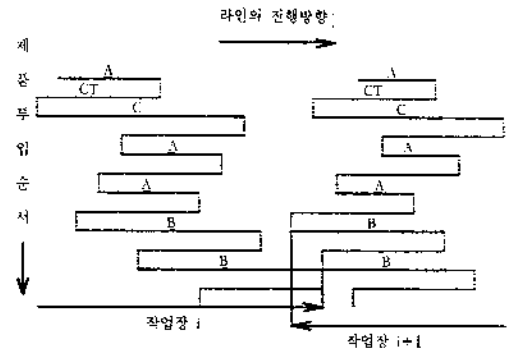


그림 2.2 개방작업장

각 작업장길이의 합으로 구해진다. 작업장은 M개($i=1, \dots, M$), 생산제품의 모델은 $N(j=1, \dots, N)$ 종류, 한 사이클동안 생산되는 제품의 총 수는 k_{max} 개, 그리고 작업장 i 에서 모델 j 에 대한 작업시간을 $T(i,j)$ 이라 하자.

작업장 i 에서 첫번째 제품의 작업시작점, $DS(i,1)$ 는

$$DS(i,1) = 0, i=1, \dots, M \dots \dots \dots (2.1)$$

로 둔다. 작업장 i 에서 k 번째 제품의 작업완료점, $DT(i,k)$ 는 작업시작 시각에 작업시간을 더한 것으로

$$DT(i,k) = DS(i,k) + T(i,N_k) \dots (2.2)$$

가 된다. 여기서 N_k 은 라인에 투입되는 k 번째 제품의 모델이다.

또한 작업장 i 에서 $k+1$ 번째 제품의 작업

시작점, $DS(i,k+1)$ 는 직전작업의 완료점에서 제품간의 투입시간간격을 뺀 것으로

$$DS(i,k+1) = DT(i,k) - CT \dots\dots (2.3)$$

가 된다. 여기서 CT는 제품간의 투입 시간간격(사이클 타임)으로 컨베이어 속도를 단위 시간당 단위거리로 가정함으로써 컨베이어에서 제품간의 간격이 된다.

따라서 작업장 i에서 라인이동 반대방향쪽으로 가장 멀리 위치한 점, $DMIN(i)$ 은

$$DMIN(i) = \text{Min} \{DS(i,k), k=1,\dots,k_{\text{max}}\} \dots\dots (2.4)$$

가 되고, 작업장 i에서 라인이동방향쪽으로 가장 멀리 위치한 점, $DMAX(i)$ 는

$$DMAX(i) = \text{Max} \{DT(i,k), k=1,\dots,k_{\text{max}}\} \dots\dots (2.5)$$

가 된다.

그러므로 작업장 i의 라인길이는 $DMAX(i) - DMIN(i)$ 이 되고, 전체 조립라인길이, SLONG은

$$SLONG = \sum_{i=1}^M (DMAX(i) - DMIN(i)) \dots\dots (2.6)$$

로 된다.

2.2 개방작업장의 라인길이

개방작업장에서는 그림 2.2와 같이 작업에 간섭이 되지 않는 한, 선후작업장을 공유할 수 있으므로 작업장간의 공유공간을 고려하여 전체 라인길이를 구해야 한다.

원점에서부터 i작업장까지의 길이를 DLNT(i), $i=1,\dots,M$ 이라고 하자. 개방작업장에서는 폐쇄작업장과 달리 각 작업장의 첫번째 제품의 시작점, $DS(i,1)$ 는

$$DS(i,1) = DLNT(i-1), i=1,\dots,M \dots (2.7)$$

로 둔다. 여기서 $DLNT(0) = 0$ 이다.

원점에서부터 첫번째 작업장까지의 길이, DLNT(1)는 식(2.4)와 (2.5)에 의해

$$DLNT(1) = DMAX(1) - DMIN(1) \dots\dots (2.8)$$

가 된다. 두번째 작업장에서 첫번째 제품의 시작점, $DS(2,1)$ 를 DLNT(1)로 두고 식(2.2)와 (2.3)에 의해 $DT(1,k)$ 와 $DS(2,k)$ 를 구하면 작업장 1의 작업에 간섭되지 않고 작업장 2의 작업들을 가장 많이 당길 수 있는 거리, DELTA(2)는

$$DELTA(2) = \text{Max} \{(DT(1,k) - DS(2,k)), k=1,\dots,k_{\text{max}}\} \dots\dots (2.9)$$

가 된다. 원점에서부터 두번째 작업장까지의 거리, DLNT(2)는 $DS(2)$ 를 DLNT(1)로 두고 식 (2.2)와 식 (2.5)로부터 구한 DMAX(2)에서 작업장 2의 작업들을 작업장 1의 작업에 간섭되지 않고 가장 많이 당길 수 있는 거리만큼 이동시킨 것이다. 즉,

$$DLNT(2) = DMAX(2) + DELTA(2) \dots (2.10)$$

가 된다. 작업장 (i-1)의 작업에 간섭되지 않고 작업장 i의 작업들을 가장 많이 당길 수 있는 거리, DELTA(i)는 식 (2.9)에서와 같이

$$DELTA(i) = \text{Max} \{(DT(i-1,k) - DS(i,k)), k=1,\dots,k_{\text{max}}\} \dots\dots (2.11)$$

로 되고, 원점에서부터 i 작업장까지의 거리, DLNT(i)는 식 (2.10)에서와 같이

$$DLNT(i) = DMAX(i) + DELTA(i) \dots\dots (2.12)$$

로 된다.

개방작업장의 총 조립라인길이(SLONG)는 원점에서부터 마지막 작업장까지의 길이, DLNT(M)과 같으므로

$SLONG = DLNT(M) = DMAX(M) + DELTA(M) \dots\dots\dots (2.13)$
 로 된다.

3. 유전 알고리즘

유전알고리즘을 적용하기 위해서는 먼저 문제에 대한 특성을 분석하여 적합한 유전알고리즘을 선택하고, 그에 따른 표현방법, 평가함수, 모집단의 초기화방법과 생존방법, 유전연산자 및 유전파라미터의 결정 등이 이루어져야 한다.

앞에서 언급했듯이 혼합모델 조립라인에서 사이클생산에 따른 생산순서를 결정하는 문제는 일종의 순서문제이나 한 사이클내에서 같은 모델의 제품이 생산될 수 있으므로, 예로 한 사이클동안의 생산제품이 모델 A, B, C가 각각 2, 3, 4개 일때 생산순서가 A, C, B, C, B, A, C, C, B인 경우 한 사이클내에서 같은 모델이 다시 나타나므로, 대표적인 순서문제인 외판원문제와는 특성이 다르게 나타난다.

본 장에서는 혼합모델 조립라인의 생산순서문제를 위한 유전알고리즘을 개발하기 위하여 사용되는 유전알고리즘 방법과 그 방법에 따른 표현, 모집단의 초기화, 평가함수, 생존, 유전연산자 및 유전 파라미터 등을 다루고자 한다.

3.1 수정 유전알고리즘(modified Genetic Algorithm : modGA)

Holland[17]에 의해 최초로 제시된 유전알고리즘(이후 '고전적 유전알고리즘'이라 부르기로 함)으로 부터 해의 성능 및 계산시간의

단축을 위해 이를 변형한 여러 유전알고리즘이 개발되었다. 본 연구에서는 Michalewicz et al.[24]가 제시한 수정 유전알고리즘을 기본으로 하고자 한다. 수정 유전알고리즘의 절차를 단계별로 정리하면 다음과 같다.

- 단계 1. 세대(Generation)를 0 으로 둔다. $t = 0$.
- 단계 2. 초기모집단(P(0))을 생산한다.
- 단계 3. 모집단의 각 개체(Chromosome, Individual)를 평가한다.
- 단계 4. 세대를 1 증가 시킨다. $t=t+1$.
- 단계 5. P(t-1)에서 (모집단의 크기-R)개의 서로 다른 개체를 중복되지 않게 선택하여 P(t)에 그대로 복사한다.
- 단계 6. P(t-1)에서 R개의 개체를 중복에 상관없이 선택한다.
- 단계 7. 단계 6.에서 선택된 R개의 개체에 교차변이 또는 돌연변이를 시켜 새로운 R개의 개체를 생산한다.
- 단계 8. 새로 생산된 R개의 개체를 P(t)에 추가한다.
- 단계 9. 모집단의 각 개체를 평가한다.
- 단계 10. 종료기준이 만족하면 알고리즘의 수행을 끝낸다. 그렇지 않으면 단계 4.로 간다.

고전적 유전알고리즘과 수정 유전알고리즘과의 중요한 차이점은 다음과 같다. 고전적 유전알고리즘에서는 P(t-1)에서 모집단 크기(N)만큼 선택(중복가능)하여 P(t)로 하고, P(t)를 교차변이와 돌연변이를 사용하여 재결합하는데 반하여 수정 유전알고리즘에서는

위의 단계 5.와 단계 6.과 같이 선택함으로써 높은 평가함수값을 갖는 초개체(super Individual)가 다음 세대에 여러개 나타나는 것을 방지함과 동시에 상대적으로 우성형질을 갖는(평가함수값이 좋은) 개체가 P(t)에서 생존가능성이 높아지며(단계 5.), 이 개체들이 다른 개체와 재결합하여 우성형질을 유전시킬 확률이 높게 된다(단계 6.). 또한 고전적 유전알고리즘에서는 돌연변이가 유전인자(Gene) 단위로 처리되어 개체의 인자 수에 따라 돌연변이율을 결정해야 하는 어려움이 있음에 반하여, 수정 유전알고리즘에서는 위의 단계 7.과 같이 개체단위로 처리되어 선택된 R개의 개체중에서 몇개의 개체는 교차변이, 나머지 개체는 돌연변이를 사용하여 자손을 생산함으로써 돌연변이율은 교차변이율과의 비율로 쉽게 구할 수 있다.

3.2 표현방법

유전알고리즘에서 표현방법은 연산자를 사용하여 가능해 공간을 효율적으로 탐색할 수 있도록 표현되어야 한다. 흔히 많은 문제에서 이진수를 사용하여 개체를 표현하고 있으나 순서문제는 이진수로 표현하기가 어렵고, 연산시에 가능해 유지의 어려움때문에 부적합한 것으로 알려져 있다[23, 34].

본 연구에서는 가장 자연스런 표현방법으로 사이클 생산내에서 생산되어야 하는 모델들의 목록을 생산순서대로 나열하는 방법을 사용하고자 한다. 예로 사이클 생산동안 모델 A, B, C가 각각 2, 3, 4개 생산되어야 한다고 하자. 그리고 하나의 가능해로 생산순서가 A, C, B, C, B, A, C, C, B라면 개체의 표현방법은 모델의 생산순서대로 (A C B C

B A C C B)와 같이 표현한다. 같은 모델이라도 다른모델인 것처럼 9가지의 다른인자로 (A1 C1 B1 C2 B2 A2 C3 C4 B3)와 같이 표현하면 외판원문제의 경로(Path) 표현방법을 그대로 사용할 수 있으나 연산자를 사용하여 연산을 할때, 같은 모델이 다른 모델처럼 취급되어(예로 A1과 A2의 교환은 같은 결과를 가져옴) 해공간을 효율적으로 탐색할 수 없다.

따라서 각 개체의 인자(Gene)에 관한 정확한 정보를 갖을 수 있도록 생산순서에 따라 모델을 차례로 나열하는 방법을 사용하기로 한다.

3.3 초기화 및 평가함수

초기 모집단의 개체들은 해공간을 가능한 넓게 탐색할 수 있도록 생성되어야 한다. 초기모집단을 만드는 방법에는 탐색적절차를 사용하는 방법과 임의의 가능해를 사용하는 방법이 있으나, 본 연구에서는 사이클생산동안 각 모델의 생산량을 만족하는 임의의 해를 사용하기로 한다.

유전알고리즘에서는 생존에 대한 적응척도로서 평가함수가 사용된다. 본 문제에서 평가함수는 제품모델의 생산순서에 따른 라인 길이를 나타내며, 폐쇄작업장 및 개방작업장의 라인길이는 각각 제 2장의 식 (2.6)과 식 (2.13)에 의해 구해진다.

3.4 생존(Selection)

생존은 자연현상의 적자생존의 법칙을 적용하여 생존의 적응도, 즉 평가함수값에 비례하여 생존하는 Roulette Wheel방법[4]을 흔히 사용하고 있다. 이 방법에서 각 개체의 생

존 확률은 그 개체의 평가함수값을 모든 개체들의 평가함수값의 합으로 나눈 값이 된다. 그러나 본 문제와 같이 최소화문제이거나 각 개체의 평가함수값이 별로 차이가 없을 때에는 Roulette Wheel을 구성하기 위하여 척도 (scaling)를 만들어야 하는 어려움이 있다.

Baker[3]가 제시한 순위생존(Rank Selection)방법은 평가함수값이 가장 좋은 개체부터 차례로 나열하여 순위에 따라 각 개체가 생존할 확률을 선형 또는 비선형함수로 나타내므로 평가함수값의 변환이 필요 없다는 장점을 갖고 있다. 본 연구에서는 평가함수값이 가장 좋은 개체부터 차례로 나열하여 순위를 1,2,3,...으로 주고, 각 개체가 생존할 확률을 아래와 같은 기하분포를 사용하여 모집단을 재생산하기로 한다.

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

여기서 q 는 생존확률파라미터로 $0 < q < 1$ 의 값을 갖는다.

3.5 유전연산자와 유전파라미터

유전연산자는 유전알고리즘의 성능에 크게 영향을 미치는 구성요소로서 연산자는 문제의 특성을 잘 반영하여 좋은 해를 효율적으로 탐색해 갈 수 있도록 개발되어야 한다[12, 30]. 본 연구에서는 문제에 적합한 유전연산자를 개발하고, 이 유전연산자를 실험을 통해 비교분석한다. 이에 관해서는 4장에서 구체적으로 다룬다.

유전파라미터의 값들은 해공간을 넓게 탐색할 수 있는 기능(exploration)과 좋은 해를 효율적으로 찾아나가는 기능(exploitation)이 조화될 수 있도록 적절히 결정되어야 한다. 수정 유전알고리즘에서 결정해야 할 파라메

터는 모집단의 크기(N), 연산자 적용비율(R), 교차변이와 돌연변이의 비율, 종료조건과 순위생존방법의 생존확률파라미터(q)가 있다.

이진표현으로 된 문제에 대해서는 이들 파라미터의 최적결정에 대한 연구가 많이 제시되어 있으나 순서문제는 이진표현이 어렵기 때문에 그대로 적용시키기 어렵다. 본 연구에서는 파라미터에 대한 기존의 연구결과를 토대로, 실험을 실시하고 결과분석을 통해 파라미터값을 결정하기로 한다. 이는 5장에서 구체적으로 다룬다.

4. 유전연산자

유전연산자는 문제의 특성을 잘 반영할 수 있는 정보를 부모로부터 효율적으로 추출하여 자손을 생산할 수 있어야 하며, 연산은 다루는 문제의 가능해가 유지되도록 해야 한다. 유전알고리즘에서 연산자는 크게 교차변이와 돌연변이로 나눌 수 있으나 순서문제에서는 교차변이와 같이 두개의 부모가 지닌 정보를 이용하여 자손을 생산하는 이원연산자(Binary Operators)와 돌연변이 대신 하나의 부모가 지닌 정보만을 이용하여 하나의 자손을 생산하는 일원연산자(Unary Operators)로 나누고 있다.

이원연산자에는 부분사상교차변이(Partially Map Crossover : PMX), 순환교차변이(Cycle Crossover : CX), 순서교차변이(Order Crossover : OX) 등이 있는데, 이들 연산자를 본 문제에 적용할 수 있도록 수정하고, 새로 개발한 이원연산자로 직후행관계교차변이(Immediate Successor Relationship Crossover :

ISR)를 제시한다. 일원연산자로는 교환(Swap), 삽입(Insertion), 역순(Inversion), 轉位(Displacement), 접합(Splice)연산자[12, 23]를 사용한다.

연산자설명에 용이성을 위해 부모1, 부모2, 자손1, 자손2를 각각 P1, P2, O1, O2로 쓰기로 하고, O2의 생산절차는 O1의 생산절차에서 P1과 P2를 바꾸면 되므로 O1의 생산절차를 설명하기로 한다.

4.1 이원 연산자

4.1.1 수정 순서교차변이

수정 순서교차변이는 Davis[11]가 제시한 OX를 본 문제에 적합하게 수정한 연산자이다. P1과 P2에서 2개의 교차점을 임의 선택하여 P1의 교차점사이에 있는 인자들을 그대로 O1에 유전시킨 후에, 유전된 인자들을 P2에서 임의로 지우고 남은 인자들을 순서대로 O1의 아직 나타나지 않은 인자에 유전하는 방법이다.

예로 그림 4.1과 같이 교차점이 임의로 선택되었다고 하자(그림에서 | 는 교차점을 의미한다). P1의 교차점사이에 있는 인자들을 O1에 그대로 유전하면 $O1 = (x\ x\ x\ B\ B\ C\ x\ x\ x)$ 가 된다. 여기서 기호 x는 현재상태에서 모르는 유전인자를 의미한다. O1에 유전된 인자를 P2에서 임의로 제거($P2[2]=B, P2[5]=B, P2[3]=C$)하면 $P2 = (A\ x\ x\ A\ x\ C\ B\ C\ C)$ 가 된다. 이를 순서대로 O1의 아직 나타나지 않은 인자에 순서대로 삽입하면 $O1 = (A\ A\ C\ B\ B\ C\ B\ C\ C)$ 가 된다.

수정 순서교차변이는 인자들의 상대적순서를 보다 잘 유지시킬 수 있도록 OX를 변형

하였다.

$$\text{부모 1 (P1)} = (A\ A\ B\ | \ B\ B\ C\ | \ C\ C\ C)$$

$$\text{부모 2 (P2)} = (A\ B\ C\ | \ A\ B\ C\ | \ B\ C\ C)$$

$$\text{자손 1 (O1)} = (A\ A\ C\ B\ B\ C\ B\ C\ C)$$

$$\text{자손 2 (O2)} = (A\ B\ B\ A\ B\ C\ C\ C\ C)$$

그림 4.1 수정 순서교차변이

4.1.2 수정 부분사상교차변이

수정 부분사상교차변이는 Goldberg와 Lingte[15]이 제시한 PMX를 본 문제에 적합하게 수정한 연산자이다. P1과 P2에서 2개의 교차점을 임의선택하여 P1의 교차점사이에 있는 인자들을 그대로 O1에 유전시킨 후에, P1과 P2의 교차점사이에 있는 인자들을 차례로 비교하여 같은 인자끼리 대응되도록 P2의 인자를 이동시킨다. 같은 인자끼리 대응되지 않은 P1의 인자들을 P2의 교차점밖에서 임의 선택하여 P2의 인자로 치환한 후, O1의 그 위치에 치환된 인자를 유전한다. 즉 P1의 인자 X가 P2의 인자 Y에 대응되었다면 P2에서 교차점밖에 있는 임의의 X를 선택하여 Y로 치환하고 O1의 선택된 X의 위치에 Y를 유전한다. 다음에 O1의 아직 나타나지 않은 인자들은 P2의 그 위치에 있는 인자를 그대로 유전한다.

예로 그림 4.2와 같이 교차점이 임의선택되었다고 하면 P1의 교차점사이에 있는 인자들을 O1에 그대로 유전하면 $O1 = (x\ x\ x\ B\ B\ C\ C\ x\ x)$ 가 된다. P1과 P2의 교차점사이에 있는 인자들을 차례대로 비교하여 같은 인자끼리 대응되도록 P2의 인자를 이동시키면 $P2[4]=B, P2[5]=B, P2[6]=C, P2[7]=$

A가 된다. 교차점 사이에 있으면서 같은 인자끼리 대응되지 않은 인자들은 $P1[7]=C$, $P2[7]=A$ 이므로 P2에서 교차점밖에 있는 C를 임의 선택($P2[9]$)하여 A로 치환하고, 그 위치로 유전하면 $O1 = (x x x B B C C x A)$ 가 된다. O1의 아직 나타나지 않은 인자들은 P2의 그 위치에 있는 인자를 유전하면 $O1 = (A B C B B C C C A)$ 가 된다.

수정 부분사상교차변이는 부모에서 각 인자들의 위치와 순서의 정보를 이용하여 자손을 생산하는 순서유전자이다.

- 부모 1 (P1) = (A A B | B B C C | C C)
- 부모 2 (P2) = (A B C | A B C B | C C)
- 자손 1 (O1) = (A B C B B C C C A)
- 자손 2 (O2) = (C A B A B C B C C)

그림 4.2 수정 부분사상교차변이

4.1.3 수정 순환교차변이

수정 순환교차변이는 Oliver et al.[28]가 제시한 CX를 본 문제에 적합하게 수정한 연산자이다. 사이클시작점으로 P1의 임의위치에 있는 인자를 선택하여 선택된 인자를 O1의 그 위치에 유전한다. 유전된 인자를 P2에서 임의선택하여 제거하고, 제거된 위치에 있는 P1의 인자를 O1의 그 위치에 유전한다. 이와 같이 반복하다가 유전할 인자의 위치가 사이클시작점과 같아지면, O1의 아직 나타나지 않은 인자들은 P2의 그 위치에 있는 인자를 유전한다.

예로 그림 4.3에서 사이클 시작점이 $P1[7]$ 로 임의로 선택되었다고 하자. 이 인자를 O1[7]에 유전하면 $O1[7] = P1[7] = C$ 가 된

다. P2에서 C를 임의로 선택($P2[3]$)하여 지우고, 그 위치에 있는 P1의 인자를 O1에 유전하면 $O1[3] = P1[3] = B$ 가 된다. 다시 P2에서 B를 임의로 선택($P2[5]$)하여 지우고, 그 위치에 있는 P1을 유전하면 $O1[5] = P1[5] = B$ 가 된다. 다시 P2에서 B를 임의로 선택하였을 때, $P2[7]=B$ 가 선택되었다면 사이클 시작점(7th)과 같아진다. 이때 O1에 아직 나타나지 않은 인자는 P2의 그 위치에 있는 인자를 유전하면 $O1[1] = P2[1] = A$, $O1[2] = P2[2] = B$, $O1[4] = P2[4] = A$, $O1[6] = P2[6] = C$, $O1[8] = P2[8] = C$, $O1[9] = P2[9] = C$ 가 된다. O2는 사이클시작점을 O1과 같게 하고 P1과 P2를 서로 바꾸어 위의 절차를 따른다.

수정 순환교차변이로 생산된 자손은 항상 두 부모중 한 부모의 인자위치를 유전받게 되어 위치에 대한 정보가 수정 부분사상교차변이보다 더 강조된다.

- 부모 1 (P1) = (A A B B B C C C C)
- 부모 2 (P2) = (A B C A B C B C C)
- 자손 1 (O1) = (A B B A B C C C C)
- 자손 2 (O2) = (A A C B B C B C C)

그림 4.3 수정 순환교차변이

4.1.4 구조교차변이

구조교차변이는 P1에서 인자를 임의로 선택한 후, 선택된 인자와 같은 인자들을 그대로 P2에 유전하여 O1를 생산한다[23]. 선택된 인자와 같은 인자이면서 유전되지 않았던 인자를 O1에서 지운 후에, 유전에 의해 지워진 인자들을 O1에 임의로 삽입한다.

예로 그림 4.4에서 P1에서 B인자가 임의로 선택되었다고 하자. P1에 있는 B인자들을 P2에 유전하여 O1를 생산하면 O1=(A B B B B C B C C)가 된다. O1에서 유전되지 않은 B들(O1[2]=B, O1[7]=B)을 지우면 O1=(A x B B B C x C C)가 된다. P2에서 유전에 의해 없어진 인자들(P2[3]=C, P2[4]=A)을 지워진 위치에 임의로 삽입하면 O1=(A C B B B C A C C)가 된다. O2는 최초의 임의선택인자를 O1과 같이 B로 하고 P1과 P2를 서로 바꾸어 위의 절차를 따른다.

구조교차변이는 부모에서 선택된 인자의 위치에 관한 정보를 이용하여 자손을 생산하는 연산자이다.

- 부모 1 (P1) = (A A B B B C C C C)
- 부모 2 (P2) = (A B C A B C B C C)
- 자손 1 (O1) = (A C B B B C A C C)
- 자손 2 (O2) = (A B C A B C B C C)

그림 4.4 구조교차변이

4.1.5 직후행관계 교차변이(Immediate Successor Relationship Crossover : ISR)

ISR 교차변이는 본 연구에서 새로 개발한 유전연산자로서 다른 연산자와 달리, 부모에서 각 인자들의 위치나 순서에 관한 정보 대신 인자별 직후행관계의 정보를 이용하여 자손을 생산하는 연산자이다. 이를 위해 부모에서 각 인자에 따른 직후행인자와 개체를 구성하고 있는 인자의 개수를 인자비율로 표시한 ISR표를 사용한다. 예로, 그림 4.5의 P1과 P2로 각 인자별 직후행관계의 밀접도를 나타내는 ISR표를 만들면, A인자(P1[1], P1

[2], P2[1], P2[4])의 직후행인자는 각각 P1[2]=A, P1[3]=B, P2[2]=B, P2[5]=B이고, B인자들의 직후행 인자들은 B, B, C, C, C, C이다. 또한 C인자들의 직후행인자들은 C, C, C, A, A, B, C, A가 되고 인자 A, B, C의 개수가 각각 2, 3, 4개 이므로 인자비율은 2, 3, 4가 되어 표 4.1과 같은 ISR표가 만들어진다.

표 4.1 ISR 표

인자	직후행인자	인자비율
A	A B B B	2
B	B B C C C C	3
C	C C C A A B C A	4

위와 같은 ISR표를 이용한 ISR 교차변이의 자손생산과정을 단계별로 표시하면 다음과 같다.

- 단계 1. P1과 P2로부터 각 인자별 직후행관계와 인자비율을 나타낸 ISR표를 만든다.
- 단계 2. 임의의 인자를 선택한다.
- 단계 3. 선택된 인자를 자손에 유전한다.
- 단계 4. ISR표에서 선택된 인자 2개를 임의로 삭제한다.
- 단계 5. ISR표에서 선택된 인자에 남아 있는 직후행 인자중에서 인자의 수가 가장 많이 나타난 인자를 선택한다. 만일 최대인자가 둘 이상이면 그 인자들의 ISR표에서 인자비율을 보정한 후 남아있는 인자의 수가 작은 것을 선택한다. 이것도 동일한 경우에는 최대인자중 하나를 임의로 선택한다.
- 단계 6. 모든 인자들이 유전될때까지 단계

3.부터 단계 6.까지를 반복한다.

위의 방법에서, ISR표는 P1과 P2의 직후행 인자를 표현하였기 때문에 단계 4.에서 인자 2개를 임의로 삭제하게 된다. 또한 단계 5.와 같이 인자를 선택하여 유전함으로써 부모의 인자들과 직후행관계가 밀접한 자손이 생산된다.

예로 그림 4.5의 P1과 P2로 표 4.1과 같은 ISR표를 만든 후에, 시작인자가 B로 임의선택되었다고 하자. 선택된 인자 B를 자손에 유전하면 O1[1]=B가 된다. B가 사용되었으므로 ISR표에서 2개의 B를 임의로 삭제한다. ISR표의 인자 A와 B에 있는 B가 삭제되었다고 하면, 선택된 B인자와의 직후행 관계는 B 인자의 ISR표에서 B가 1개, C가 4개이므로 C가 가장 밀접하다. C를 선택한 후, 유전하면 O1[2]=C가 된다. 다시 ISR표에서 2개의 C를 임의로 삭제한다. 인자 B와 C의 ISR표에 있는 C가 삭제되었다고 하면, 인자 C의 ISR표에서 남아 있는 갯수는 A와 C가 3개로 동일하다. 이 경우 A와 C인자에 대해 현 시점에서의 상대적 밀접도를 살펴보면, 인자 A와 C의 ISR표에 남아 있는 갯수는 각각 3, 7개이고 이것을 A인자와 B인자의 인자비율 1:2로 보정하면 각각 6개, 7개가 되어 A가 작으므로, 선택된 C인자의 직후행 관계는 A가 C보다 더 밀접하다. A를 선택하여 유전하면 O1[3]=A가 된다. 다시 ISR표에서 2개의 A를 임의로 삭제한다. 인자 C의 ISR표에 있는 A 2개가 삭제되었다고 하면, 선택된 A인자와의 직후행 관계는 ISR표의 인자A에서 A가 1개, B가 2개이므로 B가 더 밀접하다. 따라서 B를 선택한 후 유전하면 O1[4]=B가 된

다.

위와 같이 자손이 생산될때까지 인자를 선택하여 유전하면 O1 = (B C A B C C B C A)가 생산된다. O2의 경우에는 시작인자를 다시 임의선택하여 위와 같은 단계를 따른다.

ISR 교차변이는 두 부모의 인자별 직후행 관계에 대한 정보를 이용하여 직후행관계가 밀접한 자손을 생산하는 연산자이다.

부모 1 (P1) = (A A B B B C C C C)
 부모 2 (P2) = (A B C A B C B C C)
 자손 1 (O1) = (B C A B C C B C A)

그림 4.5 ISR 교차변이

4.2 일원 연산자

4.2.1 교환

교환연산자는 부모에서 서로 같지 않은 임의의 두 인자를 교환하여 자손을 생산하는 방법으로, 아래와 같이 부모에서 교환될 인자로 P[4]=B와 P[8]=C가 임의로 선택되었다면, 선택된 두 인자를 서로 교환하여 자손을 생산한다.

부모 (P) = (A A B B B C C C C)
 자손 (O) = (A A B C B C C B C)

4.2.2 삽입

삽입연산자는 부모에서 임의의 한 인자를 다른 임의 위치로 삽입하여 자손을 생산하는 방법으로, 아래와 같이 부모에서 삽입할 인자가 P[4]=B로 임의로 선택되고, 삽입될 위치가 (8th, 9th)의 인자사이로 임의 선택되었다면, P[4]=B를 (8th, 9th)의 인자사이로 삽입하여 자손을 생산한다.

부모 (P) = (A A B B B C C C C)

자손 (O) = (A A B B C C C B C)

4.2.3 역순

역순연산자는 부모에서 두개의 교차점을 임의로 선택하고 교차점사이의 인자들을 역순으로 하여 자손을 생산하는 방법으로, 아래와 같이 부모에서 교차점이 임의로 선택되었다면, 교차점사이의 인자들을 역순으로 하여 자손을 생산한다.

부모 (P) = (A B B | B A C C | C C)

자손 (O) = (A B B C C A B C C)

4.2.4 轉位

轉位연산자는 부모에서 두개의 교차점을 임의로 선택하고 교차점사이의 인자들을 임의의 위치로 전위하여 자손을 생산하는 방법으로, 아래와 같이 부모에서 교차점이 임의로 선택되고 전위될 위치가 (8th, 9th)의 인자사이로 임의로 선택되었다면, 교차점사이의 인자들을 (8th, 9th)의 인자사이로 전위하여 자손을 생산한다.

부모 (P) = (A A | B B B C | C C C)

자손 (O) = (A A C C B B B C C)

4.2.5 접합

접합연산자는 전위연산자의 한 형태로서 교차점사이의 인자들을 마지막 위치에 접합하여 자손을 생산하는 방법으로, 아래와 같이 부모에서 교차점이 임의로 선택되었다면 교차점사이의 인자들을 마지막 위치에 접합하여 자손을 생산한다.

부모 (P) = (A A | B B B C | C C C)

자손 (O) = (A A C C C B B B C)

5. 실험결과

혼합모델조립라인의 생산순서 결정에 적합한 유전알고리즘의 개발을 위하여, 이 장에서는 제 3장에서 제시한 수정유전알고리즘에 사용되는 유전파라미터 값과 제 4장에서 제시한 유전연산자를 실험을 통하여 비교분석하고자 한다. 결과분석은 실험의 신뢰도를 높이기 위하여 10회 반복실험한 평균치를 사용하였다.

본 연구에서는 예제로 Dar-El과 Nadivi [10]가 제시한 전화교환기 생산순서문제를 사용하였다. 이 예제는 총 27개의 작업장으로 구성된 조립라인에서 8종류의 모델을 생산하며, 각 작업장은 폐쇄작업장으로 이루어져 있다. 이때 사이클타임은 28.8분이고, 한 사이클동안에 생산되는 모델별 생산제품수는 각각 1, 4, 2, 4, 1, 2, 2, 1개이다. 한 사이클 동안에 생산할 총 제품수는 17개이므로 한 개체의 인자 수는 17개이며, 모델별 생산제품수만큼 한 개체에서 같은 인자가 다시 나타난다. 모델별 작업장에 따른 작업시간은 Dar-El과 Nadivi[10의 표4]를 사용하였다.

개방작업장도 폐쇄작업장과 같은 자료를 사용하기로 한다. 개방작업장의 경우에는 평가함수만 다르기 때문에 폐쇄작업장에 대해 분석하고, 개방작업장에 대한 분석결과는 간단히 언급하기로 한다.

5.1 유전파라미터의 비교분석

유전파라미터로는 모집단의 크기(N), 연산자의 적용비율(R), 순위생존방법의 생존확률 파라미터(q), 종료조건 등이 있다.

한 세대에서 가장 좋은 개체가 다음 세대

에서 항상 보존되는 우수개체(Elitism)생존전략을 사용하고, N은 50, 100, 200의 3 수준으로, R은 N의 30%, 50%, 70%의 3수준으로, q는 각 개체들의 평균 생존확률(1/N)의 2, 3, 4, 5배로 변화시키면서 이원연산자인 ISR연산자만을 사용하여 실험하였다. 종료조건은 재결합 회수, 세대수, 해의 변화정도 등이 있으나 본 연구에서는 재결합 회수를 사용하기로 한다. 본 예제의 경우 재결합된 개체수가 5000에 이르면 해의 변화정도가 미미하므로 이 시점을 기준으로 파라미터 분석을 하였다.

표 5.1은 5000회 재결합에서 얻은 가장 좋은 개체의 평가함수값(라인 길이)을 나타내고 있다.

표 5.1 5000회 재결합에서 가장 좋은 개체의 평가함수값

N	50			100			200		
	30%	50%	70%	30%	50%	70%	30%	50%	70%
q=2배	1992	1986	1996	1989	1993	1996	1992	1999	1988
q=3배	1988	1986	1991	1988	1984	1990	1985	1985	1984
q=4배	2005	1991	1995	1995	1987	1991	1990	1986	1984
q=5배	2013	1988	1992	1991	1991	1986	1998	1985	1986

본 예제뿐 아니라 여러 형태의 문제에 대해 실험분석한 결과, 문제에 따라 약간의 차이가 있으나 모집단의 크기, 연산자 적용 비율에 크게 유의할 만한 결과를 가져오지 않았다. 이는 재결합 회수에 의한 분석때문인 것으로 풀이된다. 표 5.1에 나타난 바와 같이 생존확률파라미터는 모집단에서 각 개체의 평균 생존확률의 3배일때 수렴속도 및 성능이 비교적 좋게 나타났다.

따라서 다음 절의 유전연산자 분석을 위한 실험에서 모집단의 크기를 100, 연산자 적용

비율은 모집단크기의 50%, 생존확률파라미터는 모집단에서 각 개체의 평균 생존확률의 3배로 두기로 한다.

5.2 유전연산자들의 비교분석

제 4장에서 개발된 연산자의 성능을 비교 분석하기 위하여 1) 이원연산자간의 성능비교분석과, 2) 일원연산자간의 성능비교분석, 그리고 3) 이원연산자와 일원연산자의 혼용에 따른 성능비교분석의 세가지 경우로 나누어 실험하였다. 이 실험에서는 우수개체 생존전략과 유전파라미터 비교분석을 통해 결정된 유전 파라미터값(N : 100, R : N의 50%, q : 3/N)을 사용하였다.

5.2.1 이원 연산자

이원연산자인 수정 OX, 수정 PMX, 수정 CX, SX, ISR에 대하여 좋은 해를 탐색하는 능력을 비교하고자 한다. 이를 위해 일원연산자를 사용하지 않고 이원연산자만을 사용하였다. 또한 각 연산자의 성능을 임의탐색(Random Search)과 비교하기 위해 각 세대마다 모집단 크기만큼 임의의 해를 발생시킨 "Random" 연산자를 사용하여 해를 탐색하였다.

그림 5.1은 각 이원연산자를 사용할 때 재결합 회수의 변화와 이때 모집단 중에서 가

장 좋은 평가함수 값(라인길이)을 나타내고 있다.

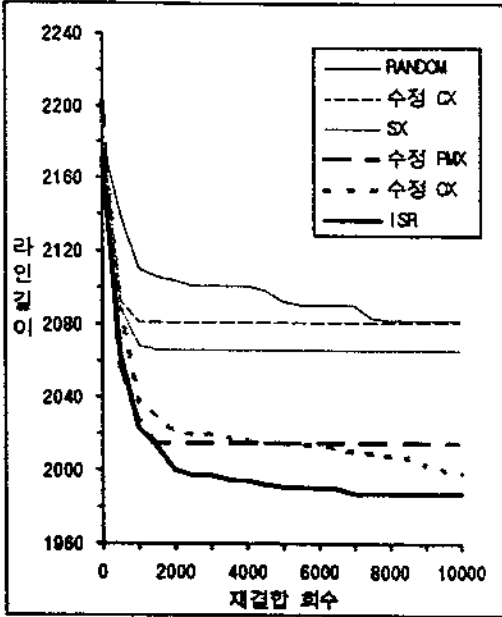


그림 5.1 이원연산자의 비교

그림 5.1에서 보여주듯이 이원연산자중에서 ISR이 가장 좋은 해를 찾고, 다음은 수정 OX, 수정 PMX, SX의 순이며, 수정 CX가 가장 나쁘게 나타났다.

ISR이 가장 좋은 것은, 혼합모델 조립라인 생산순서결정에 있어서 인자들의 순서나 위치의 정보보다 직후행관계 정보가 더 중요하다는 것을 의미한다.

ISR 다음으로 수정 OX가 좋게 나타난 것으로부터 순서의 정보 역시 중요하다는 것을 알 수 있다. 또한 수정 PMX, SX, 수정 CX의 순으로 성능이 떨어지는데, 이는 이들 연산자가 인자의 위치정보를 유전하는 정도가 강할수록 좋지 않다는 것을 알 수 있으며, 수정 CX는 자손에게 두 부모의 인자위치를 그대로 유전함으로써 임의 탐색(Random)방법

과 별 차이가 없음을 알 수 있다.

이원연산자의 역할은 평가함수값이 높은 개체들을 향해 각 개체들이 이동하면서 새로운 부분최적해를 찾는 데 있다. 그러나 그림 5.1에 나타난 바와 같이 수정 CX, SX, 수정 PMX 연산자들은 해공간을 효과적으로 탐색하지 못하고 조기수렴(Premature Convergence)함을 알 수 있다. 반면에 ISR과 수정 OX의 경우 해공간을 효과적으로 탐색함으로써 좋은 해의 발견가능성이 높아진다. 따라서 해의 수렴성이 우수한 연산자와 해공간의 탐색능력이 우수한 연산자를 조합하여 실험하여 보았다. 본 실험에서는 R의 크기가 50 일때 25개의 개체에는 SX, 수정 PMX, 수정 CX 중에 하나를, 다른 25개의 개체에는 ISR 또는 수정 OX를 사용하였다.

그림 5.2는 6가지 조합(수정 CX + ISR, SX + ISR, 수정 PMX + ISR, 수정 CX + 수정 OX, SX + 수정 OX, 수정 PMX + 수정 OX)들의 가장 좋은 평가함수값을 나타내고 있다.

그림 5.2는 서로 다른 특성을 가진 두 이원연산자를 조합하는 경우가 하나의 이원연산자만을 사용하는 경우보다 더 좋은 해를 탐색할 수 있음을 보여준다. 이는 수정 CX, SX, 수정 PMX 연산자가 갖는 해의 조기수렴성과 ISR과 수정 OX가 갖는 해공간의 효율적 탐색능력을 결합함으로써, 보다 좋은 해를 얻을 수 있음을 알 수 있다.

5.2.2 일원 연산자

교환, 삽입, 역순, 전위, 접합의 일원연산자들의 성능을 비교하기 위해 이원연산자를 사용하지 않고 일원연산자만을 사용하여 실험하였다.

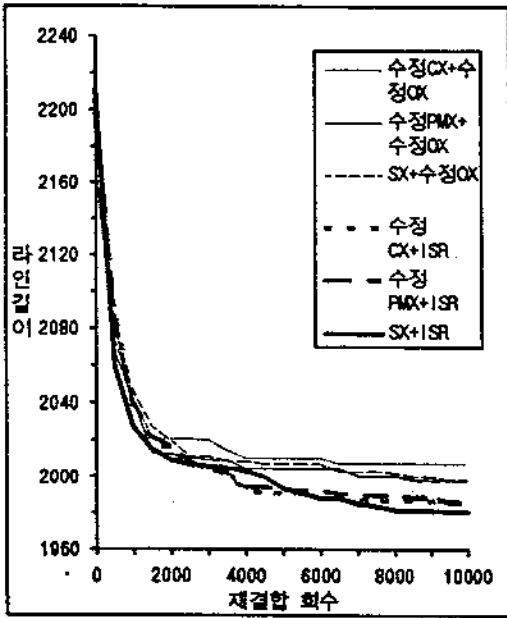


그림 5.2 2개의 이원연산자를 조합한 경우의 비교

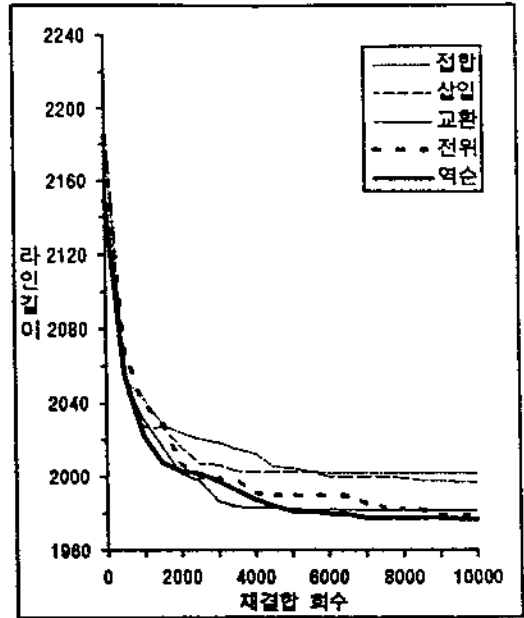


그림 5.3 일원연산자의 비교

그림 5.3은 이들 일원연산자를 사용한 경우 모집단에서 가장 좋은 평가함수값(라인길이)을 나타낸 것이다.

그림 5.3은 역순연산자가 가장 좋고, 다음은 전위, 교환, 삽입연산자 순으로 좋았으며 집합연산자가 가장 좋지 않음을 보여준다.

이원연산자와 비교할 때 일원연산자가 좋은 해를 탐색하는데 더 효과적이었으며, 이 결과는 외판문제[30]에서와 비슷하게 나타났다. 이는 부모로부터 좋은 인자의 유전과 해공간의 효율적 탐색을 잘 조화시키고 있음을 알 수 있다.

5.2.3 일원연산자와 이원연산자의 혼용

일원연산자중에서 가장 성능이 뛰어난 역순연산자와 이원연산자 중에서 성능이 뛰어난 ISR, 수정PMX+ISR, SX+ISR를 사용하여 실험하였다. 이때 일원연산자와 이원연산

자의 적절한 혼합비율을 찾기 위하여 일원연산자의 사용비율을 0.1에서 0.9까지 0.1씩 증가시키면서 실험하였다. 여기서 일원연산자의 사용비율이 0.3이란 의미는 R개중 30%를 일원연산자를, 나머지 70%는 이원연산자를 적용한다는 의미이다. 즉, 모집단 크기 N이 100, R이 50인 경우, 100개의 개체 중에서 (N - R)개인 50개는 그대로 다음 세대로 생산되고 모집단에서 중복가능하게 선택된 50개(R)의 개체 중에서 35개의 개체에 이원연산자를, 나머지 15개의 개체에 일원연산자를 적용한다는 것이다.

실험결과, 이원연산자가 ISR인 경우에 일원연산자비율이 0.5, 수정PMX+ISR인 경우에는 일원연산자비율이 0.7, SX+ISR인 경우에는 일원연산자비율이 0.3일때 가장 성능이 좋았다. 그림 5.4는 일원연산자와 이원연산자의 혼용에서, 위와 같은 일원연산자 비율을

사용하는 경우 모집단에서 가장 좋은 평가함수값을 나타내고 있다.

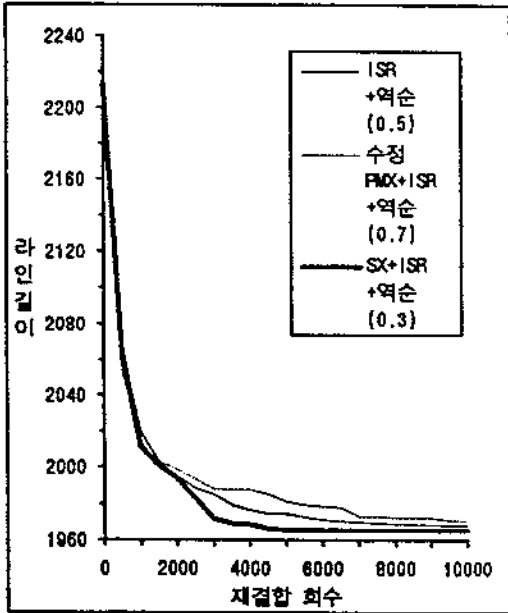


그림 5.4 일원연산자와 이원연산자의 혼용에 따른 비교

그림 5.3과 그림 5.4로 부터 일원연산자와 이원연산자를 혼용하여 사용한 경우가 이원연산자나 일원연산자만을 단독으로 사용하는 경우보다 더 좋은 해를 찾을 수 있음을 알 수 있다.

또한 그림 5.4에 나타난 바와 같이 이원연산자로 SX + ISR를 사용하고 일원연산자비율 0.3으로 역순연산자를 사용할 때 가장 좋은 해를 찾았다. 이때 10번의 실험중 8번이 10000회 재결합에서 가장 좋은 개체의 평가함수값(라인 길이)이 1964.4였고, 이때 제품의 생산순서는 (8, 4, 1, 7, 5, 2, 4, 6, 3, 2, 7, 4, 3, 6, 2, 4, 2)로 나타났다. Dar-El과 Nadivi [10]에서 구한 라인길이 2050.9와 비교해 볼 때 월등히 우수하다는 것을 알 수 있다.

지금까지는 폐쇄작업장에 대한 분석이었으나 개방작업장의 경우에도 폐쇄작업장과 같은 절차를 사용하여 실험한 결과, 유전파라미터의 값이나 유전연산자들에 대한 비교분석 결과가 폐쇄작업장의 경우와 거의 비슷하게 나타났다. 이때 가장 좋은 개체의 평가함수값은 1511.6였고, 제품의 생산순서는 (4, 2, 4, 7, 5, 6, 4, 2, 3, 7, 3, 6, 4, 2, 1, 2, 8)로 나타났다.

6. 결론

본 연구에서는 폐쇄작업장 또는 개방작업장으로 이루어진 혼합모델 조립라인의 길이를 최소로 하는 생산순서를 결정하는 유전알고리즘을 개발하기 위하여 수정 유전알고리즘을 기본으로 하여 문제에 적합한 표현방법을 찾고, 유전연산자를 수정 및 개발하였으며, 또한 실험을 통해 유전파라미터와 유전연산자를 비교분석하였다.

유전파라미터와 유전연산자의 실험결과는 아래와 같았다.

1) 유전파라미터의 분석에서 알고리즘의 성능은 생존확률파라미터가 평균 생존확률의 3배일때 비교적 좋게 나타났고, 재결합 회수의 변화에 따른 알고리즘의 성능은 모집단의 크기와 연산자 적용비율에 크게 영향을 받지 않았다.

2) 이원연산자중에서는 ISR연산자가 좋은 해를 탐색하는데 가장 효과적이었고, 다음은 수정 OX, 수정 PMX, SX연산자순이었으며 수정 CX가 가장 좋지 않았다. 이는 혼합모델조립라인의 생산순서결정에서는 직후행인자의 정보가 가장 중요하며 다음은 순서에

관한 정보임을 알 수 있다. 그리고 인자의 위치를 유전하는 정도가 강할 수록 좋지 않다는 것을 알 수 있다.

3) 일원연산자중에서는 역순연산자가 가장 좋았으며, 다음은 전위, 교환, 삽입연산자순으로 좋았으며 집합연산자가 가장 좋지 않게 나타났다. 또한 좋은 해를 탐색하는데 있어서 일원연산자가 이원연산자보다 효과적이었다.

4) 서로 다른 특성을 가진 2개의 이원연산자를 조합하여 사용한 경우, 단 하나의 이원연산자를 사용하는 경우보다 좋은 해를 얻을 수 있었으며, 각기 좋은 일원연산자와 이원연산자를 적절히 혼용하여 사용할 때 다른 어떤 경우보다 더 좋은 해를 찾을 수 있었다.

본 연구에서 개발된 유전알고리즘은 Dar-El과 Nadivi[10]가 구한 해보다 더 좋은 결과를 가져왔다.

참 고 문 헌

- [1] 김여근, "대형트럭라인의 생산성 향상에 관한 연구", Asia 자동차공업(주), 1990년 7월
- [2] 김여근, "중형트럭라인과 소형차라인의 생산성 향상에 관한 연구", Asia 자동차공업(주), 1991년 7월
- [3] Baker, J.E., "Adaptive Selection Methods for Genetic Algorithms", Proceedings of An International Conference on Genetic Algorithms and Their Applications, pp. 100-111, 1985.
- [4] Baker, J.E., "Reducing Bias and Inefficiency in the Selection Algorithm", Proceedings of the Second International Conference on Genetic Algorithms, pp.14-21, 1987.
- [5] Bard, J.F., Dar-El, E.M. and Shrub, A., "An analytic framework for sequencing mixed model assembly lines", International Journal of Production Research, Vol.30, No.1, pp.35-48, 1992.
- [6] Back, T., Reeves, C.R., "Optimal Mutation Rates in Genetic Search", Proceedings of the fifth International Conference on Genetic Algorithms, pp. 2-8, 1993.
- [7] Burns, L.D. and Daganzo, C.F., "Assembly line job sequencing principles", International Journal of Production Research, Vol.25, No.1, pp.71-99, 1987.
- [8] Dar-El, E.M., "Mixed-Model Assembly Line Sequencing Problems", Omega, Vol. 6, No.4, pp.313, 1978.
- [9] Dar-El, E.M. and Cocher, R.F., "Assembly Line Sequencing for Model-Mix", Management Science, Vol.13, No.5, pp.463-477, 1975.
- [10] Dar-El, E.M. and Navidi A., "A Mixed-Model Sequencing Application", International Journal of Production Research, Vol.19, No.1, pp.69-84, 1981.
- [11] Davis, L., "Applying Adaptive Algorithms to Epistatic Domains", Proceedings of the International Joint Conference on Artificial Intelligence, pp.162-164, 1985.
- [12] Fox, B.R. and McMahon, M.B., "Genetic Operators for Sequencing Problems",

- Foundation of Genetic Algorithms, Edited by Gregory J.E., Morgan Kaufmann Publishers, 1991.
- [13] Goldberg, D.E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, 1989.
- [14] Goldberg, D.E. and Deb, K., "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.
- [15] Goldberg, D.E. and Lingle, R., "Alleles, Loci and the TSP", Proceedings of An International Conference on Genetic Algorithms and Their Applications, pp. 154-159, 1985.
- [16] Grefenstette, J.J., "Optimization of Control Parameters for Genetic Algorithms", IEEE Transactions on System, Man, and Cybernetics, Vol.16, No.1, pp.122-128, 1986.
- [17] Holland, J.H. and Reitman, J.S., Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.
- [18] Homaifar, A., Guan, S. and Liepins, G., "A New Approach on the Traveling Salesman Problem by Genetic Algorithms", Proceedings of the fifth International Conference on Genetic Algorithms, pp. 460-466, 1993.
- [19] Inman, R.R. and Bulfin, R.L., "Sequencing JIT Mixed-Model Assembly Lines", Management Science, Vol.37, No.7, pp. 901-904, 1991.
- [20] Jog, P., Suh, J.Y., Gucht, D.V., "The Effects of Population Size, Heuristic Crossover, and Local Improvement on a Genetic Algorithm for the Traveling Salesman Problem", Proceedings of the Third International Conference on Genetic Algorithms, pp.110-115, 1989.
- [21] Kilbridge, M., and Webster, L., "The assembly line model-mix sequencing problems", Proceeding of the Third International Conference on Operations Research, Oslo, pp.247, 1963.
- [22] McCormick, T.S., Pinedo, M.L. and Wolf, B., "Sequencing in an Assembly Line with Blocking to Minimize Cycle Time", Operations Research, Vol.37, No.6, pp. 925-937, 1989.
- [23] Michalewicz, Z., Genetic Algorithms + Data Structure = Evolution Programs, Springer-Verlag, pp.58-62, 1992.
- [24] Michalewicz, Z. and Janikow, C., and Krawczyk, J., "A Modified Genetic Algorithm for Optimal Control Problems", Computers & Mathematics with Applications, Vol.23, No.12, pp.83-94, 1992.
- [25] Miltenburg, J., "Scheduling Mixed-Model Multi-Level Just-In-Time Production System", Management Science, Vol.27, No.9, pp.1487-1509, 1989.
- [26] Miltenburg, J. and Sinnamon, G., "Algo-

- rithms for Scheduling Multi-Level Just-In-Time Production Systems”, IIE Transactions, Vol.24, No.2, pp.121-130, 1992.
- [27] Okamura, K., Yamashina, H., “Heuristic Algorithm for the Assembly Line Model-Mix Sequencing Problem to Minimize the Risk of Stopping the Conveyor”, International Journal of Production Research, Vol.17, No.3, pp.223-247, 1979.
- [28] Oliver, I.M., Smith, D.J. and Holland, J. R.C., “A Study of Permutation Crossover Operators on the Traveling Salesman Problem”, Proceedings of the Second International Conference on Genetic Algorithms, pp.224-230, 1987.
- [29] Reeves, C.R., “Using Genetic Algorithms with small Populations”, Proceedings of the fifth International Conference on Genetic Algorithms, pp. 92-99, 1993.
- [30] Starkweather, T., McDaniel, S., Mathias, K., Whitley, C., Whitley, D., “A Comparison of Genetic Sequencing Operators”, Proceedings of the Fourth International Conference on Genetic Algorithms, pp.69-76, 1991.
- [31] Sumichrast, R.T., Russel, R.S. and TaylorIII, B.W., “A Comparative analysis of sequencing procedures for mixed-model assembly lines in a just-in-time production system”, International Journal of Production Research, Vol.30, No.1, pp. 199-214, 1992.
- [32] Thomopoulos, N.T., “Line Balancing-Sequencing for Mixed Model Assembly”, Management Science, Vol.14, No.2, pp. 593, 1967.
- [33] Whitley, D., “GENITOR Algorithm and Selection Pressure : Why Rank-Based Allocation of Reproduction Trials is Best”, Proceedings of the third International Conference on Genetic Algorithms and Their Applications, pp.116-121, 1989.
- [34] Whitley, D., Starkweather T. and Fuquay, D., “Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator”, Proceedings of the Third International Conference on Genetic Algorithms, pp.133-140, 1989.