복합 유전자 알고리즘을 이용한 경제적 로트 일정계획 문제

문일경† · Edward Silver‡ · 최상진†
† 부산대학교 산업공학과
‡ Faculty of Management, University of Calgary, CANADA

경제적 로트 일정계획 문제(Economic Lot Scheduling Problem : ELSP)는 지난 수십 여 년간 많은 연구가 이루어진 생산일정계획 문제 중의 한 분야이다. 이 문제는 NP-hard 문제이기 때문에 수많은 발견적 기법이 제안되고, 사용되어져 왔다. 그 중에서도 Dobson[1]의 발견적 기법이 그 수행도의 우수성으로 보아 최고의 기법으로 여겨지고 있는데, 본 연구에서는 Dobson[1]의 시변 로트 크기(time varying lot size) 접근방법에 유전자 알고리즘을 이용한 새로운 발견적 기법을 제안하고, 수치실험을 통해서 새로운 기법이 기존의 Dobson[1]의 기법보다 더 우수하다는 것을 보이고자 한다.

1. 서론

대량생산의 경제성에 기인하여 여러 제품을 한 대의 설비나 기계에서 생산하는 것은 드문 일이 아니다. 전형적으로 이러한 기계나 설비들은 한번에 한 제품을 생산할 수밖에 없고 또한 다른 제품을 생산하기 위해서는 우선 기계를 멈추고 다른 제품의 생산 준비시간과 준비비용이 필요하다. 따라서 이러한 추가 비용 때문에 경제적인 생산 일정계획이 필요하게 되었다. 본 연구의 대상문제인 경제적 로트 일정계획 문제(ELSP)는 다품목을 단일설비로 제조하는 경우, 각 품목의 수요를 만족시키며 연간 평균 준비비용과 재고유지비용을 최소로 하는 일정계획을 수립하는 문제이다. Boctor[2]가 예시한 바와 같이 경제적 로트 일정계획 문제는 다음과 같은 여러 생산현장에서 찾아볼 수 있는 실제적인 문제이다.

(a) 각 품목이 작업준비를 위해 다른 종류의 다이(die)를 필요로 하는 금형가공 라인이나 플라스틱 압출(extrusion) 라인
(b) 여러 가지 제품들 또는 여러 가지 모델들을 생산하는 조립라인
(c) 페인트, 음료수 등처럼 다른 품목들이 다른 용기들에 섞이는 연접(blending) 및 혼합가공라인
(d) 방직제품처럼 주 품목들이 여러 가지 색상, 규격, 등급 등으로 제조되는 직조(weaving) 가공라인

일반적으로 하나의 제품만을 생산하는 설비를 여러 대 구매하는 것보다, 여러 제품을 동시에 생산해 낼 수 있는 설비를 구입하는 것이 훨씬 더 경제적이다. 이러한 다기능의 성능을 지닌 설비를 어떻게 운용하는 가에 대한 논쟁은 제품의 생산순서와 생산크기를 정하는 것으로 결정지어진다. 생산일정계획 문제에 대해서 지난 40여 년간 많은 연구가 이루어져 왔는데, 그 이유는 이러한 종류의 문제가 일정계획 문제에서 자주 다루는 대표적인 문제이고 또한 해를 구하기가 어렵기 때문이다.

경제적 로트 일정계획 문제는 비선형성(nonlinearity), 조합성 (combinatorial characteristics), 그리고 복잡성(complexity) 때문에 NP-hard 문제로 알려져 있다[3]. 따라서 많은 발견적 기법들이 개발되었는데, 접근방법에 따라서 크게 세 가지로 나누어진다.

(1) 공통 주기 방법(Common Cycle Approach) : 이 접근 방법은 모든 제품의 생산주기는 동일하다는 가정을 하고 있다. 한 제품에 대한 생산주기란 그 제품의 생산을 시작하는 시점에서 그 제품을 다시 한번 더 생산할 때까지 걸리는 시간을 말한다. 이 방법은 간단한 해법절차를 통해서 항상 실현 가능한 해를 제공해준다. 하지만 몇몇의 경우에는 하한 값(lower bound)과 차이가 많이 나는 해를 제공한다.

(2) 기본 주기 방법(Basic Period Approach) : 이 접근 방법에서는 제품들은 서로 다른 생산주기를 가질 수 있다. 하지만 각 제품의 생산주기는 기본 주기(basic period)의 정수 배이어야 한다는 제약조건이 있다. 또한 각 제품의 로트 크기는 일정하다. 일반적으로 이 접근 방법은 공통주기 방법보다는 더 나은 해를 제공하지만, 실현 가능한 해를 제공하지 않을 수도 있다는 단점이 있다.

(3) 시변 로트크기 방법(Time-Varying Lot Sizes Approach) : 이 접근 방법은 생산주기내의 각 제품에 대해서 서로 다른 로트 크기를 허용한다. 또한 항상 실현 가능한 해를 제공해 준다는 것이 Dobson[1]에 의해서 증명되었다. 이 방법은 이전의 두 가지 접근방법보다는 항상 나은 해를 제공해 준다.

2. 경제적 로트 일정계획에 관한 기존 연구

지난 40 여 년 동안 경제적 로트 일정계획에 관한 많은 연구가 이루어졌으며, 100편이상의 논문들이 여러 종류의 국제저널에 투고되었다. 이와 같이 국제적으로는 경제적 로트 일정계획에 대해서 많은 연구가 이루어지고 있지만, 국내에서는 아직까지는 많은 연구가 이루어지고 있지 않은 실정이다. 따라서 본 장

에서는 이러한 논문들 중에서 몇몇 논문들에 대해서 간단하게나마 요약을 하고자 한다.

경제적 로트 일정계획 문제의 초기 연구들은 Eilon[4], Rogers[5] 그리고 Hanssmann[6]에 의해서 이루어졌다. 이 문제에 대한 최적해는 구하기가 어렵기 때문에 대부분의 연구들은 발견적 기법을 이용하여, 하한 값을 구하거나 아니면 실현 가능한 해를 구하는 방법을 중심으로 이루어 졌다. 만약 모든 제품들의 생산 주기 시간이 같아야 한다고 제약을 한다면 이 문제는 좀 더 간단한 경제적 로트 일정계획문제인 공통 주기 방법을 이용하여 해를 구해낼 수 있다. 이 방법으로부터 구한 최적해는 일반적인 경제적 로트 일정계획 문제의 상한 값으로 대신할 수 있다. 또한, Jones and Inman[7]과 Gallego[8]은 공통 주기 방법으로 구한 해는 몇몇의 경우에 좋은 해를 낸다는 것을 실험을 통해서 보였다.

기본 주기 방법은 모든 제품의 생산주기는 기본 주기의 정수배가 되어야 한다는 조건 이외에 ZSR 규칙이 필요하다. 이 방법을 이용하는 대부분의 발견적 기법들은 먼저 각 제품이 주기내에서 몇 번씩 생산되어야 하는 가에 대한 생산빈도수를 먼저 계산하고, 이러한 빈도수를 만족시키는 실현 가능한 해를 찾는다[9]. Elmaghraby[10]는 이 접근 방법에 대해서 1970년대 후반까지 연구들을 아주 자세하게 정리하였다. 또한 Khouza외 2인[11]은 기본주기 방법에 기초한 경제적 로트 일정계획 문제에 유전자 알고리즘을 적용하였다. 하지만 기존의 Dobson[1]의 발견적 기법을 이용한 해보다는 더 나은 값을 찾지 못했다.

기본 주기 방법의 단점을 보완한 시변 로트 크기 방법은 Maxwell[12]과 Delporte and Thomas[13]에 의해서 제안되고 연구되어졌다. 또한 Dobson[1]은 어떠한 생산순서일지라도, 생산 준비시간을 고려하는 실현가능한 해로 바뀌어 질 수 있다는 것을 보였다. Dobson[1]의 발견적 기법과 Zipkin[14]의 알고리즘을 결합하면 거의 최적에 가까운 해를 구해낼 수 있는데, Zipkin[14]의 알고리즘이란 제품의 생산순서가 주어진 상태에서 각 제품에 대한 생산시간과 기계 유휴시간을 구해내는 것이다. Gallego and Roundy[15]는 경제적 로트 일정계획 문제에 부재고를 고려한 경우에 대해서 시변 로트 크기 방법을 이용하였다. 또한 Dobson[16]은 그의 이전 논문[1]에 대해서, 생산 준비시간이 작업순서에 의존하는 경우 (sequence dependent setup time)를 고려하였다.

3. 문제의 정의 및 복합 유전자 알고리즘 기법

3.1 가정 및 사용기호

경제적 로트 일정계획 문제에서 사용되는 가정들은 아래와 같다.

(1) 한 대의 설비에 여러 가지의 제품이 가공될 수 있다.
(2) 연간 수요, 생산율, 생산 준비비용, 그리고 생산 준비시간에 대한 정보는 주어져 있으며 일정한 상수 값을 갖는다.
(3) 부재고는 허락되지 않는다.

또한 사용되는 기호들은 아래와 같다.

제품 인덱스	i
생산순서와 관련한 위치 인덱스	j
생산율 (개/일)	p_i
수요율 (개/일)	d_i
재고 비용 ($/개/일)	h_i
생산 준비 비용 ($)	A_i
생산 준비 시간 (일)	s_i
j 위치에서 생산되는 제품	f^j
j 위치에서 생산되는 제품 생산시간(일)	t^j
j 위치에서의 제품 유휴시간(일)	u^j
생산 주기 (일)	T

i 의 인덱스는 1, 2 ,..., m 까지 이고, j 의 인덱스는 1, 2 ,..., n 까지 이다. 앞에서 제시한 사용기호를 이용하여 경제적 로트 일정계획 문제를 다시 설명하면 다음과 같다. 먼저 m 종류의 서로 다른 제품들이 한 대의 설비나 기계에서 생산된다. 주어진 생산시간 내에서 각각의 제품들은 주어진 생산순서대로 생산되고, 제품의 수요를 맞추면서, 재고비용과 생산 준비비용의 합이 최소로 되는 생산주기(T), 생산 순서 ($f=(f^1,\cdots,f^n)$), 생산 시간($t=(t^1,\cdots,t^n)$), 그리고 유휴 시간($u=(u^1,\cdots,u^n)$) 을 구해내는 것이다. 여기서 $f^j \in \{1,\cdots,m\}$이다.

상수 χ 는 생산 주기동안 준비시간에 필요한 시간의 비율이라고 한다면, 이 값은 다음의 식에서 구해낼 수 있다.

$$\chi = 1 - \sum_{i=1}^{m} \frac{d_i}{p_i}$$

계획기간이 무한한 경우(infinite horizon)에는 양수인 χ의 값이 실현가능한 해를 찾아주는 필요조건이 된다. Dobson[1]은 만약 χ의 값이 양수인 경우에, 어떠한 생산순서도 충분히 큰 생산시간과 시변(time-varying) 생산일 경우에는, 실현가능한 해로 바뀌어 질 수 있다는 것을 보였다.

3.2 복합 유전자 알고리즘 기법

이 절에서는, 복합 유전자 알고리즘 기법에 대해서 살펴보고자 한다. 먼저 Dobson[1]이 제안한 원래의 모형을 소개하도록 한다. 여기서 아래첨자 i 는 제품 i 를 가리키고 위첨자 j 는 생산순서 중에서 j 번째 생산되는 제품을 가리킨다. 그리고, F 는 모든 가능한 해들을 가리키는 집합이다. J_i 는 j 번째 생산되는 제품 i 를 가리킨다. 또한 L_k 는 주어진 생산순서

의 k 번째에서부터 시작하여, k 번째 생산된 제품, 즉 f^k 제품이 다시 생산되기 전까지의 생산순서들을 말한다. Dobson[1]에 의한 모델은 아래와 같다.

$$\inf_{j\in F} \ \min_{t\geq 0 \ u\geq 0 \ T\geq 0}$$
$$\frac{1}{T}\left(\sum_{j=1}^{n}\frac{1}{2}h^j(p^j-d^j)\left(\frac{p^j}{d^j}\right)(t^j)^2+\sum_{j=1}^{n}A^j\right) \quad (1)$$

subject to

$$\sum_{j\in J_i} p_i t^j = d_i T \qquad i=1,\ldots,m \quad (2)$$
$$\sum_{j\in L_k}(t^j+s^j+u^j)=(p^k/d^k)t^k \quad k=1,\ldots,n \quad (3)$$
$$\sum_{j\in L_k}(t^j+s^j+u^j)=T \quad (4)$$

제약식 (2)는 제품 i 에 대한 수요, d_iT 를 만족 시켜주기 위해서 생산주기에 대하여 충분한 생산시간을 할당하게 하는 제약식이다. 제약식 (3)은 각 제품에 대해서 그 제품이 다음 번 생산을 시작하기 전까지 충분한 양을 생산해야 한다는 식이다. 제약식 (4)는 생산주기 T는 모든 제품에 대한 생산 시간, 생산 준비시간 그리고 유휴시간의 합이어야 한다는 식이다.

Dobson[1]의 모델은 시변 로트 크기 방법에 기초를 둔 접근 방법이기 때문에 최적해를 구한다는 것은 거의 불가능하기 때문에 많은 발견적 기법이 제안되어왔다. 그 중 대다수의 발견적 기법들은 문제 해결의 용이성 때문에 공통 주기 방법이나 기본 주기 방법을 이용하였다. 하지만 시변 로트 크기를 이용하면 방법상의 어려움은 있겠지만 더 좋은 해를 구해낼 수 있다. 따라서 본 연구에서는 시변 로트 크기 방법을 이용하여 해를 구하고자 한다. 본 연구에서 제안하는 복합 유전자 알고리즘 기법은 아래와 같다.

단계 1. 아래의 LB 식으로부터 생산 빈도수를 구한다. 아래의 하한 값은 기계의 용량에 대한 제약식을 추가하여 경제적 생산량을 구한 것이다. 기계의 용량에 대한 제약식은 작업 준비시간을 고려하기 위한 식이다. 그러나, 두 제품이 동시에 한 대의 설비에서 생산될 수 없다는 동시 제약식(synchronization constraint)은 고려되지 않았다. 결과적으로 아래의 비선형 문제에서 나오는 해는 원 문제의 하한 값이 된다.

LB

$$Min_{T_1,\ldots,T_m}\sum_{i=1}^{m}\left[\frac{A_i}{T_i}+\frac{h_id_iT_i}{2}\left(1-\frac{d_i}{p_i}\right)\right]$$

subject to

$$\sum_{i=1}^{m}\frac{s_i}{T_i} \leq x \quad (5)$$
$$T_i \geq 0 \qquad i=1,\ldots,m$$

위 모델에서의 목적식과 제약식들은 T_i 값들에 대해서는 아래로 볼록인 모양(convex)이기 때문에 LB의 최적의 해는 Karush - Kuhn - Tucker (KKT) 조건을 만족하는 점이다. 따라서

$$T_i=\sqrt{\frac{A_i+\lambda s_i}{H_i}} \qquad \forall i$$

$\lambda\geq 0$ complemetary slacknes with $\sum_{i=1}^{m}\frac{s_i}{T_i}\leq x$

여기서 $H_i=h_id_iT_i\left(1-\frac{d_i}{p_i}\right)/2$ 이다. 아래의 절차를 이용하면 최적의 T_i 값들을 찾을 수 있다.

하한 값을 구하기 위한 알고리즘

[단계1] 먼저 λ = 0인 경우에 조건을 만족하는 지 확인한다. 만약 만족하면 최적해이다. 그렇지 않은 경우는 아래의 식을 이용하여 T_i 를 구한다.

$$T_i=\sqrt{\frac{A_i}{H_i}} \qquad \forall i$$

[단계2] 만약 $\sum_{i=1}^{m}\frac{s_i}{T_i}\leq x$이면, T_i 값들은 최적의 해이다. 그렇지 않으면 [단계3]으로 간다.

[단계3] 임의의 λ(양수)에서 시작한다.

[단계4] 모든 i 에 대해서 $T_i=\sqrt{\frac{A_i+\lambda s_i}{H_i}}$ 를 계산한다.

[단계5] 만약 $\sum_{i=1}^{m}\frac{s_i}{T_i}<x$이면, λ 값을 감소시키고, [단계4]로 간다. 만약 $\sum_{i=1}^{m}\frac{s_i}{T_i}>x$이면, λ값을 증가시키고, [단계4]로 간다. 만약 $\sum_{i=1}^{m}\frac{s_i}{T_i}=x$이면, 이 때의 값이 최적의 T_i 값이다.

LB에서 구한 제품 i 에 대한 최적의 생산 주기를 T_i^* 라고 하자. 그리고 x_i 는 제품 i 에 대한 상대적인 생산빈도수라고 하면, x_i 는 아래의 식으로부터 구해낼 수 있다.

$$x_i=\frac{\mathrm{Max}\{T_i^*\}}{T_i^*} \qquad i=1,2,\ldots,m$$

단계 2. 단계 1에서 구한 생산빈도수는 실수 값이기 때문에 먼저 이 실수 값을 가장 가까운 정수 값으로 반환한다.

단계 3. 주어진 생산빈도수를 가지고 복합 유전자 알고리즘을 이용하여 제품의 생산순서 f 를 구한다. 복합 유전자 알고리즘에 대한 자세한 방법에 대해서는 다음 장에서 설명하도록 한다.

단계 4. 이 단계에서는, 생산순서 f 가 정해지고 난 다음 생산시간 t 와 유휴시간 u 를 구한다. 만약 유휴시간이 없다($u=0$)라고 가정한다면, 생산빈도수 f 가 주어진 상태에서는 수식 (3)을 이용하여 생산시간 t 를 구해낼 수 있다.

Dobson[1]이 제시한 발견적 기법과 복합 유전자 알고리즘 기법의 가장 큰 차이는 생산빈도수와 생산순서를 구하는 방법에 있다. 다시 말해서, Dobson[1]의 발견적 기법은 본 연구에서 제시한 단계중 단계 2와 단계 3에서 명확한 차이가 나게 된다.

단계 2. 단계 1 에서 구한 실수 값의 생산빈도수를 2의 승수 형태의 정수(power-of-two integers)로 변환한다. Roundy[17]에 의하면 실제 생산빈도수를 2의 승수 형태로 변환했을 경우의 추가비용은 6%를 초과하지 않는다. 또한 이러한 형태의 생산빈도수 값은 단계 3 에서 생산순서를 구하기 위함이다. 제품 i 에 대한 2의 승수에 의한 생산빈도수를 y_i 라고 한다면, y_i는 다음의 식에서 구해낼 수 있다.

$$\text{If } x_i \in \left[\frac{1}{\sqrt{2}}2^p, \sqrt{2}2^p\right), \text{ then } y_i = 2^p \; p = 0, 1, \ldots$$

단계 3. Dobson[1]에 의해서 제안된 bin-packing 발견적 기법을 이용하여 생산순서(f)를 구한다. (좀 더 자세한 내용은 Dobson[1]을 참조하기 바란다.)

Dobson[1]의 발견적 기법에서 생산빈도수를 2의 승수 형태의 정수로 바꾸는 이유는 기존의 bin-packing 발견적 기법을 사용하기 위해서는 생산빈도수가 반드시 2의 승수 형태의 정수로 되어야 하기 때문이다. 물론 이렇게 함으로써 추가비용이 최고 6%정도밖에 되지 않지만, 총비용의 1% 정도를 절감하는 것도 상당히 중요한 과제라고 할 수 있다. 따라서 본 논문에서 제시하고자 하는 유전자 알고리즘은 2의 승수로 변환한 정수 값을 사용하는 것이 아니라, LB에서 계산된 실수의 생산빈도수를 가장 가까운 정수 값으로 바꾸는 것이다. 또한 유전자 알고리즘을 사용하면 알고리즘의 특성상, 3단계의 bin-packing 발견적 기법을 사용하는 것보다 더 나은 해를 찾을 수 있다.

4. 유전자 알고리즘을 이용한 생산순서의 결정

이 장에서는 3.2절에서 설명한 복합 유전자 알고리즘의 기법의 단계 3에 대해서 좀 더 자세히 설명하도록 하겠다. 먼저 유전자 알고리즘의 기본적인 이론에 대해서 설명하고, 경제적 로트 일정계획 문제에 어떻게 유전자 알고리즘을 적용하게 되는 지에 대해서 살펴보기로 한다.

유전자 알고리즘은 기존의 다른 탐색기법들과는 달리, 하나의 해에서 탐색을 시작하는 것이 아니라, 해의 집단(population)에서부터 탐색을 시작한다. 해의 집단에서 하나의 해는 염색체(chromosome)라고 하는데, 각 염색체는 유전인자(gene)로 이루어져 있다. 유전인자는 문제에 따라서 형태가 달라지며, 대개 0 또는 1의 수를 가지는 문자열의 형태를 가지지만, 정의하기에 따라서 여러 가지의 형태를 가질 수 있다.

4.1 해의 표현과 초기화

유전자 알고리즘을 이용하여 문제를 풀기 위해서는 먼저 문제의 후보해를 문자열(string)로 표현하는 것이 제일 중요한 과제이다. 본 연구에서는, 염색체(chromosome)의 길이는 제품 생산빈도수의 합이 되고, 염색체내의 각각의 유전인자(gene)는 제품의 index를 가리킨다. 주어진 제품의 생산빈도수에 맞는 염색체를 만든다는 것이 쉽지 않기 때문에, 염색체내에서 유전자의 절대위치 값을 가지는 보조 염색체를 하나 더 만든다. 이 두 개의 염색체를 Info A와 Info B라고 두면, Info A는 염색체내의 유전인자의 절대적인 위치를 나타내고, Info B는 어떤 제품인지를 가리키고 있다. 예를 들어, 총 다섯 개의 제품이 있고 각각의 제품이 두 번, 두 번, 세 번, 세 번, 그리고 한번씩 한 사이클 동안에 생산되어야 한다면, 길이가 11인 하나의 염색체를 만들어야 한다. 이때의 염색체의 구조는 그림 1과 같다.

Info A	1	2	3	4	5	6	7	8	9	10	11
Info B	1	1	2	2	3	3	3	4	4	4	5

그림 1. 염색체의 구조

실제 하나의 염색체가 만들어지면 Info B를 이용해서, 각각의 유전인자에 대한 제품을 알 수 있게 된다. 예를 들어, 염색체 (5, 3, 8, 6, 1, 9, 4, 7, 11, 10, 2)는 실제로 염색체 (3, 2, 4, 3, 1, 4, 2, 3, 5, 4, 1)을 의미하게 된다. 하지만 실제로 적합도 함수 값을 계산하는 경우에는 Info A의 정보가 아니라 Info B에 대한 정보를 가지고 적합도 함수 값을 계산해야 한다.

4.2 목적함수와 적합도 함수

본 연구에서는 최근 들어 아주 널리 사용되고 있는 유전자 알고리즘 프로그램, GAlib를 사용하는데, 이 프로그램은 많은 종류의 목적함수를 제공하고 있다[18]. 따라서 단순히 제공하는 여러개의 목적함수중 하나를 택하면 된다. 본 연구에서는 ga.minimize() 함수를 선택하기로 한다. GAlib 프로그램은 C++로 이루어진 유전자 알고리즘 모듈들을 가지고 있는데, 이 모듈들을 이용하여 여러 종류의 염색체 표현과 유전 연산 등을 할 수 있다.

적합도 함수는 각각의 해 집단에 있는 모든 염색체들에 대해서 계산되는데, 목적함수가 최소값을 찾는 것이므로, 적합도 값이 최소가 되는 염

색체들을 찾는 것이 목적이다. 만약 하나의 염색체가 주어진다면 3.2절의 단계 4에서 생산시간을 구해낼 수 있고, 수식 (1)의 목적함수로부터 적합도 함수 값을 찾아 낼 수 있다.

4.3 재생산(reproduction)
재생산이란 이전 세대의 해들이 목적함수의 값에 따라 이 후 세대에도 존재하는 것을 말한다. 적합도 값이 좋은 염색체들을 다음 세대로 복사를 하는 이유는, 이 염색체들을 교배나 돌연변이 등을 이용해서 나온 다음 세대의 자손이 좀 더 우성이 될 확률이 높기 때문이다. 이러한 방법은 다윈에 의해 알려진 자연선택의 인공적인 형태이다.

재생산 연산자는 여러 가지 방법으로 실현될 수 있는데, 본 연구에서는 stochastic tournament 방법을 사용하기로 한다. 이 방법은 일반적인 방법으로 염색체의 선택확률을 계산하지만, 연속되는 두 개의 해를 선택해서 더 높은 적합도 값을 가지는 염색체를 선정한다. 이러한 방법으로 원하는 해의 수를 만족할 때까지 계속한다.

4.4 교배(crossover)와 돌연변이(mutation)
일반적인 교배 연산자는 하나의 교배위치가 정해지면 부모의 염색체에 대한 정보를 서로 교환한다. 하지만 본 연구에서는 유전자가 가지는 값이 이진수가 아닌 순서를 가지는 염색체이기 때문에 좀 더 복잡한 교배연산자가 필요하게 된다. 따라서 염색체를 교환하면서 중복되지 않는 염색체를 만들어 줄 수 있는 새로운 형태의 연산자가 필요하다. 본 연구에서는, 이러한 용도에 가장 많이 사용되고 있는 PMX(Partial Matched Crossover) 연산자를 사용하기로 한다. 이 연산자에 대해서 좀더 자세히 알아보기 위해서 그림 2에서는 Mallya[19]의 예를 이용하여 PMX 연산자를 설명하도록 한다.

PMX 연산자는 위치교환을 함으로써 염색체들을 교배한다. 먼저 P2 를 P1에 할당(mapping)하면 염색체의 두 번째 부분, 1 과 2, 9 와 7, 6 과 5, 그리고 4 와 3 이 위치를 바꾸게 된다. 유사한 방법으로 P1을 P2에 할당하면 2 와 1, 7 과 9, 5 와 6, 그리고 3 과 4 가 위치를 교환하게 된다. 이러한 교환 작업을 통하여 O1과 O2가 새롭게 생겨나게 된다. 하지만 O1과 O2는 Info A의 정보이기 때문에, 최종적으로 O1과 O2의 적합도 값을 계산하기 위해서는 Info B로 바뀌어 져야 한다. 따라서, O1의 경우는 (4, 3, 2, 1, 4, 3, 2, 5, 4, 1, 3)으로 바뀌게 된다.

```
P1   8  9  4   |  2  7  5  3  |  11 10  1  6
P2   2  8  10  |  1  9  6  4  |  3  7  5  11
                      ⇓
O1   8  7  3   |  1  9  6  4  |  11 10  2  5
O2   1  8  10  |  2  7  5  3  |  4  9  6  11
```

그림 2. PMX 연산자

돌연변이(mutation)는 염색체내의 임의의 유전자 값을 바꾸는 것을 말한다. 유전자 알고리즘에서 돌연변이 연산자가 미치는 영향은 절대 과소평가할 수 없다. 왜냐하면 돌연변이 연산자로 하여금 새로운 탐색영역이 생기고, 이전의 세대에서 없었던 새로운 형태의 염색체들도 만들어 주기 때문이다. 본 연구에서 사용하는 돌연변이 연산을 그림으로 나타내면 아래와 같다.

```
8  7  3  1  9  6  4  11  10  2  5
               ⇓
8  9  3  1  7  6  4  11  10  2  5
```

그림 3. 돌연변이 연산자

4.5 적합도 스케일링(fitness scaling)
스케일링 작업은 시뮬레이션 전반을 통해서 염색체들간의 적절한 우선 순위를 조절해주는 역할을 한다. 일반적으로 탐색의 초기에는, 개체집단 내의 적합도 분산(fitness variance)은 대체적으로 높고, 소수의 개체들이 다른 염색체들보다 적합도가 훨씬 높게 된다. 또한 적합도 비례 선택하에서, 이들과 이들 자손들이 개체집단 내에서 빠르게 증식되기 때문에, 사실상 유전자 알고리즘이 더 이상의 개발을 하지 못하도록 한다. 이러한 문제를 다루기 위해서 스케일링 방법이 제안되고 연구되었는데[20], 본 연구에서는 시그마 scaling 방법을 사용하기로 한다, 이 방법은 선택 강도(다시 말하면, 적합도가 높은 개체들이 많은 자손을 생산하도록 허용되는 정도)가 개체집단내의 적합도 분산에 의존하기보다는 차라리 실행의 과정에 걸쳐서 상대적으로 일정하게 해준다.

5. 수치 실험

예제 1. 먼저 Mallya[19]의 다섯 가지 제품에 대한 문제를 복합 유전자 알고리즘을 이용하여 해를 찾아보도록 한다. Mallya[19]의 데이터는 표 1과 같다.

표 1. Mallya의 데이타

제품	p_i	d_i	S_i	A_i	c_i
1	1800	474	0.20	80	0.00379
2	2500	413	0.35	140	0.00252
3	4000	528	0.15	60	0.00391
4	3200	985	0.25	100	0.00282
5	1500	166	0.15	60	0.00108

여기서 $h_i = ic_i$ 로써 i 값은 0.35으로 두고 있다. 따라서 c_i 값과 i 값을 알면 h_i 를 구할 수 있다. 먼저 Mallya[19]의 데이터를 가지고 Dobson[1]의 발견적 기법에 적용하여 보면 그 때의 제품의 생산빈도는

$$y_1=2,\ y_2=2,\ y_3=4,\ y_4=2,\ y_5=1$$

이고, 이 때의 총비용은 $61.63이다. 복합 유전자 알고리즘을 이용하기 위한 생산빈도수는

$y_1=2,\ y_2=2,\ y_3=3,\ y_4=3,\ y_5=1$
이고, 총비용은 $60.91으로써 Dobson[1]의 발견적 기법보다 1.1%의 비용이 낮음을 알 수 있다. 본 연구에서 제시한 복합 유전자 알고리즘으로 Mallya[19]의 문제를 푼 결과 Dobson[1]의 해보다 1.1%의 비용을 개선하였다. 이 1.1%의 비용은 Dobson[1]의 발견적 기법에서 나온 비용이 하한 값과 상당히 근접하다는 사실로 볼 때 상당히 의미있는 값이라고 할 수 있다.

예제 2. 경제적 로트 일정계획문제에서 사용되는 대부분 문제가 Bomberger[21]의 10개의 제품에 대한 문제이다. 본 예제에서는 부하가 아주 많이 걸려있는 설비 즉 x가 0.01인 경우에 대해서 알아보기로 한다.

먼저 공통 주기 방법에 기초를 둔 Khouja 외 2인[11]의 유전자 알고리즘의 결과는 하루 $231.44의 비용이 드는 반면, 본 연구에서 제안한 복합 유전자 알고리즘을 이용하면 하루 $125.81의 비용이 든다. 또한 이 비용은 시변 로트 크기 방법을 이용한 Dobson[1]의 발견적 기법보다도 좋은 것을 알 수 있다.

표 2. Bomberger의 데이터에 대한 결과
(x=0.01 case) 단위 : ($/일)

알고리즘	비 용
Lower Bound	$122.96
복합 유전자 알고리즘	$125.81
Dobson의 발견적 기법	$128.43
Khouja 외 2인의 유전자 알고리즘	$231.44
공통 주기 방법	$196.14

본 연구에서 제시한 복합 유전자 알고리즘에 대한 실험은 Dobson[1]의 발견적 기법에서 나온 해와 비교를 하였으며, 알고리즘에 필요한 데이터들은 표 3에 나타나 있다. 모든 데이터들은 주어진 범위내에서 일양분포에 의해서 임의로 추출되었으며, 총 50문제에 대해서 실험을 하였다. 이미 알려진 바대로 경제적 로트 일정계획은 x가 적은 값일 때 의미가 있으며 실제로는 풀기가 더 어려워진다. 따라서 본 문제에서는 x값이 0.01보다 적은 데이터만을 사용하였다. 본 연구에서는 실제로 두 종류의 데이터 파일이 사용되는데, 먼저 표 3에서는 일반적인 경제적 로트 일정계획 문제에서 필요한 데이터가 나타나 있다. 두 번째 종류의 데이터는 유전자 알고리즘을 실행하기 위해서 필요한 매개변수들의 값들로써, 본 연구에서 사용하는 데이터는 아래와 같다.

- 모집단의 크기 : 100
- 엘리티스트 전략(Elitist Strategy: 한 세대에서 제일 우성인 염색체는 다음 세대로 그대로 복사됨)
- 염색체의 유전인자들은 중복되지 않는 숫자로 이루어짐.
- 세대횟수가 1000번(실제 대부분의 경우 300세대에서 종료함)을 넘어가거나 한 세대에서 제일 좋은 해가 150번의 세대를 거쳐도 더 좋은 결과가 나오지 않는 경우에 프로그램을 종료한다.
- 교배율(crossover rate) : 0.9
- 돌연변이율(mutation rate) = 1/(염색체의 길이=제품의 생산빈도수의 합)

표 3. 모수의 발생범위

모 수	범 위
제품의 개수(개)	[5, 15]
생산율(개/일)	[2000, 20000]
수요율(개/일)	[1500, 2000]
생산 준비 시간(일)	[1, 4]
생산 준비 비용($)	[50, 100]
재고 유지 비용($/개/일)	[1/240, 6/240]

표 4에서는 Dobson[1]의 발견적 기법에서 나온 비용과 복합적 유전자 알고리즘에서 나온 비용에 대해서 각각 하한 값으로부터 나온 비용을 서로 비교하였다. 표 4에서 보면, Dobson[1]의 발견적 기법에서보다 복합 유전자 알고리즘이 더 좋은 결과를 보이는 것을 알 수 있다. 실제 총 50 문제 중에서 38 문제에서 복합 유전자 알고리즘이 더 나은 해를 보였다. 표 5에서는 Dobson[1]의 발견적 기법에서 나온 비용에 대한 유전자 알고리즘의 비용의 비율을 나타내었다. 평균 비율이 1을 넘는 것으로 보아, 유전자 알고리즘에서 나온 비용이 Dobson[1]의 발견적 기법에서 나온 비용보다는 적다는 것을 알 수 있다.

표 4. Dobson의 발견적 기법과 복합 유전자 알고리즘의 비교(I)

	Mean Ratio	Minimum Ratio	Maximum Ratio
Dobson의 발견적 기법	1.0424	1.0112	1.0889
복합 유전자 알고리즘	1.0302	1.0122	1.0564

표 5. Dobson의 발견적 기법과 복합 유전자 알고리즘의 비교(II)

Mean Ratio	Minimum Ratio	Maximum Ratio
1.0119	0.9835	1.0567

6. 결론

최근 수십 년간 경제적 로트 일정계획에 대해서 수많은 연구가 이루어졌다. 경제적 로트 일정계획 문제는 일반적인 일정계획문제에서 자주 발생하는 상황을 고려한 아주 현실적인 문제이다. 하지만 문제의 비선형화와 조합성 때문에 최적의 해를 찾기 어려운 문제이다. 따라서 대부분의 연구는 최적의 해와 가까운 해를 구하는 발견적 기법을 구하는 방법에 초점을 맞추어 왔다. 또한 이러한 발견적 기법에

서 나온 해의 척도는 하한치와 항상 비교되어 평가된다. 본 연구에서는 지금까지 경제적 로트 일정계획문제에 대한 발견적 기법들중 가장 좋은 결과값을 제공하는 Dobson[1]의 발견적 기법을 개선하는 복합 유전자 알고리즘을 제시하고, 실제 그 결과값을 기존의 발견적 기법과 비교해 보았다. 그 결과 본 연구에서 제시하는 복합 유전자 알고리즘이 대부분의 경우 Dobson[1]의 발견적 기법보다 나은 해를 제공함을 알 수 있었다.

7. 참고문헌

[1] Dobson, G. (1987) The economic lot scheduling problem: achieving feasibility using time-varying lot sizes, *Operations Research*, Vol. 35, pp. 764-771.

[2] Boctor, F. (1987) The G-group heuristic for single machine lot scheduling, *International Journal of Production Research*, Vol. 25, pp. 363-379.

[3] Hsu, W. (1983) On the general feasibility test for scheduling lot sizes for several products on one machine, *Management Science*, Vol. 29, pp. 93-105.

[4] Eilon, S. (1957) Scheduling for batch production, *Institute of Production Engineering Journal*, Vol. 36, pp. 549-579.

[5] Rogers, J. (1958) A computational approach to the economic lot scheduling problem, *Management Science*, Vol. 4, pp. 264-291.

[6] Hanssmann, F. (1962) *Operations Research in Production and Inventory Control*, John Wiley and Sons, New York.

[7] Jones, P. and Inman, R. (1989) When is the economic lot scheduling problem easy? *IIE Transactions*, Vol. 21, pp. 11-20.

[8] Gallego, G. (1990) An extension to the class of easy economic lot scheduling problem easy?, *IIE Transactions*, Vol. 22, pp. 189-190.

[9] Doll, L. and Whybark, C. (1973) An iterative procedure for the single machine, multi-product lot scheduling problem, *Management Science*, Vol. 20, pp. 50-55.

[10] Elmaghraby, S. (1978) The economic lot scheduling problem (ELSP): review and extensions, *Management Science*, Vol. 24, pp. 587-598.

[11] Khouza, M., Michalewicz, Z. and Wilmot, M. (1998) The use of genetic algorithms to solve the economic lot size scheduling problem, *European Journal of Operational Research*, Vol. 110, pp. 509-524.

[12] Maxwell, W. (1964) The scheduling of economic lot sizes, *Naval Research Logistics Quarterly*, Vol. 11, pp. 89-124.

[13] Delporte, C. and Thomas, L. (1978) Lot sizing and sequencing for N products on one facility, *Management Science*, Vol. 23, pp. 1070-1079.

[14] Zipkin, P. (1991) Computing optimal lot sizes in the economic lot scheduling problem, *Operations Research*, Vol. 39, pp. 56-63.

[15] Gallego, G. and Roundy, R. (1992) The economic lot scheduling problem with finite backorder costs, *Naval Research Logistics*, Vol. 39, pp. 729-739.

[16] Dobson, G. (1992) The cyclic lot scheduling problem with sequence-dependent setups, *Operations Research*, Vol. 40, pp. 736-749.

[17] Roundy, R. (1989) Rounding off to powers of two in continuous relaxation of capacitated lot sizing problems, *Management Science*, Vol. 35, pp. 1433-1442.

[18] GAlib: A C++ Library of Genetic Algorithm Components. Version 2.4.3 Documentation on the internet (http://lancet.mit.edu/ga/).

[19] Mallya, R. (1992) Multi-product scheduling on a single machine: A case study, *Omega*, Vol. 20, pp. 529-534.

[20] Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York.

[21] Bomberger, E. (1966) A dynamic programming approach to a lot size scheduling problem, *Management Science*, Vol. 12, pp. 778-784.