經營科學
第30卷 第1號
2013年 3月, pp.15-24
http://dx.doi.org/10.7737/KMSR.2013.30.1.015

복구조정 활동과 복구조정 후 시간경과에 따라 퇴화하는 작업시간을 갖는 단일기계의 일정계획에 관한 연구

김병수* · †주철민**

A Study on Single Machine Scheduling with a Rate-Modifying Activity and Time-Dependent Deterioration After the Activity

Byung Soo Kim* · †Cheol Min Joo**

■ Abstract ■

We consider the single machine scheduling problem with a rate-modifying activity and time-dependent deterioration after the activity. The class of scheduling problems with rate-modifying activities and the class of scheduling problems with time-dependent processing times have been studied independently. However, the integration of these classes is motivated by human operators of tasks who has fatigue while carrying out the operation of a series of tasks. This situation is also applicable to machines that experience performance degradation over time due to mal-position or mal-alignment of jobs, abrasion of tools, and scraps of operations, etc. In this study, the integration of the two classes of scheduling problems is considered. We present a mathematical model to determine job-sequence and a position of a rate-modifying activity for the integration problem. Since the model is difficult to solve as the size of real problem being very large, we propose genetic algorithms. The performance of the algorithms are compared with optimal solutions with various problems.

Keywords : Rate-Modifying Activity, Time-Dependent Deterioration, Scheduling, Genetic Algorithm

논문접수일 : 2012년 08월 30일 논문수정일 : 2012년 10월 30일 논문게재확정일 : 2012년 12월 07일

* 부경대학교 기술경영(MOT) 일반대학원
** 동서대학교 산업경영공학과
† 교신저자, iekbs@pknu.ac.kr

1. 서 론

일반적으로 일정계획문제들은 작업에 따라 처리시간이 항상 일정하다고 가정한다. 그러나 현실 상황에서의 작업처리시간은 기계의 가공시간이 경과함에 따라 작업물의 위치이탈, 공구의 마모, 혹은 가공과정에서의 찌꺼기 발생 등에 의해 작업처리시간이 점차 증가하는 것이 일반적이다. 이와 같이 기계의 가공시간이 경과함에 따라 작업처리시간이 증가하는 문제를 퇴화(Deterioration)를 고려한 생산일정계획 문제라고 한다. 이러한 기계의 퇴화는 정비(Maintenance) 및 청소(Cleaning) 작업등을 통해 작업처리시간을 원상태로 회복시켜줄 수 있다. 기계와 같은 자원(Resource)의 작업 처리시간을 원상복귀 또는 향상시키는 행위를 복구조정 활동(Rate Modifying Activity : RMA)라고 한다[12].

Gupta and Gupta[8]와 Browne and Yechiali[4]이 최초로 퇴화를 고려한 생산일정계획 문제를 제시한 이후로 지금까지 관련 연구들이 지속적으로 증가하고 있다[2, 6]. Alidaee and Womer[1]는 작업처리시간의 증가(퇴화)를 작업시작시간의 함수로 가정하고 그 함수를 선형, 단계적 선형, 그리고 비선형로 분류하여 관련 연구들을 조사하였다. 그들이 조사한 퇴화함수 유형들은 이후의 퇴화를 고려한 대부분의 단일기계 일정계획에 관한 관련 연구들의 퇴화함수의 형태로 사용되어 왔다. Cheng and Ding[5]은 납기와 작업처리속도의 퇴화를 고려한 최대작업시간(Makespan)을 최소화하는 단일기계 일정계획문제에 대해 동적계획법(Dynamic programming) 알고리즘을 제시하였다. Bachman et al.[2]은 작업처리시간이 작업시작시간 비례하여 선형으로 증가(퇴화)하는 경우, 총 가중 작업완료시간을 최소화 하는 문제의 NP-hardness를 증명하였다. Jeng and Lin[12]은 작업처리시간이 작업처리시간의 함수이고 각 작업의 납기가 동일한 상황에서 총 작업처리시간을 최소화하는 문제의 단일기계 일정계획문제를 다루었다. Wu et al.[24]은 작업처리시간이 단계적 선형(Piecewise linear) 퇴화하는 경우, 최대작업시간(Makespan)을 최소화하는 단일기계 일정계획 문제에 대해 Branch and Bound 해법과 두 가지의 휴리스틱들을 제시하였다. Wang and Wang[22]과 Husang et al.[10]은 작업처리시간이 작업시작시간 비례함수로 가정하고 총 가중 조기작업완료시간(Earliness)을 최소화하는 단일기계 일정계획문제를 다루었다. Wang et al.[23]은 작업시작시간 종속적인 퇴화를 고려한 총 작업완료시간을 최소화하는 단일기계 일정계획문제에 대해 V형태 최적해 특성[20]을 이용한 두 가지의 휴리스틱을 제안하고 그 성능을 SPT(Shortest processing time) 법칙을 이용한 지역해 탐색 휴리스틱과 비교하였다. Bahalke et al.[3]은 작업의 순서에 종속적인 작업준비시간과 작업의 퇴화를 고려한 최대작업시간(Makespan)을 최소화하는 단일기계 일정계획문제에 대해 하이브리드 타부서치 알고리즘과 유전알고리즘을 제시하였다.

Lee and Leon[17]이 최초로 RMA를 고려한 단일기계 생산일정계획문제를 연구하였다. 그들은 RMA를 고려한 일정계획 문제를 4개의 목적함수로 분류하고, 각 문제에 대한 최적해의 성질들을 규명하여 동적계획법(Dynamic programming)을 이용한 알고리즘을 제시하였다. Qi et al.[21]는 복수의 예방정비를 고려한 배치생산방식의 단일기계 일정계획문제를 연구하였다. Lee and Lin[15]은 확률적 프로세스(Stochastic process)에 의해 발생되는 단일기계의 고장에 의한 사후정비와 예방정비를 동시에 고려한 작업순서 결정에 관한 연구를 수행하였다. Mosheiov and Sarig[19]는 단일 납기가 주어진 작업들의 일정과 RMA 시점을 동시에 결정하는 문제를 다루었다. Lee and Chen[16]은 계획기간 내에 모든 기계에 한번의 RMA가 수행된다는 제약 하에 병렬기계 일정계획 문제를 다루었다. Zhao et al.[26]은 두 개의 병렬기계에 한번의 RMA를 고려한 총 완료시간을 최소화하는 생산일정계획 문제를 연구하였다. Iranpoor et al.[11]는 작업 순서에 따라 달라지는 작업준비시간을 갖는 단일기계의 생산일정계획 문제에 단일 RMA를 고려한 문제를 연구하였다.

이상의 연구들에서는 기계 가공시간의 경과에 따른 작업처리속도의 퇴화만을 고려하거나, 작업처리속도의 퇴화는 고려하지 않고 RMA만을 고려하고 있다. 퇴화와 RMA는 생산일정계획에 상호 밀접한 연관성과 중요성을 가지고 있으나, 퇴화와 RMA 두 가지를 동시에 고려하는 생산일정계획 문제에 관한 연구는 많지 않다. Lodree and Geiger [18]는 작업처리시간이 작업시작시간에 비례하여 선형으로 증가(퇴화)하는 단일기계 생산일정계획에 단일 RMA를 고려한 연구를 수행하였다. 이들은 주어진 문제의 특수한 경우들에 대한 최적해의 성질들을 규명하고, 최적 RMA의 위치와 작업의 순서 결정에 이용하였다. Zhao and Tang[25]은 단일 납기창(common due-window)하에 단일 RMA와 작업의 퇴화를 동시에 고려한 단일기계의 생산일정계획을 연구하였다. 이들은 납기창을 위반하는 비용을 최소화 하는 일정계획을 위한 최적해의 성질들을 경우(Case)별로 분류하여 규명하였다. Lodree and Geiger[18]와 Zhao and Tang[25]의 연구에서 RMA는 모든 작업의 표준 작업처리시간을 개선하는 활동이라 정의하고, 퇴화는 초기부터 해당 작업의 작업시작시간까지 지속적으로 발생한다고 가정하였고, 모든 작업에 동일한 퇴화율을 적용하였다. Kim and Joo[14]는 RMA를 작업 처리시간을 원상 복귀하는 활동이라 정의하고, RMA와 RMA 이후 작업순서에 따라 비선형으로 퇴화하는 작업시간을 동시에 고려한 단일기계 생산일정계획에 관한 연구를 수행하였다.

본 연구에서는 Kim and Joo[14]와 같이 작업 처리시간을 원상 복귀하는 활동이라 정의하는 단일 RMA를 고려하고, 작업처리시간이 초기 및 RMA이후의 경과시간에 비례하여 증가(퇴화)하는 단일기계 생산일정계획문제를 다룬다. 초기 및 RMA 직후의 최초 작업은 정상적인 작업처리시간을 가지나, 작업시작시점이 초기 및 RMA 시점과 차이가 발생하면 그 경과시간에 따라 정상적인 작업처리시간에 비해 증가(퇴화)하는 작업처리시간을 갖는다. 본 연구에서는 퇴화율도 작업에 따라 다르게 주어진다고 가정한다. 이는 기존의 모든 작업의 표준 작업처리시간을 동시에 개선하는 RMA와 작업처리시간이 초기부터 해당 작업의 작업시작시간에 따라 퇴화하는 경우[18, 25]와 작업 순서에 따라 퇴화하는 경우[14] 보다 현실적이다.

본 연구의 목적은 총 작업시간(Makespan)을 최소화하기 위한 작업일정과 RMA 위치(시점)을 동시에 결정하는 것이다. 제 2장에서 주어진 단일 RMA와 작업의 퇴화를 동시에 고려한 단일기계의 생산일정계획 문제에 대한 최적화 수리모형을 제시하였다. 제 3장에서 해의 효율적인 도출을 위해 서로 다른 염색체 표현을 갖는 두 가지 유전알고리즘을 제안하고, 제 4장에서는 제시한 알고리즘들의 성능을 임의로 생성된 예제를 통해 평가하였다.

2. 혼합 정수모형

단일 RMA를 가지고 작업처리시간이 초기 및 RMA이후의 경과시간에 따라 비례하여 증가(퇴화)하는 단일기계 생산일정계획문제에 대해 총 작업시간(Makespan)을 최소화하는 작업일정과 RMA 위치(시점)를 동시에 결정하는 혼합정수계획 모형(Mixed Integer Programming Model)은 다음과 같다.

<파라미터>

N : 작업수

p_i : 작업 i의 초기 작업처리시간

r_i : 작업 i의 작업처리시간 증가(퇴화)율

Q : RMA에 걸리는 시간

M : 양의 큰 수(무한대)

k : 작업 영역을 나타내는 인덱스(1 : RMA 이전 영역, 2 : RMA 이후 영역)

<의사결정 변수>

x_i : 직전 RMA와 작업 i의 작업시작시간 사이의 차이

C_k : 작업영역 k의 총 작업완료시간

$$y_{ik}=\begin{cases}1 & \text{작업 } i\text{가 작업영역 } k\text{에서 처리되어지면}\\ 0 & \text{otherwise}\end{cases}$$

$$z_{ijk}=\begin{cases}1 & \text{작업영역 } k\text{에 작업 } i\text{와 } j\text{가 할당되고,}\\ & \text{작업 } i\text{가 } j\text{보다 먼저 처리되면}\\ 0 & \text{otherwise}\end{cases}$$

<혼합정수계획 모형>

$$Min\ \ z=\sum_{k=1}^{2}C_k+Q \tag{1}$$

$$s.t.\ \ x_i+p_i+r_i\cdot x_i\le x_j+M\cdot\left(1-\sum_{k=1}^{2}z_{ijk}\right)\quad \text{for } \forall i,j \text{ and } i\neq j \tag{2}$$

$$x_i+p_i+r_i\cdot x_i\le C_k+M\cdot(1-y_{ik})\quad \text{for } \forall i,k \tag{3}$$

$$\sum_{k=1}^{2}y_{ik}=1\quad \text{for } \forall i \tag{4}$$

$$\sum_{i=1}^{N}z_{iik}=1\quad \text{for } \forall k \tag{5}$$

$$\sum_{j=1}^{N}z_{jik}=y_{ik}\quad \text{for } \forall i,k \tag{6}$$

$$\sum_{\substack{j=1\\ j\neq i}}^{N}z_{ijk}\le y_{ik}\quad \text{for } \forall i,k \text{ for } \forall k \tag{7}$$

$$x_i\ge 0\quad \text{for } \forall i \tag{8}$$

$$C_k\ge 0\quad \text{for } \forall k \tag{9}$$

$$y_{ik}=0 \text{ 또는 } 1\quad \text{for } \forall i,j \tag{10}$$

$$z_{ijk}=0 \text{ 또는 } 1\quad \text{for } \forall i,j,k \tag{11}$$

제약식 (2)는 동일 작업영역상의 작업은 작업의 선후관계를 고려하여 각 작업의 시작시간을 제약하며, 제약식 (3)은 각 작업영역에 할당된 모든 작업의 총 작업완료시간을 계산하기 위한 제약식이다. 제약식 (4)는 각 작업은 반드시 각 작업영역 중 한 곳에만 할당되어야 됨을 나타낸다. 제약식 (5)는 각 작업영역에 할당된 모든 작업들 중 첫 번째 순서에 반드시 하나의 작업이 존재해야 함을 나타내며, 제약식 (6)은 특정 작업영역에 할당된 작업이 존재한다면, 그 작업의 선행 작업은 반드시 한 개만 존재해야 함을 제약한다. 제약식 (7)은 만약 특정 작업영역에 할당된 작업이 존재한다면, 그 작업의 후속작업은 1개 이상 존재하지 않음을 제약한다.

3. 유전알고리즘

제 2장에서 제시한 혼합정수계획모형은 RMA이전과 RMA이후에 평균적으로 5개의 작업이상이 할당되는 문제의 경우 CPLEX 등의 상용 패키지를 사용하여 제한된 시간 내에 최적해를 구하기 어렵다. 따라서 본 연구에서는 규모가 큰 경우의 문제들을 위한 효과적인 유전알고리즘(Genetic algorithm) 개발에 초점을 맞추었다.

일반적으로 유전알고리즘은 생물의 진화과정인 적자생존의 원칙을 활용해 해를 탐색해가는 메타휴리스틱 기법의 하나로 Holland[9]에 의해 처음 소개된 후로 다양한 문제에 적용되었다. 보편적인 탐색기법들과는 달리 유전알고리즘은 염색체(Chromosome)라 불리는 해들로 이루어진 해집단(Population)을 하나의 세대(Generation)로 정의하고, 적합도(Fitness)라는 평가척도를 이용하여 다음 세대로의 진화(Evolution)를 반복함으로써 해를 탐색하게 된다. 유전알고리즘은 생산일정계획과 같은 조합(Combinatorial) 문제에 대해 비교적 짧은 시간 내에 근사 최적해를 도출하는 것으로 알려져 있다[7]. 또한, 본 연구에서 제시된 문제에서 염색체 구조가 비슷한 유전알고리즘을 이용한 병렬기계 생산일정계획 연구에 의해 알고리즘의 효율성을 입증하였다[13]. 따라서 본 연구에서는 효율적인 해의 도출을 위해 병렬기계일정계획 문제에서 일반적으로 많이 소개되는 2차원 배열(Double String) 염색체 표현을 이용한 2차원 배열 유전알고리즘(Genetic Algorithm with Double String : GA_DS)과 특수문자(Special Character)를 사용한 염색체 표현방법을 사용한 특수문자 유전 알고리즘(Genetic Algorithm with Special Character : GA_SC)의 두 가지 유전알고리즘을 제시한다. 2차원 배열을 이용한 염색체 표현 방법은 첫 번째 배열은 n개의 1부터 n까지의 정수를 이용하여 작업의 순서를 나타내고, 두 번째

배열은 n개의 RMA 전후를 의미하는 이진표현(0 혹은 1)을 이용하여 RMA 전후 할당을 나타낸다. 특수 문자를 이용한 염색체 표현 방법은 RMA 전후를 구분하는 1개의 특수문자 *와 n개의 1부터 n까지의 정수를 이용한 $(n+1)$ 1차원 배열 방법을 이용하여 해를 표현한다. 여기서, 특수문자 * 전후의 숫자들은 각각 RMA 전후에 가공되는 작업들의 순서를 의미한다. 따라서 두 가지 염색체 표현방법들 모두 RMA의 위치 및 RMA 전후에 할당된 작업들의 작업순서를 동시에 결정하게 된다.

<그림 1>은 동일한 해에 대한 두 가지 염색체 표현을 예시하고 있다. <그림 1>(a)는 2차원 배열을 이용한 염색체 표현으로 첫 번째 배열은 작업의 순서 및 선후관계를 결정하고 두 번째 배열은 해당 작업의 RMA 전후 할당을 결정한다. 여기서 0은 작업이 RMA 이전에 할당될 경우를, 1은 해당 작업이 RMA 이후에 할당될 경우를 의미한다. <그림 1>(b)는 특수문자를 이용한 염색체 표현이다. 두 가지 염색체 표현 모두 작업 2, 5, 7, 4, 그리고 9가 RMA 이전에 작업 10, 1, 6, 8, 그리고 3이 RMA 이후에 순차적으로 할당되는 일정계획을 나타낸다. RMA 전과 후에 최초로 할당되는 작업 2와 10은 작업처리시간이 증가(퇴화)되지 않지만, 이후 작업들은 각 최초작업 대비 작업 시작시간과 작업별 증가율에 따라 작업처리시간이 증가(퇴화)하게 된다.

초기세대의 해집단을 형성하기 위한 초기해들은 임의(Random)로 생성된다. 해집단(Population) 내의 염색체들은 적합도(Fitness) 함수로써 총 작업완료시간을 갖는다. 일반적으로 유전알고리즘은 재생산(Reproduction), 교차변이연산(Crossover), 돌연변이연산(Mutation)의 세 가지 유전 작업들을 이용하여 유전형질을 다음세대로 계승한다. 교차변이연산으로 하나의 교차점을 임의로 선택하고 두 부모의 유전자를 교차하여 자식을 생성하는 한점교차(One cut-point crossover)방법을 사용한다. 한점교차변이는 일정계획 문제의 염색체 표현에서 가장 보편적으로 이용되는 방법으로 실행불가능(Infeasible)해를 발생시키지 않고 다양한 해를 탐색하는 방법이다. <그림 2>(a)와 <그림 2>(b)는 2차원 배열구조를 이용한 염색체의 한점교차 변이를 나타내고 있다. <그림 2>(c)에서는 특수문자를 이용한 염색체의 한점 교차변이를 나타내고 있다.

돌연변이연산으로는 <그림 3>와 같이 부모의 유전자에서 임의로 선택된 두 점을 상호 교환하여 자식을 생성하는 교환 돌연변이(Swap-mutation)방법을 사용한다.

다음 세대를 구성하기 위한 자식 해는 기본적으로 임의로 짝 지워진 부모 해들을 교차변이율 P_c와 돌연변이율 P_m과 [0, 1]사이의 난수 *R1, R2*에 따라 세 가지 유전 작업을 통해 재생산 된다. 생성된 자식들 중 룰렛휠 방식에 따라 선택된 해집단으로 다음세대를 구성하고, 최대세대수에 도달할 때 까지 반복 수행한다. <그림 4>는 제안된 GA_DS와 GA_SC의 알고리즘 전체흐름도이다.

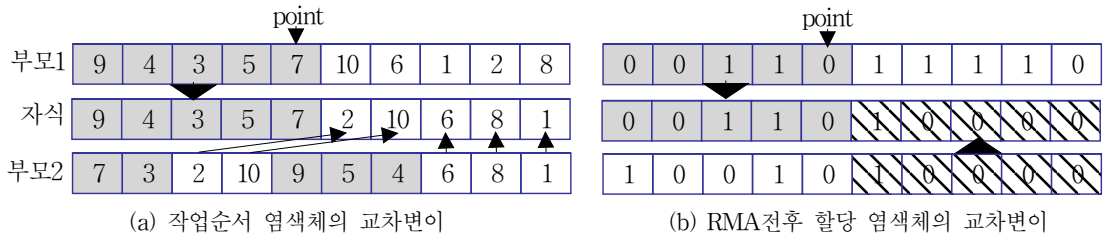
작업순서	2	5	10	1	7	6	8	4	3	9
RMA 전후 할당	0	0	1	1	0	1	1	0	1	0

(a) 2차원 배열을 이용한 염색체 표현

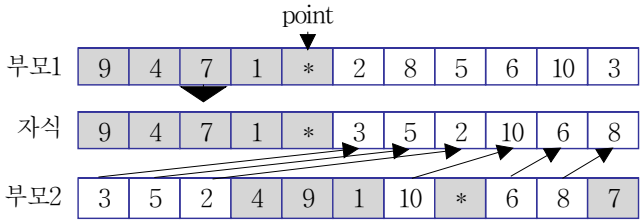
2	5	7	4	9	*	10	1	6	8	3

(b) 특수문자를 이용한 염색체 표현

〈그림 1〉 두 가지 염색체 표현

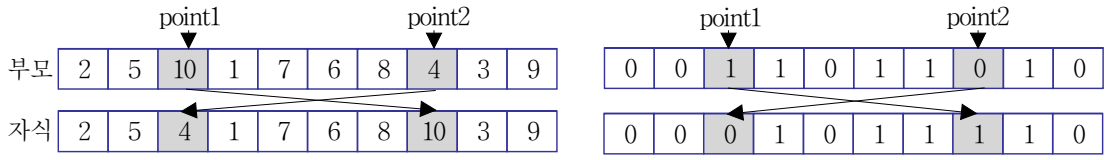


(a) 작업순서 염색체의 교차변이 (b) RMA전후 할당 염색체의 교차변이

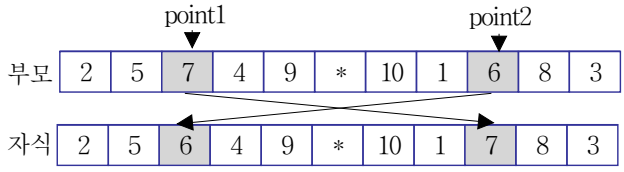


(c) 특수문자를 이용한 염색체의 교차변이

〈그림 2〉 한점 교차변이



(a) 작업순서 염색체의 돌연변이 (b) RMA전후 할당 염색체의 돌연변이



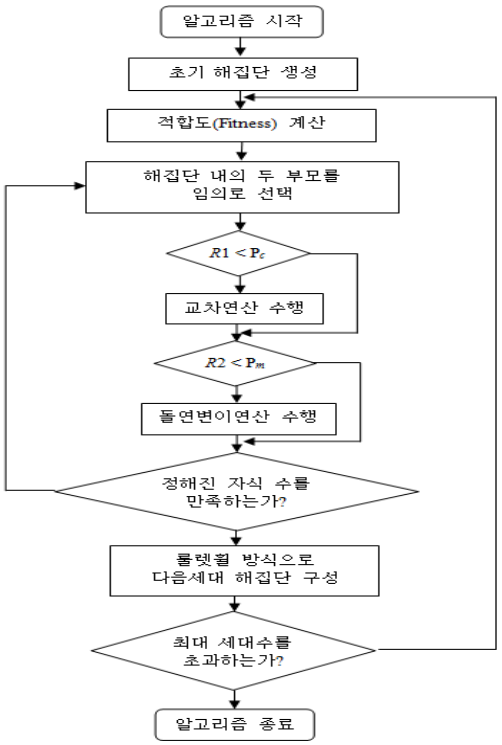
(c) 특수문자를 이용한 염색체의 돌연변이

〈그림 3〉 교환 돌연변이

4. 알고리즘 검증

본 연구에서 제시한 GA_DS와 GA_SC의 성능을 검증하기 위하여 임의로 실험데이터를 생성하였다. 각 작업의 초기 작업시간은 p_i는 [30, 60] 단위시간 구간 내에서 임의로 생성하고 각 작업의 퇴화율 r_i는 [0.1, 0.4] 구간 내에서 임의로 생성하였다. RMA에 걸리는 시간 Q는 생성된 초기 작업시간들의 평균값의 배수로 설정하였다. 일반적으로 일정계획문제의 복잡도는 작업수가 늘어남에 따라 증가하게 되므로, 작업수에 따라 두 가지 분류의 실험을 시행하였다. 첫 번째 분류는 최적해와의 비교를 위하여 총 작업수를 10개 이내로 제한하였으며, 두 번째 분류는 GA_SC과 GA_DS의 상대적 성능평가를 위해 작업 수를 20, 30, 40, 50, 그리고 60개로 증가시켜 실험을 시행하였다.

제 2장에서 제시한 혼합정수모형에 의한 최적해는 ILOG CPLEX 12.4을 적용하여 구하였고 두 가지 유전알고리즘은 C#을 이용하여 구현하였으며, 모든 실험은 1.86GHz Intel Core 2 Processor, 2GB



〈그림 4〉 GA_DS와 GA_SC의 알고리즘 전체흐름도

RAM의 PC을 이용하여 수행되었다. 유전알고리즘 파라메터 설정은 제시된 GA_DS와 GA_SC 모두 해집단의 수를 $2 \cdot n$으로, 총 세대수를 1,000으로 하였고, 교차변이율과 돌연변이율은 여러 가지 조합의 예비 실험결과에서 가장 좋은 성능을 보였던 0.8과 0.2로 수행하였다. GA_DS와 GA_SC의 성능을 평가하기 위한 척도로 해의 상대비율편차(Relative Percentage Deviation : *RPD*)를 이용하였다. *RPD*는 최선해에 대한 알고리즘에 의한 해의 편차의 비를 의미하고 식 (12)로 표현된다.

$$RPD(\%) = \frac{\text{알고리즘해} - \text{최선해}}{\text{최선해}} \times 100 \qquad (12)$$

여기서, *RPD*가 α%란 것은 알고리즘에 의한 해가 최선해(Best Solution : 최선해는 최적해가 될수도 있음)에 비해 α% 초과한다는 것이므로, *RPD*값이 적을수록 알고리즘의 효과(Effectiveness)가 좋다는 것을 의미한다. 또 한 가지 척도로 평균 절대편차(Mean Absolute Deviation : *MAD*)를 이용하였다. *MAD*는 반복실험을 통해 나온 해들의 평균값에 대한 각 해의 절대편차의 비를 의미하고 식 (13)으로 표현되며, 해의 산포 정도를 나타낸다.

$$MAD(\%) = \frac{|\text{해} - \text{해평균}|}{\text{해평균}} \times 100 \qquad (13)$$

<표 1>에 규모가 제한된 첫 번째 분류의 문제들에 대한 실험결과를 정리하였다. 각 실험데이터에 대해 CPLEX에 의한 최적해 및 최적해 탐색 시간과 GA_DS와 GA_SC를 각각 10회씩 실행하여 해의 평균값(*Mean*), 최적해 대비 *RPD*의 평균, *MAD*의 평균, 그리고 해 탐색시간의 평균을 정리하였다.

<표 1>의 실험 결과를 보면 최적해는 작업수가 늘어나면 해 탐색시간이 급격히 늘어나는 것을 알 수 있다. 두 가지의 유전알고리즘인 GA_DS와 GA_SC 모두 대다수의 경우 최적해 탐색하였고, *RPD* 전체 평균도 각각 0.13%와 0.30%로 아주 작아 그 성능이 대단히 우수함을 알 수 있다. 또한, 최적해와의 *MAD*는 GA_DS와 GA_SC 각각 평균 0.10%와 0.18%로 이는 같은 실험 데이터로 여러 번 반복 실행하여도 큰 변화 없이 매번 유사한 해를 찾는다는 것을 의미한다. 첫 번째 분류의 실험결과, GA_DS가 GA_SC보다 *RPD*와 *MAD*에 있어서 상대적으로 우수함을 나타내지만 평균 해 탐색시간에 있어서는 GA_DS가 GA_SC가 각각 0.02초와 0.01초로 GA_SC가 GA_DS보다 덜 소요되는 것으로 나타났다.

상대적으로 규모가 큰(작업수 10개 이상) 두 번째 분류의 문제들은 CPLEX를 이용하여 제한된 시간에 최적해를 찾는 것이 불가능 하므로, <표 2>에는 모든 실험 데이터의 10번의 반복에 대한 GA_DS와 GA_SC의 *RPD*, *MAD*, 그리고 해 탐색시간을 상대 비교하였다. <표 2>의 실험 결과를 보면 GA_DS와 GA_SC의 평균 *RPD*가 각각 3.44%와 7.27%, *MAD*가 각각 3.30%와 6.67%로 GA_DS가 GA_SC보다 상대적으로 우수함을 알 수 있다. 하지

〈표 1〉 최적해와 GA_DS해와 GA_SC해의 성능비교

실험데이터		CPLEX		GA_DS				GA_SC			
번호	작업수	최적해	Time (Sec.)	*Mean*	*RPD* (%)	*MAD* (%)	Time (Sec.)	*Mean*	*RPD* (%)	*MAD* (%)	Time (Sec.)
1	6	320.06	1.421	320.06	0.00	0.00	0.01	320.07	0.00	0.01	0.01
2	6	394.44	1.702	394.44	0.00	0.00	0.01	394.44	0.00	0.00	0.01
3	6	367.46	1.190	367.46	0.00	0.00	0.01	367.47	0.00	0.00	0.01
4	7	483.11	2.796	483.11	0.00	0.00	0.01	483.38	0.05	0.09	0.01
5	7	496.35	2.774	496.35	0.00	0.00	0.01	497.02	0.14	0.12	0.01
6	7	460.13	2.260	460.20	0.01	0.02	0.01	460.28	0.03	0.04	0.01
7	8	544.66	9.329	544.83	0.03	0.04	0.02	545.80	0.21	0.15	0.01
8	8	515.39	8.819	516.07	0.13	0.16	0.02	516.75	0.26	0.23	0.01
9	8	521.39	8.214	521.78	0.07	0.12	0.02	522.67	0.25	0.24	0.01
10	9	615.97	70.800	617.97	0.32	0.23	0.02	618.94	0.48	0.26	0.02
11	9	766.19	70.373	768.48	0.30	0.10	0.02	770.59	0.57	0.17	0.02
12	9	696.37	72.105	697.95	0.23	0.15	0.02	699.33	0.42	0.26	0.02
13	10	673.08	1432.979	675.59	0.37	0.29	0.02	675.57	0.37	0.26	0.02
14	10	778.47	1641.303	780.37	0.24	0.19	0.02	785.08	0.85	0.49	0.02
15	10	712.73	1609.133	714.33	0.22	0.16	0.02	719.00	0.88	0.31	0.02
평균				557.27	0.13	0.10	0.02	558.42	0.30	0.18	0.01

〈표 2〉 GA_DS해와 GA_SC해 간의 상대적 성능비교

실험데이터		최선해	GA_DS				GA_SC			
번호	작업수		*Mean*	*RPD* (%)	*MAD* (%)	CPU time	*Mean*	*RPD* (%)	*MAD* (%)	CPU time
1	20	2331.35	2361.59	1.30	1.28	0.11	2398.33	2.87	2.79	0.08
2	20	2202.27	2234.79	1.48	1.46	0.10	2269.24	3.04	2.95	0.07
3	20	2375.89	2406.81	1.30	1.28	0.09	2430.74	2.31	2.26	0.07
4	30	9372.55	9590.24	2.32	2.27	0.24	9833.08	4.91	4.68	0.16
5	30	7983.85	8380.66	4.97	4.73	0.23	8516.90	6.68	6.26	0.16
6	30	7336.67	7502.44	2.26	2.21	0.25	7722.65	5.26	5.00	0.18
7	40	29038.70	30597.41	5.37	5.09	0.50	31715.05	9.22	8.44	0.34
8	40	34741.77	35678.58	2.70	2.63	0.49	36597.48	5.34	5.07	0.34
9	40	19336.40	19985.21	3.36	3.25	0.47	20861.86	7.89	7.31	0.32
10	50	45024.73	46968.45	4.32	4.14	0.82	50471.34	12.10	10.79	0.55
11	50	94132.74	96665.11	2.69	2.62	0.83	99018.74	5.19	4.93	0.57
12	50	95221.33	98824.35	3.78	3.65	0.81	103401.73	8.59	7.91	0.56
13	60	143911.89	153648.11	6.77	6.34	1.32	161486.97	12.21	10.88	0.87
14	60	162285.30	170634.54	5.14	4.89	1.31	185286.92	14.17	12.41	0.88
15	60	132080.65	137181.43	3.86	3.72	1.30	144233.33	9.20	8.43	0.87
평균		52491.74	54843.98	3.44	3.30	0.59	57749.62	7.27	6.67	0.40

만, 평균 해 탐색시간에 있어서는 GA_SC와 GA_DS가 각각 0.40초와 0.59초로 GA_SC가 GA_DS보다 덜 소요되는 것으로 나타났다. 실험으로부터 해의 탐색시간은 총 작업수가 많아질수록 증가폭이 커지고, GA_DS는 염색체표현의 이중성 때문에 GA_SC보다 상대적으로 긴 시간이 소요되나, 모든 실험에 대해 GA_SC와 GA_DR 모두 1초 이내일 정도로 적어 실제문제에 적용시 효율적인(Efficient) 일정계획의 수립이 가능하다고 판단된다.

5. 결 론

본 연구에서는 단일 RMA를 고려하고, 작업처리시간이 초기 및 RMA 이후의 경과시간에 비례하여 증가(퇴화)하는 단일기계 생산일정계획문제를 다루었다. 최적해의 도출을 위해 총 작업시간(Makespan)을 최소화하는 작업일정과 RMA 위치(시점)를 동시에 결정하는 혼합정수계획 최적화 모형을 제시하였다. 최적화 모형을 이용한 최적해 탐색은 기계당 평균 작업수가 10개를 초과하면 현실적으로 제한된 시간에 최적해의 탐색이 어려워진다. 따라서 보다 효율적인 해의 도출을 위해 본 연구에서는 2차원 배열 염색체 표현을 이용한 유전알고리즘(GA_DS)과 특수문자를 사용한 염색체 표현방법을 사용한 유전 알고리즘(GA_SC)의 두 가지 유전알고리즘을 제시하였다. 실험으로부터 GA_DS는 염색체 표현의 복잡성 때문에 GA_SC보다 상대적으로 긴 시간이 소요되나, GA_SC보다 낮은 *RPD*와 *MAD*를 갖는 효과적인 해를 제공한다. 또한, GA_DS와 GA_SC 모두 규모가 큰 문제에도 1초 이내에 해를 탐색하는 탁월한 효율성을 입증하였다.

참 고 문 헌

[1] Alidaee, B. and N.K. Womer, "Scheduling with time dependent processing times : review and extensions," *Journal of* the *Operational Research Society*, Vol.50, No.7(1999), pp.711-721.

[2] Bachman, A., A. Janiak, and M.Y. Kovalyov, "Minimizing the total weighted completion time of deteriorating jobs," *Information Processing Letters*, Vol.81, No.2(2002), pp.81-84.

[3] Bahalke, U., A.M. Yolmeh, and K. Shahanaghi, "Meta-heuristics to solve single-machine scheduling problem with sequence-dependent setup-time and deteriorating jobs," *International Journal of Advanced Manufacturing Technology*, Vol.50(2010), pp.749-759.

[4] Browne, S. and U. Yechiali, "Scheduling deteriorating jobs on a single processor," *Operations Research*, Vol.38(1999), pp.495-498.

[5] Cheng, T.C.E. and Q. Ding, "Single machine scheduling with deadlines and increasing rates of processing times," *Acta Informatica*, Vol.36(2000), pp.673-692.

[6] Cheng, T.C.E., Q. Ding, and B.M.T. Lin, "A concise survey of scheduling with time-dependent processing times," *European Journal of Operational Research*, Vol.152(2004), pp.1-13.

[7] Gen, M. and R. Cheng, *Genetic Algorithms and Engineering Optimization,* New York : Wiley, 2000.

[8] Gupta, J.N.D. and S.K. Gupta, "Single facility scheduling with nonlinear processing times," *Computers and Industrial Engineering*, Vol. 14(1988), pp.387-393.

[9] Holland, J.H., *Adaptation in natural and artificial systems,* Ann Arbor, IL : University of Michigan Press, 1975.

[10] Husang, H., J. Wang, and X. Wang, "A generalization for single-machine scheduling with deteriorating jobs to minimize earliness penalties," I*nternational Journal of Advanced Manufacture Technology*, Vol.47(2010), 1225-1230.

[11] Iranpoor, M., S.M.T. Ghomi, and M. Zandieh, "Machine scheduling in the presence of sequence-dependent setup times and a rate-modifying activity," *International Journal of Production Research*, Forthcoming, 2012.

[12] Jeng, A. and B. Lin, "Minimizing the total completion time in single-machine scheduling with step-deteriorating jobs," *Computers and Operations Research*, Vol.32(2005), pp. 521-536.

[13] Joo, C.M. and B.S. Kim, "Genetic Algorithm with an Effective Dispatching Method for Unrelated Parallel Machine Scheduling with Sequence Dependent and Machine Dependent Setup Times," *IE Interface*, Vol.25, No.3(2012), pp.357-364.

[14] Kim, B.S. and C.M. Joo, "Single-machine total completion time scheduling with position-based deterioration and multiple rate-modifying activities," *Industrial Engineering and Management Systems*, Vol.10, No.4 (2011), pp.247-254.

[15] Lee, C.Y. and C.S. Lin, "Single-machine scheduling with maintenance and repair rate-modifying activities," *European Journal of Operational Research*, Vol.135(2001), pp.493-513.

[16] Lee, C.Y. and Z.L. Chen, "Scheduling of jobs and maintenance activities on parallel machines," *Naval Research Logistics*, Vol.47 (2000), pp.61-67.

[17] Lee, C.Y. and V.J. Leon, "Machine scheduling with a rate-modifying activity," *European Journal of Operational Research*, Vol. 128(2001), pp.119-128.

[18] Lodree, E.J. and C.D. Geiger, "A note on the optimal sequence position for a rate-modifying activity under simple linear deterioration," *European Journal of Operational Research*, Vol.201, No.2(2010), pp.644-648.

[19] Mosheiov, G. and A. Sarig, "Scheduling a maintenance activity and due-window assignment on a single machine," *Computers and Operations Research*, Vol.36 No.9(2009), pp. 2541-2545.

[20] Mosheiov, G., "V-Shaped polices to scheduling deteriorating jobs," *Operation Research*, Vol.39(1991), pp.979-991.

[21] Qi, X., T. Chen, and F. Tu, "Scheduling the maintenance on a single machine," *Journal of the Operational Research Society*, Vol.50 (1999), pp.1071-1078.

[22] Wang, D. and J. Wang, "Single-machine scheduling with simple linear deterioration to minimize earliness penalties," *International Journal of Advanced Manufacture Technology*, Vol.46(2010), pp.285-290.

[23] Wang, J.L., L. Sun, and L. Sun, "Single-machine total completion time scheduling with a time-dependent deterioration," *Applied Mathematical Modelling*, Vol.35(2011), pp. 1506-1511.

[24] Wu, C., Y. Shiau, L. Lee, and W. Lee, "Scheduling deteriorating jobs to minimize the makespan on a single machine," *International Journal of Advanced Manufacture Technology*, Vol.44(2009), pp.1230-1236.

[25] Zhao, C. and H. Tang, "A note to due-window assignment and single machine scheduling with deteriorating jobs and a rate-modifying activity," *Computers and Operations Research*, Vol.39, No.6(2012), pp.1300-1303.

[26] Zhao, C., H. Tang, and C. Cheng, "Two-parallel machines scheduling with rate-modifying activities to minimize total completion time," *European Journal of Operational Research*, Vol.198(2009), pp.354-357.