Machine scheduling of time-dependent deteriorating jobs with determining the optimal number of rate modifying activities and the position of the activities

Cheol Min JOO* and Byung Soo KIM**
*Department of Industrial and Management Engineering, Dongseo University
San 69-1, Jurye-dong, Shah-gu, Busan, 617-716, Republic of Korea
**Department of Industrial and Management Engineering, Incheon National University
119, Academy-ro, Songdo-dong, Yeonsu-gu, Incheon, 406-772, Republic of Korea
E-mail: bskim@incheon.ac.kr

Received 12 August 2014

Abstract
In this paper, we consider a single machine scheduling problems integrating with deterioration effect and rate-modifying activities (RMAs). The scheduling problem assumes that the machine may have multiple RMAs and each job has the processing time with deterioration effect increased depending upon the gap between recent RMA and starting time of the job. In practical industrial applications, the determination of the number and position of RMAs is a critical issue, because the RMAs recover deteriorated processing time of jobs bringing back to normal processing time. In this paper, we simultaneously determine the schedule of time-dependent deteriorating jobs and the number and positions of RMAs to minimize the makespan. To solve the problem, a mathematical model is derived to obtain the optimal solution and hybrid meta-heuristic algorithms are proposed for the problem. The performance of the algorithms is compared using randomly generated examples.

Key words: Machine scheduling, Deterioration, Rate-modifying activities, Hybrid meta-heuristic algorithm, Makespan minimization

1. Introduction

The machine scheduling problems with deteriorating jobs and rate-modifying activities have been received increasing attention in recent years. The machine production rate becomes less than normal due to a mal-position of tools, mal-alignment of jobs, abrasion of tools, and scraps of operations, etc. In these situations, the processing time of the jobs increases depending upon the sequence of jobs or the starting time of jobs. Such problems are generally known as machine scheduling problems with deterioration (Gupta and Gupta, 1988; Brown and Yechiali, 1990). The activity that changes the production efficiency of a machine is called rate-modifying activity (RMA) (Lee and Leon, 2001). Preventive maintenance is a special type of RMA because it brings the actual processing time with deterioration back to a normal processing time. Although the deterioration of jobs and RMAs are interrelated with each other, there are many research works on scheduling problems with deterioration, which include Kang and Ng (2007), Shiau et al. (2007), Ji and Cheng (2008), Lee and Wu (2008), Lee et al. (2009), Wang et al. (2009), Wang(2010), Wang and Wang (2010a), Rudek and Rudek (2012) and scheduling problems with RMAs, which include Lee and Leon(2001), Lee and Lin(2001), Mosheiov and Oron (2006), Gordon and Tarasevich (2009), Zhao et al. (2009), Mosheiov and Sideny (2010), Hsu et al. (2011), Wang and Wang(2010b).

Research on scheduling simultaneously with the deterioration and the RMA has received increasing attraction in recent years. In single machine, Lee and Leon (2001) initially introduced a single machine scheduling with a RMA.
They assumed that the machine may have at most one rate-modifying activity during the planning horizon and the rate-modifying activity can be considered as a special type of scheduling with maintenance activity. They separately classified the scheduling problem to following four objective functions: makespan, total completion time, total weighted completion time, and maximum lateness and each of the cases proposed a polynomial or pseudo-polynomial dynamic programming algorithm after they found optimality properties. He et al. (2005) explored a restricted version of the problem proposed by Lee and Leon (2001). They showed that some related problems become NP-hard with the restriction. Mosheiov and Sarig (2009) analyzed a single machine scheduling determining both sequencing of jobs and positioning of a RMA with common due-window assignment. Lodree and Geiger (2010) studied scheduling problems integrated with time-dependent processing times and single RMA. Under a certain condition, they provided the optimal policy determining optimal position of RMA and sequencing of jobs. Zhao and Tang (2012) considered single machine scheduling problem with positioning of single RMA and deterioration of jobs. Lodree and Geiger (2010) and Zhao and Tang (2012) defined RMA as an activity changing the normal processing times of the jobs following this activity, and assumed that the processing time of a job is a linear function of its starting time and the job-independent deterioration rates are identical for all jobs. In multiple RMAs, Ozturkoglu and Bulfin (2011) investigated single machine scheduling problems with position-based deterioration effects. They formulated a unique integer program to solve for the makespan and the total completion time. Kim and Joo (2011) and Kim and Ozturkoglu (2013) extended the problems of Ozturkoglu and Bulfin (2011) to solve large-sized problems using genetic algorithms. Joo and Kim (2013) studied a single machine scheduling problem with time-dependent deterioration effects and multiple RMAs. They assumed that the deterioration effects increased depending upon the gap between recent RMA and starting time of the job. They derived a mixed integer programming model to solve the optimal solution for the makespan. For large sized problems, they also proposed several genetic algorithms with different chromosome representations.

In parallel machine, there are several papers with single RMA on each machine. Zhao et al. (2009) extended some objectives studied by Lee and Leon (2001) to the two identical parallel-machine environment. They showed that the minimization of the total completion time is solvable in \(O(n^{2m+3})\) time, where \(n\) is the total number of jobs and \(m\) is the number of machines. Wang et al. (2011) also studied an identical parallel-machine scheduling problem with a rate-modifying activity to minimize the total completion time. They assumed that each machine may have at most one rate-modifying activity. They proved that the complexity of the problem is \(O(n^{2m+3})\). Iranpoor et al. (2012) presented a single machine scheduling problem with sequence-dependent setup times and single RMA. Hsu et al. (2011) extended the model studied by Zhao et al. (2009) to an unrelated parallel-machine setting. For given the number of machines, they showed that the minimization of the total completion time is solvable in \(O(n^{m+3})\) if the rate modifying rate of job \(j\) on machine \(i\), \(\delta_{ij}\) is in \(0 < \delta_{ij} \leq 1\) and they also showed that the time complexity of the total completion time minimization problem remains \(O(n^{2m+3})\). For multiple RMAs each machine, Ji and Cheng (2009) investigated scheduling problems with learning effects and multiple rate-modifying activities on an identical parallel-machine problem. They assumed that each machine may have multiple rate-modifying activities during the planning horizon. The objectives were to minimize the total completion time and the total machine load. They showed that both objectives can be solved \(O(n^{m+2+\sum_{k=1}^{\infty} k_i})\) time, where \(k_i\) is the upper bound of the rate-modifying activity frequency on machine \(i\). Yang and Yang (2013) investigated scheduling problems with multiple rate-modifying activities on an unrelated parallel-machine. They proposed two efficient polynomial time algorithms with a fixed number of machines and processing time being restricted. Joo and Kim (2013) proposed genetic algorithms for single machine scheduling problem with the time-dependent deterioration and the RMAs. In the study, they assumed the number of RMAs is assumed to be fixed or predetermined and then the jobs are scheduled based on the RMAs. However, this assumption might not be true. The determination of the number and position of RMAs is a critical issue in practical industrial applications, because it affects the machine production rate, as a result, determines the performance of the makespan of the machine.

To the best of our knowledge, the scheduling problem of deteriorated jobs and determining the number and positions of RMAs has never been studied simultaneously. In this study, we investigate single machine scheduling problem simultaneously determining the number and positions of RMAs and the schedule of jobs with time-dependent deterioration. The time-dependent deterioration of a job depends upon the gap between recent RMA and starting time of the job, and the RMAs are defined as a full recovering process bringing back to a normal processing time. We propose two hybrid meta-heuristic algorithms based on GA and compared with two GA with different chromosome structures.
2. Mixed integer programming model

In this section, a mixed integer programming model is derived for determining simultaneously the schedule of time-dependent deteriorating jobs and the number and positions of RMAs to minimize the makespan. If the working horizons between RMAs are regarded as parallel machines, the scheduling problem with time-dependent deterioration and RMAs are similar to the parallel machine scheduling problem with time-dependent deterioration. So, the mathematical model of the problem is formulated with the similar structure of parallel machine scheduling model. The working horizons between RMAs including horizons before the first and after the last RMAs are called \( \text{buckets} \). Thus, the model is formulated for determining simultaneously the job assignments to buckets and the sequences of assigned jobs in each bucket, along with the number and positions of RMAs to minimize the makespan. The parameters and decision variables in the mathematical model are defined as follows:

<Parameters>

\[ J \] number of independent jobs  
\[ B \] number of potential buckets, so the number of maximum RMAs is \( B-1 \)  
\[ p_j \] original processing time of job \( j \)  
\[ r_j \] time-dependent deterioration rate of job \( j \)  
\[ R \] RMA time

<Continuous decision variables>

\[ x_j \] gap between recent RMA and starting time of job \( j \) \( (x_j=0, \text{if } j \text{ is the first job in a bucket}) \)  
\[ C_k \] total actual processing time of all the assigned jobs in \( k \)th bucket

<Binary decision variables>

\[ y_{jk} \] equals to 1, if job \( j \) is processed in \( k \)th bucket, 0, otherwise  
\[ z_{ijk} \] equals to 1, if job \( i \) precedes job \( j \) in \( k \)th bucket, 0, otherwise

The actual processing time of job \( j \) is increased (deteriorated) depending upon the gap \( x_j \) between recent RMA and starting time of the job, hence the time becomes \( p_j \cdot (1 + r_j \cdot x_j) \). The deteriorated processing time of jobs is recovered to an original processing time by RMAs. The goal of the problem is to find a schedule minimizing the makespan which is the sum of total actual processing times of each bucket and all RMA times. The variable \( z_{ijk} \) is specially used in the model (equals to 1, if job \( i \) is the first job of \( k \)th bucket, 0, otherwise). Figure 1 describes the scheduling structure of time-dependent deteriorating jobs with determining the number and position of rate modifying activity.

The mixed integer programming (MIP) can be formulated as follows:

\[
\text{Minimize} \quad \sum_{k=1}^{B} C_k + \left( \sum_{k=1}^{B} \sum_{i=1}^{j} z_{ijk} - 1 \right) \cdot R
\]

\[
\text{Subject to} \quad x_i + p_i \cdot (1 + r_i \cdot x_i) \leq x_j + M \cdot \left( 1 - \sum_{k=1}^{B} z_{ijk} \right), \quad \text{for } \forall i, j, \text{and } i \neq j,
\]
\[ x_i + p_i \cdot (1 + r_i \cdot x_i) \leq C_k + M \cdot (1 - y_{ik}), \quad \text{for } \forall i, k, \]

\[ \sum_{j=1}^{I} z_{ijk} \leq 1, \quad \text{for } \forall k, \]

\[ \sum_{j=1}^{I} z_{ijk} = y_{ik}, \quad \text{for } \forall i, k, \]

\[ \sum_{j \neq i}^{I} z_{ijk} \leq y_{ik}, \quad \text{for } \forall i, k, \]

\[ \sum_{k=1}^{I} y_{ik} = 1, \quad \text{for } \forall i, \]

\[ x_i \geq 0, \quad \text{for } \forall i, \]

\[ C_k \geq 0, \quad \text{for } \forall k, \]

\[ y_{ik} = 0 \text{ or } 1, \quad \text{for } \forall i, k, \]

\[ z_{ijk} = 0 \text{ or } 1, \quad \text{for } \forall i, j, k. \]

Objective function (1) minimizes the makespan of the machine. It is calculated by total actual processing times of each bucket and RMA times. Note that a RMA time is not required to the bucket to which no jobs are assigned. Thus, the value in the parenthesis determines the number of RMAs that RMA time is needed. Constraint (2) ensures the precedence relation of jobs assigned in the same bucket and calculates starting time of each job. Constraint (3) calculates the total actual processing time of each bucket by calculating the completion time of each job in the bucket. Constraints (4)-(6) ensure that jobs assigned in the same bucket must be appeared once in their sequence. Constraint (4) guarantees that at most one job is assigned to a potential bucket and positioned at the beginning of the sequence before all the jobs in the bucket. Constraint (5) depicts that if a job is assigned to a bucket, then it will be immediately preceded by one job. Similarly Constraint (6) describes that if a job is assigned to a bucket then it can be succeeded by at most one job. Constraint (7) confirms that each job is processed in exactly one in the each bucket. The job in the last position of the sequence in a bucket will not have a succeeding job. The model has \( JB(B + 1) \) binary variables, \( (J + B) \) continuous variables and \( J^2(B + 1) + f(4B + 1) + 2B \) constraints.

**Fig. 1** Scheduling structure of time-dependent deteriorating jobs with determining the number and position of rate modifying activities

### 3. Meta-heuristic algorithms

The single machine scheduling problem with time-dependent deterioration and multiple RMAs considered in this paper is typical combinatorial optimization problem and the mixed integer programming model is not suitable for the
problems over 7 jobs because of the long computation time (See Table 1 in Section 4). Thus, we propose effective and efficient meta-heuristic algorithms. In this paper, we apply two genetic algorithms (GAs) with different chromosome structures, and propose two hybrid meta-heuristic algorithms based on GA. GA is known as an effective and efficient algorithm for combinatorial optimization problems (Gen and Cheng, 2000).

In GAs, the representation of a solution (chromosome) has a great influence on the performances of the algorithms. The chromosome should simultaneously represent the job assignments to buckets and the sequences of assigned jobs in each bucket, along with the number and positions of RMAs. In this paper, two GAs are applied by using different chromosome structures. They are called GA using chromosomes with double strings (GA_DS) and GA using chromosomes with special character (GA_SC). For applying the GAs to the problem simultaneously determining the number of RMAs of this paper, the number of potential buckets is assumed to be the number of independent jobs for the GAs. In Figure 2, the same example of the corresponding single machine scheduling problem with time-dependent processing times and multiple RMAs decoded from the two chromosome types is illustrated. The chromosomes represent the assigning and sequencing of ten jobs and three buckets. In the figure, only two RMAs are determined between bucket 2 and 5 and, between bucket 5 and 7 among nine possible RMAs. The sum of the three total actual processing times of each bucket and two RMA times becomes the makespan. The chromosome of GA_DS consists of a double-dimensional string array – a sequence array and assignment array. The sequence array is expressed by only J digits from 1 to J without duplication; whereas the assignment array is expressed by any integer between 1 to B. Figure 2 (a) describes an example the chromosome of GA_DS. There are only three digits 2, 5, and 7 in the assignment array, and the job groups 7-2-3, 5-10-6-1, and 8-4-9 are appeared in order coincide with each digit in the sequence array. Hence the decided number of buckets is three, and the jobs in each job groups are assigned to each bucket in order as the corresponding schedule in the figure. The chromosome of GA_SC consists of a single-dimensional string array, expressed by J digits from 1 to J and, as well as the (B-1) potential bucket separation indicator **’s. The digits between discontinuous ** represent the sequence of jobs assigned to a same bucket. Figure 2 (b) describes an example the chromosome of GA_SC. In the single-dimensional string array, the job groups 7-2-3, 5-10-6-1, and 8-4-9 are separated by the separation indicators. Hence the decided number of buckets is three, and the jobs in each job groups are assigned to each bucket in order as the corresponding schedule in the figure.

(a) GA_DS

\[
\begin{array}{cccccccc}
5 & 7 & 8 & 10 & 4 & 6 & 2 & 3 & 9 & 1 \\
5 & 2 & 7 & 5 & 7 & 5 & 2 & 7 & 7 & 5 \\
\end{array}
\]

sequence

assignment

(b) GA_SC

\[
\begin{array}{cccccccc}
* & ? & 2 & 3 & * & * & * & 5 & 10 & 6 & 1 & * & 8 & 4 & 9 & * & * & * \\
\end{array}
\]

<Corresponding schedule>

Fig. 2 Two chromosome representations and their corresponding schedule

We propose two hybrid meta-heuristic algorithms based on GA (HGA) with rule-based assigning heuristics. They are called HGA compounded with dispatching rule (HGA_DR) and HGA compounded with cutting rule (HGA_CR). The chromosomes of HGA_DR and HGA_CR are consists of only a single-dimensional string array, expressed by J digits from 1 to J. To generate a corresponding schedule with the simple chromosome, a job assigning rule to one of the buckets is needed. The digits in the chromosome represent a job order to apply the assigning rule. The difference between HGA_DR and HGA_CR is only the rule for representing the number of buckets, the assignment of jobs to one of buckets and the sequence of jobs in each bucket with the simple chromosome. The assigning rules are called a dispatching rule in HGA_DR and a cutting rule in HGA_CR.

The dispatching rule for HGA_DR is based on the incumbent makespan as shown in Figure 3. The incumbent schedule in the figure implies the schedule of jobs before job k in the chromosome with currently generated n buckets.
To assign job k in the chromosome to the incumbent schedule, the (n+1) candidate schedules are compared. The candidate schedules 1 to n are the schedules in which job k is assigned to the end of sequence in the corresponding generated bucket. The candidate schedule (n+1) is the schedule in which job k is assigned to a new bucket (n+1). The temporary makespans of all the candidate schedules are calculated, and then the candidate schedule n with the smallest temporary makespan is selected for the next incumbent schedule. So, job k is assigned to bucket n after job j, and the number of currently generated buckets remains n. The jobs in the chromosome are sequentially assigned to buckets according to the dispatching rule, and the corresponding schedule is finally generated.

The cutting rule for HGA_CR which decides the number and positions of RMAs is introduced to generate the corresponding schedule with the simple chromosome. The cutting rule is based on cumulative deterioration time as shown in Figure 4. The incumbent schedule in the figure implies the schedule of jobs before job k in the chromosome with currently generated n buckets. To decide job k is assigned to bucket n or a new bucket (n+1), the cumulative deterioration time of all the assigned jobs in bucket n is calculated and compared with the RMA time R. If the cumulative deterioration time is less than or equal to R, job k is assigned to the end of the sequence in bucket n for the next incumbent schedule and the number of currently generated buckets remains n as shown in Figure 4(a). Otherwise, a new bucket (n+1) is added and job k is assigned to the new bucket for the next incumbent schedule as shown in Figure 4(b). The jobs in the chromosome are sequentially assigned to buckets according to the cutting rule, and the corresponding schedule is finally generated.
In GAs, a set of chromosomes forms a population. The initial population is generated randomly for the first generation. The chromosomes in the population are evaluated using a fitness value. As a measure of fitness, the makespan, which is the objective function value proposed in Section 2, is used. In this study, the population of the next generation is primarily composed of the following: best chromosomes migrating from the current generation and new chromosomes (i.e., children) reproduced by crossover and mutation operators (with randomly selected parents from the population of current generation). The one-point crossover and swap mutation is used for all the GAs in this paper. For one-point crossover, one point is randomly selected for dividing one parent. The set of genes on left side is inherited from the parent to the child, and the other genes are placed in order of their appearance in the other parent. For swap mutation, two genes of a parent randomly selected are interchanged. The population of next generation is composed of the chromosomes selected by roulette-wheel from the children. The next generation is evaluated and this process is repeated until a stopping criterion (maximum number of generations) is met.

5. Computational results

In this section, extensive computational experiments are conducted to evaluate the performance of GA_DS, GA_SC, HGA_DR, and HGA_CR. Two problem groups are randomly generated according to the total number of jobs because the complexity of the problem highly depends on the number of jobs. First group, which involves problems with more than 3 jobs and less than or equal to 7 jobs, is for comparing solutions obtained by the GAs and the HGAs with the optimal solution. To obtain the optimal solution through mathematical programming presented in Section 2, ILOG CPLEX 10.2 is used. A 3600(sec.) time limit is imposed and a particular run is simply terminated if the optimal solution is not found and verified in the amount of time given. Second group, which involves problems with more than 10 total numbers of jobs, is for comparing the relative performance of each algorithm. To resolve all the problems, the GAs and the HGAs are run with a population size of $2J$, a generation size of 1000, and fixed crossover and mutation rates of 0.8 and 0.2, which are predetermined by extensive preliminary experiments.
We generate 12 test problems for the first group with different number of jobs and fixed RMA time 30, and 60 test problems for the second group with different number of jobs and RMA times because the performance of the all the algorithms highly depends upon the number of jobs and RMA times. The deterioration rate of jobs is generated from uniform (0.001, 0.01). The working time per day is assumed to 480 minutes and the processing time of jobs is randomly generated from the average processing time with 20% gap allowed. Thus, the processing time of jobs is generated from uniform (480 / total number of jobs – 20%, 480 / total number of jobs + 20%). All experiments of the first group utilize CPLEX, and the GAs and the HGAs are executed on a PC with 2.67 GHz Intel Xeon CPU processor and 48 GB RAM.

To compare the performance of the GAs and the HGAs, the relative percent deviation (RPD) is calculated by expression (12).

$$RPD(\%) = \frac{Best - Heu_{Sol}}{Best} \times 100,$$

where Heu_{Sol} is a meta-heuristic solution obtained by the GAs and the HGAs and Best is the best solution of all experiments for each test problem. Best can be the optimal solution if CPLEX obtains the optimal solution.

The test results of the small sized problems in the first group are summarized in Table 1. The optimal solution and computing time by CPLEX and mean value of the 10 replications, the average RPD and mean absolute deviation (MAD) of 10 replications, and the average computing time by GA_DS, GA_SC, HGA_DR, and HGA_CR for all test problems are compared. The computational time of CPLEX of the small sized problems in the first group significantly increases as the number of jobs increases. In the test, CPLEX was not able to obtain the optimal solution for problems over 7 jobs in a 3600(sec.) time limit. But the GAs and the HGAs are able to provide near optimal solutions within a second. Furthermore, the extremely small values of RPD and MAD in Table 1 indicate that all implemented algorithms provide good performances for the small sized problems.

| J | Opt. Time | Mean RPD | MAD Time | Mean RPD | MAD Time | Mean RPD | MAD Time | Mean RPD | MAD Time |
|---|---|---|---|---|---|---|---|---|---|
| 4 | 609 | 1.48 | 618 | 1.59 | 1.88 | 0.02 | 609 | 0.00 | 0.00 | 0.02 | 609 | 0.00 | 0.00 | 0.03 | 609 | 0.00 | 0.00 | 0.02 |
| 4 | 534 | 1.72 | 535 | 0.21 | 0.33 | 0.02 | 534 | 0.00 | 0.00 | 0.02 | 534 | 0.00 | 0.00 | 0.02 | 534 | 0.00 | 0.00 | 0.02 |
| 4 | 522 | 1.69 | 530 | 1.61 | 1.27 | 0.02 | 522 | 0.00 | 0.00 | 0.02 | 522 | 0.00 | 0.00 | 0.02 | 522 | 0.00 | 0.00 | 0.02 |
| 5 | 553 | 2.84 | 553 | 0.00 | 0.03 | 0.03 | 553 | 0.00 | 0.00 | 0.03 | 553 | 0.00 | 0.00 | 0.04 | 553 | 0.00 | 0.00 | 0.03 |
| 5 | 564 | 1.81 | 564 | 0.05 | 0.03 | 0.03 | 564 | 0.00 | 0.00 | 0.04 | 564 | 0.00 | 0.00 | 0.04 | 564 | 0.00 | 0.00 | 0.03 |
| 5 | 547 | 2.34 | 547 | 0.00 | 0.04 | 0.04 | 547 | 0.00 | 0.00 | 0.04 | 547 | 0.00 | 0.00 | 0.04 | 547 | 0.00 | 0.00 | 0.03 |
| 6 | 605 | 9.38 | 605 | 0.08 | 0.13 | 0.05 | 605 | 0.00 | 0.00 | 0.05 | 605 | 0.00 | 0.00 | 0.05 | 605 | 0.00 | 0.00 | 0.05 |
| 6 | 599 | 18.11 | 599 | 0.01 | 0.00 | 0.05 | 599 | 0.01 | 0.00 | 0.05 | 599 | 0.01 | 0.00 | 0.05 | 599 | 0.01 | 0.00 | 0.05 |
| 6 | 584 | 18.89 | 584 | 0.00 | 0.00 | 0.05 | 584 | 0.00 | 0.00 | 0.05 | 584 | 0.00 | 0.00 | 0.05 | 584 | 0.00 | 0.00 | 0.05 |
| 7 | 602 | 28.56 | 603 | 0.16 | 0.24 | 0.06 | 602 | 0.01 | 0.00 | 0.07 | 602 | 0.01 | 0.00 | 0.07 | 602 | 0.01 | 0.00 | 0.07 |
| 7 | 597 | 19.39 | 597 | 0.00 | 0.06 | 0.06 | 597 | 0.00 | 0.00 | 0.07 | 597 | 0.03 | 0.05 | 0.07 | 597 | 0.00 | 0.00 | 0.06 |
| 7 | 606 | 37.88 | 606 | 0.01 | 0.00 | 0.06 | 606 | 0.01 | 0.00 | 0.07 | 606 | 0.01 | 0.00 | 0.07 | 606 | 0.01 | 0.00 | 0.06 |

| Avg. | 12.01 | 0.31 | 0.32 | 0.04 | 0.00 | 0.00 | 0.05 | 0.00 | 0.00 | 0.05 | 0.00 | 0.00 | 0.04 |

The average RPD, MAD, the number of decided RMAs (n(RMA)), and computing time of 30 replications by GA_DS, GA_SC, HGA_DR, and HGA_CR of large sized problems in the second group with different job sizes (J) and RMA times (R) are summarized in Table 2 and Table 3. In order to validate the above results, it is interesting to check if the observed differences in the RPD values of each implemented algorithm are statistically significant. Figure 5 shows the mean plots and Tukey HSD intervals at the 95% confidence level of all large sized problems. Figure 5 indicates that there are statistically significant differences between the RPD values of all the implemented algorithms.
The observed differences between chromosome designs are more statistically significant as the number of job (J) and the RMA time (R) increase, as shown in graph (a) and (b) in Figure 6. In these plots, GA_SC and HGA_CR shows lower RPD with small variance than GA_DS and HGA_DR. GA_DS and HGA_DR are regarded as allocation based algorithm, because the each job in the sequence array of the chromosome are separately assigned to corresponding buckets according to the assignment array or dispatching rule. Whereas GA_SC and HGA_CR are regarded as

**Table 2** Test results of large sized problems in the second group with different job size (J)

| J  | RPD | MAD | n(RMA) | Time | RPD | MAD | n(RMA) | Time | RPD | MAD | n(RMA) | Time | RPD | MAD | n(RMA) | Time |
|----|-----|-----|--------|------|-----|-----|--------|------|-----|-----|--------|------|-----|-----|--------|------|
| 10 | 0.32| 0.36| 4.39   | 0.12 | 0.09| 0.11| 4.19   | 0.14 | 0.06| 0.04| 4.10  | 0.13 | 0.01| 0.02| 4.14   | 0.12 |
| 20 | 2.58| 1.06| 6.36   | 0.51 | 0.62| 0.34| 5.27   | 0.66 | 1.52| 0.21| 4.27  | 0.55 | 0.06| 0.05| 5.18   | 0.52 |
| 30 | 11.53| 2.10| 8.00   | 1.25 | 1.76| 0.73| 5.21   | 1.75 | 3.79| 0.28| 3.46  | 1.40 | 0.18| 0.14| 4.74   | 1.30 |
| 40 | 18.10| 2.10| 10.28  | 2.41 | 3.07| 1.10| 6.02   | 3.54 | 8.09| 0.51| 3.25  | 2.74 | 0.12| 0.10| 5.06   | 2.55 |
| 50 | 23.95| 2.04| 12.44  | 4.02 | 4.99| 1.66| 6.91   | 6.21 | 11.47| 0.52| 3.36  | 4.65 | 0.18| 0.08| 5.44   | 4.39 |
| 60 | 33.95| 2.18| 14.26  | 6.22 | 6.52| 1.80| 6.86   | 9.88 | 12.58| 0.30| 2.83  | 7.29 | 0.28| 0.17| 5.20   | 6.94 |
| 70 | 44.74| 2.61| 16.44  | 8.97 | 7.45| 2.12| 7.04   | 14.72| 17.91| 0.40| 2.64  | 10.61| 0.35| 0.19| 5.21   | 10.26|
| 80 | 53.17| 2.79| 18.42  | 12.43| 8.63| 2.36| 7.32   | 20.87| 22.05| 0.36| 2.50  | 14.72| 0.43| 0.28| 5.22   | 14.46|
| 90 | 66.40| 2.84| 20.07  | 16.68| 8.69| 2.07| 6.81   | 28.54| 31.53| 0.38| 2.38  | 19.72| 0.52| 0.32| 4.91   | 19.60|
| 100| 65.56| 2.84| 22.05  | 21.64| 10.26| 2.00| 7.90   | 37.45| 25.01| 0.37| 2.50  | 25.37| 0.35| 0.23| 5.31   | 25.45|

| Avg.| 32.03| 2.08| 13.27  | 7.43 | 5.21| 1.43| 6.35   | 12.38| 13.40| 0.34| 3.13  | 8.72 | 0.25| 0.16| 5.04   | 8.56 |

**Table 3** Test results of large sized problems of second group with different RMA time (R)

| R  | RPD | MAD | n(RMA) | Time | RPD | MAD | n(RMA) | Time | RPD | MAD | n(RMA) | Time | RPD | MAD | n(RMA) | Time |
|----|-----|-----|--------|------|-----|-----|--------|------|-----|-----|--------|------|-----|-----|--------|------|
| 10 | 6.46| 0.77| 13.97  | 7.41 | 3.63| 0.70| 9.98   | 12.21| 3.72| 0.24| 5.16  | 8.48 | 0.08| 0.05| 7.70   | 8.16 |
| 20 | 16.53| 1.52| 13.31  | 7.41 | 5.15| 1.09| 7.38   | 12.33| 8.72| 0.41| 3.61  | 8.67 | 0.17| 0.08| 5.71   | 8.45 |
| 30 | 27.13| 2.08| 13.17  | 7.44 | 5.77| 1.45| 6.22   | 12.39| 12.02| 0.32| 2.87  | 8.61 | 0.21| 0.15| 4.83   | 8.45 |
| 40 | 37.77| 2.41| 13.13  | 7.42 | 5.79| 1.57| 5.31   | 12.46| 14.89| 0.38| 2.55  | 8.84 | 0.54| 0.39| 4.28   | 8.72 |
| 50 | 47.53| 2.72| 13.00  | 7.44 | 5.87| 1.87| 4.81   | 12.42| 19.04| 0.40| 2.36  | 8.84 | 0.18| 0.12| 3.92   | 8.74 |
| 60 | 56.77| 2.96| 13.05  | 7.43 | 5.05| 1.91| 4.41   | 12.46| 22.01| 0.28| 2.23  | 8.86 | 0.31| 0.13| 3.80   | 8.82 |

| Avg.| 32.03| 2.08| 13.27  | 7.43 | 5.21| 1.43| 6.35   | 12.38| 13.40| 0.34| 3.13  | 8.72 | 0.25| 0.16| 5.04   | 8.56 |

The observed differences between chromosome designs are more statistically significant as the number of job (J) and the RMA time (R) increase, as shown in graph (a) and (b) in Figure 6. In these plots, GA_SC and HGA_CR shows lower RPD with small variance than GA_DS and HGA_DR. GA_DS and HGA_DR are regarded as allocation based algorithm, because the each job in the sequence array of the chromosome are separately assigned to corresponding buckets according to the assignment array or dispatching rule. Whereas GA_SC and HGA_CR are regarded as
clustering based algorithm, because the jobs in the sequence array of the chromosome are grouped by separation indicator or cutting rule. For the scheduling problem in this paper, the clustering based algorithms give better performance than the allocation based algorithms. The reason of this result is that the clustering based algorithms is more tightly assign jobs in the bucket and reduce the number of RMAs than allocation based algorithms.

Furthermore, the test results indicate that HGA_CR offers the best performance in any job sizes and RMA times. The computation times of GAs and HGAs are small enough to obtain solutions in a reasonable time. Thus, HGA_CR is an effective and efficient algorithm with low variation for the machine scheduling problem with time-dependent deterioration and multiple RMAs simultaneously determining the number and positions of RMAs.

![Image of mean plots and Tukey HSD intervals at the 95% confidence level for groups of different job size (J) and RMA time (R)](image_url)

**Fig. 6** Mean plots and Tukey HSD intervals at the 95% confidence level for groups of different job size (J) and RMA time (R)

6. Conclusions

Two emerging classes of scheduling problems, which are the class of scheduling problems with time-dependent processing times and the class of scheduling problems with RMAs were considered. In this paper, we proposed meta-heuristic algorithms for the single machine scheduling problem to minimize the makespan simultaneously determining the number and positions of RMAs and the schedule of jobs with time-dependent deterioration.

To solve the problem, a mixed integer programming model was derived to search the optimal solution using CPLEX. Since the CPLEX is inefficient and impractical for solving the large sized problems due to the increased computational time requirement, CPLEX was only used for solving small sized problems to compare the solution of the
proposed meta-heuristics to the optimal solution.

For large sized problems, we applied two genetic algorithms with different chromosome structures, and proposed two hybrid meta-heuristic algorithms based on GA to increase solution efficiency: GA using chromosomes with double strings (GA_DS), GA using chromosomes with special character (GA_SC), HGA compounded with dispatching rule (HGA_DR) and HGA compounded with cutting rule (HGA_CR). Computational experiments were conducted to evaluate the relative solution performance and the computation time of the proposed meta-heuristics under different job sizes and RMA times. The test results conclude that among all the implemented algorithms for machine scheduling problems with time-dependent deterioration and multiple RMAs, HGA_CR provides the highest quality in the solution performance and offers the reasonable computation time to solve the problem.

Extensions of the paper may consider the RMA time of the model in this paper is regarded as to be fixed to a constant value. But normally, the longer previous total actual processing time, the longer RMA time is required. The model should be extended to handle the time dependent RMA time in the further study. In addition, the case of the model that the actual processing times of jobs are not fully recovered is considered, if the RMA time is not enough offered.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future (Grant number: NRF-2013R1A1A1012454).

References

Browne, S. and Yechiali, U., Scheduling deteriorating jobs on a single processor, Operations Research, Vol. 38, (1990), pp. 495-498.

Gen, M. and Cheng, R., Genetic Algorithms and Engineering Optimization (2000), Wiley, New York.

Gordon, V. S. and Tarasevich, A. A., A note: common due date assignment for a single machine scheduling with the rate modifying activity, Computers & Operations Research, Vol. 36, (2009), pp. 325–328.

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

He, Y., Ji, M., and Cheng, T. C. E., Scheduling with a restricted rate modifying activity, Naval Research Logistics, Vol. 52 (2005), pp. 361–369.

Hsu, C. J., Cheng, T. C. E., and Yang, D. L., Unrelated parallel-machine scheduling with rate-modifying activities to minimize the total completion time, Information Sciences, Vol. 181, (2011), pp. 4799–4803.

Iranpoor, M., Ghomi, S. M. T., and Zandieh, M., Machine scheduling in the presence of sequence-dependent setup times and a rate-modifying activity, International Journal of Production Research, Vol. 50, No. 24 (2012), pp. 7401-7414.

Ji, M. and Cheng, T. C. E., Parallel-machine scheduling with simple linear deterioration to minimize total completion time, European Journal of Operational Research, Vol. 188, (2008), pp. 342–347.

Ji, M. and Cheng, T. C. E., Parallel-machine scheduling with simple linear deterioration, Theoretical Computer Sciences, Vol. 410 No. 38-40 (2009), pp. 3761–3768.

Joo, C. M. and Kim, B. S., Genetic algorithms for single machine scheduling with time-dependent deterioration and rate-modifying activities, Expert Systems with Applications, Vol. 40, (2013), pp. 3036-3043.

Kang L. Y. and Ng, C. T., A note on a fully polynomial-time approximation scheme for parallel-machine scheduling with deteriorating jobs, International Journal of Production Economics, Vol. 109, (2007), pp. 180–184.

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

Kim, B. S. and Ozturkoglu, Y., Scheduling a single machine with multiple preventive maintenance activities and position-based deteriorations using genetic algorithms, International Journal of Advanced Manufacturing Technology, Vol. 67, No. 5-8 (2013), pp. 1127-1137.
Lee, C. Y. and Leon, V. J., Machine scheduling with a rate-modifying activity, European Journal of Operational Research, Vol. 128, (2001), pp. 119-128.

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

Lee, K., Kim, B. S., and Joo, C. M., Genetic algorithms for door-assigning and sequencing of trucks at distribution centers for the improvement of operational performance, Experts with Applications, Vol. 39, (2012), pp. 12975-12983.

Lee, W. C. and Wu, C. C., Multi-machine scheduling with deteriorating jobs and scheduled maintenance, Applied Mathematical Model, Vol. 32, (2008), pp. 362–373.

Lee, W. C., Wu, C. C., and Liu, H. C., A note on single-machine makespan problem with general deteriorating function, International Journal Advanced Manufacturing Technology, Vol. 40, (2009), pp. 1053–1056.

Lodree, E. J. and Geiger, C. D., 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.

Mosheiov, G. and Oron, D., Due-date assignment and maintenance activity scheduling problem, Mathematical Computer Modeling, Vol. 44, (2006), pp. 1053–1057.

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

Mosheiov, G. and Sidney, J. B., Scheduling a deteriorating maintenance activity on a single machine, Journal of Operational Research Society, Vol. 61, (2010), pp. 882–887.

Ozturkoglu, Y. and Bulfin, R. L. A unique integer mathematical model for scheduling deteriorating jobs with rate-modifying activities on a single machine, International Journal Advanced Manufacturing Technology, Vol. 57, (2011), pp. 753–762.

Rudek, A. and Rudek, R., On flowshop scheduling problems with the aging effect and resource allocation, International Journal Advanced Manufacturing Technology, Vol. 62, (2012), pp. 135-145.

Shiau, Y. R., Lee, W. C., Wu, C. C., and Chang, C. M., Two-machine flowshop scheduling to minimize mean flow time under simple linear deterioration, International Journal Advanced Manufacturing Technology, Vol. 34, (2007), pp. 774–782.

Wang, D. and Wang, J. B., Single-machine scheduling with simple linear deterioration to minimize earliness penalties, International Journal Advanced Manufacturing Technology, Vol. 46, (2010a), pp. 285–290.

Wang, J. B., Flow shop scheduling with deteriorating jobs under dominating machines to minimize makespan, International Journal Advanced Manufacturing Technology, Vol. 48, (2010), pp. 719–723.

Wang, J. B., Wang, L. Y., Wang, D., and Wang, X. Y., Single-machine scheduling with a time-dependent deterioration, International Journal Advanced Manufacturing Technology, Vol. 43, (2009), pp. 805–809.

Wang, J. -J., Wang, J.-B., and Liu, F., Parallel machines scheduling with a deteriorating maintenance activity, Journal of the Operational Research Society, Vol. 62, (2011), pp. 1898–1902.

Wang, X. Y. and Wang, M. Z., Single machine common flow allowance scheduling with a rate-modifying activity, Computers and Industrial Engineering, Vol. 59, (2010b), pp. 898–902.

Yang, D. L. and Yang, S-J., Unrelated parallel-machine scheduling problems with multiple rate-modifying activities, Information Sciences, Vol. 235, (2013), pp. 280-286.

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

Zhao, C. L., Tang, H. Y., and Cheng, C. D., Two-parallel machines scheduling with rate-modifying activities to minimize total completion time, European Journal of Operational Research, Vol. 198, (2009), pp. 354–357.