*ICIC Express Letters*
*Part B: Applications*
Volume **7**, Number **10**, October **2016**

ICIC International ©2016 ISSN 2185-2766
pp. **2117–2122**

# A STUDY ON IDENTICAL PARALLEL MACHINE SCHEDULING WITH DETERIORATION AND RATE-MODIFYING ACTIVITIES

Young-Bin Woo and Byung Soo Kim*

Department of Industrial and Management Engineering
Incheon National University
119, Academy-ro, Songdo-dong, Yeonsu-gu, Incheon 406-772, Korea
*Corresponding author: bskim@inu.ac.kr

Received March 2016; accepted June 2016

Abstract. *In this article, we investigate identical parallel machine scheduling (IPMS) with time-dependent deterioration and multiple rate-modifying activities (RMAs). The deteriorated processing times of jobs in identical parallel machines increase actual processing times of the jobs. The deteriorated processing times of the jobs are recovered to an original processing time by RMAs. There is no restriction on the number of the RMAs. The goal is to determine the number and positions of RMAs and the sequence of jobs on each machine to minimize the makespan. We suggest a novel genetic algorithm embedding mathematical model (GA_EM). The performance of GA_EM is compared with a rule-based local search heuristic using randomly generated problem instances.*
**Keywords:** Multi-machine scheduling, Time-dependent deterioration, RMA, Makespan, Mixed integer linear programming, Genetic algorithm embedding mathematical model (GA_EM)

1. **Introduction.** In recent years, machine scheduling problems considering various job deteriorations and rate-modifying activities have been receiving increasing attention to many researchers on the scheduling area [1-8]. Unlike classical machine scheduling problems, this scheduling problem simultaneously deals with the deterioration effect and the activities that recover the deteriorated production time. The deterioration effect decreases the production efficiency of machines due to various reasons such as a mal-position of tools, mal-alignment of jobs, abrasion of tools, and scraps of operations. The deteriorated processing times of jobs are recovered to an original processing time by the maintenance or cleaning process of machines. The recovering process bringing back to original processing times is called *rate-modifying activity* (RMA). Lee and Leon [1] initially introduced a single machine scheduling problem with deteriorating jobs and only one RMA. In this research, the processing time of the job is changed by a given rate depending on whether the job is scheduled before or after the RMA. They assumed that the machine has at most one RMA during the planning horizon and developed a dynamic programming algorithm by using the optimality properties corresponding to the problem. Since then, many researchers have conducted researches focusing on various deteriorating strategies and multiple RMAs for parallel machine scheduling problems. Zhao et al. [2] extended some objectives studied by Lee and Leon [1] to the two identical parallel machines environment. They suggested a polynomial time algorithm for the problem to find an optimal solution but they still allowed only one of RMAs. Yang and Yang [7] and Yang [8] investigated a scheduling problem with multiple RMAs on unrelated parallel machines. They proposed a polynomial time algorithm for solving the problems. However, they limit research by fixing two major decisions. First they determined the number and positions of RMAs and the sequence of jobs under the fixed number of machines. Furthermore, the global

optimal solution is found from the local optima by a derived mathematical model enumerating all the combination of RMAs under given upper bound RMAs. Therefore, the polynomial algorithms proposed by Yang and Yang [7] and Yang [8] are hard to find the global optimal solution in a reasonable time, if the number of jobs and the number of RMAs are increased. To the best of our knowledge, no research has been found proposing good heuristic algorithms for identical parallel machines scheduling (IPMS) problem with time-dependent deterioration and multiple RMAs that are not limited by the number of machines and the number of RMAs. In real-life manufacturing scheduling processes, simultaneous consideration of deteriorations and multiple RMAs enables us to increase the productivity of the process. Previous studies [7,8] have a significant limitation to find the optimal solution when the number of RMAs in each machine is larger than two and the number of machines is larger than two because they are not able to obtain the optimal solution due to increasing the complexity of problems. In this paper, we propose a novel genetic algorithm embedding mathematical model (GA_EM) for the IPMS problem with deterioration and RMAs. The paper is organized as follows. Section 2 describes the problem statement of this problem. In Section 3, heuristic algorithms are proposed. In Section 4, computational experiments are conducted by randomly generated test problems. Finally, the conclusions and future studies are discussed in Section 5.

2. **Problem Statement.** In this section, we consider an IPMS problem with time-dependent deterioration and multiple RMAs. There are $n$ jobs expressed by $J_1, J_2, \ldots, J_n$, to be processed on $k$ identical parallel machines. All jobs are available at time zero and no preemption is allowed. Associated with job $J_i$ ($i = 1, 2, \ldots, n$) there is an original processing time $p_i$ and a deterioration rate $\alpha_i$. Each machine has the different number of RMAs to improve production rate and the RMA time is the same and fixed as $\gamma$. We assume that RMA can be performed right after completing the processing of any job and any job cannot be processed during the RMA. The RMAs are defined as a recovering process bringing back to original processing times like Lee and Leon [1] and Kim and Joo [3]. In other words, if job $J_i$ is processed at time zero or right after the RMA, its processing time is $p_i$. If $J_i$ is processed before the RMA, its processing time is $p_i + x_i\alpha_i$, where $x_i$ is the starting time of job $J_i$ ($i = 1, 2, \ldots, n$). Figure 1 describes an example of IPMS for $n$ jobs with multiple RMAs on $k$ machines. Note that the working horizons between RMAs including horizons before the first and after the last RMAs are called *buckets* [4]. There are $n$ potential buckets to be assigned on machines. The completion time of bucket $i$ is defined as $BC_i$ if the bucket includes more than one job. Each machine includes one or more buckets. A completion time corresponding to each machine is equal to the sum of completion times of buckets and the total RMA times. For example in Figure 1, the completion time of machine 1 can be written as $BC_1 + BC_2 + BC_3 + 2\gamma$. The main decisions in the problem are to simultaneously determine the number and positions of RMAs and the sequence of jobs on each machine to minimize the makespan which minimizes the maximum of the sum of processing times of jobs and total RMA times on each machine.

3. **Heuristics.** In this section, we suggest two heuristic methods since an exact approach would be impractical for solving the problem to determine the number and positions of RMAs and the sequence of jobs on each machine. We first present a priority rule based heuristic and then suggest a genetic algorithm embedding mathematical model (GA_EM) to solve the problem. The priority rule based heuristic applies a longest processing time list scheduling (LPT) heuristic. The LPT heuristic is known that it returns a makespan for a parallel machine scheduling problem within the worst-case $(4/3 - 1/3k) \cdot OPT$ where OPT is optimum [9]. Thus, the priority rule based heuristic is proposed and applied to the problem as a comparison for GA_EM.

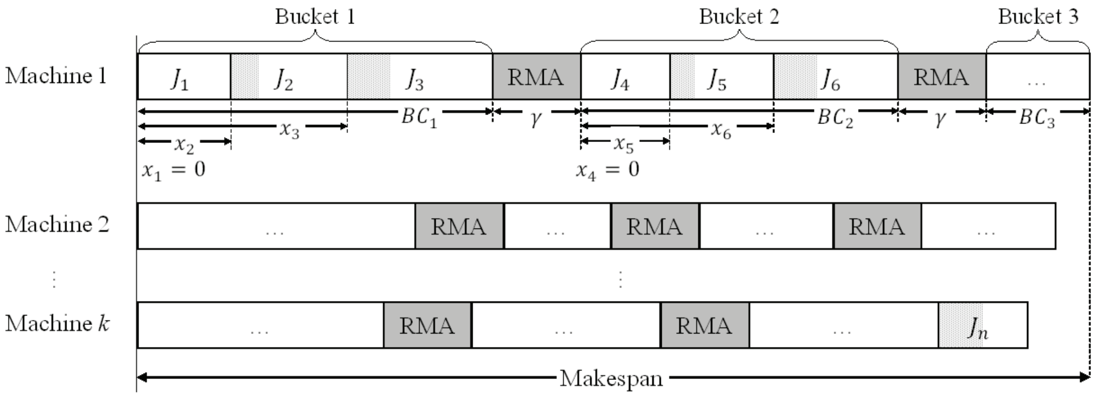


FIGURE 1. An example of IPMS for $n$ jobs with multiple RMAs on $k$ machines

3.1. **Modified LPT list scheduling heuristic.** Heuristics is a method to find a 'good' solution within a short time by implementing devised procedures on the complex or large-sized problems. Graham [9] has found that LPT list scheduling heuristic tends to provide a tight makespan for a parallel machine scheduling problem. In this sub-section, we suggest a modified LPT list scheduling (MLPT) heuristic to apply to our problem that considers deteriorating jobs and RMAs. The procedure of MLPT heuristic is as follows.
*Procedure of MLPT heuristic*

*Step 1.* Sort the list of jobs by LPT.
*Step 2.* Choose an unscheduled job in order from the list and schedule it on a machine that can perform the job as early as possible by comparing the candidate-completion-times.
  *Step 2.1.* Calculate two candidate-completion-times for each machine. First one is calculated by adding RMA time and original processing time to incumbent-completion-time, and second one is calculated by adding actual processing time to incumbent-completion-time.
  *Step 2.2.* Find the early candidate-completion-time and schedule the job corresponding to machine. If candidate-completion-time includes RMA time, additionally schedule an RMA in front of the job.
*Step 3.* Repeat Step 2 until all jobs are scheduled.

3.2. **Genetic algorithm embedding mathematical model (GA_EM).** Genetic algorithms (GAs), which are introduced by Holland [10], are known as an effective and efficient approach by applying the evolution theory. GAs find a near-optimal solution for combinatorial optimization problems in a relatively reasonable time. To solve the problem effectively, we propose a GA_EM that is used to guarantee the local optimal solution corresponding to a chromosome. Chromosome means the representation of a candidate solution. In the decoding process, it should be decoded to obtain several data, which are required to complete a mathematical model such as the number of buckets and the completion times of each bucket. Each chromosome is based on single dimensional array that consists of string values related to job indices and no duplicate string values are allowed. Figure 2(a) describes an example of a chromosome in the case of 10 jobs scheduling problem. For the example, the processing times and deterioration rates are explained in Table 1 and the RMA time is given as 20. To construct a set of buckets corresponding to the chromosome, we apply a modified dispatching rule that offers not only a set of buckets but also the assignment of jobs and the sequence of jobs in each bucket. The procedure of the modified dispatching rule is explained in the following manner.

*Decoding procedure of GA_EM*

*Step 1.* Select a job in order from the chromosome.
*Step 2.* Create first candidate bucket.
*Step 3.* Calculate the deterioration time of the job by assuming that the job is assigned to the end in the bucket.
*Step 4.* If the deterioration time of assigned job is less than RMA time, assign the job to the candidate bucket. Otherwise, close assigning the job to the candidate bucket and then create a new candidate bucket.
*Step 5.* Calculate the completion times of the candidate buckets.
*Step 6.* Repeat Step 3 to Step 5 until all jobs are assigned to the buckets.

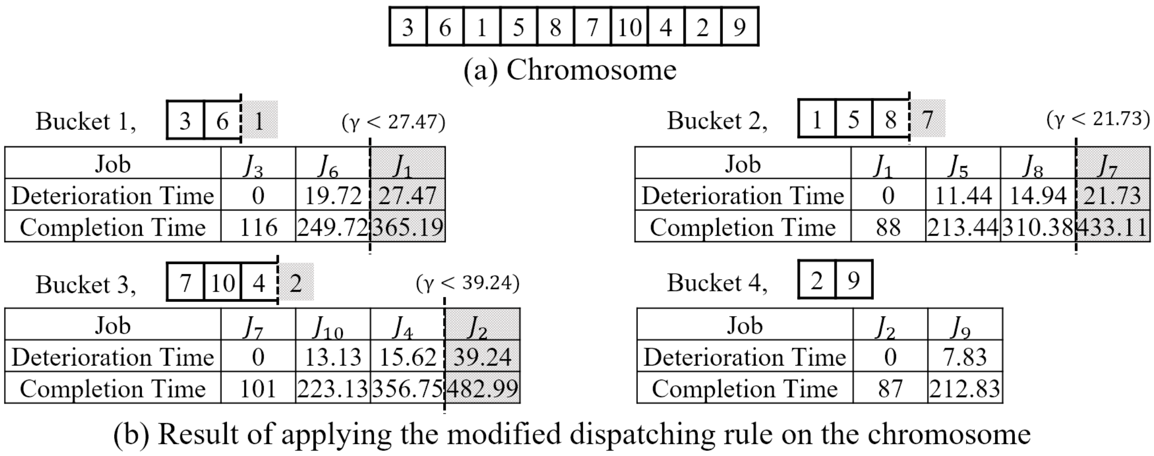


FIGURE 2. Chromosome representation and its corresponding buckets

TABLE 1. Information for processing times and deterioration rates

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $p_i$ | 88 | 87 | 116 | 118 | 114 | 114 | 101 | 82 | 118 | 109 |
| $\alpha_i$ | 0.11 | 0.11 | 0.13 | 0.07 | 0.13 | 0.17 | 0.07 | 0.07 | 0.09 | 0.13 |

Figure 2(b) shows the result of applying the modified dispatching rule on the chromosome. In this example, the number of buckets is 4 and the completion times (i.e., $BC_1$, $BC_2$, $BC_3$ and $BC_4$) of each bucket are 249.72, 310.38, 356.75, and 212.83, respectively. In evaluation process, the mathematical model which is completed by inputting bucket information, is used to calculate the makespan corresponding to the chromosome. Note that if the jobs are assigned to several buckets according to the rule, this scheduling problem is the same as an assignment problem, which is assigning the buckets to the machines. The mathematical representation is as follows:

*Decision variables*

$Y_{ij}$ 1 if bucket $i \in \{B_1, B_2, \ldots, B_l\}$ is assigned to machine $j \in \{M_1, M_2, \ldots, M_k\}$, 0 otherwise

$MC_j$ completion time of machine $j \in \{M_1, M_2, \ldots, M_k\}$

$$Min \max_{j \in \{M_1, M_2, \ldots, M_k\}} \{MC_j\} \tag{1}$$

$$\text{s.t.} \sum_{j \in \{M_1, M_2, \ldots, M_k\}} Y_{ij} = 1 \qquad \forall i \in \{B_1, B_2, \ldots, B_l\} \tag{2}$$

$$\sum_{i \in \{B_1, B_2, \ldots, B_l\}} BC_i \cdot Y_{ij} + \gamma \cdot \left( \sum_{i \in \{B_1, B_2, \ldots, B_l\}} Y_{ij} - 1 \right) \leq MC_j$$

$$\forall j \in \{M_1, M_2, \ldots, M_k\} \tag{3}$$

where $k$ and $l$ are the number of machines and the number of buckets, respectively, $\boldsymbol{M}$ and $\boldsymbol{B}$ are the index of a machine and the index of a bucket, respectively.

Objective function (1) is to minimize the makespan for IPMS problem. The constraints (2) ensure that each bucket must be assigned to only one machine. The constraints (3) define that the completion time of a machine is calculated by adding the sum of the completion time of the bucket assigned to the machine and the total RMA times. The makespan, which is obtained by solving the mathematical model, is used to evaluate each chromosome in the same generation as the measure of fitness. GA_EM has two genetic operators, crossover and mutation. These operators improve the fitness of solutions by passing down similarities and unexpected genetic characteristics to offsprings. For a crossover, position-based uniform crossover is used. For mutation, swap mutation is used. Using three genetic operators such as crossover operator, mutation operator and reproduction operator, the selected parents breed new chromosomes to generate a population for the next generation. In the reproduction process, the chromosomes in the top 10% of current population were cloned to the next generation and the rest of chromosomes in the next generation are probabilistically formed by roulette wheel method and genetic operators. Next, the chromosomes in the generation are evaluated and this process is repeated until a stopping criterion (the maximum number of generations) is met.

4. **Computational Experiments.** In this section, GA_EM is compared with MLPT heuristic. Extensive computational experiments are implemented to evaluate GA_EM and MLPT heuristic. GA_EM and MLPT heuristic are tested on 18 combinations of the number of jobs (10, 12, 14, 16, 18 and 20) and the number of machines (3, 4, and 5) and each combination is repeated for 10 times. The processing times of jobs were generated from a random value between $U[80, 120]$, the deterioration rates of jobs were generated in a random value of $U[0.05, 0.20]$ and the RMA time is fixed by 20. In GA_EM, the population size is set by $10 \times n$, and a generation size is set by 200. A crossover rate and a mutation rate are 0.8 and 0.005, respectively, which are predetermined by extensive preliminary experiments.

Table 2 shows the problem instances and computational results of GA_EM and MLPT heuristic. The results incorporate the best makespan and the mean of makespans. To evaluate the performance of GA_EM and MLPT heuristic, the relative percentage deviation (*RPD*) was used. *RPD* is percentage deviation from best objective value for the value obtained in all replications of each problem instance. We can see that MLPT heuristic takes much less computing (CPU) time ($<$ 1 sec.) while the mean of CPU times for GA_EM is computed as 399.97 sec. Nevertheless, the results show that GA_EM is more effective with low variation than MLPT heuristic. The mean of *RPDs* for GA_EM is 0.17% and the *RPD* for MLPT heuristic is 6.49%.

5. **Conclusions.** In this paper, we developed an IPMS problem considering various job deteriorations and RMAs. The objective of the problem is to simultaneously determine the number and positions of RMAs and the sequence of jobs on each machine to minimize the makespan. To effectively solve the complex scheduling problem, GA_EM is proposed and is compared with MLPT heuristic using randomly generated problem instances. GA_EM shows the good quality performance with low variation for the IPMS problem with deterioration and RMAs.

In the future, we will develop a novel mathematical model for the suggested problem to find the optimal solution for the IPMS problem with deterioration and RMAs. To overcome intractability of the model, future work will include more effective and efficient methodologies.

Table 2. The problem instances and computational results of two heuristics

| | | | | MLPT Heuristic | | GA_EM | | |
|---|---|---|---|---|---|---|---|---|
| No. | Jobs | Machine | Best Makespan | Makespan | *RPD* | Mean of Makespans | Mean of *RPDs* | Mean of CPU Times |
| 1 | 10 | 3 | 399.05 | 440.17 | 10.30% | 399.97 | 0.23% | 79.9 sec |
| 2 | | 4 | 305.10 | 324.25 | 6.28% | 306.81 | 0.56% | 84.6 sec |
| 3 | | 5 | 217.85 | 221.10 | 1.49% | 217.85 | 0.00% | 63.0 sec |
| 4 | 12 | 3 | 411.03 | 423.21 | 2.96% | 411.79 | 0.18% | 96.6 sec |
| 5 | | 4 | 333.47 | 336.08 | 0.78% | 333.71 | 0.07% | 154.1 sec |
| 6 | | 5 | 283.48 | 307.00 | 8.30% | 283.96 | 0.17% | 98.7 sec |
| 7 | 14 | 3 | 518.68 | 552.45 | 6.51% | 520.61 | 0.37% | 253.1 sec |
| 8 | | 4 | 398.74 | 436.39 | 9.44% | 398.74 | 0.00% | 226.1 sec |
| 9 | | 5 | 306.32 | 325.04 | 6.11% | 307.81 | 0.48% | 165.4 sec |
| 10 | 16 | 3 | 618.21 | 683.75 | 10.60% | 618.84 | 0.10% | 385.4 sec |
| 11 | | 4 | 432.79 | 445.09 | 2.84% | 433.52 | 0.17% | 270.9 sec |
| 12 | | 5 | 365.48 | 441.06 | 20.68% | 365.54 | 0.02% | 361.4 sec |
| 13 | 18 | 3 | 658.56 | 667.17 | 1.31% | 659.11 | 0.08% | 466.6 sec |
| 14 | | 4 | 504.93 | 556.18 | 10.15% | 506.08 | 0.23% | 371.4 sec |
| 15 | | 5 | 402.66 | 435.42 | 8.14% | 403.74 | 0.27% | 378.3 sec |
| 16 | 20 | 3 | 741.60 | 790.58 | 6.60% | 742.30 | 0.09% | 559.5 sec |
| 17 | | 4 | 572.30 | 578.33 | 1.05% | 573.11 | 0.14% | 673.3 sec |
| 18 | | 5 | 440.12 | 454.90 | 3.36% | 441.33 | 0.27% | 521.9 sec |
| Average | | | | | 6.49% | | 0.17% | 399.97 sec |

**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 Planning (Grant number: NRF-2016R1C1B2010224).

## REFERENCES

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

[2] C. L. Zhao, H. Y. Tang and C. D. Cheng, Two-parallel machines scheduling with rate-modifying activities to minimize total completion time, *European Journal of Operational Research*, vol.198, pp.354-357, 2009.

[3] B. S. Kim 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, pp.247-254, 2011.

[4] C. M. Joo and B. S. Kim, Genetic algorithms for single machine scheduling with time-dependent deterioration and rate-modifying activities, *Expert Systems with Applications*, vol.40, pp.3036-3043, 2013.

[5] B. D. Chung and B. S. Kim, A hybrid genetic algorithm with two-stage dispatching heuristic for a machine scheduling problem with step-deteriorating jobs and rate-modifying activities, *Computers & Industrial Engineering*, vol.98, pp.113-124, 2016.

[6] C. M. Joo and B. S. Kim, Machine scheduling of time-dependent deteriorating jobs with determining the optimal number of rate modifying activities and the position of the activities, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol.9, JAMDSM0007-JAMDSM0007, 2015.

[7] D. L. Yang and S. J. Yang, Unrelated parallel-machine scheduling problems with multiple rate-modifying activities, *Information Sciences*, vol.235, pp.280-286, 2013.

[8] S. J. Yang, Unrelated parallel-machine scheduling with deterioration effects and deteriorating multi-maintenance activities for minimizing the total completion time, *Applied Mathematical Modelling*, vol.37, pp.2995-3005, 2013.

[9] R. L. Graham, Bounds on multiprocessing timing anomalies, *SIAM Journal on Applied Mathematics*, vol.17, pp.416-429, 1969.

[10] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The MIT Press, 1975.