AN ANTICIPATIVE SCHEDULING APPROACH WITH CONTROLLABLE PROCESSING TIMES

SİNAN GÜREL, ERSİN KÖRPEOĞLU, AND M. SELİM AKTÜRK

ABSTRACT. In practice, machine schedules are usually subject to disruptions which have to be repaired by reactive scheduling decisions. Preparing initial schedules by considering possible disruption times along with rescheduling objectives is critical for the performance of reactive decisions. In this paper, we show that if the processing times are controllable then an anticipative approach can be used to form an initial schedule that could improve the performance of rescheduling decisions. Specifically, we consider a non-identical parallel machining environment, where processing times can be controlled at a certain compression cost. When there is a disruption during the execution of the initial schedule, a match-up time strategy is utilized such that a repaired schedule has to catch-up initial schedule at some point in future. This requires changing machine-job assignments and processing times for the rest of the schedule which implies increased manufacturing costs. We show that making anticipative job sequencing decisions, based on failure and repair time distributions and flexibility of jobs, one can repair schedules by incurring less manufacturing cost. Our computational results show that the match-up time strategy is very sensitive to initial schedule and the proposed anticipative scheduling algorithm can be very helpful to reduce rescheduling costs.

Keywords: Anticipative Scheduling, Controllable Processing Times, Reactive Scheduling, Match-up Time.

January 2009

1. INTRODUCTION

Unexpected events such as machine breakdowns or new job arrivals necessitate rescheduling remaining jobs in a schedule. Processing time controllability provides us flexibility in rescheduling against unexpected disruptions by allowing changes on the processing times of the jobs. However, the performance of rescheduling decisions, such as replanning the processing times or reallocating jobs between machines, highly depends on the state of the schedule at the time of disruption. Thus, it is critical to prepare initial schedules by considering possible disruption times and the ability of jobs to absorb disruptions. In this study, we develop an anticipative scheduling approach to form an initial schedule that could improve the performance of rescheduling decisions with controllable processing times.

M. S. Aktürk, E. Körpeoğlu: Department of Industrial Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey (akturk@bilkent.edu.tr, ersink@bilkent.edu.tr)
S. Gürel: University of Warwick, Centre for Discrete Mathematics and its Applications (DIMAP), Coventry, CV4 7AL, United Kingdom (Sinan.Gurel@wbs.ac.uk) .

To illustrate the anticipative scheduling idea we design a scheduling algorithm for a set of jobs to be scheduled on parallel machines with given machining time capacities on each machine. Initial objective for this problem is to minimize the total manufacturing cost of the jobs. We first find the optimal machine–job assignment and the optimal compression levels on the processing times of the jobs. Having found the machine–job assignments, the problem is to find a job sequence on each machine. We consider the situation that if a machine breakdown occurs on one of the machines, a reactive scheduling problem is solved and the remaining schedule is repaired. We assume that failure and repair times are uncertain with given probability distributions. In the considered reactive scheduling problem, the objective is to minimize the manufacturing cost increase due to disruption which is denoted as rescheduling cost. We assume a restriction that the repaired schedule has to match up with the initial schedule at a given time point after disruption. We provide a scheduling algorithm which determines a job sequence on each machine by considering possible downtime periods on the machines along with rescheduling cost minimization objective.

1.1. **Literature.** In the scheduling literature, reactive and predictive scheduling approaches have been considered extensively. In those studies usually the aim is to prepare an initial schedule in such a way that the schedule can be repaired with simple adjustments and within a slight performance degradation. Aytug et al. (2005) gave an extensive literature survey on scheduling under uncertainty and generating robust schedules. Jensen (2001) defined a robust schedule as the one which performs good when there was a disruption and the schedule was right shifted. Leon et al. (1994) considered finding robust schedules in a job shop scheduling environment which is subject to a single disruption. They proposed a genetic algorithm to minimize expected makespan and expected delay measures. To the best of our knowledge, studies in the literature assume fixed processing times. Here, we consider anticipative scheduling with controllable processing times.

1.1.1. *Idle time insertion.* In order to minimize the effects of possible disruptions on a schedule, a well known predictive approach is inserting idle times in it, so that disruptions can be absorbed without disturbing the system. Almost all of the existing reactive scheduling strategies (including match-up and right-shift scheduling techniques) try to accommodate disruptions by using the available idle times on initial schedules. Inserting idle times, as a predictive scheduling approach, is first proposed by Mehta and Uzsoy (1998) for a job-shop scheduling problem. Recently Leus and Herroelen (2007) presented a new model for single machine scheduling problems with stability objective and a common deadline, and proposed a branch and bound algorithm for an approximate formulation of the model to determine when and where to place an inserted idle time. Their algorithm gives the optimal job sequence and the optimal length of idle time following each job in the schedule when exactly one job deviates from its expected processing time. Yang and Geunes (2008) considered inserting idle times on a single machine scheduling problem where there exists uncertain future jobs that may arrive. They proposed a heuristic dynamic programming algorithm to minimize the expected sum of tardiness cost, the disruption cost and the wasted idle time cost. A similar idea (e.g. inserting a time buffer to protect a deterministic baseline schedule in order to cope with uncertainties) is also proposed for the project management problems, called buffer sizing, in the critical chain scheduling and buffer management (CC/BM) software as discussed in Herroelen and Leus (2001). In rescheduling with fixed processing times, inserting idle time is an efficient predictive approach. However, in case of controllable processing times, when the machining time capacity is limited and fully utilized, inserting idle times into a schedule require applying extra compression on the processing times of jobs. This increases compression costs. If no disruption occurs or if a disruption occurs after the inserted idle times, then the inserted idle time becomes useless. If the processing times are controllable, an alternative rescheduling approach to inserting idle times is reacting to disruptions by replanning the processing times. Consequently, the limited capacity of production resources are utilized more effectively.

In any idle time insertion approach, a critical decision to be made is when and how much idle time should be placed into an initial schedule so that the new schedule achieved by rescheduling after a disruption has the best scheduling performance. Analogously, in rescheduling with controllable processing times, it is critical to find the positions of the jobs in the initial schedule in an appropriate order so that a possible disruption is absorbed immediately and with a reasonable manufacturing cost increase. Therefore, we propose a new anticipative scheduling algorithm to form an initial schedule that takes flexibility of jobs along with probability distributions of failure and repair time of machines into account. Proposed flexibility measures estimate the ability of the jobs to absorb disruptions with less compression and reallocation costs, so that we schedule the most flexible jobs to the time zones where the downtime probability of a machine is higher.

1.1.2. *Controllable processing times.* A well known example to a controllable processing time is the processing time of CNC machining operations in flexible manufacturing. We can control the processing time of a job by setting the cutting speed and/or the feed rate on the machine. In turning operation as you increase the cutting speed and the feed rate, the processing time of the operation is compressed whereas the compression cost of the operation is increased due to increased tooling costs (Gürel and Aktürk, 2007). Shabtay and Steiner (2007) give an extensive literature review on the scheduling problems with controllable processing times. To the best of our knowledge, generating flexible schedules for the scheduling environments with controllable processing times has not been considered in the literature yet. Our work is the first attempt to employ anticipative scheduling with controllable processing times.

1.1.3. *Match-up scheduling.* When a disruption occurs, in order to stay consistent with the initial schedule, a critical rescheduling goal is to catch up with the initial schedule as soon as possible. The new schedule catches up with the initial schedule at the time point where the new schedule is exactly at the same state as the initial schedule. This time point is called the match-up time. Minimizing the match-up time helps to reduce the effects of a disruption on the production plan and on the schedules at the other stages of the production. For example, an extensive change in the completion times of jobs in the schedule of a department may cause unavailability of parts for the scheduled production in another department. In the literature, there exists few match-up scheduling studies such as Bean et al. (1991) and Aktürk and Görgülü (1999), which consider heuristic approaches to find match-up times under the existence of inserted idle times and fixed processing times. In rescheduling with controllable processing times, catching up an initial schedule earlier is possible by extensively compressing the jobs that are scheduled just after the disruption. With convex compression costs, absorbing a downtime by compressing a smaller set of jobs in the schedule results higher compression costs. Hence, there is a trade-off between the match-up time and the cost of the new schedule. Aktürk et al. (2009b) considered match-up time minimization and cost minimization problems for a parallel machine environment and showed the trade-off between two objectives.

1.2. **Contribution.** In this study, we introduce an anticipative scheduling approach with controllable processing times. We show that using the reactive scheduling objective and constraints, uncertainty data for downtimes, manufacturing cost and processing time controllability simultaneously, one can prepare initial schedules which could result improved rescheduling cost performance in case of a disruption.

As a specific problem we consider generating flexible initial schedules for the manufacturing cost objective by using an anticipative scheduling approach. For the rescheduling problem, we will consider minimizing rescheduling cost subject to a given match-up time point. We show that the rescheduling cost objective is quite sensitive to the set of jobs that are affected by the machine breakdown. Our scheduling algorithm uses the failure and repair time distributions and the manufacturing cost functions of the jobs in order to find the initial schedules which can be repaired at lower rescheduling cost levels. The Proposed approach in this study incurs no additional cost in terms of match-up time and manufacturing cost, but gives less rescheduling costs in case of a machine breakdown. Our computational experiments show that our approach can achieve an average improvement of 25% in rescheduling costs.

1.3. **Organization.** In Section 2, we define the considered scheduling environment, formulate the reactive cost minimization problem considered in this study and then discuss the related scheduling problem. In Section 3, we introduce our anticipative scheduling algorithm. We first introduce machine job allocation problem briefly, then present a probabilistic analysis and discuss proposed flexibility measures. Finally, we give a probabilistic sequencing algorithm for the cost minimization problem. Section 4 gives the results of the computational experiments and we conclude with Section 5.

2. RESCHEDULING COST MINIMIZATION PROBLEM

We consider n jobs to be processed on m non-identical parallel CNC machines. Processing time of job j on machine i is p^u_{ij}. Processing time of a job on machine can be compressed. Amount of compression y_{ij} is a decision variable and has an upper bound u_{ij}. Manufacturing cost of job j on machine i is c_{ij}. Compression cost function for job j on machine i is $f_{ij}(y_{ij})$. On each machine, there is a given available machining time capacity D_i. For the considered rescheduling problem initial machine–job assignment, denoted by $\mathcal{A}$, is obtained by solving a minimum cost machine-job assignment problem which will be introduced in Section 3.1.

Given $\mathcal{A}$, an initial schedule, called $\mathcal{S}$, with the start and end times of jobs on each machine is to be formed by finding a job sequence on each machine. Different disruptions may occur to a schedule $\mathcal{S}$ during its execution. In this study, we assume that a breakdown could occur on one of the machines at an uncertain time.

We also assume that since the failed machine has to be fixed, it will be down for an uncertain amount of time which will be known at the time of breakdown. If the breakdown occurs in the middle of the processing a job, the job has to be reprocessed in its entirety. This situation is called the preempt-repeat case in the literature.

Given such a downtime period on one of the machines, $\mathcal{S}$ is no longer executable. A subset of jobs in $\mathcal{S}$ has already been finished before the disruption. We assume that the jobs being processed on the machines other than the disrupted machine at the time of breakdown will finish their process as planned in $\mathcal{S}$. The other jobs which have not started processing yet at the time of breakdown and the job which is disrupted by the breakdown on the failed machine have to be rescheduled. These jobs are either to be reallocated to other machines and/or replanned to calculate their new processing times.

We consider a rescheduling cost minimization problem which is to be solved after a breakdown occurs. As one of the machines is disrupted and the schedule for the remaining jobs has to be repaired, one can look for alternative machine–job assignments and processing time decisions. Repaired schedule is required to satisfy the scheduling and processing time related constraints at a minimum rescheduling cost. It is also required that the repaired schedule catches up the initial schedule as soon as possible after a breakdown. Therefore, this problem could be formulated as to minimize the rescheduling cost of remaining jobs for a given limit on the match-up time. In this problem, a match-up time on a machine implies that the schedule, i.e. the job sequence and start-end times of the jobs, following the match-up point is exactly the same as in initial schedule $\mathcal{S}$. As we consider a non-preemptive rescheduling environment, we select match-up times out of the start times of the jobs previously determined in $\mathcal{S}$.

2.1. **Manufacturing cost function.** The manufacturing cost of a job on a machine is a fixed amount c_{ij}, which is the cost if the job is operated at p^u_{ij}, plus the compression cost which is incurred if the processing time of the job is compressed. Compressing the processing time of a job requires using additional resource. As we increase the cutting speed and/or feed rate on a CNC machine, the tool life is reduced and hence the manufacturing cost is increased. The compression cost of each job can be expressed as a function of $y \geq 0$ as

$$f(y) = ky^{(a/b)},$$

where $a \geq b > 0$ and $k > 0$ so that f is increasing and convex. As we decrease the processing time of a job, it requires more additional resource to compress it further. As discussed in Kayan and Aktürk (2005), in turning operation, the length and the diameter of the job, the required surface roughness, machine horsepower, and the required tool type determine the cost coefficients k, a, and b for each machine-job pair.

2.2. **Rescheduling Problem Formulation.** In the rescheduling cost minimization problem, for each job to be rescheduled, a machine-job assignment decision has to be made. x_{ij} is the assignment variable which is 1 if job j is assigned to machine i and 0, otherwise. Also, for each job a new compression amount (y_{ij}) has to be determined. Given an upper bound W on the match-up times, one can set the match-up time on machine i to be $W_i = \max_{j \in J_i}\{s_j : s_j \leq W\}$. This is because the match-up times can be selected out of the start times of the jobs in the initial schedule. We define the set of jobs to be rescheduled as J^W, i.e. set of jobs that precede selected match-up times on the machines. Furthermore, we define $C_j^{\mathcal{S}}$ to be the manufacturing cost of job j in $\mathcal{S}$. We denote the machining time capacity on machine i used by $\mathcal{S}$ as $D_i^{\mathcal{S}}$. Then, we can formulate the problem of minimizing total manufacturing cost of jobs in J^W with given match-up times as:

$$
\begin{aligned}
&\min \quad \sum_i \sum_{j \in J^W} (c_{ij}x_{ij} + f_{ij}(y_{ij})) - \sum_{j \in J^W} C_j^{\mathcal{S}} \\
&\text{s.t.} \quad \sum_{j \in J^W} (p_{ij}^u x_{ij} - y_{ij}) \leq W_i - D_i^{\mathcal{S}}, && i = 1, \ldots, m \quad (1)\\
&\text{(RCM)} \qquad y_{ij} \leq x_{ij}u_{ij}, && i = 1, \ldots, m \text{ and } j \in J^W \quad (2)\\
&\qquad \sum_{i=1}^{m} x_{ij} = 1, && j \in J^W \quad (3)\\
&\qquad x_{ij} \in \{0,1\},\ y_{ij} \in \mathbb{R}_+, && i = 1, \ldots, m \text{ and } j \in J^W. \quad (4)
\end{aligned}
$$

RCM is a mixed integer nonlinear programming problem for which Aktürk et al. (2009a) provided a strengthened conic quadratic formulation, and hence it can be solved efficiently by a commercial branch-and-bound software which employs second-order cone programming algorithms in solving subproblems.

Given the rescheduling problem above we focus on developing an anticipative scheduling approach to form an initial schedule. Given $\mathcal{A}$, under the assumption of a single disruption on one of the machines and the assumption that RCM problems to be solved to reschedule the remaining jobs, the problem that we deal with is to make the job sequencing decisions on each machine to form the initial schedule $\mathcal{S}$ so that the optimal solution of RCM can be improved. In the next section, we explore the probabilistic nature of downtime period on a machine and propose flexibility measures which estimate the ability of the jobs to absorb downtimes.

3. ANTICIPATIVE SCHEDULING ALGORITHM

We develop an anticipative scheduling approach to form an initial schedule. It is uncertain which machine will fail, at what time and how long it will take to repair a failed machine. We assume that the probability distributions for failure and repair times are known. When a disruption occurs it is critical to absorb the downtime as soon as possible and at minimum rescheduling cost. Therefore, it is critical which jobs are scheduled at and immediately after the downtime interval. So, we provide a set of flexibility measures to be evaluated for each job. We will use the flexibility measures in deciding which jobs are appropriate to schedule at risky time zones. An outline of proposed anticipative scheduling algorithm is given below.

Algorithm 1 Anticipative Scheduling Algorithm

Step 1. *Initial machine-job assignment, $\mathcal{A}$:* Find the minimum cost machine-job assignment for given jobs and machining time capacity levels (D_i);
Step 2. *Downtime Probability:* For each machine find the downtime probability function which gives the probability that the machine will be down at a time point t;
Step 3. *Flexibility Measures:* Develop a flexibility measure F_j for each job with respect to:

- Processing time,
- Compressibility range,
- Second derivative of the compression cost function,
- Average slope of the compression cost function,
- Machine-job reallocation cost estimate;

Step 4. *Probabilistic Sequencing Algorithm:* Sequence the jobs on the machines by placing the most flexible job, i.e. job with the highest F_j, to the time zone where the machine is most likely to be down.

3.1. **Initial Machine-Job Assignment.** As a first step of our anticipative scheduling algorithm, we solve a machine-job assignment problem to minimize the total manufacturing cost of given n jobs to be completed on m non-identical machines. A mathematical formulation of the problem is as follows:

$$
\text{(MJA)} \qquad
\begin{aligned}
\min \quad & \sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij}x_{ij}+f_{ij}(y_{ij})) \\
\text{s.t.} \quad & \sum_{j=1}^{n}(p_{ij}^{u}x_{ij}-y_{ij}) \leq D_i, && i=1,\ldots,m, && (5) \\
& y_{ij} \leq x_{ij}u_{ij}, && i=1,\ldots,m,\ j=1,\ldots,n, && (6) \\
& \sum_{i=1}^{m} x_{ij} = 1, && j=1,\ldots,n, && (7) \\
& x_{ij}\in\{0,1\},\ y_{ij}\in\mathbb{R}_+, && i=1,\ldots,m,\ j=1,\ldots,n. && (8)
\end{aligned}
$$

The difference between MJA and RCM is that MJA is solved for n jobs at the beginning when capacity on each machine is the initially available machining time D_i. MJA is a mixed-integer nonlinear programming problem which can be solved similar to the RCM problem by using the conic quadratic reformulation approach proposed by Aktürk et al. (2009a). Next, we define a downtime probability function and show how it is derived.

3.2. **Downtime Probability.** Given the failure time and repair time distributions for a machine, one can calculate the probability that it will be down at a certain time t. Let $\mathcal{X}_i$ be the random variable defining the failure time of machine i and $\mathcal{Y}_i$ be the random variable defining the repair time after a failure occurs, then we can define the probability that machine i will be down at time t as below:

$$P_i^d(t) = P(\mathcal{X}_i \leq t \leq \mathcal{X}_i + \mathcal{Y}_i)$$

While preparing the initial schedule $\mathcal{S}$, we can use $P_i^d(t)$ as a benchmark to sequence the jobs on the machine. In the rest of the paper, when it is not necessary to include index i, we will drop it from notation $\mathcal{X}_i, \mathcal{Y}_i$ and $P_i^d(t)$. We can calculate $P^d(t)$ as shown in Lemma 3.1.

Lemma 3.1. *Let $f_\mathcal{X}, F_\mathcal{X}, f_\mathcal{Y}$, and $F_\mathcal{Y}$ be probability density functions and distribution functions of continuous random variables $\mathcal{X}$ and $\mathcal{Y}$, respectively. Then,*

$$\begin{aligned} P^d(t) &= P(\mathcal{X} \le t \le \mathcal{X} + \mathcal{Y}) \\ &= \int_{-\infty}^{t} (1 - F_\mathcal{Y}(t-x)) \cdot f_\mathcal{X}(x)dx \\ &= \int_{0}^{\infty} (F_\mathcal{X}(t) - F_\mathcal{X}(t-y)) f_\mathcal{Y}(y)dy \end{aligned}$$

Proof. $P(\mathcal{X} \le t \le \mathcal{X} + \mathcal{Y}) = \int_{-\infty}^{t} P(\mathcal{X} \le t \le \mathcal{X} + \mathcal{Y} | \mathcal{X} = x) \cdot f_\mathcal{X}(x)dx$
$= \int_{-\infty}^{t} P(\mathcal{Y} \ge t - x) \cdot f_\mathcal{X}(x)dx$
$= \int_{-\infty}^{t} (1 - F_\mathcal{Y}(t-x)) \cdot f_\mathcal{X}(x)dx.$
Similarly conditioning on y immediately brings up the second equality. □

Lemma 3.1 defines $P^d(t)$ which gives the probability that a machine will fail before or at time t and will not available at time t. The next property states that $P^d(t)$ can have a unique local maximum in the interval $[0, \infty]$.

Lemma 3.2. *If f_x is unimodal in the interval $[0, \infty]$, i.e. it has a unique local maximum in the interval $[0, \infty]$, then $P^d(t)$ is also unimodal in the interval $[0, \infty]$.*

Proof. The first derivative of $P^d(t)$ is:

$$\frac{\partial P^d(t)}{\partial t} = (1 - F_\mathcal{Y}(0)) f_\mathcal{X}(t) - \int_{-\infty}^{t} f_\mathcal{Y}(t-x) f_\mathcal{X}(x)dx$$

The second term in the derivative expression is an integral of the multiplication of nonnegative functions. Hence, the term is nonnegative and increasing in given interval. Since the first term is unimodal by definition, the derivative of $P^d(t)$ can take the value zero only at a single point in the interval and hence $P^d(t)$ is unimodal. □

Lemma 3.2 implies that the downtime probability is first increasing and then decreasing. So, there is a time point where the downtime probability is at its maximum. Lemma 3.2 is quite important in designing the probabilistic sequencing algorithm which will be discussed in Section 3.4. Lemma 3.2 implies that $P^d(t)$ takes its minimum value at one of the boundary points of operating interval $[0, D_i]$. If D_i is large enough such that the $P^d(t)$ is minimized in the interior of $[0, D_i]$, then the jobs which are not flexible to reschedule should be scheduled close to the boundary points.

For the experimental study given in Section 4, we considered four probability distribution pairs for failure and repair times, which are Normal-Normal (Norm-Norm), Triangular-Normal (Tri-Norm), Exponential-Normal (Exp-Norm), and Exponential-Exponential (Exp-Exp) distributions. The first distribution of each pair is the failure time distribution and the second distribution is for the repair time. In each case, density function for the failure time distribution is unimodal in the considered interval, so they satisfy the condition of Lemma 3.2 and hence $P^d(t)$

is a unimodal function in each case. We present the derivation of $P^d(t)$ for each distribution pair in Appendix. Next, we define the flexibility measures which we use to assess the flexibility of each job with respect to considered rescheduling problem.

3.3. **Flexibility Measures.** In our anticipative approach, the goal is to prepare an initial schedule, i.e. find a job sequence on each machine, which is flexible against machine breakdowns with respect to rescheduling cost. Thus, as the third step of our approach, we introduce new flexibility measures. We consider a rescheduling problem in which the objective is to minimize the rescheduling cost subject to a given match-up time. We define a "flexible" schedule as the one which can be repaired at minimum possible manufacturing cost increase after a machine breakdown. In order to find a job sequence on a machine, it is crucial to use a measure which ranks jobs by their ability to absorb a disruption at minimum cost. Below we list the measures which could affect our anticipative scheduling decisions and we explain why each measure is critical for the considered rescheduling problem.

Processing time (p_j): is the processing time of a job j in the initial schedule, i.e. $p_j = p^u_{ij} - y^{\mathcal{S}}_{ij}$ where i is the machine that job j is assigned in $\mathcal{A}$. Processing time is critical for the rescheduling problem since placing shorter jobs around a downtime period could allow to distribute the required compression to more jobs and hence improve cost performance since the cost functions are convex.

Compressibility(w_j): is the available amount of further compression for job j on its current machine in $\mathcal{A}$. It is assigned to $w_j = \{u_{ij} - y^{\mathcal{S}}_{ij}\}$ where i is job j's current machine. Compressibility of a job is the ability to occupy less capacity on a machine and hence gives us a measure on how much of the downtime it can absorb after a disruption. The higher the compressibility of jobs in the downtime zone, it is possible to achieve the smaller match-up times.

Second derivative of compression cost function(f''_j): Suppose that job j is assigned to machine i in $\mathcal{A}$ and selected optimal compression level is $y^{\mathcal{S}}_{ij}$, then $f''_j = \frac{\partial^2 f(y^{\mathcal{S}}_{ij})}{\partial y^2_{ij}}$. By definition, the second derivative of a function gives the change rate of the first derivative at a point where it is evaluated. Lemma 3.3 gives an optimality property for the problem MJA for the compression levels on the processing times and first derivatives of cost functions for the jobs assigned on the same machine.

Lemma 3.3. *Let $y^*_{ij_1}$ and $y^*_{ij_2}$ be the optimal compression levels for jobs j_1 and j_2 assigned on machine i in the optimal solution to MJA. Let the corresponding first derivatives of the compression cost functions be $\lambda_{j_1} = (\partial f_{ij_1}/\partial y_{ij_1})(y^*_{ij_1})$ and $\lambda_{j_2} = (\partial f_{ij_2}/\partial y_{ij_2})(y^*_{ij_2})$. Then, one of the following statements holds:*
i. $\lambda_{j_1} = \lambda_{j_2}$ *and* $0 \leq y^*_{ij_1} \leq u_{ij_1}$ *and* $0 \leq y^*_{ij_2} \leq u_{ij_2}$*;*
ii. $\lambda_{j_1} < \lambda_{j_2}$ *and* $y^*_{ij_1} = u_{ij_1}$ *and* $0 \leq y^*_{ij_2} \leq u_{ij_2}$*;*
iii. $\lambda_{j_2} < \lambda_{j_1}$ *and* $0 \leq y^*_{ij_1} \leq u_{ij_1}$ *and* $y^*_{ij_2} = u_{ij_2}$*.*

Proof. It can easily be observed that a solution, in which there exists two jobs which violate the lemma, can be improved by changing the compression levels of the jobs. □

Lemma 3.3 states that, in $\mathcal{A}$, on each machine the first derivatives of compression cost functions of jobs at optimal compression levels are equal. Lemma 3.3 shows that an exception can be fully compressed jobs ($y^*_{ij} = u_{ij}$). This implies that in $\mathcal{A}$ marginal compression cost values are equal for the jobs assigned to the same machine. Then, it is intuitive to look at the second derivatives of the cost functions to estimate the cost function behaviors. If $f''_{j_1} > f''_{j_2}$, then we can say that the increase rate of the derivative of job j_1 is higher than j_2 and hence we can expect the cost increase rate of job j_1 to be higher around the optimal solution. As a result, in order to minimize the compression cost required to absorb a downtime, we may place the jobs with smaller second derivatives to the regions where a possible downtime is more likely to occur.

Delta(Δ_i): Δ is the average slope of the compression cost function of job j on machine i given in $\mathcal{A}$ in the interval $[y^{\mathcal{S}}_{ij}, u_{ij}]$. We calculate this flexibility measure as

$$\Delta = \frac{f(u_{ij}) - f(y^{\mathcal{S}}_{ij})}{u_{ij} - y^{\mathcal{S}}_{ij}}.$$

Δ is another measure which provides us information on the behavior of compression cost function. Different than f'', Δ not only considers a local behavior but it looks ahead to see what happens if the job is fully compressed. When sequencing the jobs on a machine, it would be better to place jobs with smaller Δ values to the time periods with higher probability of downtime.

When rescheduling jobs, we may need to reallocate some jobs to other machines in order to minimize the rescheduling cost. Usually, it is more likely to move jobs from the disrupted machine to other machines. Then, estimating the cost change that will occur when we move a job from its original machine to another machine can also help to rank the flexibility of the job. The cost change lower bound for moving job j from machine i_1 to machine i_2 can be calculated as below:

Lemma 3.4. *For a given machine-job assignment $\mathcal{A}$, let λ_{i_1} and λ_{i_2} be the derivative values of compression cost functions of jobs on machines i_1 and i_2 respectively, and $y^{\mathcal{A}}_{i_1 j}$ be the compression of job j on machine i_1. Then, a lower bound for the cost change that will result by moving job j from machine i_1 to i_2 is as stated below:*

$$LB(j:(i_1 \to i_2)) = -\lambda_{i_1}(p_{i_1 j} - y^{\mathcal{A}}_{i_1 j}) - c_{i_1 j} - f_{i_1 j}(y^{\mathcal{A}}_{i_1 j}) + c_{i_2 j} + f_{i_2 j}(\hat{y}_{i_2 j}) + \lambda_{i_2}(p_{i_2 j} - \hat{y}_{i_2 j}),$$

where $\hat{y}_{i_2 j} = \min((\partial f_{i_2 j}/\partial y_{i_2 j})^{-1}(\lambda_{i_2}), u_{i_2 j})$.

For the proof of Lemma 3.4, we refer the reader to Gürel and Aktürk (2007). Given the cost change lower bounds for moving a job from its current machine to the other machines, we can define the following flexibility measure for each job.

Minimum Re-allocation Cost Lower Bound(LB_j): The minimum cost change lower bound for moving job j from its initially assigned machine in $\mathcal{A}$ to the some other machine can be calculated as follows:

$$LB_j = \min_{i_2}\{LB(j:(i_1 \to i_2)) : \forall i_2,\ i_2 \neq i_1\}$$

where i_1 is the initially assigned machine of job j. This measure can be used such that we can place the jobs with smaller reallocation cost to the time periods where the downtime probability is higher.

We have defined a set of measures which may help to make sequencing decisions. We can also combine these measures to form a new flexibility measure as defined below:

Definition 1. A flexibility measure F is a multiplication of integer powers of several flexibility factors. More formally,

$$F_j = (p_j)^{\alpha_1} \times (w_j)^{\alpha_2} \times (f''_j)^{\alpha_3} \times (\Delta_j)^{\alpha_4} \times (LB_j)^{\alpha_5}$$

where $\alpha_k \in \mathbb{Z}$.

In order to clarify how these flexibility factors could be used as a sequencing rule, $\max\{\frac{1}{p}\}$ corresponds to the SPT rule, whereas $\max\{\frac{w^2}{p\cdot\Delta\cdot LB}\}$ is a composite rule that combines four of them into a single sequencing rule.

Next, we give an algorithm which schedules the jobs on their assigned machines by considering the downtime probability $P^d(t)$ function of each machine and the flexibility measure F_j for each job.

3.4. **Probabilistic Sequencing Algorithm.** Probabilistic sequencing algorithm finds a job sequence on a given machine by considering the flexibility measures of the jobs and the downtime probability function for the machine. The goal is to place the jobs with maximum flexibility to the positions with the maximum probability of downtime. The interval considered for machine i in this algorithm is $[0, D_i]$. Let F_j be the flexibility measure of job j. F_j can be easily computed for all jobs. In the first step of the algorithm, we order the jobs in J_i in ascending order of F_j. Then, starting with the first job in the list, the algorithm places each job into the schedule one by one. For the first job, say job j, the available interval is $[0, D_i]$. Proposed algorithm evaluates two alternatives. The first one is placing the job at the beginning of the available interval. The second alternative is placing it at the end. The algorithm checks the downtime probability at the mid-point of the job in both cases, i.e. checks $P^d(p_j/2)$ and $P^d(D_i - p_j/2)$. If the first probability is less, then the algorithm places the job at $[0, p_j]$. Else, the job is placed at $[D_i - p_j, D_i]$. Then, the algorithm updates the available interval and takes the next job from the list.

We check only the boundaries of the available interval, since we know from Lemma 3.2 that if $f_{\mathcal{X}}$ is a unimodal function then the probability function $P^d(t)$ is also unimodal in the interval $[0, D_i]$ for machine i. $P^d(t)$ being unimodal implies that the minimum downtime probability in the interval is found at one of the boundary points of the interval. Therefore, the algorithm tries to place the least flexible jobs first to the start or end points of the interval, i.e. to the position with minimum downtime probability. We give the step by step definition in Algorithm 2. In the next section we give the experimental results for the probabilistic sequencing algorithm.

Algorithm 2 Probabilistic Sequencing Algorithm

Require: Machine i with $P^d(t)$ and available interval $[t_s, t_e]$.
Require: Set of jobs J_i with F_j and p_j for each $j \in J_i$.
  Initialize: Order the jobs in J_i in ascending order of F_j's;
  Initialize: $t_s = 0$ and $t_e = D_i$;
  for each job $j \in J_i$ **do**
    if $P^d(t_s + p_j/2) \ \leq \ P^d(t_e - p_j/2)$ **then**
      Schedule job j at $[t_s, t_s + p_j]$.
      $t_s = t_s + p_j$.
    else
      Schedule job j at $[t_e - p_j, t_e]$.
      $t_e = t_e - p_j$.

4. COMPUTATIONAL STUDY

In the computational study, we tested the performance of Algorithm 2 using alternative flexibility measures F_j described in Section 3.3. We compared rescheduling performance on the initial schedules achieved by Algorithm 2 against the performance of initial schedules achieved by using the SPT rule. Adiri et al. (1989) consider, for the first time, the flow-time scheduling problem when the machine faces breakdowns at stochastic time epochs, the repair time is stochastic, but the processing times are constant. They prove that the problem is NP-hard and show that the SPT rule minimizes the expected total flow time if the time to breakdown is exponentially distributed. Lee and Liman (1992) study the deterministic equivalent of this problem in the context of a single scheduled maintenance and find a tight performance bound of 9/7 for the SPT rule.

In the test problems, number of jobs is $n = 100$, and number of machines is $m = 3$. We generated manufacturing cost (c_{ij}) for each machine-job pair randomly from Uniform[2.0,6.0]. We generated k_{ij} coefficient of the compression cost function $\left(f_{ij}(y_{ij}) = k_{ij} y_{ij}^{a_{ij}/b_{ij}}\right)$ from Uniform[1.0,3.0] and a_{ij}/b_{ij} from Uniform [1.1,3.1]. We generated processing time upper bound p_{ij}^u from Uniform [1.0,5.0]. In practice, one can expect a correlation between processing time upper bound and the maximum compressibility at least due to the fact that processing time upper bound is an upper bound for the maximum compressibility. Thus, we generated the compression bound u_{ij} from $p_{ij}^u \times$ Uniform [0.5, 0.9]. We set the machining capacity of each machine as below:

$$D_i = 0.2 \times \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij}^u}{m} \ .$$

In order to construct initial schedules, we first solved the machine-job assignment problem given in Section 3.1. We sequenced the jobs assigned on each machine by using Algorithm 2 which employed each of the following proposed flexibility measures: $\frac{1}{pf''}$, $\frac{1}{f''LB}$, $\frac{1}{pf''LB}$, $\frac{1}{p\Delta LB}$, $\frac{w}{pf''LB}$, $\frac{w}{p\Delta LB}$ and $\frac{w^2}{p\Delta LB}$. We also formed an initial schedule by using the SPT rule on each machine, which gives the minimum total completion time.

For $\mathcal{X}_i$ and $\mathcal{Y}_i$, we used four different distribution pairs consisting of Normal-Normal, Triangular-Normal, Exponential - Normal, and Exponential - Exponential. Having formed initial schedules, we randomly selected a machine to fail. We generated a failure time, $\mathcal{X}_i$, and a repair time $\mathcal{Y}_i$ for each machine i.

In failure time distribution, mean time to fail is MTTF $= 0.3 \cdot D[i]$. For exponential distribution, $\lambda = 1/\text{MTTF}$. For normal distribution, standard deviation is generated by $\sigma = 0.5 \cdot \text{MTTF} \cdot Z$ where $Z \sim$ Uniform$[0,1]$. We used 0, $D[i]$ and MTTF as the parameters of a triangular distribution. In repair time distribution, we used two different levels of mean time to repair, denoted as MTTR. For all distributions except exponential distribution, we used MTTR $= 0.1 \cdot D[i]$ and MTTR $= 0.15 \cdot D[i]$. For exponential distribution, we adjusted MTTR and MTTF parameters in order to avoid high variability, since high variability leads to long failure or repair times which would result infeasible rescheduling problems.

For each $\mathcal{S}$, we first solved the minimum match-up time problem to find ($\bar{W}_{min}$). Then, for $\bar{W} = \bar{W}_{min} + \beta \times (D[i] - \bar{W}_{min})$ we solved the RCM problem for four different levels of $\beta = 0.1, 0.15, 0.2, 0.25$, so that we could generate alternative time/cost trade-offs. We took 10 replications for each setting. All experiments were

TABLE 1. Mean Rescheduling Cost Performance $R(\%)$ for the Norm-Norm Case

MTTR	β	Flexibility Measures $\frac{w^2}{p\Delta LB}$	$\frac{w}{p\Delta LB}$	$\frac{1}{f''LB}$	$\frac{w}{pf''LB}$	$\frac{1}{pf''LB}$	$\frac{1}{p\Delta LB}$
Low	0.1	24.6	14.7	17.2	5.9	12.8	18.5
	0.15	19.8	6.2	8.4	3.3	8.5	9.0
	0.20	15.0	6.7	9.0	1.2	9.5	9.6
	0.25	15.4	5.0	8.3	1.1	6.7	8.7
	Total	18.7	8.2	10.7	2.9	9.4	11.4
High	0.1	31.7	22.3	20.3	23.9	12.2	13.4
	0.15	23.1	18.2	16.9	18.4	9.1	6.5
	0.20	22.5	19.3	13.8	15.6	11.3	4.7
	0.25	16.2	12.6	9.3	8.2	6.1	1.6
	Total	23.4	18.1	15.1	16.5	9.7	6.5
Total		21.0	13.1	12.9	9.7	9.5	9.0

performed using ILOG Cplex Version 9.1 on a 12×400 MHz UltraSPARC CPU and 3GB memory workstation Sun HPC 4500 with the operating system Solaris 2.7.

For each instance, we calculated a relative difference between rescheduling costs of schedules achieved by Algorithm 2 and SPT rule. We define the relative difference R as follows:

$$R = 100 \times \frac{Cost_{SPT} - Cost_F}{Cost_F}.$$

in which $Cost_{SPT}$ is the rescheduling cost of an SPT schedule for the considered failure-repair times and match-up time. $Cost_F$ is the rescheduling cost of a schedule achieved by Algorithm 2 using flexibility measure F.

Table 1 shows average R results for Norm-Norm case. Flexibility measure $\frac{w^2}{p\Delta LB}$ achieves the best cost performance against the SPT rule with an average relative difference of 21%. Among all flexibility measures tested, the worst average value for R is 9%. From the first two flexibility measures given in Table 1, we observe that as we multiply the second measure with w, we get the first measure which performs significantly better than the second one. As discussed in Section 3.3, w measures the available amount of compression on a job and hence it is important in solving rescheduling problems. From Table 1, it can be observed that as the match-up time level increases, average value of R is likely to decrease. This means as we allow distributing the effect of a disruption to a larger portion of initial schedule, we can expect that the gain to be achieved by considering flexibility of jobs declines. In other words, as the match-up time level decreases, it becomes more critical to place more flexible jobs around downtime period.

For the same flexibility measures, first two columns of Table 2 gives 95% confidence interval bounds for the average value of R. Given bounds clearly indicate that they are significantly better than the SPT rule in achieving lower rescheduling costs. The highest lower bound for a confidence interval is achieved by the measure $\frac{w^2}{p\Delta LB}$ which is 16%. In the same table, we also report the number times Algorithm 2 achieve better rescheduling cost performance than the SPT rule. The best performance is by $\frac{w^2}{p\Delta LB}$ with 68 problems out of 80. The next best performance is by $\frac{1}{pf''LB}$ with 65 out of 80. We see that all measures perform better than the SPT sequence with the worst one performing better in 52 problems out of 80.

TABLE 2. Confidence Intervals for the mean R and number of times best for the Norm-Norm Case

Flexibility	95 % CI on Mean R		# of times best		
Measures	Lower Bound	Upper Bound	MTTR Low	MTTR High	total
$\frac{w^2}{p\Delta LB}$	16.0	26.0	33	35	68
$\frac{w}{p\Delta LB}$	8.0	18.2	25	32	57
$\frac{1}{f''LB}$	7.5	18.3	25	32	57
$\frac{w}{pf''LB}$	5.0	14.4	23	29	52
$\frac{1}{pf''LB}$	5.3	13.7	33	32	65
$\frac{1}{p\Delta LB}$	4.9	13.1	27	25	52

Table 3 shows average values of R for selected flexibility measures for the Tri-Norm case. The best performing flexibility measure is again $\frac{w^2}{p\Delta LB}$ with 24.5%. The next best performance on the average is by $\frac{1}{f''LB}$. On the other hand as MTTR is increased, $\frac{w}{p\Delta LB}$ achieves the second best cost performance against the SPT rule. The maximum value of R observed in experimental results is 102.6% which means the rescheduling cost of SPT sequence is more than twice of the rescheduling cost of a schedule prepared by using the flexibility measure $\frac{w^2}{p\Delta LB}$.

TABLE 3. Mean Rescheduling Cost Performance $R(\%)$ for the Tri-Norm Case

		Flexibility Measures					
MTTR	β	$\frac{w^2}{p\Delta LB}$	$\frac{1}{f''LB}$	$\frac{w}{p\Delta LB}$	$\frac{w}{pf''LB}$	$\frac{1}{pf''LB}$	$\frac{1}{pf''}$
Low	0.1	27.2	26.2	8.8	20.0	11.7	15.1
	0.15	27.4	22.1	11.6	19.4	11.4	11.5
	0.2	21.1	17.1	6.6	12.3	6.6	8.0
	0.25	19.3	15.0	10.4	10.5	5.1	7.0
	Total	23.8	20.1	9.4	15.5	8.7	10.4
High	0.1	34.4	20.7	21.2	12.3	16.0	15.9
	0.15	24.2	15.7	18.7	7.7	10.9	9.1
	0.20	24.8	13.3	20.0	5.3	11.0	7.4
	0.25	17.7	7.7	13.4	0.1	7.2	2.3
	Total	25.3	14.4	18.3	6.3	11.3	8.7
Total		24.5	17.2	13.8	10.9	10.0	9.5

Table 4 gives the 95% confidence intervals for average R. Proposed sequencing algorithm using the selected flexible measures achieve a significant improvement in the rescheduling cost compared to the SPT rule. The best lower bound for the 95% confidence interval for the mean R is 18.6 for the flexibility measure $\frac{w^2}{p\Delta LB}$. From Table 4, we can see how many times each flexibility measure achieves better rescheduling cost than the SPT after rescheduling. $\frac{w^2}{p\Delta LB}$ is the best measure which outperformed the SPT rule in 68 cases out of 80. $\frac{w}{p\Delta LB}$ is the second best measure with 59 times and $\frac{1}{f''LB}$ is the third with 58 times. All flexibility measures included in Table 4 outperformed the SPT sequenced schedules in most of the cases.

TABLE 4. Confidence Intervals for the mean R and number of times best for the Tri-Norm Case

Flexibility	95 % CI on Mean R		# of times best		
Measures	Lower Bound	Upper Bound	MTTR Low	MTTR High	total
$\frac{w^2}{p\Delta LB}$	18.6	30.4	34	34	68
$\frac{1}{f''LB}$	12.0	22.4	27	31	58
$\frac{w}{p\Delta LB}$	9.2	18.4	29	30	59
$\frac{w}{pf''LB}$	6.1	15.8	31	21	52
$\frac{1}{pf''LB}$	5.6	14.4	19	37	56
$\frac{1}{pf''}$	4.7	14.4	26	21	47

Table 5 shows average R for different flexibility measures tested in Exp-Norm case. Flexibility measure $\frac{w}{p\Delta LB}$ achieves the best rescheduling cost performance on the average. We observe that performance of our algorithm against the SPT rule degrades slightly when exponential failure is considered. Exponential failure implies a decreasing failure rate which requires placing flexible jobs first in the sequence. When exponential failure is considered, we can expect SPT rule to find a job sequence which is quite similar to a sequence that Algorithm 2 would generate by using flexibility measure $\frac{1}{p}$. Hence, we can expect SPT to perform better in exponential failure case compared to other failure distributions.

TABLE 5. Mean Rescheduling Cost Performance $R(\%)$ for the Exp-Norm Case

		Flexibility Measures				
MTTR	β	$\frac{w}{p\Delta LB}$	$\frac{1}{f''LB}$	$\frac{1}{p\Delta}$	$\frac{w}{pf''LB}$	$\frac{w^2}{p\Delta LB}$
Low	0.1	6.5	5.6	1.5	2.4	2.8
	0.15	3.0	12.1	6.7	-0.8	2.7
	0.2	5.5	9.3	6.2	3.5	6.9
	0.25	3.7	5.5	5.2	-0.8	2.6
	Total	4.7	8.1	4.9	1.1	3.8
High	0.1	19.6	9.5	6.8	13.5	9.2
	0.15	13.6	7.1	7.9	9.6	7.5
	0.20	13.2	5.8	6.3	9.8	6.2
	0.25	13.4	7.0	5.7	7.5	4.7
	Total	14.9	7.4	6.7	10.1	6.9
Total		9.8	7.7	5.8	5.6	5.3

In Table 6, we give the confidence intervals for the mean R for Exp-Norm case. The results show that proposed sequencing algorithm significantly outperforms the SPT sequenced schedules in terms of rescheduling cost. Table 6 also includes how many times each flexibility measure achieves lower rescheduling cost compared to the SPT rule. The best measure is $\frac{w}{p\Delta LB}$ which finds a smaller cost in 59 problems out of 80. This rule achieves a better result than the SPT rule in almost all problems when the MTTR is high. The second best measure is $\frac{1}{f''LB}$ with 52 better solutions. We observe that except the measure $\frac{w}{pf''LB}$, the other measures perform better than the SPT rule both in terms of the average cost difference and the number of times achieve better rescheduling cost.

TABLE 6. Confidence Intervals for the mean R and number of times best for the Exp-Norm Case

Flexibility Measures	95 % CI on Mean R		# of times best		
	Lower Bound	Upper Bound	MTTR Low	MTTR High	total
$\frac{w}{p\Delta LB}$	5.7	13.9	22	37	59
$\frac{1}{f''LB}$	3.7	11.8	26	26	52
$\frac{1}{p\Delta}$	2.6	9.0	22	26	48
$\frac{w}{pf''LB}$	0.3	10.8	15	23	38
$\frac{w^2}{p\Delta LB}$	0.4	10.2	21	23	44

Table 7 provides the R values for the best five flexibility measures for the Exp-Exp case. The results show that $\frac{1}{f''LB}$ has achieved the best R performance of 9.1% on the average. For low MTTR, $\frac{1}{f''LB}$ achieves an R level of 11.8% which is best among all flexibility measures. On the other hand, when MTTR is high the best performance is by the flexibility measure $\frac{w}{p\Delta LB}$. As MTTR is increased, performance of flexibility measures improve except the measure $\frac{1}{f''LB}$.

TABLE 7. Mean Rescheduling Cost Performance $R(\%)$ for the Exp-Exp Case

MTTR	β	Flexibility Measures				
		$\frac{1}{f''LB}$	$\frac{w}{pf''LB}$	$\frac{1}{pf''LB}$	$\frac{1}{pf''}$	$\frac{w}{p\Delta LB}$
Low	0.1	8.7	5.7	-0.2	-0.1	1.9
	0.15	14.3	6.6	4.3	2.2	1.2
	0.2	13.0	5.6	5.7	2.8	-0.2
	0.25	11.1	6.1	7.6	0.7	-3.0
	Total	11.8	6.0	4.4	1.4	0.0
High	0.1	7.1	13.4	6.8	14.6	14.5
	0.15	7.2	10.5	6.8	11.1	15.6
	0.20	6.6	9.8	15.0	16.7	16.2
	0.25	4.8	10.4	16.5	13.2	12.4
	Total	6.4	11.0	11.3	13.9	14.7
Total		9.1	8.5	7.8	7.6	7.3

Despite lower average R values in exponential failure case, on the average our algorithm's performance is still significantly better than the SPT rule. Table 8 shows that the lower bound values of the 95% confidence intervals are greater than zero, so we can say that in terms of rescheduling cost, Algorithm 2 performs statistically better than the SPT rule. Flexibility measure $\frac{1}{f''LB}$ achieves the highest lower and upper bounds for the confidence interval. Table 8 also gives the number of times that the sequence achieved by Algorithm 2 achieves a lower rescheduling cost than the SPT sequence in the number of times best section. The results show that out of 80 problems solved, the algorithm using either flexibility measure $\frac{1}{f''LB}$ or $\frac{w}{pf''LB}$ achieves a lower cost in 54 problems. For lower MTTR, $\frac{1}{f''LB}$ performs best with better rescheduling cost in 29 problems out of 40. For higher MTTR, $\frac{1}{pf''}$ performs better in 30 problems out of 40. In general, we observe that all flexibility measures perform better both in terms of average cost difference and in terms of number of times achieving lower rescheduling cost compared to the schedules formed by the SPT rule.

TABLE 8. Confidence Intervals for the mean R and number of times best for the Exp-Exp Case

Flexibility Measures	95 % CI on Mean R		# of times best		
	Lower Bound	Upper Bound	MTTR Low	MTTR High	total
$\frac{1}{f''LB}$	4.2	14.0	29	25	54
$\frac{w}{pf''LB}$	3.3	13.7	25	29	54
$\frac{1}{pf''LB}$	2.9	12.7	24	25	49
$\frac{1}{pf''}$	2.5	12.8	22	30	52
$\frac{w}{p\Delta LB}$	0.9	13.8	22	28	50

When we check the worst case performance of proposed flexibility measures against the SPT sequence, we can state that worst performing initial schedules were achieved when both MTTR and β are low. When it is required to catch up initial schedule in a very short time, i.e. β is low, one can expect that SPT sequenced schedules can achieve best results by taking the advantage of placing more jobs within a shorter time slot.

Using given probability distributions of failure and repair times, we anticipate when and how long each machine could be down and by using designed flexibility measures we schedule the most flexible jobs to the most critical time zones on each machine. Our computational results indicate that combining the proposed probabilistic sequencing idea with proposed flexibility measures are quite efficient in preparing flexible schedules for solving rescheduling cost problems under match-up time limitations. We have tested proposed approach against the SPT sequencing rule and observed a statistically significant difference in rescheduling cost performance. We have also observed that in most of the cases our anticipative scheduling approach performs better than the SPT rule based initial schedules in terms of rescheduling costs. Our results indicate that when the failure-repair behavior pattern is known for a machine, it is quite critical to use the cost function and compression related information in forming initial schedules so that in case of a failure a schedule can be repaired at a reasonable cost. For example, our algorithm outperforms the SPT rule for normal distribution case since proposed downtime probability, $P^d(t)$, calculations more accurately capture the disruptive events due to gradual wear (e.g. expected values have an approximately symmetric behavior around a mean value), as opposed to random failures that are represented by an exponential distribution. In the next section, we give concluding remarks.

5. CONCLUSION

In this paper, we have proposed an anticipative scheduling approach for scheduling with controllable processing times. We showed that anticipative decision making in preparing initial schedules can avoid excessive rescheduling costs that may result by reactive processing time adjustments.

We have considered a rescheduling problem to minimize the increase in total manufacturing cost subject to a match-up time constraint. We have designed an anticipative scheduling algorithm which uses proposed flexibility measures that can estimate which jobs can absorb a possible disruption at lowest cost. Proposed algorithm also uses downtime probability functions in determining the job sequence on each machine. Computational results show that considering flexibility measures of jobs and probabilistic nature of machine breakdowns in preparing an initial schedule can significantly improve rescheduling cost performance. As a future research direction, it is possible to consider different reactive scheduling problems in different scheduling environments. This would require developing problem specific flexibility measures. We think that it may also be interesting to consider risky jobs as well as risky machines in preparing initial schedules.

Appendix. Derivation of $P^d(t)$ for the distributions used in the computational study

Norm-Norm Case: In this combination, both failure and repair times are assumed to have a normal distribution. If the failure time is expected to be symmetrically distributed around a mean, this combination is suitable. This is actually a realistic case if the machine breakdown is due to a gradual wear process.

Lemma A.1. *Let $\mathcal{X} \sim Normal(\mu_1, \sigma_1)$ and $\mathcal{Y} \sim Normal(\mu_2, \sigma_2)$.*
$P^d(t) = \int_0^\infty \int_{t-y}^t f_\mathcal{X}(x) \cdot f_\mathcal{Y}(y) dx dy$ *where* $f_\mathcal{Y}(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$ *and*
$f_\mathcal{X}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

Tri-Norm Case: This combination is triangular failure time and normal repair time. Tri-Norm is suitable if there is no distribution information for the failures but only the mean values are available.

Lemma A.2. *Let $\mathcal{X} \sim Triangular(a, b, c)$ and $\mathcal{Y} \sim Normal(\mu, \sigma)$. Then,*

$$P^d(t) = \begin{cases} \frac{2}{(b-a)(c-a)} \int_0^{t-a} \left(\frac{y(2t-y)}{2} - ay\right) f_\mathcal{Y}(y) dy + \int_{t-a}^\infty \frac{(t-a)^2}{(b-a)(c-a)} f_\mathcal{Y}(y) dy & if\; a \le t \le c \\ \\ \int_{t-a}^\infty A(t) f_\mathcal{Y}(y) dy + \int_0^{t-c} B(t,y) f_\mathcal{Y}(y) dy + \int_{t-c}^{t-a} C(t,y) f_\mathcal{Y}(y) dy & if\; c \le t \le b \end{cases}$$

where $A(t) = \frac{\frac{c-a}{b-a} + 2(bt - t^2/2 - bc + c^2/2)}{(b-a)(b-c)}$, $B(t,y) = \frac{2(by - ty + y^2/2)}{(b-a)(b-c)}$,

$C(t,y) = \frac{2(c^2/2 - ac - (t-y)^2/2 + c(t-y))}{(b-a)(c-a)} + \frac{2(bt - t^2/2 - bc + c^2/2}{(b-a)(b-c)}$.

Exp-Norm Case: Exponential failure is generally a logical approach as it has memoryless property. On the other hand, it may not be appropriate to use exponential repair time since memoryless property is suitable in a machining environment. We generally expect to have an approximately symmetric behavior around a mean value when we consider the repairing time of a machine. $P^d(t)$ of Exp-Norm case can be calculated as below:

Lemma A.3. *Let $\mathcal{X} \sim Exponential(\lambda)$ and $\mathcal{Y} \sim Normal(\mu, \sigma)$. Then, the down probability is calculated as*

$P^d(t) = \int_0^t (e^{-\lambda(t-y)} - e^{-\lambda t}) f_{\mathcal{Y}}(y)dy + \int_t^{\infty} (1 - e^{-\lambda t}) f_{\mathcal{Y}}(y)dy$
where $f_{\mathcal{Y}}(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$.

Exp-Exp Case: Exponential failure time and exponential repair time (Exp-Exp) is widely used in the stochastic literature. Therefore, we considered this case although we do not consider exponential repair time as realistic in our problem. Below, we derive the $P^d(t)$ for the Exp-Exp combination.

Lemma A.4. *Let $\mathcal{X} \sim Exponential(\lambda_x)$ and $\mathcal{Y} \sim Exponential(\lambda_y)$. Then,*
$P^d(t) = \frac{\lambda_x}{\lambda_y - \lambda_x}(e^{-\lambda_x t} - e^{-\lambda_y t})$.

Proof. By Lemma 3.1, $P^d(t) = \int_{-\infty}^t (1 - F_{\mathcal{Y}}(t)) \cdot f_{\mathcal{X}}(x)dx$
$= \int_0^t e^{-\lambda_y(t-x)} \cdot \lambda_x e^{-\lambda_x t} dx$
$= \lambda_x e^{-\lambda_y t} \cdot \int_0^t e^{(\lambda_y - \lambda_x)x} dx$
$= \frac{\lambda_x}{\lambda_y - \lambda_x}(e^{-\lambda_x t} - e^{-\lambda_y t})$. □

From Lemmas A.1- A.4, we see that a closed form expression for $P^d(t)$ is only available for the Exp-Exp combination, that might explain why it is widely used in the literature. For the other combinations, $P^d(t)$ can only be approximately calculated.

References

I. Adiri, J. Bruno, E. Frostig, and A.H.G. Rinnooy Kan. Single machine flow-time scheduling with a single breakdown. *Acta Informatica*, 26:679–696, 1989.

M.S. Aktürk and E. Görgülü. Match-up scheduling under a machine breakdown. *European Journal of Operational Research*, 112:81–97, 1999.

M.S. Aktürk, A. Atamtürk, and S. Gürel. A strong conic quadratic reformulation for machine-job assignment with controllable processing times. *Operations Research Letters*, 2009a. to appear.

M.S. Aktürk, A. Atamtürk, and S. Gürel. Match-up scheduling with manufacturing cost considerations. *Journal of Scheduling*, 2009b. to appear.

H. Aytug, M.A. Lawley, K. McKay, S. Mohan, and R. Uzsoy. Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, 161:86–110, 2005.

J.C. Bean, J.R. Birge, J. Mittenthal, and C.E. Noon. Match-up scheduling with multiple resources, release dates and disruptions. *Operations Research*, 39(3): 470–483, 1991.

S. Gürel and M.S. Aktürk. Optimal allocation and processing time decisions on non-identical parallel CNC machines: ϵ-constraint approach. *European Journal of Operational Research*, 183:591–607, 2007.

W. Herroelen and R. Leus. On the merits and pitfalls of critical chain scheduling. *Journal of Operations Management*, 19:559–577, 2001.

M.T. Jensen. Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures. *Applied Soft Computing*, 1:35–52, 2001.

R.K. Kayan and M.S. Aktürk. A new bounding mechanism for the CNC machine scheduling problems with controllable processing times. *European Journal of Operational Research*, 167:624–643, 2005.

C.Y. Lee and S.D. Liman. Single machine flow-time scheduling with scheduled maintenance. *Acta Informatica*, 29:375–382, 1992.

J. Leon, S.D. Wu, and R.H. Storer. Robustness measures and robust scheduling for job shops. *IIE Transactions*, 26:32–43, 1994.

R. Leus and W. Herroelen. Scheduling for stability in single-machine production systems. *Journal of Scheduling*, 10:223–235, 2007.

S.V. Mehta and R.M. Uzsoy. Predictable scheduling of a job shop subject to breakdowns. *IEEE Trans. Robot. Autom.*, 14:365–378, 1998.

D. Shabtay and G. Steiner. A survey of scheduling with controllable processing times. *Discrete Applied Mathematics*, 155(13):1643–1666, 2007.

B. Yang and J. Geunes. Predictive-reactive scheduling on a single resource with uncertain future jobs. *European Journal of Operational Research*, 189:1267–1283, 2008.