Predictive heuristics to generate robust and stable schedules in single machine systems under disruptions

Zeinab Abtahi[a] [1], Rashed Sahraeian*[a] [2], Donya Rahmani[b] [3]

[a]*Department of Industrial Engineering, College of Engineering, Shahed University, Persian Gulf Expressway, Tehran, Iran.*

[b]*Department of Industrial Engineering, K.N. Toosi University of Technology, Mirdamad Ave, Tehran, Iran.*

[1] Email: *z.abtahi@shahed.ac.ir*, Telephone: (98 21) 51212040, Mobile no: (98) 9371190269, Postal Code: *6817734456*.

[2] Corresponding author. Email: *sahraeian@shahed.ac.ir*, Telephone: (98 21) 51212040, Mobile no: (98) 9122145489, Postal Code: *3319118651*.

[3] Email: *drahmani@kntu.ac.ir*, Telephone: (98 21) 84063357, Mobile no: (98)9122820180, Fax: (98 21) 88674858, Postal Code: *1999143344*.

Abstract

The present paper examines the problems of single machine scheduling under disruptions with uncertain processing times. The goal is to achieve schedules that are simultaneously stable and robust. In order to handle such problems, in addition to exact solution approaches, a general predictive two-stage heuristic algorithm is proposed. In the first stage of the algorithm, the optimal robust schedule is generated by only considering the uncertain job processing times and forgoing the breakdown disruptions. In the second stage, adequate additional times are embedded in job processing times to enhance stability. Extensive computational experiments are carried out to test the performances of the proposed methods. The achieved results show the superiority of the proposed general predictive heuristic approach over the common methods in the literature.

Keywords: Predictive heuristic, uncertain processing times, disruption, stable-robust scheduling, single machine.

1. Introduction

Scheduling problems are exposed to uncertainties resulting from unexpected disruptions, such as machine breakdowns, processing time variations, uncertain due dates and other stochastic events, which in turn will affect the availability of machines (e.g. [1-12]). This type of availability limitation increases the complexity of any scheduling problem, even in a single machine environment, and can prevent the schedule from its planned performance.

Among different approaches used to handle the uncertainties in machine availability, the reactive and predictive scheduling methods have attracted the most attention due to their high applicability (e.g. [1-4]). As the name suggests, in reactive scheduling, the system tries to find the highest level of performance after the occurred disruptions. In other words, in reactive methods, the initial schedule is revised to suit the new changes. For better performance, rescheduling is done in a way that the differences between the initial schedule and the reactive schedule are minimized. In predictive methods, however, possible disruptions are considered while generating initial schedules. In such cases, policy systems, backup plans or extra resources are set in advance to respond to the future disruptions. This way the final goal remains the same, whether or not the disruption occurs.

While defining goals in uncertain environments, in addition to the common objectives, such as the makespan, flow time, total tardiness, etc., two other measures should be considered; robustness (quality robustness) and stability (solution robustness) of the schedules. Despite various definitions in the literature for robustness, the main idea is to "*find a solution to the optimization problem that is not necessarily optimal but remains feasible and still has good performance when the parameters of the problem change*" [5]. On the other hand, the variability caused by the unforeseen disruptions is addressed by stability, i.e. when a realized schedule does not deviate from the initial one despite the disruptions, this schedule is stable [2]. To gauge the robustness of a schedule in uncertain environments, usually the expected value of the objective is considered, e.g. the expected total (realized) flow time [[2], [4]] and the expected total (realized) tardiness [[2], [6]].

$$\begin{cases} RM1 = E\left[\sum_{j=1}^{n} C_j^r\right] \\ RM2 = E\left[\sum_{j=1}^{n} \max(0,\ C_j^r - d_j)\right] \\ RM3 = \left(E\left[\sum_{j=1}^{n} C_j^r\right] - E\left[\sum_{j=1}^{n} C_j\right]\right) \end{cases}$$

where C_j^r is the realized (expected) completion times of job j, C_j is the (expected) initial completion time of job *j* and *d* represents the common due date of jobs. The most frequent way to measure *stability* (the deviation between initial and realized schedules) is to compare job completion times [2]. Based on this comparison, three stability measures are commonly used in the related papers: the sum of the squared differences, the sum of variances of the realized completion times and the sum of absolute differences [2].

$$\begin{cases} SM1 = E\left[\sum_{j=1}^{n} (C_j - C_j^r)^2\right] \\ SM2 = E\left[\sum_{j=1}^{n} \left(E\left(C_j^r\right) - C_j^r\right)^2\right] \\ SM3 = E\left[\sum_{j=1}^{n} \left|C_j - C_j^r\right|\right] \end{cases}$$

In general, the performance of the realized schedule is the main concern of practitioners rather than the planned or estimated performance of the initial schedule. Hence, optimizing the former may be more appropriate than optimizing the latter and robustness is a practical performance measure. A schedule serves as a master plan for other shop floor activities in addition to production tasks, such as determining delivery dates, release times, and planning requirements for secondary resources such as tools, fixtures, etc. Any deviation from the production schedule can disrupt these secondary activities and increase system nervousness. Thus, stability (solution robustness) is more and more important nowadays, especially for the Just-In-Time (JIT) production systems.

Based on the literature, stability and robustness were considered separately to cope with the stochastic disruptions in the scheduling problems (e. g. [2], [6-9]). However, considering bi-objective robustness and stability optimization problem enhances the flexibility of the schedule against changes in addition to preserving the feasibility of the schedule. A linearized combination of individual objective functions is a common approach to form multiple-objective problems [1]. Ergo, in this paper, using the linear combination of robustness and stability measures, first *three* scheduling problems with disruptions are defined.

$$Z1 = \alpha.RM3 + (1 - \alpha).SM1 \quad (1)$$

$$Z2 = \alpha.RM3 + (1 - \alpha).SM2 \quad (2)$$

$$Z3 = \alpha.RM2 + (1 - \alpha).SM3 \quad (3)$$

where $0 < \alpha < 1$. The objective function of the first, second, and third problems are respectively represented by *Z1*, *Z2* and *Z3*.

In the first and second problems, two bi-objective problems of finding an optimal robust and stable schedule for a single machine under job processing time uncertainty and machine breakdowns disruption are optimized analytically based on some theorems. The third problem is defined with total tardiness as the primary objective for single machine scheduling with uncertain job processing times and the random breakdowns. The problem of minimizing total tardiness is known to be *NP*-hard even if certain job processing times are considered and no machine breakdowns occur [6]. Briskorn et al. [5] proposed a pseudo-polynomial time algorithm based on the dynamic programming to solve this problem. Lin Liu et al. [1] applied the genetic algorithm (GA) to produce a robust and stable schedule to minimize the total weighted tardiness as the main objective of a single machine problem with random machine breakdowns. We propose predictive-reactive heuristic methods to solve the third problem, and show the effectiveness of the proposed methods in comparison to the righting shift (*RS*) method, which is the preferred policy in the face of machine disruption [10].

- The uncertain processing times and the machine breakdowns are regarded as system disruptions.
- Stability and robustness are considered simultaneously in three stochastic single-machine scheduling problems.
- Two special cases with simultaneous stability and robustness measures are analytically optimized based on some theorems.
- Predictive robust and stable approaches are proposed to cope with disruptions.

The remainder of the paper is organized as follows. Section 2 reviews the related literature. In section 3, the bi-objective single-machine scheduling problems are defined. In section 4, the exact and heuristic solution methods are described. Next, the algorithms are tested using benchmark instances and the results are reported in section 5. Finally, the paper is concluded in section 6.

2. Literature review

When dealing with uncertainties, job-related properties are considered to be random, or the machine is subjected to random breakdowns or both. Adiri et al. [11] considered the problem of minimizing the total flow time in a single-machine environment subject to random machine breakdowns. In their study, only one machine breakdown occurs. They showed that if the distribution function of the time to breakdown is concave, then the flow time could be stochastically minimized by the Shortest Processing Time first (SPT) rule. For the case of multiple breakdowns, it was proven that the SPT rule minimizes the total flow time when the times to breakdowns are exponentially distributed [11]. Ganji et al. [12] focus on single machine scheduling with flexible unavailability constraint (with the unknown starting time of the unavailability period) to minimize the maximum earliness.

Wu et al. [13] studied the single machine rescheduling problem with machine disruption failures. They rescheduled the jobs so that the minimum makespan was obtained with a high degree of schedule stability. They considered two criteria for stability; the deviation of the revised schedule in terms of job starting times (similar to *SM*3) and the deviation of the revised schedule from the original schedule in terms of the sequence of the jobs. They also used pairwise swapping methods and a genetic algorithm to obtain non-dominated solution sets.

Mehta and Uzsoy [7] worked on generating a stable schedule in a single machine system with machine disruption failures. They used the maximum lateness as a performance measure and generated stable initial schedules by inserting idle times in schedules that optimize system performance. O'Donovan et al. [6] worked on generating stable schedules with machine breakdowns. They used the total tardiness as the performance measure; the stability was measured by the absolute completion time deviations from the initial schedule. Lin Liu et al. [1] proposed a robust and stable schedule based on GA to minimize the total weighted tardiness of a single machine with random machine breakdowns.

In the classic scheduling literature, the job processing times are assumed known and constant which may not be true in all situation, such as deteriorating job [14], cases with learning effect [[15], [16], [17]], and the uncertainty in job processing times' durations [18]. Yang et al. [9] proposed a robust approach based on some heuristics in cases with job processing time uncertainties to minimize the sum of the completion times. They showed that the robust version of the sum of the completion time problem is *NP*-complete even for very restricted cases. Goren et al. [2] studied a single-machine problem where the performance measure is the total flow time and the source of uncertainty is the processing time variability and random machine breakdowns. They proposed a branch-and-bound algorithm and two $O(n \log n)$ surrogate relaxation heuristics that utilized this procedure to generate robust schedules, and compared their solutions to the Shortest Expected Processing Time (*SEPT*) solution. They observed that the *SEPT* performs poorly in terms of the robustness. Moreover, a novel algorithm is proposed to minimize the makespan under at most one machine breakdown to schedule the uniform processors [19].

Rahmani [18] proposed a proactive-reactive two-stage method to hedge against the processing time uncertainty and the unexpected machine breakdowns in two-machine flow shop scheduling problem. Multi-factor measure is proposed to apply a good reaction after disruption and robust optimization is applied to produce a robust schedule in the first-stage. Kacema et al. [20] examined a single machine weighted completion time problem with a fixed non-availability interval. Zhiqiang et al. [20] considered robustness (measured by *RM1*) and stability (measured by *SM3*) simultaneously with machine breakdowns as the only source of uncertainty and Genetic Algorithm (GA) is applied to solve this dual criteria optimization problem. To the best of our knowledge, except in Rahmani [18], no other papers simultaneously consider the robustness and the stability measures, with uncertain processing times and random machine breakdowns. This paper proposes effective heuristics for the same problem, previously discussed in [18].

3. Problem Definition

There are different factors that lead to disruptions in systems such as arrival of a new job, due date uncertainty, breakdown occurrence, the uncertainty of job processing times, etc., which are commonly known as scheduling uncertainties. The current paper simultaneously considers the uncertain job processing times and the machine breakdowns as the system uncertainties. Table 1 summarizes the indices used in the model.

Table 1.

Also, the following assumptions are considered:

- Job j is available at the beginning of the scheduling.
- The machine has availability limitations; i.e. random breakdown may occur during the processing time of job j .
- The time between two consecutive breakdowns follows an exponential distribution. Also, a fixed repair time is allocated after each failure.
- The rest of the disrupted job will be performed once the machine is repaired.
- The objective function is the simultaneous minimization of the defined robustness and stability measures.

When the real value of uncertain parameters are not known in advance, surrogate measures are commonly used to obtain robust and stable schedules [3]. In this paper, we arbitrarily consider some combinations of robustness and stability measures to define the objective functions of the predefined problems. The objective functions of the first, second, and third problems are respectively represented as *Z1*, *Z2*, and *Z3*. In the next section, we analytically show the optimality of *SEPT* for the first and the second problems. For other combinations of *RM*s and *SM*s, such as $Z = \alpha.RM1 + (1-\alpha).SM1$ and $Z = \alpha.RM1 + (1-\alpha).SM2$, the optimality of *SEPT* can be shown easily.

In order to solve the third problem, general two-stage heuristics are proposed. These approaches can be adjusted to solve other combinations of *RM*s and *SM*s, such as $Z = \alpha.RM2 + (1-\alpha).SM1$, $Z = \alpha.RM2 + (1-\alpha).SM2$ and $Z = \alpha RM3 + (1-\alpha).SM3$. The first stage produces predictive schedule to optimize the robustness measure while assuming the job processing times as the only source of uncertainty. In the second stage, we keep this predictive job sequence, and job processing time modification is performed to hedge against the machine breakdown disruption. We show the effectiveness of the proposed method by comparing the results with the righting shift (RS) rescheduling method, which is the preferred policy in the case of machine breakdown disruption [10].

4. Solution Methods

In this section, based on proved theorems, we optimally obtain robust and stable schedules for the first and second problems. For the third problem, we propose two-stage predictive methods. In the first stage, we optimize robustness without considering machine breakdowns. In the second stage, we embedded additional times into job processing times to hedge against machine breakdowns.

4.1. The Analytical Approach of the First Two Problems

According to the classification defined by Graham et al. [21], a robust and stable single machine problem under uncertain job processing times and machine breakdowns (when the processing time of job j follows an exponential distribution with rate λ_j and the time between two consecutive breakdowns follows the exponential distribution with rate θ) can be represented by $1/p_j \sim \exp(\lambda_j); brkdwn: U \sim \exp(\theta), D \sim G(t)/Z_i$.

In the stochastic version of $1//\sum_{j=1}^{n}C_j$, when the job processing times follow an arbitrary distribution, the shortest expected processing time first rule (*SEPT*), which sorts the jobs in non-decreasing order of $E(p_j)$ gives the optimal sequence [22]. The optimality of *SEPT* is held even in the generalization of the single machine expected total completion times problem under machine breakdowns and job processing times variability, i.e. *SEPT* solves $1/p_j \sim \exp(\lambda_j); brkdwn: U \sim \exp(\theta), D \sim G(t) / E\left(\sum_{j=1}^{n}C_j\right)$ optimally and to take into account the machine unavailability impacts, the processing time of job j is modified via Equation 4 [22].

$$E(q_j) = E(p_j)\left(1 + r/\theta\right) \quad (4)$$

where $E(q_j)$ is the modified job processing time after breakdown. We prove that the optimality of *SEPT* is also held for *RM3* (See Appendix). In addition, *SEPT* solves $1/X_j \sim \exp(\lambda_j); brkdwn: U \sim \exp(\lambda), D \sim G_2(t) / SM1(SM2)$ optimally [2], (where $SM1$ is the sum of the squared differences, and $SM2$ is the sum of variances of the realized completion times). Based on the above, we can conclude that if $E[p_i] > E[p_j]$ implies that $\text{var}[p_i] \geq \text{var}[p_j]\ \forall(i,j)$, then the following corollaries are solved optimally by the SEPT rule:

Corollary1: $1|p_j \sim \exp(\lambda_j); brkdwn: U \sim \exp(\theta), D \sim G(t)|\alpha.RM3 + (1-\alpha).SM1$ is solved optimally according to *SEPT* (See Appendix for proof).

Corollary2: $1|p_j \sim \exp(\lambda_j); brkdwn: U \sim \exp(\theta), D \sim G(t)|\alpha.RM3 + (1-\alpha).SM2$ is solved optimally according to SEPT (See Appendix for proof).

4.2. The Proposed Heuristics

In this section, we propose two-stage heuristics to handle the following problem $1|p_j \sim \exp(\lambda_j); brkdwn: U \sim \exp(\theta), D \sim G(t)|\alpha.RM2 + (1-\alpha).SM3$. That is we propose heuristics to the robust and stable single machine problem under the uncertainty of job processing times and machine breakdowns when the processing time of job j follows the exponential distribution with rate λ_j, and the time between two consecutive breakdowns follows the exponential distribution with rate θ, and the robustness and stability measures are respectively the expected total (realized) tardiness, and the sum of absolute differences of the realized completion times. The expected total tardiness is taken as a primary objective of this problem. The problem $1//\sum_j T_j$ is known to be *NP*-hard even if deterministic job processing times are considered and no machine breakdowns occur [5]. Assuming Erlang distribution for job processing times, Bożejko et al. proposed the Tabu search algorithm to handle the single machine stable total weighted tardiness problem [23].

Goren and Sabuncuoglu [2] analytically proved the optimality of *SEPT* for single machine expected total tardiness problem when the job processing times follow the exponential distribution with rate λ_j .

Corollary3: SEPT gives the optimal sequence for $1|\, p_j \sim \exp(\lambda_j); d_j = d \,|RM\, 2$.

To handle the third problem, heuristic methods are proposed based on corollary 3 and the idea of a predictive two-stage approach called optimized surrogate measure heuristic (*OSMH). OSMH* is proposed to minimize the maximum lateness in the job shop environment with random machine breakdowns [7]. In *OSMH*, a predictive schedule is generated to minimize the primary objective assuming no breakdowns, then the same job sequence is kept and the idle time is inserted into the schedule to minimize the difference between the real and the planned completion times (stability) without considering the effects on the primary objective. Donovan [6] modified *OSMH* to minimize the total tardiness in a single machine scheduling under uncertainty of random machine breakdowns; *ATC* (a priority rule to produce a feasible schedule in a single machine total tardiness problem) is applied to generate a predictive initial schedule in the first stage. A modified two-stage GA based on the idea of (OSMH) inserting ideal times, was proposed to obtain robust and stable schedule in single machine problem under machine breakdown disruption [1]. We propose two-stage predictive heuristics to solve the third problem. In the first stage, the initial robust schedule is generated without considering breakdowns. In the second stage, idle time are inserted to enhance the schedule stability. Different methods are proposed to generate the adequate idle times. The details of the proposed predictive heuristics are presented below.

4.2.1. Predictive *SEPT-OSMH*

First stage- Robustness optimization: Generate the initial robust schedule according to SEPT (without considering machine breakdown, to minimize robustness measure ($E\left(\sum_{j=1}^{n} T_j\right)$)).

Second stage- Stability enhancement: Take into account the machine unavailability impacts by modifying the job processing times according to Equation 4.

Additional times (the total expected required repair times during the processing of a job) are obtained from Equation 5, where r is equal to the required expected repair time. The amount of mean time between failures (*MTBF*) is calculated from failure function distribution. According to the Equation 6, there is no setup time before the first job. Equation 7 gives the expected completion time of the first job. The completion time of the first job is acquired from the sum of the expected processing time and the additional time. The completion time of job *j* is obtained from Equation 8.

$$ADT_j = r.\mathrm{E}(p_j) / MTBF \tag{5}$$

$$EC_0 = 0 \tag{6}$$

$$EC_1 = E\left(p_1\right) + ADT_1 = E\left(p_1\right) * (1 + r\theta) \quad (7)$$

$$EC_j = EC_{j-1} + E\left(p_j\right)(1 + r\theta) \quad (8)$$

4.2.2. Linear Programming Based Heuristics

While more additional time insertion, enhances the solution robustness, it degrades the quality robustness. To control the expected degradation of quality robustness, linear programming based methods are provided.

4.2.2.1. Predictive *SEPT-LPOSMH*

In this method, the amount of the additional time is constrained by the difference between the initial and final primary objective to control the realized schedule primary objective degradation. The procedure of the LP-based heuristic is presented below.

Step 1. Robustness optimization; Generate the initial robust schedule according to *SEPT* (to minimize the robustness measure $E\left(\sum_{j=1}^{n} T_j\right)$ without considering machine breakdown).

Step 2. Calculate additional time for all jobs from the following LP model where $E(C_j)$, $E\left(C_j^{LP}\right)$ and $E\left(C_j^{P}\right)$ denote the completion time of j^{th} job in the sequence obtained by *SEPT*, *LP* model and predictive *SEPT-OSMH*, respectively. The objective function (Relation 9) calculates total expected tardiness. Constraints (10) and (11) guarantee the upper bound of the precedence relationships. Constraint (12) controls the degradation in the completion time of the realized schedule. We define $0 \le \eta \le 1$ as the control parameter. Degradation in the expected total tardiness of the *LP*-based model is controlled by constraint (13).

$$\min\left\{E\left[\sum_{j=1}^{n} \max(0,\ C_j^{LP} - d)\right]\right\} \quad (9)$$

$$s.t.$$

$$E\left(C_j^{LP}\right) \ge E\left(p_j\right) \quad (10)$$

$$E\left(C_j^{LP}\right) - E\left(C_{j-1}^{LP}\right) \ge E\left(p_{j-1}\right),\ \mathrm{j} = 1, 2, ..., n \quad (11)$$

$$E\left(C_j^{LP}\right) \le E\left(C_j^{P}\right),\ \mathrm{j} = 1, 2, ..., n \quad (12)$$

$$E\left(\max(0,\ C_j^{LP} - d)\right) \le E\left[\sum_{j=1}^{n} \max(0,\ C_j - d)\right] + ... \quad (13)$$

$$... + \ \eta\left\{E\left[\sum_{j=1}^{n} \max(0, C_j^{P}\ \text{-} d)\right] \text{-} E\left[\sum_{j=1}^{n} \max(0,\ C_j\ \text{-} d)\right]\right\}\ j = 1, 2, ..., n$$

In the next section, we show that except in the case of low machine breakdown rate and duration, the robustness and the stability of the schedule generated by *LP*-based method improved significantly over those generated by the predictive *SEPT-OSMH* method.

5. Computational Results

To examine the performance of the proposed predictive schedules for the third problem, a series of computational experiments using randomly generated test problems are conducted. The test instances were generated as in [7]. These algorithms are coded in MATLAB R2013b and executed on an Intel Core PC with 3.0 GHz CPU and 8.0 GB RAM.

5.1. The comparison between *SEPT-OSMH* and *SEPT-LPOSMH*

There are five categories for the number of jobs as $n = 10, 30, 50, 70, 90$. The processing times follow different exponential distributions, with uniformly-distributed, random rates of λ_j. Therefore, we have a total of 5 problems with different parameter combinations. For each combination, 100 instances are generated, increasing the number of tests to the total of 500 (see Table 2).

Table 2.

Inspired by Mehta and Uzsoy [8], a common due date is considered, which is equal to the five times of the maximum expected processing time of jobs.

The time between two consecutive machine breakdowns is exponentially distributed with mean $\theta E\left[p_j\right] = \theta\lambda_j$ where $E\left(p_j\right)$ is the expected job processing time, and $\theta = 10, 5, 2$. The machine breakdown durations or repair times are generated from a uniform distribution ($r \in \left[\beta_1 E\left[p_j\right], \beta_2 E\left[p_j\right]\right] = \left[\beta_1\lambda_j, \beta_2\lambda_j\right]$).

Therefore, the unit considered for the job processing times (minute, hour, day, or ...) is the same unit considered for the common due date, the time between two consecutive machine breakdowns, and the machine breakdown durations.

The steady state availability of repairable systems is obtained by $A = MTBF/\left(MTBF + MTTR\right) = \theta/\left(\theta + \mu\right)$ [24], so the machine availabilities for B_1, B_2, B_3, B_4, B_5 and B_6 are 97.1%, 94.3%, 87%, 87%, 76.9% and 57%, respectively, calculated via the binomial approximation (see Table 3).

Table 3.

Therefore, we have 500 instances that are subject to 6 types of machine breakdowns and a total of 3000 combinations of the problem and breakdown types.

The problem type is denoted by $\left(B_j, n\right)$, where we represent the breakdown type by B_j, the number of jobs by n and the sign $*$ represents all of the possible values of the parameter.

AET_{SEPT} and AEC_{SEPT} represent the average expected realized schedule tardiness and the average realized completion time for problem Q using *SEPT*. Similarly, $AET_{SEPT-OSMH}$ and

$AEC_{SEPT-OSMH}$ represent the average expected realized schedule tardiness and the average realized completion time for problem *Q* using *SEPT-OSMH*. The notation $AETI$ represents the average expected realized schedule tardiness improvement for problem Q using *SEPT-OSMH* method to *SEPT*, and $AECI$ represents the average expected completion time improvement for problem *Q* using *SEPT-OSMH* method to *SEPT*.

$$AETI = \frac{\sum_{Q} AET_{SEPT} - \sum_{Q} AET_{SEPT-OSMH}}{\sum_{Q} AET_{SEPT}} \quad (14)$$

$$AECI = \frac{\sum_{Q} AEC_{SEPT} - \sum_{Q} AEC_{SEPT-OSMH}}{\sum_{Q} AEC_{SEPT}} \quad (15)$$

Table 4 presents the values of *AEC, AET, AECI* and *AETI* for various problem classes. The bold positive values in Table 4 indicate that the performance of *SEPT-OSMH* is better than *SEPT*. It should be noted that *SEPT* is considered as one of the most commonly used reaction methods in scheduling under uncertainty. The closer the values to one, the more impressive the performance improvement of *SEPT-OSMH* to *SEPT*. According to Table 4, we can draw the following conclusion.

When the type of machine breakdowns are *B1, B2, B3,* and *B4,* the objective degradation of the predictive scheduling generated by the *SEPT-OSMH* algorithm improves significantly compared to *SEPT* (Figure 1).

Figure 1.

This conclusion is logical since the low (or the moderate) length and the frequency of the machine breakdown have not created much disturbances to the initial schedule. In such cases, the predictive methods are more appropriate. Moreover, the usage of reactive scheduling methods such as *SEPT* in scheduling the systems with a high degree of uncertainty is recommended [3]. This point is also confirmed here; i.e., whenever the type of machine breakdowns is *B6*, the objective degradation of the schedule generated by *SEPT* improves significantly compared to *SEPT-OSMH* algorithm.

Also a greater the number of jobs shows a lower objective degradation of the predictive schedule from the *SEPT-OSMH* compared to *SEPT* (Figure 2).

Figure 2.

That is, when the number of jobs increases, the effect of predictive scheduling is more evident.

Table 4.

To compare the effectiveness of *SEPT-OSMH* and *SEPT-LPOSMH*, Equation 16 and Equation 17 are defined. $AETI$ represents the average expected tardiness (robustness) improvement and $AEADCI$ indicates the average expected absolute differences completion time (stability) improvement for problem *Q* using the proposed *SEPT-LPOSMH* heuristic to *SEPT-OSMH*.

$$AETI = \frac{\sum_{Q \in \delta} AET_{SEPT-OSMH} - \sum_{Q \in \delta} AET_{SEPT-LPOSMH}}{\sum_{Q \in \delta} AET_{SEPT-OSMH}} \quad (16)$$

$$AEADCI = \frac{\sum_{Q \in \delta} AEC_{SEPT-OOSMH} - \sum_{Q \in \delta} AEC_{SEPT-LPOSMH}}{\sum_{Q \in \delta} AEC_{SEPT-OSMH}} \quad (17)$$

From the overview of Table 5, we can conclude that the *LP*-based method is more effective than *SEPT-OSMH* especially for a small η, and that 0.1 is the most appropriate value for η.

Also, the scheduling generated by *SEPT-OSMH* is more robust than *SEPT-LPOSMH* only when the machine breakdown frequency and duration are small (*B1*). In other cases, i.e. when the type of machine breakdowns are *B2*, *B3*, *B4*, *B5*, and *B6*, the robustness and stability of the schedules generated by *LP*-based algorithm improve significantly over those generated by *SEPT-OSMH*.

The reason for this can be that with increasing frequency and duration of machine breakdown, scheduling disturbance increases, so the *LP*-based algorithm which generates more stable (controlled) schedule, shows much better performance than *SEPT-OSMH*.

Table 5.

If a schedule with maximum stability improvement is desired, then 0.1 is the advisable value for η (see Figure 3). For $\eta = 0.1$, the robustness and stability improvement of *SEPT-LPOSMH* in comparison to *SEPT-OSMH* is higher when the number of jobs is 70.

If a schedule with maximum robustness improvement is desired, then 0.8 is the advisable value for η (see Figure 4).

Figure 3.

Figure 4.

There is a logical contradiction between stability and robustness since to enhance the schedule robustness, sequence manipulation may be necessary, which leads to stability degradation [18]. Figure 5 confirms this conflict.

According to Figure 5, If a robust *and* stable schedule is required, then the appropriate amount of η depends on the number of jobs. For example, when the number of jobs is 70, then 0.1 is the advisable value for η, and when the number of jobs is 50, then 0.3 is the advisable value for η, and so on.

Figure 5.

The increase in the η means that the Equation 13 is less restricted. That is, in order to simultaneously enhance the robustness and the stability, the robustness should worsen in favor of upgrading the stability.

6. Conclusions

The generation of high robust and stable schedule in stochastic single machine environments has become the focus of many researches recently, but only a few studies consider robustness and stability, simultaneously. Even fewer studies consider both the machine breakdown and the variable processing time as the sources of uncertainty. No exact/optimum solution for these problems has been proposed in the literature. In this paper, bi-objective problems of robustness and stability optimizations in stochastic, single-machine environments were considered and solved optimally by an analytical approach. Also, predictive heuristics were proposed to solve intractable problems of finding robust and stable solution with *RM2* (the expected total realized tardiness) as the robustness measure. Based on extensive computational experiments over 3000 combinations of problems and breakdown characteristics, in the case of the large number of jobs and a small/medium machine breakdown duration, *SEPT-OSMH* performs significantly better than *SEPT*. Additionally, scheduling generated by predictive *SEPT-OSMH* is only preferred to *SEPT-LPOSMH* when the machine breakdown frequency and duration are low. In other words, the *LP*-based method has higher prediction and the disturbance in the scheduling generated by this method is significantly lower.

The general predictive approach in the paper can be extended to any other complex machine environments such as job shop or open shop systems to achieve robust and stable schedules. Additionally researchers can present other measures of robustness and stability as predictive-reactive methods in more disrupted systems.

7. Appendix

Proof of corollary1: The proof is by contradiction. Suppose that p_j is the processing time of job *j*, θ is the rate of machine breakdowns, r is the average time of repair and q_j is the total remaining time of the job *j* on a machine. We have $E[q_j] = (1+\theta r)E[p_j]$ [22].

Let *S* be an optimal sequence, assume that there exists a pair of adjacent jobs ***i*** and ***j*** such that $E[p_j] > E[p_i]$ and job ***j*** succeeds job *i* in *S*. Consider a sequence S' from S *by* swapping the positions of jobs *i* and *j* .we show that S' is better than *S*, i.e. $[\alpha.RM(S) + (1-\alpha).SM(S)] - [\alpha.RM(S') + (1-\alpha).SM(S')] > 0$ which contradicts with the optimality of *S*;

$$[\alpha.RM(S) + (1-\alpha).SM(S)] - [\alpha.RM(S') + (1-\alpha).SM(S')] > 0$$
$$or \quad \alpha.[RM(S) - RM(S')] + (1-\alpha).[SM(S) - SM(S')] > 0$$

It is sufficient to show: $[SM(S) - SM(S')] > 0 \;\; and \; [RM(S) - RM(S')] > 0$

The proof of $\left[RM(S) - RM(S')\right] > 0$ **:** We ignore the contribution of jobs other than *i* and *j* in the comparison of S' and S, since nothing changes for them and suppose it is a constant as c. suppose the index set of jobs that precedes job *i* in *S* denoted by BS_i. We have:

$$RM(S) = E(\sum_{m \in BS_i} q_m + q_i) + E(\sum_{m \in BS_i} q_m + q_i + q_j) + c - [E(\sum_{m \in BS_i} p_m + p_i) + E(\sum_{m \in BS_i} p_m + p_i + p_j) + c']$$

$$RM(S') = E(\sum_{m \in BS_i} q_m + q_j) + E(\sum_{m \in BS_i} q_m + q_j + q_i) + c - [E(\sum_{m \in BS_i} p_m + p_j) + E(\sum_{m \in BS_i} p_m + p_j + p_i) + c']$$

$$\rightarrow \left[RM(S) - RM(S')\right] = E(q_i) - E(q_j) - [E(p_i) - E(p_j)] =$$

$$= (1+\theta r).E(p_i) - (1+\theta r).E(p_j) - [E(p_i) - E(p_j)] = \theta r,[E(p_i) - E(p_j)] > 0 \therefore$$

This contradicts with the optimality of *S*. The proof of $\left[SM1(S) - SM1(S')\right] > 0$ is discussed in Goren and Sabuncuoglu [2].

References

[1] Liu, L., Gu, H. and Xi, Y., "Robust and stable scheduling of a single machine with random machine breakdowns," *Int J Adv Manuf Technol,* **31**, pp. 645–654 (2007).

[2] Goren, S., and Sabuncuoglu, I., "Optimization of schedule robustness and stability under random machine breakdowns and processing time variability," *IIE Transactions,* **42** (3), pp. 203-220 (2009).

[3] Rahmani, D. and Heydari, M., "Robust and stable flow shop scheduling with unexpected arrivals of new jobs anduncertain processing times," *Journal of Manufacturing Systems,* **33** (1), pp. 84-92 (2014).

[4] Zhiqiang Lu, Weiwei Cui, and Xiaole Han, "Integrated production and preventive maintenance scheduling for a single machine with failure uncertainty," *Computers & Industrial Engineering,* **80**, pp. 236-244 (2015).

[5] Briskorn, D., Leung, J. and Pinedo, M-L., "Robust scheduling on a single machine using time buffers," *IIE Transactions,* 43 (6), pp. 383-398 (2011).

[6] O'Donovan, R., Uzsoy, R. and McKay, K-N., "Predictable scheduling of a single machine with breakdowns and sensitive jobs," *International Journal of Production Research*, **37** (18), pp. 4217–4233 (1999).

[7] Mehta, S-V. and Uzsoy, R-M., "Predictable scheduling of a job shop subject to breakdown," *IEEE Transactions on Robotics and Automation,* **14** (3), pp. 365–378 (1998).

[8] Mehta, S-V., "Predictable scheduling of a single machine subject to breakdowns," *International Journal of Computer Integrated Manufacturing,* **12** (1), pp. 15-38 (1999).

[9] Yang, J. and Yu, G., "On the Robust Single Machine Scheduling Problem," *Journal of Combinatorial Optimization,* **6** (1), pp. 17-33 (2002).

[10] Gören, S., "Robustness and Stability measure for scheduling policies in a single machine environment", MS Dissertation, *Bilkent University*, *Turkey* (2002).

[11] Adiri, I., Bruno, J., Frostig, E. et al., "Single machine flow-time scheduling with a single breakdown," **26** (7), pp. 679 (1989).

[12] Ganji, F., Moslehi, G., "Minimizing maximum earliness in single-machine scheduling with flexible maintenance time," *Scientia Iranica,* **24** (4), pp. 2082-2094 (2017).

[13] Wu, S-D., Storer, R-H. and Pei-Chann, C., "One-machine rescheduling heuristics with efficiency and stability as criteria," *Computers & Operations Research,* **20** (1), pp. 1-14 (1993).

[14] Jafari, A. and Lotfi, M., "Single machine scheduling to minimize the maximum tardiness under piecewise linear deteriorating jobs," *Scientia Iranica*, **25**(1), pp. 370-385 (2018).

[15] Mousavi, S-M., Mahdavi, I., Rezaeian, J. et al., "Bi-objective scheduling for re-entrant hybrid flow shop with learning effect and setup times," *Scientia Iranica*, **25**(4), pp. 2233-2253 (2017).

[16] Rudek, R., "The single machine total weighted completion time scheduling problem with the sum-of-processing time based models: Strongly NP-hard," *Applied Mathematical Modelling,* **50**, pp. 314–332 (2017).

[17] N. Hamta, N., Fatemi Ghomi, S-M-T., Tavakkoli-Moghaddam, et al., "A hybrid meta-heuristic for balancing and scheduling assembly lines with sequence-independent setup times by considering deterioration tasks and learning e," *Scientia Iranica,* **21** (3), pp. 963-979 (2014).

[18] Rahmani, D., "A new proactive-reactive approach to hedge against uncertain processing times and unexpected machine failures in the two-machine flow shop scheduling problems," *scientia Iranica,* **24** (3), pp. 1571-1584 (2017).

[19] Grigoriu, L. and Friesen, D. K., "Scheduling on uniform processors with at most one downtime on each machine," *Discrete Optimization,* **17**, pp. 14-24 (2015).

[20] Kacem, I. and Paschos, V-T., "Weighted completion time minimization on a single-machine with a fixed non-availability interval: Differential approximability," *Discrete Optimization,* **10** (1), pp. 61-68 (2013).

[21] Graham, R-L., Lawler, E-L., Lenstra, J-K. et al., "Optimization and approximation in deterministic sequencing and schedulingA survey," *Annals of Discrete Mathematics*, vol. 5, p. 287–326, 1979.

[22] Pinedo, M-L., Scheduling: theory, algorithms, and systems, Fifth Edn., Springer, USA (2016).

[23] Bożejko, W., Rajba, P. and Wodecki, M., "Stable scheduling of single machine with probabilistic parameters," *Bulletin of the Polish Academy of Sciences Technical Sciences,* **65** (2), pp. 219-231 (2017).

[24] Elsayed, E-A., "Reliability Engineering", Prentice Hall, USA (1996).

Rashed Sahraeian is an Associate Professor in the Department of Industrial Engineering at Shahed University. He holds his BS degree in Industrial Engineering from Amirkabir University of Technology (Tehran Polytechnic) and MS and PhD degrees in Industrial Engineering from Tarbiat Modares University. His research interests is optimization in scheduling, facility location problem and supply chain management. He is author and co-author of more than 100 papers, including book chapters, conferences and refereed journals.

Donya Rahmani received the PhD in industrial Engineering from the University of Science &Technlogy at Tehran in 2009. She worked as an assistant professor of Industrial Engineering of K.N. Toosi University of Technology. As a researcher in academics, she has authored over 25 technical papers including book chapters, conference papers, and refereed journal articles. Her research interests include robust optimization, stochastic scheduling, queuing theory, and advanced linear programming.

Zeinab Abtahi is the PhD candidate at Shahed University. She had graduated from Sharif University of technology and received Master's degree from Shahed University. Her research interest includes stochastic scheduling, robust optimization, and integer programming.

Figure 1.The superiority of the predictive scheduling generated by *SEPT-OSMH* compared to *SEPT* for different breakdown type.

Figure 2.The superiority of the predictive schedule from *SEPT-OSMH* compared to *SEPT* for different number of jobs.

Figure 3. Stability improvement for different breakdown type.

Figure 4. Robustness improvement for different breakdown type

Figure 5. The robustness and stability conflict

Table 1. Indices used in the model

Table2. Problem parameters

Table3. Type of machine breakdown

Table 4. *AEC, AET, AECI* and *AETI* values for various problem classes

Table 5. Stability and robustness improvement of *SEPT-LPOSMH* compared to *SEPT-OSMH*

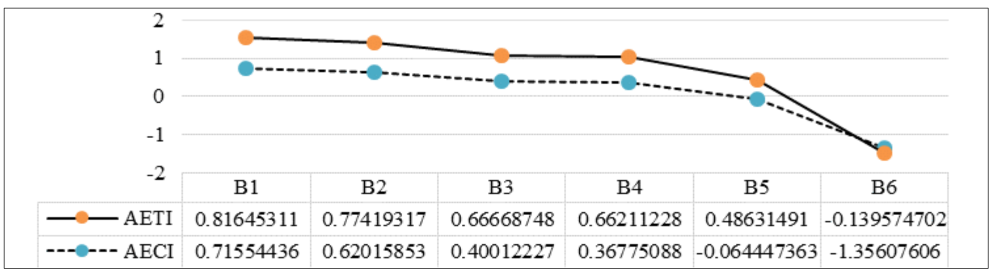
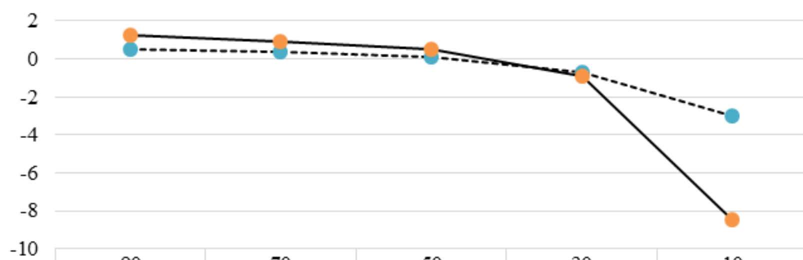
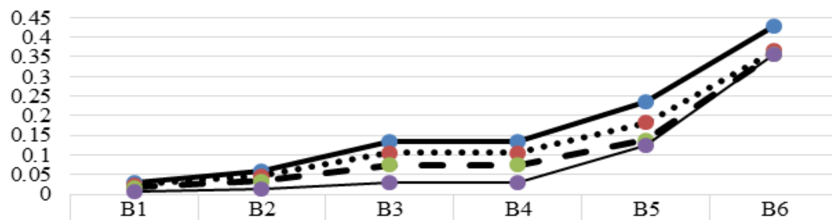


Figure 1. The superiority of the predictive scheduling generated by *SEPT-OSMH* compared to *SEPT* for different breakdown type.



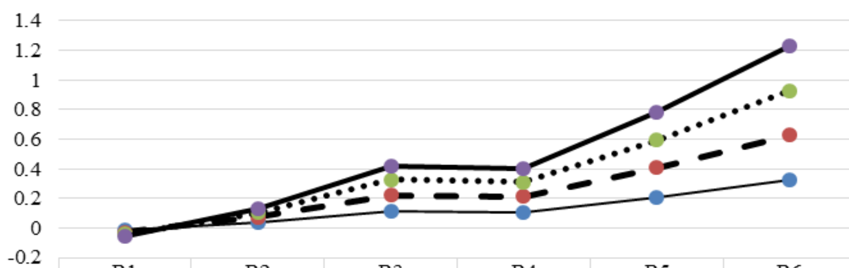
	90	70	50	30	10
AETI	0.69485	0.57138	0.37358	-0.267866	-5.41781
AECI	0.52633	0.3412	0.11592	-0.672258	-3.020424

Figure 2. The superiority of the predictive schedule from *SEPT-OSMH* compared to *SEPT* for different number of jobs.



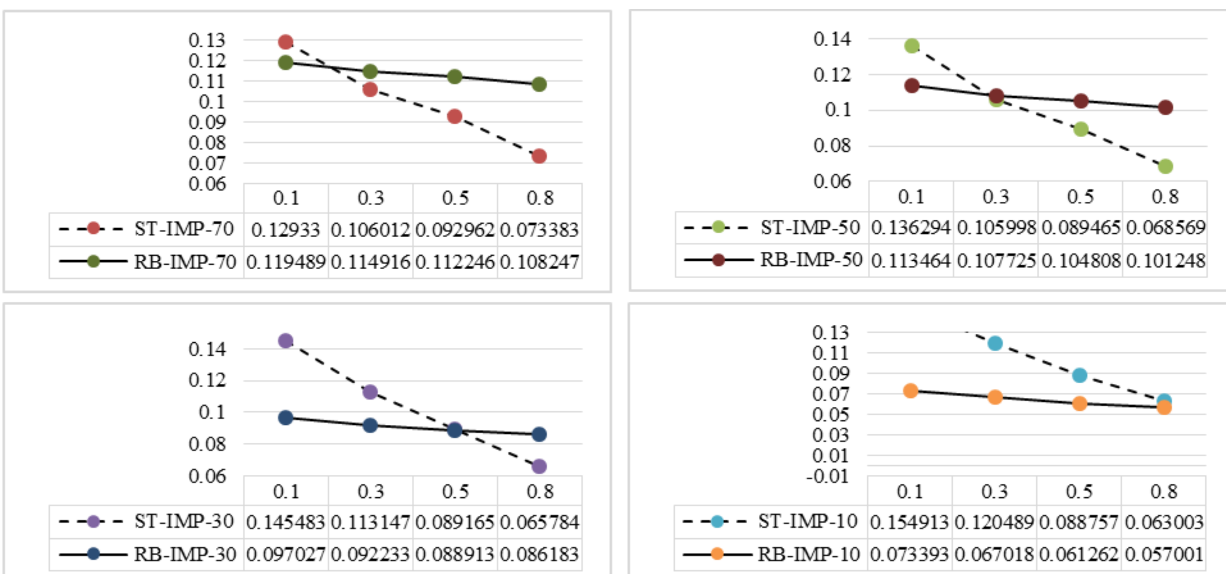
	B1	B2	B3	B4	B5	B6
STABILITY-0.1	0.030731	0.05895	0.135014	0.134817	0.233731	0.427242
STABILITY-0.3	0.023905	0.045856	0.105005	0.104857	0.181775	0.365048
STABILITY-0.5	0.017072	0.032767	0.075005	0.074897	0.13883	0.356749
STABILITY-0.8	0.006829	0.013131	0.029974	0.029965	0.124121	0.356044

Figure 3. Stability improvement for different breakdown type.



	B1	B2	B3	B4	B5	B6
ROBUSTNESS-0.8	-0.01602	0.027537	0.09533	0.08917	0.183684	0.299395
ROBUSTNESS-0.5	-0.01458	0.031745	0.103582	0.097954	0.18893	0.300391
ROBUSTNESS-0.3	-0.01361	0.034579	0.109082	0.103811	0.198132	0.305
ROBUSTNESS-0.1	-0.01265	0.037467	0.114581	0.109673	0.208608	0.321501

Figure 4. Robustness improvement for different breakdown type



ST-IMP-70 means the Stability Improvement when the number of jobs is 70
RB-IMP-70 means the Robustness Improvement when the number of jobs is 70

Figure 5. The robustness and stability conflict

Table 1. Indices used in the model

D	Downtimes; the time required to return the machine to its operational mood (following the general distribution; $D \sim G(t)$)
U	Uptimes; The time between two consecutive machine breakdowns (follow from exponential distribution with rate θ)
j	Job index, $j = 1,2,\ldots,n$
d_j	Due date of job *j*
r	(*tr*) The expected value of repair times after each breakdown
$E(p_j)$	The expected processing time of job *j*
λ_j	The job processing times following an exponential distribution with rate λ_j in the first three problems
C_j	The (expected) initial completion time of job *j*
C_j^{RS}	The (expected) completion times of job *j* assuming righting shift policy
C_j^{P}	The (expected) proposed predictive method's completion time of job *j*
C_j^{LP}	The (expected) linearized predictive completion time of job *j*
C_j^{r}	The realized (expected) completion times of job *j*

Table 2. Problem parameters

parameter	Value	Number of value
Number of jobs	$n = 10, 30, 50, 70, 90$	5
Processing times	$\lambda_1, \lambda_2, \ldots\lambda_i \ldots\lambda_n$	
	$\lambda_j \in Uniform\left[0.1, 1\right]$	
	Problem combination	100
	Total problems	500

Table 3. Type of machine breakdown

Type of machine breakdown B_i	The mean time between breakdowns $\theta E\left[p_j\right]$	Breakdown durations $uniform\left[\beta_1 E\left[p_j\right], \beta_2 E\left[p_j\right]\right]$	Machine availability (%) $A = \theta/(\theta+\mu)$
B_1	10	$(\beta_1, \beta_2) = (0.1, 0.5)$	0.97
B_2	5	$(\beta_1, \beta_2) = (0.1, 0.5)$	0.94
B_3	2	$(\beta_1, \beta_2) = (0.1, 0.5)$	0.869
B_4	10	$(\beta_1, \beta_2) = (1, 2)$	0.869
B_5	5	$(\beta_1, \beta_2) = (1, 2)$	0.769
B_6	2	$(\beta_1, \beta_2) = (1, 2)$	0.57

Table 4. *AEC*, *AET*, *AECI* and *AETI* values for various problem classes*

		SEPT		*SEPT-OSMH*		*SEPT-OSMH to SEPT*	
		AEC	***AET***	***AEC***	***AET***	***AECI***	***AETI***
Breakdown type	$(B_1,*)$	307.2872	3932.187	87.40957315	721.74069	**0.71554436**	**0.8164531**
	$(B_2,*)$	301.3732	3864.157	114.4740443	872.553014	**0.62015853**	**0.7741932**
	$(B_3,*)$	302.6828	3946.514	181.5726484	1315.42243	**0.40012227**	**0.6666875**
	$(B_4,*)$	301.6363	3836.459	190.7093178	1296.29248	**0.36775088**	**0.6621123**
	$(B_5,*)$	293.3526	3808.482	312.2583845	1956.36057	-0.06444736	**0.4863149**
	$(B_6,*)$	302.7338	3859.856	713.2638034	4398.59409	-1.3560761	-0.139575
Number of jobs	$(*,90)$	3239.797	58392.16	1534.58	17818.420	**0.5263300**	**0.694850**
	$(*,70)$	2581.589	35150.6	1700.756	15066.220	**0.3412000**	**0.571380**
	$(*,50)$	1781.805	16697.13	1575.251	10459.370	**0.1159200**	**0.373580**
	$(*,30)$	1111.572	5638.05	1858.836	7148.2950	-0.672258	-0.267870
	$(*,10)$	330.566	360.328	1329.015	2312.5160	-3.020424	-5.417810

* The bold values show the superiority of *SEPT-OSMH* to *SEPT*.

Table 5. Stability and robustness improvement of *SEPT-LPOSMH* compared to *SEPT-OSMH*

		$\eta = 0.1$		$\eta = 0.3$		$\eta = 0.5$		$\eta = 0.8$	
		RI	SI	RI	SI	RI	SI	RI	SI
Breakdown type	$(B_1, *)$	-0.01265	0.030731	-0.01361	0.023905	-0.01458	0.017072	-0.01602	0.006829
	$(B_2, *)$	0.037467	0.05895	0.034579	0.045856	0.031745	0.032767	0.027537	0.013131
	$(B_3, *)$	0.114581	0.135014	0.109082	0.105005	0.103582	0.075005	0.09533	0.029974
	$(B_4, *)$	0.109673	0.134817	0.103811	0.104857	0.097954	0.074897	0.08917	0.029965
	$(B_5, *)$	0.208608	0.233731	0.198132	0.181775	0.18893	0.13883	0.183684	0.124121
	$(B_6, *)$	0.321501	0.427242	0.305	0.365048	0.300391	0.356749	0.299395	0.356044
Number of jobs	$(*, 90)$	0.047857	0.128115	0.042456	0.107778	0.039962	0.094805	0.036212	0.075344
	$(*, 70)$	0.119489	0.12933	0.114916	0.106012	0.112246	0.092962	0.108247	0.073383
	$(*, 50)$	0.113464	0.136294	0.107725	0.105998	0.104808	0.089465	0.101248	0.068569
	$(*, 30)$	0.097027	0.145483	0.092233	0.113147	0.088913	0.089165	0.086183	0.065784
	$(*, 10)$	0.073393	0.154913	0.067018	0.120489	0.061262	0.088757	0.057001	0.063003

RI: Robustness Improvement *(AETI)*
SI: Stability Improvement *(AEADCI)*