Scientia Iranica E (2020) 27(5), 2592–2603

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
http://scientiairanica.sharif.edu

Predictive heuristics for generating robust and stable schedules in single-machine systems under disruption

Z. Abtahi[a], R. Sahraeian[a,*], and D. Rahmani[b]

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, Tehran, Iran.*

Received 27 April 2018; received in revised form 9 January 2019; accepted 23 February 2019

KEYWORDS
Predictive heuristic;
Uncertain processing times;
Disruption;
Stable-robust scheduling;
Single machine.

Abstract. The present paper examines the problems of single-machine scheduling under disruption with uncertain processing times. The goal is to achieve schedules that are simultaneously stable and robust. To resolve 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 only by taking into account uncertain job processing times and forgoing 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.

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 uncertainties in machine availability, reactive and predictive scheduling methods have attracted the most attention due to their potential applicability (e.g., [1–4]). As the name suggests, in reactive scheduling, the system attempts to find the highest level of performance following disruptions. In other words, in reactive methods, the initial schedule is revised to suit the new changes. To ensure better performance, the differences between the initial schedule and the reactive schedule are minimized as part of applying the whole rescheduling process. 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 future disruptions. In doing so, the final goal remains the same regardless of any possible disruption.

While defining goals in uncertain environments and common objectives such as makespan, flow time, total tardiness, etc., two other measures should be considered: robustness (quality robustness) and stability (solution robustness) of the schedules. Despite different definitions of robustness in the literature, 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 unforeseen disruptions is addressed

*. *Corresponding author. Tel.: +98 21 51212040*
E-mail addresses: z.abtahi@shahed.ac.ir (Z. Abtahi); sahraeian@shahed.ac.ir (R. Sahraeian); drahmani@kntu.ac.ir (D. Rahmani)

doi: 10.24200/sci.2019.50809.1873

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].

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

where C_j^r is the realized (expected) completion time 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].

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

In general, practitioners concern themselves with the performance of the realized schedule rather than the planned or estimated performance of the initial schedule. Hence, optimizing the former may be more appropriate than 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) has become more and more important nowadays, especially for just-in-time 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 forming multiple-objective problems [1]. Therefore, in this paper, by using the linear combination of robustness and stability measures, the first three scheduling problems with disruptions are defined:

$$Z1 = \alpha.RM3 + (1-\alpha).SM1, \tag{1}$$

$$Z2 = \alpha.RM3 + (1-\alpha).SM2, \tag{2}$$

$$Z3 = \alpha.RM2 + (1-\alpha).SM3, \tag{3}$$

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

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 of single-machine scheduling with uncertain job processing times and 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 dynamic programming to solve this problem. Liu et al. [1] applied 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. This study proposed predictive-reactive heuristic methods to solve the third problem and demonstrated 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].

With a glimpse at the previous attempts in this realm of research, we can state contributions of this paper:

- Uncertain processing times and machine breakdowns are regarded as system disruptions;
- Stability and robustness are considered simultaneously in three stochastic single-machine scheduling problems;
- Two particular 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

Under uncertainty, job-related properties are considered to be random or the machine is subject 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 was concave, then the flow time could be stochastically minimized by the Shortest Processing Time (SPT) first rule. In the case of multiple breakdowns, it was proven that the SPT rule minimized the total flow time when the times to breakdown were exponentially distributed [11]. Ganji and Moslehi [12] focused on single-machine scheduling with a flexible unavailability constraint (with the unknown starting time of the unavailability period) to minimize 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 could be obtained with a high degree of schedule stability. They considered two criteria for stability: deviation of the revised schedule in terms of job starting times (similar to $SM3$) and 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 GA 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 maximum lateness as a performance measure and generated stable initial schedules by inserting idle times in schedules to optimize system performance. O'Donovan et al. [6] worked on generating stable schedules with machine breakdowns. They used total tardiness as the performance measure; the stability was measured based on the deviations of the absolute completion time from the initial schedule. 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, job processing times are assumed known and constant which may not be true in all conditions such as deteriorating jobs [14], cases with learning effect [15–17], and uncertainty in the job processing time duration [18]. Yang and Yu [9] proposed a robust approach based on some heuristics in cases of job processing time uncertainties to minimize the sum of the completion times. They showed that the robust version of the sum of the completion times was an NP-complete problem, even for very restricted cases. Goren and Sabuncuoglu [2] studied a single-machine problem where the performance measure was the total flow time and the source of uncertainty was 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 found that SEPT performed poorly in terms of robustness. Moreover, a novel algorithm is proposed to minimize the makespan under one machine breakdown at most 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 the two-machine flow shop scheduling problem. Multi-factor measure was proposed to apply a good reaction after disruption, and robust optimization was applied to produce a robust schedule in the first stage. Kacema and Paschos [20] examined a single-machine weighted completion time problem at a fixed non-availability interval. Zhiqiang et al. [4] considered robustness (measured by $RM1$) and stability (measured by $SM3$) simultaneously with machine breakdowns as the only source of uncertainty, and GA was applied to solve the dual-criteria optimization problem. To the best of our knowledge, no other papers, except [18], have simultaneously considered robustness and stability measures, with uncertain processing times and random machine breakdowns. This paper proposes effective heuristics to solve the same problem, as previously discussed in [18].

3. Problem definition

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

Moreover, the following assumptions are considered:

- Job J is available at the beginning of the scheduling;
- The machine has availability limitations: random

Table 1. Indices used in the model.

D	Downtimes: the time required for the machine to return to its operational mood (following the general distribution; $D \sim G(t)$)
U	Uptimes: The time between two consecutive machine breakdowns (following exponential distribution with rate θ)
j	Job index, $j = 1, 2, ..., n$
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

breakdown may occur during the processing time of job j;

- The time between two consecutive breakdowns follows an exponential distribution. Moreover, fixed repair time is allocated after each failure;
- The rest of the disrupted jobs 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 is not known in advance, surrogate measures are commonly used to obtain robust and stable schedules [3]. This study arbitrarily considers a combination 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 denoted by $Z1$, $Z2$, and $Z3$, respectively. The next section analytically shows the optimality of SEPT for the first and second problems. For other combinations of RMs and SMs such as $Z = \alpha.RM1 + (1-\alpha).SM1$ and $Z = \alpha.RM1 + (1-\alpha).SM2$, the optimality of SEPT is shown easily.

To solve the third problem, general two-stage heuristics are proposed. These approaches can be adjusted to solve other combinations of RMs and SMs 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 a predictive schedule to optimize the robustness measure while assuming the job processing times as the only source of uncertainty. In the second stage, this predictive job sequence is kept while job processing time modification is performed to hedge against the machine breakdown disruption. The effectiveness of the proposed method is demonstrated by comparing the results with the 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, robust and stable schedules for the first and second problems are optimally obtained. For the third problem, two-stage predictive methods are proposed. In the first stage, robustness is optimized regardless of the effect of machine breakdowns. In the second stage, additional times are embedded in 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 SEPT, first rule (SEPT), which sorts jobs in non-decreasing order of $E(p_j)$, gives the optimal sequence [22]. The optimality of SEPT still holds in the case of the generalized problem of single-machine expected total completion times under machine breakdowns and variability of job processing times, 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; to take into account the machine unavailability impacts, the processing time of job is modified via Eq. (4) [22].

$$E(q_j) = E(p_j)\,(1 + r/\theta)\,, \tag{4}$$

where $E(q_j)$ is the modified job processing time after breakdown. It is shown that the optimality of SEPT also holds 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, it can be concluded that if $E\,[p_i] > E\,[p_j]$ which 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:

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

Corollary 2: $1|p_j \sim \exp(\lambda_j); brkdwn : U \sim \exp(\theta)$, $D \sim G(t)|\alpha.RM3 + (1-\alpha).SM2$ is solved optimally according to SEPT (see Appendix for proof).

4.2. The proposed heuristics

In this section, two-stage heuristics are proposed to solve the following problem:

$$1\left|p_j \sim \exp(\lambda_j); brkdwn : U \sim \exp(\theta), D\right.$$
$$\left.\sim G(t)\right|\alpha.RM2 + (1-\alpha).SM3.$$

In other words, a heuristics approach to the robust and stable single-machine problem under the uncertainty of job processing times and machine breakdowns is proposed 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. In addition, robustness and stability measures are the expected total (realized) tardiness and the sum of absolute differences of the realized completion times, respectively. The expected total tardiness is taken as the 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]. Having assumed Erlang distribution for job processing times, Bożejko et al. [23] proposed the Tabu search algorithm to resolve the single-machine stable total weighted tardiness problem.

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

Corollary 3: SEPT gives the optimal sequence for $1|p_j \sim \exp(\lambda_j)\,; d_j = d|RM2$.

To solve 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 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 planned completion times (stability) regardless of the effects on the primary objective. O' Donovan et al. [6] modified OSMH to minimize total tardiness in single-machine scheduling under uncertain, 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 a robust and stable schedule in a single-machine problem under machine breakdown disruption [1]. Two-stage predictive heuristics are proposed to solve the third problem. In the first stage, the initial robust schedule is generated regardless of breakdowns. In the second stage, idle time is inserted to enhance the stability of a schedule. Different methods are proposed to generate adequate idle times. The details of the proposed predictive heuristics are presented below.

4.2.1. Predictive SEPT-OSMH

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

The second stage. *Stability enhancement:* consider the machine unavailability impacts by modifying the job processing times according to Eq. (4).

Additional times (the total expected required repair times during the processing of a job) are obtained through Eq. (5), where r is equal to the required expected repair time. The amount of mean time between failures is calculated from failure function distribution. According to Eq. (6), there is no setup time before the first job. Eq. (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 determined through Eq. (8):

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

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

$$EC_1 = E(p_1) + ADT_1 = E(p_1) * (1 + r\theta), \tag{7}$$

$$EC_j = EC_{j-1} + E(p_j)(1 + r\theta). \tag{8}$$

4.2.2. Linear programming based heuristics

While 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.

Predictive SEPT-LPOSMH

In this method, the amount of the additional time is constrained by the difference between the initial and final stages of the primary objective to control degradation in the realized schedule. 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 $E(\sum_{j=1}^{n} T_j)$ regardless of machine breakdown).

Step 2. Calculate the additional time of all jobs using 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 the jth 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. Herein, $0 \leq \eta \leq 1$ is defined as the control parameter. Degradation in the expected total tardiness of the LP-based model is controlled by Constraint (13).

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

s.t. :

$$E\left(C_j^{LP}\right) \geq E(p_j), \tag{10}$$

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

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

$$\begin{aligned} E\left(\max(0, C_j^{LP} - d)\right) \leq E\left[\sum_{j=1}^{n} \max(0, C_j - d)\right] + ... \\ ... + \eta \left\{ E\left[\sum_{j=1}^{n} \max(0, C_j^{P} - d)\right] \right. \\ \left. - E\left[\sum_{j=1}^{n} \max(0, C_j - d)\right] \right\}, \quad j = 1, 2, ..., n. \end{aligned} \tag{13}$$

In the next section, except for the case of low machine breakdown rate and duration, the robustness and stability of the schedule generated by the 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 in [7]. These algorithms are coded in MATLAB R2013b and executed by an Intel Core PC with 3.0 GHz CPU and 8.0 GB RAM.

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

The number of jobs is categorized 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, thus increasing the number of tests to the total of 500 (see Table 2).

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

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

Therefore, the unit considered for the job processing times (minute, hour, day, etc.) 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 [24]:

$$A = MTBF/(MTBF + MTTR) = \theta/(\theta + \mu),$$

Table 2. Problem parameters.

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

Table 3. Type of machine breakdown.

Type of machine breakdown B_i	The mean time between breakdowns $\theta E\,[p_j]$	Breakdown durations uniform $[\beta_1 E\,[p_j]\,, \beta_2 E\,[p_j]]$	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

therefore, 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).

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 (B_j, n), where B_j and n denote the breakdown type and the number of jobs, respectively, and the sign $*$ represents all 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, $AET_{\mathrm{SEPT-OSMH}}$ and $AEC_{\mathrm{SEPT\,-OSMH}}$ represent the average expected realized schedule tardiness and the average realized completion time for the problem Q using SEPT-OSMH. The notation represents the average expected realized schedule tardiness improvement for the problem Q using SEPT-OSMH method to SEPT, and represents the average expected completion time improvement for the problem Q using SEPT-OSMH method to SEPT.

$$AETI = \frac{\sum_Q AET_{\mathrm{SEPT}} - \sum_Q AET_{\mathrm{SEPT-OSMH}}}{\sum_Q AET_{\mathrm{SEPT}}}, \qquad (14)$$

$$AECI = \frac{\sum_Q AEC_{\mathrm{SEPT}} - \sum_Q AEC_{\mathrm{SEPT-OSMH}}}{\sum_Q AEC_{\mathrm{SEPT}}}. \qquad (15)$$

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 for scheduling under uncertainty. The closer the values to one, the more impressive the performance improvement of SEPT-OSMH to SEPT. According to Table 4, one can draw the following conclusion.

When the types 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).

This conclusion is logical since the small (or moderate) length and the frequency of the machine breakdown have not caused much disturbances to the initial schedule. In such cases, the predictive methods are more appropriate. Moreover, the application of reactive scheduling methods, such as SEPT, to scheduling the systems with a high degree of uncertainty is recommended [3]. To confirm the above, take the following as an example: whenever the type of machine breakdowns is $B6$, the objective degradation of the schedule generated by SEPT improves significantly compared to SEPT-OSMH algorithm.

Moreover, the larger number of jobs shows a lower objective degradation of the predictive schedule from the SEPT-OSMH compared to SEPT (Figure 2).

In other words, when the number of jobs increases, the effect of predictive scheduling is more evident.

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 over SEPT.

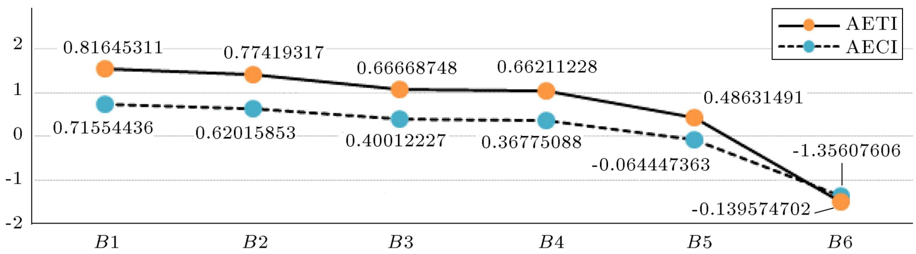

Figure 1. The superiority of the predictive scheduling generated by SEPT-OSMH over SEPT for different breakdown types.

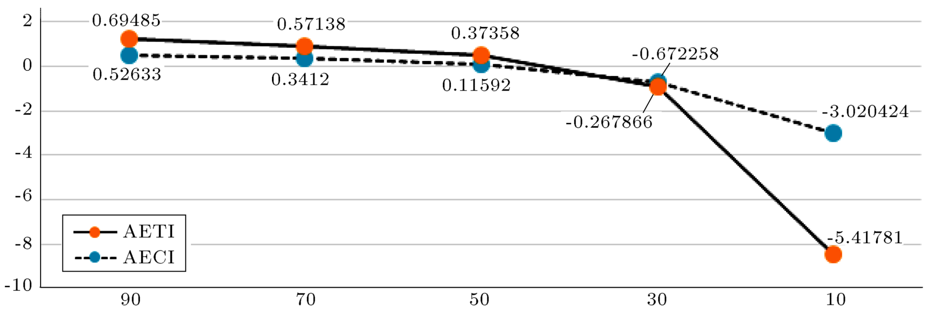

Figure 2. The superiority of the predictive schedule from SEPT-OSMH over SEPT for different number of jobs.

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

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

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

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

Moreover, the scheduling generated by SEPT-OSMH is more robust than SEPT-LPOSMH only when the machine breakdown frequency and duration

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).

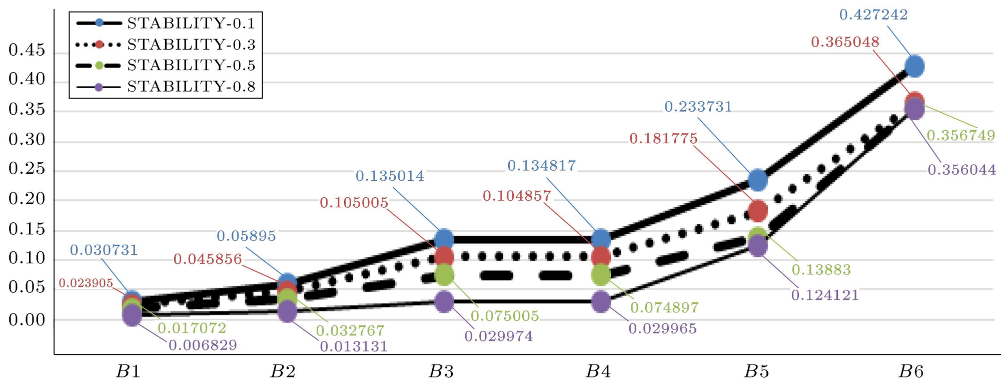

Figure 3. Stability improvement for different breakdown types.

are small ($B1$). In other cases, i.e., when the type of machine breakdowns is $B2$, $B3$, $B4$, $B5$, and $B6$, the robustness and stability of the schedules generated by LP-based algorithm improve significantly over those generated by SEPT-OSMH, because as the frequency and duration of machine breakdown increase, scheduling disturbance increases; therefore, the LP-based algorithm generating a more stable (controlled) schedule shows much better performance than SEPT-OSMH.

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

If a schedule with maximum robustness improvement is desired, the advisable value of η is 0.8 (see 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. In this figure, ST-IMP-70 means that the stability improvement when the number of jobs is

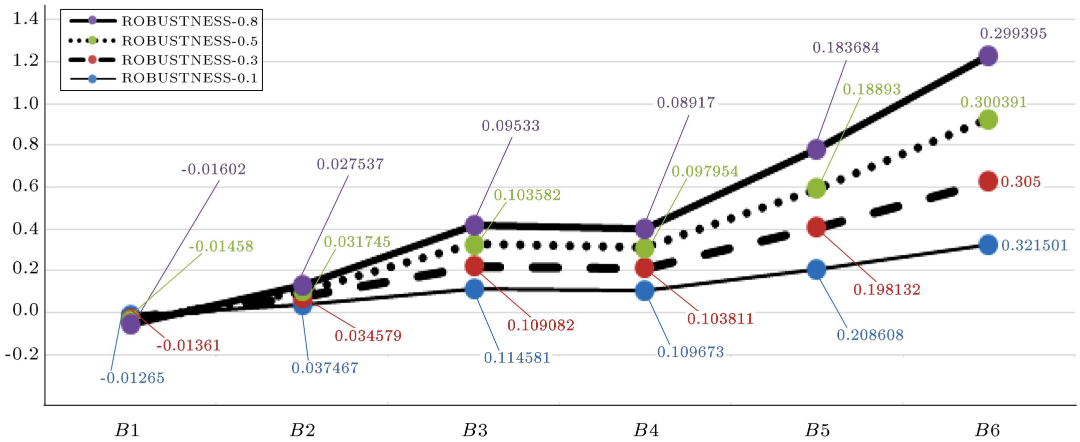

Figure 4. Robustness improvement for different breakdown types.

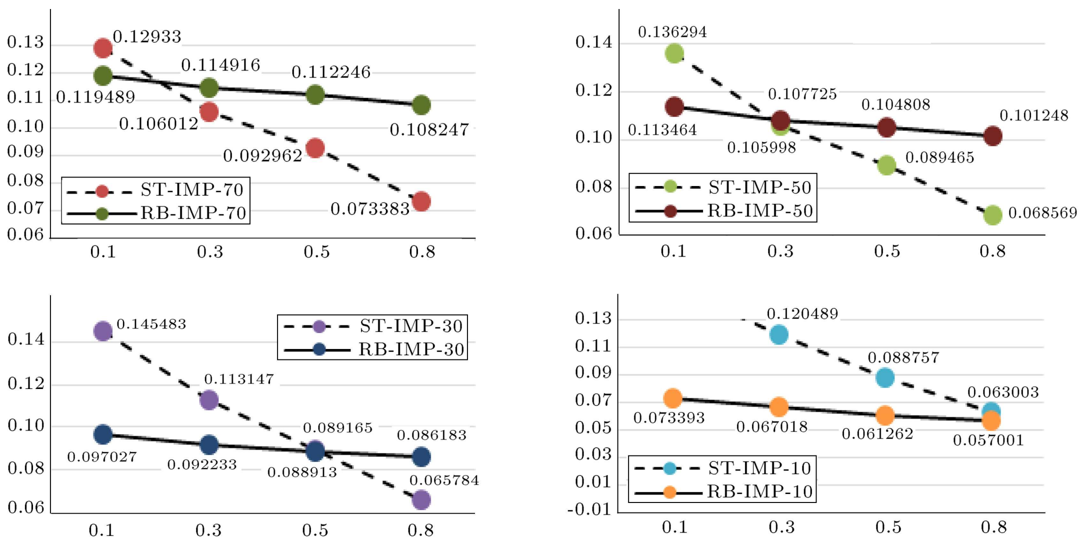

Figure 5. The robustness and stability conflict.

70, and RB-IMP-70 means the robustness improvement when the number of jobs is 70.

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

An increase in the value of η that Eq. (13) is less restricted. That is, in order to enhance the robustness and stability simultaneously, the former should worsen so that stability can be ensured.

6. Conclusions

The generation of a high robust and stable schedule in stochastic single-machine environments has become the focus of many studies recently; however, only a few studies have considered 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 to 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. Moreover, predictive heuristics were proposed to solve intractable problems of finding a robust and stable solution with $RM2$ (the expected total realized tardiness) as the robustness measure. Based on the results of extensive computational experiments applied to 3000 combinations of problems and breakdown characteristics, in the case of a 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 enjoys higher prediction accuracy and is subject to negligible disturbance in the scheduling generated by this method.

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

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 and uncertain processing times", *Journal of Manufacturing Systems*, **33**(1), pp. 84–92 (2014).

4. Lu, Z., Cui, W., and Han, X. "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), p. 679 (1989).

12. Ganji, F. and 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. Hamta, N., Fatemi Ghomi, N., Tavakkoli-Moghaddam, S.-M.-T., 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 scheduling A survey", *Annals of Discrete Mathematics*, **5**, pp. 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).

Appendix

Proof of Corollary 1: The proof is shown 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 job j on a machine. We have $E[q_j] = (1+\theta r)E[p_j]$ [22].

Let S be an optimal sequence, assuming that there exists a pair of adjacent jobs i and j such that $E\left[p_j\right] > E\left[p_i\right]$ and job j succeeds job i in S. Consider a sequence S' from S by swapping the positions of jobs i and j. It is shown that S' is better than S, i.e.:

$$[\alpha.RM\left(S\right) + (1-\alpha).SM\left(S\right)] - \big[\alpha.RM\left(S'\right)$$
$$+(1-\alpha).SM\left(S'\right)\big] > 0,$$

which contradicts the optimality of S;

$$[\alpha.RM\left(S\right) + (1-\alpha).SM\left(S\right)] - \big[\alpha.RM\left(S'\right)$$
$$+(1-\alpha).SM\left(S'\right)\big] > 0,$$

or:

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

It suffices to show that $\left[SM\left(S\right) - SM(S')\right] > 0$ and $\left[RM\left(S\right) - RM(S')\right] > 0$.

The proof of $\boldsymbol{[RM\left(S\right) - RM(S')] > 0}$***:*** The contribution of jobs other than i and j in the comparison of S' and S is ignored, since no changes occur and a constant called c is assumed. Supposing that the index set of jobs that precedes job i in S is denoted by BS, we have:

$$RM(S) = E\left(\sum_{m\hat{I}BS_i} q_m + q_i\right)$$
$$+E\left(\sum_{m\hat{I}BS_i} q_m + q_i + q_j\right)$$
$$+c - \left[E\left(\sum_{m\hat{I}BS_i} p_m + p_i\right)\right.$$
$$\left.+E\left(\sum_{m\hat{I}BS_i} p_m + p_i + p_j\right) + c'\right],$$

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

$$\rightarrow \left[RM\left(S\right) - RM(S')\right] = E(q_i)$$
$$-E(q_j) - \left[E(p_i) - E(p_j)\right]$$
$$= (1+\theta r).E(p_i) - (1+\theta r).E(p_j)$$
$$-\left[E(p_i) - E(p_j)\right] = \theta r.\left[E(p_i) - E(p_j)\right] > 0.$$

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

Biographies

Zeinab Abtahi is a PhD Candidate at Shahed University. She graduated from Sharif University of Technology and received Master's degree from Shahed University. Her research interests include stochastic scheduling, robust optimization, and integer programming.

Rashed Sahraeian is an Associate Professor at 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 include optimization in scheduling, facility location problem, and supply chain management. He is the 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 & Technology, Tehran in 2009. She worked as an Assistant Professor of Industrial Engineering at 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.