A new proactive-reactive approach to hedge against uncertain processing times and unexpected machine failures in scheduling problems

Donya Rahmani

Department of industrial engineering, K.N. Toosi University of Technology, Tehran, Iran.
drahmani@kntu.ac.ir

Abstract

In this paper, a proactive-reactive approach has been considered for achieving stable and robust schedules despite uncertain processing times and unexpected machine failures in a two-machine flow shop system. In the literature, Surrogate Measures (SMs) have been developed for achieving stable and robust solutions against the occurrence of stochastic disruptions. These measures proactively provide an approximation of the real conditions of the system in the event of a disruption. Because of the discrepancies of these measures with their real values, a different approach is developed in this paper in a two-step structure. First, an initial robust schedule is produced, and then, based on a multi-component measure, an appropriate reaction is adopted against unexpected machine failures. Computational results indicate that this method produces better solutions compared to the other two classical scheduling approaches considering their effectiveness and performance.

Keywords: Disruption, Robustness, Stability, Nervousness, Flow shop, Proactive-reactive Approach

1. Introduction

In the classical scheduling problems, it is assumed that the information about all jobs and their characterizations are known and full prior and the objective is usually to optimize some classical performance measures such as tardiness and makespan under deterministic assumptions (Beck, 2007). However, in dynamic manufacturing environments, the scheduling problems with uncertainty are addressed. In *stochastic* scheduling problems, the solving approaches often try to optimize the expected value or some other probabilistic measure of the objective function (Beck & Wilson, 2007). Terekhov, Tran , Down & Beck (2014) integrated the queueing theory and scheduling to address the stochastic due to uncertainty about the arrival times and processing times of jobs.

In uncertain scheduling problems other measures are proposed to incorporate the uncertainty (Sabuncuoglu, I. and Goren, S. 2009). Ouelhadj and Petrovi (2009) presented a survey on dynamic scheduling in manufacturing systems. In fact, because of random disruptions that may occur in the system, additional measures (such as robustness and stability) should be considered. Some disruptions that may occur in real-world manufacturing systems include:

- Machine breakdowns
- Cancellation of orders
- Changes in delivery times
- Uncertain due dates
- Uncertain processing times
- Equipment overhaul
- Addition or removal of operations

In practice, the scheduling process starts by determining an initial schedule. Then, when a disruption arises, the initial schedule should be revised in order to maintain its feasibility and quality. The type of schedule that is actually carried out in shops is known as *real schedule*. Obviously, real schedule can be different from the initial schedule because of occurrence of unexpected events. This depends on the level of failures, disruptions, and the changes of the setting. In the literature, there are two strategies for obtaining high system performance from the real schedule after any disruption. These strategies are called *reactive scheduling* and *proactive scheduling*.

The *reactive scheduling* approach does not initially consider uncertainty, but revises and improves the schedule when unpredicted events occur. In fact, reactive scheduling is a method that provides good reactions when encountering failure and disruption. This reaction could be modifying the initial schedule or generating a completely new schedule. However, *proactive scheduling* strategy considers future disruptions when generating an initial schedule. It is actually seeking a schedule that also controls the effects of future disruptions by some predictive performance measures such as *robustness* and *stability*. Optimization of stability is concerned with the deviation of the real schedule relative to the initial schedule. Optimization of robustness is concerned with deference in terms of objective function between initial and modified schedules, Kuchta (2011). Of course, a combined proactive-reactive approach can also be considered (O'Donovan et al., 1999).

In this paper, a two stage proactive-reactive method is presented for coping with uncertain and unexpected events. In the first stage, it is attempted to produce an initially robust schedule by using the robust optimization approach. The initial robust schedule counters the fluctuations of processing times. In the second stage, when an unexpected disruption occurs (i.e. machine failure), an appropriate reaction is adapted to rescheduling.

This paper is organized in the following manner. In Section 2, related technical literatures are reviewed, and a brief description of robust optimization approach is presented. The main two-step approach of the paper is presented in Section 3. In the first step, a robust model of two-machine flow shop scheduling problem has been presented and solved, and in the second step, the appropriate reactive approach has been described. Computational results and relevant comparisons have been presented in Section 4; and finally, conclusions and recommendations for future studies have been discussed in the last section.

2. Literature review

Flow-shop is one of the most practical and real-word production environments especially in assembly facilities (Pinedo, 2008). Some researchers including Fattahi et al. (2012) proposed mathematical models to formulate the flow shop environment, and some heuristics to solve the problem.

In stochastic and dynamic manufacturing environments, due to the possible occurrence of random disruptions, it is not sufficient to just establish initial schedules that minimize classical objectives like makespan. In these conditions, the timely and proper responses to the random events are particularly important. In deterministic production environments without any random disruptions, the two-machine flow shop-scheduling problem will be easily solved by the Johnson algorithm (Johnson, 1954) where the makespan is to be minimized. However, with uncertain processing times and the probability of machine breakdowns, there will be machine non-availability intervals; hence, this algorithm may not work optimally. Braun et al. (2002) investigated the circumstances in which the Johnson algorithm could operate optimally despite the existence of machine non-availability intervals. They also showed that this problem would be NP-hard even if there were only one non-availability interval. In this paper, a two-machine flow shop system is studied in which the processing times are uncertain and unexpected disruptions such as machine failures may occur in the system.

In order to decrease the effect of uncertain processing times, some researchers considered a specific distribution function for them and solved the problem based on the stochastic optimization approach. Some other researchers used the robust optimization approach so that this approach can improve the performance of the presented schedule by facing the fluctuations of uncertain processing times concerning all possible future scenarios. Daniels and Kouvelis (1995), Kouvelis et al. (1992), Mulvey et al. (1995) and Rossi (2010) presented a proactive scheduling method that deals with the future fluctuations of uncertain parameters using the robust optimization approach.

Considering disruptions and unexpected events in scheduling systems, the researchers either used iteration based simulation methods (Kutanoglu and Sabuncuoglu, 2001), or attempted to develop robust and stable schedules to face these disruptions. Wu et al. (1993) considered increase of stability in the single-machine rescheduling problem with machine breakdowns. They rescheduled the jobs in response to machine failures so that the minimum makespan could achieve a high scheduling stability. Leon et al. (1994) studied the issue of robustness in the job shop environment. Their goal was to obtain a robust initial schedule. They developed an offline pre-scheduling plan to achieve high performance for the system in case of machine failure. In their model, machine breakdowns are considered as failures and the right-shifting is used to response to the failures. They assumed that the times of failure and repairing are known values, and makespan is the shop performance measure that should be minimized. For analysis of the effects of machine failures and changes of the processing times, the authors proposed a *slack time based robustness measure*. The most promising robustness measure is expressed as:

$$Z_{r=1}(s) = MS_{\min} - RD3(s) \tag{1}$$

where MS_{min} is the makespan of schedule s and $RD3(s)$ is the average slack in schedule s. They tested the performance of their proposed algorithm under random machine breakdown and variable processing time. Their results showed that the proposed algorithm prevails over classical algorithms that only emphasize makespan minimization. The difference was significant when the variability of the processing time was sufficiently large. Lawrence and Sevelle (1997) studied the performance of the simple dispatching heuristics against the algorithmic solution techniques in a job shop environment with uncertain processing times. A similar study was undertaken by Sabunkuglu and Karabuk (1999), in which they showed that the dispatching rules for interruptions are more robust compared to the optimum search algorithms for offline schedules. Jensen (2003) generated robust schedules in a job shop environment with respect to machine breakdowns for minimizing makespan. The author defined two neighborhood-based robustness measures. The first measure is the average makespan of the given schedule's neighbors. The scheduled neighborhood is considered as the all-scheduling plans that can be achieved through pair displacement of two consecutive jobs on a machine. The second robustness measure is estimation (upper limit) of the first measure. Jensen's idea is based on the principle that the robust optimal solution is found in the wider regions of the distribution (objective) function, while the non-robust and fragile optimal solutions are located on the narrow peaks of the distribution function. In fact, he considered the neighborhood-based robustness measure, for including the schedule *s* and all the achievable schedules. These plans can be produced from schedule *s* by swapping two consecutive jobs on a machine. This measure is a weighted average of makespans of schedules in $N_1(s)$. His formula is as follows:

$$Z_{MS_{nib}}(s) = \frac{1}{|N_1(s)|} \sum_{s' \in N_1(s)} MS_{\min}(s') \tag{2}$$

where $MS_{min}(s')$ is the makespan of schedule s'. The neighborhood $N_1(s)$ contains *s* and all feasible schedules that can be created from *s* by interchange of two consecutive operations on the same machine. Rangsaritratsamee et al. (2004) proposed a rescheduling method based on the local search genetic algorithm for solving the job shop scheduling problem with the consideration that the jobs arrive dynamically. Their proposed algorithm simultaneously considers efficiency along with the preservation of makespan, tardiness and stability, and considers robustness by minimizing deviation of the job startup time. Lambrechts et al. (2008) proposed a tabu search algorithm that uses a free slack-based objective function for producing robust reactive-proactive scheduling plans in spite of uncertain renewable resource availabilities. Goren and Sabuncuoglu (2008) investigated the problem of robust and stable schedule with random failures in a single machine environment. They presented two surrogate measures for robustness and stability, and used the tabu search algorithm to solve the problem. Ultimately, they solved and analyzed the problem. by considering a linear function which combines the two criteria of robustness and stability at the same time (y = *w*robustness measure* + *(1-w)*stability measure*; *w* is the weight of robustness) . Sotskov et al. (2009, 2012) presented a number of approaches based on interval processing times for the evaluation of robustness and stability in a single-machine environment. Yang and Yu (2002) show that the robust version of the the single machine scheduling problem with sum of completion times criterion is NP-complete even for very restricted cases. They present an algorithm for finding optimal solutions for the robust problem using dynamic programming. Bouyahia et al. (2009) proposed a probable comprehensive approach for the robustness design of pre-scheduling, which assumes that the number of jobs to be processed on parallel machines is a random variable. They studied the total weighted flow time as an objective function. Ghezail et al. (2010) proposed a qualitative graphical approach for responding to the disruptions in the flow shop problem. They proposed a graphical approach that helps the decision maker to observe the consequences of random failures and to choose the best sequence. Al-Hinai and ElMekkawy (2011) produced proactive robust and stable solutions for the flexible job shop scheduling problem with random failures. They introduced a new methodology that combines the approach of insertion of non-idle time and a hybrid genetic algorithm proposed by Al-Hinai and ElMekkawy (2011). Based on the literature, the researchers separately considered stability and robustness to face the stochastic disruptions in scheduling problems. To produce robust and stable solutions, the true value of uncertain parameters should be determined. However, since the exact values of these parameters are not specified from the start, either iteration based time-consuming simulation methods or surrogate measures are used in the literature to obtain robust and stable solutions. Because of the discrepancies of these measures with their true values, they may not show the true performance of the system. We proposed a new proactive-reactive approach instead of SMs to overcome their weaknesses and achieve good-quality solutions. We also considered a new practical measure called "nervousness" in the two-machine flow shop scheduling in addition to the stability and robustness. Accordingly, a multi criteria measure is presented in the reactive stage of proposed method.

In the following, the structure of robust optimization approach used to formulate our considered problem is explained (Leung and Wu, 2004).

Robust optimization approach: The aim of the robust optimization approach is to get a set of solutions for the problem that remains robust despite the changes that may occur in the real values of data and input parameters (shown by a set of scenarios). The structure of robust optimization is briefly explained in the following section.

Consider the following linear model:

$$\min \mathrm{c}^{\mathrm{T}} x + \mathrm{d}^{T} y, \tag{3}$$

Subject to:

$$Ax = b, \tag{4}$$

$$Bx + Cy = e, \tag{5}$$

$$x, y \geq 0$$

where x is the vector of decision variables and y is the vector of control variables. B and C are coefficient vectors and e is the right hand side vector. Assume that there exists a set of scenarios, $\Omega = \{1, 2, \ldots, \tilde{\lambda}\}$. Under each scenario λ, the uncertain coefficients are defined as $\{d^{\lambda}, B^{\lambda}, C^{\lambda}, e^{\lambda}\}$ and the probability of occurrence of each scenario is p^{λ} $(\sum_{\lambda} p^{\lambda} = 1)$. Each scenario comprises a set of data that may occur in the future. Since y (vector of control variables) is determined in the constraint depending on the scenario that occurs, it is defined as y^{λ}. Due to the existence of uncertain parameters, the model may become infeasible for some scenarios which in this case, δ^{λ} is the degree of infeasibility of the model under scenario λ. If the model is feasible, then $\delta^{\lambda} = 0$. So δ^{λ} is an error vector that will measure the infeasibility allowed in the control constraints under scenario λ. The ultimate goal of this approach is the optimization of the problem with two kinds of robustness: *solution robustness*, which guarantees a near-optimum solution for all the scenarios and *model robustness* which guarantees the problem solution to be feasible for all the possible scenarios. Therefore, the robust model is made as follows:

$$\text{Min } \sigma(x, y^1, y^2, \ldots, y^{\tilde{\lambda}}) + \omega\rho(\delta^1, \delta^2, \ldots, \delta^{\tilde{\lambda}}) \tag{6}$$

Subject to:

$$Ax = b, \tag{7}$$

$$\mathrm{B}^{\lambda} x + C^{\lambda} y + \delta^{\lambda} = e^{\lambda} \qquad \forall \lambda \in \Omega, \tag{8}$$

$$x \geq 0 \ \ , \mathrm{y}^{\lambda} \geq 0, \ \ \delta^{\lambda} \geq 0 \qquad \forall \lambda \in \Omega, \tag{9}$$

where σ(.) denotes the solution robustness and ρ(.) is the model robustness. In fact $\rho(.)$ is a penalty function for the solution possibility, which is used to penalize the deviations of control constraints under some of the scenarios. Also, coefficient ω establishes the equilibrium between the solution robustness and model robustness.

The first term in the objective function can be defined with a random variable such as $\xi = c^T x + d^T y$ that takes the value $\xi^{\lambda} = c^T x + d^{\lambda^T} y^{\lambda}$ with probability p^{λ} under scenario λ. To define σ(.), Mulvey et al. (1995) used the following relation:

$$\sigma(.) = \sum_{\lambda \notin \Omega} p^{\lambda} \xi^{\lambda} + \gamma \sum_{\lambda \in \Omega} p^{\lambda} (\xi^{\lambda} - \sum_{\lambda' \in \Omega} p^{\lambda'} \xi^{\lambda'})^2, \tag{10}$$

In which, the solution has a lower sensitivity to the changes of uncertain data as γ increases. However, since this term is quadratic, its solving will be complicated, so Yu and li (2000) defined another term for σ(.) as:

$$\sigma(.) = \sum_{\lambda \notin \Omega} p^{\lambda} \xi^{\lambda} + \gamma \sum_{\lambda \in \Omega} p^{\lambda} \, | \, \xi^{\lambda} - \sum_{\lambda' \in \Omega} p^{\lambda'} \xi^{\lambda'} \, |, \tag{11}$$

But since this term is non-linear, by defining a non-negative deviational variable, the problem is converted to a linear model as follows:

$$\sigma(.) = \sum_{\lambda \notin \Omega} p^{\lambda} \xi^{\lambda} + \gamma \sum_{\lambda \in \Omega} p^{\lambda} [(\xi^{\lambda} - \sum_{\lambda' \in \Omega} p^{\lambda'} \xi^{\lambda'}) + 2\theta^{\lambda}], \tag{12}$$

Subject to:

$$\begin{aligned} &\xi^{\lambda} - \sum_{\lambda' \in \Omega} p^{\lambda'} \xi^{\lambda'} + \theta^{\lambda} \geq 0 \ , \quad \forall \lambda \in \Omega \\ &\theta^{\lambda} \geq 0 \ , \quad \forall \lambda \in \Omega \end{aligned} \tag{13}$$

3. Proposed Method

In this paper, a two-machine flow shop problem with uncertain job processing times is considered. There is some information about uncertain processing times, and are estimated by scenarios. In addition to uncertain processing times, unexpected failures may occur in the future, but there is no predictive information in this regard. Therefore, in the first stage of the proposed method, unexpected machine failure is not considered, because they are completely unexpected events and there is no information to formulate them. Hence, in the first stage, we obtain an initial solution for scheduling by only considering the problem with uncertain processing times. The robust optimization approach is used in order for this initial solution to be a robust solution as well. Actually, to reduce the effect of uncertainty on processing times, which is a random disruption in the future, we first formulate the problem using the robust optimization approach and attempt to produce robust initial solutions. After an initially robust schedule is determined, the machines start to process the jobs according to this schedule. However, each of the two machines may break down during the processing of jobs. Therefore, in the second stage, a reactive approach is presented to deal with such unexpected failures. In fact, when a machine failure occurs, an appropriate reaction should be adopted to handle this disruption. Suppose that different reactions, like regeneration, right shifting or any other heuristic reaction, can be implemented following the failure. When adopting a reactive response, we define a multi-component measure based on a classic objective and three other performance measures. This measure helps us choose the most appropriate reaction to counter the effect of machine failures. A flow chart of our proposed approach is presented in Figure. 1.

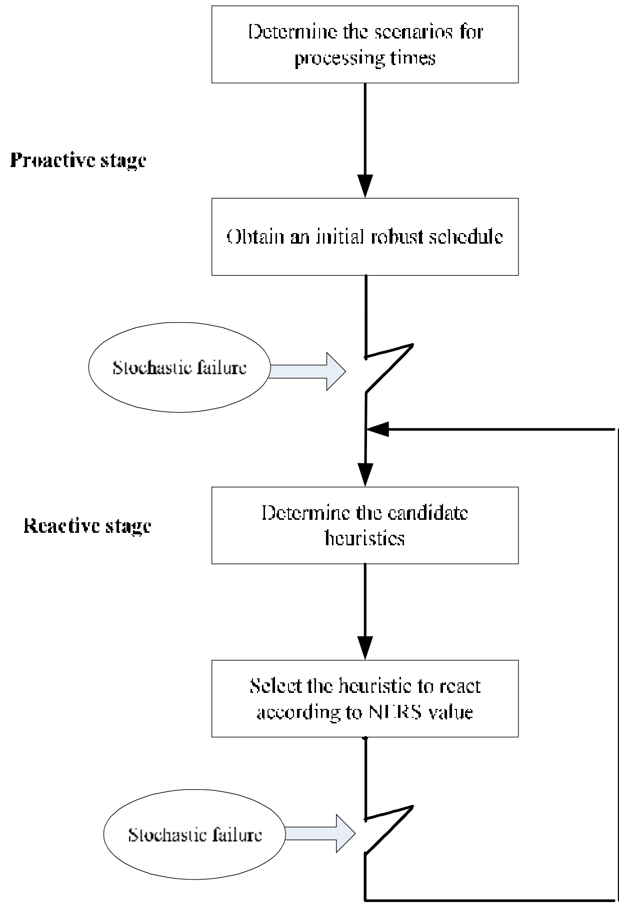

Figure 1: The flow chart of the proposed proactive-reactive method

3.1. Proactive scheduling stage

In this stage, we only consider the problem with uncertain processing times that are estimated with scenarios. We use the robust optimization approach to formulate the two-machine flow shop problem to reduce the effects of fluctuations of the processing times in the future. Consider a set $J = \{1, 2,\ldots, n\}$ of n independent jobs that require processing on each of the two machines. Assume that there exists a limited set of $\Omega = \{1, 2, 3\}$: one pessimistic, one optimistic and one normative. In fact, we proactively generate a more robust solution as an initial schedule without considering machine breakdowns in the first stage.

3.1.1 Notations

- *Indices:*

n The number of jobs

λ, λ' Indexes for scenarios {1, 2, 3}

r Index for orders $\{1, 2, \ldots, n\}$
j Index for jobs $\{1, 2, \ldots, n\}$
i Index for machines $\{1, 2\}$

- *Parameters:*

t_{ij}^{λ} : The processing time of job j on machine i under scenario λ

p^{λ} : The probability for happening of scenario λ

M : A large number

- *Variables:*

x_{ijr} : 1 if the job j processed on machine i in the order r; 0 otherwise

CP_{ijr}^{λ} : The predictive completed time of job j on the machine i in the order r under scenario λ

3.1.2. Proactive Robust Model

Now, we formulate the considered problem based on the robust optimization approach described in the previous section. The noteworthy point is that in this model, due to the unequal form of the existing constraints, it is not necessary to define parameter δ^{λ} and function $\rho(\delta^1, \delta^2, ..., \delta^{\tilde{\lambda}})$ to guarantee solution robustness. Therefore, the issue of creating a balance between solution robustness and model robustness no longer exists. Therefore, our developed robust model is as follows:

$$Min \quad \sum_{\lambda \in \Omega} p^{\lambda} \sum_{j=1}^{n} CP_{2jn}^{\lambda} + \sum_{\lambda \in \Omega} p^{\lambda} \left| (\sum_{j=1}^{n} CP_{2jn}^{\lambda} - \sum_{\lambda' \in \Omega} p^{\lambda'} \sum_{j=1}^{n} CP_{2jn}^{\lambda'} \right| \tag{14}$$

Subject to:

$$\sum_{r=1}^{n} x_{ijr} = 1, \ \forall i, \forall j \tag{15}$$

$$\sum_{j=1}^{n} x_{ijr} = 1, \ \forall i, \forall r \tag{16}$$

$$\sum_{j=1}^{n} CP_{ij(r+1)}^{\lambda} \geq \sum_{j=1}^{n} CP_{ijr}^{\lambda} + \sum_{j=1}^{n} x_{ij(r+1)} . t_{ij}^{\lambda}; \ \forall i, \forall \lambda, r = 1, ..., n-1 \tag{17}$$

$$\sum_{r=1}^{n} CP_{(i+1)jr}^{\lambda} \geq \sum_{r=1}^{n} CP_{ijr}^{\lambda} + \sum_{r=1}^{n} x_{(i+1)jr} . t_{(i+1)j}^{\lambda}; \ \forall j, \forall \lambda, i = 1, ..., m-1 \tag{18}$$

$$CP_{ijr}^{\lambda} \leq M . x_{ijr}; \ \forall i, \forall j, \forall r, \forall \lambda \tag{19}$$

$$x_{ijr} = \{0,1\}; \ \forall i, \forall j, \forall r \tag{20}$$

To linearize the objective function, we used the Yu and Li (2000) method and defined parameter θ^{λ} described in the previous section. Therefore, we have:

$$Min \quad \sum_{\lambda\in\Omega} p^{\lambda} \sum_{j=1}^{n} CP_{2jn}^{\lambda} + \sum_{\lambda\in\Omega} p^{\lambda} [(\sum_{j=1}^{n} CP_{2jn}^{\lambda} - \sum_{\lambda'\in\Omega} p^{\lambda'} \sum_{j=1}^{n} CP_{2jn}^{\lambda'}) + 2\theta^{\lambda}], \tag{21}$$

$$-\theta^{\lambda} - \sum_{j=1}^{n} CP_{2jn}^{\lambda} + \sum_{\lambda'\in\Omega} p^{\lambda'} \sum_{j=1}^{n} CP_{2jn}^{\lambda'} \leq 0; \quad \forall\lambda \tag{22}$$

Thus, objective (21) and constraints (22) ensure that the optimal schedule conforms to the definition of robust schedule based on a linear model. Constraints (15) to (20) are necessary scenario based constraints in a two-machine flow shop system to calculate appropriate makespan. These constraints guarantee that the robust schedule is feasible.

3.2. Reactive Scheduling Stage

Assume that in the first stage, an initial robust schedule is obtained. The machines may break down during the processing of jobs unexpectedly. Suppose that machine *i* fails at time t_f. In this case, the main point is to choose an appropriate heuristic as a good reaction after this failure. The important point that should be mentioned is that the proposed approach to respond to the machine failure is a reactive one. Therefore, there is no need to know the distribution function of the failure occurrence time and repair duration.

Suppose that a set *Π*= $\{H_1, H_2, ..., H_h\}$ of heuristic methods exists, which can be used after the occurrence of failure. These heuristic methods can be the *right shifting*, *regeneration* or any other heuristic. In this research a multicomponent measure is defined to choose the most appropriate heuristic method from the set *Π*. This measure is defined based on a classical objective and three other performance measures. The considered classical objective in this research is makespan that indicates the scheduling effectiveness. Other performance measures include robustness, stability and nervousness that control the unexpected disruptions. We call this measure "NERS value" (include **N**ervousness, **E**ffectiveness, **R**obustness and **S**tability). These components are described in the following subsections.

3.2.1. Scheduling Effectiveness

This measure indicates the degree of optimality for a schedule. In this paper, this criterion is measured by the classical objective "*makespan*". We should mention that because of disruptions such as machine failures in the system the real completion time of affected jobs may change. The stochastic variable CR_{ijr} is defined as the real completion time of job *j* on machine *i* at position *r*. Therefore, in a two-machine flow shop system $\sum_j CR_{2jn}$ is the real makespan for realized schedule that explains scheduling effectiveness.

3.2.2. Robustness

Dooley and Mahmoodi (1992) stated that the robustness of a schedule refers to its ability to perform well under different operational environments including dynamic and uncertain conditions. Some robustness measures are based on the actual performance of the realized schedules. In fact, the robustness is concerned with the difference in terms of objective function value. It refers to the insensitivity of scheduling performance to the disruptions. In general, the performance of the realized schedule is the main concern of practitioners rather than the estimated performance of the initial schedule. Some robustness measures are based on the actual performance of the realized schedules, and some are based on *regret.* (Sabuncuoglu, I. and Goren, S. (2009)). In this research this measure is defined as the closeness of the realized schedule performance to the initial schedule. The higher the predictability, the lower the costs of under-utilization or overtime.

It should be mentioned that in the first stage, in order to determine the initial schedule, the processing times were estimated as scenarios. In fact some issues such as the condition of machines, the state of operators, the environmental conditions, etc effect on processing times and the real states of these issues will be determined only at schedule execution time. So three scenarios (one pessimistic, one optimistic and one normative) were considered to define the uncertain processing times to obtain a robust initial schedule. The presumption is that in reality, only one of these scenarios will occur in the future. During the planning an initial schedule the real scenario that will occur really in system is not determined. But when the production process begins, it is certain which scenario has occurred. Considering this notion, suppose $\lambda'' \in \Omega$ is the scenario that has really occurred after the start of jobs processing. In this case, the real completion times without considering unexpected failures are specified based on scenario $\lambda'' \in \Omega$. Based on this matter and the definition of robustness (absolute deviation in system performance) , the robustness measure is calculated as follows:

$$\textit{Robustness measure}: \left| \sum_{\mathrm{j}} \mathrm{CR}_{2\mathrm{jn}} - \sum_{\mathrm{j}} \mathrm{CP}_{2\mathrm{jn}}^{\lambda''} \right| \tag{23}$$

That $\sum_j CR_{2jn}$ is the real makespan of the perturbed schedule, and $\sum_j CP_{2jn}^{\lambda''}$ is the predictive makespan according to the initial schedule that is determined according to the occurred scenario λ''.

3.2.3. Stability

This is the degree of rearrangement of jobs (sequence, start-times and so on) after rescheduling (Gan and Wirth, 2005). In this paper this measure is defined as the difference between the completion times of the jobs in the initial schedule and the realized ones (Al-Hinai and ElMekkawy, 2011). When a disruption occurs, the real sequence may change after a needed rescheduling. This matter may lead to additional costs including the cost of reallocation of tools and equipment, and cost of reordering raw materials etc. However, when the real schedule is closer to the initial one these costs are reduced and stability is increased. On the other hand, stability is concerned with the difference between initial and realized schedules themselves, rather than between their performances. Therefore, stability measure is defined as absolute deviation in job completion times as follows:

$$Stability\ measure: \sum_{i=1}^{2}\sum_{j=1}^{n}\sum_{r=1}^{n}\left|CR_{ijr} - CP_{ijr}^{\lambda''}\right| \tag{24}$$

CR_{ijr} is the real completion time of job *j* on the machine *i* in order *r*, and $CP_{ijr}^{\lambda''}$ is the predictive completion time of job *j* on machine *i* in order *r* under occurred scenario λ''.

3.2.4. Nervousness

Schedules are intrinsically nervous and fragile with some unexpected information. This information is not known a priori in the planning phase and revealed over time. So dynamic or on-line scheduling techniques are usually used (Gan and Wirth, 2005 and Rahmani and Heydari, 2013). To show the influence of nervousness in scheduling problems another approach is developed in this paper and so a new measure is defined such as the measures of robustness and stability. This measure includes the effect of frequency of rescheduling in the system after disruptions. When a disruption occurs in a system and the sequence of jobs changes after a rescheduling, the changing in the sequence causes nervousness in the system. In fact, internal system components like the operators, will fall into disarray due to change in the sequence, and the more this rescheduling frequency becomes, the higher the nervousness of the system will be. Therefore, when the sequence of one or several jobs in the real schedule changes relative to the initial schedule, the system becomes turbulent and chaotic. In this paper, we have differentiated between the concepts of instability and nervousness. For example, the jobs may be right-shifted after a failure. In this way, due to the change in the completion time of jobs, the stability measure will be a positive number, while the nervousness is zero, because the sequence of jobs does not change.

We assume that $PO_{ij}^{\lambda''}$ is the predictive order of job *j* on machine *i* under occurred scenario λ'' and RO_{ij} is the real order of job *j* on machine *i*, following a failure. We define a variable as follows:

$N_{ij} = 1$ if $RO_{ij} \neq PO_{ij}^{\lambda''}$, and $N_{ij} = 0$ otherwise.

In fact, if the order of job j in the real schedule be different from the predictive one, then $N_{ij} = 1$. Now, we define variable TCO as the total changing in order of jobs following a unexpected failure, therefore $TCO = \sum_{i=1}^{2}\sum_{j=1}^{n} N_{ij}$ .The TCO describes nervousness of scheduling.

*3.2.5. **NERS** value as a selector measure*

The NERS value is a multicomponent measure for the selection of a suitable heuristic to deal with machines failure. The final definition of NERS value based on the above defined components is as follows:

Selection measure = *α (Effectiveness) +β (Robustness measure) +ν (Stability measure) +φ (Nervousness)* (25)

The existing coefficients in this measure indicate the importance of each component. These values are determined by system's decision makers such as managers. So in the proposed reactive stage, when a failure occurs, first the values of these coefficients are determined by the managers of that production system. To determine the weight of each measure we can use a sensitivity analysis approach such as the research that is presented by (Gan and Wirth, 2005). They similarity define a comparison metric with four components to comparing deterministic, robust and online scheduling. They used a sensitivity analysis to determine the weights of each component. So in a real case the analyst can considered the different amounts of classical measure and other three components and the manager can choose the better one according to her/his preferences. Moreover, the approaches for determining preference weights in multi criteria decision making such as the eigenvector method, weighted least-square method, entropy method, and linear programming technique for multidimensional analysis of preference (LINMAP). The readers are referred to Hwang and Yoon, 1981) for more study. The definition of NERS value is as follows:

$$\text{NERS value}_{\mathrm{H}} = \alpha\left(\frac{(\sum_{\mathrm{j}} \mathrm{CR}_{2\mathrm{jn}})_{\mathrm{H}}}{\max_{\mathrm{H}\in\Pi}(\sum_{\mathrm{j}} \mathrm{CR}_{2\mathrm{jn}})_{\mathrm{H}}}\right) + \beta\left(\frac{\left|\sum_{\mathrm{j}} \mathrm{CR}_{2\mathrm{jn}} - \sum_{\mathrm{j}} \mathrm{CP}_{2\mathrm{jn}}^{\lambda''}\right|_{\mathrm{H}}}{\max_{\mathrm{H}\in\Pi}\left|\sum_{\mathrm{j}} \mathrm{CR}_{2\mathrm{jn}} - \sum_{\mathrm{j}} \mathrm{CP}_{2\mathrm{jn}}^{\lambda''}\right|_{\mathrm{H}}}\right) + \nu\left(\frac{\sum_{i=1}^{2}\sum_{j=1}^{n}\sum_{r=1}^{n}\left|CR_{ijr} - CP_{ijr}^{\lambda''}\right|_{H}}{\max_{H\in\Pi}\sum_{i=1}^{2}\sum_{j=1}^{n}\sum_{r=1}^{n}\left|CR_{ijr} - CP_{ijr}^{\lambda''}\right|_{H}}\right) + \varphi\left(\frac{TCO_H}{\max_{H\in\Pi} TCO_H}\right) \qquad (26)$$

$$\alpha + \beta + \nu + \phi = 1 \tag{27}$$

A set of heuristics can be compared using the NERS value that is defined term 26. In fact, after determining the weights, the NERS value is calculated for each heuristics such as right shifting, regeneration and etc, and the heuristic with lowest amount of NERS value is selected as a proper reaction to response the disruption. The terms in (25) will be normalized to enable a reasonable comparison between heuristics (Gan and Wirth, 2005).

3.3. Heuristic methods

There are different heuristics to deal with unexpected disruptions. More common methods in the literature are regenerations and right shifting methods that explain in the following subsections.

3.3.1. Right shifting

One of the simplest reactions which are commonly implemented in the literature, following the occurrence of a disruption, is right shifting. In this approach, when a failure happens, the jobs are processed with the same previous sequence following the repair of failed machine. In this case, the completion time of jobs will be shifted to the right within the repair duration (d_f). The schedule may be less stable in this method because the real completion time of jobs may very change. The advantage of this method is that, due to not changing the sequence, the costs of setup and startup will be less, and also the system nervousness will be zero. As mentioned earlier, when the manufacturing process begins, it is absolutely certain which scenario has occurred. Considering this notion, suppose that $\lambda'' \in \Omega$ is the scenario which has really occurred after the start of the jobs processing. Now, we assume that the system has a failure and the scheduled activities are shifted to the right following the failure. Two hypothetical machine failures are shown in Figure. 2. Assume that the j1, j2 and j3 are scheduled based on occurred scenario λ'' according to Figure 2(a). The Figure 2(b) and Figure. 2(c) are presented machine failures in different times on the machine 1 and machine 2, respectively. Figure. 2 shows the steps of right shifting method following the considered failures.

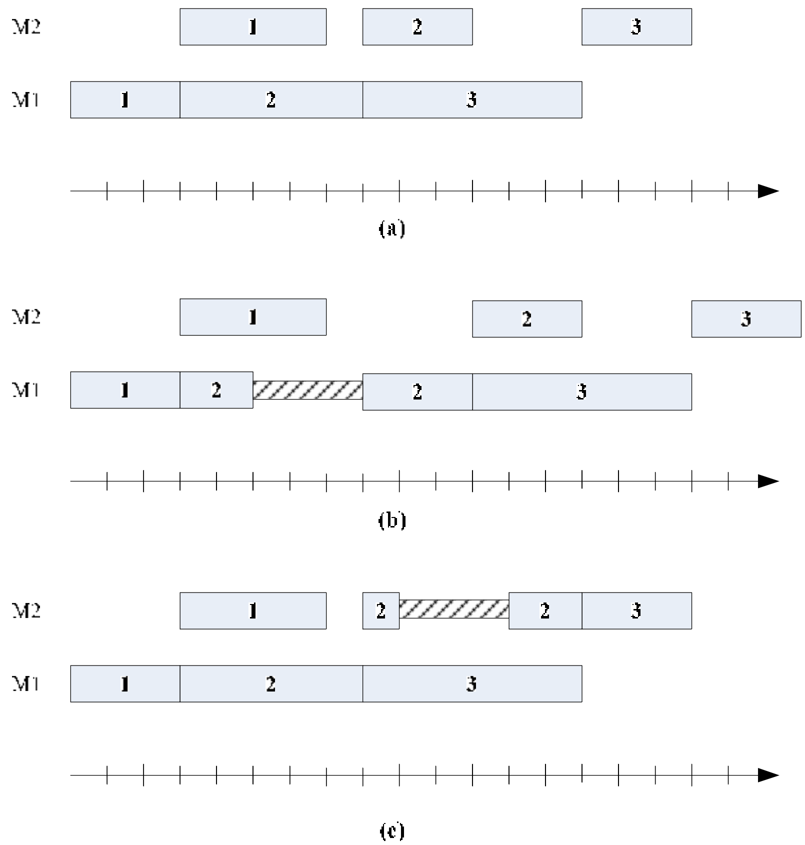

Figure. 2. (a) An initial schedule. (b) Right shifting following a failure on machine 1. (c) Right shifting following a failure on machine 2.

In general, with a repair duration d_f and occurred scenario λ'', the value of real completion time of jobs is obtained as follows:

$$r'_j = \left\{ r \in n \;\middle|\; x_{2\text{jr}} = 1 \right\} \quad ;\forall \text{j} \tag{28}$$

$$r''_j = \left\{ r \in n \;\middle|\; x_{1\text{jr}} = 1 \right\} \quad ; \quad \forall \text{j} \tag{29}$$

If the machine 1 to be failed:

$$\begin{aligned} CR_{1jr''_j} &= CP^{\lambda''}_{1jr''_j} + d_f \quad \text{if} \quad CP^{\lambda''}_{1jr''_j} \rangle t_f \quad \text{and} \\ CR_{1jr''_j} &= CP^{\lambda''}_{1jr''_j} \quad \text{if} \quad CP^{\lambda''}_{1jr''_j} \le t_f \quad ; \forall \text{j} \end{aligned} \tag{30}$$

$$CR_{2jr'_j} = \max\left\{ \sum_j CR_{2j(r'_j-1)}, CR_{1jr''_j} \right\} + \text{t}_{2\text{j}} \quad ; \forall \text{j} \tag{31}$$

If the machine 2 to be failed:

$$CR_{1jr_j''} = CP^{\lambda''}_{1jr_j''} \quad ; \forall \text{j} \tag{32}$$

$$SP_{ijr} = CP^{\lambda''}_{ijr} - t_{ij} ; \quad \forall \text{i,j,r} \tag{33}$$

$$CR_{2j'r_{j'}'} = CP^{\lambda''}_{2j'r_{j'}'} \qquad \text{if } \text{j}' = \left\{ j \in J \;\middle|\; CP^{\lambda''}_{2j'r_{j'}'} \prec t_f \text{ and } SP^{\lambda''}_{2jr_j'} \prec t_f \right\} \; ; \forall \text{j}' \tag{34}$$

$$CR_{2j''r_{j''}'} = CP^{\lambda''}_{2j''r_{j''}'} + d_f \qquad \text{if } \text{j}'' = \left\{ j \in J \;\middle|\; CP^{\lambda''}_{2jr_j'} \succ t_f \text{ and } SP^{\lambda''}_{2jr_j'} \prec t_f \right\} \; ; \forall \text{j}'' \tag{35}$$

$$\begin{aligned} &CR_{2j'''r_{j'''}'} = \max\left\{ \sum_j CR_{2j(r_j'-1)}, CR_{1j'''r_{j'''}''} \right\} + \text{t}_{2j'''} \quad \text{if} \\ &\text{j}''' = \left\{ j \in J \;\middle|\; CP^{\lambda''}_{2jr_j'} \succ t_f \text{ and } SP^{\lambda''}_{2jr_j'} \succ t_f \right\} \; ; \forall \text{j}''' \end{aligned} \tag{36}$$

3.3.2. Regeneration

Another method that can be adopted as a rescheduling method following a failure is regeneration heuristic. It reschedules the set of jobs not processed before the rescheduling point and are affected by the disruption. In this approach, all jobs that have not yet been processed are completely rescheduled. In this case, the sequence of jobs may change, and the system may become stressed and chaotic, but on the other hand, this change of sequence causes good improvement in the effectiveness (objective function). Therefore by knowing the real scenario that has occurred in the system, the remaining jobs are scheduled once again.

3.4. A numerical example

Consider a two-machine flow shop scheduling problem with five jobs, which the processing times of jobs fewer than three scenarios are given in Table 1. The probability of each scenario is $p^1 = 0.3, p^2 = 0.5, p^3 = 0.2.$, respectively. In the first stage, we should solve this problem based on the proposed model that is developed based on the robust optimization approach. We solved that model with the software GAMS23.6. Based on this software the initial schedule for machines 1 and 2 is *"j2, j1, j3, j4, j5"*. This solution is an initial robust schedule for this problem. We assume that when processing begins, the second scenario will occur really ($\lambda'' = 2$).

Table 1: Processing time for job *j* on machine *i* under scenario *λ*

Scenario	Machine	Job processing time (minute)				
		1	2	3	4	5
1	1	1	3	7	9	3
	2	2	5	7	3	3
2	1	4	2	6	8	5
	2	4	6	9	5	2
3	1	2	4	8	7	6
	2	3	4	10	2	3

Now, assume that the machine 1 is failed at the time point 9 (t_f =9) and the length of the repair time to the failed machine is d_f =3. We should do a good reaction to this unexpected disruption and reduce its effects based on the proposed multi criteria measure "NERS value". We assume that our heuristic methods that we can choose are regeneration and right shifting, so we calculate the NERS value for each of them and select the best. At first we should determine this measure's coefficients. These parameters are determined as: α=0.4, β=0.2, ν=0.3 and φ=0.1. Table 2 shows the obtained results for both regeneration and right shifting. The results indicate that following this assumptive failure, the regeneration method should choose because has smaller NERS value.

Table 2: The comparing two heuristics to select following the machine failure

Approach	Effectiveness	Robustness	Stability	Nervousness	NERS value
Regeneration	31	3	33	5	0.581
Right shifting	41	13	74	-	0.900

Since the time point of machine failure and the machine that may fail and also d_f (length of repair time) are stochastic variables, so we extended this problem with simulating failures. We assume that t_f is generated from a uniform distribution such as uniform= [0.001, 20], d_f also is generated based on uniform= [5, 10] and the machine will be failed is chosen stochastically. Then we consider three levels of the coefficients in the NERS value that is shown in Table 3. We have run the problem for each level of coefficients at 100 repetitions and finally the mean values of the effectiveness and performance objectives have calculated. The results are shown in Table 4. The worst makespan of all iterations which occurs simulated failures is shown in column 7 and the NERS values are presented in column 8.

We call our proactive-reactive method PRM for simplification. Recall that the classical approach is also used as a benchmark.

Classical approach: The algorithm calculates the initial schedules based on the expected value approach. In fact, the objective function in this approach is based on the minimizing the expected value of all makespans that were computed for each scenarios $(Z = Min \sum_{\lambda \in \Omega} p^{\lambda} \sum_{j=1}^{n} CP_{2jn}^{\lambda})$. Then when a disruption accurses, two reactions may use. In the first classical approach after a disruption, the affected jobs shift to right with the length of the repair time. We call this method *CA-Ri*. It means that the right shifting method will use following any disruptions. In the second classical approach after a disruption, the affected jobs regenerate a new schedule based on the objective function. We call this method *CA-Re* method.

Table 3: Three kinds of test problem for setting the coefficients

Test problem	Coefficient			
	α	β	γ	φ
1	0.5	0.15	0.3	0.05
2	0.4	0.2	0.3	0.1
3	0.4	0.3	0.1	0.2

Table 4: The mean values of effectiveness and performance objectives according to accomplished simulations

Prob.	Approach	Effectiveness	Robustness	Stability	Nervousness	Worst makespan	NERS value
	PRM	29.02	2.32	24.16	3	30.24	0.712
1	CA-Ri	34.97	5.77	58.01	---	36.38	0.924
	CA-Re	28.05	2.91	65.03	2.5	32.61	0.946
	PRM	31.73	2.73	32.76	2	29.96	0.590
2	CA-Ri	34.61	6.11	55.87	---	35.23	0.745
	CA-Re	31.04	3.20	85.33	3.9	33.00	0.815
	PRM	31.37	2.76	32.32	2.2	30.95	0.657
3	CA-Ri	33.99	6.09	58.18	---	37.75	0.737
	CA-Re	30.52	2.94	51.37	2.6	35.61	0.705

It should mention that there is a logical conflict between solution's stability and robustness. Because for a more robust solution may be necessary to change sequence of jobs which can increase stability. To show the conflict between the stability and robustness for this problem, we assume that the $\varphi=0$, $\alpha=1$ and also we suppose that $v=1-\beta$. Then for three values of β we solved the proposed example. The results are shown in the Table 5 and Figure 3. According to the plots it can see easily that larger value of β leads to increase stability and reduce robustness.

Table 5: The conflict between stability and robustness

Measure	Robustness coefficient		
	β=0	β=0.5	β=1
Stability	20	22	27
Robustness	6.6	5	3

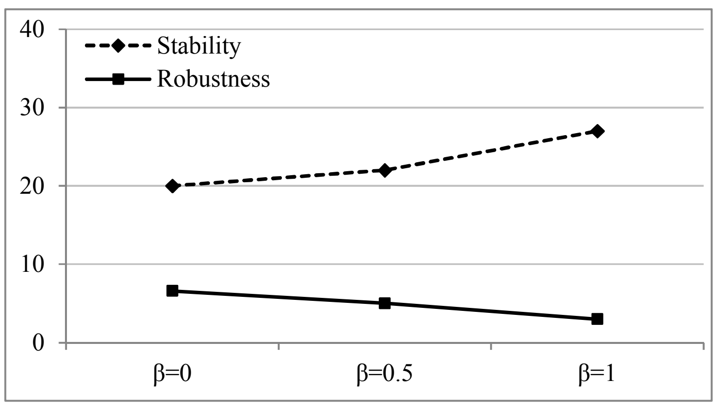

Figure 3: The robustness and stability values

4. Experimental Design

We conducted extensive computational experiments to show the performance of the proposed method. We solved a set of test problems, whose details are given in the following section.

4.1. Data Generation

Number of jobs (n): We considered seven levels of the number of jobs (n = 5, 8, 10, 12, 15, 20 and 25). In general, the number of jobs specifies the size of the problem.

*Processing time (t_{ij}^{λ}):*Job processing times were generated from discrete uniform distributions. The unit of the job processing time is minutes. The distribution was used for t_{ij}^{λ} is – Uniform [a, η^{λ} b], where *a*=10 and *b*=40. The parameter η^{λ} led to different intervals for processing times for each scenario; we assumed $\eta^1 = 1$, $\eta^2 = 1.5$ and $\eta^3 = 2$.

Break down time (t_f): This parameter explains the time point that a machine fails. The interval between two failures usually follows an exponential distribution with *MTBF* as the mean time between failures. But since our proposed method is a reactive one, we should determine a time point in any simulated run of the problem instead of the interval between failures. Since a machine may fail from the moment zero it begins processing, to the moment of proactive makespan, we assumed that this parameter was generated from a discrete uniform distribution – *Uniform [0,* $CP^{\lambda'}_{2jn}$ *]*.

Duration of repair time (d_f): This parameter follows of an exponential distribution with *MTTR* as the mean time to repair. Therefore, d_f = *exp-rand (MTTR)* and we assume *MTTR*=20 time units for any machines.

4.2. Experimental results

We considered five-type problems defined above based on the number of jobs. In the first stage, we formulated each problem based on the proposed robust model and solved them with GAMS23.6 software, and their solutions were determined as the initial robust schedules. Then we simulated machine failures in the scheduling problem with MATLAB R2007b software and run on a personal computer with a 2.26 GHz processor with 3.00 GB of RAM.

It should be mentioned that our candidate heuristics chosen as suitable reactions toward the failures were regeneration and right shifting methods, which were explained in the It should mention that our candidate heuristics to choose as a suitable reaction to response the failures were regeneration and right shifting methods that explained at previous section.

All the five test problems simulated in 1000 repetitions and the mean value of the each defined criteria was calculated for them. The obtained results for PRM, CA-Ri and CA-Re are shown in Table 6 and Figure 4.

Table 6: The mean values of effectiveness and performance criteria according to accomplished simulations

Number of job	Method	Effectiveness	Robustness	Stability	Nervousness	Worst makespan	NERS value
n=5	PRM	307.14	20.14	214.22	5.2	320.42	0.6564
	CA-Ri	290.97	51.17	415.87	---	316.14	0.7674
	CA-Re	285.78	48.15	395.01	6.8	305.29	0.7381
n=8	PRM	324.5	29.06	320.58	7.7	346.45	0.6895
	CA-Ri	339.73	23.33	356.56	---	371.45	0.8388
	CA-Re	312.8	27.42	357.91	6.1	337.00	0.8651
n=10	PRM	375.14	32.67	379.58	11.5	404.87	0.7162

	CA-Ri	419.02	49.14	423.13	---	505.47	0.9017
	CA-Re	370.66	30.76	409.76	8.4	420.12	0.8703
	PRM	384.03	37.89	388.5	10.7	429.00	0.7501
n=12	CA-Ri	423.91	34.02	421.09	---	450.53	0.9382
	CA-Re	373.76	36.87	499.32	8.3	457.91	0.9234
	PRM	418.17	35.65	419.77	11.9	456.86	0.8019
n=15	CA-Ri	436.90	41.60	446.81	---	470.90	0.9278
	CA-Re	401.87	47.98	497.06	13.9	450.07	0.9631
	PRM	493.40	46.33	467.50	14.0	550.32	0.8315
n=20	CA-Ri	512.94	57.87	469.45	---	578.98	0.9590
	CA-Re	456.82	52.80	517.90	16.1	498.55	0.9351
	PRM	562.06	49.65	499.77	16.5	581.06	0.8498
n=25	CA-Ri	587.21	65.40	476.81	---	600.53	0.96.01
	CA-Re	521.87	65.36	547.06	19.9	530.07	0.9839
	PRM	409.21	35.91	384.27	11.07	441.28	0.76
Average	CA-Ri	430.10	46.08	429.96	---	470.57	0.89
	CA-Re	389.08	44.19	460.57	11.36	428.43	0.90

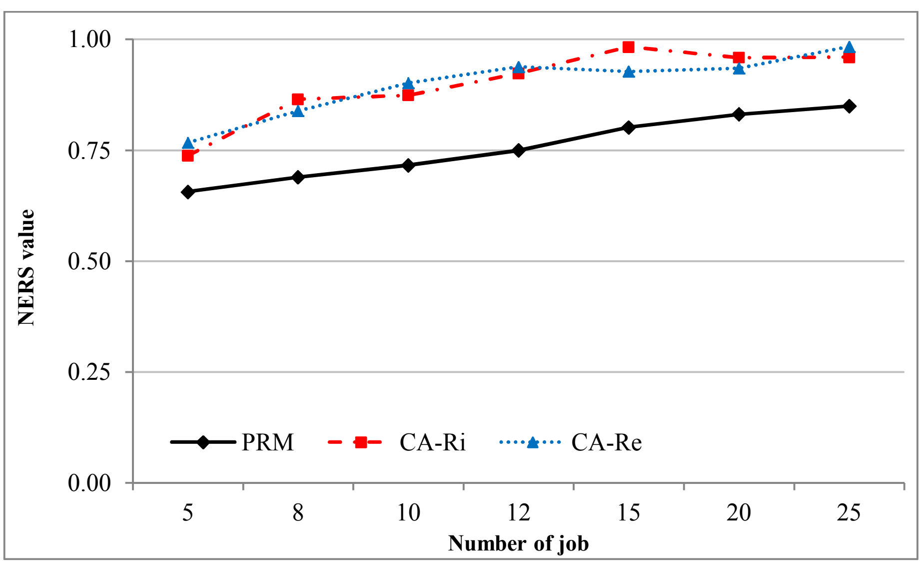

Figure 4: The comparison of NERS values of different solution methods

Table 6 shows details of computational results obtained using PRM, CA-Ri and CA-Re heuristics. These results can be used to evaluate the proposed solution methods effectiveness and efficiency. A general review of the results in Table 6 and Figure. 4 shows that:

- The NERS value of PRM is lower than the NERS value of the two other methods in each level of jobs. These results show the efficiency of the proposed approach compared to two common classical methods.
- In addition to NERS value, the average amounts of effectiveness, stability and robustness of PRM are lower than the CA-Ri and CA-Re, and only the average nervousness of PRM is greater than CA-Re.
- The NERS value for all solution methods will be worse when the problem size increases.
- For the test problems, CA-Ri and CA-Re algorithms have relatively the same average of NERS value.
- The average stability of CA-Ri is lower than the CA-Re and the average robustness of CA-Ri is greater than CA-Re. This is because that in CA-Ri, changing in the previous schedule is less.

With regard to Table 6, the effectiveness of PRM is better than other methods for n>=10. This fact indicates that in addition to NERS, this method also works much better in terms of makespan value in systems with large number jobs.

As mentioned before, the existing coefficients in the NERS value measure indicate the importance level of each component. These values are determined by system decision makers such as managers. Changing the values of these coefficients will cause change in the proper heuristic, which is selected as the reactive method. To consider the impact of these coefficients, we simulated the previous system for 10 different levels of coefficients for two different number of jobs (n=5 and n=8). The values of each component are reported in Table 7. The NERS value in column 10 is the best one obtained by two heuristics (i.e. right shifting and regeneration).

Table 7: Values of NERS value and its components for different coefficients

No.	α	β	γ	φ	**Effectiveness**	**Robustness**	**Stability**	**Nervousness**	**NERS value**
	0.6	0.3	0.05	0.05	289.09	21.30	366.08	4.3	0.76
	0.4	0.5	0.05	0.05	293.12	18.06	397.21	6.1	0.69
5	0.6	0.05	0.3	0.05	341.90	57.32	187.29	3.3	0.63
	0.4	0.05	0.5	0.05	361.03	64.25	170.32	3.5	0.88
	0.6	0.05	0.05	0.3	351.82	53.56	276.01	1.7	0.67

	0.4	0.05	0.05	0. 5	359.01	68.21	235.56	2.5	0.80
	0.5	0.1	0.3	0.1	339.31	46.05	190.03	2.1	0.77
	0.5	0.2	0.2	0.1	320.99	34.57	201.98	2.5	0.59
	0.5	0.3	0.1	0.1	295.89	25.70	279.17	3.7	0.55
	0.5	0.35	0.05	0.1	306.85	22.87	319.65	3.9	0.61
8	0.6	0.3	0.05	0.05	351.05	51.33	472.91	7.5	0.77
	0.4	0.5	0.05	0.05	355.95	56.98	490.04	8.2	0.73
	0.6	0.05	0.3	0.05	408.22	87.34	287.12	4.5	0.71
	0.4	0.05	0.5	0.05	412.90	93.33	229.72	5.8	0.91
	0.6	0.05	0.05	0.3	391.56	82.09	320.76	3.5	0.83
	0.4	0.05	0.05	0. 5	393.02	81.32	308.88	2.9	0.89
	0.5	0.1	0.3	0.1	410.55	89.25	291.30	4.5	0.85
	0.5	0.2	0.2	0.1	401.99	80.91	309.07	4.1	0.69
	0.5	0.3	0.1	0.1	381.34	63.21	387.26	6.8	0.73
	0.5	0.35	0.05	0.1	368.39	56.43	431.82	7.2	0.86

In the table 6 for n=5 (row 2 to row 7), the results shows that when a coefficient of a performance criteria (stability, robustness or nervousness) increases while two other coefficients are fixed, the value of this criteria decreases because its importance increases in the NERS measure. The NERS values in last column indicate that the coefficient of stability has a strong influence on NERS measure. In fact much larger values for γ has more negative impact on the NERS value and when it is much lesser the NERS value increase too. So a moderate amount is required. For high value of φ the NERS value increases too, but when the β increases while two other coefficients are fixed the NERS value decreases so it has a positive influence on NERS measure.

To more analysis, we assumed that α and φ are fixed and β and γ change to consider the natural conflict between stability and robustness. Based on table 7 when β increases and γ decreases then stability and robustness are changed conversely and this shows the conflict between these two measures. On the other hand, the NERS values for these four levels of coefficients are considered to gain a proper tradeoff between these two parameters. The conclusion shows for the coefficients in the 10^{th} row the NERS value is the best. So the different levels of the coefficient lead to different amounts for each component and NERS value and decision makers can choose the proper ones based their opinions. There are similar conclusions for n=8.

Comparing effectiveness of solution methods

The NERS measure is used to evaluate the performance of the solution methods and also to show the effectiveness of the proposed method. A single factor ANOVA is used to find out whether there is a significant difference among the performance of solution methods. Data are being normal and have equal variance (tested by MINITAB14 software). ANOVA results are presented in Table 8. These results with p-value = 0.003 confirm that there is at least one method with different mean response when confidence level is set at 0.95. Thus, Fisher's least significant difference method is used to compare the performance of methods (see Table 9). The results confirm that there is a significant difference between PRM and CA-Ri and CA-Re. Also, Table 9 shows no significant difference between CA-Ri and CA-Re.

As Figure. 5 illustrates, among of the solution methods, proactive-reactive method for NERS value is better than both classical approach with right shifting and classical approach with regeneration. Therefore, the superiority of the proposed solution approach is concluded over other heuristics including CA-Ri and CA-Re due to the computational results.

Table 8: ANOVA results for solution methods

Source	**df**	**SS**	**MS**	**F**	**P-value**
Algorithm	2	0.09347	0.04674	8.07	0.003
Error	18	0.10426	0.00579		
Total	20	0.19773			

Table 9: Fisher 95% individual confidence intervals all pair-wise comparisons

Algorithms	**Lower**	**Center**	**Upper**	**Significant difference at 95% level**
PRM vs. CA-Ri	0.057	0.143	0.228	Yes
PRM vs. CA-Re	0.055	0.141	0.226	Yes
CA-Ri vs. CA-Re	-0.087	-0.002	0.083	No

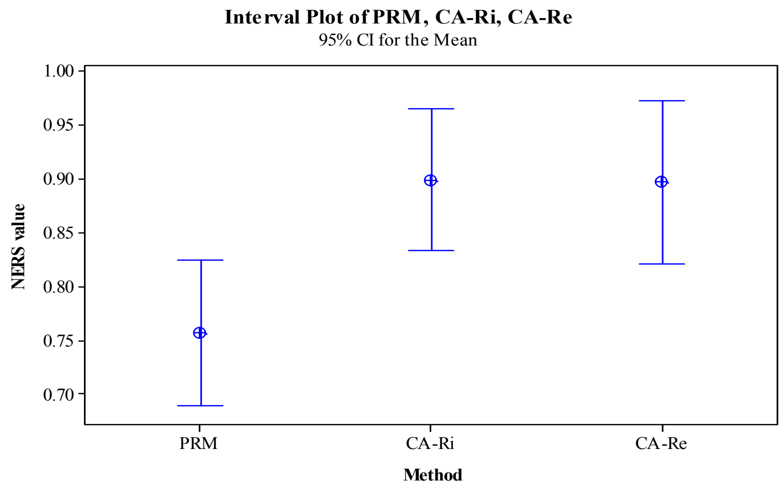

Figure 5: Means and interval plot for NERS value

5. Conclusions

In this paper, a proactive-reactive approach was presented instead of SMs. This two stage approach produced robust and stable solutions for a two-machine scheduling problem in a flow shop environment. In this approach, firstly by considering uncertain processing times and using the robust optimization approach, the problem was solved and a robust initial solution was proactively produced. Then in case of machine failure, the appropriate reaction was adopted based on the defined performance measure. This measure is a multi-criteria measure defined in terms of solution effectiveness, robustness, stability, and reduction of system nervousness. Computational results indicated that this method is much more effective, compared to the ordinary scheduling methods that operate on the basis of makespan alone. The results showed efficiency of the proposed approach compared to two common classical methods in the face of unexpected failures. For future researches, this problem can be considered for other scheduling problems of shop floor, and other classical objectives can be used to evaluate this method. As another research subject, random distributions can be considered for the time of failure occurrence and for repair duration, because they can be used for obtaining initial solutions that are proactively able to reduce the effect of machine breakdown. Moreover, the effect of this approach on other random disruptions such as the arrival of new jobs, and order cancellations etc. can be investigated and analyzed. Another important future research could be the presentation of proper heuristic methods for machine failure, which can be applied in the second step of this approach.

References

Al-Hinai N, ElMekkawy TY (2011) Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm. *International Journal of Production Economics* 132: 279–291

Al-Hinai N, ElMekkawy TY (2011) An efficient hybridized genetic algorithm architecture for the flexible job-shop scheduling problem. *Flexible Services and Manufacturing Journal* 23: 64–85

Beck, J. C. (2007). Solution-guided multi-point constructive search for job shop scheduling. *Journal of Artificial Intelligence Research,* 29, 49-77.

Beck, J. C., & Wilson, N. (2007). Proactive algorithms for job shop scheduling with probabilistic durations. *Journal of Artificial Intelligence Research* , 28 , 183-232.

Bouyahia Z, Bellalouna M, Jaillet P, Ghedira K (2010) A priori parallel machines scheduling. *Computers & Industrial Engineering* 58(3): 488‑500

Braun O, Lai TC, Schmidt G Sotskov YN (2002) Stability of Johnson's schedule with respect to limited machine availability. *International Journal of Production Research* 40(17): 4381–4400

Daniels R Kouvelis P (1995) *Robust scheduling to hedge against processing time uncertainty in single stage production. Management Science* 41(2): 363-376

Fattahi P, Hosseini SMH, Jolai F (2012) A mathematical model and extension algorithm for assembly flexible flow shop scheduling problem. *International Journal of advanced manufacturing technology* 65(5-8): (787-802)

Gan, H. -S. and Wirth, A. (2005) Comparing deterministic, robust and online scheduling using entropy, *International Journal of Production Research* 43(10): 2113-2134

Ghezail F, Pierreval H, Hajri-Gabouj S (2010) Analysis of robustness in proactive scheduling: A graphical approach. *Computers & Industrial Engineering* 58:193–198

Goren S, Sabuncuoglu I (2008) Robustness and stability measures for scheduling: single machine environment. *IIE Transactions* 40(1): 66 – 83

Guo B, Nonaka Y (1999) Rescheduling and optimization of schedules considering machine failure. *International Journal of Production Economics* 60–61: 503–513

Hwang CL, Yoon K (1981) Multiple Attribute Decision Making: Methods and Applications, *Springer-Verlag, Berlin*.

Jensen MT (2001b) Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measure. *Applied Soft Computing* 1: 35–52

Jensen MT (2003) Generating robust and flexible job shop schedules using genetic algorithms. *IEEE Transactions on Evolutionary Computation* 7 (3): 275–288

Johnson SM (1954) Optimal two- and three-stage production schedules with set-up times included*, Naval Research Logistics Quarterly* 1: 61-68

Kouvelis P, Kurawarwala AA, Gutierrez GJ (1992) Algorithms for robust single- and multiple-period layout planning for manufacturing systems. *European Journal of Operational Research* 63: 287-303

Kutanoglu E, Sabuncuoglu I (2001) Experimental investigation of iterative simulation-based scheduling in a dynamic and stochastic job shop. *Journal of Manufacturing Systems* 20(4): 264–279

Kuchta D (2011) A concept of a robust solution of a multicriterial linear programming problem. *Central European Journal of Operations Research* 19:605–613

Lambrechts O, Demeulemeester E, Herroelen, W (2008) A tabu search procedure for developing robust predictive project schedules. *International Journal of Production Economics* 111: 493–508

Lawrence SR, Sewell EC (1997) Heuristic, optimal, static, and dynamic schedules when processing times are uncertain. *Journal of Operations Management* 15: 71–82

Leon VJ, Wu SD, Storer RH (1994) Robustness measures and robust scheduling for job shops. *IIE Transactions* 26(5): 32–43

Leung S C H, Wu Y (2004) A robust optimization model for stochastic aggregate production planning. *Production Planning & Control* 15(5): 502–514

Mulvey JM, Vanderbei RJ Zenios SA (1995) Robust optimization of large-scale systems. *Operations Research*, 43: 264- 281

O'Donovan R, Uzsoy R, McKay KN (1999) Predictable scheduling of a single machine with breakdowns and sensitive jobs. *International Journal of Production Research* 37(18): 4217–4233

Ouelhadj D, Petrovic S. (2009) A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling* 12: 417-43

Pinedo ML (2008) Scheduling: Theory, Algorithms, and Systems, Springer, 3rd Edition

Rahmani, D. & Heydari, M. (2014). Robust and stable flow shop scheduling with unexpected arrivals of new jobs and uncertain processing times. *Journal of Manufacturing Systems*, 33, 1, 84–92.

Rangsaritratsamee R, Ferrell WG, Kurtz MB (2004) Dynamic rescheduling that simultaneously considers efficiency and stability. *Computers & Industrial Engineering* 46 (1): 1–15

Rossi A (2010) A robustness measure of the configureuration of multi-purpose machines, *International Journal of Production Research*. 48(4): 1013-1033

Sabuncuoglu I, Karabuk S (1999) Rescheduling frequency in an FMS with uncertain processing times and unreliable machines. *Journal of Manufacturing* 18 (4): 268–283

Sabuncuoglu I, Goren S (2009) Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. *International Journal of Computer Integrated Manufacturing* 22 (2): 138–157

Sotskov YN Lai TC (2012) Minimizing total weighted flow time under uncertainty using dominance and a stability box. *Computers & Operations Research* 39(6): 1271–1289

SotskovYN, Egorova NG, Lai TC (2009) Minimizing total weighted flow time of a set of jobs with interval processing times. *Mathematical and Computer Modelling*. 50: 556–73

Sevaux M and Sorensen K (2004) A genetic algorithm for robust schedules in a one- machine environment with ready times and due dates. *Quarterly Journal of the Belgian, French and Italian Operations Research* Societies 4 (2): 129–147

Terekhov, D., Tran, T. T., Down, D. G. & Beck, J. C. (2014). Integrating Queueing Theory and Scheduling for Dynamic Scheduling Problems. *Journal of Artificial Intelligence Research* 50, 535-572

Wu SD, Storer RN, Chang P (1993) One-machine rescheduling heuristics with efficiency and stability as criteria. *Computers & Operations Research* 20(1): 1–14

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